
Unterschrift des Betreuers



TECHNISCHE
UNIVERSITÄT
WIEN
Vienna University of Technology

DIPLOMARBEIT

Numerische Berechnung des magnetischen Streufelds

Ausgeführt am Institut für
Analysis und Scientific Computing
der Technischen Universität Wien

unter der Anleitung von
Ao.Univ.Prof. Dipl.-Math. Dr.techn. Dirk Praetorius

durch

Thomas Führer
Rottenhauserstrasse 28
3250 Wieselburg

Datum

Unterschrift

Inhaltsverzeichnis

Kurzfassung	iii
Danksagung	iv
1 Einleitung	1
1.1 Magnetostatische Maxwellgleichungen	1
1.2 Streufeld und skalares Potential	2
2 Analytische Grundlagen	6
2.1 Sobolev-Räume	6
2.2 Schwache Formulierung	9
2.3 Integraloperatoren	11
2.4 Eindeutige Lösbarkeit	15
3 Numerische Verfahren	18
3.1 Finite Elemente Methode	18
3.1.1 Galerkin Verfahren	18
3.1.2 P1-Finite Elemente Methode	19
3.2 Interpolationsoperatoren	29
4 Ansätze zur Lösung des magnetostatischen Problems	34
4.1 Ansatz von García-Cervera & Roma	35
4.1.1 Schwache Formulierung	35
4.1.2 Numerische Umsetzung	37
4.1.3 A priori Fehleranalysis	38
4.2 Ansatz von Fredkin & Koehler	39
4.2.1 Schwache Formulierung	39
4.2.2 Numerische Umsetzung	41
4.2.3 A priori Fehleranalysis	42
4.3 Symmetrische FEM/BEM-Kopplungsmethode	43
4.3.1 Formulierung	43
4.3.2 Numerische Umsetzung	44

4.3.3	A priori Analysis	44
4.4	Direkte FEM/BEM - Kopplungsmethode	44
4.4.1	Formulierung	45
4.4.2	Numerische Umsetzung	45
4.4.3	A priori Analysis	46
5	Analytische Lösung	47
5.1	Zweidimensionaler Fall	48
5.2	Dreidimensionaler Fall	57
6	Implementierung in 2D	63
6.1	Datenstrukturen	63
6.2	García-Cervera & Roma	64
6.3	Fredkin & Koehler	70
6.4	Symmetrische FEM/BEM-Kopplung	77
6.5	Direkte FEM/BEM-Kopplung	81
7	Numerische Resultate in 2D	85
7.1	Kreisscheibe	85
7.2	Weitere Gebiete	89
7.3	Zusammenfassung	103
	Anhang	104
A	Notationen	104
B	Quellcodes	106
	Literaturverzeichnis	117

Kurzfassung

Diese Arbeit handelt von der numerischen Berechnung des magnetischen Streufelds, welche in vielen Bereichen zur Anwendung kommt. Beispielsweise tritt das Streufeld in der Landau-Lifschitz-Gilbert Gleichung auf, die im Themengebiet Mikromagnetismus eine tragende Rolle spielt. Diese Gleichung wird unter anderem dazu benützt, um Simulationen von magnetischen Bauteilen, wie etwa magnetischen Speichermedien oder magnetischen Sensoren, durchzuführen.

Wir werden in dieser Arbeit vier verschiedene Methoden vorstellen, wie das Streufeld berechnet werden kann. Diese Verfahren unterteilen sich dabei in zwei unterschiedliche Typen. Auf der einen Seite stehen die Ansätze von FREDKIN / KOEHLER 1990 [8] bzw. GARCÍA-CERVERA / ROMA 2005 [10], welche auch hybride FEM/BEM Kopplungen genannt werden, auf der anderen Seite stehen die symmetrische (COSTABEL 1988 [5]) und direkte (JOHNSON / NÉDÉLEC 1980 [14]) FEM/BEM - Kopplungsmethode. Wir werden diese Verfahren analysieren und auch die numerische Umsetzung betrachten. Die getroffenen Aussagen gelten sowohl für den zwei- als auch für den dreidimensionalen Fall. Zum Abschluss führen wir noch einige numerische Experimente durch, welche die Vor- und Nachteile der einzelnen Verfahren zum Vorschein bringen sollen. Hierbei werden wir uns auf den zweidimensionalen Fall beschränken.

Die Arbeit ist wie folgt aufgebaut:

- Zunächst werden wir in *Kapitel 1* die zur Berechnung des Streufelds zu Grunde liegenden Gleichungen aufstellen und diese auf die Bestimmung des magnetostatischen Potentials zurückführen. Als Ausgangspunkt dienen dabei die Maxwell'schen Gleichungen.
- In *Kapitel 2* legen wir das mathematische Fundament um die zu lösenden Gleichungen, damit wir Aussagen über die Existenz und Eindeutigkeit von Lösungen treffen können.
- *Kapitel 3* beschäftigt sich dann mit der Finiten Elemente Methode. Weiters führen wir Interpolationsoperatoren ein, die im späteren Verlauf für die Analysis Bedeutung finden.
- Die zuvor erwähnten Ansätze werden in *Kapitel 4* untersucht. Wir behandeln dabei sowohl Aspekte zur Implementierung, als auch die a priori Analysis.
- Im Allgemeinen können die Gleichungen, welche das magnetostatische Potential bestimmen, nicht exakt gelöst werden. Für spezielle Geometrien können jedoch analytische Lösungen berechnet werden. In *Kapitel 5* lösen wir die Gleichungen auf einer Kreisscheibe bzw. Kugel. Diese Lösungen verwenden wir später dazu, um die Implementierung zu verifizieren.
- In *Kapitel 6* beschreiben wir die Implementierung der verschiedenen Ansätze in Matlab für den zweidimensionalen Fall.
- Zuletzt führen wir in *Kapitel 7* verschiedene Experimente mit den einzelnen Methoden durch.

Danksagung

An dieser Stelle möchte ich mich ganz herzlichst bei meinem Betreuer Professor Dirk Praetorius bedanken, der immer Zeit gefunden hat, sich mit meinen Problemen auseinander zusetzen und mit zahlreichen Ideen diese Arbeit prägend mitgestaltet hat. Darüber hinaus hat er mir ermöglicht bei dem WWTF Projekt *Micromagnetic Simulation and Computational Design of Future Devices (WWTF Project MA09-029)* mitzuarbeiten, welches mich auch finanziell unterstützt hat. Weiters möchte ich mich bei DI Petra Goldenits bedanken, die durch zahlreiches Korrekturlesen dieser Arbeit und viele Anregungen zur wesentlichen Verbesserung derselben beigetragen hat. Außerdem sei noch der ganzen Arbeitsgruppe rund um Professor Praetorius gedankt. Die letzten Monate waren eine schöne und hilfreiche Erfahrung.

Des weiteren möchte ich allen meinen Freunden und Studienkollegen, die mich in den letzten Jahre durchs Studium begleitet haben, meinen Dank aussprechen.

Zu guter Letzt möchte ich noch meiner ganzen Familie danken, allen voran meinen Eltern Christa und Friedrich Führer, die mich während meiner Studienzeit immer unterstützt haben.

Kapitel 1

Einleitung

1.1 Magnetostatische Maxwellgleichungen

Ausgangspunkt zur Herleitung der Gleichungen des magnetostatischen Potentials stellen die Maxwell'schen Gleichungen der Magnetostatik dar. Diese Gleichungen lauten in SI-Einheiten

$$\operatorname{div} \mathbf{B} = 0, \quad (1.1)$$

$$\operatorname{rot} \mathbf{B} = \mu_0 \mathbf{J}, \quad (1.2)$$

wobei wir mit $\mathbf{B}$ die magnetische Flussdichte (bzw. Induktion, in T), mit $\mu_0 = 4\pi 10^{-7} \text{Vs/Am}$ die Permeabilität von Vakuum und mit $\mathbf{J}$ (in A/m^2) die Stromdichte bezeichnen.

Bei Anwesenheit von Materie muss noch zwischen freien und gebundenen Strömen im Körper unterschieden werden. Dazu benötigen wir die Größe $\mathbf{J}_{frei}$, welche die Stromdichte beschreibt die von freien Ladungsträgern im Material herrührt. Des weiteren führen wir noch die Stromdichte $\mathbf{J}_{geb}$ der im Material gebundenen Ströme ein. Diese gemittelte Größe erzeugt die Magnetisierung des Materials gemäß

$$\operatorname{rot} \mathbf{M} = \mathbf{J}_{geb}. \quad (1.3)$$

Außerhalb des Körpers verschwindet die Magnetisierung $\mathbf{M}$. Nach JACKSON 2006 [13, Abschnitt 5.8] ist die Magnetisierung die mittlere magnetische Momentdichte, d.h.

$$\mathbf{M}(\mathbf{x}) = \sum_i N_i \langle \mathbf{m}_i \rangle, \quad (1.4)$$

wobei N_i die mittlere Anzahl von Atomen bzw. Molekülen der Sorte i pro Volumen, und $\langle \mathbf{m}_i \rangle$ das mittlere magnetische Moment in einem Volumen um die Stelle $\mathbf{x}$ ist.

Das Gesamtfeld $\mathbf{B}$ erfüllt die Gleichungen

$$\begin{aligned} \operatorname{div} \mathbf{B} &= 0, \\ \operatorname{rot} \mathbf{B} &= \mu_0 (\mathbf{J}_{frei} + \mathbf{J}_{geb}). \end{aligned} \quad (1.5)$$

Üblicherweise definiert man ein zusätzliches Feld $\mathbf{H}$, welches magnetische Feldstärke bzw. magnetische Erregung genannt wird, über

$$\mathbf{B} = \mu_0 (\mathbf{H} + \mathbf{M}). \quad (1.6)$$

Durch Einsetzen in Gleichung 1.5 erhält man zunächst

$$\operatorname{rot} \mathbf{H} + \operatorname{rot} \mathbf{M} = \mathbf{J}_{frei} + \mathbf{J}_{geb}, \quad (1.7)$$

und nach Ausnützen von (1.3) bekommt man das Gleichungssystem

$$\begin{aligned}\operatorname{div} \mathbf{B} &= 0, \\ \operatorname{rot} \mathbf{H} &= \mathbf{J}_{\text{frei}}.\end{aligned}\tag{1.8}$$

Betrachtet man nun zwei Materialien mit verschiedenen magnetischen Eigenschaften, so lassen sich aus den Maxwell'schen Gleichungen Übergangsbedingungen an der Grenzfläche beider Medien herleiten. Seien Ω_1, Ω_2 zwei Gebiete und Γ die Grenzfläche. Mit $\mathbf{n}$ bezeichnen wir den von Ω_1 nach Ω_2 gerichteten Normalenvektor auf Γ . Dann lauten die Übergangsbedingungen nach JACKSON 2006 [13, Abschnitt I.5]

$$\begin{aligned}(\mathbf{B}_2 - \mathbf{B}_1) \cdot \mathbf{n} &= 0, \\ (\mathbf{H}_2 - \mathbf{H}_1) \times \mathbf{n} &= \mathbf{k},\end{aligned}$$

auf der Grenzfläche Γ , wobei $\mathbf{k}$ eine Flächenstromdichte beschreibt und die Felder $\mathbf{B}_i, \mathbf{H}_i$ mit $i = 1, 2$ die Einschränkung der Felder von Ω_i auf Γ sind. Mit (1.6) ergibt sich nun

$$\begin{aligned}(\mathbf{H}_2 - \mathbf{H}_1) \cdot \mathbf{n} &= (\mathbf{M}_1 - \mathbf{M}_2) \cdot \mathbf{n}, \\ (\mathbf{H}_2 - \mathbf{H}_1) \times \mathbf{n} &= \mathbf{k}.\end{aligned}\tag{1.9}$$

1.2 Streufeld und skales Potential

Wir betrachten nun einen Körper, der durch das beschränkte Gebiet Ω beschrieben wird und nehmen an, die Magnetisierung ist stetig differenzierbar auf diesem Gebiet und verschwindet außerhalb. Zusätzlich nehmen wir $\mathbf{J}_{\text{frei}} = 0$ an, d.h. es fließt kein Strom, der von freien Ladungsträgern erzeugt wird. Dann folgt aus dem Gleichungssystem (1.8) und Gleichung (1.6), für das Feld $\mathbf{H}$ in Ω

$$\begin{aligned}\operatorname{div} \mathbf{H} &= -\operatorname{div} \mathbf{M}, \\ \operatorname{rot} \mathbf{H} &= 0.\end{aligned}\tag{1.10}$$

Außerhalb des Körpers gilt, da die Magnetisierung verschwindet,

$$\begin{aligned}\operatorname{div} \mathbf{H} &= 0, \\ \operatorname{rot} \mathbf{H} &= 0.\end{aligned}\tag{1.11}$$

Wir fügen dem Feld $\mathbf{H}$ den Index $(\cdot)_s$ hinzu und nennen $\mathbf{H}_s$ das Streufeld oder Demagnetisierungsfeld. Aus den Übergangsbedingungen (1.9) lässt sich folgern, dass die Normalkomponente von $\mathbf{H}_s$ am Rand einen Sprung besitzt und die Tangentialkomponente stetig ist, d.h. es gilt

$$\begin{aligned}(\mathbf{H}_s^{\text{ext}} - \mathbf{H}_s^{\text{int}}) \cdot \mathbf{n} &= \mathbf{M} \cdot \mathbf{n}, \\ (\mathbf{H}_s^{\text{ext}} - \mathbf{H}_s^{\text{int}}) \times \mathbf{n} &= 0\end{aligned}\tag{1.12}$$

auf Γ .

Aus $\operatorname{rot} \mathbf{H}_s = 0$ folgt, dass man das Streufeld $\mathbf{H}_s$ als Gradienten eines Potentials darstellen kann, d.h.

$$\mathbf{H}_s = -\nabla u,\tag{1.13}$$

und man nennt u auch skales magnetostatisches Potential. Die Bedingung, dass das Streufeld stetig in der Tangentialkomponente am Rand ist, wird sichergestellt durch die Forderung der Stetigkeit des Potentials am Rand. Um u bestimmen zu können benötigen wir noch Informationen über das Abklingverhalten des Potentials im Unendlichen. Auf die Abklingbedingungen wird im nächsten Kapitel näher eingegangen.

Wir bezeichnen im Folgenden mit Ω ein beschränktes Gebiet in $\mathbb{R}^d$ mit $d = 2, 3$, mit $\Omega^{\text{ext}} := \mathbb{R}^d \setminus \overline{\Omega}$ den Außenraum und mit $\mathbf{n}$ den Normalenvektor auf $\Gamma := \partial\Omega$ (siehe Abbildung 1.1). Die Normalenableitung einer Funktion u wird als $\partial u / \partial \mathbf{n} = \sum_i \partial u / \partial x_i \mathbf{n}_i$ geschrieben.

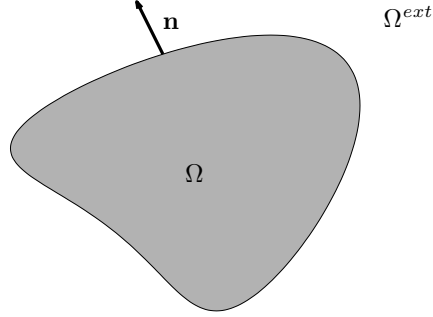


Abbildung 1.1: Aufteilung des $\mathbb{R}^2$ in das beschränkte Gebiet Ω und in den Außenraum Ω^{ext} . Der Normalenvektor $\mathbf{n}$ zeigt von Ω nach Ω^{ext} .

Um die Existenz und Eindeutigkeit der Lösung zu gewährleisten, muss man noch das asymptotische Verhalten der Lösung im Unendlichen angeben. Im zweidimensionalen Fall setzen wir

$$u(\mathbf{x}) = a + b \log \|\mathbf{x}\| + \mathcal{O}(1/\|\mathbf{x}\|) \quad \text{für } \|\mathbf{x}\| \rightarrow \infty$$

voraus. Man kann entweder $a \in \mathbb{R}$ oder $b \in \mathbb{R}$ frei wählen und den anderen Wert bestimmen. In dieser Arbeit wählen wir $a = 0$. In Kapitel 4 werden wir den Wert b bestimmen. Im dreidimensionalen Fall nehmen wir für die Abklingbedingung, i.e. das asymptotische Verhalten der Lösung,

$$u(\mathbf{x}) = c + \mathcal{O}(1/\|\mathbf{x}\|) \quad \text{für } \|\mathbf{x}\| \rightarrow \infty.$$

Hierbei kann die Konstante $c \in \mathbb{R}$ frei gewählt werden. Wir wählen in dieser Arbeit $c = 0$.

Die Abklingbedingungen geben also gewissermaßen an, dass sich u asymptotisch wie die Fundamentallösung des Laplace Operators verhält. Es sei weiters angemerkt, dass man sich üblicherweise für das Streufeld, also dem negativen Gradienten von u , interessiert, weshalb die Wahl von a und c im Grunde genommen keine Rolle spielt.

Die das Potential bestimmenden Gleichungen können nun folgendermaßen zusammengefasst werden.

Definition 1.1 (Klassische Formulierung). Sei $\Omega \in \mathbb{R}^d$ ein beschränktes Gebiet mit $d = 2$ oder $d = 3$ und $\Gamma = \partial\Omega$ dessen Rand. Mit der Magnetisierung des Gebietes $\mathbf{M} \in (C^1(\Omega))^d \cap (C^0(\overline{\Omega}))^d$ ist dann das magnetostatische Potential $u \in C^2(\mathbb{R}^d)$ die Lösung des folgenden Transmissionsproblems:

$$\begin{aligned} \Delta u &= \operatorname{div} \mathbf{M} && \text{in } \Omega, \\ \Delta u &= 0 && \text{in } \Omega^{ext}, \\ [u] &= 0 && \text{auf } \Gamma, \\ \left[\frac{\partial u}{\partial \mathbf{n}} \right] &= -\mathbf{M} \cdot \mathbf{n} && \text{auf } \Gamma, \end{aligned} \tag{1.14}$$

$$u(\mathbf{x}) = \begin{cases} b \log \|\mathbf{x}\| + \mathcal{O}\left(\frac{1}{\|\mathbf{x}\|}\right) & \text{falls } d = 2, \\ \mathcal{O}\left(\frac{1}{\|\mathbf{x}\|}\right) & \text{falls } d = 3, \end{cases} \quad \text{für } \|\mathbf{x}\| \rightarrow \infty.$$

Dabei ist $\|\mathbf{x}\| := \sqrt{\sum_{i=1}^d x_i^2}$ die Länge von $\mathbf{x}$ und $[u] := u^{ext} - u^{int}$ der Sprung der Funktion u am Rand.

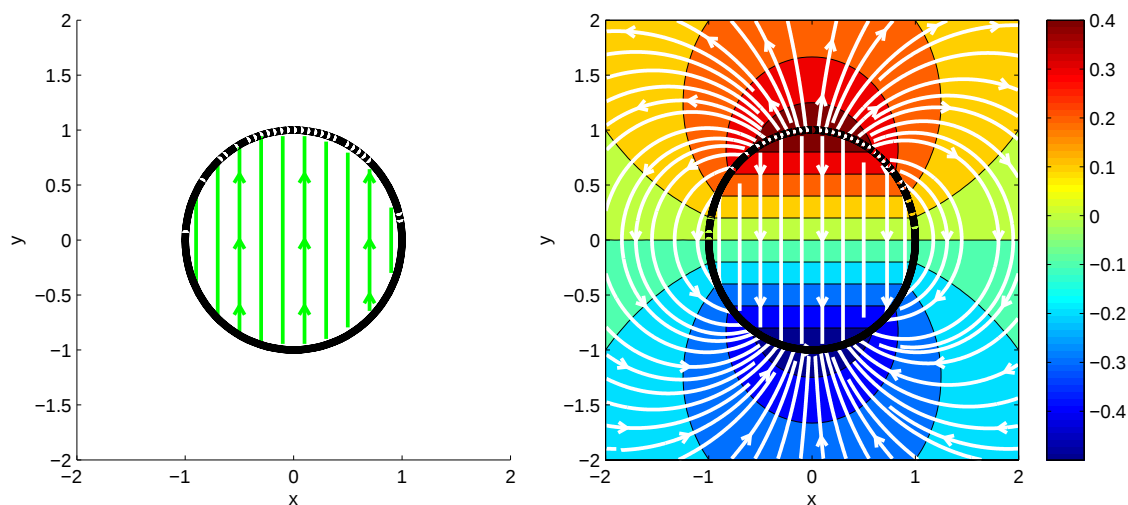


Abbildung 1.2: Das linke Bild zeigt die Magnetisierung einer homogen magnetisierten Kreisscheibe $\Omega = \{\mathbf{x} \in \mathbb{R}^2 \mid \|\mathbf{x}\| < 1\}$ mit $\mathbf{M} = (0, 1)^T$. Die schwarze Umrandung stellt die Begrenzung des magnetischen Körpers dar. Die weißen Linien im rechten Bild skizzieren die Feldlinien des Streufelds. Der Hintergrund wurde mit Bezug auf das Potential eingefärbt, d.h., dass Flächen gleicher Farbe, gleichen Potentialwerten entsprechen.

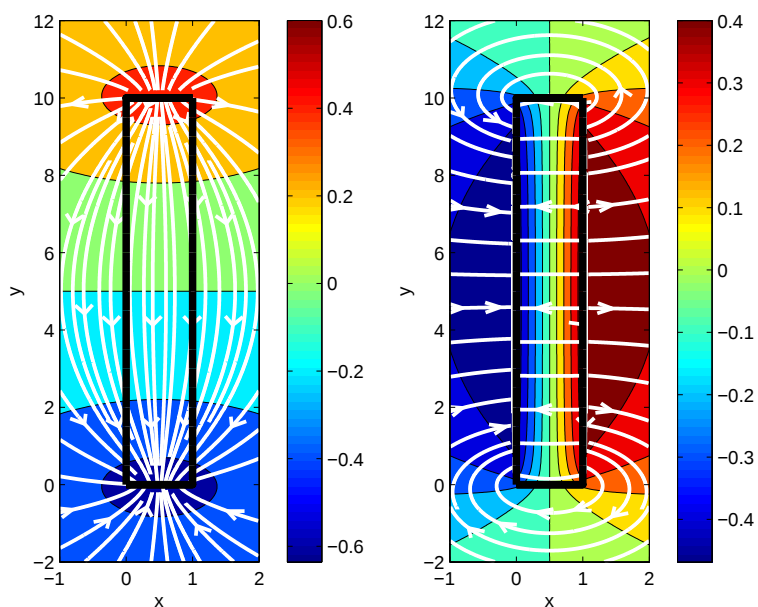


Abbildung 1.3: Die beiden Bilder zeigen jeweils die Feldlinien des Streufelds eines homogen magnetisierten rechteckigen Körpers $\Omega = \{(x, y) \in \mathbb{R}^2 \mid 0 < x < 1, 0 < y < 10\}$. Für die Magnetisierung im linken Bild gilt $\mathbf{M} = (0, 1)^T$ für die im rechten $\mathbf{M} = (1, 0)^T$.

Die Gleichungen in Definition 1.1 sind ähnlich der Potentialgleichungen in der Elektrostatik und man kann daher den Term $(-\operatorname{div} \mathbf{M})$ als magnetische Ladungsdichte interpretieren. Dies bedeutet, dass die Divergenz der Magnetisierung die Quelle des Streufelds darstellt. Weiters kann man die Normalkomponente $\mathbf{M} \cdot \mathbf{n}$ der Magnetisierung am Rand als magnetische Flächenladungsdichte deuten. Angenommen die Divergenz der Magnetisierung verschwindet im Inneren, dann wird das Demagnetisierungsfeld durch die Werte von $\mathbf{M} \cdot \mathbf{n}$ am Rand bestimmt.

Das Streufeld lässt sich für gewisse Gebiete, z.B. eine Kreisscheibe oder eine Kugel, exakt berechnen (eine genauere Auseinandersetzung folgt in Kapitel 5). Das linke Bild in Abbildung 1.2 zeigt eine Kreisscheibe die homogen magnetisiert ist. Die Magnetisierung wird durch den konstanten Vektor $\mathbf{M} = (0, 1)^T$ beschrieben und das Potential des Streufelds im Inneren der Kugel ist dann gegeben durch

$$u(x, y, z) = \frac{1}{2}y \quad \text{in } \Omega.$$

Im rechten Bild von Abbildung 1.2 sind die Feldlinien des Streufelds durch weiße Linien visualisiert. Man erkennt, dass das Streufeld im Inneren entgegengesetzt zur Magnetisierung ist. Das Gesamtfeld im Inneren der Kreisscheibe ergibt sich nun zu

$$\mathbf{B} = \mu_0(\mathbf{H} + \mathbf{M}) = \mu_0 \begin{pmatrix} 0 \\ -\frac{1}{2} + 1 \\ 0 \end{pmatrix} = \mu_0 \begin{pmatrix} 0 \\ \frac{1}{2} \\ 0 \end{pmatrix}.$$

Wir betrachten noch ein weiteres Beispiel. Abbildung 1.3 zeigt zwei rechteckige Körper, die homogen magnetisiert sind. Für die Magnetisierung des Körpers im linken Bild wurde $\mathbf{M} = (0, 1)^T$ gewählt, für die Magnetisierung im rechten Bild $\mathbf{M} = (1, 0)^T$. Man sieht, dass das Streufeld im Inneren nicht homogen ist, wie das z.B. im Fall der Kreisscheibe ist.

Kapitel 2

Analytische Grundlagen

In diesem Kapitel stellen wir für diese Arbeit grundlegende Ergebnisse aus der Analysis vor. Der erste Abschnitt beschäftigt sich mit den für die Lösungstheorie partieller Differentialgleichungen wichtigen Sobolev-Räumen, wobei wir uns an die Arbeiten von EVANS 2002 [6] und SAUTER / SCHWAB 2004 [18] halten. Danach stellen wir die schwache Formulierung des Transmissionsproblems bereit. Im dritten Abschnitt behandeln wir dann Integraloperatoren und deren Eigenschaften, dabei werden die wichtigsten Resultate aus SAUTER / SCHWAB 2004 [18], PRAETORIUS 2007 [16] und STEINBACH 2003 [20] entnommen. Zuletzt fassen wir noch Existenz- und Eindeutigkeitsaussagen zusammen, die wir in späterer Folge noch verwenden werden. Hierbei sei auf die Arbeit von CARSTENSEN / STEPHAN 1995 [4] verwiesen.

2.1 Sobolev-Räume

In diesem Abschnitt führen wir den Begriff der Sobolev-Räume ein und werden fundamentale Ergebnisse aus der oben erwähnten Literatur zitieren. Diese Räume sind essentiell für die Lösungstheorie von partiellen Differentialgleichungen, da mit ihnen Resultate über Existenz und Eindeutigkeit von Lösungen erzielt werden können.

Konvention 2.1. Im Folgenden bezeichnen wir mit $\Omega \subseteq \mathbb{R}^d$ ($d = 2, 3$) ein beschränktes Lipschitz-Gebiet mit Rand $\Gamma := \partial\Omega$. Das Komplement von Ω bezeichnen wir mit $\Omega^{ext} := \mathbb{R}^d \setminus \Omega$.

Konvention 2.2. Sei $U \subseteq \mathbb{R}^d$ eine offene Menge. Sei $C^\infty(U)$ der Raum der unendlich oft differenzierbaren Funktionen auf U , dann bezeichnet

$$C_0^\infty(U) := \{v \in C^\infty(U) \mid \text{supp}(v) \text{ ist kompakt in } U\}$$

den Raum aller unendlich oft differenzierbaren Funktionen mit kompakten Träger in U , wobei der Träger einer stetigen Funktion $v \in C^0(U)$ definiert ist als

$$\text{supp}(v) := \overline{\{x \in U \mid v(x) \neq 0\}}.$$

Definition 2.3. Sei $U \subseteq \mathbb{R}^d$ eine offene Menge. Mit

$$L_{lok}^1(U) := \left\{ v : U \rightarrow \mathbb{R} \text{ Lebesgue-messbar} \mid \int_K |v(x)| dx < \infty \text{ für alle } K \subseteq U \text{ mit } K \text{ kompakt} \right\} \quad (2.1)$$

bezeichnen wir den Raum aller lokal integrierbaren Funktionen.

Definition 2.4. Sei $U \subseteq \mathbb{R}^d$ eine offene Menge, α ein Multiindex und $u \in L_{lok}^1(U)$. Existiert ein $v \in L_{lok}^1(U)$ mit

$$\int_U u \partial^\alpha \phi dx = (-1)^{|\alpha|} \int_U v \phi dx \quad \text{für alle } \phi \in C_0^\infty(U), \quad (2.2)$$

dann heißt v die α -te schwache Ableitung von u und wir schreiben auch $\partial^\alpha u := v$.

Lemma 2.1 (EVANS 2002 [6, Abschnitt 5.2, Lemma]). Sei $U \subseteq \mathbb{R}^d$ eine offene Menge. Existiert die schwache Ableitung einer Funktion $u \in L^1_{lok}(U)$, dann ist sie eindeutig bestimmt bis auf eine Menge mit Maß 0.

Bemerkung 2.1. Für stetig differenzierbare Funktionen stimmt die schwache Ableitung mit der starken Ableitung (fast überall) überein.

Definition 2.5. Für $\ell \in \mathbb{N}_0$ ist der Sobolev-Raum $H^\ell(\Omega)$ definiert als

$$H^\ell(\Omega) := \{v \in L^2(\Omega) \mid \partial^\alpha v \in L^2(\Omega) \text{ für alle Multiindizes } \alpha \text{ mit } |\alpha| \leq \ell\}. \quad (2.3)$$

Der Raum $H^\ell(\Omega)$ wird mit dem Skalarprodukt

$$(u, v)_{H^\ell(\Omega)} := \sum_{|\alpha| \leq \ell} (\partial^\alpha u, \partial^\alpha v)_{L^2(\Omega)} \quad (2.4)$$

versehen. Die assoziierte Norm ist dann gegeben durch $\|u\|_{H^\ell(\Omega)} := \sqrt{(u, u)_{H^\ell(\Omega)}}$.

Satz 2.2 (EVANS 2002 [6, Abschnitt 5.2, Satz 2]). $H^\ell(\Omega)$ ist für $\ell \in \mathbb{N}_0$ ein Hilbertraum.

Durch $|u|_{H^\ell(\Omega)} := \sum_{|\alpha|=\ell} (\partial^\alpha u, \partial^\alpha v)_{L^2(\Omega)}$ wird der Sobolev-Raum $H^\ell(\Omega)$ mit einer Seminorm versehen.

Definition 2.6. $H^\ell_0(\Omega)$ wird als Abschluss von $C^\infty_0(\Omega)$ bezüglich der $\|\cdot\|_{H^\ell(\Omega)}$ -Norm definiert, d.h.

$$H^\ell_0(\Omega) := \overline{C^\infty_0(\Omega)}^{\|\cdot\|_{H^\ell(\Omega)}}.$$

Definition 2.7. Sei $U \subseteq \mathbb{R}^d$ ein Gebiet. Der Raum

$$H^\ell_{lok}(U) := \{u : U \rightarrow \mathbb{R} \text{ Lebesgue-messbar} \mid u \in H^\ell(K) \text{ für alle } K \subseteq \bar{U} \text{ mit } K \text{ kompakt}\} \quad (2.5)$$

bezeichnet den Raum aller messbaren Funktionen die auf jedem Kompaktum in H^ℓ liegen.

Des weiteren wollen wir noch einen speziellen Sobolev-Raum auf dem Rand Γ des Gebiets einführen. Dieser lässt sich mit Hilfe der Sobolev-Slobodeckij-Seminorm

$$|u|_{H^{1/2}(\Gamma)} := \left(\int_\Gamma \int_\Gamma \frac{|u(x) - u(y)|^2}{\|x - y\|^d} ds_y ds_x \right)^{1/2},$$

als Sobolev-Raum mit halbzahligem Exponenten definieren.

Definition 2.8. Mit der Norm $\|u\|_{H^{1/2}(\Gamma)} := \left(\|u\|_{L^2(\Gamma)}^2 + |u|_{H^{1/2}(\Gamma)}^2 \right)^{1/2}$ ist der Raum $H^{1/2}(\Gamma)$ definiert als

$$H^{1/2}(\Gamma) := \{u \in L^2(\Gamma) \mid \|u\|_{H^{1/2}(\Gamma)} < \infty\}.$$

Im Allgemeinen sind Funktionen aus $H^1(\Omega)$ nicht stetig und auch nicht auf dem Rand des Gebiets definiert. Es existiert aber ein Operator der jeder Funktion aus $H^1(\Omega)$ eine Funktion aus $H^{1/2}(\Gamma)$ zuordnet, wie folgender Satz zeigt.

Satz 2.3 (SAUTER / SCHWAB 2004 [18, Satz 2.6.8]). Es existieren stetige, lineare Spurooperatoren

$\gamma_0^{int} : H^1(\Omega) \longrightarrow H^{1/2}(\Gamma)$ mit

$$\gamma_0^{int} u = u|_\Gamma \quad \text{für alle } u \in C^0(\bar{\Omega}) \cap H^1(\Omega), \quad (2.6)$$

und

$\gamma_0^{ext} : H^1_{lok}(\Omega^{ext}) \longrightarrow H^{1/2}(\Gamma)$ mit

$$\gamma_0^{ext} u = u|_\Gamma \quad \text{für alle } u \in C^0(\bar{\Omega^{ext}}) \cap H^1(\Omega^{ext}). \quad (2.7)$$

Dabei wird $\gamma_0^{int} u$ als innere Spur von u und $\gamma_0^{ext} u$ als äußere Spur von u bezeichnet.

Nach Definition ist daher der Spuroperator angewandt auf eine stetige Funktion u nichts weiter als die Restriktion von u auf den Rand. Für $u \in H_0^1(\Omega)$ gilt $\gamma_0^{int}u = 0$ im Sinne von $H^{1/2}(\Gamma)$.

Umgekehrt kann man sich auch fragen, ob Funktionen aus $H^{1/2}(\Gamma)$ auf $H^1(\Omega)$ fortgesetzt werden können. Dazu zitieren wir den folgenden Satz aus PRAETORIUS 2007 [16].

Satz 2.4 (PRAETORIUS 2007 [16, Satz 2.5]). *Es existiert ein stetiger, linearer Operator $\mathcal{L} : H^{1/2}(\Gamma) \longrightarrow H^1(\Omega)$, so dass*

$$\gamma_0(\mathcal{L}v) = v \quad \text{für alle } v \in H^{1/2}(\Gamma)$$

gilt. Wir nennen $\mathcal{L}$ auch Fortsetzungsoperator.

Man beachte, dass der Fortsetzungsoperator $\mathcal{L}$ nicht eindeutig ist.

Definition 2.9. Für $\ell \in \mathbb{N}$ bezeichne $H_*^\ell(\Omega)$ den Raum

$$H_*^\ell(\Omega) := \left\{ u \in H^\ell(\Omega) \mid \int_\Omega u \, dx = 0 \right\}.$$

Analog bezeichnen wir mit $H_*^{1/2}(\Gamma)$ den Raum

$$H_*^{1/2}(\Gamma) := \left\{ v \in H^{1/2}(\Gamma) \mid \int_\Gamma v \, ds_x = 0 \right\}.$$

Funktionen aus Sobolev-Räumen lassen sich durch gewisse glatte Funktionen approximieren, wie nachfolgender Satz besagt.

Satz 2.5 (Meyers-Serrin, EVANS 2002 [6, Abschnitt 5.3, Satz 3]). *Für $\ell \in \mathbb{N}_0$ liegt der Raum $C^\infty(\overline{\Omega})$ dicht in $H^\ell(\Omega)$ bezüglich der H^ℓ -Norm.*

Satz 2.6 (Sobolev'scher Einbettungssatz, SAUTER / SCHWAB 2004 [18, Satz 2.5.4]). *Sei Ω ein beschränktes C^k -Gebiet (beschränktes Gebiet mit k -mal stetig differenzierbarem Rand). Für $k > \ell$ und $m < \ell - d/2$ besitzen Funktionen aus $H^\ell(\Omega)$ einen m -mal stetig differenzierbaren Repräsentanten. Die Einbettung*

$$H^\ell(\Omega) \hookrightarrow C^m(\overline{\Omega})$$

ist stetig, d.h.

$$\|u\|_{C^m(\overline{\Omega})} \leq C \|u\|_{H^\ell(\Omega)} \quad \text{für alle } u \in H^\ell(\Omega),$$

mit einer positiven Konstante C .

Um Funktionen aus dem Dualraum eines Sobolev-Raumes zu charakterisieren benötigen wir noch folgendes Lemma.

Lemma 2.7 (PRAETORIUS 2007 [16, Lemma 4]). *Seien X und Y reelle Hilberträume mit stetiger Einbettung $X \subseteq Y$, dann ist die Riesz-Abbildung $J_Y : Y \longrightarrow Y^*$, $J_Y y := (y, \cdot)_Y$ ein wohldefinierter, stetiger, linearer Operator $J_Y : Y \longrightarrow X^*$. Des weiteren ist $J_Y(Y)$ ein dichter Unterraum von X^* .*

Wir wenden das Lemma an auf $X = H^\ell(\Omega)$ und $Y = L^2(\Omega)$. Der Raum $L^2(\Omega)$ liegt daher dicht in $H^\ell(\Omega)^*$. Für $u \in L^2(\Omega)$ und $v \in H^\ell(\Omega)$ stimmt die duale Klammer $\langle u, v \rangle$ mit dem L^2 -Skalarprodukt $(u, v)_{L^2(\Omega)}$ überein. Analog wenden wir das Lemma auf $X = H^{1/2}(\Gamma)$ und $Y = L^2(\Gamma)$ an. Für $v \in H^{1/2}(\Gamma)$ stimmt die duale Klammer $\langle u, v \rangle$ dann mit dem L^2 -Skalarprodukt $(u, v)_{L^2(\Gamma)}$ überein, falls $u \in L^2(\Gamma)$.

Definition 2.10. Für $\ell \in \mathbb{N}$ bezeichnen wir mit

$$\tilde{H}^{-\ell}(\Omega) := H^\ell(\Omega)^*$$

den Dualraum von $H^\ell(\Omega)$ bezüglich des erweiterten L^2 -Skalarproduktes und mit

$$H^{-1/2}(\Gamma) := H^{1/2}(\Gamma)^*$$

den Dualraum von $H^{1/2}(\Gamma)$ bezüglich des erweiterten L^2 -Skalarproduktes.

2.2 Schwache Formulierung

Wir betrachten für gegebenes $f \in C^0(\overline{\Omega})$ die partielle Differentialgleichung

$$-\Delta u = f \quad \text{in } \Omega. \quad (2.8)$$

Existiert eine Lösung $u \in C^2(\overline{\Omega})$, so nennt man u eine klassische Lösung der Differentialgleichung. Im Allgemeinen jedoch lassen sich keine Aussagen über die eindeutige Existenz einer Lösung tätigen. Man geht daher zur so genannten schwachen bzw. distributionellen Formulierung über. Um dies zu motivieren gehen wir davon aus, dass u eine klassische Lösung ist und multiplizieren Gleichung (2.8) mit einer Testfunktion $v \in C^\infty(\overline{\Omega})$ und integrieren über das Gebiet Ω . Nach partieller Integration der linken Seite erhält man

$$(\nabla u, \nabla v)_{L^2(\Omega)} = (f, v)_{L^2(\Omega)} + (\partial u / \partial \mathbf{n}, \gamma_0^{int} v)_{L^2(\Gamma)} \quad \text{für alle } v \in C^\infty(\overline{\Omega}). \quad (2.9)$$

Mit $\mathbf{n}$ bezeichnen wir dabei den von Ω nach Ω^{ext} zeigenden Normalenvektor. Klarerweise sind klassische Lösungen von (2.8) auch Lösungen von (2.9).

Definition 2.11. Wir nennen eine Funktion $u \in L_{lok}^1(\Omega)$ eine schwache Lösung von $-\Delta u = f$, falls sie $-\Delta u = f$ im distributionellen Sinne erfüllt, d.h. es gilt

$$\langle -\Delta u, v \rangle = \langle f, v \rangle \quad \text{für alle } v \in C_0^\infty(\Omega).$$

Die Distribution $-\Delta u$ kann für $u \in H^1(\Omega)$ dabei über

$$\langle -\Delta u, v \rangle = (\nabla u, \nabla v)_{L^2(\Omega)} \quad \text{für alle } v \in C_0^\infty(\Omega)$$

definiert werden.

Lemma 2.8 (MCLEAN 2000 [15, Lemma 4.3]). Sei $f \in \tilde{H}^{-1}(\Omega)$, und $u \in H^1(\Omega)$ erfülle

$$-\Delta u = f$$

im distributionellen Sinne. Dann existiert ein eindeutiges lineares Funktional $\gamma_1 u \in H^{-1/2}(\Gamma)$ mit

$$(\nabla u, \nabla v)_{L^2(\Omega)} = \langle f, v \rangle + \langle \gamma_1 u, \gamma_0^{int} v \rangle_\Gamma \quad \text{für alle } v \in H^1(\Omega). \quad (2.10)$$

Des weiteren ist $\gamma_1 u$ eindeutig durch u und f bestimmt, und es gilt die Abschätzung

$$\|\gamma_1 u\|_{H^{-1/2}(\Gamma)} \leq C \left(\|u\|_{H^1(\Omega)} + \|f\|_{\tilde{H}^{-1}(\Omega)} \right). \quad (2.11)$$

Dabei ist C die Stetigkeitskonstante des Fortsetzungsoperators $\mathcal{L}$. Das Funktional $\gamma_1 u$ wird auch als Konormalenableitung bezeichnet.

Die Variationsformulierung (2.10) in Lemma 2.8 bezeichnet man als schwache Formulierung der Differentialgleichung (2.8). Gelten für die Lösung der schwachen Formulierung zusätzliche Regularitätsannahmen, d.h. $u \in C^2(\overline{\Omega})$, dann ist die schwache Lösung auch eine klassische Lösung und das Funktional $\gamma_1 u$ stimmt mit $\partial u / \partial \mathbf{n}$ überein. Jede klassische Lösung ist wiederum eine Lösung der distributionellen Formulierung.

Wir bezeichnen mit $\gamma_1^{ext} u$ die äußere Konormalenableitung und halten fest, dass für genügend glatte Funktionen $\gamma_1^{ext} u = \partial u / \partial \mathbf{n}$ gilt. Der Normalenvektor $\mathbf{n}$ zeigt dabei von Ω nach Ω^{ext} .

Als nächstes wollen wir uns der schwachen Formulierung des Transmissionsproblems widmen. Dazu betrachten wir Funktionen u auf dem Außenraum Ω^{ext} , die der Abklingbedingung $u = b \log \|x\| + \mathcal{O}(1/\|x\|)$ falls $d = 2$ bzw. $u = \mathcal{O}(1/\|x\|)$ falls $d = 3$ für $\|x\| \rightarrow \infty$ genügen. Der Sobolev-Raum $H^1(\Omega^{ext})$ enthält nicht alle Funktionen die der Abklingbedingung genügen, wie aus nachfolgendem Beispiel ersichtlich ist.

Beispiel 2.1. Sei $\Omega := \{x \in \mathbb{R}^d \mid \|x\| < R\}$ eine Kugel mit Radius $R > 0$, $u(x) = 1/\|x\|$ und $v(x) = \log \|x\|$, dann gilt $u, v \notin H^1(\Omega^{ext})$.

Beweis. Es gilt $\|u\|_{H^1(\Omega^{ext})} \geq \|u\|_{L^2(\Omega^{ext})}$. Wir bezeichnen mit $|S_d|$ das Maß der Einheitskugel in $\mathbb{R}^d$. Damit ergibt sich

$$\|u\|_{L^2(\Omega^{ext})}^2 = \int_{\Omega^{ext}} \frac{1}{\|x\|^2} dx = |S_d| \int_R^\infty r^{d-3} dr = \infty.$$

Für v gilt analog

$$\|v\|_{L^2(\Omega^{ext})}^2 = \int_{\Omega^{ext}} (\log\|x\|)^2 dx = |S_d| \int_R^\infty r^{d-1} (\log r)^2 dr = \infty.$$

Somit wäre die Aussage bewiesen. $\square$

Der Raum $H^1(\Omega^{ext})$ ist daher nicht günstig, um die magnetostatischen Gleichungen im schwachen Sinne zu definieren. Für die Lösung im Außenraum benutzen wir den Raum $H_{lok}^1(\Omega^{ext})$, der Funktionen, die die Abklingbedingung erfüllen, enthält. Des weiteren lassen sich im Raum $H_{lok}^1(\Omega^{ext})$ Existenz- und Eindeutigkeitsaussagen zeigen. Um ein Transmissionsproblem aufzustellen, benötigt man noch Informationen über das Verhalten der Funktionen am Rand. Gibt man die Sprünge der Spuren und Konormalenableitungen vor, so lässt sich das Transmissionsproblem im schwachen Sinne anschreiben.

Definition 2.12 (Transmissionsproblem (TP)). Seien $f \in L^2(\Omega)$, $g_0 \in H^{1/2}(\Gamma)$ und $g_1 \in H^{-1/2}(\Gamma)$ gegeben. Wir suchen eine Funktion $u := (u^{int}, u^{ext}) \in H^1(\Omega) \times H_{lok}^1(\Omega^{ext})$, welche

$$\begin{aligned} -\Delta u^{int} &= f & \text{in } \Omega, \\ -\Delta u^{ext} &= 0 & \text{in } \Omega^{ext}, \\ [\gamma_0 u] &= g_0 & \text{auf } \Gamma, \\ [\gamma_1 u] &= g_1 & \text{auf } \Gamma, \end{aligned} \tag{2.12}$$

$$u^{ext}(\mathbf{x}) = \begin{cases} b \log\|\mathbf{x}\| + \mathcal{O}\left(\frac{1}{\|\mathbf{x}\|}\right) & \text{falls } d = 2, \\ \mathcal{O}\left(\frac{1}{\|\mathbf{x}\|}\right) & \text{falls } d = 3, \end{cases} \quad \text{für } \|\mathbf{x}\| \rightarrow \infty.$$

erfüllt. Die Gleichungen sind dabei im distributionellen Sinne zu verstehen. Die eckigen Klammern bezeichnen die Sprünge der Spuren bzw. Konormalenableitungen von u , d.h. $[\gamma_0 u] := \gamma_0^{ext} u^{ext} - \gamma_0^{int} u^{int}$ bzw. $[\gamma_1 u] := \gamma_1^{ext} u^{ext} - \gamma_1^{int} u^{int}$.

Wir wollen jetzt noch den Ausdruck $\mathbf{M} \cdot \mathbf{n}$ definieren. Die Abbildung γ_n welche $\mathbf{v}$ auf $\mathbf{v} \cdot \mathbf{n}$ abbildet ist für glatte Funktionen $\mathbf{v}$ mit kompakten Träger in $\mathbb{R}^d$ definiert. Nach GIRAULT / RAVIART 1986 [11, Satz 2.5] kann γ_n zu einer stetigen, linearen Abbildung von $H(\text{div}; \Omega)$ auf $H^{-1/2}(\Gamma)$ erweitert werden. Dabei ist der Hilbertraum $H(\text{div}; \Omega)$ definiert als

$$H(\text{div}; \Omega) := \{ \mathbf{v} \in (L^2(\Omega))^d \mid \text{div } \mathbf{v} \in L^2(\Omega) \}.$$

Nach Definition ist daher $(H^1(\Omega))^d$ ein Teilraum von $H(\text{div}, \Omega)$.

Konvention 2.13. Für $\mathbf{M} \in (H^1(\Omega))^d$ ist der Ausdruck $\mathbf{M} \cdot \mathbf{n}$ definiert als

$$\mathbf{M} \cdot \mathbf{n} := \gamma_n(\mathbf{M})$$

Daher gilt $\mathbf{M} \cdot \mathbf{n} \in H^{-1/2}(\Gamma)$.

Bemerkung 2.2. In den nachfolgenden Resultaten kann die Voraussetzung $\mathbf{M} \in (H^1(\Omega))^d$ durch die schwächere Annahme $\mathbf{M} \in H(\text{div}, \Omega)$ ersetzt werden.

Um nun unser Modellproblem (siehe Definition 1.1) neu zu formulieren, setzen wir in Definition 2.12 $f = -\operatorname{div} \mathbf{M}$, $g_0 = 0$ und $g_1 = -\mathbf{M} \cdot \mathbf{n}$ ein wobei $\mathbf{M} \in (H^1(\Omega))^d$. Das magnetostatische Potential $u = (u^{int}, u^{ext}) \in H^1(\Omega) \times H_{lok}^1(\Omega^{ext})$ ist daher Lösung der Gleichungen

$$\begin{aligned} -\Delta u^{int} &= -\operatorname{div} \mathbf{M} && \text{in } \Omega, \\ -\Delta u^{ext} &= 0 && \text{in } \Omega^{ext}, \\ [\gamma_0 u] &= 0 && \text{auf } \Gamma, \\ [\gamma_1 u] &= -\mathbf{M} \cdot \mathbf{n} && \text{auf } \Gamma, \\ u^{ext}(\mathbf{x}) &= \begin{cases} b \log \|\mathbf{x}\| + \mathcal{O}\left(\frac{1}{\|\mathbf{x}\|}\right) & \text{falls } d = 2, \\ \mathcal{O}\left(\frac{1}{\|\mathbf{x}\|}\right) & \text{falls } d = 3, \end{cases} && \text{für } \|\mathbf{x}\| \rightarrow \infty. \end{aligned} \quad (2.13)$$

In Abschnitt 2.4 werden wir uns mit der eindeutigen Lösbarkeit des Transmissionsproblems beschäftigen.

2.3 Integraloperatoren

Zunächst betrachten wir die partielle Differentialgleichung

$$-\Delta u = f \quad \text{auf } \Omega \subseteq \mathbb{R}^d. \quad (2.14)$$

Die Fundamentallösung des Laplaceoperators $-\Delta$ ist durch die Green'sche Funktion

$$G(x, y) := \begin{cases} -\frac{1}{2\pi} \log(\|x - y\|) & \text{falls } d = 2, \\ \frac{1}{4\pi} \frac{1}{\|x - y\|} & \text{falls } d = 3 \end{cases} \quad (2.15)$$

gegeben.

Das folgende Resultat besagt, dass man Funktionen $u \in C^2(\overline{\Omega})$, die die Poisson-Gleichung (2.14) erfüllen, durch ihre Werte am Rand, ihre Normalenableitungen am Rand und f darstellen kann.

Satz 2.9 (Darstellungsformel, PRAETORIUS 2007 [16, Proposition 2]). *Sei $\Omega \subseteq \mathbb{R}^d$ ein beschränktes Gebiet mit glattem Rand $\Gamma := \partial\Omega$ und $u \in C^2(\overline{\Omega})$. Mit $f := -\Delta u \in C(\overline{\Omega})$ gilt die Darstellungsformel*

$$u(x) = \int_{\Omega} G(x, y) f(y) dy + \int_{\Gamma} G(x, y) \gamma_1^{int} u(y) ds_y - \int_{\Gamma} \gamma_{1,y}^{int} G(x, y) \gamma_0 u(y) ds_y \quad \text{für alle } x \in \Omega. \quad (2.16)$$

Der zusätzliche Index bei $\gamma_{1,y}^{int}$ gibt an, dass die Ableitung nach y erfolgt.

Die Terme in (2.16) definieren Integraloperatoren. Mit

$$(\tilde{\mathcal{N}}f)(x) := \int_{\Omega} G(x, y) f(y) dy \quad \text{für } x \in \mathbb{R}^d \setminus \Gamma \quad (2.17)$$

bezeichnen wir das Newton-Potential. Weiters wird durch

$$(\tilde{\mathcal{V}}\phi)(x) := \int_{\Gamma} G(x, y) \phi(y) ds_y \quad \text{für } x \in \mathbb{R}^d \setminus \Gamma \quad (2.18)$$

das Einfachschichtpotential gegeben, während

$$(\tilde{\mathcal{K}}\phi)(x) := \int_{\Gamma} \gamma_{1,y}^{int} G(x, y) \phi(y) ds_y \quad \text{für } x \in \mathbb{R}^d \setminus \Gamma \quad (2.19)$$

das Doppelschichtpotential definiert.

Satz 2.10 (SAUTER / SCHWAB 2004 [18, Satz 3.1.1]). Sei $\phi \in L^1(\Gamma)$. Dann ist $\tilde{\mathcal{V}}\phi \in C^\infty(\mathbb{R}^d \setminus \Gamma)$ und $\tilde{\mathcal{K}}\phi \in C^\infty(\mathbb{R}^d \setminus \Gamma)$. Es gilt

$$(\Delta \tilde{\mathcal{V}}\phi)(x) = (\Delta \tilde{\mathcal{K}}\phi)(x) = 0 \quad \text{für alle } x \in \mathbb{R}^d \setminus \Gamma. \quad (2.20)$$

Satz 2.11. Falls $\phi \in L^\infty(\Gamma)$, dann gilt $\tilde{\mathcal{V}}\phi \in C(\mathbb{R}^d)$.

Wir befassen uns im Folgenden mit den wichtigsten Eigenschaften des Einfach- und Doppelschichtpotentials. Das Einfachschichtpotential lässt sich zu einem stetigen linearen Operator von $H^{-1/2}(\Gamma)$ nach $H^1(\Omega)$ erweitern.

Satz 2.12 (STEINBACH 2003 [20, Lemma 6.3]). Der Operator $\tilde{\mathcal{V}} : H^{-1/2}(\Gamma) \rightarrow H^1(\Omega)$ ist stetig und linear, und $\tilde{\mathcal{V}}\phi$ ist Lösung der homogenen partiellen Differentialgleichung

$$\Delta \tilde{\mathcal{V}}\phi = 0 \quad \text{in } \Omega.$$

Des weiteren ist $\tilde{\mathcal{V}}\phi \in H_{lok}^1(\Omega^{ext})$, und es gilt

$$\Delta \tilde{\mathcal{V}}\phi = 0 \quad \text{in } \Omega^{ext}.$$

Wendet man die innere Spur γ_0^{int} auf $\tilde{\mathcal{V}}\phi \in H^1(\Omega)$ an, so lässt sich der Operator

$$\mathcal{V} := \gamma_0^{int} \tilde{\mathcal{V}} : H^{-1/2}(\Gamma) \rightarrow H^1(\Omega) \quad (2.21)$$

definieren.

Durch Anwenden der äußeren Spur an $\tilde{\mathcal{V}}$ kann man folgende Sprungbedingung für das Einfachschichtpotential am Rand zeigen.

Satz 2.13 (Sprungbedingung Einfachschichtpotential, STEINBACH 2003 [20, Abschnitt 6.2]). Sei $\phi \in H^{-1/2}(\Gamma)$. Das Einfachschichtpotential erfüllt

$$[\gamma_0 \tilde{\mathcal{V}}\phi] := \gamma_0^{ext}(\tilde{\mathcal{V}}\phi) - \gamma_0^{int}(\tilde{\mathcal{V}}\phi) = 0.$$

Satz 2.14 (STEINBACH 2003 [20, Satz 6.5, Satz 6.6]). Sei $\phi \in H^{-1/2}(\Gamma)$. Für $d = 3$ ist der Operator $\mathcal{V}$ elliptisch, d.h. es gilt

$$\langle \phi, \mathcal{V}\phi \rangle_\Gamma \geq C \|\phi\|_{H^{-1/2}(\Gamma)}^2 \quad \text{für alle } \phi \in H^{-1/2}(\Gamma),$$

mit einer positiven Konstanten C . Für $d = 2$ ist $\mathcal{V}$ elliptisch, falls $\text{diam}(\Omega) < 1$.

Als Nächstes betrachten wir das Doppelschichtpotential $\tilde{\mathcal{K}}$ und geben analoge Eigenschaften an.

Satz 2.15 (STEINBACH 2003 [20, Lemma 6.7]). Der Operator $\tilde{\mathcal{K}} : H^{1/2}(\Gamma) \rightarrow H^1(\Omega)$ ist stetig und linear. $\tilde{\mathcal{K}}\phi$ ist Lösung der homogenen partiellen Differentialgleichung

$$\Delta \tilde{\mathcal{K}}\phi = 0 \quad \text{in } \Omega.$$

Des weiteren ist $\tilde{\mathcal{K}}\phi \in H_{lok}^1(\Omega^{ext})$, und es gilt

$$\Delta \tilde{\mathcal{K}}\phi = 0 \quad \text{in } \Omega^{ext}.$$

Für $\phi \in H^{1/2}(\Gamma)$ lässt sich daher nach Satz 2.3 die innere Spur von $\tilde{\mathcal{K}}\phi$ bilden. Im Gegensatz zum Einfachschichtpotential besitzt das Doppelschichtpotential jedoch einen Sprung am Rand. Dazu definieren wir den stetigen linearen Operator $\mathcal{K} : H^{1/2}(\Gamma) \rightarrow H^{1/2}(\Gamma)$ durch

$$\mathcal{K} := \gamma_0^{int} \tilde{\mathcal{K}} + (1 - \sigma). \quad (2.22)$$

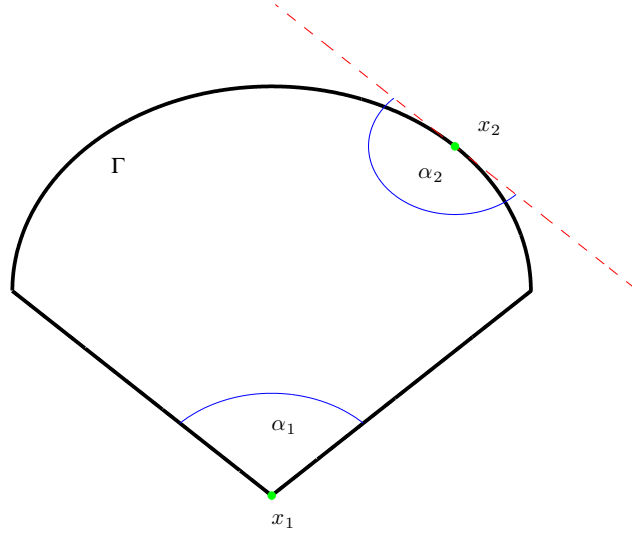


Abbildung 2.1: Für den Punkt $x_1 \in \Gamma$ ist der zugehörige Innenwinkel α_1 eingezeichnet. Der Rand Γ ist in einer Umgebung um den Punkt x_2 glatt, besitzt daher eine Tangente (rote strichlierte Linie) und somit ergibt sich für den Innenwinkel $\alpha_2 = \pi$.

Die Größe $\sigma(x)$ ist wie in [20, Lemma 6.5] definiert als

$$\sigma(x) := \lim_{\varepsilon \rightarrow 0} \frac{1}{2(d-1)\pi} \frac{1}{\varepsilon^{d-1}} \int_{\partial K_\varepsilon^{\text{int}}(x)} ds_y \quad \text{für } x \in \Gamma, \quad (2.23)$$

wobei $K_\varepsilon^{\text{int}}(x) := \{y \in \Omega \mid \|x - y\| < \varepsilon\}$ der Teil einer Kugel um den Punkt $x \in \Gamma$ mit Radius ε ist, welcher im Gebiet Ω liegt. Für einen Punkt $x \in \Gamma$ ist die Größe $\sigma(x)$ proportional zum Innenwinkel. Beispiele von verschiedenen Innenwinkeln sind etwa in Abbildung 2.1 zu finden.

Bemerkung 2.3. Besitzt Γ in einem Punkt $x \in \Gamma$ eine Tangente, dann ist nach Definition (2.23) $\sigma(x) = \frac{1}{2}$. Für Lipschitz-Gebiete existiert die Tangente für fast alle $x \in \Gamma$, daher gilt

$$\sigma(x) = \frac{1}{2} \quad \text{für fast alle } x \in \Gamma.$$

Satz 2.16 (Sprungbedingung Doppelschichtpotential, STEINBACH 2003 [20, Lemma 6.8]). Sei $\phi \in H^{1/2}(\Gamma)$. Für das Doppelschichtpotential $\tilde{\mathcal{K}}$ gilt

$$\gamma_0^{\text{ext}} \tilde{\mathcal{K}} = \sigma + \mathcal{K}.$$

Das Doppelschichtpotential besitzt daher am Rand den Sprung

$$[\gamma_0 \tilde{\mathcal{K}} \phi] = \gamma_0^{\text{ext}}(\tilde{\mathcal{K}} \phi) - \gamma_0^{\text{int}}(\tilde{\mathcal{K}} \phi) = \phi.$$

Nach Satz 2.12 bzw. Satz 2.15 und Lemma 2.8 sind $\gamma_1^{\text{int}} \tilde{\mathcal{V}} : H^{-1/2}(\Gamma) \rightarrow H^{-1/2}(\Gamma)$ und $\gamma_1^{\text{int}} \tilde{\mathcal{K}} : H^{1/2}(\Gamma) \rightarrow H^{-1/2}(\Gamma)$ wohldefinierte stetige lineare Operatoren.

Satz 2.17 (STEINBACH 2003 [20, Lemma 6.6, Lemma 6.9]). *Die Sprünge der Konormalenableitungen von $\tilde{\mathcal{V}}$ und $\tilde{\mathcal{K}}$ lauten*

$$[\gamma_1 \tilde{\mathcal{V}} \phi] = \gamma_1^{ext}(\tilde{\mathcal{V}} \phi) - \gamma_1^{int}(\tilde{\mathcal{V}} \phi) = -\phi \quad \text{für } \phi \in H^{-1/2}(\Gamma)$$

und

$$[\gamma_1 \tilde{\mathcal{K}} \phi] = \gamma_1^{ext}(\tilde{\mathcal{K}} \phi) - \gamma_1^{int}(\tilde{\mathcal{K}} \phi) = 0 \quad \text{für } \phi \in H^{1/2}(\Gamma).$$

Mit den Konormalenableitungen des Einfach- und Doppelschichtpotentials lassen sich weitere lineare stetige Integraloperatoren erklären. Das adjungierte Doppelschichtpotential $\mathcal{K}' : H^{-1/2}(\Gamma) \rightarrow H^{-1/2}(\Gamma)$ ist definiert als

$$\mathcal{K}' := \gamma_1^{int} \tilde{\mathcal{V}} - \sigma \quad (2.24)$$

während der hypersinguläre Integraloperator $\mathcal{W} : H^{1/2}(\Gamma) \rightarrow H^{-1/2}(\Gamma)$ definiert ist als

$$\mathcal{W} := -\gamma_1^{int} \tilde{\mathcal{K}}. \quad (2.25)$$

Bemerkung 2.4. *In der Literatur werden sowohl $\mathcal{V}$ als auch $\tilde{\mathcal{V}}$ als Einfachschichtpotential und sowohl $\mathcal{K}$ als auch $\tilde{\mathcal{K}}$ als Doppelschichtpotential bezeichnet.*

Man kann den Stetigkeitsbereich der Integraloperatoren noch erweitern. Dazu sei etwa auf das Buch von SAUTER / SCHWAB 2004 [18, Satz 3.1.16, Bemerkung 3.1.18] verwiesen.

Satz 2.18. *Die Operatoren $\tilde{\mathcal{V}}$, $\mathcal{V}$, $\tilde{\mathcal{K}}$ und $\mathcal{K}$ sind stetig für $|s| \leq 1/2$:*

- $\tilde{\mathcal{V}} : H^{-1/2+s}(\Gamma) \rightarrow H^{1+s}(\Omega),$
- $\mathcal{V} : H^{-1/2+s}(\Gamma) \rightarrow H^{1/2+s}(\Gamma),$
- $\tilde{\mathcal{K}} : H^{1/2+s}(\Gamma) \rightarrow H^{1+s}(\Omega),$
- $\mathcal{K} : H^{1/2+s}(\Gamma) \rightarrow H^{1/2+s}(\Gamma).$

Mit zusätzlichen Voraussetzungen an den Rand können diese Stetigkeitsbereiche noch erweitert werden: Der Lipschitz-Rand Γ sei global glatt, d.h. $\Gamma \in C^\infty$, dann gelten obige Aussagen für alle $s > -1/2$.

Satz 2.19 (Abklingverhalten von $\tilde{\mathcal{V}}$ und $\tilde{\mathcal{K}}$). *Für $\phi \in H^{1/2}(\Gamma)$ und $d = 2$ gilt*

$$\tilde{\mathcal{K}} \phi(\mathbf{x}) = \mathcal{O} \left(\frac{1}{\|\mathbf{x}\|} \right) \quad \text{für } \|\mathbf{x}\| \rightarrow \infty,$$

und falls $d = 3$ dann gilt für $\tilde{\mathcal{K}}$ das Abklingverhalten

$$\tilde{\mathcal{K}} \phi(\mathbf{x}) = \mathcal{O} \left(\frac{1}{\|\mathbf{x}\|^2} \right) \quad \text{für } \|\mathbf{x}\| \rightarrow \infty.$$

Für $\phi \in H^{-1/2}(\Gamma)$ erfüllt das Einfachschichtpotential im Fall $d = 2$

$$\tilde{\mathcal{V}} \phi(\mathbf{x}) = -\frac{1}{2\pi} \langle \phi, 1 \rangle_\Gamma \log \|\mathbf{x}\| + \mathcal{O} \left(\frac{1}{\|\mathbf{x}\|} \right) \quad \text{für } \|\mathbf{x}\| \rightarrow \infty$$

fast überall in Ω^{ext} und wenn $d = 3$, dann gilt

$$\tilde{\mathcal{V}} \phi(\mathbf{x}) = \mathcal{O} \left(\frac{1}{\|\mathbf{x}\|} \right) \quad \text{für } \|\mathbf{x}\| \rightarrow \infty.$$

Für einen Beweis des Abklingverhaltens der Integraloperatoren für $d = 2$ sei auf die Arbeit von FEISCHL 2010 [7, Satz 3.9 und 3.10] verwiesen. Im Fall $d = 3$ findet sich das Abklingverhalten in SAUTER / SCHWAB 2004 [18, Beweis von Satz 3.1.12].

Wir geben noch die Darstellungsformel für das Außenraumproblem an, die auch unter der dritten Green'schen Identität bekannt ist. Essentiell dafür sind die Abklingbedingungen, welche an die Lösungsfunktion gestellt werden, dazu sei auf MCLEAN 2000 [15, Kapitel 8] verwiesen. Ein Beweis dieser Formel ist etwa in [15, Satz 7.12] zu finden.

Satz 2.20 (Darstellungsformel im Außenraum). *Erfüllt $u \in H_{lok}^1(\Omega^{ext})$*

$$\begin{aligned} \Delta u &= 0, \\ u(\mathbf{x}) &= \begin{cases} b \log \|\mathbf{x}\| + \mathcal{O}\left(\frac{1}{\|\mathbf{x}\|}\right) & \text{falls } d = 2, \\ \mathcal{O}\left(\frac{1}{\|\mathbf{x}\|}\right) & \text{falls } d = 3, \end{cases} \quad \text{für } \|\mathbf{x}\| \rightarrow \infty, \end{aligned}$$

dann gilt die Darstellungsformel

$$u = \tilde{K}\gamma_0^{ext}u - \tilde{V}\gamma_1^{ext}u.$$

2.4 Eindeutige Lösbarkeit

Zum Abschluss dieses Kapitels listen wir in diesem Abschnitt noch einige Existenz- und Eindeutigkeitsaussagen auf. Zunächst schreiben wir wichtige Sätze über die Existenz und Eindeutigkeit der Lösung der schwachen Formulierung des Dirichlet und Neumann Problems zusammen. Danach präsentieren wir Ergebnisse der Arbeit von CARSTENSEN / STEPHAN 1995 [4], die die eindeutige Lösbarkeit des Transmissionsproblems beweisen.

Konvention 2.14. *Im Folgenden werden wir den Index $(\cdot)^{int}$ bei γ_0^{int} weglassen, d.h. die innere Spur wird mit γ_0 bezeichnet. Analoges gilt auch für die innere Konormalenableitung γ_1^{int} , welche dann durch γ_1 ersetzt wird.*

Für einen Beweis der folgenden aufgelisteten Resultate sei auf SAUTER / SCHWAB 2004 [18] bzw. PRAETORIUS 2008 [17] verwiesen. Wir betrachten das Dirichlet Problem

$$\begin{aligned} -\Delta u &= f & \text{in } \Omega, \\ u &= g_D & \text{auf } \Gamma. \end{aligned} \tag{2.26}$$

Satz 2.21. • Sei $f \in C^0(\overline{\Omega})$ und $g_D \in C^0(\Gamma)$ gegeben. Ist $u \in C^2(\overline{\Omega})$ eine Lösung von (2.26), dann erfüllt u die Variationsformulierung (schwache Formulierung)

$$\begin{aligned} (\nabla u, \nabla v)_{L^2(\Omega)} &= \langle f, v \rangle \quad \text{für alle } v \in H_0^1(\Omega), \\ \gamma_0 u &= g_D. \end{aligned} \tag{2.27}$$

- Seien $f \in L^2(\Omega)$ und $g_D \in H^{1/2}(\Gamma)$ gegeben, dann existiert für die Variationsformulierung (2.27) genau eine Lösung $u \in H^1(\Omega)$. Die schwache Lösung u hängt stetig von den Daten ab, d.h.

$$\|u\|_{H^1(\Omega)} \leq C \left(\|f\|_{L^2(\Omega)} + \|g_D\|_{H^{1/2}(\Gamma)} \right), \tag{2.28}$$

dabei hängt die Konstante C nur vom Gebiet Ω ab. Die stetige Abhängigkeit der Lösung von den Daten wird auch Stabilität der schwachen Lösung genannt.

- Gelten noch die zusätzlichen Regularitätsannahmen $u \in C^2(\overline{\Omega})$, $f \in C^0(\overline{\Omega})$ und $g_D \in C^0(\Gamma)$, dann löst u auch die starke Formulierung (2.26).

Als Nächstes gehen wir über zum Neumann Problem, welches in der starken Formulierung lautet

$$\begin{aligned} -\Delta u &= f & \text{in } \Omega, \\ \frac{\partial u}{\partial \mathbf{n}} &= g_N & \text{auf } \Gamma. \end{aligned} \quad (2.29)$$

Damit eine Lösung des Neumann Problems existieren kann, müssen die Daten f und g_N

$$\int_{\Omega} f \, dx + \int_{\Gamma} g_N \, ds_y = 0 \quad (2.30)$$

erfüllen, wie man aus der Beziehung

$$\int_{\Omega} f \, dx = - \int_{\Omega} \Delta u \, dx = - \int_{\Gamma} \nabla u \cdot \mathbf{n} \, ds_y = - \int_{\Gamma} g_N \, ds_y$$

sehen kann.

Des weiteren ist mit einer Lösung u des Neumann Problems auch $u + c$ mit $c \in \mathbb{R}$ eine Lösung. Diese Konstante wird beispielsweise mit der Bedingung

$$\int_{\Omega} u \, dx = 0$$

festgesetzt.

Satz 2.22. • Erfüllen $f \in C^0(\overline{\Omega})$ und $g_N \in C^0(\Gamma)$ die Bedingung (2.30) und ist $u \in C^2(\overline{\Omega})$ eine Lösung von (2.29), dann ist $u \in H^1(\Omega)$ und erfüllt die Variationsformulierung (schwache Formulierung)

$$(\nabla u, \nabla v)_{L^2(\Omega)} = \langle f, v \rangle + \langle g_N, \gamma_0 v \rangle_{\Gamma} \quad \text{für alle } v \in H^1(\Omega). \quad (2.31)$$

• Seien $f \in L^2(\Omega)$ und $g_N \in H^{-1/2}(\Gamma)$, dann besitzt die Variationsformulierung

$$(\nabla u, \nabla v)_{L^2(\Omega)} = \langle f, v \rangle + \langle g_N, \gamma_0 v \rangle_{\Gamma} \quad \text{für alle } v \in H_*^1(\Omega) \quad (2.32)$$

genau eine Lösung $u \in H_*^1(\Omega)$.

• Erfüllen die Daten $f \in L^2(\Omega)$ und $g_N \in H^{-1/2}(\Gamma)$ die Bedingung $\langle f, 1 \rangle + \langle g_N, 1 \rangle_{\Gamma} = 0$, dann ist die eindeutige Lösung von (2.32) auch eine Lösung der schwachen Formulierung (2.31) und es gilt die Stabilitätsabschätzung

$$\|u\|_{H^1(\Omega)} \leq C (\|f\|_{L^2(\Omega)} + \|g_N\|_{H^{-1/2}(\Gamma)}), \quad (2.33)$$

wobei die Konstante C nur vom Gebiet Ω abhängt.

• Seien $f \in C^0(\overline{\Omega})$ und $g_N \in C^0(\Gamma)$ und es gelte zusätzlich die Bedingung (2.30). Des weiteren gelte für die schwache Lösung $u \in H_*^1(\Omega)$ von (2.31) bzw. (2.32) $u \in C^2(\overline{\Omega})$, dann löst u auch die starke Formulierung (2.29).

Kommen wir nun zum Transmissionsproblem. In CARSTENSEN / STEPHAN 1995 [4] wird gezeigt, dass das Transmissionsproblem (TP), siehe Definition 2.12, äquivalent zu einem FEM/BEM-Kopplungsproblem ist. Das Kopplungsproblem (KP) ist dabei wie folgt definiert.

Definition 2.15 (Kopplungsproblem (KP)). Seien $f \in L^2(\Omega)$, $g_0 \in H^{1/2}(\Gamma)$ und $g_1 \in H^{-1/2}(\Gamma)$ gegeben. Das Kopplungsproblem lautet: Suche $(u, \phi) \in H^1(\Omega) \times H^{-1/2}(\Gamma)$, so dass

$$(\nabla u, \nabla v)_{L^2(\Omega)} + \langle \mathcal{W} \gamma_0 u + (\mathcal{K}' - \frac{1}{2}) \phi, \gamma_0 v \rangle_{\Gamma} = \langle f, v \rangle - \langle g_1 + \mathcal{W} g_0, \gamma_0 v \rangle_{\Gamma}, \quad (2.34)$$

$$\langle \psi, \mathcal{V} \phi - (\mathcal{K} - \frac{1}{2}) \gamma_0 u \rangle_{\Gamma} = \langle \psi, (\mathcal{K} - \frac{1}{2}) g_0 \rangle_{\Gamma}, \quad (2.35)$$

für alle $(v, \psi) \in H^1(\Omega) \times H^{-1/2}(\Gamma)$ gilt.

Für den Fall $d = 2$ skalieren wir das Gebiet Ω so, dass $\text{diam}(\Omega) < 1$, damit der Operator $\mathcal{V}$ nach Lemma 2.14 positiv definit ist. Mit dieser Voraussetzung gilt die Äquivalenz von (KP) und (TP), wie nachfolgender Satz besagt.

Satz 2.23 (CARSTENSEN / STEPHAN 1995 [4, Satz 1]). *Seien $f \in L^2(\Omega)$, $g_0 \in H^{1/2}(\Gamma)$ und $g_1 \in H^{-1/2}(\Gamma)$ gegeben, dann ist (KP) und (TP) äquivalent in folgendem Sinne.*

Ist $(u^{int}, u^{ext}) \in H^1(\Omega) \times H_{lok}^1(\Omega^{ext})$ eine Lösung von (TP), dann löst das Paar $(u^{int}, \phi) \in H^1(\Omega) \times H^{-1/2}(\Gamma)$ (KP) mit $\phi := \gamma_1^{ext} u^{ext}$.

Gilt umgekehrt dass $(u^{int}, \phi) \in H^1(\Omega) \times H^{-1/2}(\Gamma)$ eine Lösung von (KP) ist, dann ist $(u^{int}, u^{ext}) \in H^1(\Omega) \times H_{lok}^1(\Omega^{ext})$ eine Lösung von (TP) mit

$$u^{ext}(x) := \tilde{\mathcal{K}}(\gamma_0 u + g_0)(x) - \tilde{\mathcal{V}}\phi(x) \quad \text{für } x \in \Omega^{ext}. \quad (2.36)$$

Dabei ist (2.36) die dritte Green'sche Formel, wie sie bereits in Abschnitt 2.3 angeführt wurde.

Satz 2.24 (CARSTENSEN / STEPHAN 1995 [4, Korollar 2]). *(KP) ist eindeutig lösbar. Aus der Äquivalenz von (KP) und (TP) folgt daher auch, dass (TP) eindeutig lösbar ist.*

Das magnetostatische Problem in der schwachen Formulierung, siehe Gleichungssystem (2.13), besitzt diesem Resultat zufolge genau eine Lösung.

Kapitel 3

Numerische Verfahren

In diesem Kapitel behandeln wir die für das numerische Lösen der magnetostatischen Gleichungen wichtige Methode der Finiten Elemente. Zunächst werden wir im ersten Abschnitt die wichtigsten Resultate aus der Theorie zitieren. Dabei werden wir uns im Speziellen an das Vorlesungsskript von PRAETORIUS 2008 [17] bzw. an das Buch von BRAESS 2007 [3] halten. Der letzte Abschnitt handelt dann von Interpolationsoperatoren, allen voran werden wir hier den Scott-Zhang Operator aus SCOTT/ZHANG 1990 [19] näher untersuchen.

3.1 Finite Elemente Methode

3.1.1 Galerkin Verfahren

Die Finite Elemente Methode ist ein spezielles Galerkin Verfahren. Im Folgenden werden wir die wichtigsten Eigenschaften von Galerkin Verfahren vorstellen. Zunächst betrachten wir folgendes abstraktes Setting. Es sei $\mathcal{H}$ ein Hilbertraum mit Skalarprodukt $\langle \cdot, \cdot \rangle$ und induzierter Norm $\|\cdot\|_{\mathcal{H}} := \sqrt{\langle \cdot, \cdot \rangle}$. Des weiteren notieren wir mit $\langle\langle \cdot, \cdot \rangle\rangle$ ein äquivalentes Skalarprodukt auf $\mathcal{H}$, d.h. es existieren Konstanten $\alpha, \beta > 0$, so dass

$$\alpha\|v\|_{\mathcal{H}} \leq \|v\| \leq \beta\|v\|_{\mathcal{H}} \quad \text{für alle } v \in \mathcal{H} \quad (3.1)$$

gilt, wobei $\|v\| := \sqrt{\langle\langle v, v \rangle\rangle}$ die durch $\langle\langle \cdot, \cdot \rangle\rangle$ induzierte Norm ist. Üblicherweise wird $\|\cdot\|$ als Energienorm und $\langle\langle \cdot, \cdot \rangle\rangle$ als Energieskalarprodukt bezeichnet.

Ein abstraktes Variationsproblem liest sich wie folgt: Für gegebenes $F \in \mathcal{H}^*$ finde $u \in \mathcal{H}$ mit

$$\langle\langle u, v \rangle\rangle = F(v) \quad \text{für alle } v \in \mathcal{H}. \quad (3.2)$$

Der Satz von Riesz (ALT 2006 [1, Satz 4.1]) zeigt eindeutige Lösbarkeit dieses Problems. Das Grundprinzip von Galerkin Verfahren besteht nun darin, den unendlichdimensionalen Hilbertraum $\mathcal{H}$ in (3.2) durch einen endlichdimensionalen Unterraum $X_h \subseteq \mathcal{H}$ zu ersetzen. Somit lautet das diskretisierte Problem: Finde $u_h \in X_h$ mit

$$\langle\langle u_h, v_h \rangle\rangle = F(v_h) \quad \text{für alle } v_h \in X_h. \quad (3.3)$$

Bemerkung 3.1. Da X_h ein endlicher Teilraum des Hilbertraums $\mathcal{H}$ ist, gilt insbesondere, dass X_h abgeschlossen und somit wieder ein Hilbertraum ist. Der Satz von Riesz zeigt damit wieder eindeutige Lösbarkeit von (3.3).

Für den Rest dieses Abschnitts bezeichne u die eindeutige Lösung von (3.2) und u_h die eindeutige Lösung von (3.3).

Man nennt u_h auch Galerkin Lösung. Die Funktion u_h ist durch die wichtige Galerkin Orthogonalität

$$\langle\langle u - u_h, v_h \rangle\rangle = 0 \quad \text{für alle } v_h \in X_h \quad (3.4)$$

charakterisiert, welche aus (3.2) und (3.3) folgt.

Mit Hilfe der Galerkin Orthogonalität lässt sich das Céa-Lemma zeigen. Dieses Lemma sagt aus, dass der Galerkin Fehler $\|u - u_h\|_{\mathcal{H}}$ quasi-optimal ist, d.h. der Fehler sich wie die Bestapproximation bis auf multiplikative Konstanten verhält.

Lemma 3.1 (Céa-Lemma, PRAETORIUS 2008 [17, Lemma 1.6]). *Der Galerkin Fehler ist quasi-optimal, d.h. er erfüllt*

$$\|u - u_h\|_{\mathcal{H}} \leq \frac{\beta}{\alpha} \min_{v_h \in X_h} \|u - v_h\|_{\mathcal{H}} \quad \text{für alle } u \in \mathcal{H},$$

wobei $\alpha, \beta > 0$ die Normäquivalenzkonstanten aus (3.1) sind.

Die Galerkin Lösung u_h ist die Bestapproximation der exakten Lösung u in Bezug auf die Energienorm, d.h. es gilt

$$\|u - v_h\| = \min_{v_h \in X_h} \|u - v_h\| \quad \text{für alle } u \in \mathcal{H}.$$

Die Galerkin Methode hat den Vorteil, dass man Konvergenz für jede exakte Lösung $u \in \mathcal{H}$ zeigen kann.

Satz 3.2 (PRAETORIUS 2008 [17, Proposition 1.7]). *Sei D ein dichter Teilraum von $\mathcal{H}$ mit Approximationseigenschaft, d.h. es gilt*

$$\lim_{h \rightarrow 0} \min_{v_h \in X_h} \|v - v_h\|_{\mathcal{H}} = 0 \quad \text{für alle } v \in D.$$

Dann konvergiert die Folge der Galerkin Lösungen gegen die exakte Lösung u , d.h. es gilt

$$\lim_{h \rightarrow 0} \|u - u_h\|_{\mathcal{H}} = 0.$$

Der letzte Satz sagt jedoch noch nicht aus, wie schnell der Galerkin Fehler gegen Null strebt. Um Aussagen über die Konvergenzgeschwindigkeit zu erhalten, müssen spezielle Galerkin Methoden betrachtet werden. Wir werden im Folgenden die Theorie der Finiten Elemente Methode (FEM) behandeln.

Zuvor zitieren wir noch ein allgemeines Resultat über Galerkin Verfahren.

Satz 3.3 (PRAETORIUS 2008 [17, Satz 1.4]). *Sei $N \in \mathbb{N}$ die Dimension von X_h und $\{\phi_1, \phi_2, \dots, \phi_N\}$ eine Basis von X_h . Wir definieren die so genannte Galerkin Matrix $\mathbf{A} \in \mathbb{R}^{N \times N}$ über*

$$\mathbf{A}_{ij} := \langle \phi_j, \phi_i \rangle,$$

und den Vektor $\mathbf{b} \in \mathbb{R}^N$ über

$$\mathbf{b}_i := F(\phi_i).$$

Dann ist $\mathbf{A}$ symmetrisch und positiv definit, daher auch regulär. Des weiteren gilt $u_h = \sum_{i=1}^N \mathbf{x}_i \phi_i$, wobei der Vektor $\mathbf{x} \in \mathbb{R}^N$ die Lösung des linearen Gleichungssystems $\mathbf{A}\mathbf{x} = \mathbf{b}$ darstellt.

Bemerkung 3.2. Die Matrix $\mathbf{A}$ wird auch Steifigkeitsmatrix genannt, der Vektor $\mathbf{b}$ Lastvektor.

3.1.2 P1-Finite Elemente Methode

Eine Menge $T \subseteq \mathbb{R}^2$ wird nicht entartetes Dreieck genannt, falls es Knoten $x_T, y_T, z_T \in \mathbb{R}^2$ gibt, mit $T = \text{conv}\{x_T, y_T, z_T\}$ und $|T| > 0$. Eine Menge $T \subseteq \mathbb{R}^3$ wird nicht entarteter Tetraeder genannt, falls Knoten $a_T, b_T, c_T, d_T \in \mathbb{R}^3$ existieren, mit $T = \text{conv}\{a_T, b_T, c_T, d_T\}$ und $|T| > 0$. Wir bezeichnen mit

$$\mathcal{M}_T := \begin{cases} \{x_T, y_T, z_T\} & \text{falls } d = 2, \\ \{a_T, b_T, c_T, d_T\} & \text{falls } d = 3 \end{cases}$$

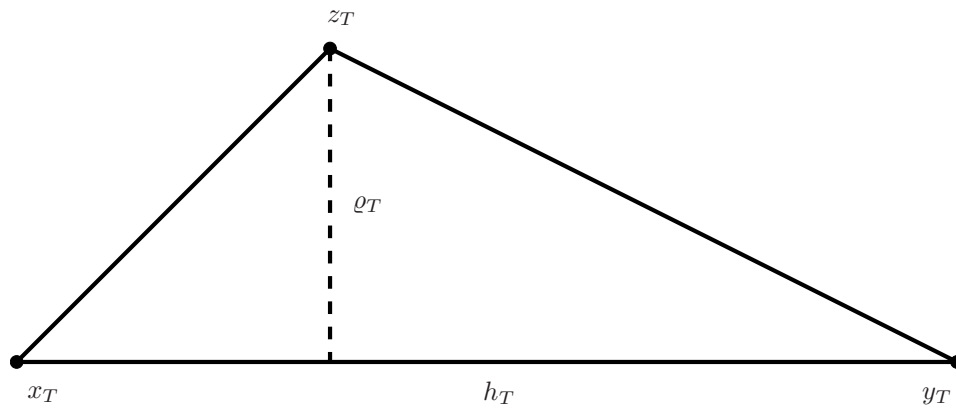


Abbildung 3.1: Ein Dreieck T wird durch seine drei Knoten $x_T, y_T, z_T \in \mathbb{R}^3$ beschrieben. Die Größe h_T bezeichnet die längste Kante und ϱ_T die Höhe darauf.

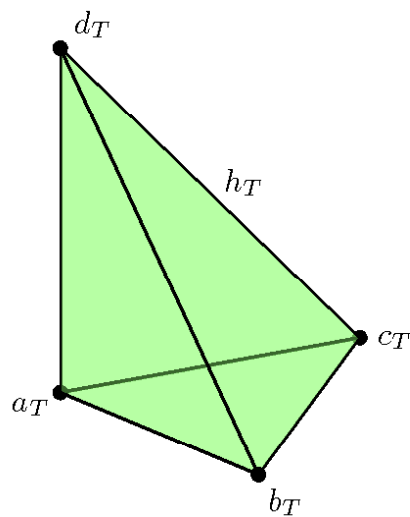


Abbildung 3.2: Ein Tetraeder T wird durch seine vier Knoten $a_T, b_T, c_T, d_T \in \mathbb{R}^3$ beschrieben. Die Größe h_T bezeichnet dabei den Durchmesser von T , d.h. die längste Kante von T .

die Menge der Knoten von T und mit

$$\mathcal{E}_T := \begin{cases} \{\text{conv}\{x_T, y_T\}, \text{conv}\{y_T, z_T\}, \text{conv}\{x_T, z_T\}\} & \text{falls } d = 2, \\ \{\text{conv}\{a_T, b_T\}, \text{conv}\{a_T, c_T\}, \text{conv}\{a_T, d_T\}, \\ \text{conv}\{b_T, c_T\}, \text{conv}\{b_T, d_T\}, \text{conv}\{c_T, d_T\}\} & \text{falls } d = 3 \end{cases}$$

die Menge der Kanten von T . Der Durchmesser von T wird mit

$$h_T := \text{diam}(T)$$

bezeichnet und stimmt mit der Länge der längsten Kante von T überein. Des weiteren definieren wir noch die Kantenlänge über

$$h_E := \text{diam}(E)$$

für alle Kanten $E \in \mathcal{E}_T$.

Die Höhe über die längste Seite eines Dreiecks T wird mit ϱ_T bezeichnet. Damit ergibt für das Volumen $|T|$ eines Dreiecks

$$|T| = \frac{h_T \varrho_T}{2}. \quad (3.5)$$

Im Fall $d = 3$ bezeichnen wir mit

$$\mathcal{F}_T := \{\text{conv}\{a_T, b_T, c_T\}, \text{conv}\{a_T, b_T, d_T\}, \text{conv}\{a_T, c_T, d_T\}, \text{conv}\{b_T, c_T, d_T\}\}$$

die Menge der Seitenflächen von T . Das Volumen eines Tetraeders lässt sich über

$$|T| = \frac{1}{6} |((b_T - a_T) \times (c_T - a_T)) \cdot (d_T - a_T)| \quad (3.6)$$

berechnen.

Ein wichtiges Beispiel für ein Dreieck bzw. einen Tetraeder ist das Referenzelement

$$T_{ref} := \begin{cases} \text{conv}\{(0, 0), (1, 0), (0, 1)\} & \text{falls } d = 2, \\ \text{conv}\{(0, 0, 0), (1, 0, 0), (0, 1, 0), (0, 0, 1)\} & \text{falls } d = 3. \end{cases}$$

Das Volumen des Referenzelements ergibt sich nach (3.5) und (3.6) zu

$$|T_{ref}| = \begin{cases} \frac{1}{2} & \text{falls } d = 2, \\ \frac{1}{6} & \text{falls } d = 3. \end{cases}$$

Wir kommen jetzt zum fundamentalen Begriff der Triangulierung.

Definition 3.1 (Triangulierung). *Eine Menge $\mathcal{T}$ ist eine Triangulierung des Gebiets Ω , genau dann wenn gilt:*

- $\mathcal{T}$ ist entweder eine endliche Menge von Dreiecken ($d = 2$) oder eine endliche Menge von Tetraedern ($d = 3$),
- der Abschluss von Ω wird durch $\mathcal{T}$ überdeckt, d.h. $\overline{\Omega} = \bigcup_{T \in \mathcal{T}} T$,
- für alle $T, T' \in \mathcal{T}$ mit $T \neq T'$ gilt $|T \cap T'| = 0$, d.h. die Schnittmenge besitzt Maß 0.

In dieser Arbeit werden wir uns ausschließlich mit regulären Triangulierungen auseinandersetzen.

Definition 3.2 (Reguläre Triangulierung). Eine Triangulierung $\mathcal{T}$ von Ω wird regulär (im Sinne von Ciarlet) genannt, falls die Schnittmenge zweier Elemente $T, T' \in \mathcal{T}$ mit $T \neq T'$ entweder

- leer, oder
- ein gemeinsamer Knoten, d.h. $T \cap T' = \{z\}$ und $z \in \mathcal{M}_T \cap \mathcal{M}_{T'}$, oder
- eine gemeinsame Kante, d.h. $E := T \cap T' \in \mathcal{E}_T \cap \mathcal{E}_{T'}$, oder
- (im Falle $d = 3$) eine gemeinsame Seitenfläche, d.h. $F := T \cap T' \in \mathcal{F}_T \cap \mathcal{F}_{T'}$

ist.

Wir bezeichnen mit $\mathcal{M} := \bigcup \{z \in \mathcal{M}_T \mid T \in \mathcal{T}\}$ die Menge der Knoten, mit $\mathcal{E} := \bigcup \{E \in \mathcal{E}_T \mid T \in \mathcal{T}\}$ die Menge der Kanten, und mit $\mathcal{F} := \bigcup \{F \in \mathcal{F}_T \mid T \in \mathcal{T}\}$ die Menge der Seitenflächen der Triangulierung $\mathcal{T}$. Des weiteren definieren wir die lokale Netzweitenfunktion $h \in L^\infty(\Omega)$ und die Volumensfunktion $V \in L^\infty(\Omega)$ über

$$h|_T := h_T$$

und

$$V|_T := |T|$$

für alle $T \in \mathcal{T}$.

Wir benötigen den Raum der stückweise polynomialen Funktionen.

Definition 3.3. Sei $\mathcal{T}$ eine reguläre Triangulierung. Für $n \in \mathbb{N}$ definieren wir den Raum $\mathcal{P}^n(\mathcal{T})$ über

$$\mathcal{P}^n(\mathcal{T}) := \{v \in L^2(\Omega) \mid v|_T \text{ ist Polynom vom Grad } \leq n \text{ für alle } T \in \mathcal{T}\}$$

Definition 3.4. Die L^2 -Orthogonalprojektion auf den Raum der $\mathcal{T}$ -stückweise konstanten Funktionen wird definiert als

$$\begin{aligned} \Pi_h^0 : L^2(\Omega) &\longrightarrow \mathcal{P}^0(\mathcal{T}) \\ v &\longmapsto v_{\mathcal{T}} \end{aligned}$$

mit

$$v_{\mathcal{T}}|_T := \frac{1}{|T|} \int_T v \, dx \quad \text{für alle } T \in \mathcal{T}.$$

Mit obiger Definition gilt für $v \in L^2(\Omega)$ und $p \in \mathcal{P}^0(\mathcal{T})$

$$(v - v_{\mathcal{T}}, p)_{L^2(\Omega)} = \sum_{T \in \mathcal{T}} \left(v - \frac{1}{|T|} \int_T v \, dx, p \right)_{L^2(T)} = \sum_{T \in \mathcal{T}} \left(\int_T v \, dx - \int_T v \, dx \right) p|_T = 0.$$

Eine Basis von $\mathcal{P}^0(\mathcal{T})$ ist durch die Menge $\mathcal{X} := \{\chi_T \mid T \in \mathcal{T}\}$ mit

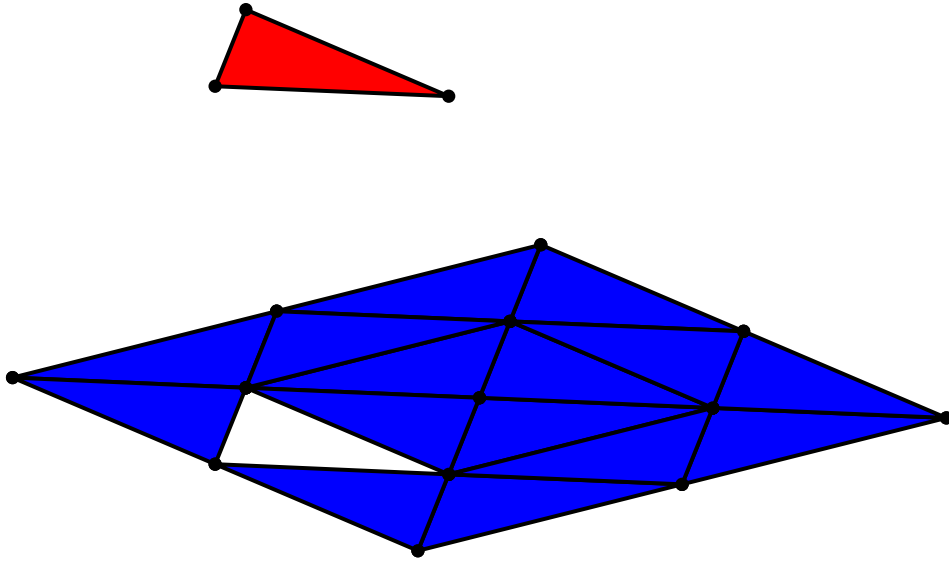
$$\chi_T(x) := \begin{cases} 1 & \text{für } x \in T \\ 0 & \text{sonst} \end{cases}$$

gegeben. Ein Beispiel solch einer Funktion ist Abbildung 3.3 zu sehen.

Definition 3.5. Für eine reguläre Triangulierung $\mathcal{T}$ definieren wir den endlichdimensionalen Raum

$$\mathcal{S}^1(\mathcal{T}) := \{v_h \in C(\Omega) \mid v_h|_T \in \mathcal{P}^1(T) \quad \text{für alle } T \in \mathcal{T}\}$$

aller $\mathcal{T}$ -stückweise affinen und global stetigen Funktionen.

Abbildung 3.3: Beispiel einer stückweisen konstanten Basisfunktion χ_T .

Satz 3.4 (PRAETORIUS 2008 [17, Proposition 3.1]). Sei $\mathcal{T}$ eine reguläre Triangulierung, dann gilt

- $\mathcal{S}^1(\mathcal{T})$ ist ein N -dimensionaler Unterraum von $H^1(\Omega)$ und $N = \#\mathcal{M}$ bezeichnet die Anzahl der Knoten.
- Für jeden Knoten $z \in \mathcal{M}$ existiert eine eindeutige Hutfunktion

$$\eta_z \in \mathcal{S}^1(\mathcal{T}) \quad \text{mit} \quad \eta_z(z') = \delta_{zz'} \quad \text{für alle } z' \in \mathcal{M}.$$

- Die Menge $\mathcal{B} := \{\eta_z \mid z \in \mathcal{M}\}$ ist eine Basis von $\mathcal{S}^1(\mathcal{T})$. Wir nennen diese Basis auch nodale Basis von $\mathcal{S}^1(\mathcal{T})$.

Um die diskreten Variationsformulierungen von (2.27) bzw. (2.32) aufzustellen, benötigen wir noch die Räume $\mathcal{S}_0^1(\mathcal{T})$ und $\mathcal{S}_*^1(\mathcal{T})$.

Satz 3.5 (PRAETORIUS 2008 [17, Korollar 3.2, Korollar 3.3]). Sei $\mathcal{T}$ eine reguläre Triangulierung von Ω . Dann ist der Raum

$$\mathcal{S}_0^1(\mathcal{T}) := \{v_h \in \mathcal{S}^1(\mathcal{T}) \mid \forall z \in \mathcal{M} \cap \Gamma : v_h(z) = 0\}$$

ein endlichdimensionaler Teilraum von $H_0^1(\Omega)$ mit der Dimension $\#\{z \in \mathcal{M} \mid z \notin \Gamma\}$.

Der Raum

$$\mathcal{S}_*^1(\mathcal{T}) := \left\{ v_h \in \mathcal{S}^1(\mathcal{T}) \mid \int_{\Omega} v_h \, dx = 0 \right\}$$

ist ein endlichdimensionaler Teilraum von $H_*^1(\Omega)$ mit Dimension $\#\mathcal{M} - 1$.

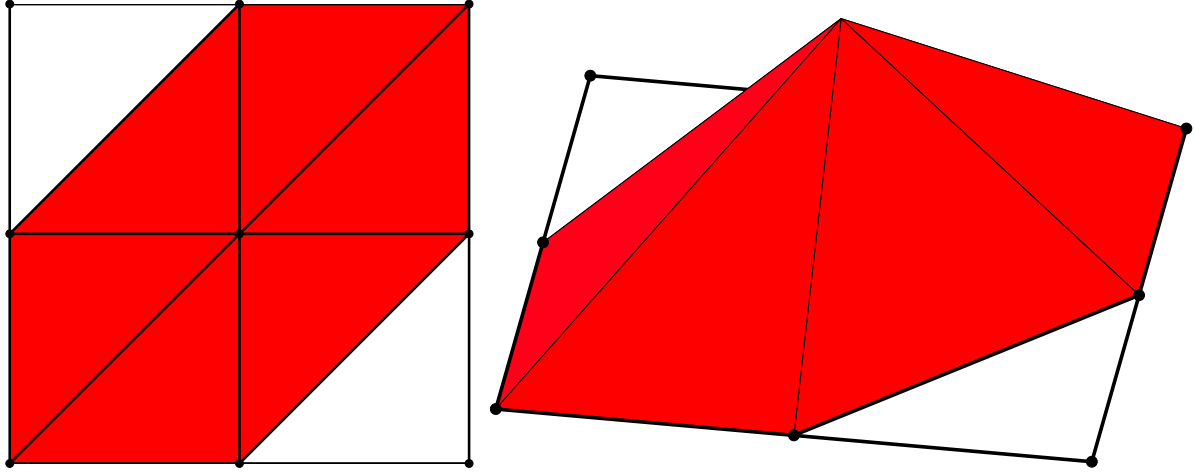


Abbildung 3.4: Beispiel einer Hutfunktion. Im linken Bild ist der Träger der Hutfunktion in rot eingefärbt. Das rechte Bild zeigt die Hutfunktion zum mittleren Knoten der Triangulierung.

Konvention 3.6. Wir werden in dieser Arbeit durchgehend mit $\mathcal{T}$ eine reguläre Triangulierung von Ω bezeichnen.

Die Triangulierung $\mathcal{T}$ des Gebiets Ω induziert in natürlicher Weise eine Triangulierung $\mathcal{T}_\Gamma$ des Randes Γ . Wir bezeichnen mit

$$\mathcal{M}_\Gamma := \{z \in \mathcal{M} \mid \{z\} \cap \Gamma = \{z\}\}$$

die Menge aller Knoten, die auf dem Rand liegen, mit

$$\mathcal{E}_\Gamma := \{E \in \mathcal{E} \mid E \cap \Gamma = E\}$$

die Menge aller Kanten, die auf dem Rand liegen und mit

$$\mathcal{F}_\Gamma := \{F \in \mathcal{F} \mid F \cap \Gamma = F\}$$

die Menge aller Seitenflächen, die auf dem Rand liegen. Es gilt

$$\mathcal{T}_\Gamma = \begin{cases} \mathcal{E}_\Gamma & \text{falls } d = 2, \\ \mathcal{F}_\Gamma & \text{falls } d = 3. \end{cases}$$

Die Definitionen der Funktionenräume $\mathcal{S}^1(\mathcal{T})$, $\mathcal{P}^n(\mathcal{T})$ etc. für die Triangulierung $\mathcal{T}$ des Gebiets Ω können für die Triangulierung $\mathcal{T}_\Gamma$ des Randes Γ übernommen werden. Die Definition des Raumes $\mathcal{S}_0^1(\mathcal{T}_\Gamma)$ macht dabei keinen Sinn.

Bemerkung 3.3. Sei $z \in \mathcal{M}_\Gamma$, dann existiert für jede Hutfunktion $\hat{\eta}_z \in \mathcal{S}^1(\mathcal{T}_\Gamma)$ genau eine Basisfunktion $\eta_z \in \mathcal{S}^1(\mathcal{T})$ mit

$$\gamma_0(\eta_z) = \hat{\eta}_z.$$

Die P1-Finite Elemente Methode besteht nun darin, die unendlichdimensionalen Räume $H_0^1(\Omega)$, $H_*^1(\Omega)$ in den Variationsformulierungen (2.27) und (2.32), durch die eben vorgestellten Räume $\mathcal{S}_0^1(\mathcal{T})$ und $\mathcal{S}_*^1(\mathcal{T})$ zu ersetzen. Bezeichnet $\langle \cdot, \cdot \rangle$ das Skalarprodukt auf $H_0^1(\Omega)$ bzw. $H_*^1(\Omega)$, dann ist nach der Friedrich'schen bzw. Poincaré'schen Ungleichung (siehe PRAETORIUS 2008 [17, Korollar 2.15 bzw. Korollar 2.10]) die Bilinearform $(\nabla u, \nabla v)_{L^2(\Omega)}$ ein auf $H_0^1(\Omega)$ bzw. $H_*^1(\Omega)$ äquivalentes Skalarprodukt.

Wir wollen nun die diskreten Variationsformulierungen von (2.27) und (2.32) aufstellen. Zuerst betrachten wir das Neumann Problem wie es in Satz 2.22 beschrieben wird. Sei also $f \in \tilde{H}^{-1}(\Omega)$ und $g_N \in H^{-1/2}(\Gamma)$. Wir suchen eine Funktion $u \in H_*^1(\Omega)$, welche

$$(\nabla u, \nabla v)_{L^2(\Omega)} = \langle f, v \rangle \quad \text{für alle } v \in H_*^1(\Omega)$$

erfüllt. Die P1-FEM dieser Variationsformulierung lautet: Finde $U \in \mathcal{S}_*^1(\mathcal{T})$ mit

$$(\nabla U, \nabla V)_{L^2(\Omega)} = \langle f, V \rangle \quad \text{für alle } V \in \mathcal{S}_*^1(\mathcal{T}). \quad (3.7)$$

Die schwache Formulierung des Dirichlet Problem (vgl. Satz 2.21) lautet bei gegebenem $f \in \tilde{H}^{-1}(\Omega)$ und $g_D \in H^{1/2}(\Gamma)$ wie folgt: Finde $u \in H^1(\Omega)$ mit

$$\begin{aligned} \gamma_0 u &= u_D, \\ (\nabla u, \nabla v)_{L^2(\Omega)} &= \langle f, v \rangle \quad \text{für alle } v \in H_0^1(\Omega). \end{aligned} \quad (3.8)$$

Satz 3.6 (PRAETORIUS 2008 [17, Proposition 3.19]). *Sei $\hat{g}_D \in H^1(\Omega)$ eine beliebige Fortsetzung der Dirichlet Daten $g_D \in H^{1/2}(\Gamma)$. Bei gegebenem $f \in \tilde{H}^{-1}(\Omega)$ existiert ein eindeutiges $u_0 \in H_0^1(\Omega)$, so dass*

$$(\nabla u_0, \nabla v)_{L^2(\Omega)} = \langle f, v \rangle - (\nabla \hat{g}_D, \nabla v)_{L^2(\Omega)} \quad \text{für alle } v \in H_0^1(\Omega) \quad (3.9)$$

gilt. Eine Funktion $u \in H^1(\Omega)$ mit $\gamma_0 u = g_D$ ist genau dann eine Lösung der schwachen Formulierung (3.8), falls $u_0 := u - \hat{g}_D \in H_0^1(\Omega)$ eine Lösung von (3.9) ist. Die Funktion u hängt dabei nicht von der Wahl von $\hat{g}_D$ ab.

Um die P1-FEM des Dirichlet Problems zu erhalten, ersetzt man den Raum $H_0^1(\Omega)$ in (3.9) durch den endlichdimensionalen Raum $\mathcal{S}_0^1(\mathcal{T})$ und u_0 durch U_0 . Man beachte jedoch, dass im Allgemeinen die Lösung $U = U_0 + \hat{g}_D$ nicht in $\mathcal{S}^1(\mathcal{T})$ liegt, da die Fortsetzung $\hat{g}_D$ der Dirichlet Daten im Allgemeinen nicht in $\mathcal{S}^1(\mathcal{T})$ liegt. Wir werden daher die Vorgangsweise wie folgt modifizieren.

- Diskretisiere die Dirichlet Daten $g_D \in H^{1/2}(\Gamma)$ mit $G_D \in \mathcal{S}^1(\mathcal{T}_\Gamma)$.
- Konstruiere eine Erweiterung $\hat{G}_D \in \mathcal{S}^1(\mathcal{T})$ von $G_D = \gamma_0(\hat{G}_D)$.
- Finde $U_0 \in \mathcal{S}_0^1(\mathcal{T})$ mit

$$(\nabla U_0, \nabla V)_{L^2(\Omega)} = \langle f, V \rangle - (\nabla \hat{G}_D, \nabla V)_{L^2(\Omega)} \quad \text{für alle } V \in \mathcal{S}_0^1(\mathcal{T}). \quad (3.10)$$

- $U := U_0 + \hat{G}_D \in \mathcal{S}^1(\mathcal{T})$ definieren wir als die Approximationslösung von u .

Die oben beschriebene Vorgangsweise nennen wir auch die P1-FEM der schwachen Formulierung (3.8). Wir zitieren noch ein wichtiges Resultat, dass ähnlich zum bereits angeführten Céa Lemma ist.

Lemma 3.7 (Céa, PRAETORIUS 2008 [17, Lemma 3.20]). *Sei $u \in H^1(\Omega)$ die Lösung von (3.8) und $\hat{G}_D \in \mathcal{S}^1(\mathcal{T})$ die Fortsetzung der approximierten Dirichletdaten $G_D = \gamma_0(\hat{G}_D) \in \mathcal{S}^1(\mathcal{T}_\Gamma)$. Sei weiters U die eindeutige Lösung von (3.10), dann gilt, dass U quasi-optimal ist und zwar in dem Sinne, dass eine Konstante $C > 0$ existiert, für die gilt*

$$\|u - U\|_{H^1(\Omega)} \leq C \min_{V \in \mathcal{S}_0^1(\mathcal{T})} \|u - (V + \hat{G}_D)\|_{H^1(\Omega)}.$$

Dabei hängt die Konstante C nur von Ω und Γ ab.

Die Formregularitätskonstante

$$\sigma(\mathcal{T}) := \|h^d/V\|_{L^\infty(\Omega)} = \max_{T \in \mathcal{T}} \frac{h_T^d}{|T|} \quad (3.11)$$

beeinflusst alle Fehlerabschätzungen, weshalb man Verfeinerungen des Netzes so wählt, dass $\sigma(\mathcal{T})$ konstant ist oder zumindest uniform beschränkt bleibt.

In dieser Arbeit werden wir uns nur mit uniformen Verfeinerungen (Rot-Verfeinerungen) beschäftigen. Sei $\mathcal{T}^{alt}$ eine reguläre Triangulierung von Ω . Für $d = 2$ erhält man eine uniform verfeinerte Triangulierung $\mathcal{T}^{neu}$, indem man jedes Dreieck T in vier ähnliche Dreiecke $T_1, T_2, T_3, T_4 \in \mathcal{T}^{neu}$ unterteilt (vgl. Abbildung 3.5). Für jeden Knoten $z \in \mathcal{M}^{neu}$ gilt dann entweder $z \in \mathcal{M}^{alt}$ oder z ist Mittelpunkt einer Kante $E \in \mathcal{E}^{alt}$. Für die neue Triangulierung gilt dann:

- $\mathcal{T}^{neu}$ ist wieder regulär.
- Die lokale Netzweitenfunktion erfüllt $h^{neu} = h^{alt}/2$ und die Volumensfunktion erfüllt

$$V^{neu} = \frac{V^{alt}}{4}.$$

- Eine wichtige Beobachtung ist, dass

$$\sigma(\mathcal{T}^{neu}) = \sigma(\mathcal{T}^{alt})$$

gilt.

Wir betrachten nun eine reguläre Verfeinerung eines Tetraeders T , wobei zu beachten ist, dass nicht alle aus der Verfeinerung entstandenen Tetraeder ähnlich zu T sind (vgl. Abbildung 3.6). Wir identifizieren das Tetraeder $T \in \mathcal{T}^{alt}$ durch die geordnete Liste $[z_1, z_2, z_3, z_4]$ seiner vier Knoten. Mit $z_{ij} = \frac{1}{2}(z_i + z_j)$ für $1 \leq i < j \leq 4$ bezeichnen wir den Mittelpunkt der Kante, die die zwei Knoten z_i, z_j verbindet. Nach VERFÜRTH 1996 [22, Abschnitt 4.1] wird $T \in \mathcal{T}^{alt}$ in 8 Tetraeder $T_1, T_2, T_3, T_4, T_5, T_6, T_7, T_8 \in \mathcal{T}^{neu}$ mit

$$\begin{aligned} T_1 &:= [z_1, z_{12}, z_{13}, z_{14}], & T_2 &:= [z_{12}, z_2, z_{23}, z_{24}], \\ T_3 &:= [z_{13}, z_{23}, z_3, z_{34}], & T_4 &:= [z_{14}, z_{24}, z_{34}, z_4], \\ T_5 &:= [z_{12}, z_{13}, z_{14}, z_{24}], & T_6 &:= [z_{12}, z_{13}, z_{23}, z_{24}], \\ T_7 &:= [z_{13}, z_{23}, z_{24}, z_{34}], & T_8 &:= [z_{13}, z_{14}, z_{24}, z_{34}], \end{aligned}$$

unterteilt. Dabei sind T_1, T_2, T_3, T_4 im Gegensatz zu T_5, T_6, T_7, T_8 ähnlich zum Ausgangstetraeder T . Die Formregularität bleibt bei dieser regulären Verfeinerung erhalten.

Nach dem Sobolev'schen Einbettungssatz (Satz 2.6) besitzen Funktionen $u \in H^2(\Omega)$ einen stetigen Repräsentanten aus $C(\overline{\Omega})$. Die Auswertung von $u \in H^2(\Omega)$ an den Knoten $z \in \mathcal{M}$ ist also wohldefiniert.

Definition 3.7 (Nodaler Interpolant). Sei $u \in H^2(\Omega)$, dann wird der nodale Interpolant $I_h : H^2(\Omega) \rightarrow \mathcal{S}^1(\mathcal{T})$ über

$$I_h u := \sum_{z \in \mathcal{M}} u(z) \eta_z \in \mathcal{S}^1(\mathcal{T})$$

definiert.

Die Beweise der beiden folgenden Aussagen sind in [17] zu finden.

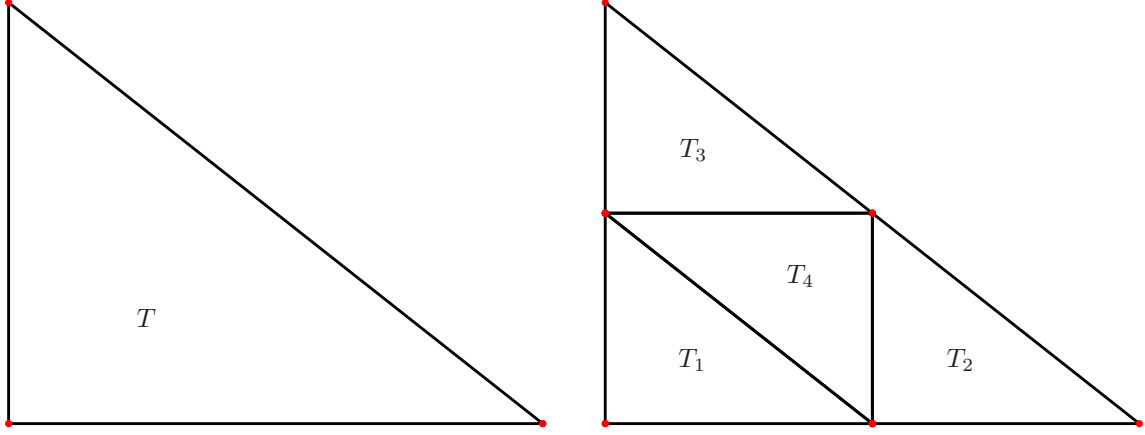


Abbildung 3.5: Rot-Verfeinerung: Ein Dreieck $T \in \mathcal{T}^{alt}$ wird in 4 ähnliche Dreiecke $T_1, T_2, T_3, T_4 \in \mathcal{T}^{neu}$ verfeinert. Für die verfeinerten Dreiecke gilt $h_T^{neu} = h_T^{alt}/2$ und $|\mathcal{T}^{neu}| = |\mathcal{T}^{alt}|/4$. Insbesondere gilt $\sigma(\mathcal{T}^{neu}) = \sigma(\mathcal{T}^{alt})$.

Satz 3.8 (Approximationssatz, PRAETORIUS 2008 [17, Satz 3.7]). Sei $u \in H^2(\Omega)$, dann gelten für jedes $\alpha \in \mathbb{R}$ die Fehlerabschätzungen

$$\|h^\alpha(u - I_h(u))\|_{L^2(\Omega)} \leq C \|h^{2+\alpha} D^2 u\|_{L^2(\Omega)}$$

und

$$\|h^\alpha \nabla(u - I_h(u))\|_{L^2(\Omega)} \leq C \sigma(\mathcal{T}) \|h^{1+\alpha} D^2 u\|_{L^2(\Omega)},$$

wobei die Konstante $C > 0$ unabhängig von u , $\mathcal{T}$ und Ω ist.

Mit dem Approximationssatz lässt sich eine Aussage über die Konvergenzgeschwindigkeit der P1-FEM treffen.

Korollar 3.9. Für $u \in H^2(\Omega) \cap H_0^1(\Omega)$ gilt $I_h u \in S_0^1(\mathcal{T})$ und daher

$$\min_{V \in S_0^1(\mathcal{T})} \|u - V\|_{H^1(\Omega)} \leq \|u - I_h u\|_{H^1(\Omega)} \leq C \sigma(\mathcal{T}) \|h D^2 u\|_{L^2(\Omega)}.$$

Für $u \in H^2(\Omega) \cap H_*^1(\Omega)$ gilt

$$\min_{V \in S_*^1(\mathcal{T})} \|u - V\|_{H^1(\Omega)} \leq \|u - I_h u\|_{H^1(\Omega)} \leq C \sigma(\mathcal{T}) \|h D^2 u\|_{L^2(\Omega)}.$$

Dabei hängt die Konstante $C > 0$ jeweils nur von $\text{diam}(\Omega)$ ab.

Bemerkung 3.4. Mit dem Céa Lemma und Korollar 3.9 folgt, dass für eine glatte exakte Lösung $u \in H^2(\Omega)$ des homogenen Dirichlet Problems bzw. Neumann Problems die P1-Finite Elemente Methode die Konvergenzordnung $\mathcal{O}(h)$ besitzt.

Bemerkung 3.5. Ein großer Vorteil der FEM besteht darin, dass die Galerkinmatrix $\mathbf{A}$ (siehe Satz 3.3) dünn besetzt ist, da jede nodale Basisfunktion nur lokalen Träger besitzt. Die Einträge der Matrix $\mathbf{A}$ ergeben sich zu

$$\mathbf{A}_{ij} = (\nabla \eta_j, \nabla \eta_i)_{L^2(\Omega)} = \int_{\Omega} \nabla \eta_j \cdot \nabla \eta_i \, dx = \int_S \nabla \eta_j \cdot \nabla \eta_i \, dx,$$

mit $S := \text{supp}(\eta_i) \cap \text{supp}(\eta_j)$.

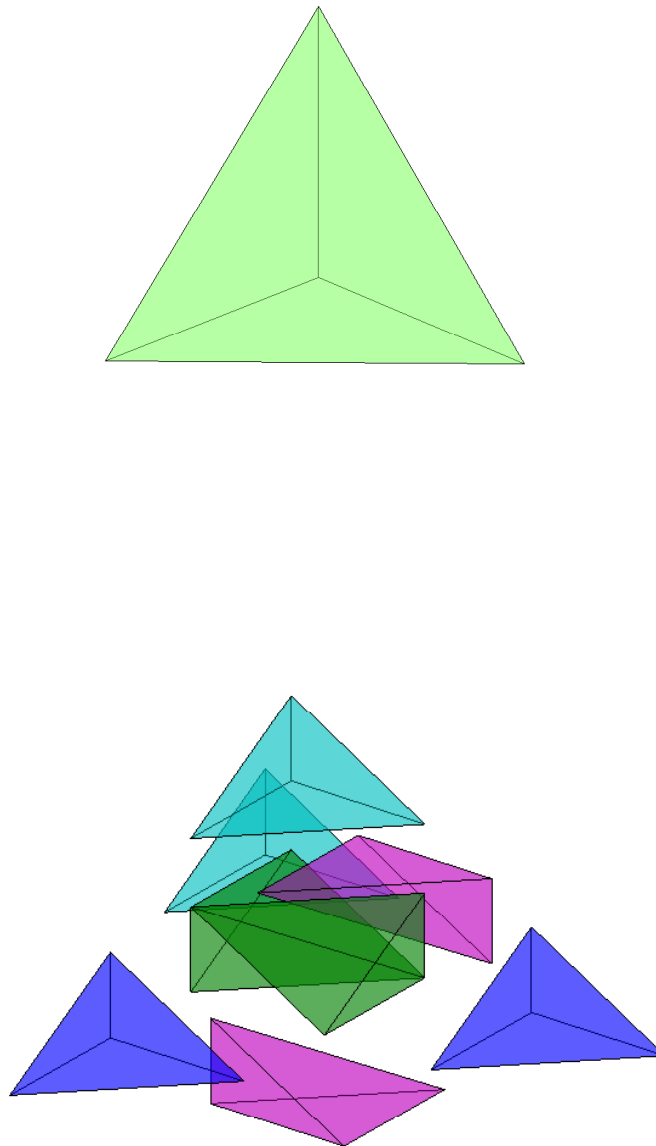


Abbildung 3.6: Rot-Verfeinerung: Ein Tetraeder T einer Triangulierung $\mathcal{T}^{alt}$ wird in 8 Tetraeder verfeinert. Die Formregularität bleibt bei dieser Verfeinerung dabei erhalten.

3.2 Interpolationsoperatoren

Für die numerische Implementierung benötigt man noch zusätzliche Approximationsoperatoren, die Funktionen aus Sobolev-Räumen auf affine, global stetige Funktionen abbilden. Ein Beispiel für solch einen Operator ist der nodale Interpolant. Der Nachteil dieses Operators ist, dass dieser nur auf $H^2(\Omega)$ definiert ist. Man kann aber auch Interpolationsoperatoren auf $H^1(\Omega)$ definieren. Dies sind Operatoren, bei denen man die Werte an den Knoten der affinen, global stetigen Funktion durch Mitteln bzw. gewichtetes Mitteln der zu approximierenden Funktion über lokale Bereiche definiert. Als Beispiele solcher Operatoren werden wir in diesem Abschnitt den Clément-Operator und den Scott-Zhang Operator aus SCOTT / ZHANG 1990 [19] definieren und deren wichtigsten Eigenschaften anführen.

Zunächst benötigen wir noch den Begriff des Patches.

Definition 3.8. Für jeden Knoten $z \in \mathcal{M}$ definiert man:

$$\tilde{\Omega}_z := \{T \in \mathcal{T} \mid z \in \mathcal{M}_T\}, \quad \Omega_z := \left\{x \in \mathbb{R}^d \mid \exists T \in \tilde{\Omega}_z : x \in T\right\}. \quad (3.12)$$

Ω_z wird als Patch zum Knoten z bezeichnet. In Abbildung 3.7 ist Ω_z für einen Knoten veranschaulicht.



Abbildung 3.7: Der grau eingefärbte Bereich im linken Bild ist der Patch Ω_z zum grün eingezeichneten Knoten z . Das rechte Bild zeigt die Wahl der Kanten beim Scott-Zhang Operator: Für Knoten im Inneren (grün) kann man eine beliebige anliegende Kante wählen (schwarz), für Knoten die am Rand liegen (rot) darf man nur eine anliegende Kante wählen die auch am Rand liegt (blau).

Bei der Definition des Clément-Operators halten wir uns an PRAETORIUS 2008 [17].

Definition 3.9 (Clément-Operator). Sei $\mathcal{T}$ eine Triangulierung von Ω und $\mathcal{M}_F := \mathcal{M} \setminus \Gamma$. Der Clément-Operator $J_h : H_0^1(\Omega) \rightarrow \mathcal{S}_0^1(\mathcal{T})$ wird definiert durch

$$J_h v := \sum_{z \in \mathcal{M}_F} v_z \eta_z \quad \text{mit} \quad v_z := \frac{1}{|\Omega_z|} \int_{\Omega_z} v \, dx, \quad (3.13)$$

wobei Ω_z der oben definierte Patch ist und die Menge der Funktionen $\{\eta_z\}_{z \in \mathcal{M}}$ stellt die nodale Basis von $\mathcal{S}^1(\mathcal{T})$ dar.

Bemerkung 3.6. Erweitert man den Definitionsbereich des Clément-Operators auf $L^2(\Omega)$ Funktionen, dann gilt für $v \in L^2(\Omega)$ und mit der L^2 -Orthogonalprojektion $\Pi_h^0 : L^2(\Omega) \rightarrow \mathcal{P}^0(\mathcal{T})$ auf den Raum der stückweise konstanten Funktionen $\mathcal{P}^0(\mathcal{T})$

$$\int_{\Omega_z} v \, dx = \sum_{T \in \tilde{\Omega}_z} \int_T v \, dx = \sum_{T \in \tilde{\Omega}_z} |T| (\Pi_h^0 v)|_T = \sum_{T \in \tilde{\Omega}_z} \int_T (\Pi_h^0 v)|_T \, dx = \int_{\Omega_z} \Pi_h^0 v \, dx. \quad (3.14)$$

Aus (3.13) folgt daher für $v \in L^2(\Omega)$

$$J_h v = J_h \Pi_h^0 v. \quad (3.15)$$

Als nächstes beschäftigen wir uns mit dem Scott-Zhang Operator. Es bezeichne im Folgenden σ_i eine Kante ($d = 2$) bzw. eine Seitenfläche ($d = 3$) eines Elements $T \in \mathcal{T}$. Nun wählt man zu jedem Knoten $z_i \in \mathcal{M}$ eine Kante bzw. eine Seitenfläche σ_i so, dass

$$z_i \in \sigma_i. \quad (3.16)$$

Gilt zusätzlich noch $z_i \in \Gamma$, dann wählt man σ_i so, dass

$$z_i \in \sigma_i \subset \Gamma. \quad (3.17)$$

Man beachte, dass die Wahl der σ_i nicht eindeutig ist! In Abbildung 3.7 ist die Wahl der σ_i im Fall $d = 2$ veranschaulicht. Die Menge der Funktionen $\{\phi_{ij}\}_{j=1}^d$ sei die nodale Basis von σ_i und $\{\psi_{ik}\}_{k=1}^d$ eine $L^2(\sigma_i)$ -duale Basis mit

$$\int_{\sigma_i} \psi_{ik} \phi_{ij} dx = \delta_{kj}, \quad (3.18)$$

wobei δ_{kj} das Kronecker Symbol ist. Die Funktionen $\{\phi_{ij}\}_{j=1}^d$ bilden also eine Basis von $\mathcal{P}^1(\sigma_i)$ und die Funktionen $\{\psi_{ij}\}_{j=1}^d$ eine Basis von $(\mathcal{P}^1(\sigma_i))^*$. Man kann die Funktion ϕ_{ij} auch als Linearkombination

$$\psi_{ij} = \sum_{k=1}^d a_{ij}^k \phi_{ik} \quad (3.19)$$

mit $a_{ij}^k \in \mathbb{R}$ darstellen. Diese Eigenschaft werden wir später benützen um die duale Basis auszurechnen.

Wir ordnen die Funktionen ϕ_{ij} und damit auch die Funktionen ψ_{ij} so an, dass $\phi_{ij}(z_i) = \delta_{1j}$, d.h. $\phi_{i1}(z_i) = 1$. Wir definieren $\phi_i := \phi_{i1}$ und $\psi_i := \psi_{i1}$. Sei $\{\eta_i\}_{i=1}^N$ die nodale Basis von $\mathcal{S}^1(\mathcal{T})$. Für eine Basisfunktion η_i gilt

$$\eta_i|_{\sigma_i} = \phi_i. \quad (3.20)$$

Daraus folgt

$$\int_{\sigma_i} \psi_i \eta_j dx = \delta_{ij} \quad \text{für alle } (i, j) \in \{1, 2, \dots, N\}^2. \quad (3.21)$$

In (3.21) geht auch ein, dass η_j genau dann Träger in σ_i hat, wenn z_j ein Knoten von σ_i ist.

Definition 3.10 (Scott-Zhang Operator). Zu jedem Knoten $z_i \in \mathcal{M}$ wählt man eine Kante ($d = 2$) bzw. Seitenfläche ($d = 3$) so wie oben beschrieben. Der Operator $S_h : H^1(\Omega) \rightarrow \mathcal{S}^1(\mathcal{T})$ wird definiert durch

$$S_h v := \sum_{i=1}^N v_i \eta_i, \quad (3.22)$$

mit

$$v_i := \int_{\sigma_i} \psi_i v dx. \quad (3.23)$$

Satz 3.10 (SCOTT / ZHANG 1990 [19, Satz 2.1]). Der in Definition 3.10 definierte Operator S_h ist eine Projektion von $H^1(\Omega)$ nach $\mathcal{S}^1(\mathcal{T})$ und erhält homogene Randbedingungen, d.h. $H_0^1(\Omega)$ wird auf $\mathcal{S}_0^1(\mathcal{T})$ abgebildet.

Beweis. Mit (3.21) folgt für $v \in \mathcal{S}^1(\mathcal{T})$ wegen $v = \sum_{j=1}^N v(z_j) \eta_j$

$$\begin{aligned} S_h v &= \sum_{i=1}^N \eta_i \int_{\sigma_i} \psi_i \sum_{j=1}^N v(z_j) \eta_j dx = \sum_{i=1}^N \eta_i \sum_{j=1}^N v(z_j) \int_{\sigma_i} \psi_i \eta_j dx = \sum_{i=1}^N \eta_i \sum_{j=1}^N v(z_j) \delta_{ij} \\ &= \sum_{i=1}^N \eta_i v(z_i) = v, \end{aligned}$$

und wir schließen daraus, dass der Operator S_h eine Projektion von $H^1(\Omega)$ auf $\mathcal{S}^1(\mathcal{T})$ ist.

Mit der Wahl von σ_i (siehe (3.16) bzw. (3.17)) erhält man die Implikation

$$v \in H_0^1(\Omega) \Rightarrow S_h v(z_i) = 0$$

für alle $z_i \in \mathcal{M}_\Gamma$ und daher insbesondere $S_h(H_0^1(\Omega)) \subseteq \mathcal{S}_0^1(\mathcal{T})$. Aufgrund von $S_h(\mathcal{S}_0^1(\mathcal{T})) = \mathcal{S}_0^1(\mathcal{T})$ folgt sogar $S_h(H_0^1(\Omega)) = \mathcal{S}_0^1(\mathcal{T})$. $\square$

Nachfolgender Satz sagt aus, dass der Scott-Zhang-Operator von optimaler Ordnung ist.

Satz 3.11 (SCOTT / ZHANG 1990 [19]). *Sei $v \in H^m(\Omega)$ und $m = 1, 2$, dann gilt für den Scott-Zhang-Operator*

$$\|v - S_h v\|_{H^1(\Omega)} \leq C_{S_h} \|h^{m-1} D^m v\|_{L^2(\Omega)}. \quad (3.24)$$

Die Konstante C_{S_h} hängt nur von der Formregularitätskonstante $\sigma(\mathcal{T})$ und der Dimension d ab.

Beweis. Ein Beweis ist etwa in SCOTT / ZHANG 1990 [19, Abschnitt 4] zu finden. $\square$

Aus dem letzten Satz folgt mit $m = 1$, dass $1 - S_h$ und daher auch S_h stetig ist.

Bemerkung 3.7. *Gilt für $v \in H^1(\Omega)$, dass $v|_{\sigma_i} = c \in \mathbb{R}$, dann vereinfacht sich die Berechnung des Knotenwertes v_i , vgl. (3.23), zu*

$$v_i = \int_{\sigma_i} \psi_i v \, dx = c \int_{\sigma_i} \psi_i 1 \, dx = c \int_{\sigma_i} \psi_i \left(\sum_{j=1}^N \eta_j \right) dx = c \sum_{j=1}^N \delta_{ij} = c,$$

wobei hier die Gleichheit (3.21) und die Tatsache, dass die Funktionen η_j eine Zerlegung der 1 darstellen, d.h. $\sum_{j=1}^N \eta_j(x) = 1$ für $x \in \Omega$, verwendet wurden.

Für $v \in H^1(\Omega)$, $v_h \in \mathcal{S}^1(\mathcal{T})$ mit $\gamma_0 v = \gamma_0 v_h$ gilt sogar

$$\gamma_0 S_h v = \gamma_0 v_h.$$

Dies folgt aus $v - v_h \in H_0^1(\Omega)$ und Satz 3.10.

Definition 3.11. *Zu jedem Knoten $z_i \in \mathcal{M}_\Gamma$ wählt man eine Kante ($d = 2$) bzw. Seitenfläche ($d = 3$) so wie oben beschrieben. Wir definieren den Operator $S_h^\Gamma : H^{1/2}(\Gamma) \rightarrow \mathcal{S}^1(\mathcal{T}_\Gamma)$ durch*

$$S_h^\Gamma v := \sum_{z_i \in \mathcal{M}_\Gamma} v_i \eta_i|_\Gamma. \quad (3.25)$$

Dabei sei v_i der in Definition 3.10 erklärte Wert zum Knoten z_i . Insbesondere folgt aus dieser Definition, dass bei gegebenem Operator S_h und daher auch einer Wahl der Kanten bzw. Seitenflächen

$$\gamma_0(S_h v) = S_h^\Gamma(\gamma_0 v) \quad \text{für alle } v \in H^1(\Omega)$$

gilt.

Für die spätere Implementierung müssen wir noch die $L^2(\sigma_i)$ -duale Basis mit der Eigenschaft (3.18) bestimmen. Mit der Eigenschaft (3.19) beläuft sich somit die Bestimmung der dualen Basisfunktion ψ_{ik} auf die Bestimmung der Koeffizienten a_{ik}^ℓ in

$$\int_{\sigma_i} \left(\sum_{\ell=1}^d a_{ik}^\ell \phi_{i\ell} \right) \phi_{ij} \, dx = \delta_{kj} \quad (3.26)$$

für alle $(j, k) \in \{1, \dots, d\}^2$.

Zunächst beschäftigen wir uns mit dem Fall $d = 2$. Wir bezeichnen mit z_{i1}, z_{i2} die zwei Endknoten der Kante σ_i . Zusätzlich gelte $z_{i1} = z_i$ und $\phi_{i1}(z_i) = 1$. Wir definieren den Weg γ_i als

$$\begin{aligned}\gamma_i : [0, 1] &\longrightarrow \sigma_i \\ t &\longmapsto z_{i1} + (z_{i2} - z_{i1})t,\end{aligned}$$

und erhalten somit

$$\int_{\sigma_i} \phi_{i\ell} \phi_{ij} dx = |\sigma_i| \int_0^1 \phi_{i\ell}(\gamma_i(t)) \phi_{ij}(\gamma_i(t)) dt.$$

Wir definieren noch die Funktionen $\widehat{\phi}_1, \widehat{\phi}_2$ auf dem Intervall $[0, 1]$ als

$$\begin{aligned}\widehat{\phi}_1(t) &:= \phi_{i1}(\gamma_i(t)), \\ \widehat{\phi}_2(t) &:= \phi_{i2}(\gamma_i(t)).\end{aligned}$$

Es gilt

$$\begin{aligned}\widehat{\phi}_1(t) &= 1 - t, \\ \widehat{\phi}_2(t) &= t.\end{aligned}$$

Man beachte, dass $\widehat{\phi}_1$ und $\widehat{\phi}_2$ nicht von der Kante σ_i abhängen. Mit

$$\left(\left(\sum_{\ell=1}^2 a_{ik}^\ell \phi_{i\ell} \right), \phi_{ij} \right)_{L^2(\sigma_i)} = |\sigma_i| \sum_{\ell=1}^2 a_{ik}^\ell \left(\widehat{\phi}_\ell, \widehat{\phi}_j \right)_{L^2([0,1])}$$

können nun die Gleichungen (3.26) folgendermaßen umgeschrieben werden:

$$|\sigma_i| \sum_{\ell=1}^2 a_{ik}^\ell \left(\widehat{\phi}_\ell, \widehat{\phi}_j \right)_{L^2([0,1])} = \delta_{kj},$$

für alle $(k, j) \in \{1, 2\}^2$. In Matrixschreibweise lautet dieses Gleichungssystem

$$|\sigma_i| \mathbf{A} \mathbf{M} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix},$$

wobei für die Einträge der 2×2 - Matrizen

$$\begin{aligned}\mathbf{A}_{k\ell} &:= a_{ik}^\ell, \\ \mathbf{M}_{k\ell} &:= \left(\widehat{\phi}_k, \widehat{\phi}_\ell \right)_{L^2([0,1])}\end{aligned}$$

gilt. Mit

$$\mathbf{M} = \begin{pmatrix} \frac{1}{3} & \frac{1}{6} \\ \frac{1}{6} & \frac{1}{3} \end{pmatrix},$$

ergibt sich die Koeffizientenmatrix $\mathbf{A}$ zu

$$\mathbf{A} = \frac{1}{|\sigma_i|} \begin{pmatrix} 4 & -2 \\ -2 & 4 \end{pmatrix},$$

und somit besitzt die duale Basis die Darstellung

$$\begin{aligned}\psi_{i1} &= \frac{1}{|\sigma_i|} (4\phi_{i1} - 2\phi_{i2}), \\ \psi_{i2} &= \frac{1}{|\sigma_i|} (-2\phi_{i1} + 4\phi_{i2}).\end{aligned}\tag{3.27}$$

Um die duale Basis für $d = 3$ zu bestimmen, kann man analog wie für $d = 2$ vorgehen. Es seien z_{i1}, z_{i2}, z_{i3} die drei Knoten der Seitenfläche σ_i . Außerdem gelte $z_{i1} = z_i$, und $\phi_{i1}(z_{i1}) = 1$. Wir definieren den Weg γ_i über

$$\begin{aligned}\gamma_i : T_{ref} &\longrightarrow \sigma_i \\ (t, s) &\longmapsto z_{i1} + (z_{i2} - z_{i1})t + (z_{i3} - z_{i1})s.\end{aligned}$$

Analoge Vorgehensweise wie im Fall $d = 2$ führt nun auf die Koeffizientenmatrix

$$\mathbf{A} = \frac{1}{|\sigma_i|} \begin{pmatrix} 18 & -6 & -6 \\ -6 & 18 & -6 \\ -6 & -6 & 18 \end{pmatrix},$$

und damit auf die duale Basis

$$\begin{aligned}\psi_{i1} &= \frac{1}{|\sigma_i|} (18\phi_{i1} - 6\phi_{i2} - 6\phi_{i3}), \\ \psi_{i2} &= \frac{1}{|\sigma_i|} (-6\phi_{i1} + 18\phi_{i2} - 6\phi_{i3}), \\ \psi_{i3} &= \frac{1}{|\sigma_i|} (-6\phi_{i1} - 6\phi_{i2} + 18\phi_{i3}).\end{aligned}\tag{3.28}$$

Mit den Beziehungen (3.27) und (3.28) lassen sich nun die Knotenwerte (3.23) des Scott-Zhang Operators bestimmen.

Kapitel 4

Ansätze zur Lösung des magnetostatischen Problems

In diesem Kapitel werden wir uns mit verschiedenen Ansätzen zur Lösung des magnetostatischen Problems auseinandersetzen. Zum einen handelt es sich dabei um die Ansätze von FREDKIN / KOEHLER 1990 [8] bzw. GARCÍA-CERVERA / ROMA 2005 [10], zum anderen um verschiedene FEM/BEM-Kopplungsstrategien. Da es sich beim magnetostatischen Problem um ein Transmissionsproblem handelt (siehe Definition 1.1), muss man das Potential im ganzen Raum $\mathbb{R}^d$ bestimmen. Ein wesentlicher Vorteil aller in dieser Arbeit vorgestellten Methoden besteht darin, dass die Lösung u nur mehr auf dem beschränkten Gebiet Ω bestimmt wird, was in vielen Anwendungen genügt. Beispielsweise tritt bei der Landau-Lifschitz-Gilbert Gleichung der Streufeldenergieterm

$$E_S = \frac{1}{2} \int_{\mathbb{R}^d} |\nabla u|^2 = \frac{1}{2} \int_{\Omega} \nabla u \cdot \mathbf{M}$$

auf, wobei aus der letzten Identität ersichtlich ist, dass das Potential nur mehr im Innenraum berechnet werden muss. Daher brauchen auch keine zusätzlichen Techniken verwendet werden, um das Potential im Außenbereich zu bestimmen.

Die Ansätze von Fredkin & Koehler und García-Cervera & Roma verwenden das Superpositionsprinzip mit dem die Gesamtlösung $u = u_1 + u_2$ als Überlagerung von Funktionen u_1 und u_2 dargestellt wird, welche Lösungen von verschiedenen Teilproblemen sind. Des weiteren benötigen wir zur Darstellung der Lösung noch das Einfach- und Doppelschichtpotential, welche in Abschnitt 2.3 beschrieben wurden. Die numerische Implementierung erfordert die Lösung von FEM Problemen sowie die Auswertung des Einfach- oder Doppelschichtpotentials auf dem Rand. Beide Ansätze werden auch hybride FEM/BEM Methoden genannt.

Im Gegensatz zu diesen stehen die FEM/BEM-Ansätze, die direkt über ein Gleichungssystem gelöst werden können. Dabei müssen verschiedenste Randintegraloperatoren ausgewertet werden. Die Approximationslösungen der hier vorgestellten FEM/BEM-Kopplungsstrategien sind Quasi-Bestapproximationen der exakten Lösung im Sinne des Céa-Lemmas.

Jedem Ansatz in diesem Kapitel ist ein eigener Abschnitt gewidmet. Der erste behandelt den Ansatz von García-Cervera & Roma, der zweite jenen von Fredkin & Koehler. Im dritten und vierten Teil des Kapitels erläutern wir einerseits die symmetrische und andererseits die direkte FEM/BEM-Kopplungsmethode. In jedem Abschnitt stellen wir die starke und schwache Formulierung der jeweiligen Methode auf. Danach beschäftigen wir uns mit der numerischen Umsetzung. Am Ende jedes Abschnitts wird dann noch auf die a priori Fehleranalyse eingegangen, um Aussagen über die Konvergenz der Verfahren treffen zu können.

In diesem Kapitel benötigen wir die magnetostatischen Gleichungen in der schwachen Formulierung, siehe dazu Abschnitt 2.2. Zur Wiederholung und wegen der besseren Übersichtlichkeit führen wir sie hier noch einmal an:

Für eine gegebene Magnetisierung $\mathbf{M} \in (H^1(\Omega))^d$ ist das magnetostatische Potential $u = (u^{int}, u^{ext}) \in H^1(\Omega) \times H_{lok}^1(\Omega^{ext})$ Lösung der Gleichungen

$$\begin{aligned} -\Delta u^{int} &= -\operatorname{div} \mathbf{M} && \text{in } \Omega, \\ -\Delta u^{ext} &= 0 && \text{in } \Omega^{ext}, \\ [\gamma_0 u] &= 0 && \text{auf } \Gamma, \\ [\gamma_1 u] &= -\mathbf{M} \cdot \mathbf{n} && \text{auf } \Gamma, \\ u^{ext}(\mathbf{x}) &= \begin{cases} b \log \|\mathbf{x}\| + \mathcal{O}\left(\frac{1}{\|\mathbf{x}\|}\right) & \text{falls } d = 2, \\ \mathcal{O}\left(\frac{1}{\|\mathbf{x}\|}\right) & \text{falls } d = 3, \end{cases} && \text{für } \|\mathbf{x}\| \rightarrow \infty. \end{aligned} \quad (4.1)$$

Weiters ist zu bemerken, dass für $u \in C^2(\mathbb{R}^d \setminus \Gamma) \cap C^0(\mathbb{R}^d)$ und gegebenes $\mathbf{M} \in (C^1(\overline{\Omega}))^d$, obige Gleichungen die klassische Formulierung beschreiben.

4.1 Ansatz von García-Cervera & Roma

In diesem Abschnitt stellen wir den Ansatz von García-Cervera & Roma vor und betrachten sowohl die schwache Formulierung als auch die numerische Umsetzung sowie die a priori Fehleranalyse.

4.1.1 Schwache Formulierung

In diesem Unterabschnitt werden wir die schwache Formulierung des Ansatzes von García-Cervera & Roma aufstellen. Zunächst betrachten wir die magnetostatischen Differentialgleichungen (siehe Gleichungssystem (4.1)) in der schwachen Form, wie sie am Ende von Abschnitt 2.2 beschrieben wurden.

Wir teilen die Lösung u auf in die Funktionen u_1 und u_2 . Sei nun $\mathbf{M} \in (H^1(\Omega))^d$ gegeben. Wir führen die Funktion $u_1 = (u_1^{int}, u_1^{ext}) \in H_0^1(\Omega) \times H_{lok}^1(\Omega^{ext})$ ein, welche Lösung von

$$\begin{aligned} \Delta u_1^{int} &= \operatorname{div} \mathbf{M} && \text{in } \Omega, \\ \gamma_0^{int} u_1^{int} &= 0 && \text{auf } \Gamma \end{aligned} \quad (4.2)$$

ist. Definiert man $u_1^{ext} := 0$, so führt dies auf die Sprünge

$$\begin{aligned} [\gamma_0 u_1] &= 0, \\ [\gamma_1 u_1] &= -\gamma_1^{int} u_1^{int}. \end{aligned}$$

Damit nun $u_1 + u_2$ das Gesamtproblem lösen kann, muss $u_2 = (u_2^{int}, u_2^{ext}) \in H^1(\Omega) \times H_{lok}^1(\Omega^{ext})$ die Gleichungen

$$\begin{aligned} \Delta u_2^{int} &= 0 && \text{in } \Omega, \\ \Delta u_2^{ext} &= 0 && \text{in } \Omega^{ext}, \\ [\gamma_0 u_2] &= 0 && \text{auf } \Gamma, \\ [\gamma_1 u_2] &= -\mathbf{M} \cdot \mathbf{n} + \gamma_1^{int} u_1^{int} && \text{auf } \Gamma, \\ u_2^{ext}(\mathbf{x}) &= \begin{cases} b \log \|\mathbf{x}\| + \mathcal{O}\left(\frac{1}{\|\mathbf{x}\|}\right) & \text{falls } d = 2, \\ \mathcal{O}\left(\frac{1}{\|\mathbf{x}\|}\right) & \text{falls } d = 3, \end{cases} && \text{für } \|\mathbf{x}\| \rightarrow \infty \end{aligned} \quad (4.3)$$

erfüllen. Die Funktionen u_1 und u_2 sind nach Kapitel 2 eindeutig bestimmt. Aus Satz 2.12 und aus den Sprungbedingungen des Einfachschichtpotentials (siehe Satz 2.13 und Satz 2.17) folgt, dass u_2 die Darstellung

$$u_2 = \tilde{\mathcal{V}}(\mathbf{M} \cdot \mathbf{n} - \gamma_1^{int} u_1^{int}), \quad (4.4)$$

besitzt. Aus dieser Darstellung lässt sich ein interessantes Resultat über das Abklingverhalten von u_2 und daher auch von u im Fall $d = 2$ ableiten.

Satz 4.1. *Sei $\mathbf{M} \in (H^1(\Omega))^d$ gegeben und u_1 die Lösung von (4.2), dann erfüllt u_2 im Fall $d = 2$ die Abklingbedingung*

$$u_2 = \mathcal{O}\left(\frac{1}{\|\mathbf{x}\|}\right) \quad \text{für } \|\mathbf{x}\| \rightarrow \infty. \quad (4.5)$$

Insbesondere gilt dann auch für das Potential $u = u_1 + u_2$

$$u = \mathcal{O}\left(\frac{1}{\|\mathbf{x}\|}\right) \quad \text{für } \|\mathbf{x}\| \rightarrow \infty,$$

und daher stimmen die Abklingbedingungen für den zwei- und dreidimensionalen Fall überein.

Beweis. Nach Lemma 2.8 gilt für die innere Konormalenableitung von u_1 mit $v \in H^1(\Omega)$

$$\begin{aligned} \langle \gamma_1^{int} u_1^{int}, \gamma_0 v \rangle_\Gamma &= (\nabla u, \nabla v)_{L^2(\Omega)} + (\operatorname{div} \mathbf{M}, v)_{L^2(\Omega)} \\ &= (\nabla u, \nabla v)_{L^2(\Omega)} - (\mathbf{M}, \nabla v)_{L^2(\Omega)} + \langle \mathbf{M} \cdot \mathbf{n}, \gamma_0 v \rangle_\Gamma. \end{aligned}$$

Dabei wurde bei obiger Umformung die erste Green'sche Identität verwendet. Wählt man als Testfunktion $v = 1$, dann ergibt sich nach Umformen

$$\langle \gamma_1^{int} u_1^{int} - \mathbf{M} \cdot \mathbf{n}, 1 \rangle_\Gamma = 0.$$

Mit der Darstellung (4.4) und Satz 2.19 erhält man dann für u_2 im Fall $d = 2$ das behauptete Abklingverhalten (4.5). $\square$

Konvention 4.1. *Wir werden im Folgenden keine Unterscheidung bei der Abklingbedingung im zwei- bzw. dreidimensionalen Fall mehr vornehmen, da diese nach Satz 4.1 übereinstimmen.*

Aus der Darstellung 4.4 könnte man theoretisch u_2 in $\mathbb{R}^d \setminus \Gamma$ bestimmen und daher auch das Potential $u = u_1 + u_2$. Stattdessen bilden wir die innere Spur von u_2 , was uns auf das folgende inhomogene Dirichlet Problem führt

$$\begin{aligned} \Delta u_2^{int} &= 0 && \text{in } \Omega, \\ \gamma_0 u_2^{int} &= \mathcal{V}(\mathbf{M} \cdot \mathbf{n} - \gamma_1^{int} u_1^{int}) && \text{auf } \Gamma, \end{aligned} \quad (4.6)$$

wobei die Randdaten fast überall auf Γ gelten. Würde man u_2 direkt über (4.4) bestimmen, so müsste man beim diskretisierten Problem in jedem Punkt $\mathbf{x}$ das Einfachschichtpotential auswerten. Das Auswerten des Einfachschichtpotentials an einem Punkt entspricht einer numerischen Berechnung eines Integrals. In der Praxis rechnet sich dieser Zugang nicht, da er zu einem erhöhten Rechenaufwand führt.

Die Methode von García-Cervera & Roma besteht nun darin, die zwei Dirichlet Probleme (4.2) und (4.6) zu lösen.

Wir haben jetzt für die Funktionen u_1^{int} und u_2^{int} jeweils eine partielle Differentialgleichung (siehe Gleichung (4.2) und (4.6)) aufgestellt, die nur noch im Innenraum Ω zu lösen ist. Die Lösung des Ausgangsproblems muss jedoch auf $\mathbb{R}^d$ bestimmt werden. Es bleibt also noch zu zeigen, dass durch u_1^{int} und u_2^{int} die Lösung des Ausgangsproblems bereits eindeutig bestimmt wird.

Satz 4.2. *Sei $\mathbf{M} \in (H^1(\Omega))^d$ gegeben. Die Gleichungen (4.2) bzw. (4.6) besitzen jeweils eine eindeutige Lösung $u_1^{int} \in H_0^1(\Omega)$ bzw. $u_2^{int} \in H^1(\Omega)$. Definiert man $u_1 := (u_1^{int}, 0) \in H_0^1(\Omega) \times H_{lok}^1(\Omega^{ext})$ und $u_2 := (u_2^{int}, u_2^{ext}) \in H^1(\Omega) \times H_{lok}^1(\Omega^{ext})$ mit $u_2^{ext} := \tilde{\mathcal{V}}(\mathbf{M} \cdot \mathbf{n} - \gamma_1^{int} u_1^{int})$, dann ist $u = u_1 + u_2 \in H^1(\Omega) \times H_{lok}^1(\Omega^{ext})$ die eindeutige Lösung der magnetostatischen Gleichungen (4.1).*

Beweis. Zunächst sei bemerkt, dass das magnetostatische Problem (4.1) nach Satz 2.24 eine eindeutige Lösung $u \in H^1(\Omega) \times H_{lok}^1(\Omega^{ext})$ besitzt. Nach Abschnitt 2.4 besitzen die Probleme (4.2) und (4.6) eindeutig bestimmte Lösungen $u_1^{int} \in H_0^1(\Omega)$ und $u_2^{int} \in H^1(\Omega)$.

Sei also nun $u_2 := (u_2^{int}, u_2^{ext})$ mit $u_2^{ext} := \tilde{\mathcal{V}}(\mathbf{M} \cdot \mathbf{n} - \gamma_1^{int} u_1^{int})$, dann ist u_2 Lösung von (4.3) und daher ist $u_1 + u_2$ mit $u_1 := (u_1^{int}, 0)$ nach Konstruktion eine und daher auch die eindeutig bestimmte Lösung von (4.1). $\square$

Da wir uns ab jetzt nur mehr mit Innenraumproblemen beschäftigen müssen, werden wir im restlichen Abschnitt den Index $(\cdot)^{int}$ weglassen.

Für eine numerische Umsetzung betrachten wir noch die Variationsformulierung der Gleichungen (4.2) und (4.6). Dazu benötigen wir folgendes Lemma.

Lemma 4.3. Sei $\mathbf{M} \in (H^1(\Omega))^d$ gegeben, dann ist die Variationsformulierung: Finde $u_1 \in H_0^1(\Omega)$ mit

$$(\nabla u_1, \nabla v)_{L^2(\Omega)} = (-\operatorname{div} \mathbf{M}, v)_{L^2(\Omega)} \quad \text{für alle } v \in H_0^1(\Omega),$$

äquivalent zu: Finde $u_1 \in H_0^1(\Omega)$ mit

$$(\nabla u_1, \nabla v)_{L^2(\Omega)} = (\mathbf{M}, \nabla v)_{L^2(\Omega)} \quad \text{für alle } v \in H_0^1(\Omega). \quad (4.7)$$

Beweis. Es gilt

$$(-\operatorname{div} \mathbf{M}, v)_{L^2(\Omega)} = \int_{\Omega} -\operatorname{div} \mathbf{M} v \, dx = \int_{\Omega} \mathbf{M} \cdot \nabla v \, dx + \underbrace{\langle \mathbf{M} \cdot \mathbf{n}, \gamma_0 v \rangle_{\Gamma}}_{=0, \text{ da } v \in H_0^1(\Omega)} = (\mathbf{M}, \nabla v)_{L^2(\Omega)}.$$

Hierbei wurde die erste Green'sche Identität benutzt. $\square$

Die Variationsformulierungen von (4.2) und (4.6) lauten bei gegebenem $\mathbf{M} \in (H^1(\Omega))^d$ somit: Finde $u_1 \in H_0^1(\Omega)$ mit

$$(\nabla u_1, \nabla v)_{L^2(\Omega)} = (\mathbf{M}, \nabla v)_{L^2(\Omega)} \quad \text{für alle } v \in H_0^1(\Omega), \quad (4.8)$$

und: Finde $u_2 \in H^1(\Omega)$ mit

$$\begin{aligned} (\nabla u_2, \nabla v)_{L^2(\Omega)} &= 0 \quad \text{für alle } v \in H_0^1(\Omega), \\ \gamma_0 u_2 &= \mathcal{V}(\mathbf{M} \cdot \mathbf{n} - \gamma_1^{int} u_1^{int}). \end{aligned} \quad (4.9)$$

4.1.2 Numerische Umsetzung

Das Ermitteln einer Lösung mit Hilfe des Ansatzes von García-Cervera & Roma benötigt die Lösung eines homogenen und eines inhomogenen Dirichlet Problems. Die Randdaten, die zur Lösung des zweiten Problems nötig sind, werden dabei mit Hilfe des Einfachschichtpotentials aus der Lösung des homogenen Dirichlet Problems und der Magnetisierung bestimmt.

Die Lösungsschritte lassen sich in folgendem Algorithmus zusammenfassen, wobei wir mit $\Pi_h^0 : L^2(\Gamma) \rightarrow \mathcal{P}^0(\mathcal{T}_{\Gamma})$ die L^2 -Orthogonalprojektion auf den Raum der stückweise konstanten Funktionen $\mathcal{P}^0(\mathcal{T}_{\Gamma})$ bezeichnen. Für eine Funktion V aus $S_0^1(\mathcal{T})$ gilt, da der Normalenvektor $\mathbf{n}$ konstant auf jedem Randelement ist, dass die Normalenableitung $\nabla V \cdot \mathbf{n}$ in $\mathcal{P}^0(\mathcal{T}_{\Gamma})$ liegt. Weiters gilt auch $\mathbf{M} \cdot \mathbf{n} \in L^2(\Gamma)$.

Der Operator $A_h : H^{1/2}(\Gamma) \rightarrow \mathcal{S}^1(\mathcal{T}_{\Gamma})$ soll dabei in gewissem Sinne die Spur eines Approximationsoperators sein, d.h. er soll

$$\gamma_0 A_h^{\Omega} v = A_h \gamma_0 v \quad \text{für alle } v \in H^1(\Omega) \quad (4.10)$$

erfüllen, wobei $A_h^{\Omega} : H^1(\Omega) \rightarrow \mathcal{S}^1(\mathcal{T})$ einen Interpolationsoperator notiert. Diese Eigenschaft wird in der später folgenden Konvergenzanalyse benötigt. Wählt man für A_h^{Ω} den Scott-Zhang-Operator S_h und für A_h den Operator

S_h^Γ , dann gilt nach Definition beider Operatoren die Eigenschaft (4.10). Schwächt man Bedingung (4.10) so ab, dass sie nur noch für alle $v \in H^2(\Omega)$ gilt, so kann auch der nodale Interpolant für A_h^Ω bzw. A_h gewählt werden.

Algorithmus 4.1. Input: Magnetisierung $\mathbf{M}$ und Triangulierung $\mathcal{T}$.

Schritt 1: Bestimme $U_1 \in \mathcal{S}_0^1(\mathcal{T})$ mit

$$(\nabla U_1, \nabla V)_{L^2(\Omega)} = (\mathbf{M}, \nabla V)_{L^2(\Omega)} \quad \text{für alle } V \in \mathcal{S}_0^1(\mathcal{T}).$$

Schritt 2: Berechne mit $w_h = \Pi_h^0(\mathbf{M} \cdot \mathbf{n}) - \nabla U_1 \cdot \mathbf{n} \in \mathcal{P}^0(\mathcal{T}_\Gamma)$ die Randdaten $g_h \in \mathcal{S}^1(\mathcal{T}_\Gamma)$ durch Auswerten von

$$g_h = A_h \mathcal{V} w_h.$$

Schritt 3: Bestimme $U_2 \in \mathcal{S}^1(\mathcal{T})$ mit

$$\begin{aligned} \gamma_0 U_2 &= g_h, \\ (\nabla U_2, \nabla V)_{L^2(\Omega)} &= 0 \quad \text{für alle } V \in \mathcal{S}_0^1(\mathcal{T}). \end{aligned}$$

Schritt 4: Berechne eine Approximationslösung $U = U_1 + U_2 \in \mathcal{S}^1(\mathcal{T})$ für das magnetostatische Potential.

Output: Approximationslösung $U \in \mathcal{S}^1(\mathcal{T})$.

Aus Algorithmus 4.1 ergibt sich die Approximation für das magnetische Streufeld zu $\mathbf{H}_h = -\nabla U \in (\mathcal{P}^0(\mathcal{T}))^d$.

4.1.3 A priori Fehleranalyse

In diesem Unterabschnitt wollen wir eine Abschätzung des Fehlers zwischen der analytischen und numerischen Lösung aufstellen.

Satz 4.4. Sei $u = u_1 + u_2$ die schwache Lösung des Ansatzes von García-Cervera & Roma und U die Approximationslösung (siehe Algorithmus 4.1). Mit der zusätzlichen Annahme $u_1, u_2 \in H^2(\Omega)$ gilt

$$\|u - U\|_{H^1(\Omega)} = \mathcal{O}(h) + \mathcal{O}(\|\gamma_1 u_1 - \nabla U_1 \cdot \mathbf{n}\|_{H^{-1/2}(\Gamma)}).$$

Beweis. Es gilt $u = u_1 + u_2$, wobei u_1 bzw. u_2 die eindeutigen Lösungen der Variationsformulierungen von (4.8) bzw. (4.9) und U_1 bzw. U_2 die eindeutigen Lösungen der Variationsformulierungen in Algorithmus 4.1 sind.

Aus der Dreiecksungleichung erhält man sofort

$$\|u - U\|_{H^1(\Omega)} \leq \|u_1 - U_1\|_{H^1(\Omega)} + \|u_2 - U_2\|_{H^1(\Omega)}. \quad (4.11)$$

Aus der Zusatzannahme $u_1 \in H^2(\Omega)$ folgt aus dem Céa-Lemma (Lemma 3.1) und Korollar 3.9

$$\|u_1 - U_1\|_{H^1(\Omega)} = \mathcal{O}(h).$$

Um den zweiten Term in (4.11) abschätzen zu können, benötigen wir das Céa-Lemma in der Form von Lemma 3.7. Dazu definieren wir eine Fortsetzung von g_h als

$$\widehat{G} := A_h^\Omega \widetilde{\mathcal{V}} w_h \in \mathcal{S}^1(\mathcal{T}).$$

Die Operatoren A_h und A_h^Ω sind gerade so gewählt, dass sie die Eigenschaft (4.10) erfüllen, es folgt daher $\gamma_0 \widehat{G} = g_h$. Weiters bemerken wir, dass die schwache Lösung u_2 nach Unterabschnitt 4.1.1 die Darstellung

$$u_2 = \widetilde{\mathcal{V}}(\mathbf{M} \cdot \mathbf{n} - \gamma_1 u_1)$$

besitzt. Das C ea-Lemma besagt nun

$$\begin{aligned} \|u_2 - U_2\|_{H^1(\Omega)} &\lesssim \min_{\mathcal{V} \in \mathcal{S}_0^1(\mathcal{T})} \|u_2 - (\mathcal{V} + \widehat{G})\|_{H^1(\Omega)} \leq \|u_2 - \widehat{G}\|_{H^1(\Omega)} = \|\widetilde{\mathcal{V}}w - A_h^\Omega \widetilde{\mathcal{V}}w_h\|_{H^1(\Omega)} \\ &\leq \|\widetilde{\mathcal{V}}w - A_h^\Omega \widetilde{\mathcal{V}}w\|_{H^1(\Omega)} + \|A_h^\Omega \widetilde{\mathcal{V}}w - A_h^\Omega \widetilde{\mathcal{V}}w_h\|_{H^1(\Omega)}. \end{aligned} \quad (4.12)$$

Wir w hlen f r A_h^Ω den Scott-Zhang-Operator S_h . F r den zweiten Term ergibt sich mit der Stetigkeit der Operatoren S_h und $\widetilde{\mathcal{V}}$

$$\begin{aligned} \|S_h \widetilde{\mathcal{V}}w - S_h \widetilde{\mathcal{V}}w_h\|_{H^1(\Omega)} &\lesssim \|\widetilde{\mathcal{V}}w - \widetilde{\mathcal{V}}w_h\|_{H^1(\Omega)} \lesssim \|w - w_h\|_{H^{-1/2}(\Gamma)} \\ &= \|\mathbf{M} \cdot \mathbf{n} - \gamma_1 u_1 - (\Pi_h^0(\mathbf{M} \cdot \mathbf{n}) - \nabla U_1 \cdot \mathbf{n})\|_{H^{-1/2}(\Gamma)} \\ &\leq \|\mathbf{M} \cdot \mathbf{n} - \Pi_h^0(\mathbf{M} \cdot \mathbf{n})\|_{H^{-1/2}(\Gamma)} + \|\gamma_1 u_1 - \nabla U_1 \cdot \mathbf{n}\|_{H^{-1/2}(\Gamma)} \end{aligned} \quad (4.13)$$

Nach PRAETORIUS 2007 [16, Satz 19] gilt

$$\|(1 - \Pi_h^0)\mathbf{M} \cdot \mathbf{n}\|_{H^{-1/2}(\Gamma)} = \mathcal{O}(h^{3/2}). \quad (4.14)$$

Der erste Term in (4.12) l sst sich mit Satz 3.11 absch tzen durch

$$\|\widetilde{\mathcal{V}}w - A_h^\Omega \widetilde{\mathcal{V}}w\|_{H^1(\Omega)} \lesssim \|hD^2 \widetilde{\mathcal{V}}w\|_{L^2(\Omega)}. \quad (4.15)$$

Nach Voraussetzung gilt $u_2 = \widetilde{\mathcal{V}}w \in H^2(\Omega)$.

Mit (4.13), (4.14) und (4.15) folgt aus (4.12)

$$\|u_2 - U_2\|_{H^1(\Omega)} = \mathcal{O}(h) + \mathcal{O}(\|\gamma_1 u_1 - \nabla U_1 \cdot \mathbf{n}\|_{H^{-1/2}(\Gamma)}),$$

und damit das behauptete Resultat. $\square$

4.2 Ansatz von Fredkin & Koehler

In diesem Abschnitt stellen wir den Ansatz von Fredkin & Koehler vor und betrachten sowohl die schwache Formulierung, als auch die numerische Umsetzung sowie die a priori Fehleranalyse.

4.2.1 Schwache Formulierung

In diesem Unterabschnitt werden wir die schwache Formulierung des Ansatzes von Fredkin & Koehler aufstellen. Zun chst betrachten wir die magnetostatischen Differentialgleichungen in der schwachen Form, wie sie am Ende von Abschnitt 2.2 beschrieben wurden. Wir teilen die L sung u auf in die Funktionen u_1 und u_2 . Sei $\mathbf{M} \in (H^1(\Omega))^d$ gegeben. Wir f hren die Funktion $u_1 = (u_1^{int}, u_1^{ext}) \in H_*^1(\Omega) \times H_{lok}^1(\Omega^{ext})$ ein, welche L sung von

$$\begin{aligned} \Delta u_1^{int} &= \operatorname{div} \mathbf{M} \quad \text{in } \Omega, \\ \gamma_1^{int} u_1^{int} &= \mathbf{M} \cdot \mathbf{n} \quad \text{auf } \Gamma \end{aligned} \quad (4.16)$$

ist. Definiert man $u_1^{ext} := 0$, so f hrt dies auf die Spr nge

$$\begin{aligned} [\gamma_0 u_1] &= -\gamma_0^{int} u_1^{int}, \\ [\gamma_1 u_1] &= 0. \end{aligned}$$

Damit nun $u_1 + u_2$ das Gesamtproblem lösen kann, muss $u_2 = (u_2^{int}, u_2^{ext}) \in H^1(\Omega) \times H_{lok}^1(\Omega^{ext})$ die Gleichungen

$$\begin{aligned} \Delta u_2^{int} &= 0 && \text{in } \Omega, \\ \Delta u_2^{ext} &= 0 && \text{in } \Omega^{ext}, \\ [\gamma_0 u_2] &= \gamma_0^{int} u_1^{int} && \text{auf } \Gamma, \\ [\gamma_1 u_2] &= 0 && \text{auf } \Gamma, \\ u_2 &= \mathcal{O}\left(\frac{1}{\|x\|}\right) && \text{für } \|x\| \rightarrow \infty \end{aligned} \quad (4.17)$$

erfüllen. Die Funktionen u_1 und u_2 sind nach Kapitel 2 eindeutig bestimmt. Aus Satz 2.15 und aus den Sprungbedingungen des Doppelschichtpotentials (siehe Satz 2.16 und Satz 2.17) folgt, dass u_2 die Darstellung

$$u_2 = \tilde{\mathcal{K}}(\gamma_0^{int} u_1^{int}) \quad (4.18)$$

besitzt. Aus (4.18) könnte man theoretisch u_2 in $\mathbb{R}^d \setminus \Gamma$ bestimmen und daher auch das Potential $u = u_1 + u_2$. Stattdessen bilden wir die innere Spur von u_2 , was uns auf das folgende inhomogene Dirichlet Problem führt

$$\begin{aligned} \Delta u_2^{int} &= 0 && \text{in } \Omega, \\ \gamma_0 u_2^{int} &= (\mathcal{K} - 1/2) \gamma_0^{int} u_1^{int} && \text{auf } \Gamma, \end{aligned} \quad (4.19)$$

wobei die Randdaten fast überall auf Γ gelten. Würde man u_2 direkt über (4.18) bestimmen, so müsste man beim diskretisierten Problem in jedem Punkt x das Doppelschichtpotential auswerten. Das Auswerten des Doppelschichtpotentials an einem Punkt entspricht einer numerischen Berechnung eines Integrals. In der Praxis rechnet sich dieser Zugang nicht, da er zu einem erhöhten Rechenaufwand führt.

Die Methode von Fredkin & Koehler besteht nun darin, das Neumann Problem (4.16) und das Dirichlet Problem (4.19) zu lösen.

Wir haben jetzt für die Funktionen u_1^{int} und u_2^{int} jeweils ein Gleichungssystem (siehe Gleichung (4.16) und (4.19)) aufgestellt, das nur noch im Innenraum Ω zu lösen ist. Die Lösung des Ausgangsproblems muss jedoch auf $\mathbb{R}^d$ bestimmt werden. Es bleibt also noch zu zeigen, dass durch u_1^{int} und u_2^{int} die Lösung des Ausgangsproblems bereits eindeutig bestimmt wird.

Satz 4.5. Sei $M \in (H^1(\Omega))^d$ gegeben. Die Gleichungen (4.16) bzw. (4.19) besitzen jeweils eine eindeutige Lösung $u_1^{int} \in H_*^1(\Omega)$ bzw. $u_2^{int} \in H^1(\Omega)$. Definiert man $u_1 := (u_1^{int}, 0) \in H_*^1(\Omega) \times H_{lok}^1(\Omega^{ext})$ und $u_2 := (u_2^{int}, u_2^{ext}) \in H^1(\Omega) \times H_{lok}^1(\Omega^{ext})$ mit $u_2^{ext} := \tilde{\mathcal{K}}(\gamma_0^{int} u_1^{int})$, dann ist $u = u_1 + u_2 \in H^1(\Omega) \times H_{lok}^1(\Omega^{ext})$ die eindeutige Lösung der magnetostatischen Gleichungen (4.1).

Beweis. Nach Abschnitt 2.4 besitzen die Probleme (4.16) und (4.19) eindeutig bestimmte Lösungen $u_1^{int} \in H_0^1(\Omega)$ und $u_2^{int} \in H^1(\Omega)$.

Sei also $u_2 := (u_2^{int}, u_2^{ext})$ mit $u_2^{ext} := \tilde{\mathcal{K}}(\gamma_0^{int} u_1^{int})$, dann ist u_2 die Lösung von (4.17) und daher ist $u_1 + u_2$ mit $u_1 := (u_1^{int}, 0)$ nach Konstruktion eine und daher auch die eindeutig bestimmte Lösung von (4.1). $\square$

Das Neumann Problem (4.16) ist im Allgemeinen eindeutig bestimmt bis auf eine Konstante. Die Forderung $u_1^{int} \in H_*^1(\Omega)$ setzt diese Konstante fest. Wir wollen kurz zeigen, dass diese Festlegung sich nicht auf die Gesamtlösung auswirkt. Dazu sei $\tilde{u}_1 := (\tilde{u}_1^{int}, \tilde{u}_1^{ext}) \in H^1(\Omega) \times H_{lok}^1(\Omega^{ext})$ eine — bis auf eine Konstante eindeutige — Lösung der Differentialgleichung (4.16) mit $\tilde{u}_1^{ext} = 0$. Es gilt daher $\tilde{u}_1^{int} = u_1^{int} + c$ mit $c \in \mathbb{R}$. Analoge Abhandlungen wie zu Beginn dieses Abschnitts führen auf eine Funktion $\tilde{u}_2 \in H^1(\Omega) \times H_{lok}^1(\Omega^{ext})$ mit der Darstellung

$$\tilde{u}_2 = \tilde{\mathcal{K}} \gamma_0^{int} \tilde{u}_1^{int}.$$

Wir wollen nun zeigen, dass $\tilde{u}_1 + \tilde{u}_2 = u_1 + u_2$ gilt. Nach PRAETORIUS 2007 [16, Lemma 14] gilt

$$\tilde{\mathcal{K}} 1(x) = \begin{cases} -1 & \text{falls } x \in \Omega, \\ 0 & \text{falls } x \in \Omega^{ext}. \end{cases} \quad (4.20)$$

Daher gilt mit der charakteristischen Funktion χ_Ω

$$\begin{aligned}\tilde{u} &= \tilde{u}_1 + \tilde{u}_2 = u_1 + c\chi_\Omega + \tilde{\mathcal{K}}\gamma_0^{int}\tilde{u}_1^{int} = u_1 + c\chi_\Omega + \tilde{\mathcal{K}}\gamma_0^{int}u_1^{int} + \tilde{\mathcal{K}}c = u_1 + u_2 + c\chi_\Omega - c\chi_\Omega \\ &= u_1 + u_2 = u.\end{aligned}$$

Da wir uns ab jetzt nur mehr mit Innenraumproblemen beschäftigen müssen, werden wir im restlichen Abschnitt den Index $(\cdot)^{int}$ weglassen.

Für eine numerische Umsetzung betrachten wir die Variationsformulierung von (4.16) und (4.19). Dazu benötigen wir noch folgendes Lemma.

Lemma 4.6. Sei $\mathbf{M} \in (H^1(\Omega))^d$ gegeben, dann ist die Variationsformulierung: Finde $u_1 \in H_*^1(\Omega)$ mit

$$(\nabla u_1, \nabla v)_{L^2(\Omega)} = (-\operatorname{div} \mathbf{M}, v)_{L^2(\Omega)} + \langle \mathbf{M} \cdot \mathbf{n}, \gamma_0 v \rangle_\Gamma \quad \text{für alle } v \in H_*^1(\Omega), \quad (4.21)$$

äquivalent zu: Finde $u_1 \in H_*^1(\Omega)$ mit

$$(\nabla u_1, \nabla v)_{L^2(\Omega)} = (\mathbf{M}, \nabla v)_{L^2(\Omega)} \quad \text{für alle } v \in H_*^1(\Omega). \quad (4.22)$$

Beweis. Mit Hilfe der ersten Green'schen Identität ergibt sich

$$(-\operatorname{div} \mathbf{M}, v)_{L^2(\Omega)} = (\mathbf{M}, \nabla v)_{L^2(\Omega)} - \langle \mathbf{M} \cdot \mathbf{n}, \gamma_0 v \rangle_\Gamma \quad \text{für alle } v \in H^1(\Omega). \quad (4.23)$$

Daher ergibt sich für die rechte Seite von Gleichung (4.21)

$$(-\operatorname{div} \mathbf{M}, v)_{L^2(\Omega)} + \langle \mathbf{M} \cdot \mathbf{n}, \gamma_0 v \rangle_\Gamma = (\mathbf{M}, \nabla v)_{L^2(\Omega)} \quad \text{für alle } v \in H_*^1(\Omega),$$

womit die Aussage des Lemmas bewiesen ist. $\square$

Die Variationsformulierung des Ansatzes von Fredkin & Koehler lautet nun: Für gegebenes $\mathbf{M} \in (H^1(\Omega))^d$ finde $u_1 \in H_*^1(\Omega)$ mit

$$(\nabla u_1, \nabla v)_{L^2(\Omega)} = (\mathbf{M}, \nabla v)_{L^2(\Omega)} \quad \text{für alle } v \in H_*^1(\Omega), \quad (4.24)$$

und: Finde $u_2 \in H^1(\Omega)$ mit

$$\begin{aligned}(\nabla u_2, \nabla v)_{L^2(\Omega)} &= 0 & \text{für alle } v \in H_0^1(\Omega), \\ \gamma_0 u_2 &= (\mathcal{K} - 1/2)\gamma_0 u_1.\end{aligned} \quad (4.25)$$

4.2.2 Numerische Umsetzung

Das Ermitteln einer Lösung mit Hilfe des Ansatzes von Fredkin & Koehler erfordert die Lösung eines Neumann und eines inhomogenen Dirichlet Problems. Die Randdaten die zur Lösung des inhomogenen Dirichlet Problems nötig sind, werden dabei mit Hilfe des Doppelschichtpotentials aus der Lösung des Neumann Problems bestimmt.

Die Lösungsschritte lassen sich in folgendem Algorithmus zusammenfassen, wobei wir zur Wahl des Operators $A_h : H^{1/2}(\Gamma) \rightarrow \mathcal{S}^1(\mathcal{T}_\Gamma)$ auf den Abschnitt 4.1.2 verweisen.

Algorithmus 4.2. Input: Magnetisierung $\mathbf{M}$ und Triangulierung $\mathcal{T}$.

Schritt 1: Bestimme $U_1 \in \mathcal{S}_*^1(\mathcal{T})$ mit

$$(\nabla U_1, \nabla V)_{L^2(\Omega)} = (\mathbf{M}, \nabla V)_{L^2(\Omega)} \quad \text{für alle } V \in \mathcal{S}_*^1(\mathcal{T}).$$

Schritt 2: Berechne die Randdaten $g_h \in \mathcal{S}^1(\mathcal{T}_\Gamma)$ durch Auswerten von

$$g_h = A_h(\mathcal{K} - 1/2)\gamma_0 U_1.$$

Schritt 3: Bestimme $U_2 \in \mathcal{S}^1(\mathcal{T})$ mit

$$\begin{aligned} \gamma_0 U_2 &= g_h, \\ (\nabla U_2, \nabla V)_{L^2(\Omega)} &= 0 \quad \text{für alle } V \in \mathcal{S}_0^1(\mathcal{T}). \end{aligned}$$

Schritt 4: Berechne eine Approximationslösung

$$U = U_1 + U_2 \in \mathcal{S}^1(\mathcal{T})$$

für das magnetostatische Potential.

Output: Approximationslösung $U \in \mathcal{S}^1(\mathcal{T})$.

Aus Algorithmus 4.2 ergibt sich die Approximation für das magnetische Streufeld zu $\mathbf{H}_h = -\nabla U \in (\mathcal{P}^0(\mathcal{T}))^d$.

4.2.3 A priori Fehleranalyse

In diesem Unterabschnitt wollen wir eine Abschätzung des Fehlers zwischen der analytischen und numerischen Lösung aufstellen.

Satz 4.7. Sei $u = u_1 + u_2$ die schwache Lösung des Ansatzes von Fredkin & Koehler und U die Approximationslösung (siehe Algorithmus 4.2). Mit der zusätzlichen Annahme $u_1, u_2 \in H^2(\Omega)$ gilt

$$\|u - U\|_{H^1(\Omega)} = \mathcal{O}(h).$$

Beweis. Es gilt $u = u_1 + u_2$ und $U = U_1 + U_2$, wobei u_1 bzw. u_2 die eindeutigen Lösungen der Variationsformulierungen von (4.24) bzw. (4.25) und U_1 bzw. U_2 die eindeutigen Lösungen der Variationsformulierungen in Algorithmus 4.2 sind.

Aus der Dreiecksungleichung erhält man sofort

$$\|u - U\|_{H^1(\Omega)} \leq \|u_1 - U_1\|_{H^1(\Omega)} + \|u_2 - U_2\|_{H^1(\Omega)}. \quad (4.26)$$

Aus der Zusatzannahme $u_1 \in H^2(\Omega)$ folgt mit dem Céa-Lemma (Lemma 3.1) und Korollar 3.9 für den ersten Term

$$\|u_1 - U_1\|_{H^1(\Omega)} = \mathcal{O}(h).$$

Um den zweiten Term in (4.26) abschätzen zu können, benötigen wir das Céa-Lemma in der Form von Lemma 3.7. Dazu definieren wir eine Fortsetzung von g_h als

$$\widehat{G} := A_h^\Omega \widetilde{\mathcal{K}} \gamma_0 U_1 \in \mathcal{S}^1(\mathcal{T}).$$

Aus den Eigenschaften von A_h und A_h^Ω — siehe Gleichheit (4.10) — folgt $\gamma_0 \widehat{G} = g_h$. Weiters bemerken wir, dass die schwache Lösung u_2 nach Unterabschnitt 4.2.1 die Darstellung

$$u_2 = \widetilde{\mathcal{K}} \gamma_0 u_1$$

besitzt. Das Céa-Lemma besagt nun

$$\begin{aligned} \|u_2 - U_2\|_{H^1(\Omega)} &\lesssim \min_{V \in S_h^1(\mathcal{T})} \|u_2 - (V + \widehat{G})\|_{H^1(\Omega)} \lesssim \|u_2 - \widehat{G}\|_{H^1(\Omega)} = \|\widetilde{\mathcal{K}}\gamma_0 u_1 - A_h^\Omega \widetilde{\mathcal{K}}\gamma_0 U_1\|_{H^1(\Omega)} \\ &\leq \|\widetilde{\mathcal{K}}\gamma_0 u_1 - A_h^\Omega \widetilde{\mathcal{K}}\gamma_0 u_1\|_{H^1(\Omega)} + \|A_h^\Omega \widetilde{\mathcal{K}}\gamma_0 u_1 - A_h^\Omega \widetilde{\mathcal{K}}\gamma_0 U_1\|_{H^1(\Omega)}. \end{aligned} \quad (4.27)$$

Wir wählen $A_h^\Omega = S_h$, dann gilt für den zweiten Term in (4.27) aufgrund der Stetigkeit der Operatoren S_h , $\widetilde{\mathcal{K}}$ und γ_0

$$\begin{aligned} \|S_h \widetilde{\mathcal{K}}\gamma_0 u_1 - S_h \widetilde{\mathcal{K}}\gamma_0 U_1\|_{H^1(\Omega)} &\lesssim \|\widetilde{\mathcal{K}}(\gamma_0 u_1 - \gamma_0 U_1)\|_{H^1(\Omega)} \lesssim \|\gamma_0(u_1 - U_1)\|_{H^{1/2}(\Gamma)} \\ &\lesssim \|u_1 - U_1\|_{H^1(\Omega)} = \mathcal{O}(h). \end{aligned} \quad (4.28)$$

Mit der Approximationseigenschaft von S_h (siehe Satz 3.11) kann man den ersten Term in (4.27) abschätzen durch

$$\|\widetilde{\mathcal{K}}\gamma_0 u_1 - S_h \widetilde{\mathcal{K}}\gamma_0 u_1\|_{H^1(\Omega)} \lesssim \|hD^2 \widetilde{\mathcal{K}}\gamma_0 u_1\|_{L^2(\Omega)}. \quad (4.29)$$

Man beachte, dass nach Voraussetzung $u_2 = \mathcal{K}\gamma_0 u_1 \in H^2(\Omega)$ gilt. Mit (4.28) und (4.29) folgt aus (4.27)

$$\|u_2 - U_2\|_{H^1(\Omega)} = \mathcal{O}(h) \quad (4.30)$$

und daher das behauptete Resultat. □

Bemerkung 4.1. Die Voraussetzung $u_2 \in H^2(\Omega)$ in Satz 4.7 wird im zugehörigen Beweis in (4.29) benötigt. Diese zusätzliche Voraussetzung kann unter der Annahme, dass der Rand Γ glatt ist, wegfallen, da aus Satz 2.18 $\widetilde{\mathcal{K}}\gamma_0 u_1 \in H^2(\Omega)$ folgt, falls $\gamma_0 u_1 \in H^{3/2}(\Gamma)$.

4.3 Symmetrische FEM/BEM-Kopplungsmethode

In diesem Abschnitt der Arbeit werden wir die bereits in Kapitel 2 angeführte FEM/BEM-Kopplungsmethode behandeln. Dabei betrachten wir die Variationsformulierung, die numerische Umsetzung und die a priori Analysis.

Die hier angeführten Resultate sind dabei aus CARSTENSEN / STEPHAN 1995 [4] entnommen. Die symmetrische Kopplung erschien zuerst in COSTABEL 1988 [5].

4.3.1 Formulierung

Das symmetrische FEM/BEM-Kopplungsproblem wurde bereits in Definition 2.15 aufgestellt. Zur Wiederholung sei es hier nochmals angeführt: Seien $f \in L^2(\Omega)$, $g_0 \in H^{1/2}(\Gamma)$ und $g_1 \in H^{-1/2}(\Gamma)$ gegeben. Finde $(u, \phi) \in H^1(\Omega) \times H^{-1/2}(\Gamma)$, so dass

$$\begin{aligned} (\nabla u, \nabla v)_{L^2(\Omega)} + \langle \mathcal{W}\gamma_0 u + (\mathcal{K}' - 1/2)\phi, \gamma_0 v \rangle_\Gamma &= \langle f, v \rangle - \langle g_1 + \mathcal{W}g_0, \gamma_0 v \rangle_\Gamma, \\ \langle \psi, \mathcal{V}\phi - (\mathcal{K} - 1/2)\gamma_0 u \rangle_\Gamma &= \langle \psi, (\mathcal{K} - 1/2)g_0 \rangle_\Gamma \end{aligned} \quad (4.31)$$

für alle $(v, \psi) \in H^1(\Omega) \times H^{-1/2}(\Gamma)$ gilt. Für den Fall $d = 2$ skalieren wir das Gebiet Ω so, dass $\text{diam}(\Omega) < 1$, damit der Operator $\mathcal{V}$ positiv definit ist.

Um nun das Kopplungsproblem speziell für die magnetostatischen Gleichungen zu formulieren, setzen wir, bei gegebenem $\mathbf{M} \in (H^1(\Omega))^d$, $f = -\text{div } \mathbf{M}$, $g_0 = 0$ und $g_1 = -\mathbf{M} \cdot \mathbf{n}$. Mit der ersten Green'schen Identität (4.23) können wir noch die rechte Seite der ersten Gleichung von (4.31) umschreiben:

$$(-\text{div } \mathbf{M}, v)_{L^2(\Omega)} + \langle \mathbf{M} \cdot \mathbf{n}, \gamma_0 v \rangle_\Gamma = (\mathbf{M}, \nabla v)_{L^2(\Omega)} \quad \text{für alle } v \in H^1(\Omega).$$

Somit ergibt sich das Kopplungsproblem für die magnetostatischen Gleichungen zu: Finde $(u, \phi) \in H^1(\Omega) \times H^{-1/2}(\Gamma)$, so dass

$$\begin{aligned} (\nabla u, \nabla v)_{L^2(\Omega)} + \langle \mathcal{W}\gamma_0 u + (\mathcal{K}' - 1/2)\phi, \gamma_0 v \rangle_\Gamma &= (\mathbf{M}, \nabla v)_{L^2(\Omega)}, \\ \langle \psi, \mathcal{V}\phi - (\mathcal{K} - 1/2)\gamma_0 u \rangle_\Gamma &= 0 \end{aligned} \quad (4.32)$$

für alle $(v, \psi) \in H^1(\Omega) \times H^{-1/2}(\Gamma)$ gilt.

Wie bereits in Abschnitt 2.4 erwähnt wird, ist das symmetrische Kopplungsproblem eindeutig lösbar und äquivalent mit dem Ausgangsproblem.

4.3.2 Numerische Umsetzung

Als Ansatzraum für die Approximationslösung wählen wir $\mathcal{S}^1(\mathcal{T}) \subset H^1(\Omega)$ und $\mathcal{P}^0(\mathcal{T}_\Gamma) \subset H^{-1/2}(\Gamma)$. Dabei bezeichnet $\mathcal{P}^0(\mathcal{T}_\Gamma)$ wieder den Raum der stückweisen konstanten Funktionen. Die Vorgangsweise beim Lösen des Problems lässt sich in folgendem Algorithmus zusammenfassen.

Algorithmus 4.3. Input: Magnetisierung $\mathbf{M}$ und Triangulierung $\mathcal{T}$.

Schritt 1: Bestimme $(U, \Phi) \in \mathcal{S}^1(\mathcal{T}) \times \mathcal{P}^0(\mathcal{T}_\Gamma)$ mit

$$\begin{aligned} (\nabla U, \nabla V)_{L^2(\Omega)} + \langle \mathcal{W}U + (\mathcal{K}' - 1/2)\Phi, V \rangle_\Gamma &= (\mathbf{M}, \nabla V)_{L^2(\Omega)}, \\ \langle \Psi, \mathcal{V}\Phi - (\mathcal{K} - 1/2)U \rangle_\Gamma &= 0 \end{aligned}$$

für alle $(V, \Psi) \in \mathcal{S}^1(\mathcal{T}) \times \mathcal{P}^0(\mathcal{T}_\Gamma)$.

Output: Approximationslösung $U \in \mathcal{S}^1(\mathcal{T})$.

Aus Algorithmus 4.3 ergibt sich die Approximation für das magnetische Streufeld zu $\mathbf{H}_h = -\nabla U \in (\mathcal{P}^0(\mathcal{T}))^d$.

Bemerkung 4.2. Es existiert eine eindeutige Lösung der Variationsformulierung in Schritt 1 in Algorithmus 4.3. Ein Beweis ist etwa in FEISCHL 2010 [7] zu finden.

4.3.3 A priori Analysis

Die Approximationslösung aus Algorithmus 4.3 ist eine Quasi-Bestapproximation der exakten Lösung im Sinne des Céa-Lemmas.

Satz 4.8. Sei $\mathbf{M} \in (H^1(\Omega))^d$ gegeben. Des weiteren sei $(u, \phi) \in H^1(\Omega) \times H^{-1/2}(\Gamma)$ die eindeutig bestimmte Lösung von (4.32) und $(U, \Phi) \in \mathcal{S}^1(\mathcal{T}) \times \mathcal{P}^0(\mathcal{T}_\Gamma)$ die Lösung von Algorithmus 4.3, dann gilt

$$\|u - U\|_{H^1(\Omega)} + \|\phi - \Phi\|_{H^{-1/2}(\Gamma)} \leq C \inf_{(V, \Psi) \in X \times Y} (\|u - V\|_{H^1(\Omega)} + \|\phi - \Psi\|_{H^{-1/2}(\Gamma)})$$

mit einer Konstanten $C > 0$ und $X := \mathcal{S}^1(\mathcal{T})$ bzw. $Y := \mathcal{P}^0(\mathcal{T}_\Gamma)$.

4.4 Direkte FEM/BEM - Kopplungsmethode

Als nächsten Ansatz behandeln wir die direkte FEM/BEM-Kopplung, welche auch JOHNSON/NÉDÉLEC Kopplung genannt wird. Wir werden die wichtigsten Resultate über eindeutige Lösbarkeit und Konvergenz aus der Arbeit von JOHNSON / NÉDÉLEC 1980 [14] zitieren.

Eine Analyse der symmetrischen und direkten FEM/BEM - Kopplungsmethode ist etwa auch in FEISCHL 2010 [7] zu finden.

4.4.1 Formulierung

Seien $f \in L^2(\Omega)$, $g_0 \in H^{1/2}(\Gamma)$ und $g_1 \in H^{-1/2}(\Gamma)$ gegeben. Dann lautet der Ansatz: Finde $(u, \phi) \in H^1(\Omega) \times H^{-1/2}(\Gamma)$ mit

$$\begin{aligned} (\nabla u, \nabla v)_{L^2(\Omega)} - \langle \phi, \gamma_0 v \rangle_\Gamma &= \langle f, v \rangle - \langle g_1, \gamma_0 v \rangle_\Gamma, \\ \langle \psi, \mathcal{V}\phi - (\mathcal{K} - 1/2)\gamma_0 u \rangle_\Gamma &= \langle \psi, (\mathcal{K} - 1/2)g_0 \rangle \end{aligned} \quad (4.33)$$

für alle $(v, \psi) \in H^1(\Omega) \times H^{-1/2}(\Gamma)$. Für den Fall $d = 2$ skalieren wir das Gebiet Ω so, dass $\text{diam}(\Omega) < 1$, damit der Operator $\mathcal{V}$ positiv definit ist und somit eine eindeutige Lösung gewährleistet ist.

Es existiert eine eindeutige Lösung von (4.33). Ein Beweis dieser Aussage ist in JOHNSON / NÉDÉLEC 1980 [14] zu finden.

Satz 4.9. Seien $f \in L^2(\Omega)$, $g_0 \in H^{1/2}(\Gamma)$ und $g_1 \in H^{-1/2}(\Gamma)$ gegeben, dann ist (4.33) und das Transmissionsproblem (TP) — siehe Definition 2.12 — äquivalent in folgendem Sinne.

Ist $(u^{int}, u^{ext}) \in H^1(\Omega) \times H^1_{lok}(\Omega^{ext})$ eine Lösung von (TP), dann löst das Paar $(u^{int}, \phi) \in H^1(\Omega) \times H^{-1/2}(\Gamma)$ das Problem (4.33) mit $u := u^{int}$ und $\phi := \gamma_1^{ext} u^{ext}$.

Gilt umgekehrt, dass $(u^{int}, \phi) \in H^1(\Omega) \times H^{-1/2}(\Gamma)$ eine Lösung von (4.33) ist, dann ist $(u^{int}, u^{ext}) \in H^1(\Omega) \times H^1_{lok}(\Omega^{ext})$ eine Lösung von (TP) mit $u^{int} := u$ und

$$u^{ext}(x) := \tilde{\mathcal{K}}(\gamma_0 u + g_0)(x) - \tilde{\mathcal{V}}\phi(x) \quad \text{für } x \in \Omega^{ext}. \quad (4.34)$$

Stellen wir also nun das Kopplungsproblem für die magnetostatischen Gleichungen auf. Bei gegebenem $\mathbf{M} \in (H^1(\Omega))^d$ $f = -\text{div } \mathbf{M}$ setzen wir $g_0 = 0$ und $g_1 = -\mathbf{M} \cdot \mathbf{n}$. Mit der ersten Green'schen Identität (4.23) können wir die rechte Seite der ersten Gleichung von (4.33) umschreiben:

$$(-\text{div } \mathbf{M}, v)_{L^2(\Omega)} + \langle \mathbf{M} \cdot \mathbf{n}, \gamma_0 v \rangle_\Gamma = (\mathbf{M}, \nabla v)_{L^2(\Omega)} \quad \text{für alle } v \in H^1(\Omega).$$

Somit ergibt sich das Kopplungsproblem für die magnetostatischen Gleichungen zu: Finde $(u, \phi) \in H^1(\Omega) \times H^{-1/2}(\Gamma)$, so dass

$$\begin{aligned} (\nabla u, \nabla v)_{L^2(\Omega)} - \langle \phi, \gamma_0 v \rangle_\Gamma &= (\mathbf{M}, \nabla v)_{L^2(\Omega)}, \\ \langle \psi, \mathcal{V}\phi - (\mathcal{K} - 1/2)\gamma_0 u \rangle_\Gamma &= 0, \end{aligned} \quad (4.35)$$

für alle $(v, \psi) \in H^1(\Omega) \times H^{-1/2}(\Gamma)$ gilt.

4.4.2 Numerische Umsetzung

Als Ansatzraum für die Approximationslösung wählen wir $\mathcal{S}^1(\mathcal{T}) \subset H^1(\Omega)$ und $\mathcal{P}^0(\mathcal{T}_\Gamma) \subset H^{-1/2}(\Gamma)$. Die Vorgangsweise beim Lösen des Problems lässt sich in folgendem Algorithmus zusammenfassen.

Algorithmus 4.4. Input: Magnetisierung $\mathbf{M}$ und Triangulierung $\mathcal{T}$.

Schritt 1: Bestimme $(U, \Phi) \in \mathcal{S}^1(\mathcal{T}) \times \mathcal{P}^0(\mathcal{T}_\Gamma)$ mit

$$\begin{aligned} (\nabla U, \nabla V)_{L^2(\Omega)} - (\Phi, V)_{L^2(\Gamma)} &= (\mathbf{M}, \nabla V)_{L^2(\Omega)}, \\ (\Psi, \mathcal{V}\Phi - (\mathcal{K} - 1/2)U)_{L^2(\Gamma)} &= 0 \end{aligned}$$

für alle $(V, \Psi) \in \mathcal{S}^1(\mathcal{T}) \times \mathcal{P}^0(\mathcal{T}_\Gamma)$.

Output: Approximationslösung $U \in \mathcal{S}^1(\mathcal{T})$.

Aus Algorithmus 4.4 ergibt sich die Approximation für das magnetische Streufeld zu $\mathbf{H}_h = -\nabla U \in (\mathcal{P}^0(\mathcal{T}))^d$.

Bemerkung 4.3. Es existiert eine eindeutige Lösung der Variationsformulierung in Schritt 1 in Algorithmus 4.4.

4.4.3 A priori Analysis

Die Approximationslösung aus Algorithmus 4.4 ist eine quasi-Bestapproximation der exakten Lösung im Sinne des Céa-Lemmas.

Satz 4.10. *Sei $\mathbf{M} \in (H^1(\Omega))^d$ gegeben. Des weiteren sei $(u, \phi) \in H^1(\Omega) \times H^{-1/2}(\Gamma)$ die eindeutig bestimmte Lösung von (4.35) und $(U, \Phi) \in \mathcal{S}^1(\mathcal{T}) \times \mathcal{P}^0(\mathcal{T}_\Gamma)$ die Lösung von Algorithmus 4.4, dann gilt*

$$\|u - U\|_{H^1(\Omega)} + \|\phi - \Phi\|_{H^{-1/2}(\Gamma)} \leq C \inf_{(V, \Psi) \in X \times Y} (\|u - V\|_{H^1(\Omega)} + \|\phi - \Psi\|_{H^{-1/2}(\Gamma)})$$

mit einer Konstanten $C > 0$ und $X := \mathcal{S}^1(\mathcal{T})$ bzw $Y := \mathcal{P}^0(\mathcal{T}_\Gamma)$.

Kapitel 5

Analytische Lösung

Die Lösungen der magnetostatischen Gleichungen lassen sich im Allgemeinen nur für spezielle Geometrien analytisch darstellen. Ein Beispiel für solch ein Gebiet ist etwa eine Kreisscheibe bzw. eine Kugel. Zunächst setzen wir voraus, dass $\operatorname{div} \mathbf{M} = 0$ in Ω gilt und daher vereinfachen sich die zu lösenden Gleichungen zu

$$\begin{aligned} -\Delta u^{int} &= 0 && \text{in } \Omega, \\ -\Delta u^{ext} &= 0 && \text{in } \Omega^{ext}, \\ [\gamma_0 u] &= 0 && \text{auf } \Gamma, \\ [\gamma_1 u] &= -\mathbf{M} \cdot \mathbf{n} && \text{auf } \Gamma, \\ u^{ext}(\mathbf{x}) &= \mathcal{O}\left(\frac{1}{\|\mathbf{x}\|}\right) && \text{für } \|\mathbf{x}\| \rightarrow \infty. \end{aligned} \tag{5.1}$$

Aus den Eigenschaften des Einfeldschichtpotentials ergibt sich als Lösung

$$u = \tilde{\mathcal{V}}(\mathbf{M} \cdot \mathbf{n}).$$

Man könnte nun versuchen dieses Integral zu berechnen, um auf eine Lösung zu kommen. Wir wählen jedoch einen anderen Ansatz, bei dem wir die Lösung im Innen- bzw. Außenraum als Überlagerung von Funktionen, die die Laplace-Gleichung erfüllen, darstellen, d.h.

$$\begin{aligned} u^{int} &= \sum_i \alpha_i f_i, \\ u^{ext} &= \sum_j \beta_j g_j, \end{aligned}$$

mit

$$\begin{aligned} \Delta f_i &= 0 && \text{in } \Omega, \\ \Delta g_j &= 0 && \text{in } \Omega^{ext}. \end{aligned}$$

Die Koeffizienten im Superpositionsansatz werden dabei so gewählt, dass die Randbedingungen in (5.1) erfüllt werden.

Für spezielle Magnetisierungen mit $\operatorname{div} \mathbf{M} \neq 0$ können ebenso Lösungen gefunden werden. Dazu überlagert man eine Lösung von (5.1) mit einer partikulären Lösung, also einer Lösung u^{par} , die $\Delta u^{par} = \operatorname{div} \mathbf{M}$ in Ω erfüllt.

Die im Folgenden angegebenen Beispiele werden in Kapitel 7 dazu benutzt, um die numerische Implementierung zu testen.

5.1 Zweidimensionaler Fall

Als Gebiet Ω wählen wir eine offene Kreisscheibe mit Mittelpunkt $(0, 0)$ und Radius $R > 0$,

$$\Omega := K_R(0, 0) = \{(x, y) \in \mathbb{R}^2 \mid x^2 + y^2 < R^2\}.$$

Der Außenraum ergibt sich daher zu

$$\Omega^{ext} := \mathbb{R}^2 \setminus \overline{\Omega}.$$

Wir wollen das Laplace-Problem

$$-\Delta u = 0 \tag{5.2}$$

lösen. Aufgrund der gewählten Geometrie, liegt es nahe das Problem in Polarkoordinaten (r, φ) zu beschreiben. Zwischen Polarkoordinaten und kartesischen Koordinaten (x, y) gelten die Beziehungen

$$\begin{aligned} x &= r \cos \varphi, \\ y &= r \sin \varphi \end{aligned}$$

mit $(x, y) \in \mathbb{R}^2$ und $(r, \varphi) \in [0, \infty) \times [0, 2\pi)$. Dabei wird der Polarwinkel des Punktes $(x, y) = (0, 0)$ als $\varphi = 0$ gewählt, damit die obigen Beziehungen eindeutig sind.

Nun gehen wir analog wie in HEUSER 2004 [12, Nr. 146] vor und stellen den Laplace-Operator in Polarkoordinaten dar. Für eine Funktion $u(r, \varphi)$ lautet die Laplace-Gleichung (5.2)

$$\frac{\partial^2 u}{\partial r^2} + \frac{1}{r} \frac{\partial u}{\partial r} + \frac{1}{r^2} \frac{\partial^2 u}{\partial \varphi^2} = 0 \quad \text{für } r \neq 0. \tag{5.3}$$

Wir betrachten nun zum einen Lösungen die unabhängig von φ und zum anderen Lösungen die von φ abhängig sind. Eine Lösung u die unabhängig von φ ist, erfüllt die Differentialgleichung

$$\frac{\partial^2 u}{\partial r^2} + \frac{1}{r} \frac{\partial u}{\partial r} = 0.$$

Nach Umformung erhält man die Gleichung

$$\frac{d}{dr} \left(r \frac{du}{dr} \right) = 0,$$

welche durch zweimaliges Integrieren auf die Lösung

$$u(r, \varphi) = \alpha + \beta \ln r \tag{5.4}$$

führt, wobei $\alpha, \beta \in \mathbb{R}$ beliebige Konstanten sind.

Um nun Lösungen von (5.3) zu erhalten, die auch von φ abhängen, benutzen wir den Separationsansatz $u(r, \varphi) = v(r)w(\varphi)$ und setzen in die Laplace-Gleichung ein. Dies führt auf

$$r^2 \frac{v''(r)}{v(r)} + r \frac{v'(r)}{v(r)} = -\frac{\ddot{w}(\varphi)}{w(\varphi)},$$

wobei die Differentiation nach r mit einem Strich und die Differentiation nach φ mit einem Punkt bezeichnet wird. Da die linke Seite nur von r und die rechte Seite nur von φ abhängt und obige Beziehung für alle $(r, \varphi) \in (0, \infty) \times [0, 2\pi)$ gilt, muss die linke bzw. rechte Seite konstant sein. Als Konstante wählen wir $\lambda \in \mathbb{R}$ mit

$$\lambda := r^2 \frac{v''(r)}{v(r)} + r \frac{v'(r)}{v(r)} = -\frac{\ddot{w}(\varphi)}{w(\varphi)}.$$

Dies führt uns daher auf folgende zwei Gleichungen:

$$r^2 v''(r) + r v'(r) - \lambda v(r) = 0, \quad (5.5)$$

$$\ddot{w}(\varphi) + \lambda w(\varphi) = 0. \quad (5.6)$$

Zu beachten ist, dass die Gleichung (5.6) für alle $\varphi \in \mathbb{R}$ bei beliebigem λ Lösungen besitzt. Wir interessieren uns jedoch nur für Lösungen, die 2π periodisch und stetig differenzierbar sind, es gilt also

$$w(0) = w(2\pi) \quad \text{und} \quad \dot{w}(0) = \dot{w}(2\pi). \quad (5.7)$$

Des weiteren gilt

$$\dot{w}(\varphi) \neq 0 \quad \text{für fast alle } \varphi,$$

da ansonsten bereits (5.4) die Lösung der Laplace-Gleichung darstellt. Aus den letzten Bedingungen können wir noch die Werte von λ einschränken, denn es gilt

$$\lambda \int_0^{2\pi} w^2 d\varphi = - \int_0^{2\pi} w \ddot{w} d\varphi = \underbrace{-w \dot{w}}_{=0} \Big|_0^{2\pi} + \int_0^{2\pi} \dot{w}^2 d\varphi > 0,$$

und daher muss $\lambda > 0$ sein.

Eine allgemeine Lösung von Gleichung (5.6) ist durch

$$w(\varphi) = \alpha e^{i\sqrt{\lambda}\varphi} + \beta e^{-i\sqrt{\lambda}\varphi}$$

gegeben, wobei α, β beliebige komplexe Konstanten sind. Einsetzen der allgemeinen Lösung in die Beziehungen (5.7) führt auf

$$\begin{aligned} \alpha + \beta &= \alpha e^{i\sqrt{\lambda}2\pi} + \beta e^{-i\sqrt{\lambda}2\pi}, \\ \alpha i\sqrt{\lambda} - \beta i\sqrt{\lambda} &= \alpha i\sqrt{\lambda} e^{i\sqrt{\lambda}2\pi} - \beta i\sqrt{\lambda} e^{-i\sqrt{\lambda}2\pi}, \end{aligned} \quad (5.8)$$

wodurch sich sich weitere Einschränkungen an λ erzielen lassen. Zunächst geben wir die Gleichungen (5.8) in Matrixschreibweise an

$$\begin{pmatrix} 1 - e^{i\sqrt{\lambda}2\pi} & 1 - e^{-i\sqrt{\lambda}2\pi} \\ i\sqrt{\lambda}2\pi (1 - e^{i\sqrt{\lambda}2\pi}) & i\sqrt{\lambda} (e^{-i\sqrt{\lambda}2\pi} - 1) \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}. \quad (5.9)$$

Die Gleichungen (5.8) können daher nur gelten, falls die Matrix in (5.9) nicht regulär ist, d.h. falls die Determinante dieser Matrix verschwindet. Nach Berechnen der Determinante und Nullsetzen derselben erhält man die Bedingung

$$4i\sqrt{\lambda} (\cos(\sqrt{\lambda}2\pi) - 1) = 0,$$

und nach kurzer Rechnung bekommt man für λ die Einschränkung $\sqrt{\lambda} \in \mathbb{N}$ und somit

$$\lambda = n^2 \quad \text{mit } n \in \mathbb{N}.$$

Wir müssen nun die Differentialgleichungen

$$r^2 v''(r) + r v'(r) - n^2 v(r) = 0 \quad \text{und} \quad \ddot{w}(\varphi) + n^2 w(\varphi) = 0 \quad \text{mit } n \in \mathbb{N}$$

für $r \neq 0$ lösen. Die erste Gleichung wird auch Euler'sche Differentialgleichung genannt. Nach HEUSER 2004 [12, Nr. 146, Aufgabe 2] lassen sich alle Lösungen der Gleichung in der Form

$$v(r) = Ar^n + Br^{-n}$$

mit beliebigen Konstanten A, B schreiben.

Somit ist für $r \neq 0$ mit

$$u_n(r, \varphi) := (\alpha_n e^{in\varphi + \beta_n} + \beta_n e^{-in\varphi}) r^n + (\alpha_{-n} e^{-in\varphi} + \beta_{-n} e^{in\varphi}) r^{-n}$$

eine Lösung der Laplace-Gleichung $-\Delta u_n = 0$ gegeben.

Mit (5.4) und u_n ist daher

$$u(r, \varphi) = \alpha + \beta \ln r + \sum_{m \in \mathbb{Z} \setminus \{0\}} r^m (\alpha_m e^{im\varphi} + \beta_m e^{-im\varphi}) \quad (5.10)$$

formal eine Lösung der Laplace-Gleichung.

Wir betrachten nun diesen Lösungsansatz für u^{int} bzw. u^{ext} . Damit u^{ext} die Abklingbedingung erfüllen kann, müssen die Koeffizienten $\alpha, \beta, \alpha_m, \beta_m$ mit $m > 0$ in (5.10) allesamt verschwinden. Des weiteren haben wir vorausgesetzt, dass $\operatorname{div} \mathbf{M}(x, y) = 0$ für alle $(x, y) \in K_R(0, 0)$, weshalb u^{int} keine Singularität am Nullpunkt besitzen darf. Daher müssen für u^{int} die Koeffizienten β, α_m, β_m mit $m < 0$ verschwinden. Da das magnetostatische Potential eine reelle Funktion ist, muss auch $u^{int} = \overline{u^{int}}$ und $u^{ext} = \overline{u^{ext}}$ gelten.

Insgesamt ergibt sich daher für u^{int} bzw. u^{ext} der Ansatz

$$u^{int}(r, \varphi) = u_0 + \sum_{m=1}^{\infty} r^m (\alpha_m e^{im\varphi} + \overline{\alpha_m} e^{-im\varphi}), \quad (5.11)$$

bzw.

$$u^{ext}(r, \varphi) = \sum_{m=1}^{\infty} r^{-m} (\beta_m e^{im\varphi} + \overline{\beta_m} e^{-im\varphi}). \quad (5.12)$$

Die Koeffizienten $u_0 \in \mathbb{R}$ und $\alpha_m, \beta_m \in \mathbb{C}$ ergeben sich aus den Übergangsbedingungen am Rand. Dazu setzen wir noch zusätzlich voraus, dass $\mathbf{M} \cdot \mathbf{n} = \mathbf{M} \cdot \mathbf{n}(\varphi) \in L^2([0, 2\pi])$ gilt.

Satz 5.1 (TESCHL 2009 [21, Satz 10.5]). *Die Funktionen $f_m(\varphi) := \frac{1}{\sqrt{2\pi}} e^{im\varphi}$ mit $m \in \mathbb{Z}$ bilden ein vollständiges Orthonormalsystem in $L^2([0, 2\pi])$. Es gilt daher*

$$(f_n, f_m)_{L^2([0, 2\pi])} = \int_0^{2\pi} f_n \overline{f_m} d\varphi = \frac{1}{2\pi} \int_0^{2\pi} e^{i(n-m)\varphi} d\varphi = \begin{cases} 1 & \text{falls } n = m, \\ 0 & \text{sonst.} \end{cases}$$

Wir können also die Randfunktion $\mathbf{M} \cdot \mathbf{n}$ in die Reihe (Fourierreihe)

$$\mathbf{M} \cdot \mathbf{n} = M_0 + \sum_{m=1}^{\infty} M_m e^{im\varphi} + \overline{M_m} e^{-im\varphi} \quad (5.13)$$

zerlegen, wobei der Koeffizient M_0 reell und die Koeffizienten M_m komplex sind, da die Magnetisierung eine reelle Funktion ist.

Die Koeffizienten von $\mathbf{M} \cdot \mathbf{n}$ ergeben sich zu

$$M_m = \frac{1}{2\pi} \int_0^{2\pi} e^{-im\varphi} \mathbf{M} \cdot \mathbf{n}(\varphi) d\varphi \quad \text{für } m \in \mathbb{N}_0.$$

Da $\operatorname{div} \mathbf{M} = 0$ und

$$\int_{\Omega} \operatorname{div} \mathbf{M} dx = \int_{\Gamma} \mathbf{M} \cdot \mathbf{n} ds_y = R \int_0^{2\pi} \mathbf{M} \cdot \mathbf{n}(\varphi) d\varphi$$

gilt, folgt $M_0 = 0$.

Bevor wir noch zu diversen Beispielen kommen, möchten wir für eine weitreichendere Analysis des Problems noch auf HEUSER 2004 [12, Nr. 146] verweisen. Die Theorie von Fourierreihen wird etwa in HEUSER 2004 [12, Kapitel 17] behandelt.

Wir benötigen noch einige Konventionen. Mit $\partial_i f$ bezeichnen wir die Ableitung einer Funktion f nach der i -ten Komponente. Die i -te Komponente des Vektorfelds $\mathbf{F}$ notieren wir mit $\mathbf{F}^i$.

Als Richtungsvektor zum Punkt $(x, y) = (r \cos \varphi, r \sin \varphi) \in \mathbb{R}^2$ bezeichnen wir den Vektor

$$e_r = \begin{pmatrix} \cos \varphi \\ \sin \varphi \end{pmatrix},$$

mit den Eigenschaften $\|e_r\|^2 = \sum_{i=1}^2 e_r^i e_r^i = 1$. Wir definieren den dazu orthogonalen Vektor

$$e_\varphi := \begin{pmatrix} -\sin \varphi \\ \cos \varphi \end{pmatrix},$$

mit $\|e_\varphi\|^2 = \sum_{i=1}^2 e_\varphi^i e_\varphi^i = 1$ und $\sum_{i=1}^2 e_\varphi^i e_r^i = 0$. Man kann die Vektoren e_r bzw. e_φ auch durch Ableiten von r bzw. φ erhalten

$$\begin{aligned} e_r^i &= \partial_i r, \\ \frac{1}{r} e_\varphi^i &= \partial_i \varphi. \end{aligned}$$

Wir können jetzt in die Übergangsbedingungen des Gleichungssystems (5.1) einsetzen. Die erste Bedingung besagt, dass das Potential stetig ist, daher müssen u^{int} und u^{ext}

$$[\gamma_0 u] = \lim_{r \rightarrow R^+} u^{ext} - \lim_{r \rightarrow R^-} u^{int} = 0 \quad (5.14)$$

erfüllen. Einsetzen unserer Ansätze für u^{int} bzw. u^{ext} liefert

$$-u_0 + \sum_{m=1}^{\infty} (R^{-m}(\beta_m e^{im\varphi} + \overline{\beta_m} e^{-im\varphi}) - R^m(\alpha_m e^{im\varphi} + \overline{\alpha_m} e^{-im\varphi})) = 0.$$

Wir multiplizieren die letzte Gleichung mit $e^{-in\varphi}$ und integrieren über das Intervall $[0, 2\pi]$. Anwendung von Satz 5.1 bedingt für die Koeffizienten

$$\begin{aligned} u_0 &= 0, \\ \beta_n &= \alpha_n R^{2n} \quad \text{für } n \in \mathbb{N}. \end{aligned} \quad (5.15)$$

Bei dem Sprung der Normalenableitungen am Rand kommt nun auch die Magnetisierung ins Spiel. Daher gilt

$$[\gamma_1 u] = \lim_{r \rightarrow R^+} \nabla u^{ext} \cdot \mathbf{n} - \lim_{r \rightarrow R^-} \nabla u^{int} \cdot \mathbf{n} = -\mathbf{M} \cdot \mathbf{n}. \quad (5.16)$$

Wir setzen wiederum unsere Ansätze für u^{int} , u^{ext} und $\mathbf{M} \cdot \mathbf{n}$ ein und benutzen, dass $\sum_{i=1}^2 (\partial_i \varphi) e_r^i = \sum_{i=1}^2 \frac{1}{r} e_\varphi^i e_r^i = 0$

und $\sum_{i=1}^2 e_r^i e_r^i = 1$ gilt und erhalten daher

$$\begin{aligned} \sum_{i=1}^2 (\partial_i (u^{ext} - u^{int}) e_r^i)|_{r=R} &= \sum_{m=1}^{\infty} ((-m) R^{-m-1} (\beta_m e^{im\varphi} + \overline{\beta_m} e^{-im\varphi}) - m R^{m-1} (\alpha_m e^{im\varphi} + \overline{\alpha_m} e^{-im\varphi})) \\ &= -M_0 - \sum_{m=1}^{\infty} (M_m e^{im\varphi} + \overline{M_m} e^{-im\varphi}). \end{aligned}$$

Multiplikation der letzten Gleichung mit $e^{-in\varphi}$ und integrieren über das Intervall $[0, 2\pi]$ ergibt für die Koeffizienten die Gleichung

$$-nR^{-n-1}\beta_n - nR^{n-1}\alpha_n = -M_n, \quad (5.17)$$

für $n \in \mathbb{N}$. Setzen wir nun (5.15) in (5.17) ein, so ergibt sich

$$\alpha_n = \frac{M_n}{2nR^{n-1}},$$

$$\beta_n = \frac{M_n}{2n}R^{n+1},$$

mit $n \in \mathbb{N}$.

Wir wollen uns nun mit einigen Beispielen beschäftigen, anhand derer wir dann in Kapitel 7 die numerische Implementierung testen können.

Beispiel 1

Das Gebiet sei die Einheitskreisscheibe und als Magnetisierung wählen wir

$$\mathbf{M} = \begin{pmatrix} M_x \\ M_y \end{pmatrix} \in \mathbb{R}^2.$$

Der Normalenvektor auf dem Rand ist durch

$$\mathbf{n} = \begin{pmatrix} \cos \varphi \\ \sin \varphi \end{pmatrix}$$

gegeben. Die Koeffizienten M_n ergeben sich zu

$$M_n = \frac{1}{2\pi} \int_0^{2\pi} e^{-in\varphi} (M_x \cos \varphi + M_y \sin \varphi) d\varphi = \begin{cases} \frac{1}{2}(M_x - iM_y) & n = 1, \\ 0 & n > 1, \end{cases}$$

und daher gilt

$$\alpha_n = \begin{cases} \frac{1}{4}(M_x - iM_y) & n = 1, \\ 0 & n > 1, \end{cases}$$

bzw.

$$\beta_n = \begin{cases} \frac{1}{4}(M_x - iM_y) & n = 1, \\ 0 & n > 1. \end{cases}$$

Somit können wir die Lösungen für u^{int} und u^{ext} anschreiben:

$$u^{int}(r, \varphi) = \frac{r}{2}(M_x \cos \varphi + M_y \sin \varphi), \quad (5.18)$$

$$u^{ext}(r, \varphi) = \frac{1}{2r}(M_x \cos \varphi + M_y \sin \varphi). \quad (5.19)$$

Die Funktion u^{int} hat in kartesischen Koordinaten die Darstellung

$$u^{int}(x, y) = \frac{1}{2}(M_x x + M_y y),$$

welche auch in Abbildung 5.1 mit $\mathbf{M} = (1, 0)^T$ visualisiert wird.

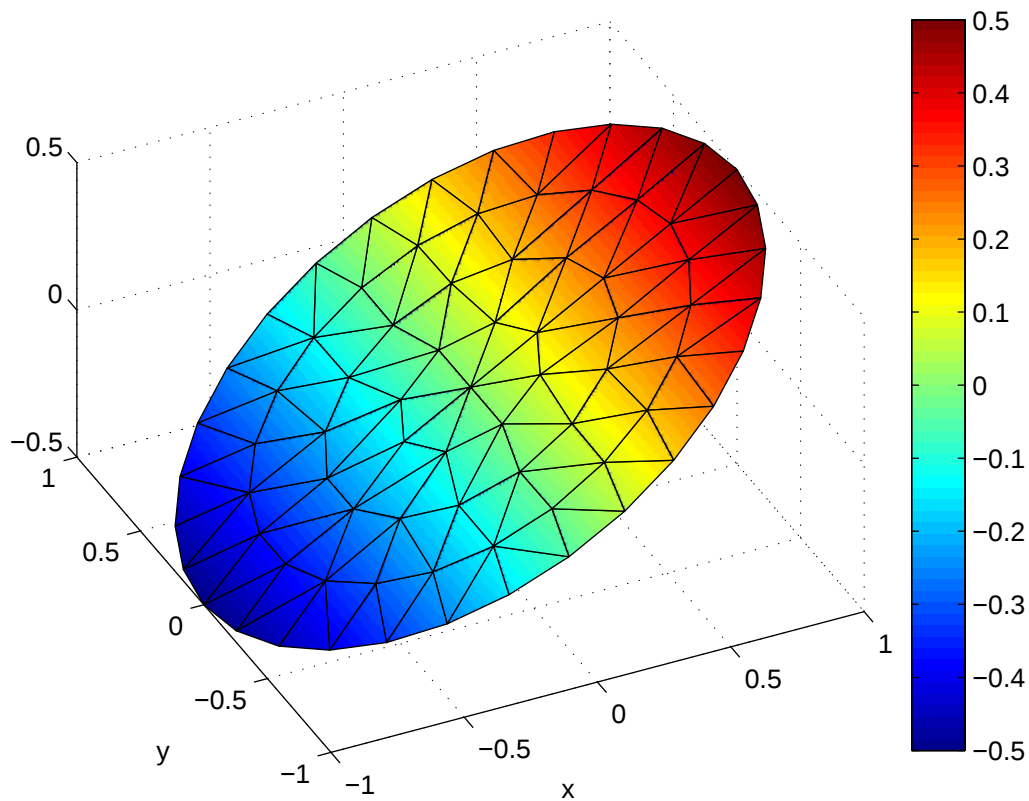


Abbildung 5.1: Lösung der magnetostatischen Gleichungen im Innenraum bei konstanter Magnetisierung $\mathbf{M} = (1, 0)^T$ (vgl. Beispiel 1).

Beispiel 2

Es sei wiederum Ω die Einheitskreisscheibe. Wir wählen als Magnetisierung

$$\mathbf{M}(x, y) = \begin{pmatrix} x \\ -y \end{pmatrix},$$

und daher ist $\mathbf{M} \cdot \mathbf{n} = \cos^2 \varphi - \sin^2 \varphi$. Nach kurzer Rechnung ergibt sich

$$M_n = \begin{cases} \frac{1}{2} & \text{falls } n = 2, \\ 0 & \text{sonst,} \end{cases}$$

und daher

$$\alpha_n = \beta_n = \begin{cases} \frac{1}{8} & \text{falls } n = 2, \\ 0 & \text{sonst.} \end{cases}$$

Die Lösungen u^{int} und u^{ext} ergeben sich mit $\frac{r^2}{8}(e^{i2\varphi} + e^{-i2\varphi}) = \frac{r^2}{4} \cos 2\varphi$ somit zu

$$\begin{aligned} u^{int}(r, \varphi) &= \frac{r^2}{4} \cos(2\varphi), \\ u^{ext}(r, \varphi) &= \frac{1}{4r^2} \cos(2\varphi). \end{aligned}$$

Wir können die Funktion u^{int} auch in kartesischen Koordinaten darstellen

$$u^{int}(x, y) = \frac{1}{4}(x^2 - y^2), \quad (5.20)$$

welche auch in Abbildung 5.2 visualisiert ist.

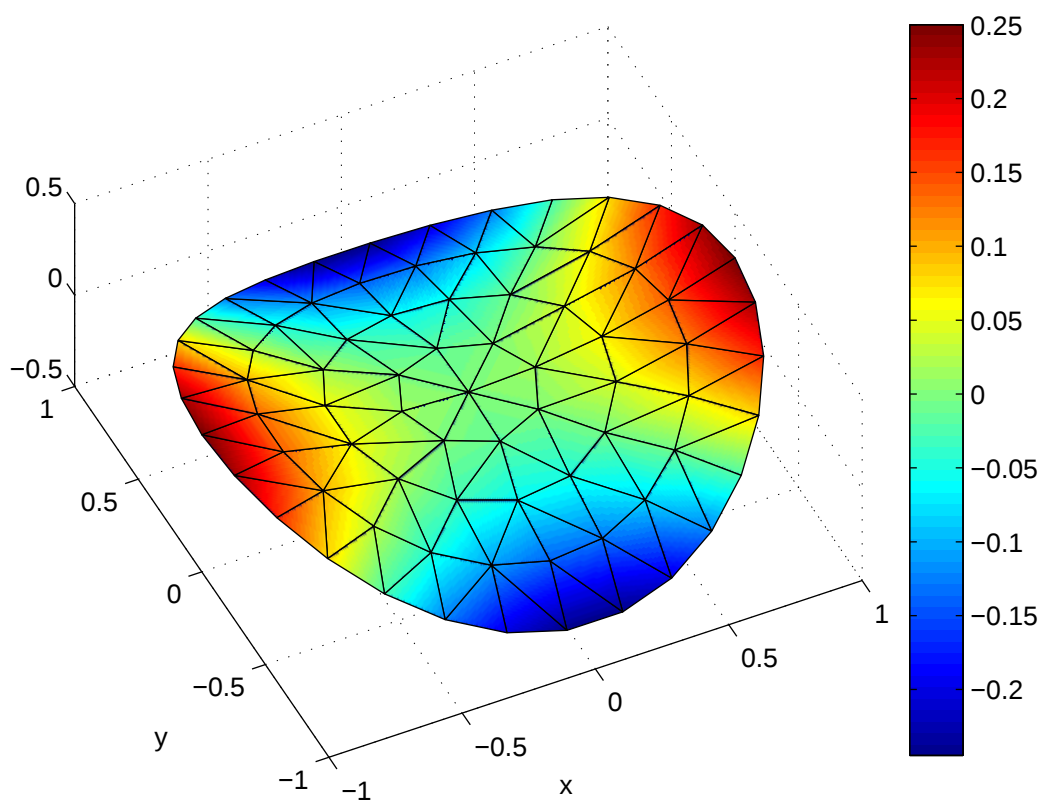


Abbildung 5.2: Lösung der magnetostatischen Gleichungen im Innenraum mit Magnetisierung $\mathbf{M} = (x, -y)^T$ (vgl. Beispiel 2).

Beispiel 3

Wir betrachten noch ein Beispiel, wo der Quellenterm $\operatorname{div} \mathbf{M}$ nicht verschwindet, d.h. es gilt $\operatorname{div} \mathbf{M} \neq 0$. Dazu wählen wir den Superpositionsansatz $u^{int} = u^{hom} + u^{par}$, wobei u^{hom} eine Lösung der Laplace-Gleichung $\Delta u^{hom} = 0$ und u^{par} eine Lösung der Poisson-Gleichung $\Delta u^{hom} = \operatorname{div} \mathbf{M}$ ist. Für die Funktion u^{hom} wählen wir den Ansatz wie in (5.11).

Als Magnetisierung benützen wir in diesem Beispiel

$$\mathbf{M} = \begin{pmatrix} x \\ y \end{pmatrix}.$$

Das Gebiet Ω sei wiederum die Einheitskreisscheibe. Mit dieser Annahme erhalten wir

$$\mathbf{M} \cdot \mathbf{n} = \begin{pmatrix} \cos \varphi \\ \sin \varphi \end{pmatrix} \cdot \begin{pmatrix} \cos \varphi \\ \sin \varphi \end{pmatrix} = (\cos^2 \varphi + \sin^2 \varphi) = 1.$$

Die Divergenz der Magnetisierung ergibt sich zu

$$\operatorname{div} \mathbf{M} = 2.$$

Wir suchen nun eine partikuläre Lösung u^{par} die der Differentialgleichung

$$-\Delta u^{par} = -2 \quad \text{in } \Omega \quad (5.21)$$

genügt. Durch Einsetzen von

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

in (5.21) erkennt man, dass es sich um eine Lösung handelt.

Für die Lösung im Außenraum kann wiederum der Ansatz (5.12) gewählt werden. Somit sind unsere Lösungsansätze für Innen- und Außenraum durch

$$\begin{aligned} u^{int}(r, \varphi) &= u^{par}(r, \varphi) + u^{hom}(r, \varphi) = \frac{1}{2}r^2 + u_0 + \sum_{m=1}^{\infty} r^m (\alpha_m e^{im\varphi} + \overline{\alpha_m} e^{-im\varphi}), \\ u^{ext}(r, \varphi) &= \sum_{m=1}^{\infty} r^{-m} (\beta_m e^{im\varphi} + \overline{\beta_m} e^{-im\varphi}) \end{aligned}$$

gegeben. Die Koeffizienten u_0, α_m, β_m können nun wieder durch die Übergangsbedingungen bestimmt werden. Die Vorgangsweise ist dabei die gleiche wie zu Beginn dieses Abschnitts. Es ergeben sich die Koeffizienten zu

$$\begin{aligned} u_0 &= -\frac{1}{2}, \\ \alpha_m &= \beta_m = 0 \quad \text{für } m \in \mathbb{N}. \end{aligned}$$

Somit ist die Lösung der magnetostatischen Gleichungen gegeben durch

$$\begin{aligned} u^{int}(r, \varphi) &= \frac{1}{2}(r^2 - 1), \\ u^{ext}(r, \varphi) &= 0. \end{aligned}$$

Wir können die Funktion u^{int} auch in kartesischen Koordinaten anschreiben:

$$u^{int}(x, y) = \frac{1}{2}(x^2 + y^2 - 1). \quad (5.22)$$

Bemerkung 5.1. Normalerweise ist bei der Berechnung des magnetostatischen Potentials die Magnetisierung gegeben. Da wir die analytische Lösung des Problems später dazu benutzen wollen, um die in Kapitel 4 beschriebenen Ansätze zu testen, können wir auch eine beliebige Funktion $\tilde{u}$ vorgeben. Die Magnetisierung definieren wir dann als

$$\mathbf{M} := \nabla \tilde{u}.$$

Da $\Delta \tilde{u} = \operatorname{div}(\nabla \tilde{u})$ ist, gilt dann für die Funktion $\tilde{u}$

$$-\Delta \tilde{u} = -\operatorname{div} \mathbf{M} \quad \text{in } \Omega.$$

Wir führen dann den oben beschriebenen Lösungsweg durch und benützen dabei $\tilde{u}$ als partikuläre Lösung.

Beispiel 4

Als Gebiet verwenden wir eine Kreisscheibe mit Radius $R > 0$. Man kann auch Magnetisierungen betrachten, die nicht im Sobolevraum $(H^1(\Omega))^2$ liegen. Dazu wählen wir als Magnetisierung

$$\mathbf{M}(x, y) := \frac{\begin{pmatrix} x \\ y \end{pmatrix}}{\|(x, y)\|},$$

wobei wir mit den zu Beginn dieses Abschnitts getroffenen Konventionen auch

$$\mathbf{M}^i := e_r^i$$

schreiben können. Die Magnetisierung an einem Punkt (x, y) stimmt also mit dem Richtungsvektor in diesem Punkt überein. Zu beachten ist, dass diese Magnetisierung im Punkt $(0, 0)$ nicht stetig definiert werden kann.

Lemma 5.2. *Das Vektorfeld $\mathbf{M} = e_r$ liegt in $(L^2(\Omega))^2$ aber nicht in $(H^1(\Omega))^2$.*

Beweis. Es gilt $\sum_{i=1}^2 e_r^i e_r^i = 1$ und daher

$$\|\mathbf{M}^i\|_{L^2(\Omega)}^2 = \int_{\Omega} (e_r^i)^2 dx \leq \int_{\Omega} \sum_{i=1}^2 (e_r^i)^2 dx = |\Omega|.$$

Somit folgt die erste Behauptung.

Wir bezeichnen nun mit x^i die i -te Komponente des Vektors

$$\begin{pmatrix} x \\ y \end{pmatrix},$$

und mit r die Länge von (x, y) , dann gilt

$$\partial_i \frac{x^1}{r} = \delta_{i1} \frac{1}{r} - \frac{x^1 x^i}{r^2}.$$

Es ist

$$\begin{aligned} \sum_{i=1}^2 \left(\delta_{i1} \frac{1}{r} - \frac{x^1 x^i}{r^2} \right) \left(\delta_{i1} \frac{1}{r} - \frac{x^1 x^i}{r^2} \right) &= \sum_{i=1}^2 \left(\delta_{i1} \delta_{i1} \frac{1}{r^2} - 2 \frac{x^1 \delta_{i1} x^i}{r^4} + \frac{(x^1)^2 x^i x^i}{r^6} \right) \\ &= \frac{1}{r^2} - 2 \frac{(x^1)^2}{r^4} + \frac{(x^1)^2 r^2}{r^6} = \frac{1}{r^2} - 2 \frac{\cos^2 \varphi}{r^2} + \frac{\cos^2 \varphi}{r^2} = \frac{\sin^2 \varphi}{r^2} \end{aligned}$$

und daher

$$\|\nabla(\mathbf{M}^1)\|_{L^2(\Omega)}^2 = \int_0^{2\pi} \int_0^R r \frac{\sin^2 \varphi}{r^2} dr d\varphi = \infty.$$

□

Die Divergenz der Magnetisierung ergibt sich für $r \neq 0$ zu

$$\sum_{i=1}^2 \partial_i e_r^i = \sum_{i=1}^2 \partial_i \frac{x^i}{r} = \sum_{i=1}^2 \left(\frac{\partial_i x^i}{r} - \frac{x^i x^i}{r^2} \right) = \frac{1}{r}.$$

Wir gehen nun wie in Beispiel 3 vor. Mit $\partial_i r = e_r^i$ und obiger Umformung erhält man, dass

$$-\Delta r = -\frac{1}{r} \quad \text{in } \Omega$$

für $r \neq 0$ gilt und wir wählen daher als Partikulärlösung $u^{par} = r$. Die Ansätze für die Lösung im Innen- und Außenraum ergeben sich zu

$$u^{int}(r, \varphi) = u^{par}(r, \varphi) + u^{hom}(r, \varphi) = r + u_0 + \sum_{m=1}^{\infty} r^m (\alpha_m e^{im\varphi} + \overline{\alpha_m} e^{-im\varphi}),$$

$$u^{ext}(r, \varphi) = \sum_{m=1}^{\infty} r^{-m} (\beta_m e^{im\varphi} + \overline{\beta_m} e^{-im\varphi}).$$

Wir wählen nun den Radius der Kreisscheibe zu $R = 1$, damit gilt $\mathbf{M} \cdot \mathbf{n} = 1$. Einsetzen in die Übergangsbedingungen (5.14) und (5.16) liefert für die Koeffizienten

$$u_0 = -1,$$

$$\alpha_m = \beta_m = 0 \quad \text{für } m \in \mathbb{N},$$

und somit ist

$$u^{int}(r, \varphi) = r - 1,$$

$$u^{ext}(r, \varphi) = 0$$

für $r \neq 0$.

In kartesischen Koordinaten ist dann

$$u^{int}(x, y) = \sqrt{x^2 + y^2} - 1.$$

Die Funktion kann an der Stelle $(x, y) = (0, 0)$ stetig fortgesetzt werden.

5.2 Dreidimensionaler Fall

Als Gebiet wählen wir eine Kugel mit Radius $R > 0$ und Mittelpunkt $(x, y, z) = (0, 0, 0)$

$$\Omega := K_R(0, 0, 0) = \{(x, y, z) \in \mathbb{R}^3 \mid x^2 + y^2 + z^2 < R^2\}.$$

Nach Definition gilt dann $\Omega^{ext} = \mathbb{R}^3 \setminus \overline{\Omega}$. Wir werden jetzt ähnlich wie im vorigen Abschnitt vorgehen. Zunächst werden wir aufgrund der Geometrie unser Problem in Kugelkoordinaten beschreiben. Es gelten die Beziehungen

$$x = r \sin \theta \cos \varphi,$$

$$y = r \sin \theta \sin \varphi,$$

$$z = r \cos \theta,$$

mit $(r, \theta, \varphi) \in [0, \infty) \times [0, \pi) \times [0, 2\pi)$. Der Laplace-Operator lautet in Kugelkoordinaten für $r \neq 0$

$$\Delta u = \frac{1}{r} \frac{\partial^2}{\partial r^2} (ru) + \frac{1}{r^2} \left(\frac{1}{\sin \theta} \frac{\partial}{\partial \theta} \left(\sin \theta \frac{\partial}{\partial \theta} u \right) + \frac{1}{\sin^2 \theta} \frac{\partial^2}{\partial \varphi^2} u \right).$$

Um eine Lösung u von $-\Delta u = 0$ zu bestimmen, wählt man einen Separationsansatz der Form

$$u(r, \theta, \varphi) = R(r)\Theta(\theta)\Phi(\varphi).$$

Wir wollen uns im Folgenden kurz fassen (eine ausführlichere Behandlung dieses Problems ist z.B. in JACKSON 2006 [13, Kapitel 3] zu finden) und bemerken nur, dass dieser Ansatz auf Lösungen der Form

$$u_{\ell m}(r, \theta, \varphi) = \left(\alpha_{\ell m} r^\ell + \beta_{\ell m} \frac{1}{r^{\ell+1}} \right) Y_\ell^m(\theta, \varphi),$$

mit $\ell \in \mathbb{N}_0$ und $|m| \leq \ell$ führt. Dabei wird die Funktion

$$Y_\ell^m(\theta, \varphi) := \sqrt{\frac{2\ell+1}{4\pi} \frac{(\ell-m)!}{(\ell+m)!}} P_\ell^m(\cos \theta) e^{im\varphi}$$

als Kugelflächenfunktion bezeichnet und P_ℓ^m heißt assoziierte Legendre-Funktion, welche beispielsweise über

$$P_\ell^m(x) := \frac{(-1)^m}{2^\ell \ell!} (1-x^2)^{m/2} \frac{d^{\ell+m}}{dx^{\ell+m}} (x^2-1)^\ell$$

für $x \in [-1, 1]$ mit $\ell \in \mathbb{N}_0$ und $|m| \leq \ell$ definiert werden kann.

Wir können Lösungen des Laplace-Problems formal anschreiben als

$$u(r, \theta, \varphi) = \sum_{\ell=0}^{\infty} \sum_{m=-\ell}^{\ell} (\alpha_{\ell m} r^\ell + \beta_{\ell m} r^{-\ell-1}) Y_\ell^m(\theta, \varphi),$$

mit beliebigen Konstanten $\alpha_{\ell m}$ und $\beta_{\ell m}$.

Um nun einen Ansatz für die Lösung im Außenraum anzuschreiben, bemerken wir, dass die Koeffizienten vor r^ℓ mit $\ell \geq 0$ verschwinden müssen, damit die Abklingbedingungen erfüllt werden können. Wegen $\operatorname{div} \mathbf{M}(x, y, z) = 0$, darf die Lösung im Innenraum keine Singularitäten an der Stelle $(x, y, z) = (0, 0, 0)$ aufweisen, daher müssen die Koeffizienten vor $r^{-\ell-1}$ im obigen Ansatz wegfallen. Somit ergeben sich die Ansätze für die Lösung im Innen- bzw. Außenraum zu

$$u^{int}(r, \theta, \varphi) = \sum_{\ell=0}^{\infty} \sum_{m=-\ell}^{\ell} \alpha_{\ell m} r^\ell Y_\ell^m(\theta, \varphi), \quad (5.23)$$

bzw.

$$u^{ext}(r, \theta, \varphi) = \sum_{\ell=0}^{\infty} \sum_{m=-\ell}^{\ell} \beta_{\ell m} r^{-\ell-1} Y_\ell^m(\theta, \varphi). \quad (5.24)$$

Die Koeffizienten in (5.23) und (5.24) ergeben sich aus den Randbedingungen.

Im Folgenden bezeichnen wir mit $L^2([0, \pi], \sin \theta d\theta) \otimes L^2([0, 2\pi])$ den Raum aller messbaren Funktionen $f(\theta, \varphi)$ mit

$$\int_0^{2\pi} \int_0^\pi |f(\theta, \varphi)|^2 \sin \theta d\theta d\varphi < \infty.$$

Satz 5.3 (TESCHL 2009 [21, Satz 10.7]). *Die Kugelflächenfunktionen bilden ein vollständiges Orthonormalsystem in $L^2([0, \pi], \sin \theta d\theta) \otimes L^2([0, 2\pi])$. Sie erfüllen die Beziehung*

$$\int_0^{2\pi} \int_0^\pi \overline{Y_k^n}(\theta, \varphi) Y_\ell^m(\theta, \varphi) \sin \theta d\theta d\varphi = \delta_{k\ell} \delta_{nm},$$

für $k, \ell \in \mathbb{N}_0$ mit $|n| \leq k$ und $|m| \leq \ell$.

Wir setzen $\mathbf{M} \cdot \mathbf{n}(\theta, \varphi) \in L^2([0, \pi], \sin \theta d\theta) \otimes L^2([0, 2\pi])$ voraus und können die Funktion daher in die Reihe

$$\mathbf{M} \cdot \mathbf{n} = \sum_{\ell=0}^{\infty} \sum_{m=-\ell}^{\ell} M_{\ell m} Y_\ell^m(\theta, \varphi)$$

entwickeln. Die Koeffizienten $M_{\ell m}$ können dabei folgendermaßen bestimmt werden

$$M_{\ell m} = \int_0^{2\pi} \int_0^\pi \overline{Y_\ell^m}(\theta, \varphi) \mathbf{M} \cdot \mathbf{n}(\theta, \varphi) \sin \theta \, d\theta d\varphi.$$

Falls $\operatorname{div} \mathbf{M} = 0$, wie bei uns vorausgesetzt, folgt aus

$$\int_\Omega \operatorname{div} \mathbf{M} \, dx = \int_\Gamma \mathbf{M} \cdot \mathbf{n} \, ds_x = R^2 \int_0^{2\pi} \int_0^\pi \mathbf{M} \cdot \mathbf{n} \sin \theta \, d\theta d\varphi,$$

und $Y_0^0 = \frac{1}{\sqrt{4\pi}}$, dass $M_0^0 = 0$ gilt.

Um die Koeffizienten $M_{\ell m}$ zu bestimmen benötigt man im Allgemeinen die explizite Darstellung der Kugelflächenfunktionen. Eine kurze Auflistung dieser Funktionen (bis $\ell = 3$) ist zum Beispiel in JACKSON 2006 [13, Abschnitt 3.5] zu finden.

Wie bereits erwähnt, können wir aus den Randbedingungen die Koeffizienten unserer Lösungsansätze für Innenraumproblem (5.23) und Außenraumproblem (5.24) bestimmen. Zunächst setzen wir diese Ansätze in die erste Randbedingung ein (Stetigkeit des Potentials am Rand), was uns auf

$$\sum_{\ell=0}^{\infty} \sum_{m=-\ell}^{\ell} (\beta_{\ell m} R^{-\ell-1} - \alpha_{\ell m} R^\ell) Y_\ell^m(\theta, \varphi) = 0$$

führt. Ausnützen der Orthogonalitätsbeziehungen der Kugelflächenfunktionen aus Satz 5.3 liefert die Beziehung

$$\beta_{\ell m} = \alpha_{\ell m} R^{2\ell+1} \quad (5.25)$$

zwischen den Koeffizienten. Nun setzen wir in die Bedingung $[\gamma_1 u] = -\mathbf{M} \cdot \mathbf{n}$ ein

$$\sum_{\ell=0}^{\infty} \sum_{m=-\ell}^{\ell} (\beta_{\ell m}(-\ell-1)R^{-\ell-2} - \alpha_{\ell m} \ell R^{\ell-1}) Y_\ell^m(\theta, \varphi) = - \sum_{\ell=0}^{\infty} \sum_{m=-\ell}^{\ell} M_{\ell m} Y_\ell^m(\theta, \varphi).$$

Nützen wir wieder die Orthogonalitätsbeziehungen der Kugelflächenfunktionen aus, dann kommen wir auf

$$\beta_{\ell m}(-\ell-1)R^{-\ell-2} - \alpha_{\ell m} \ell R^{\ell-1} = -M_{\ell m}.$$

Mit (5.25) ergibt sich somit insgesamt

$$\alpha_{\ell m} = \frac{M_{\ell m}}{2\ell+1} R^{\ell-1}, \quad (5.26)$$

und

$$\beta_{\ell m} = \frac{M_{\ell m}}{2\ell+1} R^{\ell+2}. \quad (5.27)$$

Beispiel 1

Für den Radius der Kugel wählen wir $R = 1$. Die Magnetisierung sei gegeben durch

$$\mathbf{M} = \begin{pmatrix} M_x \\ M_y \\ M_z \end{pmatrix} \in \mathbb{R}^3,$$

und daher gilt mit dem Normalenvektor

$$\mathbf{n} = \begin{pmatrix} \cos \theta \sin \varphi \\ \sin \theta \sin \varphi \\ \cos \theta \end{pmatrix}$$

auf Γ ,

$$\mathbf{M} \cdot \mathbf{n} = M_x \sin \theta \cos \varphi + M_y \sin \theta \sin \varphi + M_z \cos \theta.$$

Mit Hilfe der Kugelflächenfunktionen können wir auch schreiben

$$\mathbf{M} \cdot \mathbf{n} = M_z \sqrt{\frac{4\pi}{3}} Y_1^0 - \frac{1}{2} \sqrt{\frac{3}{8\pi}} (M_x - iM_y) Y_1^1 - \frac{1}{2} \sqrt{\frac{3}{8\pi}} (M_x + iM_y) Y_1^{-1}.$$

Die Koeffizienten $\alpha_{\ell m}$ und $\beta_{\ell m}$ unserer Lösungsansätze (5.23) und (5.24) ergeben sich somit zu

$$\alpha_{\ell m} = \begin{cases} \frac{1}{3} \sqrt{\frac{4\pi}{3}} M_z & \text{für } \ell = 1, m = 0, \\ -\frac{1}{6} \sqrt{\frac{3}{8\pi}} (M_x - iM_y) & \text{für } \ell = 1, m = 1, \\ -\frac{1}{6} \sqrt{\frac{3}{8\pi}} (M_x + iM_y) & \text{für } \ell = 1, m = -1, \\ 0 & \text{sonst,} \end{cases}$$

und

$$\beta_{\ell m} = \begin{cases} \frac{1}{3} \sqrt{\frac{4\pi}{3}} M_z & \text{für } \ell = 1, m = 0, \\ -\frac{1}{6} \sqrt{\frac{3}{8\pi}} (M_x - iM_y) & \text{für } \ell = 1, m = 1, \\ -\frac{1}{6} \sqrt{\frac{3}{8\pi}} (M_x + iM_y) & \text{für } \ell = 1, m = -1, \\ 0 & \text{sonst.} \end{cases}$$

Daher lauten die Lösungen für den Innen- und Außenraum

$$u^{int}(r, \theta, \varphi) = \sum_{m=-1}^1 \alpha_{1m} r Y_1^m(\theta, \varphi),$$

$$u^{ext}(r, \theta, \varphi) = \sum_{m=-1}^1 \beta_{1m} r Y_1^m(\theta, \varphi).$$

Gilt nun speziell $M_x = 0$, $M_y = 0$ so vereinfachen sich die Lösungen zu

$$u^{int}(r, \theta, \varphi) = \frac{1}{3} M_z r \cos \theta,$$

$$u^{ext}(r, \theta, \varphi) = \frac{1}{3} M_z \frac{1}{r^2} \cos \theta.$$

Beispiel 2

Wir wählen nun eine Magnetisierung, deren Divergenz nicht verschwindet

$$\mathbf{M}(x, y, z) = \begin{pmatrix} x \\ y \\ z \end{pmatrix}.$$

Daher gilt $\operatorname{div} \mathbf{M} = 3$. Als Gebiet wählen wir die Einheitskugel. Nun gehen wir wie im ersten Abschnitt dieses Kapitels vor und suchen eine partikuläre Lösung des Problems $\Delta u = \operatorname{div} \mathbf{M}$, die wir mit einer homogenen Lösung überlagern. Es lässt sich schnell überprüfen, dass

$$u^{par} = \frac{x^2 + y^2 + z^2}{2}$$

eine Partikulärlösung darstellt. Für die homogene Lösung wählen wir den Ansatz (5.23). Mit den Kugelflächenfunktionen können wir auch $u^{par} = \frac{r^2}{2}\sqrt{4\pi}Y_0^0$ schreiben. Für $\mathbf{M} \cdot \mathbf{n}$ ergibt sich

$$\mathbf{M} \cdot \mathbf{n}(\theta, \varphi) = \begin{pmatrix} \sin \theta \cos \varphi \\ \sin \theta \sin \varphi \\ \cos \theta \end{pmatrix} \cdot \begin{pmatrix} \sin \theta \cos \varphi \\ \sin \theta \sin \varphi \\ \cos \theta \end{pmatrix} = 1 = \sqrt{4\pi}Y_0^0.$$

Somit ergeben sich für unsere Ansätze

$$u^{int}(r, \theta, \varphi) = \frac{r^2}{2}\sqrt{4\pi}Y_0^0 + \sum_{\ell=0}^{\infty} \sum_{m=-\ell}^{\ell} \alpha_{\ell m} r^{\ell} Y_{\ell}^m(\theta, \varphi),$$

bzw.

$$u^{ext}(r, \theta, \varphi) = \sum_{\ell=0}^{\infty} \sum_{m=-\ell}^{\ell} \beta_{\ell m} r^{-\ell-1} Y_{\ell}^m(\theta, \varphi).$$

Analog wie am Beginn dieses Abschnitts können wir nun die Koeffizienten $\alpha_{\ell m}$ und $\beta_{\ell m}$ aus den Übergangsbedingungen bestimmen. Dies führt auf

$$\begin{aligned} \alpha_{00} &= -\frac{\sqrt{4\pi}}{2}, \\ \beta_{00} &= 0, \\ \alpha_{\ell m} &= \beta_{\ell m} = 0 \quad \text{für } \ell \geq 1, |m| \leq \ell. \end{aligned}$$

Die Lösungen für Innen- und Außenraum lauten daher

$$\begin{aligned} u^{int}(r, \theta, \varphi) &= \frac{1}{2}(r^2 - 1), \\ u^{ext}(r, \theta, \varphi) &= 0. \end{aligned}$$

Beispiel 3

Wir wählen jetzt

$$\mathbf{M}(x, y, z) := \frac{\begin{pmatrix} x \\ y \\ z \end{pmatrix}}{\|(x, y, z)\|},$$

oder kurz $\mathbf{M}^i := e_r^i$. Für die Divergenz der Magnetisierung gilt

$$\operatorname{div} \mathbf{M} = \sum_{i=1}^3 \partial_i \mathbf{M}^i = \sum_{i=1}^3 \left(\frac{\partial_i x^i}{r} - \frac{x^i x^i}{r^2} \frac{1}{r} \right) = \frac{2}{r},$$

und kurzes Nachrechnen zeigt, dass $u^{par} = r$ eine partikuläre Lösung für $r \neq 0$ ist. Als Gebiet verwenden wir wiederum die Einheitskugel und es gilt

$$\mathbf{M} \cdot \mathbf{n} = \sum_{i=1}^3 e_r^i e_r^i = 1.$$

Die Ansätze für die Lösung im Innen- und Außenraum ergeben sich nun zu

$$u^{int}(r, \theta, \varphi) = r\sqrt{4\pi}Y_0^0 + \sum_{\ell=0}^{\infty} \sum_{m=-\ell}^{\ell} \alpha_{\ell m} r^{\ell} Y_{\ell}^m(\theta, \varphi),$$

$$u^{ext}(r, \theta, \varphi) = \sum_{\ell=0}^{\infty} \sum_{m=-\ell}^{\ell} \beta_{\ell m} r^{-\ell-1} Y_{\ell}^m(\theta, \varphi).$$

Einsetzen in die Übergangsbedingungen liefert für die Koeffizienten

$$\begin{aligned} \alpha_{00} &= -\sqrt{4\pi}, \\ \beta_{00} &= 0, \\ \alpha_{\ell m} &= \beta_{\ell m} = 0 \quad \text{für } \ell \geq 1, |m| \leq \ell. \end{aligned}$$

Somit ergeben sich die Lösungen

$$\begin{aligned} u^{int}(r, \theta, \varphi) &= r - 1 \\ u^{ext}(r, \theta, \varphi) &= 0. \end{aligned}$$

Kapitel 6

Implementierung in 2D

Im Folgenden werden die in Matlab erstellten Programme zur Berechnung des magnetostatischen Potentials mit den verschiedensten in Kapitel 4 vorgestellten Methoden im zweidimensionalen Fall beschrieben. Der Aufbau der Steifigkeitsmatrizen für die P1-FEM wurde von [9] übernommen. Die Matrizen, die zur Implementierung der Randintegraloperatoren nötig sind, wurden mit Hilfe von Funktionen aus dem Programmpaket HILBERT [2] berechnet.

6.1 Datenstrukturen

Bevor wir uns den Matlab Programmen im Speziellen widmen, treffen wir noch einige Konventionen bezüglich der Datenstrukturen.

Die Triangulierung $\mathcal{T}$ bzw. $\mathcal{T}_\Gamma$ von Ω bzw. Γ wird durch die drei Felder `coordinates`, `elements` und `boundary` beschrieben. Dabei ist `coordinates` ein $n_C \times 2$, `elements` ein $n_E \times 3$ und `boundary` ein $n_B \times 2$ Feld. Der Skalar n_C bezeichnet die Anzahl der Knoten der Triangulierung, n_E die Anzahl der Elemente und n_B die Anzahl der Randelemente.

Die Koordinaten der einzelnen Knoten werden im Array `coordinates` gespeichert,

```
coordinates(i,:) = [x y]
```

gibt an, dass der i -te Knoten die Koordinaten $(x, y) \in \mathbb{R}^2$ besitzt. Durch das Feld `coordinates` wird automatisch eine Nummerierung der Knoten durchgeführt. Wir legen fest, dass die ersten n_B Koordinaten den Knoten am Rand Γ entsprechen.

Ein Element wird durch seine drei Knoten $z_p, z_q, z_r \in \mathcal{M}$ beschrieben. Das i -te Element wird abgespeichert als

```
elements(i,:) = [p q r],
```

hierbei sind p, q, r die Indizes der Knoten im Feld `coordinates`. Weiters legen wir fest, dass die Knoten der Elemente im mathematisch positiven Sinn gespeichert sind. Dies hat den Vorteil, dass man den äußeren Normalenvektor auf den Rand eines Elements T mit einer geschlossenen Formel bestimmen kann. In Abbildung 6.1 wird dargestellt, in welcher Reihenfolge die Knoten eines Elements gespeichert werden.

Das Array `boundary` beschreibt die Triangulierung des Randes. Es beinhaltet die Indizes der Knoten $z_p, z_q \in \mathcal{M}_\Gamma$ eines Randelements $E_i \in \mathcal{T}_\Gamma$ bezüglich dem Feld `coordinates`,

```
boundary(i,:) = [p q].
```

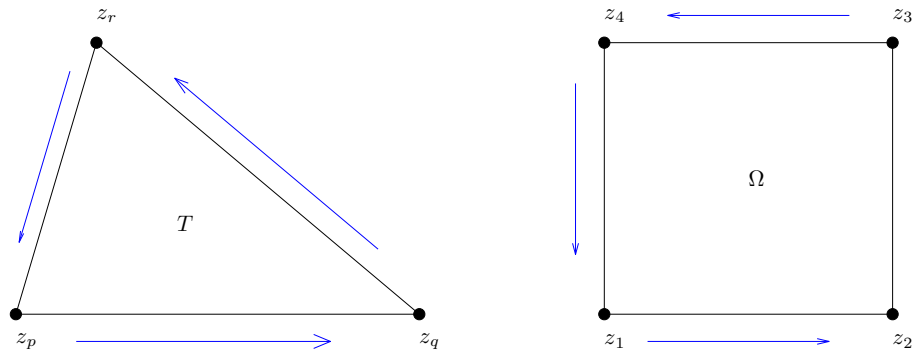


Abbildung 6.1: Die Knoten eines Elements T werden in mathematisch positiver Richtung abgespeichert. Die Knoten eines Randelements (vgl. rechtes Bild) werden so gespeichert, dass der Tangentialvektor in die mathematisch positive Richtung in Bezug auf den Rand zeigt.

Wir legen fest, dass die Knoten so abgespeichert sind, dass der Tangentialvektor vom ersten zum zweiten Knoten in mathematisch positiver Richtung in Bezug auf den Rand Γ zeigt. Dies wird auch in Abbildung 6.1 visualisiert.

Im Vektor `boundary2elements` steckt die Information, welche Randkante auf welchem Element liegt, d.h. `boundary2elements(i)=j` bedeutet dass die i -te Kante auf dem j -ten Element liegt.

Zusätzlich werden wir noch öfters das Feld `node2edges` benötigen, wobei

$$\text{node2edges}(i, j) = [k \ m]$$

angibt, dass der Knoten z_i der erste Knoten der k -ten Kante ist und der Knoten z_j ist der zweite Knoten der m -ten Kante. Die n -te Kante ist gegeben durch `boundary(n, :)= [p q]`. Der Knoten z_p wird als erster und z_q als zweiter Knoten der Kante bezeichnet.

6.2 García-Cervera & Roma

Dieser Abschnitt behandelt die Implementierung von Algorithmus 4.1 im Fall $d = 2$.

Listing 6.1: GCR2D.m

```

1 function [x x1 x2] = ...
2     GCR2D(coordinates,elements,boundary,boundary2elements,M,interpV)
3
4 *** Number of nodes, elements and boundary elements
5 nC = size(coordinates,1);
6 nE = size(elements,1);
7 nB = size(boundary,1);
8
9 *****
10 *** STEP 1, compute -laplace(u1) = -div(M)
11 ***           u1 = 0 on Gamma
12 *****
13
14 *** First vertex of elements and corresponding edge vectors

```

```

15 c1 = coordinates(elements(:,1),:);
16 d21 = coordinates(elements(:,2),:) - c1;
17 d31 = coordinates(elements(:,3),:) - c1;
18
19 %*** Vector of element areas 4*|T|
20 area4 = 2*(d21(:,1).*d31(:,2)-d21(:,2).*d31(:,1));
21
22 %*** Assembly of stiffness matrix
23 a = (sum(d21.*d31,2)./area4)';
24 b = (sum(d31.*d31,2)./area4)';
25 c = (sum(d21.*d21,2)./area4)';
26 A = [-2*a+b+c;a-b;a-c;a-b;b;-a;a-c;-a;c];
27 I = reshape(elements(:,[1 2 3 1 2 3 1 2 3]),9*nE,1);
28 J = reshape(elements(:,[1 1 1 2 2 2 3 3 3]),9*nE,1);
29 A = sparse(I,J,A(:));
30
31 %*** |T|*grad(Vj) where Vj is hat function on T
32 grad1 = [d21(:,2)-d31(:,2) d31(:,1)-d21(:,1)]/2;
33 grad2 = [d31(:,2) -d31(:,1)]/2;
34 grad3 = [-d21(:,2) d21(:,1)]/2;
35
36 %*** Compute M(sT)
37 MsT = feval(M, c1+(d21+d31)/3);
38
39 %*** Assembly of right-hand side
40 b = [sum(grad1.*MsT,2) sum(grad2.*MsT,2) sum(grad3.*MsT,2)];
41 b = accumarray(elements(:), b(:), [nC, 1]);
42
43 %*** Computation of P1-FEM approximation
44 boundaryNodes = 1:nB;
45 freenodes = nB+1:nC;
46 x1 = zeros(nC,1);
47 x1(freenodes) = A(freenodes,freenodes)\b(freenodes);
48
49 %*** Free memory
50 clear I J a b c
51
52 %*****
53 %*** STEP 2, compute Dirichlet data g_h = J_h V_h w_h
54 %*** w_h = M_h.n - nabla(U1).n
55 %*****
56
57 %*** Vector of element volumes 2*|T|
58 area2 = area4(boundary2elements)/2;
59
60 %*** Elementwise gradient of uh
61 u21 = repmat( (x1(elements(boundary2elements,2)))....
62 -x1(elements(boundary2elements,1)))./area2, 1,2);
63 u31 = repmat( (x1(elements(boundary2elements,3)))....
64 -x1(elements(boundary2elements,1)))./area2, 1,2);
65
66 %*** Compute grad(u1_h)
67 gradh = (d31(boundary2elements,:).*u21 ...
68 - d21(boundary2elements,:).*u31)*[0 -1 ; 1 0];
69
70 %*** Compute projection of M on P^0
71 cb1 = coordinates(boundary(:,1),:);
72 cb2 = coordinates(boundary(:,2),:);
73
74 M_h = feval(M, (cb1+cb2)./2);
75
76 %*** Outer normal vector
77 nVec = [cb2(:,2)-cb1(:,2), cb1(:,1)-cb2(:,1)];
78 edgeLength = sqrt(sum(nVec.^2,2));
79 nVec = nVec./repmat(edgeLength,1,2);
80

```

```

81  *** Compute g_h = J_h(V*w_h)
82  w_h = sum( (M_h - gradh).*nVec, 2);
83  g_h = feval(interpV, coordinates(1:nB,:), boundary, w_h);
84
85  %*****
86  *** STEP 3, compute -laplace(u2) = 0
87  ***                               u2 = g_h on Gamma
88  %*****
89
90  *** Prescribe values at boundary nodes
91  x2 = zeros(nC,1);
92  x2(boundaryNodes) = g_h;
93
94  *** Assembly of right-hand side
95  b = - A*x2;
96
97  *** Computation of P1-FEM approximation
98  x2(freenodes) = A(freenodes, freenodes)\b(freenodes);
99
100 %*****
101 *** STEP 4, obtain discrete solution u = u1 + u2
102 %*****
103 x = x1 + x2;

```

- Zeile 1-2: Die Funktion GCR2D erwartet als Eingabe die Triangulierung, welche durch `coordinates`, `elements` und `boundary` beschrieben wird, den Vektor `boundary2elements` und die Magnetisierung `M`. Mit `interpV` wird der Name der Funktion übergeben, mit der die Randdaten approximiert werden (siehe Schritt 2 in Algorithmus 4.1). Zurückgegeben wird die Gesamtlösung `x` und die Teillösungen von Schritt 1 bzw. Schritt 3, `x1` bzw. `x2`.
- Zeile 4-7: Bestimmung der Anzahl der Knoten `nC`, Elemente `nE` und Randelemente `nB`.
- Zeile 9-50: Schritt 1: Bestimme die Lösung der schwachen Formulierung: Finde $U_1 \in \mathcal{S}_0^1(\mathcal{T})$ mit

$$(\nabla U_1, \nabla V_j)_{L^2(\Omega)} = (\mathbf{M}, \nabla V_j)_{L^2(\Omega)} \quad \text{für } j = 1 \dots N. \quad (6.1)$$

$N = \#\mathcal{M}_F$ ist die Anzahl der Knoten im Inneren und $\{V_j\}_{j=1}^N$ bezeichnen die Basisfunktionen (Hutfunktionen) von $\mathcal{S}_0^1(\mathcal{T})$.

- Zeile 14-29: Aufbau der Steifigkeitsmatrix A wie in [9].
- Zeile 31-34: Berechnung der stückweise konstanten Gradienten ∇V_j der Basisfunktionen, gewichtet mit der Fläche eines Elements.
- Zeile 36-37: Berechnung von $\mathbf{M}(s_T)$, s_T bezeichnet dabei den Massenschwerpunkt eines Elements T .
- Zeile 39-41: Aufbau der rechten Seite von 6.1. Es gilt

$$(\mathbf{M}, \nabla V_j)_{L^2(\Omega)} = \sum_{T \in \mathcal{T}} (\mathbf{M}, \nabla V_j)_{L^2(T)},$$

und das Integral $(\mathbf{M}, \nabla V_j)_{L^2(T)}$ wird durch eine 1-Punkt Quadratur approximiert:

$$\int_T \mathbf{M} \cdot \nabla V_j dx = \nabla V_j \cdot \int_T \mathbf{M} dx \approx |T| \mathbf{M}(s_T) \cdot \nabla V_j.$$

Die Matlab Funktion `accumarray` dient zum schnellen Aufbau von dichten Matrizen.

- Zeile 43-47: Lösung des linearen Gleichungssystems $Ax = b$.
- Zeile 49-50: Freigabe des Speichers von nicht mehr benötigten Arrays.

- Zeile 52-83: Schritt 2: Berechnung von $g_h = A_h \mathcal{V} w_h \in \mathcal{S}^1(\mathcal{T}_\Gamma)$, wobei A_h ein Interpolationsoperator ist, der durch das Function Handle `interpV` spezifiziert wird. Als Operator können hierbei der Clément-Operator, der Scott-Zhang-Operator oder der nodale Interpolant gewählt werden. Der Ausdruck $w_h := \mathbf{M}_h \cdot \mathbf{n} - \nabla U_1 \cdot \mathbf{n}$ ist die L^2 Orthogonalprojektion von $\mathbf{M} \cdot \mathbf{n} - \nabla U_1 \cdot \mathbf{n}$ auf $\mathcal{P}^0(\mathcal{T}_\Gamma)$.

- Zeile 57-68: Berechnung von $\nabla U_1|_\Gamma$. Da U_1 stückweise affin ist, gilt daher, dass ∇U_1 stückweise konstant ist. Mit dem Vektor `boundary2elements` lassen sich die Werte von ∇U_1 am Rand bestimmen. Dazu sei bemerkt dass eine Kante am Rand nur auf einem Element liegen kann, was für Kanten im Inneren nicht der Fall ist.

- Zeile 70-74: Es wird die L^2 -Orthogonalprojektion von $\mathbf{M}$ auf $(\mathcal{P}^0(\mathcal{T}_\Gamma))^2$ approximiert:

$$\Pi_h^0(\mathbf{M})|_E = \frac{1}{|E|} \int_E \mathbf{M} \, dx \approx \frac{1}{|E|} |\mathbf{M}(s_E)| = \mathbf{M}(s_E).$$

Dabei ist s_E der Mittelpunkt der Kante E .

- Zeile 76-79: Bestimmung des Normalenvektors.
- Zeile 81-83: Aufbau von $g_h = A_h \mathcal{V} w_h \in \mathcal{S}^1(\mathcal{T})$. Da der Normalenvektor auf jeder Kante konstant ist, gilt $\Pi_h^0(\mathbf{M} \cdot \mathbf{n}) = \Pi_h^0(\mathbf{M}) \cdot \mathbf{n}$. Für A_h kann dann einer der weiter unten angeführten Interpolationsoperatoren verwendet werden.
- Zeile 85-98: Schritt 3: Bestimmung der Lösung der schwachen Formulierung. Finde $U_2 \in \mathcal{S}^1(\Omega)$ mit

$$\begin{aligned} (\nabla U_2, \nabla V_j)_{L^2(\Omega)} &= 0 \quad \text{für } j = 1 \dots N, \\ \gamma_0 U_2 &= g_h, \end{aligned}$$

dabei gibt $N = \#\mathcal{M}$ die Anzahl der Knoten an.

- Zeile 90-95: Aufbau der Randwerte g_h .
- Zeile 97-98: Lösung des linearen Gleichungssystems $Ax = b$.
- Zeile 100-103: Schritt 4: Es werden die Knotenwerte der Gesamtlösung $U = U_1 + U_2$ berechnet.

Wir betrachten jetzt noch die Interpolationsoperatoren, die in Schritt 2 in Listing 6.1 gewählt werden können.

Zunächst werden wir den Clément-Operator behandeln (siehe Definition 3.9).

Listing 6.2: `interpVCL.m`

```

1 function g_h = interpVCL(coordinates,boundary,w_h)
2
3 %*** Number of boundary elements
4 nB = size(boundary,1);
5
6 %*** Compute edge length
7 cb1 = coordinates(boundary(:,1),:);
8 cb2 = coordinates(boundary(:,2),:);
9 edgeLength = sqrt( sum( (cb1-cb2).^2,2) );
10
11 %*** Build V-Matrix V_ij = (\chi_i, V\chi_j)
12 V = buildV(coordinates,boundary);
13 w_h = V*w_h;
14
15 %*** Compute node2edges
16 node2edges = zeros(nB,2);
17 node2edges(boundary(:,1),1) = (1:nB)';
18 node2edges(boundary(:,2),2) = (1:nB)';
19
20 %*** Return J_h V w_h
21 g_h = sum(w_h(node2edges),2)./( sum(edgeLength(node2edges),2) );

```

- Zeile 1: Die Funktion `interpVCL` erwartet als Eingabe das $nB \times 2$ Feld `coordinates`, in welchem die Koordinaten der Randknoten gespeichert sind, das $nB \times 2$ Feld `boundary`, das die Randelemente (Kanten) beschreibt und `w_h` (siehe Schritt 2 in Listing 6.1). Zurückgegeben wird das Feld `g_h` welches die Knotenwerte von $g_h = J_h \mathcal{V} w_h$ enthält. Bei J_h handelt es sich um den Clément-Operator.
- Zeile 3-4: Bestimmung der Anzahl der Randelemente.
- Zeile 6-9: Berechnung der Kantenlängen.
- Zeile 11-12: Mit Hilfe der Funktion `buildV` von [2] wird eine Matrix $\mathbf{V} \in \mathbb{R}^{M \times M}$ aufgebaut, wobei wir mit M die Anzahl der Knoten in $\mathcal{T}_\Gamma$ bezeichnen. Die Einträge ergeben sich zu

$$\mathbf{V}_{ij} = (\mathcal{V} \chi_j, \chi_i)_{L^2(\Gamma)},$$

wobei die Funktionen χ_i eine Basis von $\mathcal{P}^0(\mathcal{T}_\Gamma)$ bilden mit

$$\chi_i = \begin{cases} 1 & \text{auf } E_i, \\ 0 & \text{sonst,} \end{cases}$$

und $\mathcal{V} : H^{-1/2}(\Gamma) \rightarrow H^{1/2}(\Gamma)$ der Einfachschichtpotentialoperator ist.

- Zeile 13: Berechnung von $(\mathcal{V} w_h, \chi_i)_{L^2(\Gamma)}$.
- Zeile 15-18: Berechnung von `node2edges`, vgl. Abschnitt 6.1.
- Zeile 20-21: Sei z ein Knoten, welcher auf den Kanten E_ℓ und E_m liegt und sei k ein Vektor mit den Einträgen $k_i = \Pi_h^0(\mathcal{V} w_h)|_{E_i}$. Dann folgt

$$v_z = \frac{k_\ell |E_\ell| + k_m |E_m|}{|E_\ell| + |E_m|},$$

hierbei ist v_z der in Definition 3.9 definierte Knotenwert des Clément-Operators $J_h = \sum_{z \in \mathcal{M}_\Gamma} v_z \eta_z$. Weiters sei auf die hier verwendete Bemerkung 3.6 verwiesen.

Als Nächstes behandeln wir die Implementierung des Scott-Zhang-Operators (siehe Definition 3.10).

Listing 6.3: `interpVSZ.m`

```

1 function g_h = interpVSZ(coordinates, boundary, w_h)
2
3 %*** Number of boundary elements
4 nB = size(boundary, 1);
5
6 %*** Coordinates of boundary nodes
7 cb1 = coordinates(boundary(:, 1), :);
8 cb2 = coordinates(boundary(:, 2), :);
9
10 %*** Determine for every boundary node the related boundary edges
11 node2edges = zeros(nB, 2);
12 node2edges(boundary(:, 1), 1) = (1:nB)';
13 node2edges(boundary(:, 2), 2) = (1:nB)';
14
15 %*** Choose edge for every boundary node
16 %*** Remaining code is independent of that choice!
17 edgesSZ = zeros(nB, 1);
18 edgesSZ = node2edges(:, 1);
19
20 %*** Coordinates of the endpoints of the chosen edges
21 cedgesSZ1 = cb1(edgesSZ, :);
22 cedgesSZ2 = cb2(edgesSZ, :);

```

```

23
24 %*** Choose quadrature nodes and weights on [-1 1]
25 %*** Remaining code is independent of that choice!
26 quadN = [-1 1]/sqrt(3);
27 quadW = [1; 1];
28 nN = length(quadN);
29
30 %*** Evaluation points
31 sx = reshape(cedgesSZ1, 2*nB, 1)*(1-quadN) ...
32     + reshape(cedgesSZ2, 2*nB, 1)*(1+quadN);
33 sx = 0.5*reshape(sx', nB*nN, 2);
34
35 %*** Compute V at evaluation points (V_q ... nB x nN -matrix)
36 V_q = evaluateV(coordinates, boundary, w_h, sx);
37 V_q = reshape(V_q, nN, nB)';
38
39 %*** Compute hat functions at quadrature nodes
40 sx1 = 0.5*(1-quadN);
41 sx2 = 0.5*(1+quadN);
42
43 %*** Decide whether a node is "left" or "right" endpoint of the chosen edge
44 idx = (node2edges(:, 1)==edgesSZ);
45
46 %*** Compute L^2 dual basis functions at quadrature nodes
47 P_q = zeros(nB, nN);
48 P_q(idx, :) = repmat(2*sx1-sx2, nnz(idx), 1);
49 P_q(~idx, :) = repmat(2*sx2-sx1, nnz(~idx), 1);
50
51 %*** Integral Approximation
52 g_h = (V_q.*P_q)*quadW;

```

Im Folgenden werden die Bezeichnungen und Definitionen aus Abschnitt 3.2 übernommen.

- Zeile 1: Die Funktion `interpVSZ` erwartet als Eingabe das $nB \times 2$ Feld `coordinates`, in welchem die Koordinaten der Randknoten gespeichert sind, das $nB \times 2$ Feld `boundary`, das die Randelemente (Kanten) beschreibt und `w_h` (siehe Schritt 2 in Listing 6.1). Zurückgegeben wird das Feld `g_h` welches die Knotenwerte von $g_h = S_h^\Gamma \mathcal{V}w_h$ enthält. Bei S_h^Γ handelt es sich um den in (3.11) definierten Operator.
- Zeile 3-4: Bestimmung der Anzahl der Randelemente.
- Zeile 6-8: Bestimmung der Koordinaten der Randknoten.
- Zeile 10-13: Berechnung von `node2edges`, vgl. Abschnitt 6.1.
- Zeile 15-18: Zu jedem Knoten $z_i \in \mathcal{M}_\Gamma$ wird eine Kante σ_i gewählt mit $z_i \in \sigma_i$. Der restliche Programmcode ist unabhängig von der Wahl der Kanten!
- Zeile 20-22: Bestimmung der Koordinaten der Eckpunkte der gewählten Kanten.
- Zeile 24-28: Wahl der Quadraturknoten q_j und -gewichte w_j auf dem Referenzintervall $[-1, 1]$. Es wird das Integral $\int_{-1}^1 f dx$ approximiert durch

$$\sum_j f(q_j)w_j.$$

Die Knoten werden im Feld `quadN` und die Gewichte in `quadW` gespeichert.

Der restliche Code ist unabhängig von der gewählten Quadraturregel. In dem hier abgebildeten wird eine 2-Punkt Gauß-Quadratur verwendet.

- Zeile 30-33: Transformation der Quadraturknoten vom Referenzintervall auf die Kante σ_i .
- Zeile 35-37: Werte $\mathcal{V}w_h$ an den transformierten Knoten aus.

- Zeile 39-41: Werte die Hutfunktionen auf dem Referenzintervall $[-1, 1]$ an den Quadraturknoten aus. Dies entspricht einer Auswertung der Hutfunktionen auf σ_i an den transformierten Quadraturknoten.
- Zeile 43-44: Bestimme ob z_i linker oder rechter Eckpunkt der gewählten Kante σ_i ist.
- Zeile 46-52: Berechne die Knotenwerte von $g_h = S_h^\Gamma \mathcal{V}w_h$. Für den i -ten Knotenwert ergibt sich

$$\int_{\sigma_i} \psi_i \mathcal{V}w_h dx \approx \frac{|\sigma_i|}{2} \sum_j w_j \psi_i(x_j) (\mathcal{V}w_h)(x_j),$$

wobei wir mit x_j die transformierten Quadraturknoten bezeichnen. Mit der Darstellung (3.27) der dualen Basis ergibt sich somit

$$\begin{aligned} \frac{|\sigma_i|}{2} \sum_j w_j \psi_i(x_j) (\mathcal{V}w_h)(x_j) &= \frac{|\sigma_i|}{2} \sum_j w_j \frac{1}{|\sigma_i|} (4\phi_1 - 2\phi_2)(x_j) (\mathcal{V}w_h)(x_j) \\ &= \sum_j w_j (2\phi_1 - \phi_2)(x_j) (\mathcal{V}w_h)(x_j), \end{aligned}$$

dabei ist nach Konvention $\phi_1(z_i) = 1$.

Als Interpolationsoperator kann man auch den nodalen Interpolanten wählen.

Listing 6.4: interpVNI.m

```
1 function g_h = interpVNI(coordinates,boundary,w_h)
2
3 %*** Return (V w_h)(coordinates)
4 g_h = evaluateV(coordinates,boundary,w_h,coordinates);
```

- Zeile 1: Die Funktion `interpVNI` erwartet als Eingabe das $nB \times 2$ Feld `coordinates`, in welchem die Koordinaten der Randknoten gespeichert sind, das $nB \times 2$ Feld `boundary`, das die Randelemente (Kanten) beschreibt und `w_h` (siehe Schritt 2 in Listing 6.1). Zurückgegeben wird das Feld `g_h` welches die Knotenwerte von $g_h = I_h \mathcal{V}w_h$ enthält. Bei I_h handelt es sich um den nodalen Interpolanten.
- Zeile 3-4: Mit Hilfe der Funktion `evaluateV` aus [2] wird $\mathcal{V}w_h$ an den Knoten am Rand ausgewertet.

6.3 Fredkin & Koehler

Wir betrachten die Implementierung von Algorithmus 4.2 im Falle $d = 2$.

Listing 6.5: FK2D.m

```
1 function [x x1 x2] = FK2D(coordinates,elements,boundary,M,interpK)
2
3 %*** Number of nodes, elements and boundary elements
4 nC = size(coordinates,1);
5 nE = size(elements,1);
6 nB = size(boundary,1);
7
8 %*****
9 %*** STEP 1, compute -laplace(u1) = -div(M)
10 %*** du1/dn = M.n on Gamma
11 %*****
12
13 %*** First vertex of elements and corresponding edge vectors
```

```

14 c1 = coordinates(elements(:,1),:);
15 d21 = coordinates(elements(:,2),:) - c1;
16 d31 = coordinates(elements(:,3),:) - c1;
17
18 %*** Vector of element areas 4*|T|
19 area4 = 2*(d21(:,1).*d31(:,2)-d21(:,2).*d31(:,1));
20
21 %*** Assembly of stiffness matrix
22 a = (sum(d21.*d31,2)./area4)';
23 b = (sum(d31.*d31,2)./area4)';
24 c = (sum(d21.*d21,2)./area4)';
25 A = [-2*a+b+c;a-b;a-c;a-b;b;-a;a-c;-a;c];
26 I = reshape(elements(:,[1 2 3 1 2 3 1 2 3]),9*nE,1);
27 J = reshape(elements(:,[1 1 1 2 2 2 3 3 3]),9*nE,1);
28 A = sparse(I,J,A(:));
29
30 %*** |T|*grad(Vj) where Vj is hat function on T
31 grad1 = [d21(:,2)-d31(:,2) d31(:,1)-d21(:,1)]/2;
32 grad2 = [d31(:,2) -d31(:,1)]/2;
33 grad3 = [-d21(:,2) d21(:,1)]/2;
34
35 %*** Compute M(sT)
36 MsT = feval(M, c1+(d21+d31)/3);
37
38 %*** Assembly of right-hand side
39 b = [sum(grad1.*MsT,2) sum(grad2.*MsT,2) sum(grad3.*MsT,2)];
40 b = accumarray(elements(:), b(:), [nC, 1]);
41
42 %*** Computation of P1-FEM approximation, Neumann problem
43 freenodes = 1:nC-1; % solve restricted system
44 x1 = zeros(nC,1);
45 x1(freenodes) = A(freenodes,freenodes)\b(freenodes);
46
47 %*** Free memory
48 clear I J a b c
49
50 %*****
51 %*** STEP 2, compute Dirichlet data g_h = J_h w_h
52 %*****
53 g_h = feval(interpK, coordinates(1:nB,:),boundary,x1(1:nB));
54
55 %*****
56 %*** STEP 3, compute -laplace(u2) = 0
57 %*** u2 = g_h on Gamma
58 %*****
59
60 %*** Prescribe values at boundary nodes
61 boundaryNodes = 1:nB;
62 x2 = zeros(nC,1);
63 x2(boundaryNodes) = g_h;
64
65 %*** Assembly of right-hand side
66 b = -A*x2;
67
68 %*** Computation of P1-FEM approximation
69 freenodes = nB+1:nC;
70 x2(freenodes) = A(freenodes,freenodes)\b(freenodes);
71
72 %*****
73 %*** STEP 4, obtain discrete solution u = u1 + u2
74 %*****
75 x = x1 + x2;

```

- Zeile 1: Die Funktion FK2D erwartet als Eingabe die Magnetisierung M und die Triangulierung, welche durch coordinates, elements und boundary beschrieben wird. Mit interpK wird der Name der

Funktion übergeben, mit der die Randdaten approximiert werden (siehe Schritt 2 in Algorithmus 4.2). Zurückgegeben wird die Gesamtlösung x und die Teillösungen von Schritt 1 bzw. Schritt 3, x_1 bzw. x_2 .

- Zeile 3-6: Bestimmung der Anzahl der Knoten n_C , Elemente n_E und Randelemente n_B .
- Zeile 8-48: Schritt 1: Bestimme Lösung von der schwachen Formulierung: Finde $U_1 \in \mathcal{S}_*^1(\mathcal{T})$, mit

$$(\nabla U_1, \nabla V)_{L^2(\Omega)} = (\mathbf{M}, \nabla V)_{L^2(\Omega)} \quad \text{für alle } V \in \mathcal{S}_*^1(\mathcal{T}). \quad (6.2)$$

- Zeile 13-28: Aufbau der Steifigkeitsmatrix A wie in [9].
- Zeile 30-33: Berechnung der stückweise konstanten Gradienten ∇V_j der Basisfunktionen, gewichtet mit der Fläche eines Elements.
- Zeile 35-36: Berechnung von $\mathbf{M}(s_T)$, s_T bezeichnet dabei den Massenschwerpunkt eines Elements T .
- Zeile 38-40: Aufbau der rechten Seite von 6.2. Es gilt

$$(\mathbf{M}, \nabla V_j)_{L^2(\Omega)} = \sum_{T \in \mathcal{T}} (\mathbf{M}, \nabla V_j)_{L^2(T)},$$

und das Integral $(\mathbf{M}, \nabla V_j)_{L^2(T)}$ wird durch eine 1-Punkt Quadratur approximiert:

$$\int_T \mathbf{M} \cdot \nabla V_j dx = \nabla V_j \cdot \int_T \mathbf{M} dx \approx |T| \mathbf{M}(s_T) \cdot \nabla V_j.$$

Die Matlab Funktion `accumarray` dient zum schnellen Aufbau von dichten Matrizen.

- Zeile 42-45: Lösung des linearen Gleichungssystems $Ax = b$. Da es sich um ein Neumann Problem handelt, löst man nicht das volle Gleichungssystem, sondern ein restringiertes (dies liegt daran, dass der Kern der Matrix eindimensional ist, daher löst man ein um die Dimension 1 verringertes System) und zieht dann das Integralmittel von der Lösung dieses restringierten Problems ab. Man reduziert das Gleichungssystem indem man eine zu einem beliebigen Knoten gehörende Zeile und Spalte aus der Steifigkeitsmatrix entfernt. In unserem Problem aber ist die Gesamtlösung $U = U_1 + U_2$ eindeutig, daher muss man nicht verlangen dass das Integralmittel von U_1 Null wird. Man beachte jedoch, dass die Konditionszahl der Steifigkeitsmatrix stark davon abhängig ist, welcher Freiheitsgrad, also Knoten, herausgenommen wird.
- Zeile 47-48: Freigabe des Speichers von nicht mehr benötigten Arrays.
- Zeile 50-53: Schritt 2: Berechnung von $g_h = A_h(\mathcal{K} - 1/2)U_1 \in \mathcal{S}^1(\mathcal{T}_\Gamma)$, wobei A_h ein Interpolationsoperator ist, der durch das Function Handle `interpK` spezifiziert wird. Als Operator können hierbei der Clément-Operator, der Scott-Zhang-Operator oder der nodale Interpolant gewählt werden. Verwenden wir den nodalen Interpolanten, so müssen die Innenwinkel an Eckpunkten berücksichtigt werden, d.h. es muss $g_h = A_h(\mathcal{K} - 1 + \sigma)U_1 \in \mathcal{S}^1(\mathcal{T}_\Gamma)$ bestimmt werden, wobei σ die in (2.23) definierte Größe ist. Die Interpolationsoperatoren werden weiter unten vorgestellt.
- Zeile 55-70: Schritt 3: Bestimmung der Lösung der schwachen Formulierung. Finde $U_2 \in \mathcal{S}^1(\Omega)$ mit

$$\begin{aligned} (\nabla U_2, \nabla V_j)_{L^2(\Omega)} &= 0 \quad \text{für } j = 1 \dots N, \\ \gamma_0 U_2 &= g_h, \end{aligned}$$

dabei gibt $N = \#\mathcal{M}$ die Anzahl der Knoten an.

- Zeile 60-66: Aufbau der Randwerte g_h .
- Zeile 68-70: Lösung des linearen Gleichungssystems $Ax = b$.
- Zeile 72-75: Schritt 4: Es wird die Gesamtlösung $U = U_1 + U_2$ berechnet.

Wir betrachten jetzt noch die Interpolationsoperatoren die in Schritt 2 in Listing 6.5 gewählt werden können. Zunächst werden wir den Clément-Operator behandeln (siehe Definition 3.9).

Listing 6.6: interpKCL.m

```

1 function g_h = interpKCL(coordinates, boundary, x1)
2
3 %*** Number of boundary elements
4 nB = size(boundary,1);
5
6 % L^2 projection of U1|_Gamma on P^0
7 x1_h = 1/2*( x1(boundary(:,1)) + x1(boundary(:,2)) );
8
9 %*** Compute edge length
10 cb1 = coordinates(boundary(:,1),:);
11 cb2 = coordinates(boundary(:,2),:);
12 edgeLength = sqrt( sum( (cb1-cb2).^2,2) );
13
14 %*** Build (K-1/2)
15 K = buildK(coordinates, boundary);
16 w_h = K*(x1(1:nB))./edgeLength-1/2*x1_h;
17
18 %*** Compute node2edges
19 node2edges = zeros(nB,2);
20 node2edges(boundary(:,1),1) = (1:nB)';
21 node2edges(boundary(:,2),2) = (1:nB)';
22
23 %*** Return J_h (K-1/2)x1
24 w_h = w_h.*edgeLength;
25 g_h = sum(w_h(node2edges),2)./( sum(edgeLength(node2edges),2) );

```

- Zeile 1: Die Funktion `interpKCL` erwartet als Eingabe das $nB \times 2$ Feld `coordinates`, in welchem die Koordinaten der Randknoten gespeichert sind, das $nB \times 2$ Feld `boundary`, das die Randelemente (Kanten) beschreibt und das Feld `x1`, welches die Werte von U_1 auf den Kanten beinhaltet. Zurückgegeben wird der Wert `g_h` welcher sich als $g_h = A_h(K - 1/2)U_1$ berechnet. Bei A_h handelt es sich um den Clément-Operator.
- Zeile 3-4: Bestimmung der Anzahl der Randelemente.
- Zeile 6-7: Berechnung der L^2 -Orthogonalprojektion von $U_1|_\Gamma$ auf $\mathcal{P}^0(\mathcal{T}_\Gamma)$. Sei E eine Kante dann gilt

$$(\Pi_h^0 U_1)|_E = \frac{1}{|E|} \int_E U_1 dx = \frac{1}{2}(U_1(z_1) + U_1(z_2)),$$

wobei z_1, z_2 die zwei Knoten der Kante E sind.

- Zeile 9-12: Berechnung der Kantenlängen.
- Zeile 14-15: Mit Hilfe der Funktion `buildK` von [2] wird eine Matrix $\mathbf{K} \in \mathbb{R}^{M \times M}$ aufgebaut, wobei wir mit M die Anzahl der Knoten in $\mathcal{T}_\Gamma$ bezeichnen. Die Einträge ergeben sich zu

$$\mathbf{K}_{ij} = (\mathcal{K}\phi_j, \chi_i)_{L^2(\Gamma)},$$

wobei die Funktionen χ_i eine Basis von $\mathcal{P}^0(\mathcal{T}_\Gamma)$ bilden, mit

$$\chi_i = \begin{cases} 1 & \text{auf } E_i, \\ 0 & \text{sonst,} \end{cases} \quad (6.3)$$

und die Funktionen ϕ_j bilden eine Basis von $\mathcal{S}^1(\mathcal{T}_\Gamma)$. $\mathcal{K} : H^{1/2}(\Gamma) \rightarrow H^{1/2}(\Gamma)$ ist der Doppelschichtpotentialoperator.

- Zeile 16: Berechnung von $\Pi_h^0((\mathcal{K} - 1/2)U_1)$.
- Zeile 18-21: Berechnung von `node2edges`, vgl. Abschnitt 6.1.
- Zeile 23-25: Sei z ein Knoten, welcher auf den Kanten E_ℓ und E_m liegt und sei k ein Vektor mit den Einträgen $k_i = \Pi_h^0((\mathcal{K} - 1/2)U_1)|_{E_i}$. Dann gilt

$$v_z = \frac{k_\ell |E_\ell| + k_m |E_m|}{|E_\ell| + |E_m|},$$

wobei v_z der in Definition 3.9 definierte Knotenwert des Clément-Operators $J_h = \sum_{z \in \mathcal{M}_\Gamma} v_z \eta_z$ ist. Weiters sei auf die hier verwendete Bemerkung 3.6 verwiesen.

Als Nächstes behandeln wir die Implementierung des Scott-Zhang-Operators (siehe Definition 3.10).

Listing 6.7: `interpKSZ.m`

```

1  function g_h = interpKSZ(coordinates,boundary,x1)
2
3  %*** Number of boundary elements
4  nB = size(boundary,1);
5
6  %*** coordinates of boundary nodes
7  cb1 = coordinates(boundary(:,1),:);
8  cb2 = coordinates(boundary(:,2),:);
9
10 %*** Determine for every boundary node the related boundary edges
11 node2edges = zeros(nB,2);
12 node2edges(boundary(:,1),1) = (1:nB)';
13 node2edges(boundary(:,2),2) = (1:nB)';
14
15 %*** Choose edge for every boundary node
16 %*** Remaining code is independent of that choice!
17 edgesSZ = zeros(nB,1);
18 edgesSZ = node2edges(:,1);
19
20 %*** Coordinates of the endpoints of the chosen edges
21 cedgesSZ1 = cb1(edgesSZ,:);
22 cedgesSZ2 = cb2(edgesSZ,:);
23
24 %*** Choose quadrature nodes and weights on [-1 1]
25 %*** Remaining code is independent of that choice!
26 quadN = [-1 1]/sqrt(3);
27 quadW = [1; 1];
28 nN = length(quadN);
29
30 %*** Evaluation points
31 sx = reshape(cedgesSZ1,2*nB,1)*(1-quadN) ...
32       + reshape(cedgesSZ2,2*nB,1)*(1+quadN);
33 sx = 0.5*reshape(sx',nB*nN,2);
34
35 %*** Compute K at evaluation points (K_q ... nB x nN -matrix)
36 K_q = evaluateK(coordinates,boundary,x1,sx);
37 K_q = reshape(K_q,nN,nB)';
38
39 %*** Compute hat functions at quadrature nodes
40 sx1 = 0.5*(1-quadN);
41 sx2 = 0.5*(1+quadN);
42
43 %*** Decide whether a node is "left" or "right" endpoint of the chosen edge
44 idx = (node2edges(:,1)==edgesSZ);
45
46 %*** Compute L^2 dual basis functions at quadrature nodes

```

```

47 P_q = zeros(nB,nN);
48 P_q(idx,:) = repmat(2*sx1-sx2,nnz(idx),1);
49 P_q(~idx,:) = repmat(2*sx2-sx1,nnz(~idx),1);
50
51 %*** Integral Approximation
52 g_h = (K_q.*P_q)*quadw;
53 g_h = g_h-0.5.*x1;

```

Im Folgenden werden die Bezeichnungen und Definitionen aus Abschnitt 3.2 übernommen.

- Zeile 1: Die Funktion `interpKSZ` erwartet als Eingabe das $nB \times 2$ Feld `coordinates`, in welchem die Koordinaten der Randknoten gespeichert sind, das $nB \times 2$ Feld `boundary`, das die Randelemente (Kanten) beschreibt und das Feld `x1`, welches die Werte von U_1 auf den Kanten beinhaltet. Zurückgegeben wird das Feld `g_h` welches die Knotenwerte von $g_h = S_h^\Gamma(\mathcal{K} - 1/2)U_1$ enthält. Bei S_h^Γ handelt es sich um den in (3.11) definierten Operator.
- Zeile 3-4: Bestimmung der Anzahl der Randelemente.
- Zeile 6-8: Bestimmung der Koordinaten der Randknoten.
- Zeile 10-13: Berechnung von `node2edges`, vgl. Abschnitt 6.1.
- Zeile 15-18: Zu jedem Knoten $z_i \in \mathcal{M}_\Gamma$ wird eine Kante σ_i gewählt mit $z_i \in \sigma_i$. Der restliche Programmcode ist unabhängig von der Wahl der Kanten.
- Zeile 20-22: Bestimmung der Koordinaten der Eckpunkte der gewählten Kanten.
- Zeile 24-28: Wahl der Quadraturknoten q_j und -gewichte w_j auf dem Referenzintervall $[-1, 1]$. Es wird das Integral $\int_{-1}^1 f dx$ approximiert durch

$$\sum_j f(q_j)w_j.$$

Die Knoten werden im Feld `quadN` und die Gewichte in `quadW` gespeichert.

Der restliche Code ist unabhängig von der gewählten Quadraturregel. In dem hier abgebildeten wird eine 2-Punkt Gauß-Quadratur verwendet.

- Zeile 30-33: Transformation der Quadraturknoten vom Referenzintervall auf die Kante σ_i .
- Zeile 35-37: Werte $\mathcal{K}U_1$ an den transformierten Knoten aus.
- Zeile 39-41: Werte die Hutfunktionen auf dem Referenzintervall $[-1, 1]$ an den Quadraturknoten aus. Dies entspricht einer Auswertung der Hutfunktionen auf σ_i an den transformierten Quadraturknoten.
- Zeile 43-44: Bestimme ob z_i linker oder rechter Eckpunkt der gewählten Kante σ_i ist.
- Zeile 46-53: Berechne die Knotenwerte von $g_h = S_h^\Gamma(\mathcal{K} - 1/2)U_1$. Für den i -ten Knotenwert ergibt sich

$$\int_{\sigma_i} \psi_i(\mathcal{K} - 1/2)U_1 dx \approx \frac{|\sigma_i|}{2} \sum_j w_j \psi_i(x_j)((\mathcal{K} - 1/2)U_1)(x_j),$$

wobei wir mit x_j die transformierten Quadraturknoten bezeichnen. Mit der Darstellung (3.27) der dualen Basis ergibt sich somit

$$\begin{aligned} \frac{|\sigma_i|}{2} \sum_j w_j \psi_i(x_j)(\mathcal{K}U_1)(x_j) &= \frac{|\sigma_i|}{2} \sum_j w_j \frac{1}{|\sigma_i|} (4\phi_1 - 2\phi_2)(x_j)(\mathcal{K}U_1)(x_j) \\ &= \sum_j w_j (2\phi_1 - \phi_2)(x_j)(\mathcal{K}U_1)(x_j), \end{aligned}$$

dabei ist nach Konvention $\phi_1(z_i) = 1$. Wir nutzen hier noch zusätzlich die Projektionseigenschaften des Operators S_h^Γ aus. Es gilt, da $U_1|_\Gamma \in \mathcal{S}^1(\mathcal{T}_\Gamma)$,

$$S_h^\Gamma U_1|_\Gamma = U_1|_\Gamma.$$

Als Interpolationsoperator kann man auch den nodalen Interpolanten wählen

Listing 6.8: interpKNI.m

```

1 function g_h = interpKNI(coordinates,boundary,x1)
2
3 %*** Number of boundary elements
4 nB = size(boundary,1);
5
6 %*** Compute edge length
7 cb1 = coordinates(boundary(:,1),:);
8 cb2 = coordinates(boundary(:,2),:);
9 edgeLength = sqrt( sum( (cb1-cb2).^2,2) );
10
11 %*** Compute node2edges
12 node2edges = zeros(nB,2);
13 node2edges(boundary(:,1),1) = (1:nB)';
14 node2edges(boundary(:,2),2) = (1:nB)';
15
16 %*** Direction vectors
17 t = (cb2-cb1)./repmat(edgeLength,1,2);
18 r1 = t(node2edges(:,2),:);
19 r2 = t(node2edges(:,1),:);
20
21 %*** Orientation, if det(x,y)>0 => positive rotational direction
22 orient = (r1(:,1).*r2(:,2)-r1(:,2).*r2(:,1)));
23
24 tmp = sum(r1.*r2,2);
25
26 %*** To avoid imaginary values
27 tmp(tmp>1) = 1;
28 tmp(tmp<-1) = -1;
29
30 alpha = zeros(nB,1);
31 alpha(orient<=0) = pi+acos( tmp(orient<=0) );
32 alpha(orient>0) = pi-acos( tmp(orient>0) );
33
34 %*** Nodal evaluation
35 g_h = evaluateK(coordinates,boundary,x1,coordinates);
36 g_h = g_h + (-1+alpha/(2*pi)).*x1;

```

- Zeile 1: Die Funktion `interpKNI` erwartet als Eingabe das $nB \times 2$ Feld `coordinates`, in welchem die Koordinaten der Randknoten gespeichert sind, das $nB \times 2$ Feld `boundary`, das die Randelemente (Kanten) beschreibt und das Feld `x1`, welches die Werte von U_1 auf den Kanten beinhaltet. Zurückgegeben wird der Wert `g_h` welcher die Knotenwerte von $g_h = I_h(\mathcal{K} - 1 + \alpha/(2\pi))U_1$ enthält. Bei I_h handelt es sich um den nodalen Interpolanten. Dabei ist $\alpha/(2\pi)$ die in Gleichung (2.23) definierte Größe σ .
- Zeile 3-4: Bestimmung der Anzahl der Randelemente.
- Zeile 6-9: Berechnung der Kantenlängen.
- Zeile 11-14: Berechnung von `node2edges`, vgl. Abschnitt 6.1.
- Zeile 16-19: Richtungsvektoren `r1` und `r2` der Kanten bestimmen. Jeder Knoten liegt an zwei Kanten. `r1(i, :)` ist dabei der Richtungsvektor der Kante, wo z_i der rechte Endpunkt ist und `r2` ist der Richtungsvektor der Kante, wo z_i linker Endpunkt ist.

- Zeile 21-32: Bestimmung des Winkels α zwischen zwei Kanten. Der Vektor `orient` = $\det(\mathbf{r1}, \mathbf{r2})$ beinhaltet die Information über die Drehrichtung zwischen den Vektoren $\mathbf{r1}$ und $\mathbf{r2}$. Seien $r_1 \in \mathbb{R}^2, r_2 \in \mathbb{R}^2$ zwei Vektoren mit $\|r_1\|_2 = \|r_2\|_2 = 1$ und sei $\beta \in [0, \pi]$ der Winkel zwischen diesen beiden, dann bedeutet $\det(r_1, r_2) > 0$, dass wenn man den Vektor r_1 um den Winkel β in mathematisch positiver Richtung dreht, man den Vektor r_2 erhält. Gilt $\det(r_1, r_2) \leq 0$, dann muss man den Vektor r_1 in mathematisch negativer Richtung um den Winkel β drehen um den Vektor r_2 zu erhalten. Der Winkel β lässt sich durch die Formel $\cos \beta = r_1 \cdot r_2$ bestimmen.
- Zeile 24-28: In dem Vektor `tmp` wird das Skalarprodukt der Richtungsvektoren gespeichert. Die Funktion `arccos` besitzt den Definitionsbereich $[-1, 1]$. Um bei der Berechnung des Winkels β imaginäre Werte zu vermeiden, werden alle Zahlenwerte des Skalarprodukts die größer als 1 sind auf 1 gesetzt und alle jene die kleiner als -1 sind auf -1 gesetzt.

Sei $z \in \mathcal{M}_\Gamma$ und E_1, E_2 die linke bzw. die rechte Kante des Knotens z . Mit r_1, r_2 bezeichnen wir die Richtungsvektoren der beiden Kanten. Dann gilt für den Winkel α zwischen den Kanten:

$$\alpha = \begin{cases} \pi - \arccos(r_1 \cdot r_2) & \text{falls } \det(r_1, r_2) > 0, \\ \pi + \arccos(r_1 \cdot r_2) & \text{sonst.} \end{cases} \quad (6.4)$$

- Zeile 20-32: In `alpha` werden die Winkel an jedem Knoten gespeichert. `alpha(i)` gibt dabei den Winkel zwischen den beiden im Knoten z_i anliegenden Kanten an.
- Zeile 34-36: Mit Hilfe der Funktion `evaluateK` aus [2] wird die punktweise Auswertung des Doppelschichtpotentials bestimmt und in Zeile 35 wird dann $(\mathcal{K} - 1 + \alpha/(2\pi))U_1$, mit den zuvor berechneten Winkel α bestimmt.

6.4 Symmetrische FEM/BEM-Kopplung

In diesem Abschnitt betrachten wir die numerische Implementierung von Algorithmus 4.3, also die Realisierung der symmetrischen FEM/BEM-Kopplung.

Listing 6.9: FEMBEMsymm.m

```

1 function x = FEMBEMsymm(coordinates,elements,boundary,M)
2
3 *** Number of nodes, elements and boundary elements
4 nC = size(coordinates,1);
5 nE = size(elements,1);
6 nB = size(boundary,1);
7
8 *** First boundary node
9 first_b_n = nC-nB+1;
10
11 *** Renumber boundary nodes
12 boundary_new = boundary-first_b_n+1;
13
14 *** Build K,V,W matrices
15 K = buildK(coordinates(first_b_n:end,:),boundary_new);
16 V = buildV(coordinates(first_b_n:end,:),boundary_new);
17 W = buildW(coordinates(first_b_n:end,:),boundary_new);
18
19 *** Length of boundary edges
20 cb1 = coordinates(boundary(:,1),:);
21 cb2 = coordinates(boundary(:,2),:);
22 edgeLength = sqrt(sum((cb2-cb1).^2,2));
23
24 *** Build Matrix L, L_ij = (\chi_i, \eta_j)_L^2(\Gamma)

```

```

25 I = reshape(repmat(1:nB,2,1),2*nB,1);
26 J = reshape(boundary_new', 2*nB,1);
27 L = reshape(repmat(0.5*edgeLength',2,1),2*nB,1);
28 L = sparse(I,J,L);
29
30 %*** First vertex of elements and corresponding edge vectors
31 c1 = coordinates(elements(:,1),:);
32 d21 = coordinates(elements(:,2),:) - c1;
33 d31 = coordinates(elements(:,3),:) - c1;
34
35 %*** Vector of element areas 4*|T|
36 area4 = 2*(d21(:,1).*d31(:,2)-d21(:,2).*d31(:,1)));
37
38 %*** Build matrix for the FEM part
39 I = reshape(elements(:, [1 2 3 1 2 3 1 2 3]))',9*nE,1);
40 J = reshape(elements(:, [1 1 1 2 2 2 3 3 3]))',9*nE,1);
41 a = (sum(d21.*d31,2)./area4)';
42 b = (sum(d31.*d31,2)./area4)';
43 c = (sum(d21.*d21,2)./area4)';
44 A = [-2*a+b+c;a-b;a-c;a-b;b;-a;a-c;-a;c];
45
46 %*** Assembly of stiffness matrix
47 I = [I; repmat(first_b_n:nC,1,2*nB)']; repmat(nC+1:nC+nB,1,2*nB)';
48 J = [J; reshape(repmat(first_b_n:nC,nB,1),nB*nB,1); ...
49      reshape(repmat(nC+1:nC+nB,nB,1),nB*nB,1); ...
50      reshape(repmat(first_b_n:nC,nB,1),nB*nB,1); ...
51      reshape(repmat(nC+1:nC+nB,nB,1),nB*nB,1)];
52 A = [A(:); W(:); reshape(K',nB*nB,1)-0.5*reshape(L',nB*nB,1); ...
53      -K(:)+0.5*L(:); V(:)];
54 A = sparse(I,J,A);
55
56 %*** Free memory
57 clear I J a c K V L W
58
59 %*** Assemble right hand side ***
60 %*** |T|*grad(Vj) where Vj is hat function on T
61 grad1 = [d21(:,2)-d31(:,2) d31(:,1)-d21(:,1)]/2;
62 grad2 = [d31(:,2) -d31(:,1)]/2;
63 grad3 = [-d21(:,2) d21(:,1)]/2;
64
65 %*** Compute M(sT)
66 MsT = feval(M, c1+(d21+d31)/3);
67
68 %*** Assembly of right-hand side
69 b = zeros(nC+nB,1);
70 bdata = [sum(grad1.*MsT,2) sum(grad2.*MsT,2) sum(grad3.*MsT,2)];
71 b(1:nC) = accumarray(elements(:), bdata(:), [nC, 1]);
72
73 %*** Computation of P1-FEM Approximation
74 x = A\b;

```

- Zeile 1: Die Funktion FEMBEMsymm erwartet als Eingabe die Triangulierung, welche durch die Felder `coordinates`, `elements` und `boundary` beschrieben wird, und die Magnetisierung `M`. Zurückgegeben wird das Feld `x`, welches die Einträge des Lösungsvektors $\mathbf{x} \in \mathbb{R}^{N+M}$ beinhaltet, wobei N die Anzahl der Knoten in $\mathcal{T}$ und M die Anzahl der Knoten in $\mathcal{T}_\Gamma$ ist. Es gilt dann für die Lösungen $U \in \mathcal{S}^1(\mathcal{T})$ und $\Psi \in \mathcal{P}^0(\mathcal{T}_\Gamma)$ von Algorithmus 4.3

$$U = \sum_{i=1}^N \mathbf{x}_i \eta_i, \quad \Psi = \sum_{i=N+1}^{N+M} \mathbf{x}_i \chi_{(i-N)}.$$

Wir setzen weiters noch voraus, dass die Randknoten den letzten Knoten im Array `coordinates` entsprechen.

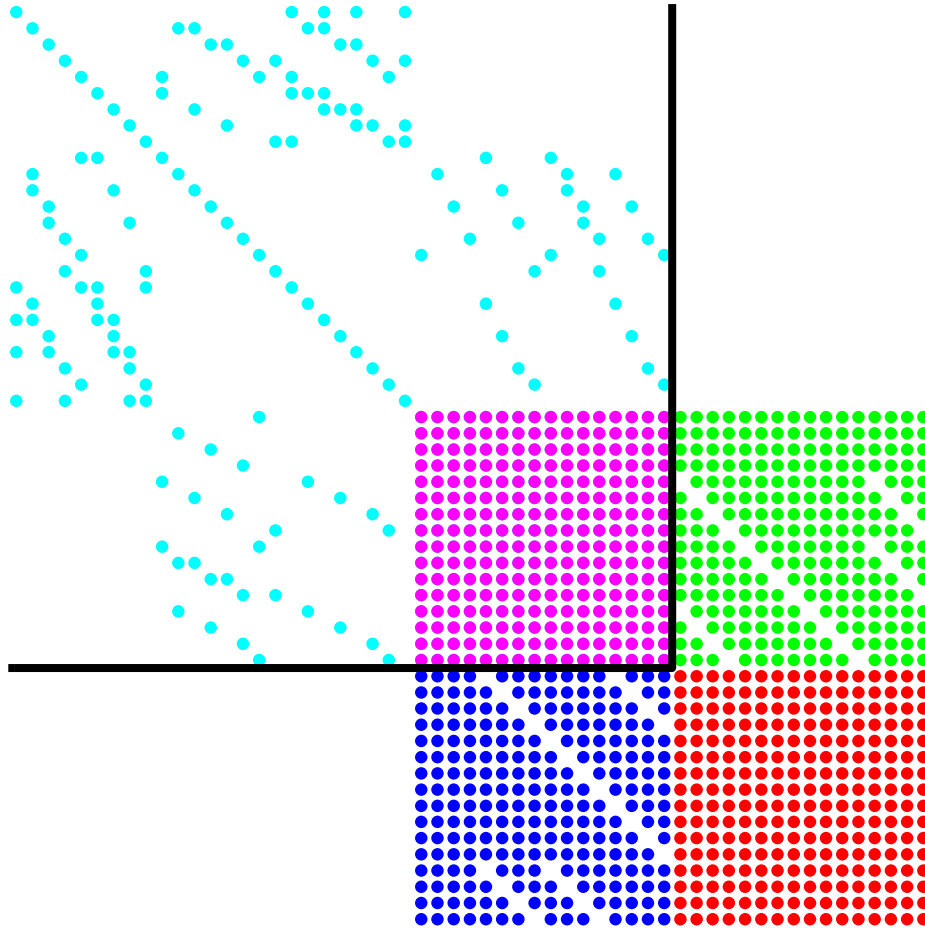


Abbildung 6.2: Besetzungsstruktur der Matrix $\mathbf{A}$ aus Abschnitt 6.4. Der rote Block entspricht den Einträgen der Einfachschichtpotentialmatrix $\mathbf{V}$, der rosa Block den Einträgen der Matrix $\mathbf{W}$. Der blaue Bereich visualisiert die Matrix $\frac{1}{2}\mathbf{L} - \mathbf{K}$, der grüne Bereich deren Adjungierte $(\frac{1}{2}\mathbf{L} - \mathbf{K})^T$. Die türkisen Markierungen entsprechen der dünn besetzten Matrix $\mathbf{B}$. Mit den schwarzen Linien wird die Grenze zwischen den Freiheitsgraden von $\mathcal{S}^1(\mathcal{T})$ und $\mathcal{P}^0(\mathcal{T}_\Gamma)$ dargestellt.

- Zeile 3-6: Bestimmung der Anzahl der Knoten nC , Elemente nE und Randelemente nB .
- Zeile 8-9: Bestimmung des ersten Randknotens im Feld `coordinates`.
- Zeile 11-12: Reindizierung der Randelemente.
- Zeile 14-17: Aufbau der diskreten Randintegraloperatoren. Dabei sind $\mathbf{K}, \mathbf{V}, \mathbf{W} \in \mathbb{R}^{M \times M}$ Matrizen mit den Einträgen

$$\begin{aligned}\mathbf{K}_{ij} &= (\mathcal{K}\eta_{j+N-M}, \chi_i)_{L^2(\Gamma)}, \\ \mathbf{V}_{ij} &= (\mathcal{V}\chi_j, \chi_i)_{L^2(\Gamma)}, \\ \mathbf{W}_{ij} &= (\mathcal{W}\eta_{j+N-M}, \eta_{i+N-M})_{L^2(\Gamma)}\end{aligned}$$

welche mit Hilfe der Programmbibliothek `HILBERT` aufgebaut werden. Nähere Informationen zum Aufbau dieser Matrizen ist in `HILBERT` 2009 [2] zu finden.

- Zeile 19-22: Berechnung der Kantenlängen.
- Zeile 24-28: Berechnung der Massenmatrix $\mathbf{L}$ mit Einträgen

$$\mathbf{L}_{ij} = (\eta_{j+N-M}, \chi_i)_{L^2(\Gamma)}.$$

- Zeile 30-54: Aufbau der Steifigkeitsmatrix $\mathbf{A} \in \mathbb{R}^{(N+M) \times (N+M)}$. Diese Matrix besitzt eine spezielle Struktur: Wir schreiben $\mathbf{A} = \mathbf{B} + \mathbf{C}$, wobei $\mathbf{B}, \mathbf{C} \in \mathbb{R}^{(N+M) \times (N+M)}$ Matrizen mit den Einträgen

$$\mathbf{B}_{ij} = \begin{cases} (\nabla\eta_j, \nabla\eta_i)_{L^2(\Omega)} & \text{für } (i, j) \in \{1, 2, \dots, N\}^2, \\ 0 & \text{sonst,} \end{cases}$$

und

$$\mathbf{C}_{ij} = \begin{cases} \mathbf{W}_{(i-N+M)(j-N+M)} & \text{für } (i, j) \in \{N-M+1, N-M+2, \dots, N\}^2, \\ \mathbf{K}_{(i-N+M)(j-N)}^T - \frac{1}{2}\mathbf{L}_{(i-N+M)(j-N)}^T & \text{für } N-M+1 \leq i \leq N, N+1 \leq j \leq N+M, \\ -(\mathbf{K}_{(i-N)(j-N+M)} - \frac{1}{2}\mathbf{L}_{(i-N)(j-N+M)}) & \text{für } N+1 \leq i \leq N+M, N-M+1 \leq j \leq N, \\ \mathbf{V}_{(i-N)(j-N)} & \text{für } (i, j) \in \{N+1, N+2, \dots, N+M\}^2, \\ 0 & \text{sonst,} \end{cases}$$

sind. Die Besetzungsstruktur der Matrix $\mathbf{A}$ wird in Abbildung 6.2 dargestellt.

- Zeile 56-57: Freigabe des von den Matrizen $\mathbf{K}, \mathbf{V}, \mathbf{W}, \mathbf{L}$ benutzten Speichers. Zusätzlich wird Speicher von temporär benutzten Variablen freigegeben.
- Zeile 59-63: Berechnung der stückweise konstanten Gradienten ∇V_j der Basisfunktionen, gewichtet mit der Fläche eines Elements.
- Zeile 65-66: Berechnung von $\mathbf{M}(s_T)$, s_T bezeichnet dabei den Massenschwerpunkt eines Elements T .
- Zeile 68-71: Aufbau des Vektors $\mathbf{b} \in \mathbb{R}^{N+M}$ der rechten Seite, mit den Einträgen

$$\mathbf{b}_i = \begin{cases} (\mathbf{M}, \nabla\eta_i)_{L^2(\Omega)} & \text{für } 1 \leq i \leq N, \\ 0 & \text{sonst.} \end{cases}$$

Die Werte $(\mathbf{M}, \nabla\eta_i)_{L^2(\Omega)}$ werden dabei wie in Listing 6.1 Zeile 39-41 aufgebaut.

- Zeile 73-74: Lösung des linearen Gleichungssystems $\mathbf{Ax} = \mathbf{b}$.

6.5 Direkte FEM/BEM-Kopplung

Wir betrachten die Implementierung von Algorithmus 4.4, also die Realisierung der direkten FEM/BEM-Kopplung.

Listing 6.10: FEMBEMJohNed.m

```

1  function x = FEMBEMJohNed(coordinates,elements,boundary,M)
2
3  %*** Number of nodes, elements and boundary elements
4  nC = size(coordinates,1);
5  nE = size(elements,1);
6  nB = size(boundary,1);
7
8  %*** First boundary node
9  first_b_n = nC-nB+1;
10
11 %*** Renumber boundary nodes
12 boundary_new = boundary-first_b_n+1;
13
14 %*** Build K,V,W matrices
15 K = buildK(coordinates(first_b_n:end,:),boundary_new);
16 V = buildV(coordinates(first_b_n:end,:),boundary_new);
17
18 %*** Length of boundary edges
19 cb1 = coordinates(boundary(:,1),:);
20 cb2 = coordinates(boundary(:,2),:);
21 edgeLength = sqrt(sum((cb2-cb1).^2,2));
22
23 %*** Build Matrix L, L_ij = (chi_i, eta_j)_L^2(\Gamma)
24 I = reshape(repmat(1:nB,2,1),2*nB,1);
25 J = reshape(boundary_new', 2*nB,1);
26 L = reshape(repmat(0.5*edgeLength',2,1),2*nB,1);
27 L = sparse(I,J,L);
28
29 %*** First vertex of elements and corresponding edge vectors
30 c1 = coordinates(elements(:,1),:);
31 d21 = coordinates(elements(:,2),:) - c1;
32 d31 = coordinates(elements(:,3),:) - c1;
33
34 %*** Vector of element areas 4*|T|
35 area4 = 2*(d21(:,1).*d31(:,2)-d21(:,2).*d31(:,1)));
36
37 %*** Build matrix for the FEM part
38 I = reshape(elements(:, [1 2 3 1 2 3 1 2 3])),9*nE,1);
39 J = reshape(elements(:, [1 1 1 2 2 2 3 3 3])),9*nE,1);
40 a = (sum(d21.*d31,2)./area4)';
41 b = (sum(d31.*d31,2)./area4)';
42 c = (sum(d21.*d21,2)./area4)';
43 A = [-2*a+b+c;a-b;a-c;a-b;b;-a;a-c;-a;c];
44
45 %*** Assembly of stiffness matrix
46 I = [I; repmat(first_b_n:nC,1,nB)'; repmat(nC+1:nC+nB,1,2*nB)'];
47 J = [J; reshape(repmat(nC+1:nC+nB,nB,1),nB*nB,1); ...
48      reshape(repmat(first_b_n:nC,nB,1),nB*nB,1); ...
49      reshape(repmat(nC+1:nC+nB,nB,1),nB*nB,1)];
50 A = [A(:); reshape(-L',nB*nB,1); -K(:)+0.5*L(:); V(:)];
51 A = sparse(I,J,A);
52
53 %*** Free memory
54 clear I J a c K V L
55
56 %*** |T|*grad(Vj) where Vj is hat function on T
57 grad1 = [d21(:,2)-d31(:,2) d31(:,1)-d21(:,1)]/2;

```

```

58 grad2 = [d31(:,2) -d31(:,1)]/2;
59 grad3 = [-d21(:,2) d21(:,1)]/2;
60
61 %*** Compute M(sT)
62 MsT = feval(M, c1+(d21+d31)/3);
63
64 %*** Assembly of right-hand side
65 b = zeros(nB+nC,1);
66 bdata = [sum(grad1.*MsT,2) sum(grad2.*MsT,2) sum(grad3.*MsT,2)];
67 b(1:nC) = accumarray(elements(:), bdata(:), [nC, 1]);
68
69 %*** Computation of P1-FEM Approximation
70 x = A\b;

```

- Zeile 1: Die Funktion FEMBEMJohNed erwartet als Eingabe die Triangulierung, welche durch die Felder `coordinates`, `elements` und `boundary` beschrieben wird, und die Magnetisierung `M`. Zurückgegeben wird das Feld `x`, welches die Einträge des Lösungsvektors $\mathbf{x} \in \mathbb{R}^{N+M}$ beinhaltet, wobei N die Anzahl der Knoten in $\mathcal{T}$ und M die Anzahl der Knoten in $\mathcal{T}_\Gamma$ ist. Es gilt dann für die Lösungen $U \in \mathcal{S}^1(\mathcal{T})$ und $\Psi \in \mathcal{P}^0(\mathcal{T}_\Gamma)$ von Algorithmus 4.4

$$U = \sum_{i=1}^N \mathbf{x}_i \eta_i, \quad \Psi = \sum_{i=N+1}^{N+M} \mathbf{x}_i \chi_{(i-N)}.$$

Wir setzen weiters noch voraus, dass die Randknoten den letzten Knoten im Array `coordinates` entsprechen.

- Zeile 3-6: Bestimmung der Anzahl der Knoten `nC`, Elemente `nE` und Randelemente `nB`.
- Zeile 8-9: Bestimmung des ersten Randknotens im Feld `coordinates`.
- Zeile 11-12: Reindizierung der Randelemente.
- Zeile 14-16: Aufbau der diskreten Randintegraloperatoren. Dabei sind $\mathbf{K}, \mathbf{V} \in \mathbb{R}^{M \times M}$ Matrizen mit den Einträgen

$$\begin{aligned} \mathbf{K}_{ij} &= (\mathcal{K} \eta_{j+N-M}, \chi_i)_{L^2(\Gamma)}, \\ \mathbf{V}_{ij} &= (\mathcal{V} \chi_j, \chi_i)_{L^2(\Gamma)} \end{aligned}$$

welche mit Hilfe der Programmbibliothek HILBERT aufgebaut werden. Nähere Informationen zum Aufbau dieser Matrizen ist in HILBERT 2009 [2] zu finden.

- Zeile 18-21: Berechnung der Kantenlängen.
- Zeile 23-27: Berechnung der Massenmatrix $\mathbf{L}$ mit den Einträgen

$$\mathbf{L}_{ij} = (\eta_{j+N-M}, \chi_i)_{L^2(\Gamma)}.$$

- Zeile 29-51: Aufbau der Steifigkeitsmatrix $\mathbf{A} \in \mathbb{R}^{(N+M) \times (N+M)}$. Diese Matrix besitzt eine spezielle Struktur: Wir schreiben $\mathbf{A} = \mathbf{B} + \mathbf{C}$, wobei $\mathbf{B}, \mathbf{C} \in \mathbb{R}^{(N+M) \times (N+M)}$ Matrizen mit den Einträgen

$$\mathbf{B}_{ij} = \begin{cases} (\nabla \eta_j, \nabla \eta_i)_{L^2(\Omega)} & \text{für } (i, j) \in \{1, 2, \dots, N\}^2, \\ 0 & \text{sonst,} \end{cases}$$

und

$$\mathbf{C}_{ij} = \begin{cases} -\mathbf{L}_{(i-N+M)(j-N)}^T & \text{für } N-M+1 \leq i \leq N, N+1 \leq j \leq N+M, \\ -(\mathbf{K}_{(i-N)(j-N+M)} - \frac{1}{2} \mathbf{L}_{(i-M)j}) & \text{für } N+1 \leq i \leq N+M, N-M+1 \leq j \leq N, \\ \mathbf{V}_{(i-N)(j-N)} & \text{für } (i, j) \in \{N+1, N+2, \dots, N+M\}^2, \\ 0 & \text{sonst,} \end{cases}$$

sind. Die Besetzungsstruktur der Matrix $\mathbf{A}$ wird in Abbildung 6.3 dargestellt.

- Zeile 53-54: Freigabe des von den Matrizen $\mathbf{K}$, $\mathbf{V}$, $\mathbf{L}$ benutzten Speichers. Zusätzlich wird Speicher von temporär benutzten Variablen freigegeben.
- Zeile 56-59: Berechnung der stückweise konstanten Gradienten ∇V_j der Basisfunktionen, gewichtet mit der Fläche eines Elements.
- Zeile 61-62: Berechnung von $\mathbf{M}(s_T)$, s_T bezeichnet dabei den Massenschwerpunkt eines Elements T .
- Zeile 64-67: Aufbau des Vektors $\mathbf{b} \in \mathbb{R}^{N+M}$ der rechten Seite, mit den Einträgen

$$\mathbf{b}_i = \begin{cases} (\mathbf{M}, \nabla \eta_i)_{L^2(\Omega)} & \text{für } 1 \leq i \leq N, \\ 0 & \text{sonst.} \end{cases}$$

Die Werte $(\mathbf{M}, \nabla \eta_i)_{L^2(\Omega)}$ werden dabei wie in Listing 6.1 Zeile 39-41 aufgebaut.

- Zeile 69-70: Lösung des linearen Gleichungssystems $\mathbf{Ax} = \mathbf{b}$.

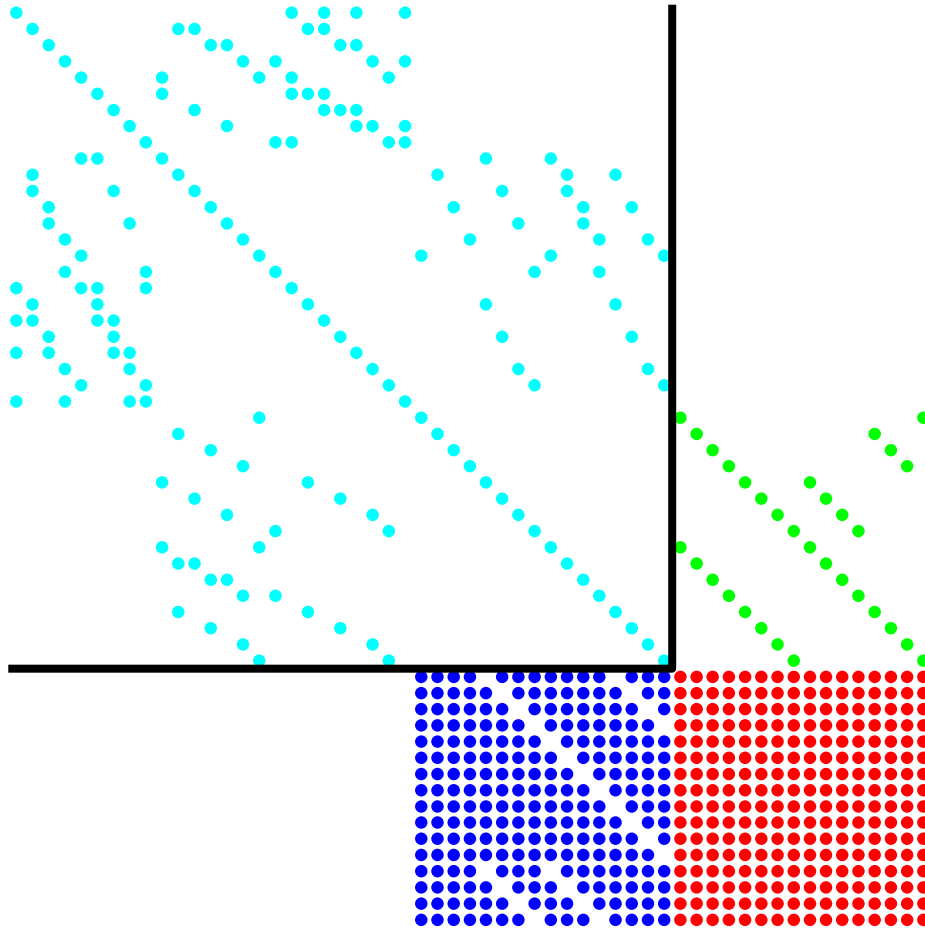


Abbildung 6.3: Besetzungsstruktur der Matrix $\mathbf{A}$ aus Abschnitt 6.5. Der rote Block entspricht den Einträgen der Einfachschichtpotentialmatrix $\mathbf{V}$. Der blaue Bereich visualisiert die Matrix $\frac{1}{2}\mathbf{L} - \mathbf{K}$. Die grünen Einträge entsprechen der dünn besetzten Matrix $(-\mathbf{L}^T)$ und die türkisen Markierungen entsprechen der dünn besetzten Matrix $\mathbf{B}$. Mit den schwarzen Linien wird die Grenze zwischen den Freiheitsgraden von $\mathcal{S}^1(\mathcal{T})$ und $\mathcal{P}^0(\mathcal{T}_\Gamma)$ dargestellt.

Kapitel 7

Numerische Resultate in 2D

In diesem Kapitel betrachten wir verschiedenste numerische Experimente in 2D. Die erhaltenen Resultate wurden allesamt auf einem 64-Bit Linux System mit 32 GB RAM und Matlab Version 2009b gerechnet. Wir befassen uns unter anderem mit den Konvergenzraten der in Kapitel 4 behandelten Verfahren. Zu Beginn bemerken wir noch, dass die schwarz strichlierten Linien in den hier angeführten Graphen als Orientierungshilfe dienen. Darüber hinaus sei noch darauf hingewiesen, dass die Größen in den Abbildungen, jeweils über der Anzahl der Elemente geplottet werden. Weiters bezeichnen wir mit

$$U_{MET} \in \mathcal{S}^1(\mathcal{T})$$

die numerisch berechnete Lösung des magnetostatischen Problems mit Hilfe des Verfahrens *MET*. Dabei kann *MET* folgende Werte annehmen:

- *GCR*, Verfahren von García-Cervera & Roma,
- *FK*, Verfahren von Fredkin & Koehler,
- *SYM*, Symmetrische FEM/BEM-Kopplung,
- *JN*, Johnson-Nédélec FEM/BEM-Kopplung.

Bei den hybriden Verfahren *GCR* und *FK* muss noch zusätzlich ein Interpolationsoperator gewählt werden (siehe Schritt 2 in Algorithmus 4.1 bzw. Algorithmus 4.2).

Im ersten Abschnitt untersuchen wir die Konvergenzraten auf der Kreisscheibe bei verschiedenen Magnetisierungen. Der zweite Abschnitt behandelt die Konvergenz der vorgestellten Verfahren und weitere Experimente auf unterschiedlichen Gebieten.

7.1 Kreisscheibe

Wie bereits in Kapitel 5 behandelt, können auf Kreisscheiben analytische Lösungen der magnetostatischen Gleichungen gefunden werden. In diesem Abschnitt werden wir verschiedene Magnetisierungen betrachten und mit den analytischen Lösungen vergleichen. Zunächst benötigen wir jedoch noch eine Routine die eine Triangulierung auf Kreisscheiben erzeugt. Im Zuge dieser Arbeit sind dazu zwei Matlab-Programme entwickelt worden, um dies zu bewerkstelligen. Zum einen wird mit Hilfe des Matlab-Befehls `de1aunay` und zum anderen wird mit NETGEN über die MEX Schnittstelle von Matlab ein Netz erzeugt. Das Programm, welches den Delaunay Algorithmus zur Generierung des Netzes verwendet lautet `genMeshC`. Die Routine `buildMeshIN2D` benützt NETGEN. Beide Funktionen sind im Anhang aufgelistet. In diesem Abschnitt werden wir meistens die zuletzt genannte Routine verwenden um eine Triangulierung zu erzeugen. Für das letzte Experiment benützen wir dann das Programm

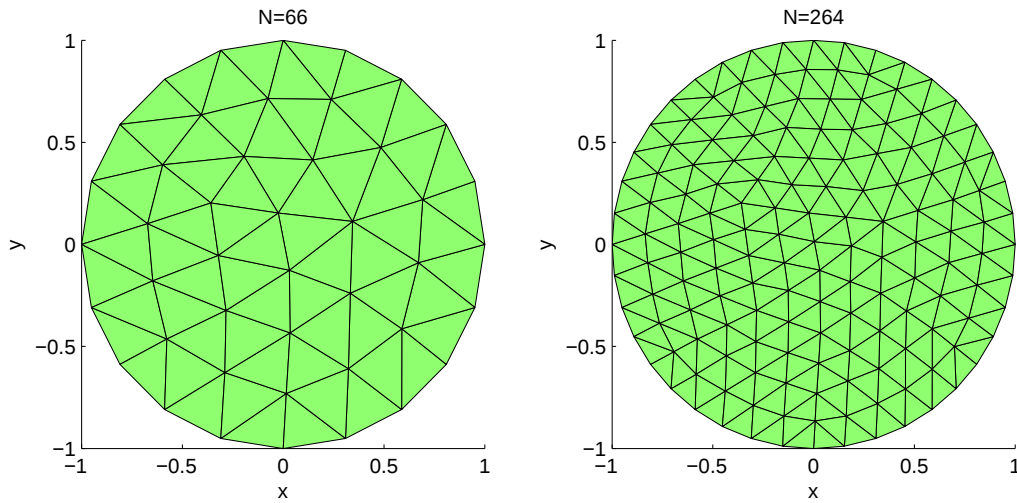


Abbildung 7.1: Das linke Bild zeigt die Starttriangulierung des Einheitskreises mit $N = 66$ Elementen. Wird dieses Netz uniform verfeinert, so ergibt sich die rechte Triangulierung mit $N = 264$ Elementen. Der Rand wird dabei feiner aufgelöst.

genMeshC, welches gewährleistet, dass der Punkt $(0, 0)$ ein Knoten des Netzes ist. Ein Beispiel für eine Triangulierung liefert etwa Abbildung 7.1. Die dort abgebildeten Netze wurden mit der Funktion **buildMeshIN2D** erstellt. Das zweite Netz geht dabei aus uniformer Verfeinerung des ersten Netzes hervor.

Im Folgenden werden wir Kreisscheiben mit Radius $R = 1$ betrachten. Dabei muss man beachten, dass der Operator $\mathcal{V}$ nur für $D := \text{diam}(\Omega) < 1$ elliptisch ist und damit eine eindeutige Lösung der beiden Kopplungsmethoden garantiert ist. Für $R = 1$ gilt $D = 2 > 1$ und somit ist die Elliptizität von $\mathcal{V}$ nicht gewährleistet. In den numerischen Experimenten zeigte sich jedoch, dass es keinen Unterschied macht, ob man Ω so skaliert, dass $D < 1$ gilt oder nicht.

Konvergenzraten bei verschiedenen Magnetisierungen

In diesem Abschnitt untersuchen wir die Konvergenzraten bei verschiedenen Magnetisierungen auf der Kreisscheibe. Weiters wollen wir kurz die Zeiten zum Lösen mit den verschiedenen Verfahren betrachten. Wir bemerken zuvor noch:

Bemerkung 7.1. Für die Bestimmung der Lösung mit Hilfe der Verfahren von García-Cervera & Roma und Fredkin & Koehler muss noch ein Interpolationsoperator gewählt werden, vgl. Algorithmus 4.1 und Algorithmus 4.2. In diesem Abschnitt wurde durchgehend der Scott-Zhang Operator verwendet, wobei mit dem nodalen Interpolanten analoge Ergebnisse erzielt werden. Mit dem Clément Operator erhält man für konstante Magnetisierung die gleichen Konvergenzraten wie bei den anderen beiden Interpolationsoperatoren. Bei nicht konstanten Magnetisierungen wird die Konvergenzgeschwindigkeit etwas schlechter, ist aber noch h und daher *a priori* optimal.

Im linken Bild von Abbildung 7.2 sind die Fehler der verschiedenen Verfahren im Vergleich zur analytischen Lösung angeführt. Die Magnetisierung wurde in diesem Beispiel konstant zu $\mathbf{M} = (1, 0)^T$ gewählt und daher gilt für die exakte Lösung nach Kapitel 5

$$u(x, y) = \frac{1}{2}x.$$

Die schwarz strichlierte Linie in dieser Abbildung ist proportional zu $h^{1.85}$. Bei uniformer Verfeinerung wird h

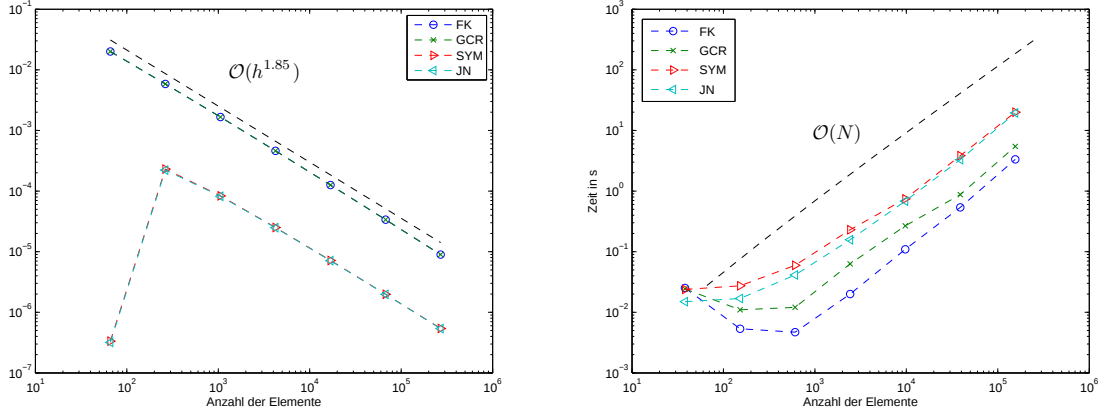


Abbildung 7.2: Das linke Bild zeigt den Fehler zwischen der mit verschiedenen numerischen Methoden berechneten Lösung und der exakten Lösung. Dabei wurde als Magnetisierung $\mathbf{M} = (1, 0)^T$ gewählt. Das rechte Bild vergleicht die Zeiten zum Lösen mit Hilfe der einzelnen Verfahren.

halbiert. Die Fehler

$$\|u - U_{MET}\|_{H^1(\Omega)}$$

der einzelnen Verfahren sind über die Anzahl der Elemente aufgetragen. Erstaunlicherweise fallen die Fehler in mit Ordnung $h^{1.85}$, obwohl man a priori nur die Ordnung h erwartet. Die absoluten Fehler der beiden Kopplungsmethoden liegen dabei deutlich unter den absoluten Fehlern der hybriden Methoden. Für $u \in \mathcal{S}^1(\mathcal{T})$ würde man auch erwarten, dass $\|u - U_{MET}\|_{H^1(\Omega)} = 0$ erfüllt ist, falls $MET = SYM$ bzw. $MET = JN$ gilt. Dies gilt aber nur dann, wenn man die genaue Geometrie nimmt und alle Daten exakt berechnet. Da wir den Kreis durch einen Polygonzug approximieren, stören wir die Daten.

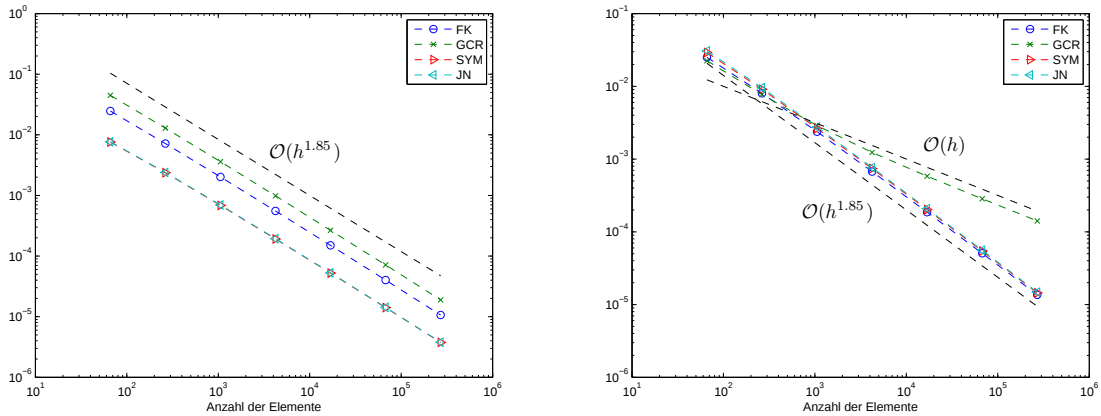


Abbildung 7.3: Fehler zwischen der mit verschiedenen numerischen Methoden berechneten Lösung und der exakten Lösung. Im linken Bild wurde $\mathbf{M} = (x, -y)^T$ und im rechten Bild $\mathbf{M} = (x, y)^T$ gewählt.

Das rechte Bild von Abbildung 7.2 zeigt die Zeiten die zum Lösen des magnetostatischen Problems mit den verschiedenen Ansätzen benötigt wurde. Dabei ist die schwarz strichlierte Linie proportional zu der Anzahl der Elemente. Es ist zu erkennen, dass alle Lösungsmethoden etwa die gleiche Steigung besitzen. Absolut gesehen

sind jedoch die hybriden Methoden schneller beim Lösen als die Kopplungsmethoden. Dies liegt unter anderem an dem Umstand, dass bei den hybriden Methoden nur ein diskretisierter Randintegraloperator zu berechnen ist, während bei den Kopplungsverfahren zwei bzw. drei diskretisierte Randintegraloperatoren aufgebaut werden. Ein weiterer Grund ist, dass bei den Kopplungsmethoden ein größeres Gleichungssystem gelöst werden muss.

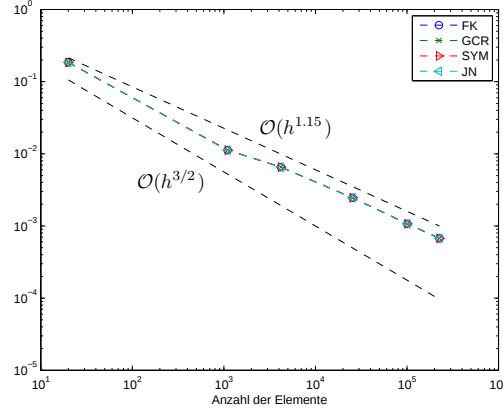


Abbildung 7.4: Fehler zwischen der mit verschiedenen numerischen Methoden berechneten Lösung und der exakten Lösung. Für die Magnetisierung gilt $\mathbf{M} = \mathbf{x}/\|\mathbf{x}\|$.

Wir wollen nun ein weiteres Beispiel untersuchen. Als Magnetisierung wählen wir hierbei

$$\mathbf{M} = \begin{pmatrix} x \\ -y \end{pmatrix},$$

und somit ergibt sich für die exakte Lösung

$$u(x, y) = \frac{1}{4}(x^2 - y^2).$$

Die Berechnung der $H^1(\Omega)$ -Norm wurde in dieser Arbeit nur für $\mathcal{S}^1(\mathcal{T})$ Funktionen implementiert. Zu beachten ist, dass $u(x, y) \notin \mathcal{S}^1(\mathcal{T})$ gilt. Wir können aber den Fehler

$$\|u - U_{MET}\|_{H^1(\Omega)}$$

durch

$$\|I_h u - U_{MET}\|_{H^1(\Omega)}$$

abschätzen, da aus der Dreiecksungleichung

$$\|u - U_{MET}\|_{H^1(\Omega)} \leq \|I_h u - U_{MET}\|_{H^1(\Omega)} + \|u - I_h u\|_{H^1(\Omega)}$$

folgt und nach dem Approximationssatz 3.8 erfüllt $u \in H^2(\Omega)$

$$\|u - I_h u\|_{H^1(\Omega)} = \mathcal{O}(h).$$

Die Werte

$$\|I_h u - U_{MET}\|_{H^1(\Omega)}$$

sind im linken Bild der Abbildung 7.3 über die Anzahl der Elemente aufgetragen. Dabei wird analog zum ersten Beispiel eine Konvergenzrate von h^2 festgestellt.

Im nächsten Beispiel wählen wir

$$\mathbf{M} = \begin{pmatrix} x \\ y \end{pmatrix}$$

und somit gilt für die exakte Lösung

$$u(x, y) = \frac{1}{2}(x^2 + y^2 - 1).$$

Die Werte

$$\|I_h u - U_{MET}\|_{H^1(\Omega)},$$

welche als Schätzung für die Fehler benutzt werden, sind im rechten Bild der Abbildung 7.3 über die Anzahl der Elemente aufgetragen. Dabei fallen die Fehler der Lösungen, welche mit den Kopplungsmethoden bzw. mit dem Ansatz von Fredkin & Koehler bestimmt wurden, mit der Ordnung $h^{1.85}$. Der Fehler der Lösung, welcher mit dem Ansatz von García-Cervera & Roma bestimmt wurde, fällt bis $N = 1.056$ mit Ordnung $h^{1.85}$, danach stellt sich eine Rate von $h^{1.15}$ ein.

In diesem Abschnitt wollen wir noch ein letztes Beispiel anführen. Dabei ist

$$\mathbf{M} = \frac{\mathbf{x}}{\|\mathbf{x}\|}$$

und für die analytische Lösung ergibt sich nach Kapitel 5

$$u(x, y) = \frac{1}{2}(\sqrt{x^2 + y^2} - 1).$$

Die Fehler der einzelnen Verfahren sind in Abbildung 7.4 illustriert. Man beachte, dass in diesem Fall das Netz nicht mit der eingangs erwähnten Funktion **buildMeshIN2D** erzeugt wurde (so wie in den vorher betrachteten Beispielen), sondern mit dem Befehl **genMeshC**. Weiters sei angeführt, dass $\mathbf{M}$ nicht am Punkt $(0, 0)$ ausgewertet werden kann. In Kapitel 6 wurde bereits beschrieben, dass $\mathbf{M}$ nur am Schwerpunkt eines Elements der Triangulierung ausgewertet wird. Mit dem Programm **genMeshC** werden Netze erzeugt, welche auch den Punkt $(0, 0)$ als Knoten enthalten und somit kann dieser Knoten nie ein Schwerpunkt eines Elements sein. Bei dem Programm **buildMeshIN2D** kann dies a priori nicht gewährleistet werden. Wie in Abbildung 7.4 erkennbar, stellt sich eine Konvergenzrate von h ein.

7.2 Weitere Gebiete

In diesem Abschnitt werden wir die Approximationslösungen auf verschiedenen Gebieten betrachten. Dabei verwenden wir durchgehend uniforme Verfeinerungen des Startnetzes.

Im Allgemeinen kann für eine beliebige Geometrie keine analytische Lösung bestimmt werden. Um dennoch den Fehler abschätzen zu können, sei Folgendes bemerkt.

Bemerkung 7.2. Nach Kapitel 4 wissen wir, dass die Kopplungsmethoden quasi-Bestapproximationen im Sinne des Céa Lemmas sind. Um eine Abschätzung für den Fehler einer numerisch berechneten Lösung zu bestimmen, gehen wir wie folgt vor. Sei $\{\mathcal{T}^{(\ell)}\}_{\ell=0}^n$ mit $n \in \mathbb{N}$ eine Folge von Triangulierungen, die durch uniforme Verfeinerungen des Startnetzes $\mathcal{T}^{(0)}$ entstehen.

- Bestimme die Lösung mit Hilfe einer Kopplungsmethode auf dem feinsten Netz $\mathcal{T}^{(n)}$, welche wir mit $U^{(n)}$ bezeichnen.
- Bestimme die Lösung mit Hilfe eines beliebigen Verfahrens auf einem gröberen Netz $\mathcal{T}^{(\ell)}$ ($0 \leq \ell \leq n-1$). Diese Lösung bezeichnen wir kurz mit $U^{(\ell)}$.

- Prolongiere diese Lösung auf das feinste Netz $\mathcal{T}^{(n)}$.
- Bestimme den Fehler zwischen $U^{(\ell)}$ und $U^{(n)}$ in der $H^1(\Omega)$ -Norm.

Wir wollen nun experimentell die Konvergenzraten der verschiedenen Verfahren bestimmen. Unter der Annahme, dass $\|U^{(n)} - U_{MET}^{(\ell)}\|_{H^1(\Omega)} = Ch^{\alpha_{MET}}$ mit einer Konstanten $C > 0$ gilt, kann man die Rate α_{MET} über

$$\log\|U^{(n)} - U_{MET}^{(\ell)}\|_{H^1(\Omega)} = \log C + \alpha_{MET} \log h_\ell$$

bestimmen. Mit $f_\ell := \|U^{(n)} - U_{MET}^{(\ell)}\|_{H^1(\Omega)}$ definieren wir

$$\alpha_{MET}^{(\ell)} := \frac{\log \frac{f_\ell}{f_{\ell-1}}}{\log \frac{h_\ell}{h_{\ell-1}}}$$

für $0 < \ell < n$.

Konvergenzraten auf dem Quadrat

Als erstes Gebiet betrachten wir ein Quadrat $\Omega_Q = (0, 0.25) \times (0, 0.25)$, welches in Abbildung 7.5 illustriert ist. Für den Durchmesser dieses Gebiets gilt die Bedingung $\text{diam } \Omega_Q < 1$, welche Elliptizität des Operators $\mathcal{V}$ impliziert.

N	$\alpha_{GCR,NI}$	$\alpha_{GCR,CL}$	$\alpha_{GCR,SZ}$	$\alpha_{FK,NI}$	$\alpha_{FK,CL}$	$\alpha_{FK,SZ}$	α_{SYM}	α_{JN}
16	0.79	0.79	0.50	0.51	0.79	0.79	0.50	0.51
64	0.85	0.85	0.76	0.73	0.85	0.85	0.76	0.73
256	0.88	0.88	0.83	0.83	0.88	0.88	0.83	0.83
1024	0.89	0.89	0.87	0.88	0.89	0.89	0.87	0.88
4096	0.91	0.91	0.90	0.90	0.91	0.91	0.90	0.90
16384	0.95	0.95	0.94	0.95	0.95	0.95	0.94	0.95
65536	1.08	1.08	1.08	1.08	1.08	1.08	1.08	1.08

Tabelle 7.1: Numerisch berechnete Konvergenzraten $\alpha_{MET}^{(\ell)}$ auf dem Gebiet Ω_Q mit konstanter Magnetisierung $\mathbf{M} = (1, 0)^T$. Der zusätzliche Index bei den Konvergenzraten $\alpha_{MET}^{(\ell)}$ der hybriden Verfahren gibt den jeweils verwendeten Interpolationsoperator an.

N	$\alpha_{GCR,NI}$	$\alpha_{GCR,CL}$	$\alpha_{GCR,SZ}$	$\alpha_{FK,NI}$	$\alpha_{FK,CL}$	$\alpha_{FK,SZ}$	α_{SYM}	α_{JN}
16	0.58	0.74	0.53	0.67	0.48	0.62	0.66	0.83
64	0.75	0.79	0.75	0.77	0.78	0.76	0.72	0.72
256	0.77	0.79	0.75	0.83	0.82	0.83	0.81	0.82
1024	0.77	0.77	0.75	0.87	0.84	0.86	0.86	0.87
4096	0.77	0.77	0.76	0.90	0.86	0.90	0.89	0.90
16384	0.79	0.78	0.78	0.95	0.89	0.94	0.94	0.95
65536	0.82	0.81	0.82	1.09	0.93	1.09	1.09	1.09

Tabelle 7.2: Numerisch berechnete Konvergenzraten $\alpha_{MET}^{(\ell)}$ auf dem Gebiet Ω_Q mit Magnetisierung $\mathbf{M} = \mathbf{x}/\|\mathbf{x}\|$. Der zusätzliche Index bei den Konvergenzraten $\alpha_{MET}^{(\ell)}$ der hybriden Verfahren gibt den jeweils verwendeten Interpolationsoperator an.

In Abbildung 7.8 sind die Fehler der zu vergleichenden Verfahren bei verschiedenen Magnetisierungen geplottet. Für das linke Bild gilt $\mathbf{M} = (1, 0)^T$, während für das rechte Bild $\mathbf{M} = \mathbf{x}/\|\mathbf{x}\|$ als Magnetisierung gewählt wurde.

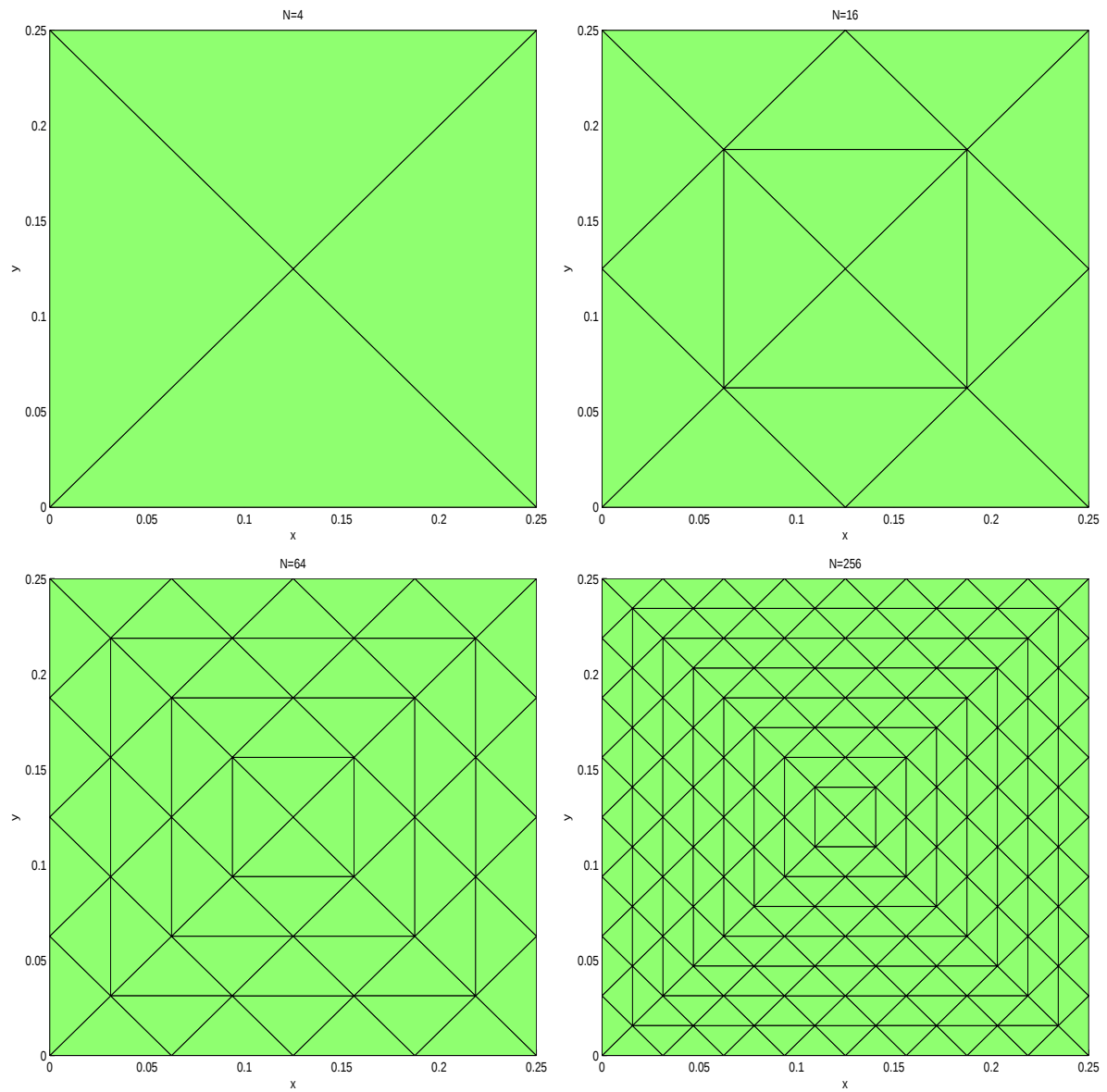


Abbildung 7.5: Startnetz und uniform verfeinerte Netze. N gibt die Anzahl der Elemente der jeweiligen Triangulierung an.

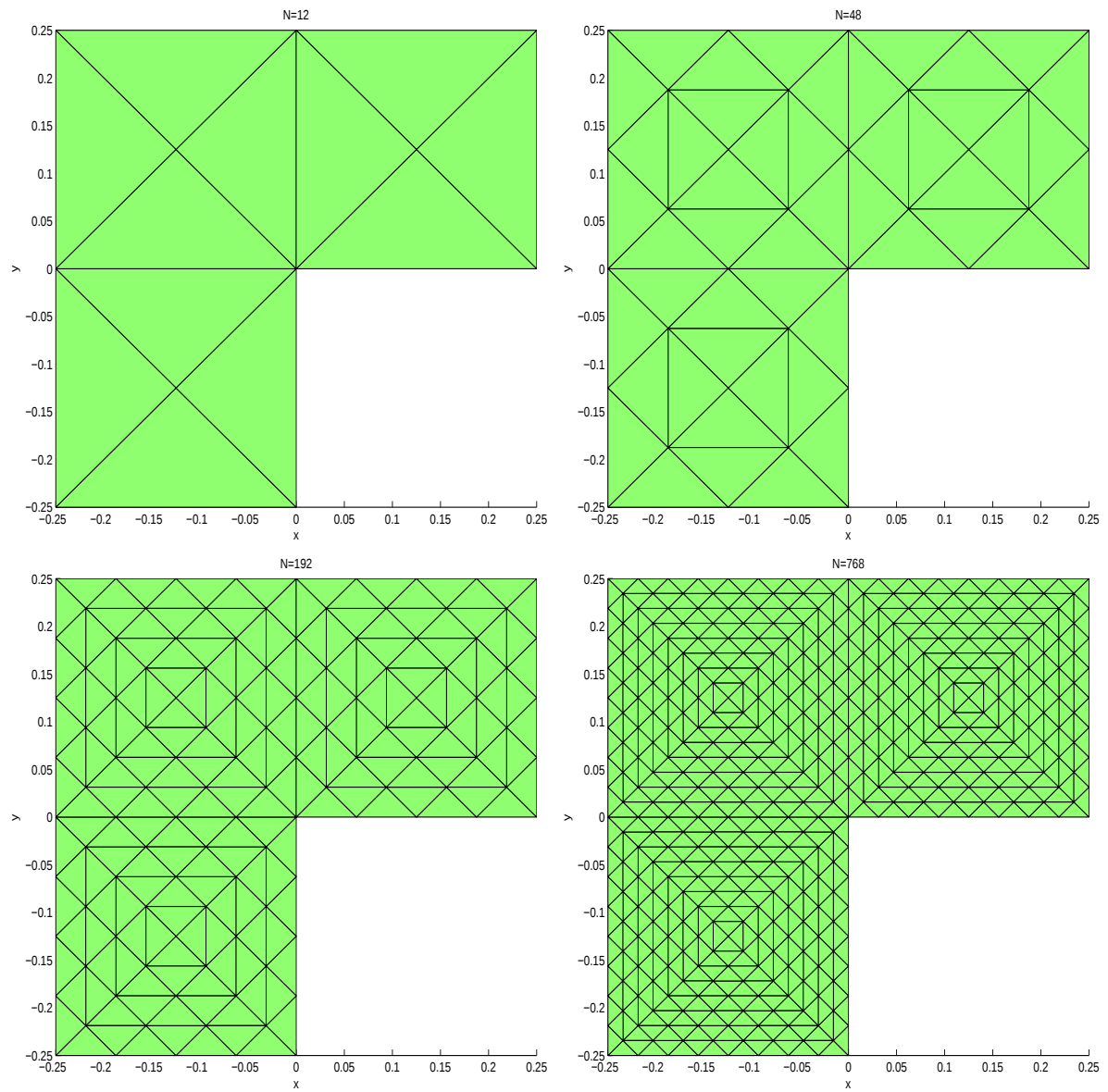


Abbildung 7.6: Startnetz und uniform verfeinerte Netze. N gibt die Anzahl der Elemente der jeweiligen Triangulierung an.

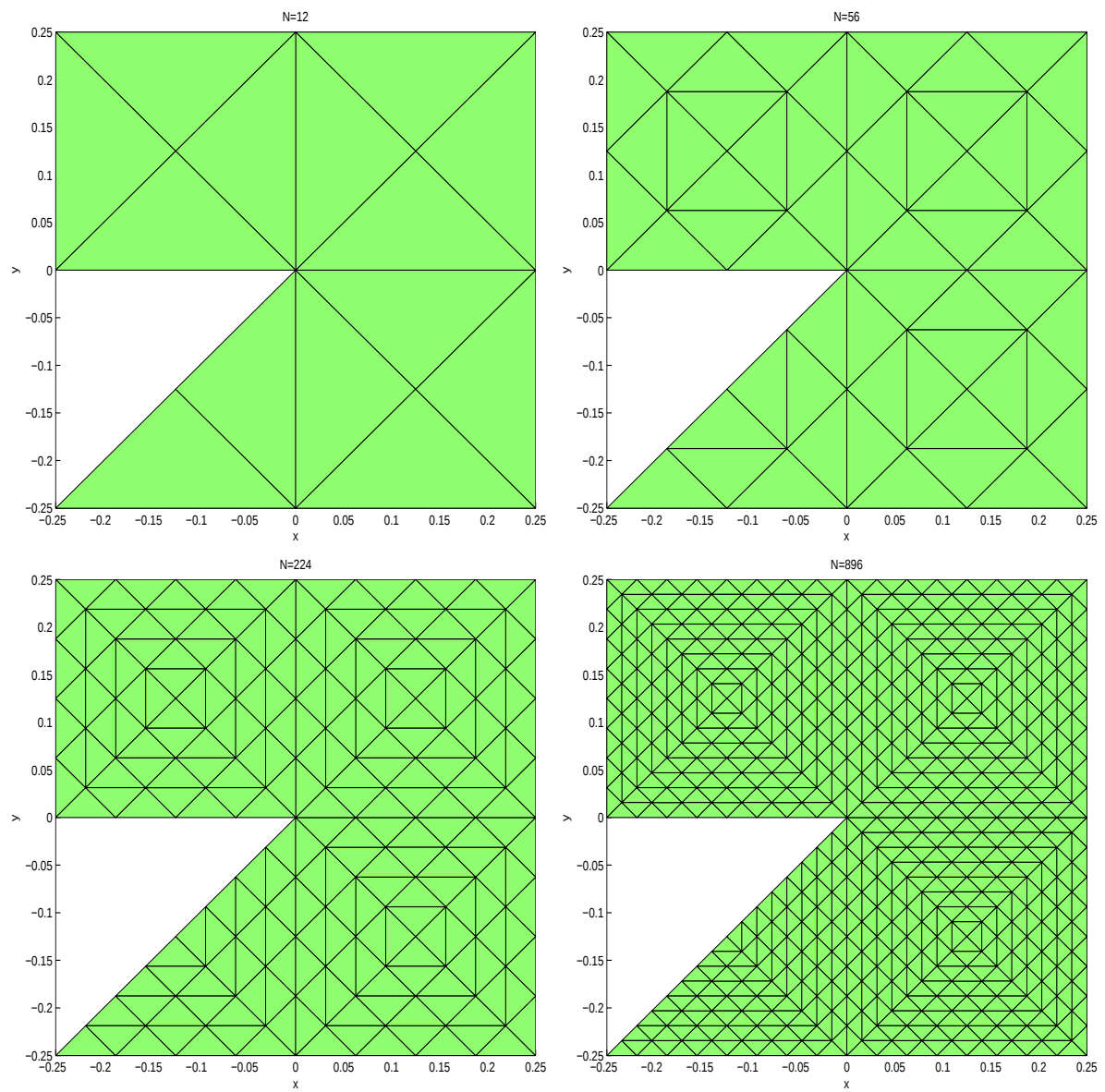
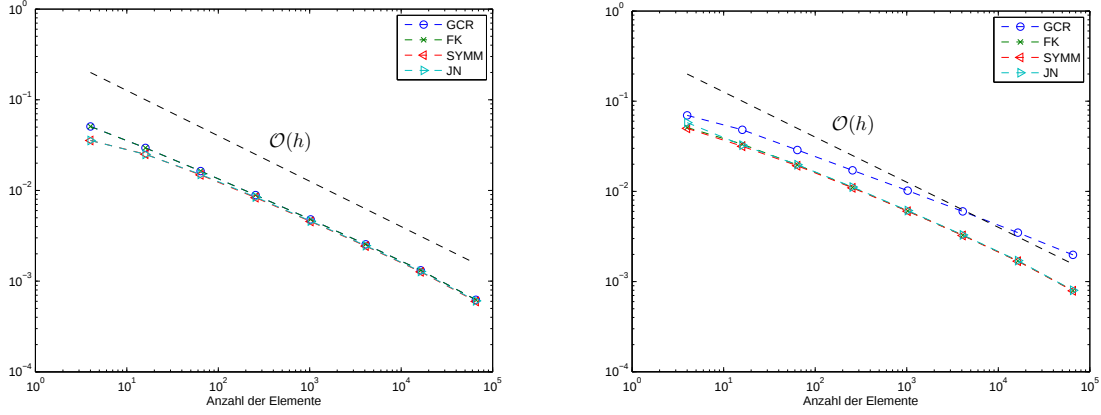


Abbildung 7.7: Startnetz und uniform verfeinerte Netze. N gibt die Anzahl der Elemente der jeweiligen Triangulierung an.



Abbildungung 7.8: Fehler zwischen der mit verschiedenen numerischen Methoden berechneten Lösung und der Lösung der symmetrischen FEM/BEM-Kopplung am feinsten Netz auf dem Gebiet Ω_Q . Für die Magnetisierung im linken Bild gilt $\mathbf{M} = (1, 0)^T$, für die im rechten $\mathbf{M} = \mathbf{x}/\|\mathbf{x}\|$. Als Interpolationsoperator wurde bei den hybriden Verfahren der nodale Interpolant I_h benützt.

Als Interpolationsoperator bei den hybriden Verfahren wurde der nodale Interpolant verwendet. In Tabelle 7.1 und Tabelle 7.2 sind die experimentell ermittelten Konvergenzraten $\alpha_{MET}^{(\ell)}$ eingetragen. Dabei gibt der zusätzliche Index bei den hybriden Verfahren den jeweils verwendeten Interpolationsoperator an. Der nodale Interpolant wird hierbei mit NI , der Scott-Zhang Operator mit SZ und der Clément Operator mit CL abgekürzt. Bei der konstanten Magnetisierung $\mathbf{M} = (1, 0)^T$ stellt sich bei jedem Verfahren experimentell die Konvergenzrate h ein. Wählt man jedoch $\mathbf{M} = \mathbf{x}/\|\mathbf{x}\|$, so ist bei der Lösung, welche mit dem Ansatz von García-Cervera & Roma bestimmt wurde, nur mehr die Ordnung 0.8 zu erkennen und zwar unabhängig vom gewählten Interpolationsoperator. Dies liegt an dem Umstand, dass bereits die Teillösung U_1 aus dem Ansatz von García-Cervera & Roma nur mehr mit der Geschwindigkeit $h^{0.8}$ konvergiert. Wir werden diese Situation im nächsten Unterabschnitt an einem speziell gewählten Beispiel näher untersuchen. Auch beim Fredkin & Koehler Ansatz ist eine schlechtere Ordnung zu erkennen, wenn man den Clément Operator verwendet. In der Analysis braucht man verschwindende Momente für Ordnung $\mathcal{O}(h)$. Der Clément Operator erfüllt diese Eigenschaft jedoch nicht.

Konvergenzraten auf Gebieten mit einspringender Ecke

Gebiet	M	$\alpha_{GCR,NI}$	$\alpha_{GCR,CL}$	$\alpha_{GCR,SZ}$	$\alpha_{FK,NI}$	$\alpha_{FK,CL}$	$\alpha_{FK,SZ}$	α_{SYM}	α_{JN}
L-shape	const.	1.08	0.96	1.08	1.08	0.94	1.08	1.09	1.08
	$x/\ x\ $	0.62	0.63	0.62	1.09	0.96	1.09	1.09	1.09
	jump	0.64	0.68	0.64	1.08	0.92	1.04	1.08	1.07
Z-shape	const.	1.09	0.94	1.08	1.09	0.94	1.08	1.09	1.09
	$x/\ x\ $	0.51	0.43	0.51	1.09	0.98	1.09	1.09	1.09
	jump	0.51	0.43	0.50	1.08	0.76	1.04	1.09	1.08

Tabelle 7.3: Experimentell bestimmte Konvergenzraten auf dem L-shape Ω_L und Z-shape Ω_Z bei verschiedenen Magnetisierungen. Der zusätzliche Index bei den Konvergenzraten α_{MET} der hybriden Verfahren gibt dabei den verwendeten Interpolationsoperator an.

Als nächstes beschäftigen wir uns mit dem in Abbildung 7.6 geplotteten L-förmigen Gebiet, welches sowohl in der Literatur als auch hier häufig als L-shape bezeichnet wird. Es gilt $\Omega_L = (-0.25, 0) \times (-0.25, 0.25) \cup$

$[0, 0.25) \times (0, 0.25)$ und $\text{diam } \Omega_L < 1$, weshalb Elliptizität des Operators $\mathcal{V}$ gewährleistet ist. Darüber hinaus werden wir auch Konvergenzraten auf dem in Abbildung 7.7 geplotteten Z-förmigen Gebiet bestimmen. Dieses Gebiet $\Omega_Z = (-0.25, 0.25) \times (-0.25, 0.25) \setminus \{(x, y) \in [-0.25, 0] \times [-0.25, 0] \mid y \geq x\}$ nennen wir im Folgenden auch Z-shape. Das Gebiet Ω_Z erfüllt $\text{diam } \Omega_Z < 1$.

Wir werden die experimentell ermittelten Konvergenzraten in Tabelle 7.3 für das L-shape und das Z-shape zusammenfassen. Dabei werden wir drei verschiedene Magnetisierungen betrachten. Die konstante Magnetisierung $\mathbf{M} = (1, 0)^T$ wird in der Tabelle mit “const.” abgekürzt. Die Magnetisierung

$$\mathbf{M}(x, y) = \begin{cases} (1, 0)^T & x > 0, \\ (0, 1)^T & \text{sonst} \end{cases}$$

werden wir im Folgenden mit “jump” bezeichnen. Als dritte Magnetisierung werden wir $\mathbf{M} = \mathbf{x}/\|\mathbf{x}\|$ benützen. Um die Konvergenzrate zu bestimmen, wählen wir wiederum die Lösung der symmetrischen FEM/BEM-Kopplung am feinsten Gitter. In Tabelle 7.3 ist zu erkennen, dass die beiden Kopplungsmethoden und der Ansatz von Fredkin & Koehler, mit nodalen Interpolanten bzw. Scott-Zhang Operator als Interpolationsoperator, optimale Ordnung besitzen. Beim Verfahren von García-Cervera & Roma sieht man eine deutlich schlechtere Rate für nicht konstante Magnetisierungen und zwar liegt beim L-shape die experimentell bestimmte Konvergenzrate zwischen 0.62 und 0.68, während sie beim Z-shape nur zwischen 0.43 und 0.51 liegt. Im Folgenden geben wir einen Grund für dieses Verhalten an. Bekanntlich besteht die Lösung U des García-Cervera & Roma Ansatzes aus zwei Teillösungen U_1 und U_2 . Man kann sich daher auch die Frage stellen, welche Konvergenzraten die einzelnen Teillösungen besitzen. Da die Randwerte von U_2 abhängig von U_1 sind, ist die Bestimmung einer Konvergenzrate für U_2 nicht aussagekräftig. Wir können aber die Konvergenzgeschwindigkeit von U_1 bestimmen. Als Vergleichslösung werden wir dabei die Lösung $U_1^{(n)}$ am feinsten Gitter verwenden und aus den Differenzen

$$\|U_1^{(\ell)} - U_1^{(n)}\|_{H^1(\Omega)}$$

die Konvergenzraten berechnen. Wir wählen als Beispiel das L-shape mit $\mathbf{M} = \mathbf{x}/\|\mathbf{x}\|$ aus, wobei die folgenden Aussagen auch für die Magnetisierung “jump” gültig sind. Im linken Bild in Abbildung 7.9 sind die Fehler über der Anzahl der Elemente aufgetragen und zum Vergleich ist die Konvergenzrate 0.6 eingetragen. Wie aus dem Plot ersichtlich, fällt der Fehler ungefähr mit $h^{0.6}$, was wiederum mit der experimentell bestimmten Konvergenzgeschwindigkeit für $U = U_1 + U_2$ aus Tabelle 7.3 übereinstimmt. Weiters sieht man (rechtes Bild), dass der Fehler zur einspringenden Ecke hin, größer wird.

Das rechte Bild in Abbildung 7.9 zeigt die Differenz zwischen $U_1^{(4)}$ und $U_1^{(f)}$. Dabei besitzt das feinste Netz $\mathcal{T}^{(f)}$ 786.432 Elemente und das Netz $\mathcal{T}^{(4)}$ 3.072 Elemente. Das L-shape besitzt am Punkt $(0, 0)$ eine generische Eckensingularität. Der Innenwinkel an dieser einspringenden Ecke beträgt $\frac{3}{2}\pi$. Man erwartet $u \in H^{1+2/3-\varepsilon}(\Omega)$ mit $\varepsilon > 0$ und deshalb Konvergenzraten mit $\mathcal{O}(h^{2/3})$.

Beim Z-shape beträgt der Innenwinkel an der einspringenden Ecke $\frac{7}{4}\pi$. Man erwartet $u \in H^{1+4/7-\varepsilon}(\Omega)$ mit $\varepsilon > 0$, was sich auch in Tabelle 7.3 widerspiegelt. Bei dem Ansatz von García-Cervera & Roma hat sich herauskristallisiert, dass die Konvergenzrate von U mit jener von U_1 zusammenfällt.

Auch bei dem Ansatz von Fredkin & Koehler hat sich gezeigt, dass die Konvergenzgeschwindigkeit von U_1 mit der von U übereinstimmt.

In Abbildung 7.10 sind die Gesamtlösung und die Teillösungen aus dem Ansatz von García-Cervera & Roma auf dem L-shape mit Magnetisierung $\mathbf{M} = \mathbf{x}/\|\mathbf{x}\|$ geplottet. Zum Vergleich dazu sind in Abbildung 7.11 die Gesamtlösung und die Teillösungen aus dem Ansatz von Fredkin & Koehler visualisiert. Als Interpolationsoperator wurde bei beiden Abbildungen der nodale Interpolant verwendet.

Vergleich der verschiedenen Ansätze

In einem weiteren Experiment vergleichen wir die verschiedenen Ansätze miteinander. Dazu sind in Tabelle 7.4 die experimentell beobachteten Konvergenzraten der Differenzen

$$\|U_{MET_1}^{(\ell)} - U_{MET_2}^{(\ell)}\|_{H^1(\Omega)}$$

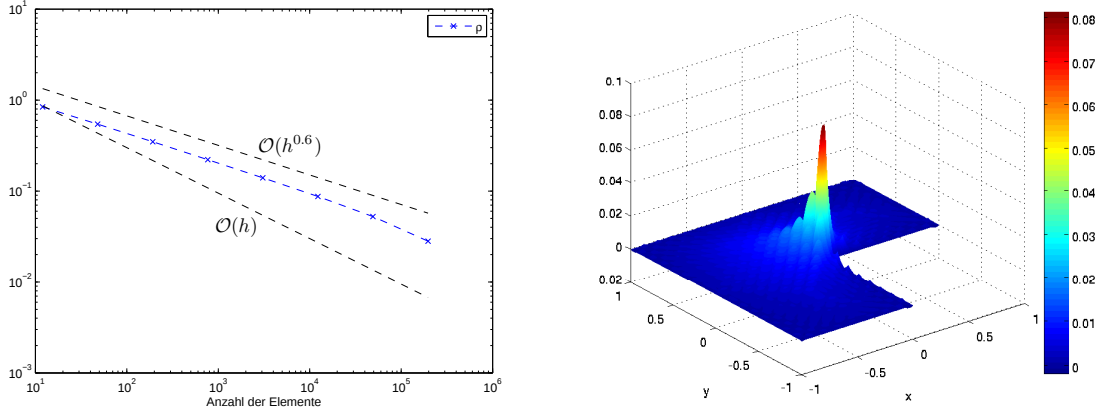


Abbildung 7.9: Im linken Bild ist die Differenz zwischen der Teillösung $U_1^{(\ell)}$ und $U_1^{(f)}$ des Verfahrens von García-Cervera & Roma aufgetragen. Es gilt $\varrho := \|U_1^{(\ell)} - U_1^{(f)}\|_{H^1(\Omega_L)}$. Dabei wurde auf dem L-shape gerechnet und als Magnetisierung wurde $\mathbf{M} = \mathbf{x}/\|\mathbf{x}\|$ verwendet. Im rechten Bild sind die Werte $|U_1^{(4)} - U_1^{(f)}|$ zu sehen. Es gilt $\#T^{(4)} = 3.072$. Der Fehler steigt zur einspringenden Ecke hin stark an.

Gebiet	$\mathbf{M}$	$\alpha_{FK/GCR}$	$\alpha_{FK/SYM}$	$\alpha_{FK/JN}$	$\alpha_{GCR/SYM}$	$\alpha_{GCR/JN}$	$\alpha_{SYM/JN}$
L-shape	const.	-2.11	1.00	1.00	1.00	1.00	1.00
	$x/\ x\ $	0.62	1.01	1.01	0.62	0.62	1.01
	jump	0.60	1.00	1.00	0.61	0.61	1.00
Z-shape	const.	-0.05	1.00	1.00	1.00	1.00	1.00
	$x/\ x\ $	0.51	1.01	1.01	0.51	0.51	1.02
	jump	0.50	0.97	0.99	0.50	0.50	1.00

Tabelle 7.4: Konvergenzraten der in der $H^1(\Omega)$ -Norm gemessenen Differenzen zwischen den einzelnen Verfahren auf dem L-shape und Z-shape mit verschiedenen Magnetisierungen. Als Interpolationsoperator wurde bei den hybriden Verfahren stets der nodale Interpolant I_h benutzt.

aufgelistet. Bei dem Interpolationsoperator, welcher bei den hybriden Verfahren verwendet wurde, handelt es sich um den nodalen Interpolanten. In der Tabelle werden die Konvergenzraten mit α_{MET_1/MET_2} bezeichnet, wobei MET_1 und MET_2 zwei verschiedene Lösungsmethoden bezeichnen. Aus der Tabelle erkennt man, dass die Differenz zwischen den beiden Kopplungsmethoden und zwischen dem Ansatz von Fredkin & Koehler und den Kopplungsmethoden gemessen in der $H^1(\Omega)$ -Norm mit der Rate h gegen Null strebt. Dies spiegelt auch die bereits erlangte Erkenntnis wider, dass das Verfahren von Fredkin & Koehler und die Kopplungsstrategien mit optimaler Rate konvergieren, das Verfahren von García-Cervera & Roma jedoch eine schlechtere Konvergenzrate besitzt. Aus der Dreiecksungleichung

$$\|U_{MET_1} - U^{(n)}\|_{H^1(\Omega)} \leq \|U_{MET_2} - U^{(n)}\|_{H^1(\Omega)} + \|U_{MET_1} - U_{MET_2}\|_{H^1(\Omega)}$$

schließen wir

$$\alpha_{MET_1} \geq \min \{ \alpha_{MET_1/MET_2}, \alpha_{MET_2} \}.$$

Mit Tabelle 7.3 und Tabelle 7.4 lässt sich dieser Sachverhalt bestätigen. Auf dem L-shape Ω_L sind in Abbildung 7.12 die Fehler mit den Magnetisierungen $\mathbf{M} = (1, 0)^T$ und $\mathbf{M} = \mathbf{x}/\|\mathbf{x}\|$ geplottet. Abbildung 7.13 zeigt die Differenz zwischen der symmetrischen Kopplungsmethode und den anderen Verfahren. Man sieht, dass der Fehler $U_{GCR} - U_{SYM}$ an der einspringenden Ecke am Größten ist.

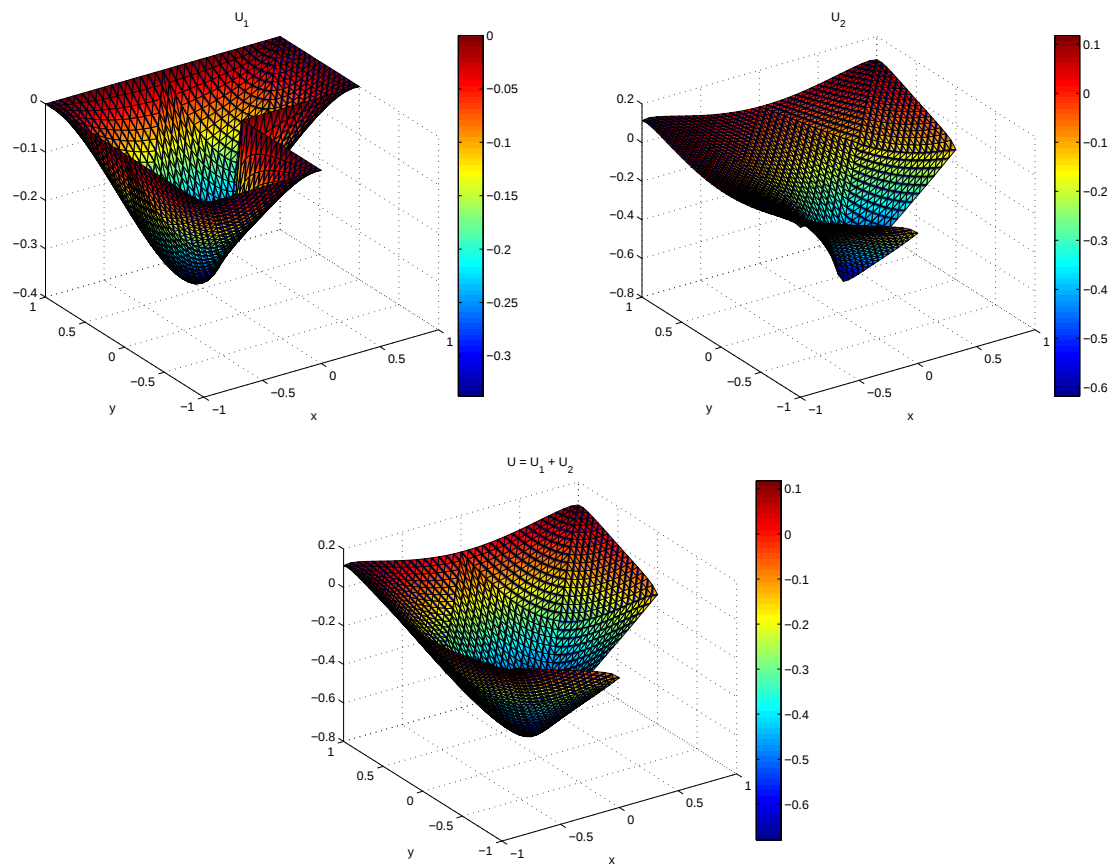


Abbildung 7.10: Vergleich der Gesamtlösung und der Teillösungen aus dem Ansatz von García-Cervera & Roma auf dem L-shape mit einer Triangulierung bestehend aus 3.072 Elementen. Als Magnetisierung wurde $\mathbf{M} = \mathbf{x}/\|\mathbf{x}\|$ gewählt und als Interpolationsoperator der nodale Interpolant I_h .

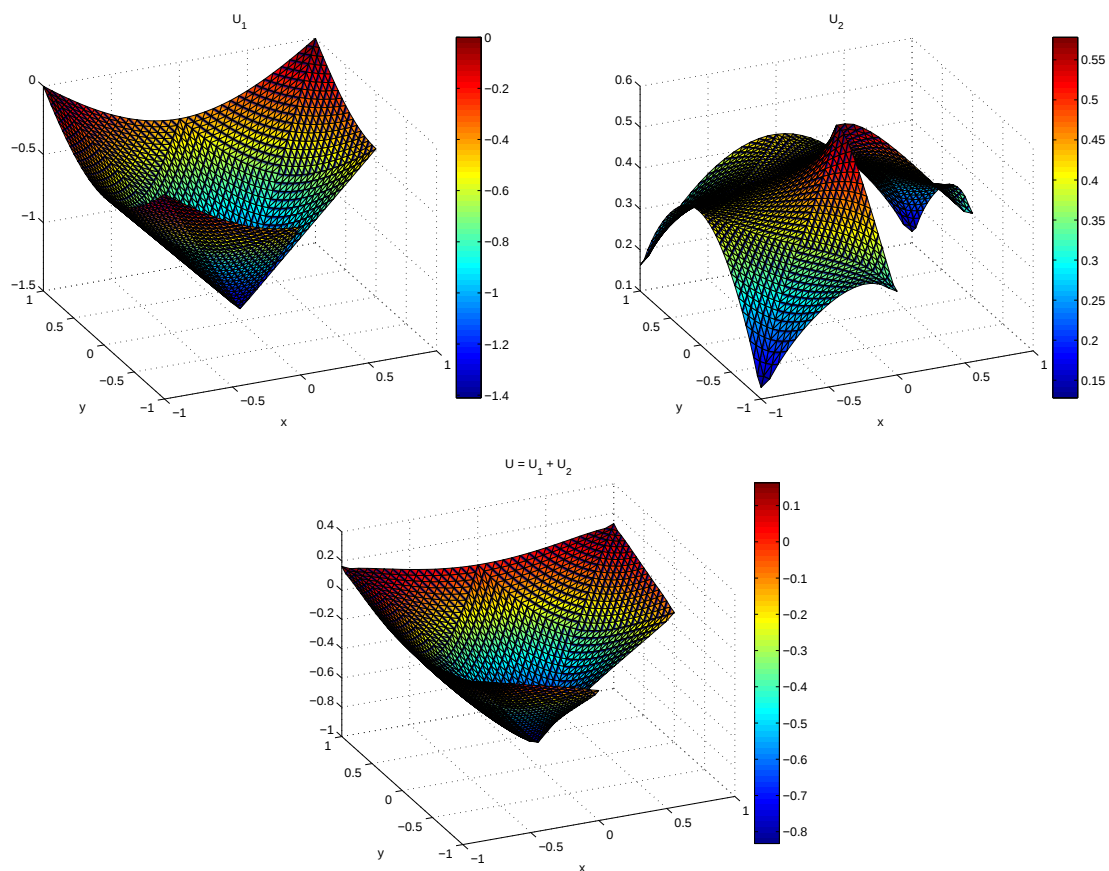


Abbildung 7.11: Vergleich der Gesamtlösung und der Teillösungen aus dem Ansatz von Fredkin & Koehler auf dem L-shape mit einer Triangulierung bestehend aus 3.072 Elementen. Als Magnetisierung wurde $\mathbf{M} = \mathbf{x}/\|\mathbf{x}\|$ gewählt und als Interpolationsoperator der nodale Interpolant I_h .

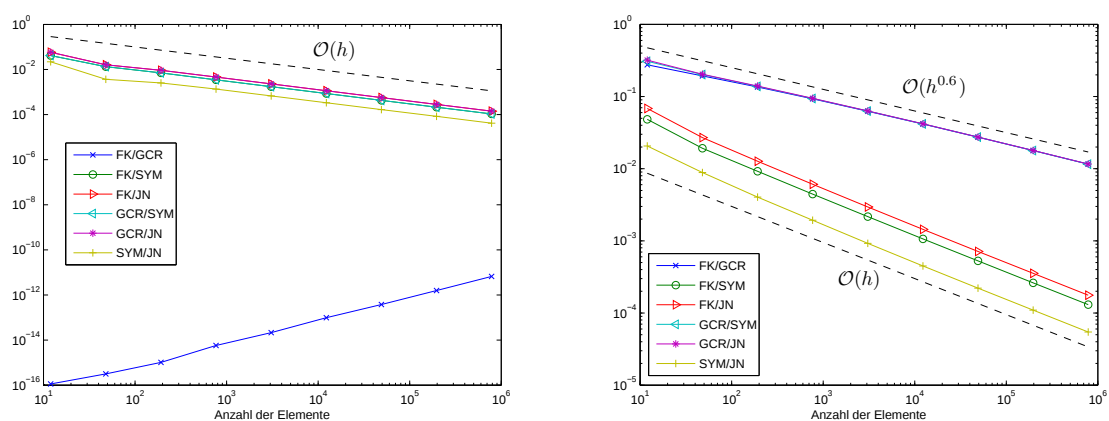


Abbildung 7.12: Differenz der verschiedenen Verfahren auf dem L-shape Ω_L gemessen in der $H^1(\Omega_L)$ -Norm. Für die Magnetisierung im linken Bild gilt $\mathbf{M} = (1, 0)^T$, für die im rechten $\mathbf{M} = \mathbf{x}/\|\mathbf{x}\|$. Als Interpolationsoperator bei den hybriden Verfahren wurde jeweils der nodale Interpolant I_h verwendet.

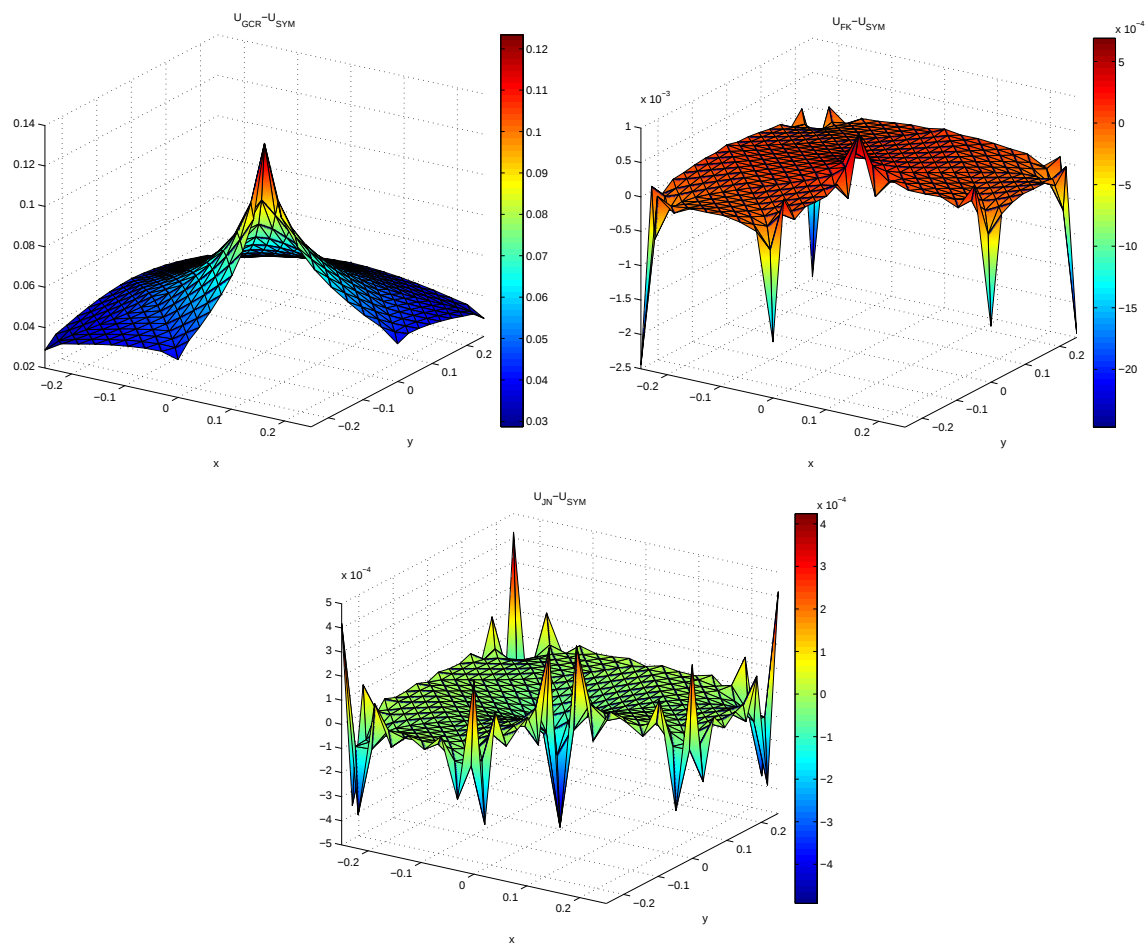


Abbildung 7.13: Vergleich der numerischen Lösungen auf dem L-shape mit einer Triangulierung bestehend aus $N = 768$ Elementen. Als Magnetisierung wurde $\mathbf{M} = \mathbf{x}/\|\mathbf{x}\|$ verwendet. Als Interpolationsoperator bei den hybriden Verfahren wurde der nodale Interpolant I_h benutzt.

Vergleich der hybriden Verfahren mit verschiedenen Interpolationsoperatoren

Wir wollen nun die Lösungen der hybriden Verfahren, welche mit verschiedenen Interpolationsoperatoren berechnet wurden, miteinander vergleichen. Dazu sind in Tabelle 7.5 die ermittelten Konvergenzraten auf dem L-shape und dem Z-shape mit verschiedenen Magnetisierungen zusammengefasst. In Abbildung 7.14 sind für das Fredkin & Koehler Verfahren die Beträge der Differenzen $|U_{INT_1} - U_{INT_2}|$ illustriert, wobei INT_1 und INT_2 jeweils zwei verschiedene Interpolationsoperatoren bezeichnen. Die Differenz der Gesamtlösungen, welche mit verschiedenen Interpolationsoperatoren berechnet werden, hängt nur von den Werten von $U_{2,INT}$ am Rand ab. Es gilt nämlich

$$\|U_{INT_1} - U_{INT_2}\|_{H^1(\Omega)} = \|U_{2,INT_1} - U_{2,INT_2}\|_{H^1(\Omega)}.$$

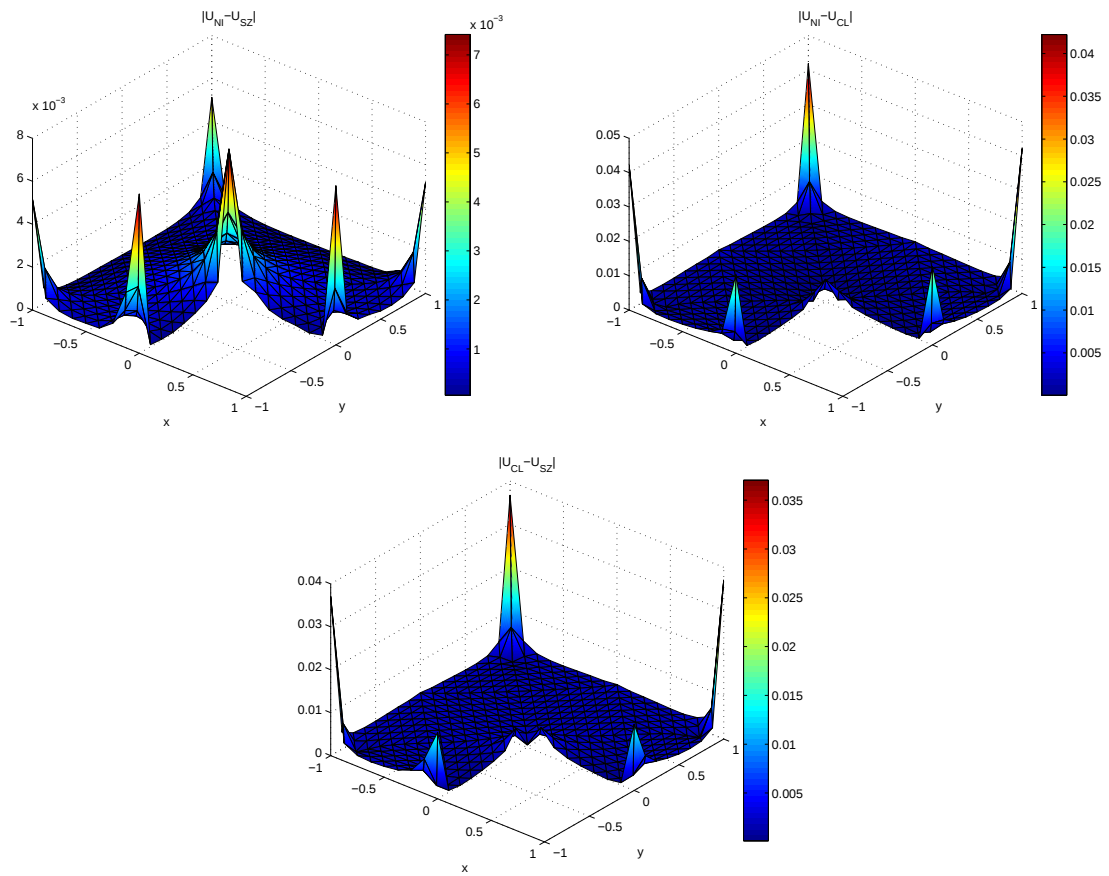


Abbildung 7.14: Differenzen $|U_{INT_1} - U_{INT_2}|$, bei Berechnung von U_{INT_i} mit dem Ansatz von Fredkin & Koehler. Bei dem verwendeten Gebiet handelt es sich um das L-shape, wobei die Triangulierung 756 Elemente enthält. Als Magnetisierung wurde $M = x/\|x\|$ gewählt.

Gebiet	M	FK, NI/CL	FK, NI/SZ	FK, CL/SZ	GCR, NI/CL	GCR, NI/SZ	GCR, CL/SZ
L-shape	const.	0.90	1.00	0.90	0.81	1.00	0.78
	$x/\ x\ $	0.86	1.00	0.84	0.49	0.61	0.42
	jump	0.87	0.86	0.81	0.72	0.79	0.73
Z-shape	const.	0.92	1.00	0.92	0.86	1.00	0.85
	$x/\ x\ $	0.87	1.00	0.87	0.45	0.53	0.43
	jump	0.68	0.64	0.61	0.47	0.66	0.43

Tabelle 7.5: Konvergenzraten der in der $H^1(\Omega)$ -Norm gemessenen Differenzen zwischen den mit verschiedenen Interpolationsoperatoren berechneten Lösungen der hybriden Verfahren auf dem L-shape und Z-shape mit verschiedenen Magnetisierungen.

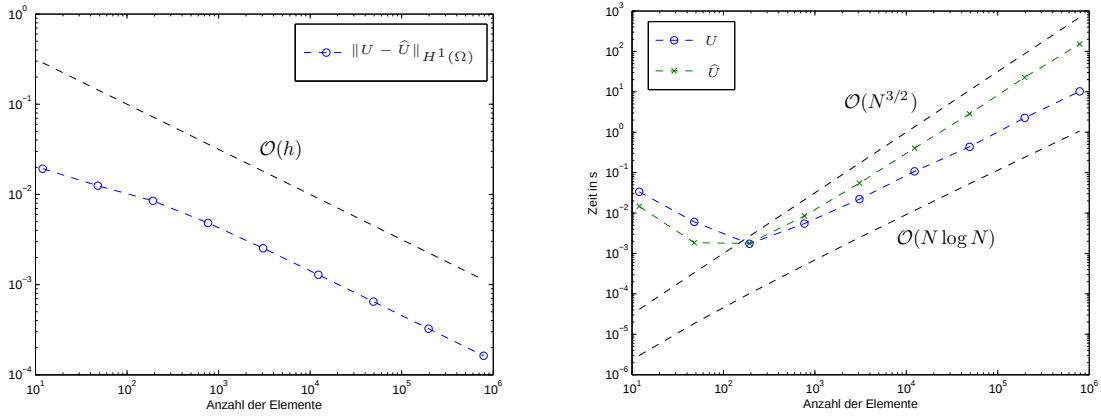


Abbildung 7.15: Das linke Bild zeigt den Fehler $\|U - \hat{U}\|_{H^1(\Omega)}$ auf dem L-shape mit Magnetisierung $\mathbf{M} = \mathbf{x}/\|\mathbf{x}\|$. Im rechten Bild sind die Zeiten, die zum Berechnen von U (blau) und $\hat{U}$ (grün) benötigt wurden, eingetragen. U ist dabei die Lösung des Ansatzes von Fredkin & Koehler und für $\hat{U}$ gilt $\hat{U} = U_1 + \hat{U}_2$, wobei $\hat{U}_2$ in (7.1) definiert wird. Als Interpolationsoperator wurde der nodale Interpolant verwendet.

Direkte Bestimmung der Teillösung U_2 der hybriden Verfahren mit Hilfe des nodalen Interpolanten

Zum Abschluss werden wir noch ein weiteres Experiment mit den beiden hybriden Ansätzen betrachten. Zuerst behandeln wir den Ansatz von Fredkin & Koehler. Dazu sei bemerkt, dass man u_2 anstatt das Problem

$$\begin{aligned} \Delta u_2 &= 0 && \text{in } \Omega, \\ \gamma_0 u_2 &= (\mathcal{K} - 1/2)\gamma_0 u_1 && \text{auf } \Gamma \end{aligned}$$

zu lösen, auch über

$$u_2 = \tilde{\mathcal{K}}\gamma_0 u_1$$

berechnen kann (vgl. Abschnitt 4.2). In der numerischen Umsetzung wird U_2 in Algorithmus 4.2 ersetzt durch $\hat{U}_2 \in \mathcal{S}^1(\mathcal{T})$ mit

$$\hat{U}_2(z) = \begin{cases} (\tilde{\mathcal{K}}\gamma_0 U_1)(z) & \text{für } z \in \mathcal{M}_F, \\ (\mathcal{K} - \frac{\alpha}{2\pi})(z) & \text{für } z \in \mathcal{M}_\Gamma, \end{cases} \quad (7.1)$$

wobei $\frac{\alpha}{2\pi}$ die in (2.23) definierte Größe ist. Die Gesamtlösung wird mit $\hat{U}$ bezeichnet und ergibt sich aus $\hat{U} := U_1 + \hat{U}_2$. Mit U_1 notieren wir die Lösung aus Schritt 1 von Algorithmus 4.2 und mit U die Gesamtlösung aus

Algorithmus 4.2. Der Doppelschichtpotentialoperator $\tilde{\mathcal{K}}$ wird dabei mit Hilfe der Funktion **evaluateK** aus der Programmbibliothek HILBERT 2009 [2] an den Knoten ausgewertet. In Abbildung 7.15 ist zum einen der Fehler zwischen U_2 und $\hat{U}_2$ auf verschiedenen Triangulierungen illustriert (linkes Bild) und zum anderen die Zeit die benötigt wurde um U bzw. $\hat{U}$ zu bestimmen (rechtes Bild). Als Gebiet wurde hier das L-shape gewählt und als Magnetisierung $\mathbf{M} = \mathbf{x}/\|\mathbf{x}\|$. Der Fehler

$$\|U - \hat{U}\|_{H^1(\Omega)} = \|U_1 + U_2 - (U_1 + \hat{U}_2)\|_{H^1(\Omega)} = \|U_2 - \hat{U}_2\|_{H^1(\Omega)}$$

fällt mit der Ordnung h . In der praktischen Anwendung sind die Zeiten, die zum Lösen aufgewandt wurden, interessant. Wie in Abbildung 7.15 zu sehen, ist der Aufwand zum Berechnen von U in der Größenordnung von $N \log N$ während der Aufwand für $\hat{U}$ in der Größenordnung von $N^{3/2}$ ist, wobei wir mit N wiederum die Anzahl der Elemente bezeichnen. Beispielsweise benötigt man bei dem feinsten Netz ($N = 786.432$) 10.22 Sekunden zum Berechnen von U und 151.47 Sekunden zum Berechnen von $\hat{U}$. Weiters ist in Abbildung 7.16 der absolute Fehler $|U_2(x) - \hat{U}_2(x)|$ zu sehen, wobei hier auffällt, dass der Fehler zum Rand hin größer wird. Man beachte auch, dass der Fehler am Rand selbst verschwindet, also

$$|U_2(x) - \hat{U}_2(x)| = 0$$

für $x \in \Gamma$ gilt, da für die Bestimmung von U_2 der nodale Interpolant verwendet wurde.

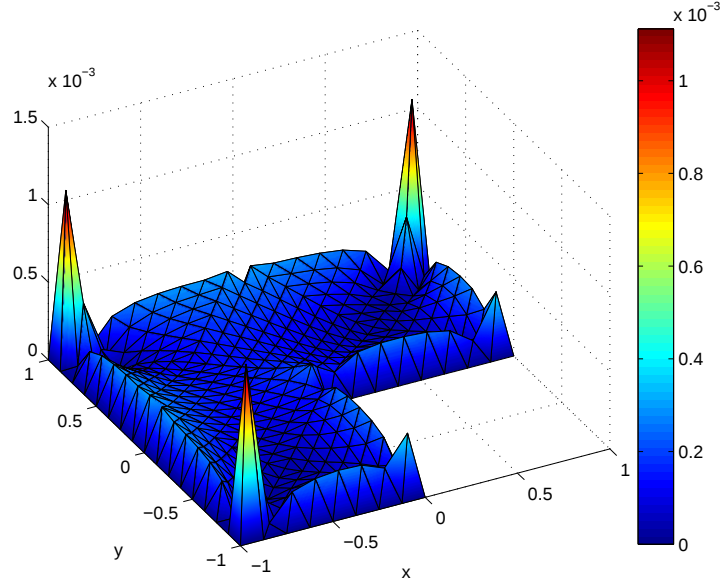


Abbildung 7.16: Absoluter Fehler $|U(x) - \hat{U}(x)| = |U_2(x) - \hat{U}_2(x)|$ auf dem L-shape mit $N = 768$ Elementen und mit Magnetisierung $\mathbf{M} = \mathbf{x}/\|\mathbf{x}\|$. U wurde dabei mit dem Fredkin & Koehler Ansatz bestimmt. Es gilt $\hat{U} = U_1 + \hat{U}_2$. Dabei wird $\hat{U}_2$ in (7.1) definiert. Als Interpolationsoperator wurde der nodale Interpolant verwendet.

Bei dem Ansatz von García-Cervera & Roma werden wir nun ähnlich wie zuvor vorgehen. Dazu ersetzen wir U_2 in Algorithmus 4.1 durch

$$\hat{U}_2(z) = \begin{cases} \tilde{\mathcal{V}}(\Pi_h^0(\mathbf{M} \cdot \mathbf{n}) - \gamma_1 U_1)(z) & \text{für } z \in \mathcal{M}_F, \\ \mathcal{V}(\Pi_h^0(\mathbf{M} \cdot \mathbf{n}) - \gamma_1 U_1)(z) & \text{für } z \in \mathcal{M}_\Gamma, \end{cases} \quad (7.2)$$

wobei $\hat{U}_2 \in \mathcal{S}^1(\mathcal{T})$ gilt. Sei U_1 die Lösung aus Schritt 1 von Algorithmus 4.1 und U die Gesamtlösung aus Algorithmus 4.1. Mit $\hat{U}$ bezeichnen wir $\hat{U} := U_1 + \hat{U}_2$. In Abbildung 7.17 sind die Fehler

$$\|U - \hat{U}\|_{H^1(\Omega)} = \|U_1 + U_2 - (U_1 + \hat{U}_2)\|_{H^1(\Omega)} = \|U_2 - \hat{U}_2\|_{H^1(\Omega)}$$

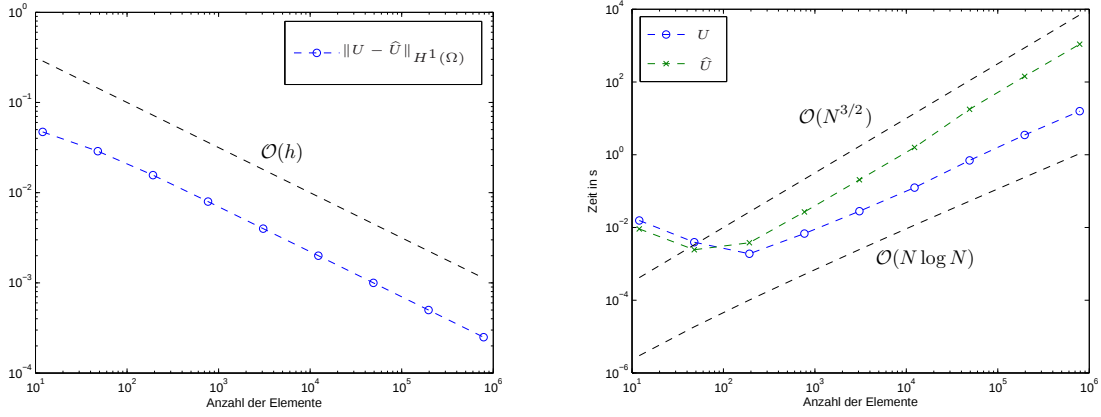


Abbildung 7.17: Das linke Bild zeigt den Fehler $\|U - \hat{U}\|_{H^1(\Omega)}$ auf dem L-shape mit Magnetisierung $\mathbf{M} = \mathbf{x}/\|\mathbf{x}\|$ aufgetragen über der Anzahl der Elemente. Im rechten Bild sind die Zeiten, die zum Berechnen von U (blau) und $\hat{U}$ (grün) benötigt wurden, eingetragen. U ist dabei die Lösung des Ansatzes von García-Cervera & Roma und für $\hat{U}$ gilt $\hat{U} = U_1 + \hat{U}_2$, wobei $\hat{U}_2$ in (7.2) definiert wird. Als Interpolationsoperator wurde der nodale Interpolant verwendet.

über der Anzahl der Elemente aufgetragen. Im rechten Bild sind dabei wiederum die Zeiten angeführt, die zum Berechnen von U und $\hat{U}$ benötigt wurden. Hierbei ist wieder zu erkennen, dass der Fehler $\|U_2 - \hat{U}_2\|_{H^1(\Omega)}$ die Konvergenzgeschwindigkeit h besitzt. Die Zeiten zum Berechnen von U besitzen einen Aufwand $\mathcal{O}(N \log N)$, während die Zeiten zum Bestimmen von $\hat{U}$ mit $N^{3/2}$ steigen.

7.3 Zusammenfassung

Zu Beginn dieses Kapitels haben wir die Konvergenzgeschwindigkeiten der verschiedenen Verfahren auf der Kreisscheibe untersucht und festgestellt, dass sich bei allen Methoden eine optimale Konvergenzrate einstellt. Man beobachtet sogar schnellere Raten als man a priori erwartet. Und zwar liegt die Ordnung zwischen h und $h^{1.85}$. Weiters haben wir gesehen, dass die Konvergenzrate auf dem L-shape bzw. Z-shape bei den beiden Kopplungsmethoden und dem Ansatz von Fredkin & Koehler optimal ist. Das Verfahren von García-Cervera & Roma zeigt dabei jedoch schlechtere Raten bei bestimmten Magnetisierungen. Für die hybriden Verfahren ist es noch notwendig einen Interpolationsoperator zu wählen. Dabei hat sich gezeigt, dass sich die besten Ergebnisse mit dem nodalen Interpolanten und dem Scott-Zhang Operator erzielen lassen.

Des weiteren wurde auch der Zeitaufwand beim Berechnen mit den verschiedenen Verfahren untersucht. Wie sich herausgestellt hat, benötigen die hybriden Verfahren im Vergleich zu den beiden FEM/BEM - Kopplungsmethoden am wenigsten Zeit.

Von diesen Ergebnissen ausgehend besteht daher beim Ansatz von Fredkin & Koehler das beste Mittelmaß aus Genauigkeit und Zeitaufwand.

Anhang A

Notationen

$d = 2, 3$	Dimension.
Ω	Beschränktes Lipschitz Gebiet in $\mathbb{R}^d$.
$\Omega^{ext} := \mathbb{R}^d \setminus \overline{\Omega}$	Außenraum.
Γ	Rand von Ω .
$\text{diam}(U)$	Durchmesser der beschränkten Menge $U \subseteq \mathbb{R}^d$.
$ U $	Lebesgue Maß der Menge U .
$\mathbf{n}$	Normalenvektor auf Γ . Zeigt von Ω nach Ω^{ext} .
$\mathbf{B}$	Magnetische Flussdichte / Magnetisches Feld.
$\mathbf{H}, \mathbf{H}_s$	Magnetische Feldstärke, Streufeld.
$\mathbf{M}$	Magnetisierung.
$\text{div } \mathbf{F}$	Divergenz eines Vektorfelds $\mathbf{F}$.
$\text{rot } \mathbf{F}$	Rotation eines Vektorfelds $\mathbf{F}$.
$\ \mathbf{x}\ $	Euklidische Norm von $\mathbf{x} \in \mathbb{R}^d$.
$C^n(U)$	Menge der n -mal stetig differenzierbaren Funktionen auf U .
$C^\infty(U)$	Menge der unendlich oft differenzierbaren Funktionen auf U .
$C_0^\infty(U)$	Menge der unendlich oft differenzierbaren Funktionen mit kompakten Träger in U .
$\text{supp}(f)$	Träger einer Funktion f .
δ_{ij}	Kronecker Delta.
$L^2(U)$	Raum der quadratisch integrierbaren Funktionen auf U .
$\ \cdot\ _{\mathcal{H}}$	Norm auf Hilbertraum $\mathcal{H}$.
$\langle \cdot, \cdot \rangle_{\mathcal{H}}$	Skalarprodukt auf Hilbertraum $\mathcal{H}$.
$\ \cdot\ $	Energienorm.
$\langle \cdot, \cdot \rangle$	Energieskalarprodukt.
$H^\ell(U)$	Sobolevraum der ℓ -ten Ordnung auf U .
$H_{lok}^\ell(U)$	Raum aller Funktionen, die auf jedem Kompaktum K in $H^\ell(K)$ liegen.
$H_*^\ell(U)$	Raum aller Funktionen aus $H^\ell(U)$ mit verschwindendem Integralmittel.
$H_0^\ell(U)$	Raum aller Funktionen aus $H^\ell(U)$ mit Nullrand.
$\tilde{H}^{-\ell}(\Omega)$	Dualraum von $H^\ell(\Omega)$ bezüglich des erweiterten L^2 -Skalarprodukts.
$H^{1/2}(\Gamma)$	Sobolevraum der Ordnung $1/2$ auf Γ , Spurraum von $H^1(\Omega)$.
$H^{-1/2}(\Gamma)$	Dualraum von $H^{1/2}(\Gamma)$ bezüglich des erweiterten L^2 -Skalarprodukts.
γ_0, γ_0^{int}	Innere Spur.
γ_0^{ext}	Äußerer Spur.
γ_1, γ_1^{int}	Innere Konormalenableitung.

γ_1^{ext}	Äußere Konormalenableitung.
$\mathbf{M} \cdot \mathbf{n}$	Normalenkomponente von $\mathbf{M}$ auf dem Rand.
$\tilde{\mathcal{K}}$	Doppelschichtpotentialoperator.
$\gamma_0 \tilde{\mathcal{K}} = \mathcal{K} - 1/2$	Spur des Doppelschichtpotentialoperators.
$\tilde{\mathcal{V}}$	Einfachschichtpotentialoperator.
$\gamma_0 \tilde{\mathcal{V}} = \mathcal{V}$	Spur des Einfachschichtpotentialoperators.
$\tilde{\mathcal{W}}$	Hypersingulärer Integraloperator.
$\gamma_0 \tilde{\mathcal{W}} = \mathcal{W}$	Spur des hypersingulären Integraloperators.
Ω_z	Knotenpatch.
I_h	Nodaler Interpolant.
J_h	Clément Operator.
S_h	Scott-Zhang Operator.
Π_h^0	L^2 -Orthogonalprojektion auf den Raum der stückweise konstanten Funktionen.
$\mathcal{T}$	Triangulierung von Ω .
$\mathcal{T}_\Gamma$	Triangulierung des Rands.
T	Element von $\mathcal{T}$.
$\mathcal{M}$	Menge der Knoten einer Triangulierung.
$\mathcal{M}_\Gamma$	Knoten am Rand.
$\mathcal{E}$	Kanten einer Triangulierung.
$\mathcal{E}_\Gamma$	Kanten am Rand.
h_T	Durchmesser von T .
h	Lokale Netzweitenfunktion, mit $h _T = h_T$.
V_T	Volumen von T .
V	Lokale Volumsfunktion, mit $V _T = V_T$.
$S^1(\mathcal{T})$ bzw. $S^1(\mathcal{T}_\Gamma)$	Raum der stückweise affinen, global stetigen Funktionen auf $\mathcal{T}$ bzw. $\mathcal{T}_\Gamma$.
$S_0^1(\mathcal{T})$	Raum aller Funktionen aus $S^1(\mathcal{T})$ mit Nullrand.
$S_*^1(\mathcal{T})$	Raum aller Funktionen aus $S^1(\mathcal{T})$ mit verschwindendem Integralmittel.
η_z	Nodale Basisfunktion von Knoten $z \in \mathcal{M}$.
$\mathcal{P}^n(\mathcal{T})$ bzw. $\mathcal{P}^n(\mathcal{T}_\Gamma)$	Raum der stückweise polynomialen Funktionen mit Grad $\leq n$ auf $\mathcal{T}$ bzw. $\mathcal{T}_\Gamma$.
χ_T	Basisfunktion von $\mathcal{P}^0(\mathcal{T})$ mit $\chi_T _{T'} = \delta_{T,T'}$.
$\sigma(\mathcal{T})$	Formregularitätskonstante.
u	Skalares magnetostatisches Potential.
u^{int}	Potential im Innenraum.
u^{ext}	Potential im Außenraum.
U	Approximationslösung in $S^1(\mathcal{T})$.
$\mathbf{H}_h$	Approximationslösung des Streufelds.

Anhang B

Quellcodes

Listing B.1: sortMesh.m

```
1  function [coordinates,elements,boundary,boundary2elements] = ...
2      sortMesh(coordinates,elements,boundary)
3
4  %[COORDINATES,ELEMENTS,BOUNDARY,BOUNDARY2ELEMENTS] = ...
5  %     SORTMESH(COORDINATES,ELEMENTS,BOUNDARY)
6  %     Sorts FEM-Mesh such that boundary nodes are the first nodes in the
7  %     coordinates field and computes the boundary2elements array.
8  %
9  %     INPUT:
10 %         COORDINATES ... field describing nodes
11 %         ELEMENTS ... elements described by their indices
12 %         BOUNDARY ... boundary elements described by their indices
13 %
14 %     OUTPUT:
15 %         COORDINATES ... sorted coordinates field
16 %         ELEMENTS ... sorted elements field
17 %         BOUNDARY ... sorted boundary field
18 %         BOUNDARY2ELEMENTS ... contains the indices of the elements that
19 %         share a boundary element, i.e.
20 %         BOUNDARY2ELEMENTS(I)=J means, that the I-th boundary element
21 %         lays on the J-th volume element.
22
23 %*** Number of nodes and elements
24 nC = size(coordinates,1);
25 nE = size(elements,1);
26
27 %*** Determine nodes on boundary and interior nodes (freenodes)
28 nodes_boundary = unique(boundary);
29 freenodes = setdiff((1:nC)',nodes_boundary);
30
31 %*** Build permutation such that boundary nodes are first
32 nodes = [nodes_boundary;freenodes];
33 [foo,permutation] = sort(nodes);
34
35 %*** Permute indices of nodes
36 coordinates(permutation,:) = coordinates;
37 elements = permutation(elements);
38 boundary = permutation(boundary);
39
40 %*** Determine which element contains a boundary edge
41 [edge2nodes,element2edges,boundary2edges] ...
42     = provideGeometricData(elements,boundary);
43 edge2elements = zeros(size(edge2nodes,1),1);
```

```

44 edge2elements(element2edges(:)) = repmat( reshape(1:nE,nE,1), 3, 1);
45 boundary2elements = edge2elements(boundary2edges);

```

Listing B.2: h1norm.m

```

1  function [h1n l2n sh1n] = h1norm(coordinates,elements,x)
2  %[H1N[,L2N,SH1N]] = H1NORM(COORDINATES,ELEMENTS,X)
3  %   Computes the H^1-Norm of a S^1 function
4  %
5  %   INPUT:
6  %       COORDINATES ... field describing nodes
7  %       ELEMENTS ... elements described by their indices
8  %       X ... S^1 function given nodewise
9  %
10 %   OUTPUT:
11 %       H1N ... H^1 Norm
12 %   OPTIONAL OUTPUT:
13 %       L2N ... L^2 Norm
14 %       SH1N ... H^1 Seminorm
15
16 %*** First vertex of elements and corresponding edge vectors
17 c1 = coordinates(elements(:,1),:);
18 d21 = coordinates(elements(:,2),:) - c1;
19 d31 = coordinates(elements(:,3),:) - c1;
20
21 %*** Vector of element volumes 2*|T|
22 area2 = d21(:,1).*d31(:,2)-d21(:,2).*d31(:,1);
23
24 %*** L2-Norm
25 norml2 = sum( 1/24*(sum(x(elements(:,1)).^2+x(elements(:,2)).^2+x(elements(:,3)).^2,2)...
26 +sum(x(elements(:,1))+x(elements(:,2))+x(elements(:,3)),2).^2).*area2);
27 l2n = sqrt(norml2);
28
29 %*** Elementwise gradient of u_h
30 u21 = repmat( (x(elements(:,2))-x(elements(:,1)))./area2, 1,2);
31 u31 = repmat( (x(elements(:,3))-x(elements(:,1)))./area2, 1,2);
32 gradh = (d31.*u21 - d21.*u31)*[0 -1 ; 1 0];
33
34 %*** H1-Seminorm
35 snorm2 = sum(gradh.^2,2);
36 snorm2 = sum( snorm2.*area2 )/2;
37 sh1n = sqrt(snorm2);
38
39 %*** H1-Norm
40 h1n = sqrt( norml2 + snorm2);

```

Listing B.3: genMeshC.m

```

1  function [coordinates, elements, boundary] = ...
2  genMeshC(R,nK,nR)
3  %[COORDINATES,ELEMENTS,BOUNDARY] = GENMESH(C,R,NK,NR)
4  %   Creates a triangulation of a disc with built-in Matlab function
5  %   delaunay.
6  %
7  %   INPUT:
8  %       R ... radius of disc
9  %       NK ... number of nodes on boundary
10 %       NR ... number of "inner" rings
11 %
12 %   OUTPUT:
13 %       COORDINATES ... field describing nodes
14 %       ELEMENTS ... elements described by their indices
15 %       BOUNDARY ... boundary elements described by their indices

```

```

16 %
17 %   ALGORITHM:
18 %       -) Create nodes on boundary.
19 %       -) Create nodes on inner circles. Number of nodes
20 %           on an inner circle decreases as the radius of
21 %           the circle decreases.
22 %       -) Create triangulation with delaunay algorithm
23 %       -) Determine boundary elements
24
25 %*** Nodes on boundary
26 x = R*cos((1:nK)/nK*2*pi)';
27 y = R*sin((1:nK)/nK*2*pi)';
28
29 %*** Inner rings
30 for i = 1:nR
31     nN = round(i/(nR+1)*nK);
32     x = [x; i/(nR+1)*R*cos((1:nN)/nN*2*pi)'];
33     y = [y; i/(nR+1)*R*sin((1:nN)/nN*2*pi)'];
34 end
35
36 %*** Add node (0,0) to mesh
37 x = [x; 0];
38 y = [y; 0];
39
40 %*** Create mesh with delaunay algorithm
41 elements = delaunay(x,y);
42 coordinates = [x y];
43
44 %*** Sort elements such that longest edge of an element
45 %*** is the first one. Conserves math. pos. direction of
46 %*** the nodes.
47 l1 = sqrt(sum( (coordinates(elements(:,1),:)-...
48               coordinates(elements(:,2),:)).^2,2));
49 l2 = sqrt(sum( (coordinates(elements(:,2),:)-...
50               coordinates(elements(:,3),:)).^2,2));
51 l3 = sqrt(sum( (coordinates(elements(:,3),:)-...
52               coordinates(elements(:,1),:)).^2,2));
53
54 for i = 1:size(elements,1)
55     if l1(i) < l2(i)
56         if l2(i) < l3(i)
57             elements(i,:) = elements(i,[3 1 2]);
58         else %*** l1 < l2 && l2 > l3
59             elements(i,:) = elements(i,[2 3 1]);
60         end
61     else %*** l1 > l2
62         if l1(i) < l3(i)
63             elements(i,:) = elements(i,[3 1 2]);
64         end
65     end
66 end
67
68 %*** Determine boundary elements
69 boundary = determineBoundary(coordinates,elements);
70
71 %*** Sort Mesh, such that boundary nodes are first
72 %*** Not necessary at all, due to the architecture of the program
73 [coordinates, elements, boundary] = ...
74     sortMesh(coordinates,elements,boundary);

```

Listing B.4: determineBoundary.m

```

1 function boundary = determineBoundary(coordinates,elements)
2
3 %BOUNDARY = DETERMINEBOUNDARY(COORDINATES,ELEMENTS)

```

```

4 %   Determines the boundary elements from the given triangulation
5 %
6 %   INPUT:
7 %       COORDINATES ... field describing nodes
8 %       ELEMENTS ... elements described by their indices
9 %
10 %   OUTPUT:
11 %       BOUNDARY ... boundary elements described by their indices
12 %       BOUNDARY is a nBx2 field, where nB denotes the number of
13 %       boundary elements. The nodes are stored in the usual
14 %       math.pos. sense.
15
16 %*** All the edges
17 edges = [elements(:, [1 2]); elements(:, [2 3]); elements(:, [3 1])];
18 edges = sort(edges, 2);
19
20 %*** Find unique edges
21 [foo, idx, jdx] = unique(edges, 'rows');
22 jdx = sort(jdx);
23
24 %*** Determine edges that appear only once, i.e. the boundary edges
25 kdx = find(jdx(1:end-1)==jdx(2:end));
26 kdx = [kdx; kdx+1];
27 elldx = setdiff( (1:length(jdx))', kdx);
28 jdx = jdx(elldx);
29 boundary = foo(jdx, :);
30
31 %*** Are the Boundary elements in math. pos. sense?
32 points = [elements(:, 3); elements(:, 1); elements(:, 2)];
33 cp = coordinates(points, :);
34 cp = cp(idx(jdx), :);
35 r1 = coordinates(boundary(:, 1), :)-cp;
36 r2 = coordinates(boundary(:, 2), :)-coordinates(boundary(:, 1), :);
37 idx = find(r1(:, 1).*r2(:, 2)-r1(:, 2).*r2(:, 1) < 0);
38 boundary(idx, :) = boundary(idx, [2 1]);

```

Listing B.5: buildMeshIN2D.cpp

```

1 // buildMeshIN2D.cpp
2 // Uses NETGEN to create a triangulation on a domain.
3
4 #include "mex.h"
5
6 namespace nglib {
7 #include <nglib.h>
8 }
9
10 using namespace std;
11 using namespace nglib;
12
13 void getCoordinates(Ng_Mesh * mesh, double * coordinates, int nC);
14 void getElements(Ng_Mesh * mesh, double * elements, int nE);
15 void getBoundary(Ng_Mesh * mesh, double * boundary, int nB);
16 void sortElementsLongest(double * elements, int nE, double * coordinates, int nC);
17 double square(double x);
18
19 void mexFunction(int nlhs, mxArray *plhs[], int nrhs, const mxArray *prhs[])
20 {
21     // Geomety file
22     char * geofile;
23     geofile = mxArrayToString(prhs[0]);
24
25     // Initialize NG
26     Ng_Init();
27

```

```

28 // Load geometry from *.in2d file
29 Ng_Geometry_2D * geom2d;
30 geom2d = Ng_LoadGeometry_2D(geofile);
31
32 // Set meshing parameters
33 Ng_Meshing_Parameters mp;
34 int field_n;
35 if(nrhs==3) {
36     field_n = mxGetFieldNumber(prhs[2], "uselocalh");
37     if(field_n≥0)
38         mp.uselocalh = (int) mxGetScalar(mxGetFieldByNumber(prhs[2],0,field_n));
39
40     field_n = mxGetFieldNumber(prhs[2], "maxh");
41     if(field_n≥0)
42         mp.maxh = mxGetScalar(mxGetFieldByNumber(prhs[2],0,field_n));
43
44     field_n = mxGetFieldNumber(prhs[2], "minh");
45     if(field_n≥0)
46         mp.minh = mxGetScalar(mxGetFieldByNumber(prhs[2],0,field_n));
47
48     field_n = mxGetFieldNumber(prhs[2], "fineness");
49     if(field_n≥0)
50         mp.fineness = mxGetScalar(mxGetFieldByNumber(prhs[2],0,field_n));
51
52     field_n = mxGetFieldNumber(prhs[2], "grading");
53     if(field_n≥0)
54         mp.grading = mxGetScalar(mxGetFieldByNumber(prhs[2],0,field_n));
55
56     field_n = mxGetFieldNumber(prhs[2], "elementsperedge");
57     if(field_n≥0)
58         mp.elementsperedge = mxGetScalar(mxGetFieldByNumber(prhs[2],0,field_n));
59
60     field_n = mxGetFieldNumber(prhs[2], "elementspercurve");
61     if(field_n≥0)
62         mp.elementspercurve = mxGetScalar(mxGetFieldByNumber(prhs[2],0,field_n));
63
64     field_n = mxGetFieldNumber(prhs[2], "closeedgeenable");
65     if(field_n≥0)
66         mp.closeedgeenable = (int) mxGetScalar(mxGetFieldByNumber(prhs[2],0,field_n));
67
68     field_n = mxGetFieldNumber(prhs[2], "closeedgefact");
69     if(field_n≥0)
70         mp.closeedgefact = mxGetScalar(mxGetFieldByNumber(prhs[2],0,field_n));
71
72     field_n = mxGetFieldNumber(prhs[2], "secondorder");
73     if(field_n≥0)
74         mp.secondorder = (int) mxGetScalar(mxGetFieldByNumber(prhs[2],0,field_n));
75
76     field_n = mxGetFieldNumber(prhs[2], "quad_dominated");
77     if(field_n≥0)
78         mp.quad_dominated = (int) mxGetScalar(mxGetFieldByNumber(prhs[2],0,field_n));
79
80     field_n = mxGetFieldNumber(prhs[2], "meshsize_filename");
81     if(field_n≥0)
82         mp.meshsize_filename = mxArrayToString(mxGetFieldByNumber(prhs[2],0,field_n));
83
84     field_n = mxGetFieldNumber(prhs[2], "optsurfmeshenable");
85     if(field_n≥0)
86         mp.optsurfmeshenable = (int) mxGetScalar(mxGetFieldByNumber(prhs[2],0,field_n));
87
88     field_n = mxGetFieldNumber(prhs[2], "optvolmeshenable");
89     if(field_n≥0)
90         mp.optvolmeshenable = (int) mxGetScalar(mxGetFieldByNumber(prhs[2],0,field_n));
91
92     field_n = mxGetFieldNumber(prhs[2], "optsteps_2d");
93     if(field_n≥0)

```

```

94     mp.optsteps_2d = (int) mxGetScalar(mxGetFieldByNumber(prhs[2],0,field_n));
95 }
96
97 // Generate Mesh
98 Ng_Mesh * mesh;
99 Ng_GenerateMesh_2D(geom2d, &mesh, &mp);
100
101 // Uniform refinement (second parameter)
102 if(nrhs==2 || nrhs==3) {
103     int nURef = (int) mxGetScalar(prhs[1]);
104     int i;
105     for(i=0; i<nURef; i++) {
106         Ng_2D_Uniform_Refinement(geom2d, mesh);
107     }
108 }
109
110 // Get coordinates field
111 int nC;
112 nC = Ng_GetNP_2D(mesh);
113 plhs[0] = mxCreateDoubleMatrix(nC,2,mxREAL);
114 double * coordinates = mxGetPr(plhs[0]);
115
116 getCoordinates(mesh, coordinates, nC);
117
118 // Get elements field
119 int nE;
120 nE = Ng_GetNE_2D(mesh);
121 plhs[1] = mxCreateDoubleMatrix(nE,3,mxREAL);
122 double * elements = mxGetPr(plhs[1]);
123
124 getElements(mesh, elements, nE);
125
126 // Sort elements field such that longest edge is the first
127 sortElementsLongest(elements, nE, coordinates, nC);
128
129 // Get boundary field
130 int nB;
131 nB = Ng_GetNSeg_2D(mesh);
132 plhs[2] = mxCreateDoubleMatrix(nB,2,mxREAL);
133 double * boundary = mxGetPr(plhs[2]);
134
135 getBoundary(mesh, boundary, nB);
136
137 // Save mesh on disk,
138 // just for testing issues
139 Ng_SaveMesh(mesh, "test.vol");
140 Ng_DeleteMesh(mesh);
141
142 // Exit NG
143 Ng_Exit();
144 }
145
146 // Coordinates of nodes
147 void getCoordinates(Ng_Mesh * mesh, double * coordinates, int nC)
148 {
149     int i;
150     double point[2];
151
152     for(i=0; i<nC; i++) {
153         Ng_GetPoint_2D(mesh, i+1, point);
154         coordinates[i] = point[0];
155         coordinates[i+nC] = point[1];
156     }
157 }
158
159 // Elements

```

```

160 void getElements(Ng_Mesh * mesh, double * elements, int nE)
161 {
162     int i;
163     int nodes[3];
164
165     for(i=0; i<nE; i++) {
166         Ng_GetElement_2D(mesh, i+1, nodes);
167         elements[i] = nodes[0];
168         elements[i+nE] = nodes[1];
169         elements[i+2*nE] = nodes[2];
170     }
171 }
172
173 // Boundary elements
174 void getBoundary(Ng_Mesh * mesh, double * boundary, int nB)
175 {
176     int i;
177     int nodes[2];
178
179     for(i=0; i<nB; i++) {
180         Ng_GetSegment_2D(mesh, i+1, nodes);
181         boundary[i] = nodes[0];
182         boundary[i+nB] = nodes[1];
183     }
184 }
185
186 // Sort elements field, such that longest edge is first edge
187 void sortElementsLongest(double * elements, int nE, double * coordinates, int nC)
188 {
189     int i, j;
190     double l1, l2, l3;
191     double coord[3][2];
192     int tmp[3];
193
194     for(i=0; i<nE; i++) {
195         for(j=0; j<3; j++) {
196             coord[j][0] = coordinates[(int) elements[i+j*nE]];
197             coord[j][1] = coordinates[(int) elements[i+j*nE] + nC];
198         }
199
200         l1 = square(coord[0][0] - coord[1][0]) + square(coord[0][1] - coord[1][1]);
201         l2 = square(coord[1][0] - coord[2][0]) + square(coord[1][1] - coord[2][1]);
202         l3 = square(coord[2][0] - coord[0][0]) + square(coord[2][1] - coord[0][1]);
203
204         tmp[0] = elements[i];
205         tmp[1] = elements[i+nE];
206         tmp[2] = elements[i+2*nE];
207
208         if(l1<l2) {
209             if(l2<l3) {
210                 elements[i] = tmp[2];
211                 elements[i+nE] = tmp[0];
212                 elements[i+2*nE] = tmp[1];
213             }
214             // l1 < l2 && l2 > l3
215             else {
216                 elements[i] = tmp[1];
217                 elements[i+nE] = tmp[2];
218                 elements[i+2*nE] = tmp[0];
219             }
220         }
221         // l1 ≥ l2
222         else {
223             if(l1<l3) {
224                 elements[i] = tmp[2];
225                 elements[i+nE] = tmp[0];

```

```

226         elements[i+2*nE] = tmp[1];
227     }
228 }
229 }
230 }
231
232 double square(double x) {
233     return x*x;
234 }

```

Listing B.6: GCRNI.m

```

1  function [x x1 x2] = ...
2      GCRNI(coordinates,elements,boundary,boundary2elements,M)
3  %X[,X1,X2]=GCRNI(COORDINATES,ELEMENTS,BOUNDARY,BOUNDARY2ELEMENTS,M)
4  % Solves the magnetostatic problem using the ansatz of
5  % Garcia-Cervera & Roma, but computes the partial solution
6  % U2 with the nodal interpolant.
7  %
8  % INPUT:
9  %     COORDINATES ... field describing nodes
10 %     ELEMENTS ... elements described by their indices
11 %     BOUNDARY ... boundary elements described by their indices
12 %     BOUNDARY2ELEMENTS ... contains the indices of the elements that
13 %         share a boundary element, i.e.
14 %         BOUNDARY2ELEMENTS(I)=J means, that the I-th boundary element
15 %         lays on the J-th volume element.
16 %     M ... magnetization (function handle)
17 %
18 % OUTPUT:
19 %     X ... solution U = U1 + U2
20 % OPTIONAL OUTPUT:
21 %     X1 ... partial solution U1
22 %     X2 ... partial solution U2
23
24 %*** Number of nodes, elements and boundary elements
25 nC = size(coordinates,1);
26 nE = size(elements,1);
27 nB = size(boundary,1);
28
29 %*****
30 %*** STEP 1, compute -laplace(U1) = -div(M)
31 %***         U1 = 0 on Gamma
32 %*****
33
34 %*** First vertex of elements and corresponding edge vectors
35 c1 = coordinates(elements(:,1),:);
36 d21 = coordinates(elements(:,2),:) - c1;
37 d31 = coordinates(elements(:,3),:) - c1;
38
39 %*** Vector of element areas 4*|T|
40 area4 = 2*(d21(:,1).*d31(:,2)-d21(:,2).*d31(:,1));
41
42 %*** Assembly of stiffness matrix
43 a = (sum(d21.*d31,2)./area4)';
44 b = (sum(d31.*d31,2)./area4)';
45 c = (sum(d21.*d21,2)./area4)';
46 A = [-2*a+b+c;a-b;a-c;a-b;b;-a;a-c;-a;c];
47 I = reshape(elements(:,[1 2 3 1 2 3 1 2 3]),9*nE,1);
48 J = reshape(elements(:,[1 1 1 2 2 2 3 3 3]),9*nE,1);
49 A = sparse(I,J,A(:));
50
51 %*** |T|*grad(Vj) where Vj is hat function on T
52 grad1 = [d21(:,2)-d31(:,2) d31(:,1)-d21(:,1)]/2;
53 grad2 = [d31(:,2) -d31(:,1)]/2;

```

```

54 grad3 = [-d21(:,2) d21(:,1)]/2;
55
56 %*** Compute M(sT)
57 MsT = feval(M, c1+(d21+d31)/3);
58
59 %*** Assembly of right-hand side
60 b = [sum(grad1.*MsT,2) sum(grad2.*MsT,2) sum(grad3.*MsT,2)];
61 b = accumarray(elements(:), b(:), [nC, 1]);
62
63 %*** Computation of P1-FEM approximation
64 freenodes = nB+1:nC;
65 x1 = zeros(nC,1);
66 x1(freenodes) = A(freenodes,freenodes)\b(freenodes);
67
68 %*** Free memory
69 clear I J a b c A
70
71 %*****
72 %*** STEP 2, compute
73 %*** w_h = M_h.n - nbla(U1).n
74 %*****
75
76 %*** Vector of element volumes 2*|T|
77 area2 = area4(boundary2elements)/2;
78
79 %*** Elementwise gradient of U1
80 u21 = repmat( (x1(elements(boundary2elements,2))).
81 -x1(elements(boundary2elements,1)))./area2, 1,2);
82 u31 = repmat( (x1(elements(boundary2elements,3))).
83 -x1(elements(boundary2elements,1)))./area2, 1,2);
84
85 %*** Compute grad(U1)
86 gradh = (d31(boundary2elements,:).*u21 ...
87 - d21(boundary2elements,:).*u31)*[0 -1 ; 1 0];
88
89 %*** Compute projection of M on P^0
90 cb1 = coordinates(boundary(:,1),:);
91 cb2 = coordinates(boundary(:,2),:);
92 M_h = feval(M, (cb1+cb2)./2);
93
94 %*** Outer normal vector
95 nVec = [cb2(:,2)-cb1(:,2), cb1(:,1)-cb2(:,1)];
96 edgeLength = sqrt(sum(nVec.^2,2));
97 nVec = nVec./repmat(edgeLength,1,2);
98
99 %*** Compute w_h
100 w_h = sum( (M_h - gradh).*nVec, 2);
101
102 %*****
103 %*** STEP 3, compute (\widetilde{V} w_h)(z) z\in\KK
104 %*****
105 x2 = evaluateV(coordinates(1:nB,:), boundary, w_h, coordinates);
106
107 %*****
108 %*** STEP 4, obtain discrete solution u = u1 + u2
109 %*****
110 x = x1 + x2;

```

Listing B.7: FKNI.m

```

1 function [x x1 x2] = FKNI(coordinates,elements,boundary,M)
2 % [X,X1,X2]=GCRNI(COORDINATES,ELEMENTS,BOUNDARY,BOUNDARY2ELEMENTS,M)
3 % Solves the solution of the magnetostatic problem using the
4 % ansatz of Fredkin & Koehler, but computes the partial solution
5 % U2 with the nodal interpolant.

```

```

6 %
7 % INPUT:
8 % COORDINATES ... field describing nodes
9 % ELEMENTS ... elements described by their indices
10 % BOUNDARY ... boundary elements described by their indices
11 % M ... magnetization (function handle)
12 %
13 % OUTPUT:
14 % X ... solution  $U = U_1 + U_2$ 
15 % OPTIONAL OUTPUT:
16 % X1 ... partial solution  $U_1$ 
17 % X2 ... partial solution  $U_2$ 
18 %
19 %*** Number of nodes, elements and boundary elements
20 nC = size(coordinates,1);
21 nE = size(elements,1);
22 nB = size(boundary,1);
23 %
24 %*****
25 %*** STEP 1, compute  $-\text{laplace}(U_1) = -\text{div}(M)$ 
26 %***  $du_1/dn = M.n$  on Gamma
27 %*****
28 %
29 %*** First vertex of elements and corresponding edge vectors
30 c1 = coordinates(elements(:,1),:);
31 d21 = coordinates(elements(:,2),:) - c1;
32 d31 = coordinates(elements(:,3),:) - c1;
33 %
34 %*** Vector of element areas  $4*|T|$ 
35 area4 = 2*(d21(:,1).*d31(:,2)-d21(:,2).*d31(:,1)));
36 %
37 %*** Assembly of stiffness matrix
38 a = (sum(d21.*d31,2)./area4)';
39 b = (sum(d31.*d31,2)./area4)';
40 c = (sum(d21.*d21,2)./area4)';
41 A = [-2*a+b+c;a-b;a-c;a-b;b;-a;a-c;-a;c];
42 I = reshape(elements(:,[1 2 3 1 2 3 1 2 3]),9*nE,1);
43 J = reshape(elements(:,[1 1 1 2 2 2 3 3 3]),9*nE,1);
44 A = sparse(I,J,A(:));
45 %
46 %***  $|T|*\text{grad}(V_j)$  where  $V_j$  is hat function on T
47 grad1 = [d21(:,2)-d31(:,2) d31(:,1)-d21(:,1)]/2;
48 grad2 = [d31(:,2) -d31(:,1)]/2;
49 grad3 = [-d21(:,2) d21(:,1)]/2;
50 %
51 %*** Compute  $M(sT)$ 
52 MsT = feval(M, c1+(d21+d31)/3);
53 %
54 %*** Assembly of right-hand side
55 b = [sum(grad1.*MsT,2) sum(grad2.*MsT,2) sum(grad3.*MsT,2)];
56 b = accumarray(elements(:, b(:), [nC, 1]));
57 %
58 %*** Computation of P1-FEM approximation, Neumann problem
59 freenodes = 2:nC; % solve restricted system
60 x1 = zeros(nC,1);
61 x1(freenodes) = A(freenodes,freenodes)\b(freenodes);
62 %
63 %*** Free memory
64 clear I J a b c A
65 %
66 %*****
67 %*** STEP 3, compute  $(\widetilde{K} U_1)(z)$   $z \in \widetilde{K}$ 
68 %*****
69 x2 = evaluateK(coordinates(1:nB,:),boundary,x1(1:nB),coordinates);
70 %
71 %*** Compute edge length

```

```

72 cb1 = coordinates(boundary(:,1),:);
73 cb2 = coordinates(boundary(:,2),:);
74 edgeLength = sqrt( sum( (cb1-cb2).^2,2) );
75
76 %*** Compute node2edges
77 node2edges = zeros(nB,2);
78 node2edges(boundary(:,1),1) = (1:nB)';
79 node2edges(boundary(:,2),2) = (1:nB)';
80
81 %*** Direction vectors
82 t = (cb2-cb1)./repmat(edgeLength,1,2);
83 r1 = t(node2edges(:,2),:);
84 r2 = t(node2edges(:,1),:);
85
86 %*** Orientation, if det(x,y)>0 => positive rotational direction
87 orient = (r1(:,1).*r2(:,2)-r1(:,2).*r2(:,1)));
88 tmp = sum(r1.*r2,2);
89
90 %*** To avoid imaginary values
91 tmp(tmp>1) = 1;
92 tmp(tmp<-1) = -1;
93
94 %*** Inner angle
95 alpha = zeros(nB,1);
96 alpha(orient<=0) = pi+acos( tmp(orient<=0) );
97 alpha(orient>0) = pi-acos( tmp(orient>0) );
98
99 %*** Compute U2
100 x2(1:nB) = x2(1:nB) + (-1+alpha/(2*pi)).*x1(1:nB);
101
102 %*****
103 %*** STEP 4, obtain discrete solution U = U1 + U2
104 %*****
105 X = x1 + x2;

```

Literaturverzeichnis

- [1] ALT, H. W.: *Lineare Funktionalanalysis. Eine anwendungsorientierte Einführung*. Springer Verlag, 2006
- [2] AURADA, M. ; EBNER, M. ; FERRAZ-LEITE, S. ; GOLDENITS, P. ; KARKULIK, M. ; MAYR, M. ; PRAETORIUS, D.: *HILBERT - A Matlab Implementation of Adaptive BEM*. Vienna University of Technology: Institute for Analysis and Scientific Computing, 2009
- [3] BRAESS, D.: *Finite Elemente. Theorie, schnelle Löser und Anwendungen in der Elastizitätstheorie*. Springer Verlag, 2007
- [4] CARSTENSEN, C. ; STEPHAN, E. C.: Adaptive Coupling of Boundary Elements and Finite Elements. In: *Mathematical Modelling and Numerical Analysis* 29 (1995)
- [5] COSTABEL, M.: A Symmetric Method for the Coupling of Finite Elements and Boundary Elements. In: *The Mathematics of Finite Elements and Applications IV, MAFELAP 1987*. London : Academic Press, 1988
- [6] EVANS, Lawrence C.: *Partial Differential Equations*. Providence, Rhode Island : American Mathematical Society, 2002
- [7] FEISCHL, M.: *Different Strategies for the Adaptive FEM-BEM Coupling*. 2010
- [8] FREDKIN, D.R. ; KOEHLER, T.R.: Hybrid Method for Computing Demagnetizing Fields. In: *IEEE Transactions on Magnetics* 26 (1990)
- [9] FUNKEN, S. ; PRAETORIUS, D. ; WISSGOTT, P.: Efficient Implementation of Adaptive P1-FEM in Matlab. In: *ASC Report* 19 (2008). – Institute for Analysis and Scientific Computing, Vienna University of Technology.
- [10] GARCÍA-CERVERA, C. J. ; ROMA, A. M.: Adaptive Mesh Refinement for Micromagnetics Simulations. In: *IEEE Transactions on Magnetics* 2005 (2005)
- [11] GIRAULT, V. ; RAVIART, P.-A.: *Finite Element Methods for Navier-Stokes Equations. Theory and Algorithms*. Springer Verlag, 1986
- [12] HEUSER, H.: *Lehrbuch der Analysis, Teil 2*. Stuttgart, Leipzig, Wiesbaden : B.G. Teubner Verlag, 2004
- [13] JACKSON, J. D.: *Klassische Elektrodynamik*. Walter de Gruyter, 2006
- [14] JOHNSON, J. ; NÉDÉLEC, J.C.: On the Coupling of Boundary Integral and Finite Element Methods. In: *Math. Comp.* 35 (1980)
- [15] MCLEAN, W.: *Strongly Elliptic Systems and Boundary Integral Equations*. Stuttgart, Leipzig, Wiesbaden : Cambridge University Press, 2000
- [16] PRAETORIUS, D.: *Introduction to Boundary Element Method. Lecture Notes*. 2007
- [17] PRAETORIUS, D.: *Finite Element Method. Lecture Notes*. 2008/2009

- [18] SAUTER, S. ; SCHWAB, C.: *Randelementmethoden. Analyse, Numerik und Implementierung schneller Algorithmen*. Stuttgart, Leipzig, Wiesbaden : B.G. Teubner Verlag, 2004
- [19] SCOTT, R. L. ; ZHANG, S.: Finite Element Interpolation of Nonsmooth Functions Satisfying Boundary Conditions. In: *Mathematics of Computation* 54 (1990)
- [20] STEINBACH, O.: *Numerische Näherungsverfahren für elliptische Randwertprobleme*. Stuttgart, Leipzig, Wiesbaden : B.G. Teubner Verlag, 2003
- [21] TESCHL, G.: *Mathematical Methods in Quantum Mechanics. With Applications to Schrödinger Operators*. American Mathematical Society, 2009
- [22] VERFÜRTH, R.: *A Review of A Posteriori Error Estimation and Adaptive Mesh-Refinement Techniques*. Chichester, New York, Brisbane, Toronto, Singapore, Stuttgart, Leipzig : Wiley-Teubner, 1996