

Diese Dissertation haben begutachtet:

Jens Markus Melenk

Sören Bartels



D I S S E R T A T I O N

Quadratic minimization with non-local operators and non-linear constraints

A case study in thin-film micromagnetics

ausgeführt zum Zwecke der Erlangung des akademischen Grades eines
Doktors der technischen Wissenschaften unter der Leitung von

Univ.Prof. Jens Markus Melenk, PhD und
Ao.Univ.Prof. Dipl.Math. Dr.techn. Dirk Praetorius
E101, Institut für Analysis und Scientific Computing

eingereicht an der Technischen Universität Wien
Fakultät für Mathematik und Geoinformation

von

Dipl.-Ing. Samuel Ferraz-Leite

Matrikelnummer: 0025356

Thürnlhofstraße 9/1/12, 1110 Wien

Wien, Jänner 2011

Dedicated to my beloved wife Bianca,
my source of strength and inspiration.

Kurzfassung

In physikalischen Problemstellungen wird das Verhalten komplexer Systeme oft beschrieben, indem jedem sinnvollen Zustand des Systems eine Energie zugeordnet wird. Die (meta-) stabilen Zustände sind dann gerade dadurch charakterisiert, dass sie die Energie (lokal) minimieren. Mathematisch lautet die Aufgabe also, ein gegebenes Energiefunktional $e : \mathcal{A} \rightarrow \mathbb{R}$ zu minimieren. Dabei ist \mathcal{A} die durch Nebenbedingungen festgelegte Teilmenge der zulässigen physikalisch sinnvollen Konstellationen aus einem geeigneten Funktionenraum $\mathcal{A} \subseteq \mathcal{H}$.

Die vorliegende Arbeit befasst sich mit einem Reduktionsmodell aus [DKM⁺01], welches das Verhalten uniaxialer dünner Filme im Mikromagnetismus beschreibt. Sei $\omega \subseteq \mathbb{R}^2$ das ferromagnetische Material dessen Dicke $t \ll \text{diam}(\omega)$ in der Modellierung vernachlässigt wird. Mögliche Magnetisierungen werden durch Vektorfelder $\mathbf{m} : \omega \rightarrow \mathbb{R}^2$ beschrieben, die der physikalischen Nebenbedingung $|\mathbf{m}(x)| \leq 1$ genügen. Für einen möglichen Zustand \mathbf{m} des Ferromagneten unter Einfluss eines äußeren Feldes \mathbf{f} lautet die Energie

$$e(\mathbf{m}) = \int_{\mathbb{R}^3} |\nabla u|^2 dx + q \int_{\omega} \mathbf{m}_2^2 dx - 2 \int_{\omega} \mathbf{f} \cdot \mathbf{m} dx,$$

wobei $u : \mathbb{R}^3 \rightarrow \mathbb{R}$ das von \mathbf{m} induzierte magnetostatische Potential ist und als Lösung einer reduzierten Maxwell-Gleichung bestimmt ist. Der Parameter $q \geq 0$ ist vom Material abhängig und bestimmt die Relevanz der kristallinen Anisotropie.

Dieses Modellproblem dient als Prototyp für ein vektorwertiges quadratisches Minimierungsproblem. Als schwierig erweist sich das Vorhandensein nichtlokaler Beiträge durch die Streufeldenergie $\int_{\mathbb{R}^3} |\nabla u|^2 dx$ sowie die nichtlineare skalare, jedoch konvexe Nebenbedingung $|\mathbf{m}(x)| \leq 1$. In der Dissertation [Drw08] behält der Autor, wie dies auch in der Originalarbeit [DKMO02] geschieht, eine distributionelle Sichtweise bei. Der Autor diskutiert die Simulation des Spezialfalls weichmagnetischen Materials mit $q = 0$ mittels eines Innere-Punkte-Verfahrens. Dabei nimmt die effiziente Berechnung der Streufeldenergie $\int_{\mathbb{R}^3} |\nabla u|^2 dx$ einen wesentlichen Raum ein.

Der erste Teil der vorliegenden Arbeit befasst sich mit der Konstruktion eines Hilbertraums \mathcal{H} für die Magnetisierung \mathbf{m} . Lösbarkeit und Eindeutigkeit werden in geeignetem Sinn nachgewiesen. Dabei ist das Eindeutigkeitsresultat weder in den Originalpublikationen [DKM⁺01, DKMO02] noch in der Dissertation [Drw08] enthalten.

Der zweite Teil befasst sich mit der Umsetzung eines Strafverfahrens für die numerische Approximation des analytischen Minimierers. Dabei muss der Hilbertraum \mathcal{H} geeignet durch einen endlichdimensionalen Ansatzraum X_h approximiert werden. Dadurch entsteht ein vom Parameter $h > 0$ abhängiger Diskretisierungsfehler. Andererseits wird das Strafverfahren durch einen Parameter $\varepsilon > 0$ gesteuert, der einen Modellierungsfehler bedingt. Als Kernresultat wird bewiesen, dass beliebige Folgen $(h_n, \varepsilon_n) \rightarrow (0, 0)$ zu Konvergenz der Minimierer in geeignetem Sinn führen. Unter Regularitätsannahmen und aufbauend auf punktweise hergeleiteten Euler-Lagrange Gleichungen werden a priori Abschätzungen bewiesen, die darüber hinaus Auskunft über die zumindest zu erwartende Konvergenzordnung geben. Schließlich wird ein heuristischer Fehlerschätzer vorgeschlagen, der geeignet ist, um ein h - und ε -adaptives Verfahren zu steuern.

Der dritte und vierte Teil des Dissertation befragen sich mit einer detaillierten Beschreibung einer objektorientierten Implementierung des Strafverfahrens in C++ sowie mit ausführlichen numerischen Experimenten, um die analytischen Aussagen zu überprüfen. In Beispielen mit starken Singularitäten und im Falle weicher Filme mit $q = 0$ führt der Einsatz von Adaptivität zu einer Verbesserung der Asymptotik.

Abstract

In physics, the behavior of complex systems is often described in terms of an energy associated with each possible state. The (meta-) stable states of the system, then, are characterized by minimizing the energy (locally). Mathematically, the task of finding stable states, therefore, is the minimization of an energy functional $e : \mathcal{A} \rightarrow \mathbb{R}$. Here, \mathcal{A} denotes the subset of admissible states in some appropriate function space $\mathcal{A} \subseteq \mathcal{H}$.

The present work treats the numerical analysis of a reduced model from [DKM⁺01] that describes the behavior of thin-film devices in micromagnetics. Let $\omega \subseteq \mathbb{R}^2$ denote the ferromagnetic sample whose thickness $t \ll \text{diam}(\omega)$ is neglected by the model. Possible states of the magnetic device are described by a vector field $\mathbf{m} : \omega \rightarrow \mathbb{R}^2$ under the physical side constraint $|\mathbf{m}(x)| \leq 1$. Given a magnetization \mathbf{m} and an applied field \mathbf{f} , the energy reads

$$e(\mathbf{m}) = \int_{\mathbb{R}^3} |\nabla u|^2 dx + q \int_{\omega} \mathbf{m}_2^2 dx - 2 \int_{\omega} \mathbf{f} \cdot \mathbf{m} dx,$$

where the induced magnetostatic potential $u : \mathbb{R}^3 \rightarrow \mathbb{R}$ is determined as solution of a reduced Maxwell equation. The material dependent parameter $q \geq 0$ describes the strength of the crystalline anisotropy.

This model problem serves as a prototype for a vector valued quadratic minimization problem. The non-local contributions of the stray field energy $\int_{\mathbb{R}^3} |\nabla u|^2 dx$ and the non-linear, but convex scalar side constraint $|\mathbf{m}(x)| \leq 1$ effect difficulties in the treatment of the equations. The dissertation [Drw08] has a distributional point of view, as is also the case in the original work [DKMO02]. The work [Drw08] is concerned with the numerical simulation of the model in the case of soft ferromagnetic samples, i.e. $q = 0$, by use of an interior point method. A large part of that thesis is concerned with the efficient computation of the stray-field energy $\int_{\mathbb{R}^3} |\nabla u|^2 dx$.

The first part of the present work is concerned with the construction of a suitable Hilbert space \mathcal{H} for the magnetization. Existence and uniqueness of solutions are proven in an appropriate sense. We stress that the statement on the uniqueness is neither included in the original works [DKM⁺01, DKMO02] nor in the more recent dissertation [Drw08].

The second part studies a penalty method for the numerical solution of the model problem. First, the energy space \mathcal{H} needs to be approximated by some finite-dimensional subspace $X_h \subseteq \mathcal{H}$. This introduces a discretization error dependent on $h > 0$. Second, the penalty scheme depends on a parameter $\varepsilon > 0$ which introduces a modeling error. As a key result we prove that any choice of zero sequences $(h_n, \varepsilon_n) \rightarrow (0, 0)$ leads to convergence of the discrete and penalized solutions to the analytical solution in an appropriate sense. Under regularity assumptions and based on pointwise Euler-Lagrange equations, we also provide an a priori error analysis that gives indication on what order of convergence may be expected at least in numerical simulations. Finally, a heuristic error estimator is proposed that is suitable to steer an h - and ε -adaptive algorithm.

Parts three and four describes in detail the object-oriented implementation of the proposed scheme in C++ and extensive numerical experiments are provided to verify our analysis. In the case of soft films with $q = 0$ or in presence of strong singularities, the proposed adaptive scheme improves the asymptotic behavior when compared to a uniform approach.

Contents

1	Introduction and overview	1
1.1	Model problem	1
1.2	Overview and main results	2
1.3	Acknowledgement	8
2	A model problem in thin-film micromagnetics	9
2.1	The magnetic free energy	9
2.2	A reduced model in thin-film micromagnetics	11
2.3	Sobolev spaces	13
2.4	The static Maxwell equation in thin-film micromagnetics	15
2.4.1	Strong form of the Maxwell equation	15
2.4.2	Newtonian and simple-layer potential	16
2.4.3	Solution of the magnetostatic Maxwell equation	17
2.5	The energy spaces \mathcal{H} and $B_1^2(\mathbb{R}^3)$	18
2.5.1	The energy space \mathcal{H} for the magnetization	19
2.5.2	The Beppo-Levi class $B_1^2(\mathbb{R}^3)$ for the magnetostatic potential u	21
2.6	Well posedness of the thin-film minimization problem	21
2.6.1	Existence and uniqueness of the magnetostatic potential	21
2.6.2	Existence and uniqueness of minimizer $\mathbf{m}^* \in \mathcal{H}$	23
3	Discretization, numerical solution, and error analysis	27
3.1	Discretization and penalization	28
3.1.1	The space of lowest-order Raviart-Thomas finite elements	29
3.1.2	The Discrete Minimization Problem (M_h)	31

3.1.3	The (discrete) penalized minimization problem (M_h^ε)	34
3.2	Convergence	35
3.3	A priori error estimates	39
3.3.1	The Euler-Lagrange equations (KKT conditions)	39
3.3.2	A priori analysis	45
3.4	Adaptive algorithm	48
3.4.1	Error estimation by space enrichment	49
3.4.2	A heuristic adaptive algorithm	52
4	Implementation	55
4.1	An object oriented implementation of meshes	56
4.1.1	Geometric base classes	56
4.1.2	The Mesh class	59
4.2	Finite element spaces	64
4.2.1	Basis functions	64
4.2.2	Finite element spaces	65
4.3	A damped Newton algorithm	68
4.3.1	Quadrature rules	71
4.3.2	Sparse matrix representation and HLib	73
4.3.3	Code listing	76
5	Numerical experiments	81
5.1	Overview and general remarks	81
5.2	Experimental analysis of discretization error	85
5.2.1	Soft applied field $\mathbf{f} = (0.1, 0.1)^T$	85
5.2.2	Active penalization with $\mathbf{f} = (0.6, 0.6)^T$	93
5.2.3	Non-smooth solution and h -adaptive algorithm	96
5.2.4	Influence of ε	105
5.3	Experimental analysis of penalty error	110
5.3.1	Convergence and influence of h	110
5.3.2	Empirical choice of $\varepsilon = h^\alpha$	115

5.4	A simulation with non-smooth data	121
5.5	Hard and soft material	126
	List of Figures	143
	List of Tables	149
	Bibliography	153
	Curriculum vitae	157

Chapter 1

Introduction and overview

1.1 Model problem

In physics, the behavior of complex systems is often described in terms of energy functionals. The (meta-) stable states of the system, then, are characterized by minimizing the energy (locally). Mathematically, the task of finding stable states, therefore, is the minimization of an energy functional $e : \mathcal{A} \rightarrow \mathbb{R}$, where \mathcal{A} denotes the subset of admissible states in some appropriate function space $\mathcal{A} \subseteq \mathcal{H}$.

The field of computational micromagnetics is of special mathematical and scientific interest for several reasons. The model due to Landau and Lifshits [LL35] is nowadays accepted as the relevant model to describe micromagnetic phenomena in many applications. The energy functional, however, is not convex and hysteresis governs many of the non-linear effects that are observed in experiments. The presence of various length scales make large magnetic devices hardly accessible for direct numerical simulation. On the other hand, micromagnetic phenomena are not yet fully understood [SH98], and numerical simulation has proven over the last decades to be a valuable tool to understanding physical phenomena in many applications.

To allow for efficient simulation of mesoscopic effects, various reduced models have been proposed and analyzed in the literature, cf. e.g. [DKMO06]. Each of which covers a certain limiting regime of samples. In [DKM⁺01], a reduced model in thin-film micromagnetics has been proposed and it is observed that it is compatible with the other prior models suggested in [SB89] and [vdB86].

Let $\omega \subseteq \mathbb{R}^2$ denote a ferromagnetic sample, whose thickness $t \ll \text{diam}(\omega)$ is neglected. Admissible magnetizations are described by vector fields $\mathbf{m} : \omega \rightarrow \mathbb{R}^2$ such that the constraint $|\mathbf{m}(x)| \leq 1$ is satisfied pointwise almost everywhere in ω . Given an applied field \mathbf{f} , the reduced energy reads

$$e(\mathbf{m}) = \frac{1}{2} \int_{\mathbb{R}^3} |\nabla u|^2 dx + \frac{q}{2} \int_{\omega} \mathbf{m}_2^2 dx - \int_{\omega} \mathbf{f} \cdot \mathbf{m} dx, \quad (1.1)$$

where the magnetostatic potential u solves the reduced Maxwell equation

$$\int_{\mathbb{R}^3} \nabla u \nabla v dX = \int_{\omega} \mathbf{m} \cdot \nabla v dx \quad \forall v \in \mathcal{D}(\mathbb{R}^3).$$

The material parameter $q > 0$ determines the strength of the uniaxial crystalline anisotropy. In contrast to the full micromagnetic model problem due to Landau and Lifshits, the reduced thin-film minimization problem is convex. However, it is a vector valued problem with a scalar, non-linear but

convex side constraint $|\mathbf{m}(x)| \leq 1$. In [CP01] and [Pra03], a penalty method for the simulation of the large-body limit in micromagnetics is studied. The similar nature of the model problem may indicate that the techniques used in those publications might be generalized and applied to the thin-film problem as well. However, there are some differences that need new techniques and lead to somewhat altered results: First, the presence of a negative order Sobolev norm which is a result of the limiting process of the stray field energy as $t \rightarrow 0$. In particular, in the thin-film model under consideration, the dual space contains not only Lebesgue functions but also distributions. Therefore, arguments based on properties of the function space L^2 often cannot be transferred easily. Second, the existence of the magnetostatic potential demands $\nabla \cdot \mathbf{m} \in \tilde{H}^{-1/2}(\omega)$. This means that a discretization \mathbf{m}_h of the magnetization must allow for evaluation of the divergence $\nabla \cdot \mathbf{m}_h$. The use of Raviart-Thomas finite elements [RT77] effects that certain L^2 -orthogonalities used in the a priori and a posteriori analysis of the large-body limit – where piecewise constant functions are used to discretize $\mathbf{m} \in L^2$ – are not available.

In the work [DKMO02], Γ -convergence [Bra02] of the full model by Landau and Lifshits to the reduced thin-film model has been proven under certain assumptions. This means that one cannot expect to obtain full information on the microstructure when simulating the thin-film model. Mesoscopic and macroscopic quantities, such as the magnetostatic potential and the energy, are however preserved. Moreover, a comparison of first numerical simulations with data obtained from experiments shows that the model is of relevance, at least for soft ferromagnetic samples. The recent dissertation [Drw08] is dedicated to the case of soft material, and the anisotropy is dropped, i.e. $q = 0$. Numerical calculation of minimizers is done by use of an interior point method. Special emphasis is placed on the efficient computation of the stray field energy $\frac{1}{2} \int_{\mathbb{R}^3} |\nabla u|^2 dx$ by use of \mathcal{H}^2 matrices [HB02, DO04] and the computation of minimizers of unit length. Let \mathbf{m} be a minimizer of the reduced energy (1.1). Note that the anisotropy energy contribution vanishes due to $q = 0$. Choosing some divergence free magnetization $\tilde{\mathbf{m}}$ with $\int_{\omega} \tilde{\mathbf{m}} \cdot \mathbf{f} dx = 0$ and $|\mathbf{m}(x) + \tilde{\mathbf{m}}(x)| \leq 1$ yields another minimizer. The computation of minimizers of unit length allows, under certain conditions, for reconstruction of the magnetic domains and hence restoring part of the microstructural data that is originally lost in the process of the Γ -limit.

1.2 Overview and main results

Chapter 2 is dedicated to the construction of a Hilbert space \mathcal{H} for the magnetization and the analysis of the model problem. First, in Section 2.1, we give an interpretation of the energy contributions of the full micromagnetic model by Landau and Lifshits. Then, Section 2.2 introduces the minimization problem (M), where the energy functional is defined as in (1.1).

As a first step in the analysis of the model problem, we turn our attention to the reduced magnetostatic Maxwell equation

$$\int_{\mathbb{R}^3} \nabla u \nabla v dX = \int_{\omega} \mathbf{m} \cdot \nabla v dx \quad \forall v \in \mathcal{D}(\mathbb{R}^3). \quad (1.2)$$

In Lemma 2.3, direct calculations and integration by parts yield

$$\begin{aligned} -\Delta u &= 0 && \text{in } \mathbb{R}^3 \setminus \bar{\omega} \\ \left[\frac{\partial u}{\partial x_3} \right] &= \nabla \cdot \mathbf{m} && \text{on } \omega \\ [u] &= 0 && \text{on } \omega \\ \mathbf{m} \cdot \mathbf{n} &= 0 && \text{on } \gamma = \partial\omega. \end{aligned}$$

Here, $[\cdot]$ denotes the jump $\lim_{x_3 \rightarrow 0^+} u(x) - \lim_{x_3 \rightarrow 0^-} u(x)$ across the surface ω and \mathbf{n} is the outer normal vector of ω in \mathbb{R}^2 . The simple-layer potential $S : \tilde{H}^{-1/2}(\omega) \rightarrow H_{loc}^1(\mathbb{R}^3)$ defined by

$$Sv(x) = \frac{1}{4\pi} \int_{\omega} \frac{v(y)}{\|x - y\|} ds_y$$

is used to obtain a solution of the Maxwell equation. Theorem 2.8 uses ideas from [Ste87] for the analysis of screen problems to generalize the well-known jump conditions of the simple-layer potential in the case of closed surfaces. We obtain

$$[S\varphi] = 0 \quad \left[\frac{\partial S\varphi}{\partial x_3} \right] = -\varphi$$

from which we conclude that $S(-\nabla \cdot \mathbf{m})$ is a solution of (1.2). In particular the magnetostatic Maxwell equation does have a solution if the divergence of the magnetization satisfies $\nabla \cdot \mathbf{m} \in \tilde{H}^{-1/2}(\omega)$.

This leads to the choice of our energy space

$$\mathcal{H} := \{\mathbf{m} \in L^2(\omega)^2 \mid \nabla \cdot \mathbf{m} \in \tilde{H}^{-1/2}(\omega), \mathbf{m} \cdot \mathbf{n} = 0 \text{ on } \gamma\},$$

which is a Hilbert space with the natural norm

$$\|\mathbf{m}\|^2 = \|\mathbf{m}\|_{L^2}^2 + \|\nabla \cdot \mathbf{m}\|_{\tilde{H}^{-1/2}}^2.$$

Let $V : \tilde{H}^{-1/2}(\omega) \rightarrow H^{1/2}(\omega)$ denote the trace of the simple-layer potential S . Then, V is an elliptic and continuous linear operator, and the V -norm defined by

$$\|\varphi\|_V^2 = \langle \varphi, V\varphi \rangle_{\tilde{H}^{-1/2} \times H^{1/2}}$$

is an equivalent norm in $\tilde{H}^{-1/2}(\omega)$. Theorem 2.18, finally, states that given any $\mathbf{m} \in \mathcal{H}$, there is a uniquely determined potential $u \in B_1^2(\mathbb{R}^3) := \{u \in H_{loc}^1(\mathbb{R}^3) \mid \nabla u \in L^2(\mathbb{R}^3)\} / \mathbb{R}$, and moreover it holds that

$$\|\nabla u\|_{L^2(\mathbb{R}^3)}^2 = \|\nabla \cdot \mathbf{m}\|_V^2.$$

This last identity allows for reformulation of our energy functional

$$e(\mathbf{m}) = \frac{1}{2} \|\nabla \cdot \mathbf{m}\|_V^2 + \frac{q}{2} \|\mathbf{m}_2\|_{L^2}^2 - (f, \mathbf{m})_{L^2} \quad (1.3)$$

as was also observed in [DKMO02]. Note that $e(\cdot)$ is a convex and continuous mapping with respect to the energy space \mathcal{H} .

In Section 2.6.2, we use the direct method of calculus of variations [Dac89] to prove existence and uniqueness of a minimizer \mathbf{m}^* within the set of admissible magnetizations $\mathcal{A} := \{\mathbf{m} \in \mathcal{H} \mid |\mathbf{m}(x)| \leq 1\}$. In Lemma 2.20, we prove that the set \mathcal{A} is closed with respect to the \mathcal{H} -norm. Since it is also convex, Mazur's lemma [Yos65] implies that it is also closed with respect to the weak topology. In Theorem 2.21, we prove coercivity of $e(\cdot)$. An infimizing sequence hence has a weakly convergent subsequence, and the weak limit is an element of the admissible set \mathcal{A} . Since $e(\cdot)$ is continuous and convex it is also weakly lower semi-continuous and we conclude that the weak limit is a minimizer. In Theorem 2.22 we use mollifier techniques inspired by [Pra03] to prove that

$$\|\mathbf{m}\|^2 := \|\nabla \cdot \mathbf{m}\|_V^2 + q \|\mathbf{m}_2\|_{L^2}^2 \quad (1.4)$$

defines a norm, from which we obtain uniqueness of the minimizer, provided $q > 0$. This last result is not stated in any of the publications [DKMO02, Drw08]. This may be because the authors there have a distributional point of view and do not define a Hilbert space setting. Also the focus in the original publications lies on soft material, where the anisotropy $q \ll 1$ is dropped. Then, the solution is in fact not uniquely determined. After stating the corresponding variational inequality in Lemma 3.3, we immediately obtain the continuous dependence of \mathbf{m} from the data \mathbf{f} by standard arguments. Altogether this finally provides well-posedness of the minimization problem (M) within our Hilbert space setting.

In Chapter 3, we propose a scheme for the computation of approximations to the analytical solution \mathbf{m}^* . First, we use lowest order Raviart-Thomas finite elements to discretize the energy space \mathcal{H} . This is also proposed in the work [Drw08]. The approximation of the infinite dimensional Hilbert space \mathcal{H} by a finite dimensional subspace X_h introduces a discretization error that depends on the mesh-size $h > 0$. The fact that we choose a conforming discretization allows us to apply the same arguments as for the continuous problem to prove existence and uniqueness of solutions of the discrete problem (M_h) . With the help of the equivalent variational inequality, we obtain the a priori error estimate

$$\|\mathbf{m}^* - \mathbf{m}_h\| = \mathcal{O}(h^{1/2})$$

under certain regularity assumptions.

The side constraint $|\mathbf{m}(x)| \leq 1$ is treated by use of a penalty method in the spirit of [CP01]: For $\varepsilon > 0$, we introduce the energy contribution

$$\frac{1}{2\varepsilon} \|(|\mathbf{m}| - 1)_+\|_{L^2}^2,$$

where $(\cdot)_+$ denotes the positive part. Lemma 3.7 provides the coercivity of the modified energy functional

$$e_\varepsilon(\mathbf{m}) := e(\mathbf{m}) + \frac{1}{2\varepsilon} \|(|\mathbf{m}| - 1)_+\|_{L^2}^2.$$

We drop the side constraint and seek the uniquely determined solution \mathbf{m}_h^ε of the unconstrained but non-smooth minimization problem (M_h^ε) associated with the penalized energy functional $e_\varepsilon(\cdot)$. The penalty scheme thus introduces an additional error dependent of $\varepsilon > 0$.

In the literature, penalty methods have been studied in depth for finite dimensional problems. Let \mathbf{m}_h^ε be the solution to the problem (M_h^ε) and \mathbf{m}_h the solution to the problem (M_h) . Then it is well-known, see e.g. [NW99], that one can expect weak convergence

$$\mathbf{m}_h^\varepsilon \rightharpoonup \mathbf{m}_h \quad \text{as } \varepsilon \rightarrow 0.$$

This, however, only guarantees convergence for fixed discretization parameter $h \geq 0$. We failed to find any convergence result in the literature that could be applied to our model problem to provide convergence as $h \rightarrow 0$ and $\varepsilon \rightarrow 0$. In [CP01], the authors succeed to prove convergence in the energy norm. The proof, however, relies on the fact that the energy space is L^2 and that the dual space thus consist of Lebesgue functions. In contrast, this is not the case in our context. In Theorem 3.11, we use abstract arguments to prove that also $\mathbf{m}_h^\varepsilon \rightharpoonup \mathbf{m}_0^\varepsilon$ as $h \rightarrow 0$. Finally, Corollary 3.12 states weak convergence $\mathbf{m}_h^\varepsilon \rightharpoonup \mathbf{m}^*$ for any choice of $h \rightarrow 0$ and $\varepsilon \rightarrow 0$. We stress that the arguments are largely independent of the concrete model problem and apply to a large class of energy functionals.

The scheme for the numerical computation, now, is justified in the sense that, as $(h_n, \varepsilon_n) \rightarrow (0, 0)$, we obtain weak convergence of the discrete solutions to the analytical solution of the constrained problem

(M). Section 3.3, is dedicated to establish a priori error estimates. Based on a regularity result of [DKMO02], we derive KKT-equations and prove the existence of a Lagrange multiplier $\lambda \in L^2_{loc}(\omega)$ such that

$$\begin{aligned} q \begin{pmatrix} 0 \\ \mathbf{m}_2 \end{pmatrix} + \nabla u - f + \lambda \mathbf{m} &= 0, \\ \lambda(|\mathbf{m}| - 1) &= 0, \\ \lambda &\geq 0, \\ |\mathbf{m}| &\leq 1 \end{aligned}$$

hold pointwise almost everywhere in ω . The proof of Theorem 3.18 follows the same ideas as in case of the large body limit [De 93]. However, the fact that we are dealing with functions in L^2_{loc} effects that the proof of [De 93] cannot be applied to our setting directly, and further technical details are necessary.

Based on these pointwise equations and under the additional regularity assumption $\nabla \cdot \mathbf{m} \in L^2$, Theorem 3.22 provides the a priori estimate

$$\|\mathbf{m}^* - \mathbf{m}_0^\varepsilon\| \lesssim \mathcal{O}(\varepsilon^{1/2}).$$

In the proof, we use a pointwise estimate for the non-linear contributions that stems from [CP01]. Finally, Theorem 3.24 provides an a priori estimate that yields

$$\|\mathbf{m}^* - \mathbf{m}_h^\varepsilon\| \lesssim \mathcal{O}(h^{1/2} + \varepsilon^{1/2})$$

as stated in Corollary 3.25 under suitable assumptions.

In Section 3.4, we propose a heuristic error estimation strategy: As stated in [Drw08], in the case of $q = 0$ and for constant applied field with $|\mathbf{f}| \ll 1$, the constraint is not active and the minimization problem is equivalent to Symm's integral equation

$$V\phi = \mathbf{f} \cdot x$$

where $\phi = \nabla \cdot \mathbf{m}$. The solutions to such screen problems reveal generic singularities along the edges of ω [ESAES90]. For this linear case, an $h - h/2$ based error estimation strategy has been analyzed recently, cf. [FLP08, FLOP10]. We apply the ideas of [FLP08] to our minimization problem (M_h^ε): Let \mathcal{T}_ℓ denote some triangulation of ω with associated discrete space X_ℓ . The uniform refinement of \mathcal{T}_ℓ is denoted by $\widehat{\mathcal{T}}_\ell$, and \widehat{X}_ℓ is the corresponding discrete space. Let $\varepsilon > 0$ be fixed and let \mathbf{m}_ℓ and $\widehat{\mathbf{m}}_\ell$ be the solutions of (M_h^ε) with respect to the discrete spaces X_ℓ and \widehat{X}_ℓ , respectively. Then, the quantity

$$\eta_\ell^H = \|\widehat{\mathbf{m}}_\ell - \mathbf{m}_\ell\|,$$

is used to estimate the discretization error. In Theorem 3.26, we prove that the saturation assumption

$$\|\mathbf{m}_0^\varepsilon - \widehat{\mathbf{m}}_\ell\| \leq C_{sat} \|\mathbf{m}_0^\varepsilon - \mathbf{m}_\ell\| \text{ with } \ell\text{-independent } C_{sat} \in (0, 1)$$

implies efficiency and reliability of the error estimator

$$\eta_\ell^H \lesssim \|\mathbf{m}_0^\varepsilon - \mathbf{m}_\ell\| \lesssim \eta_\ell^H.$$

Based on a local inverse estimate from [GHS05], we finally provide local error indicators $\mu_\ell^H(T)$ to steer an h -adaptive algorithm. Based on the empirical observation that $\frac{1}{2\varepsilon} \|(|\mathbf{m}_h^\varepsilon| - 1)_+\|_{L^2(\omega)}^2 \rightarrow 0$ as $\varepsilon \rightarrow 0$, we propose the use of

$$\mu_\ell^\varepsilon(T)^2 = \frac{1}{2\varepsilon} \|(|\mathbf{m}_h^\varepsilon| - 1)_+\|_{L^2(T)}^2$$

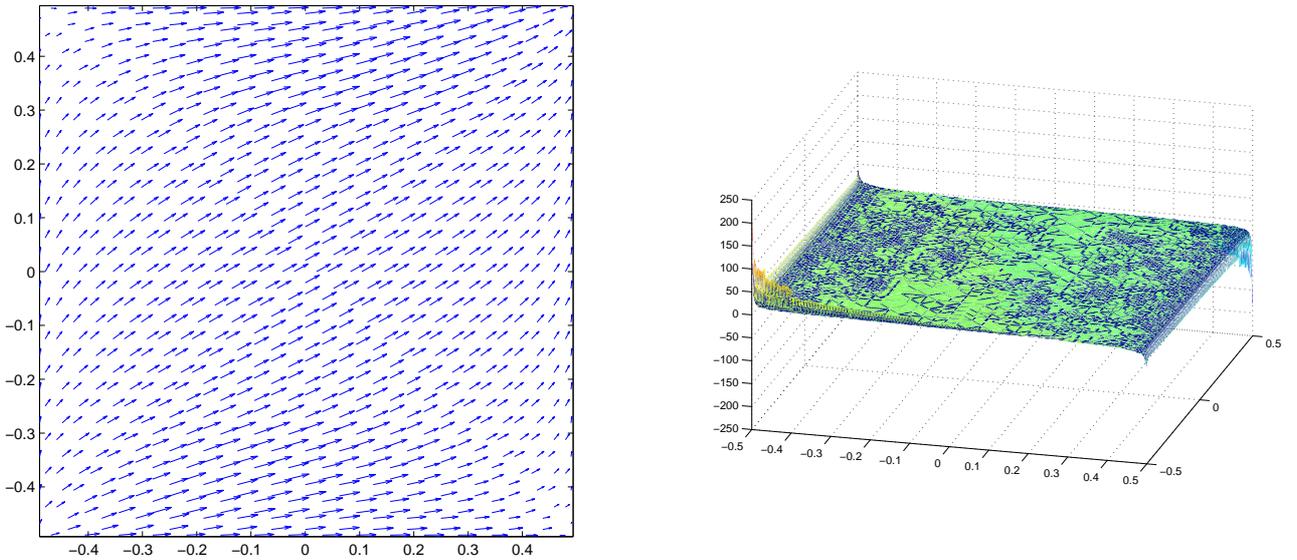


Figure 1.1: Typical solution of a problem with smooth data; left: Magnetization, right: Divergence

for steering adaptive refinements of the penalty parameter ε .

Chapter 4, reports on the implementation of the numerical scheme in C++. In Section 4.1, we describe an object oriented implementation of regular adaptive meshes. Some of the ideas are inspired by [Sha09] but our implementation differs significantly from the solution proposed in the mentioned publication. We stress that mesh administration usually is neither a bottleneck with respect to computation times nor with respect to memory consumption in numerical simulations. Therefore, we did not aim at providing a fully optimized and efficient implementation. Instead we focused on aspects such as maintainability, extensibility, and readability of the codes. All mesh routines, however, have at most a complexity of $\mathcal{O}(N \log N)$, which is asymptotically optimal.

Section 4.2, describes how the Mesh-class can be extended to represent a finite element space. In contrast to the mesh, where all quantities are stored in list containers, we store the basis functions in a vector to provide access in constant time by the index of the functions. Usually, basis functions are associated with some geometric quantity. Fast access $\mathcal{O}(\log N)$ via the geometric information seems not to be possible using the `std::map` class. Instead, we implemented our own sorted `vector<pair>` container that ensures $\mathcal{O}(N \log N)$ time for building and $\mathcal{O}(\log N)$ for access.

Finally, Section 4.3 describes how the developed finite element library can be used to implement the numerical scheme. In [Drw08], special emphasis is layed on the efficient computation of the stray-field energy by use of \mathcal{H}^2 matrices, cf [HB02]. In our simulations, we experienced that a matrix compression by use of \mathcal{H} -matrices [Hac99], although not optimal, is sufficient. For the computation of the low-rank approximation of admissible blocks, we used the black-box adaptive cross algorithm [Beb00].

The final Chapter 5 is devoted to the empirical study of the performance of our proposed numerical scheme. In the first section, we study the behavior of the discretization error and the error estimator η_ℓ^H . We observe that η_ℓ^H is efficient and reliable. Given smooth data, computations with uniform mesh-refinements reveal an asymptotic behavior of

$$\|\mathbf{m}_0^\varepsilon - \mathbf{m}_\ell\| = \mathcal{O}(h^{1/2}) = \mathcal{O}(N_T^{1-/4})$$

with N_T the number of triangles of the underlying mesh. The adaptive algorithm is not able to improve the asymptotic behavior significantly, it leads though to higher accuracy. This is because the

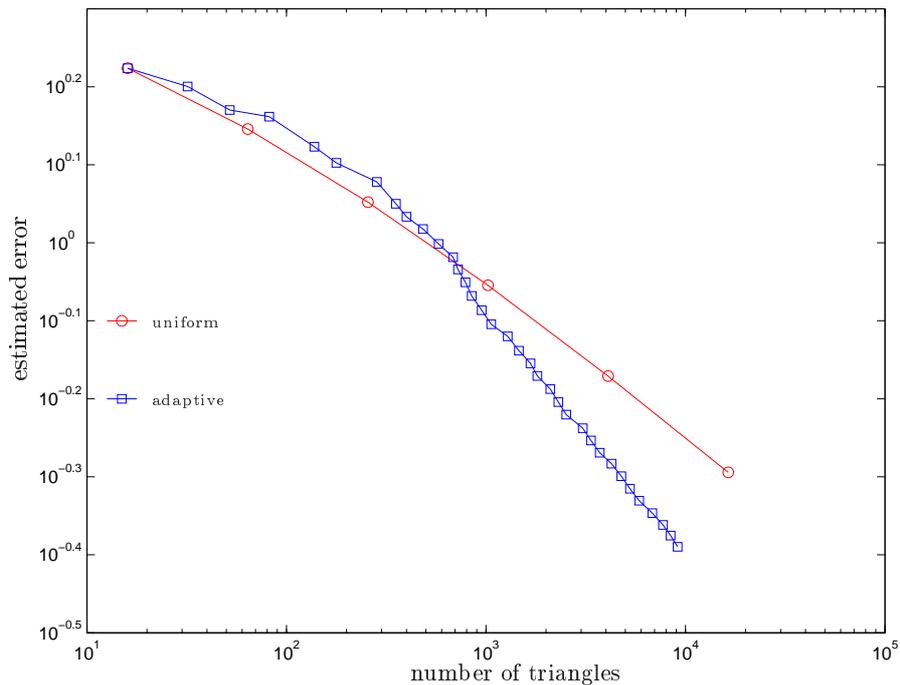


Figure 1.2: Estimated error for non-smooth data. The adaptive algorithm leads to convergence at a higher order than the uniform approach.

L^2 component of the magnetization is resolved optimally by uniform meshes. The divergence $\nabla \cdot \mathbf{m}$, however, demands strongly adapted meshes towards the edges of ω for optimal resolution. See Figure 1.1 for the typical behavior of a solution with smooth data. It seems that these goals are incompatible, at least with isotropic sequences of meshes. In a further experiment with discontinuous \mathbf{f} , the uniform computation reveals a smaller rate of convergence and the adaptive algorithm recovers a rate of at least $\mathcal{O}(N_T^{-1/4})$.

In the second section, we study the error introduced by the penalty scheme. For smooth data and uniform mesh-refinements, we observe a rate of convergence of

$$\|\mathbf{m}_h^0 - \mathbf{m}_h^\varepsilon\| = \mathcal{O}(\varepsilon^{1/2})$$

as predicted by theory. It seems that the proposed error estimator η_ℓ^ε is reliable. We observe that discretization and penalization error are not entirely independent. As $h \rightarrow 0$, the penalty error is reduced slightly, and as $\varepsilon \rightarrow 0$, it also seems that the discretization error is reduced slightly. In an experiment where we verified that the choice of $\varepsilon = h^1$ is empirically optimal, we observe an increased rate of convergence

$$\|\mathbf{m}^* - \mathbf{m}_h^\varepsilon\| = \mathcal{O}(N_T^{-\alpha})$$

with $\alpha > 1/4$.

The last two sections are concerned with simulations applying the full ε - and h -adaptive algorithm. In an experiment with non-smooth \mathbf{f} , we observe that – in contrast to the uniform approach – the adaptive scheme recovers the same rate of convergence as for smooth data, see Figure 1.2. In the case of soft films with $q = 0$, the adaptive scheme leads to linear convergence, whereas uniform mesh-refinements only reveal $h^{1/2}$. Our results suggest that the heuristic adaptive algorithm always leads to higher accuracy than a uniform approach. In case of soft films and in presence of strong singularities, it is able to improve the rate of convergence significantly when compared to the uniform approach.

1.3 Acknowledgement

First, I would like to thank Jens Markus Melenk for his willingness to share his expertise in mathematics and numerics with me. Whenever I was in need of a fruitful discussion, the doors to his office were widely open to me and I could be sure to have his dedication to answering my questions.

I also want to thank Dirk Praetorius, who inspired me to specialize in the field of numerics. His talent to share his enthusiasm as well as his excellent skills as mathematician make me feel grateful to have profited from his guidance over the past years.

Furthermore, I also want to thank Sören Bartels from the University of Bonn for examining this dissertation as external reviewer.

I want to express my thankfulness for the opportunity to be a member of the graduate school “Differential Equations - Models in Science and Engineering” supported by the Austrian Science Fund FWF under the grant WK800-N05.

The last two months of my research activities were generously funded as part of the project “Adaptive Boundary Element Methods” under the grant FWF Project P21732 of the Austrian Science Fund.

I want to thank all of my colleagues at the institute for the interesting discussions and valuable input. Specially Markus Aurada for being patient and listening to me whenever I had to talk about some topic or to get things clear. To Markus Mayr for his precious input and advice in the implementation of the algorithms in C++. And to all the other dear friends who helped me become more mature as a mathematician over the past years.

Finally, I want to express my wholehearted thankfulness to my family. My wife Bianca and my children Joel and Jan Simon whose love and support seems unconditional and infinite. My parents Elzbieta and Heber who helped me become who I am and who always support me in all possible ways. My brothers Heber and Andres who are living idols in many ways to me and help me grow as a human being. And also my friends Benedikt, Clemens, Jakob, and Serge who I consider to be part of my family: For those about to rock, I salute you!

Chapter 2

A model problem in thin-film micromagnetics

For the convenience of the reader, in Section 2.1, we briefly present the full stationary micromagnetic problem due to Landau and Lifshits [LL35]. In Section 2.2, we move on to the reduced model introduced by [DKM⁺01] that describes micromagnetic phenomena in relatively large thin-film samples. This will serve as a prototype for an infinite dimensional quadratic minimization problem with convex but non-linear inequality constraints and energy contributions stemming from a non-local operator.

At first glance the energy functional under consideration depends on two quantities, the magnetization \mathbf{m} and the stray field $-\nabla u$. These are coupled by the magnetostatic Maxwell equation, i.e. a PDE constraint.

After collecting some analytical prerequisites on Sobolev spaces in Section 2.3, we closely analyze the Maxwell equation arising in the reduced model problem in thin-film micromagnetics. In Section 2.4, we establish an explicit representation of u as a simple-layer potential of the divergence of the magnetization. Therefore, we may rewrite our energy functional $e(\mathbf{m}, u) = e(\mathbf{m}, u(\mathbf{m}))$, hence the energy only depends on \mathbf{m} . This has already been proposed in [DKMO02]. However, in contrast to the original work where the focus layed on a distributional point of view, we aim at constructing an appropriate Hilbert space setting. This is performed in Section 2.5 where we state the precise functional analytic framework. The choice of the energy space \mathcal{H} for the magnetization is not trivial but natural from the considerations so far. The construction is a new contribution to the understanding of the model problem.

Finally in Section 2.6, we establish well-posedness of the thin-film minimization problem in our function space setting. Existence of minimizers \mathbf{m}^* and uniqueness of the stray field $-\nabla u^*$ are proven. For non-vanishing anisotropy even uniqueness of $\mathbf{m}^* \in \mathcal{H}$ can be concluded, a result which is not included in any of the prior publications on this minimization problem.

2.1 The magnetic free energy

In the mesoscopic modeling approach, the atomic magnetic moments of a ferromagnetic sample Ω are replaced by a continuous vector field, the magnetization density \mathbf{M} , given in the units of Tesla. However its length, the saturation magnetization, only depends on the temperature and is therefore assumed to be constant within the sample. In a partially non-dimensional formulation, we may prescribe $|\mathbf{M}| = 1$

inside Ω and extend $\mathbf{M} = 0$ in the exterior. The interaction of magnetic particles is governed by two distinct effects: First, a long range interaction occurs due to the induced magnetic field. The stray field energy contribution is given by

$$\frac{1}{2} \int_{\mathbb{R}^3} |\nabla U|^2 dx$$

where the magnetic potential U is determined by the static Maxwell equation

$$(\nabla \cdot (\nabla U + \mathbf{M})) = 0 \in \mathcal{D}(\mathbb{R}^3)^*,$$

stated here in a distributional sense. Second, quantum-mechanical exchange interactions prefer constant alignment of the magnetization. This exchange energy acts on a short range, and in the mesoscopic model its contribution is given by

$$\frac{d^2}{2} \int_{\Omega} |\nabla \mathbf{M}|^2 dx,$$

where the material dependent exchange length d is also called Bloch line-width. It measures the relative strength of the exchange energy with respect to the stray field energy.

Additionally to these two contributions, the crystalline structure of ferromagnetic material may lead to certain preferred magnetization alignments. Here, we restrict ourselves to uniaxial material, i.e. the deviation of \mathbf{M} from a so-called easy axis is penalized. Without loss of generality, we assume our sample to be aligned in such a way that the easy axis is simply the first in-plane axis e_1 . The anisotropic energy contribution thus reads

$$\frac{Q}{2} \int_{\Omega} \mathbf{M}_2^2 + \mathbf{M}_3^2 dx.$$

The material parameter Q measures the relative strength between the anisotropic energy and the stray field contribution. In the literature material with $Q \ll 1$ are referred to as soft and material with $Q > 1$ as hard magnetic material.

Finally one is interested in identifying the stable states of a ferromagnetic sample under an applied exterior field \mathbf{F} . The Zeemann energy or applied field energy favors alignment of the magnetization density with the applied field

$$- \int_{\Omega} \mathbf{F} \cdot \mathbf{M} dx.$$

Summing all energy contributions, we obtain a partially dimensionless formulation of the magnetic free energy

$$E(\mathbf{M}) = \frac{d^2}{2} \int_{\Omega} |\nabla \mathbf{M}|^2 dx + \frac{1}{2} \int_{\mathbb{R}^3} |\nabla U|^2 dx + \frac{Q}{2} \int_{\Omega} \mathbf{M}_2^2 + \mathbf{M}_3^2 dx - \int_{\Omega} \mathbf{F} \cdot \mathbf{M} dx. \quad (2.1)$$

Local minimizers \mathbf{M} of the energy functional E describe meta-stable states of the magnetization. For a detailed description of the modeling approach, we refer to [Brw63].

We observe that the full micromagnetic problem is quite complex from a numerical point of view. Not only is it a non-convex minimization problem, due to the constraint $|\mathbf{M}| = 1$, it is also non-local due to the stray field energy contribution. The simulation of large ferromagnetic samples with $\text{diam}(\Omega)$ in the μm regime reveals to be computationally intense. The exchange energy – acting at a small scale

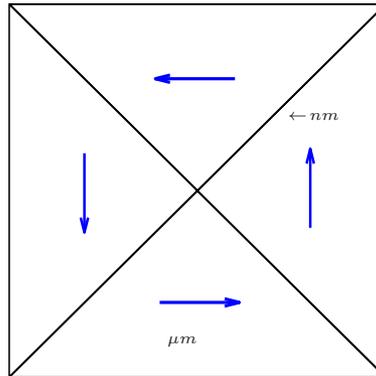


Figure 2.1: Landau state of a soft ferromagnetic film with diameter of several μm . The magnetization (blue) is nearly constant on the domains. The domain walls depicted here as lines are continuous transitions of the magnetization at nm scale.

– prefers constantly aligned magnetizations. However the long-scale stray field energy penalizes large divergence of magnetization and in particular seeks to eliminate surface charges $\mathbf{M} \cdot \mathbf{N}$ with \mathbf{N} the outer normal vector on $\partial\Omega$. The result of these competing goals is the formation of magnetic domains where the magnetization is nearly constant [LL35, SH98]. Figure 2.1 shows a typical pattern of a soft thin-film sample in the absence of an applied exterior field, the so-called Landau state. The domains are separated by domain walls which in fact are continuous transitions of the magnetization on a short nm scale. Since the location of these magnetic domain walls is in general an unknown a priori, a naive numerical approach would force one to discretize the sample of μm in size at a nm scale.

For these reasons, various reduced models for certain regimes, have been introduced and studied lately. The aim is always to simplify the full problem in order to improve computability – with as little loss of information as possible. In the following sections, we present a reduced model for thin-film micromagnetics that was first introduced in [DKM⁺01]. We will discuss the precise mathematical setting, which is fundamental to develop a sophisticated numerical analysis. This model problem serves as a prototype of an infinite dimensional quadratic minimization problem with non-linear constraints and involving non-local operators.

2.2 A reduced model in thin-film micromagnetics

We will focus on the simulation of thin ferromagnetic films. We thus restrict our attention to a simple yet relevant geometric set-up. We consider the sample Ω to be cylindrical with basis ω and thickness t . Therefore, Ω can be written in the form

$$\Omega = \omega \times (0, t),$$

where we demand $\omega \subseteq \mathbb{R}^2$ to be a bounded simply connected Lipschitz domain. Since our interest is focused on thin ferromagnetic films we assume

$$t \ll \ell,$$

where ℓ denotes the diameter of ω . Our model problem is therefore governed by four different scales, namely t, ℓ, d , and Q , where the latter two are scales stemming from material properties. We stress that these scales may vary by orders of magnitude.

In [DKM⁺01], a reduced model has been proposed to describe micromagnetic phenomena in thin films in the limit case $t \rightarrow 0$: Neglecting the thickness of Ω , we represent our ferromagnetic sample by the two-dimensional surface ω . With an in-plane applied exterior field $\mathbf{f} : \omega \rightarrow \mathbb{R}^2$, we seek a magnetization $\mathbf{m}^* : \omega \rightarrow \mathbb{R}^2$ that satisfies the convex pointwise constraint $|\mathbf{m}| \leq 1$ and minimizes the reduced energy

$$e(\mathbf{m}) = \frac{1}{2} \int_{\mathbb{R}^3} |\nabla u|^2 dx + \frac{q}{2} \int_{\omega} \mathbf{m}_2^2 dx - \int_{\omega} \mathbf{f} \cdot \mathbf{m} dx. \quad (2.2)$$

The magnetic potential $u : \mathbb{R}^3 \rightarrow \mathbb{R}$ now satisfies

$$\int_{\mathbb{R}^3} \nabla u \cdot \nabla v dx = \int_{\omega} \mathbf{m} \cdot \nabla v(x, 0) dx \quad \text{for all } v \in \mathcal{D}(\mathbb{R}^3) := \mathcal{C}_c^\infty(\mathbb{R}^3). \quad (2.3)$$

Let \mathbf{n} denote the outer normal on $\gamma = \partial\omega$. For smooth functions u, \mathbf{m} with $\mathbf{m} \cdot \mathbf{n} = 0$ on γ , integration by parts on the right-hand side of (2.3) yields

$$\int_{\mathbb{R}^3} \nabla u \cdot \nabla v dx = - \int_{\omega} \nabla \cdot \mathbf{m} v dx \quad \text{for all } v \in \mathcal{D}(\mathbb{R}^3). \quad (2.4)$$

A rigorous analysis has been performed in [DKMO02], where Γ -convergence of the full problem (2.1) to the reduced problem (2.2) is proven for vanishing thickness $t \rightarrow 0$ in the asymptotic regime

$$d^2 \frac{\log 1/t}{t} \rightarrow 0.$$

Therefore, the model is mathematically justified for samples that are sufficiently thin $t \ll 1$ and large $\frac{\ell t}{d^2} \gg \log(\frac{\ell}{t})$. First numerical experiments in [DKM⁺01] and [DKMO02] for soft samples, where $q \ll 1$ is neglected for simplicity, show very good results. The simulated behavior of the magnetization coincides well with measured data from experiments with thin permalloy films, i.e. the model seems to be of practical relevance. In particular the samples under consideration with $\ell \in [10\mu m, 100\mu m]$ and $t \in [10nm, 500nm]$ are hardly accessible through direct simulation of the full problem.

In these first implementations, an interior point method is used to compute the minimizers of the energy (2.2). Details on the methods used and implementational aspects can be found in the recent thesis [Drw08]. All experiments and most of the numerical analysis, however, were performed for the special case of soft ferromagnetic materials with small anisotropic parameter $q \ll 1$ and constant applied exterior field $\mathbf{f} \in \mathbb{R}^2$. For simplicity the anisotropy energy contribution is dropped, i.e. $q = 0$ is assumed. With these simplifications the energy (2.2) solely depends on the divergence $\nabla \cdot \mathbf{m}$ of the magnetization. The consequence of this is that the solution to the minimization problem is not uniquely determined. For any minimizer \mathbf{m}^* one may add an arbitrary divergence free magnetization $\tilde{\mathbf{m}}$ with $\int_{\omega} \tilde{\mathbf{m}} \cdot \mathbf{f} dx = 0$ to obtain another minimizer $\tilde{\mathbf{m}}^* = \mathbf{m}^* + \tilde{\mathbf{m}}$. In particular, domain structure may be reconstructed by first computing some minimizer \mathbf{m}^* and then adding an appropriate $\tilde{\mathbf{m}}$ such that $|\tilde{\mathbf{m}}^*| = 1$ and that the wall energy, i.e. the contribution of discontinuities, is minimized by some heuristics, see [Drw08].

In contrast to the prior works, we consider arbitrary uniaxial materials with $q > 0$ and the applied field $\mathbf{f} \in L^2$ may be non-constant. Interpretation of results obtained from simulation of hard material may be more difficult since there is no obvious way to reconstruct domain patterns. However, we stress that there holds $\nabla U \rightarrow \nabla u$ in a weak sense as $t \rightarrow 0$. The stray field and the reduced energy therefore preserve physical meaning. Moreover for a strong applied field $|\mathbf{f}| \gg 1$, the magnetization \mathbf{m} may be saturated $|\mathbf{m}| = 1$ within a penetration region. There the magnetization is already uniquely determined by the reduced model and is therefore also of direct physical relevance.

In the original publications [DKMO02, Drw08] the focus was on a distributional point of view. In order to analyze the reduced problem from a numerical point of view, it is useful to determine an appropriate and precise functional analytic setting, i.e. define the function spaces for all involved quantities. Obviously the magnetic field \mathbf{m} should at least satisfy $\mathbf{m} \in L^2(\omega)^2$. This space, however, is still too large, as the existence of the magnetostatic potential u from (2.4) demands some further regularity.

2.3 Sobolev spaces

We assume throughout the entire work that $\omega \subseteq \mathbb{R}^2$ is a bounded Lipschitz domain. For the convenience of the reader, we give a brief definition of Sobolev spaces of relevance in this work. For a detailed representation the reader may refer to e.g. [Ada75]. We define the Sobolev space

$$H^1(\omega) := \{u \in L^2(\omega) \mid \nabla u \in L^2(\omega)\}$$

with associated norm

$$\|u\|_{H^1(\omega)} = \left(\|u\|_{L^2(\omega)}^2 + \|\nabla u\|_{L^2(\omega)}^2 \right)^{1/2}.$$

Here, ∇u denotes the weak gradient, and it is well known that $H^1(\omega)$ is a Hilbert space. For functions $u \in H^1(G)$ for a bounded Lipschitz domain $G \subseteq \mathbb{R}^d$ with vanishing integral mean $\int_G u \, dx = 0$ there holds a Poincaré inequality [Eva98, Chapter 5.8: Theorem 1]

$$\|u\|_{L^2(G)} \leq C_P \|\nabla u\|_{L^2(G)}, \quad (2.5)$$

where the constant C_P depends only on the domain G .

For $k \in \mathbb{N}_{\geq 2}$ we inductively define Sobolev spaces of higher integer order by

$$H^k(\omega) := \{u \in L^2(\omega) \mid \nabla u \in H^{k-1}(\omega)\}$$

with the natural norm

$$\|u\|_{H^k(\omega)} = \left(\|u\|_{L^2(\omega)}^2 + \|\nabla u\|_{H^{k-1}(\omega)}^2 \right)^{1/2}.$$

Moreover we identify $H^0(\omega) = L^2(\omega)$. The dual space with respect to the extended L^2 scalar product is denoted by $\tilde{H}^{-k}(\omega) := (H^k(\omega))^*$.

For any real number $s \in (0, 1)$ we define the Sobolev-Slobodeckij semi-norm by

$$|u|_s := \left(\int_{\omega} \int_{\omega} \frac{|u(x) - u(y)|^2}{|x - y|^{2+s}} \, dy \, dx \right)^{1/2}.$$

Then, the fractional order Sobolev space H^{k+s} with $k \in \mathbb{Z}_{\geq 0}$ and $s \in (0, 1)$ is given by

$$H^{k+s} := \{u \in H^k(\omega) \mid |\partial^\alpha u|_s < \infty \text{ for } |\alpha| = k\}.$$

This space may be equipped with the norm

$$\|u\|_{H^{k+s}(\omega)} = \left(\|u\|_{H^k(\omega)}^2 + \sum_{|\alpha|=k} |\partial^\alpha u|_s^2 \right)^{1/2}$$

and is again a Hilbert space. The dual space with respect to the extended L^2 scalar product is again denoted by $\tilde{H}^{-k-s}(\omega) = (H^{k+s}(\omega))^*$. All definitions may be read componentwise to obtain Sobolev spaces for vector-valued functions.

For any positive real number $s \in \mathbb{R}_{\geq 0}$, we moreover define the space $\tilde{H}^s(\omega) := \{u|_{\omega} \mid u \in H^s(\mathbb{R}^2) \text{ and } u|_{\mathbb{R}^2 \setminus \bar{\omega}} = 0\}$. The corresponding negative order dual space is denoted by $H^{-s}(\omega) := (\tilde{H}^s(\omega))^*$.

Next, we define some Sobolev spaces that will arise in the context of functions defined on the full space \mathbb{R}^3 . We follow the lines of [SS04, Section 2.6]. For any real number $s \in \mathbb{R}$ the space $H_{loc}^s(\mathbb{R}^3)$ is defined as the set of all functions which are locally in H^s , i.e. for any test function $\phi \in \mathcal{D}(\mathbb{R}^3)$ all functions $u \in H_{loc}^s(\mathbb{R}^3)$ satisfy $\phi u \in H^s(\mathbb{R}^3)$. Here, we identified $\tilde{H}^{-t}(\mathbb{R}^3) =: H^{-t}(\mathbb{R}^3)$ for $t = |s|$.

The support of a negative order Sobolev function $u \in H^s(\mathbb{R}^3)$ with $-\infty < s \leq 0$ is defined by localization with all possible test functions. The function $u \in H^s(\mathbb{R}^3)$ is said to vanish on an open set $O \subseteq \mathbb{R}^3$ if for all test functions $\phi \in \mathcal{D}(\mathbb{R}^3)$ with $\text{supp}(\phi) \subseteq O$ there holds $\int_{\mathbb{R}^3} \phi u \, dx = 0$. Then, the support of $u \in H^s(\mathbb{R}^3)$ is defined as the maximal closed set A with $u = 0$ on $\mathbb{R}^3 \setminus A$.

We now may define the set

$$H_{comp}^s(\mathbb{R}^3) := \bigcup_A \{u \in H_{loc}^s(\mathbb{R}^3) \mid \text{supp}(u) \subseteq A\}$$

for all compact subsets $A \subseteq \mathbb{R}^3$. The L^2 scalar product may be extended to the duality brackets

$$\langle \cdot, \cdot \rangle_{H_{comp}^{-s}(\mathbb{R}^3) \times H_{loc}^s(\mathbb{R}^3)},$$

see [SS04, Satz 2.6.7].

Next we collect some fundamental theorems on existence and continuity of trace operators.

Proposition 2.1 ([McL00, Theorem 3.37]). *Let G be a smooth (\mathcal{C}^∞), bounded domain with $\omega \subseteq \partial G$ and let $s > 1/2$. Then there exists a uniquely determined trace operator*

$$t_0 \in L(H^s(G); H^{s-1/2}(\partial G))$$

such that $t_0 u = u|_{\partial G}$ for functions $u \in \mathcal{C}(\bar{G})$. Moreover, there is a uniquely determined trace operator

$$t \in L(H_{loc}^s(\mathbb{R}^3); H^{s-1/2}(\omega))$$

with $tu = u|_{\omega}$ for functions $u \in \mathcal{C}(\mathbb{R}^3)$.

Proposition 2.2 ([McL00, Lemma 4.3]). *Let $u \in H_{\Delta}^1(G) := \{u \in H^1(G) \mid \Delta u \in L^2(G)\}$ for some bounded Lipschitz domain G . Then, there exists a uniquely determined $t_1 u \in H^{-1/2}(\partial G)$ such that*

$$\int_G \Delta u v \, dx + \int_G \nabla u \nabla v = \langle t_1 u, t_0 v \rangle \quad \forall v \in H^1(G),$$

and the operator $t_1 : H_{\Delta}^1(G) \rightarrow H^{-1/2}(\partial G)$ is continuous. For $u \in \mathcal{C}^1(\bar{G})$, the function $t_1 u = \partial_n u$ is the normal derivative on the boundary.

Having collected this preliminary notation and some properties of trace operators, we may proceed to analyze the magnetostatic Maxwell equation.

2.4 The static Maxwell equation in thin-film micromagnetics

2.4.1 Strong form of the Maxwell equation

The magnetic potential is given as the solution of the variational formulation (2.3). Understanding the potential u is crucial to define the appropriate function space for the magnetization \mathbf{m} .

Lemma 2.3. *For smooth $\mathbf{m} \in \mathcal{C}^1(\overline{\omega})$, every weak solution u of the magnetostatic Maxwell equation (2.3) which is sufficiently smooth, i.e. $u \in \mathcal{C}^2(\mathbb{R}^3 \setminus \overline{\omega}) \cap \mathcal{C}(\mathbb{R}^3) \cap \mathcal{C}^1(\mathbb{R}^2 \times \mathbb{R}_{\geq 0}) \cap \mathcal{C}^1(\mathbb{R}^2 \times \mathbb{R}_{\leq 0})$, solves the strong form*

$$\begin{aligned} \Delta u &= 0 & \text{in } \mathbb{R}^3 \setminus \overline{\omega} \\ \left[\frac{\partial u}{\partial x_3} \right] &= \nabla \cdot \mathbf{m} & \text{on } \omega \\ [u] &= 0 & \text{on } \omega \\ \mathbf{m} \cdot \mathbf{n} &= 0 & \text{on } \gamma = \partial\omega \subseteq \mathbb{R}^2 \end{aligned} \tag{2.6}$$

where \mathbf{n} denotes the outer normal in \mathbb{R}^2 of ω .

Proof. We split the left hand side of (2.3) into two integrals

$$\int_{\mathbb{R}^3} \nabla u \cdot \nabla v \, dx = \int_{\mathbb{R}_-^3} \nabla u \cdot \nabla v \, dx + \int_{\mathbb{R}_+^3} \nabla u \cdot \nabla v \, dx$$

over domains $\mathbb{R}_+^3 := \{x \in \mathbb{R}^3 | x_3 > 0\}$ and $\mathbb{R}_-^3 := \{x \in \mathbb{R}^3 | x_3 < 0\}$. We use integration by parts to see

$$\int_{\mathbb{R}_-^3} \nabla u \cdot \nabla v \, dx + \int_{\mathbb{R}_+^3} \nabla u \cdot \nabla v \, dx = - \int_{\mathbb{R}^3} \Delta u v \, dx - \int_{\mathbb{R}^2} \frac{\partial u(x, 0+)}{\partial x_3} v \, dx + \int_{\mathbb{R}^2} \frac{\partial u(x, 0-)}{\partial x_3} v \, dx$$

On the other hand, integration by parts on the right-hand side of (2.3) shows

$$\int_{\omega} \mathbf{m} \cdot \nabla v(x, 0) \, dx = - \int_{\omega} \nabla \cdot \mathbf{m} v \, dx + \int_{\gamma} (\mathbf{m} \cdot \mathbf{n}) v \, ds$$

which results in

$$- \int_{\mathbb{R}^3} \Delta u v \, dx - \int_{\mathbb{R}^2} \frac{\partial u(x, 0+)}{\partial x_3} v \, dx + \int_{\mathbb{R}^2} \frac{\partial u(x, 0-)}{\partial x_3} v \, dx = - \int_{\omega} \nabla \mathbf{m} \cdot v \, dx + \int_{\gamma} (\mathbf{m} \cdot \mathbf{n}) v \, ds$$

with the equation valid for all $v \in \mathcal{D}(\mathbb{R}^3)$. By definition of the jump $[f] := f(x_1, x_2, 0^+) - f(x_1, x_2, 0^-)$ of a function $f : \mathbb{R}^3 \rightarrow \mathbb{R}$ across the plane $x_3 = 0$, it holds that

$$- \int_{\mathbb{R}^2} \frac{\partial u(x, 0+)}{\partial x_3} v \, dx + \int_{\mathbb{R}^2} \frac{\partial u(x, 0-)}{\partial x_3} v \, dx = - \int_{\mathbb{R}^2} \left[\frac{\partial u}{\partial x_3} \right] v \, dx.$$

Considering that u is smooth in $\mathbb{R}^3 \setminus \overline{\omega}$ the jump $\left[\frac{\partial u}{\partial x_3} \right]$ can only be non-zero on ω . The equation

$$- \int_{\mathbb{R}^3} \Delta u v \, dx + \int_{\omega} \nabla \cdot \mathbf{m} v \, dx = \int_{\omega} \left[\frac{\partial u}{\partial x_3} \right] v \, dx + \int_{\gamma} (\mathbf{m} \cdot \mathbf{n}) v \, ds$$

holds for all $v \in \mathcal{D}(\mathbb{R}^3 \setminus \bar{\omega})$ which yields the pointwise relation $\Delta u = 0$ in $\mathbb{R}^3 \setminus \bar{\omega}$. The remaining equation

$$\int_{\omega} \nabla \cdot \mathbf{m} v \, dx = \int_{\omega} \left[\frac{\partial u}{\partial x_3} \right] v \, dx + \int_{\gamma} (\mathbf{m} \cdot \mathbf{n}) v \, ds$$

holds for all $v \in \mathcal{D}(\omega)$ from which follows $\nabla \cdot \mathbf{m} = \left[\frac{\partial u}{\partial x_3} \right]$ and consequently $\int_{\gamma} (\mathbf{m} \cdot \mathbf{n}) v \, ds = 0$ for all $v \in \mathcal{D}(\gamma)$, hence $\mathbf{m} \cdot \mathbf{n} = 0$ on γ . \blacksquare

Remark. The proof of Lemma 2.3 reveals that the relation $\mathbf{m} \cdot \mathbf{n} = 0$ on γ is necessary for the existence of a smooth solution u . Suppose $\mathbf{m} \cdot \mathbf{n} \neq 0$ on γ . Integration by parts

$$\int_{\mathbb{R}^2} \mathbf{m} \cdot \nabla v \, dx = - \int_{\mathbb{R}^2} \nabla \cdot \mathbf{m} v \, dx + \int_{\gamma} \mathbf{m} \cdot \mathbf{n} v \, ds$$

then implies that $\nabla \cdot \mathbf{m}$ is a distribution on \mathbb{R}^2 with relevant contribution on γ . This, however, means that a solution u of the variational form (2.3) cannot satisfy $u \in H_{loc}^1$. In Section 2.5 this fact influences the choice of the energy space for \mathbf{m} which must include the restriction $\mathbf{m} \cdot \mathbf{n} = 0$ on γ in an appropriate sense. \square

2.4.2 Newtonian and simple-layer potential

We recall some well-known properties of the simple-layer potential of the Laplace operator. We mostly summarize statements from [Ste87],[SS04] and [Ste03].

Proposition 2.4 ([SS04, Satz 3.1.2, Satz 3.1.4]). *The Newtonian potential*

$$\mathcal{N}f(x) := \frac{1}{4\pi} \int_{\mathbb{R}^3} \frac{f(y)}{\|x - y\|} dy \quad \forall x \in \mathbb{R}^3 \quad (2.7)$$

may be extended to a continuous and linear operator $\mathcal{N} : H_{comp}^{-1}(\mathbb{R}^3) \rightarrow H_{loc}^1(\mathbb{R}^3)$. For any bounded domain $\Omega \subset \mathbb{R}^3$, there holds

$$\mathcal{N} \in L(\tilde{H}^{-1}(\Omega); H^1(\Omega)).$$

There holds $-\Delta \mathcal{N}f = f$ for all $f \in \tilde{H}^{-1}(\Omega)$. The Newtonian potential is self-adjoint, i.e. $\mathcal{N}' = \mathcal{N}$.

Definition. We define the simple-layer potential

$$S := \mathcal{N}t' \quad (2.8)$$

in the distributional sense, with t' the adjoint trace operator on ω , i.e.

$$\langle Sv, w \rangle = \langle \mathcal{N}t'v, w \rangle = \langle v, t\mathcal{N}'w \rangle. \quad (2.9)$$

\square

The following theorem follows immediately from the linearity and continuity of \mathcal{N} and t' as well as the definition of S .

Proposition 2.5 ([Ste03, Kapitel 6.2, Kapitel 6.3], [SS04, Satz 3.1.6, Satz 3.1.16]). *The simple-layer potential $S : \tilde{H}^{-1/2}(\omega) \rightarrow H_{loc}^1(\mathbb{R}^3)$ is linear and continuous, i.e. for every bounded domain $G \subseteq \mathbb{R}^3$, there is a constant $C(G) > 0$ such that*

$$\|Sv\|_{H^1(G)} \leq C(G) \|v\|_{\tilde{H}^{-1/2}(\omega)} \quad \forall v \in \tilde{H}^{-1/2}(\omega).$$

Furthermore, there holds

$$\Delta Sv = 0 \quad \text{in } \mathbb{R}^3 \setminus \bar{\omega}. \quad (2.10)$$

Lemma 2.6 ([SS04, Satz 3.1.6, Satz 3.1.1]). *For $v \in L^1(\omega)$, the simple-layer potential Sv may be represented by*

$$Sv(x) = \frac{1}{4\pi} \int_{\omega} \frac{v(y)}{\|x - y\|} ds_y. \quad (2.11)$$

In this case, it holds that $Sv \in C^\infty(\mathbb{R}^3 \setminus \bar{\omega})$.

Proof. The proof of the second statement can be read word by word as in [SS04]. The proof of the first statement has to be slightly modified to fit our setting, i.e. the case of a screen. Let $v \in L^1(\omega)$ and $w \in \mathcal{D}(\mathbb{R}^3)$. Then it holds that

$$\begin{aligned} (\mathcal{N}'v, w)_{L^2(\mathbb{R}^3)} &= (v, t\mathcal{N}w)_{L^2(\omega)} = \frac{1}{4\pi} \int_{\omega} v(y) \int_{\mathbb{R}^3} \frac{w(x)}{\|x - y\|} dx ds_y \\ &= \frac{1}{4\pi} \int_{\mathbb{R}^3} w(x) \int_{\omega} \frac{v(y)}{\|x - y\|} ds_y dx = (Sv, w)_{L^2(\mathbb{R}^3)}, \end{aligned}$$

where we have used Fubini's theorem and $\mathcal{N}' = \mathcal{N}$. ■

2.4.3 Solution of the magnetostatic Maxwell equation

We need one more result from the theory of elliptic partial differential equations, before we can finally prove the jump conditions of the simple-layer potential. This will enable us to write the solution of the magnetostatic Maxwell equation explicitly.

In [Ste87] one central idea for the analysis of the weakly-singular integral operator S is to define a bounded Lipschitz-domain $G_- \subset \mathbb{R}^3$ with $\omega \subset \Gamma_- := \partial G_-$, where the outer normal on the boundary additionally satisfies $\mathbf{n}|_{\omega} = (0, 0, 1)^T$. We follow the same ideas to prove the jump conditions of the simple-layer potential. We stress that we did not find the proof of Theorem 2.8 stated for screen problems in the literature.

Proposition 2.7 ([McL00, Theorem 4.4]). *Let $L_c^2(G) := \{v \in L^2 \mid \text{supp}(v) \subseteq G \text{ compact}\}$. Define $H_L^1 := \{u \in H_{loc}^1 \mid \Delta u \in L_c^2(G_{\pm})\}$ for a bounded Lipschitz domain $G_- \subset \mathbb{R}^3$ and the complementary domain $G_+ = \mathbb{R}^3 \setminus \overline{G_-}$. Then, there holds the second Green's formula*

$$-(\Delta u, v)_{L^2(G_{\pm})} + (u, \Delta v)_{L^2(G_{\pm})} = \mp(\langle t_0 u, t_1 v \rangle - \langle t_1 u, t_0 v \rangle) \quad (2.12)$$

for all $u, v \in H_L^1(G_{\pm})$, where the sign in (2.12) is contrary to the index of the domain.

Theorem 2.8 (Jump conditions). *The simple-layer potential satisfies the jump conditions*

$$[S\varphi] = 0 \in H^{1/2}(\omega), \quad [t_1 S\varphi] = \left[\frac{\partial S\varphi}{\partial x_3} \right] = -\varphi \in \tilde{H}^{-1/2}(\omega). \quad (2.13)$$

Proof. The continuity $[S\varphi] = 0$ follows from $S = \mathcal{N}t'$ and the mapping properties of t' and \mathcal{N} as well as the properties of t : Recall that $S : \tilde{H}^{-1/2}(\omega) \rightarrow H_{loc}^1(\mathbb{R}^3)$. The trace operator $t : H_{loc}^1(\mathbb{R}^3) \rightarrow H^{1/2}(\omega)$ is uniquely determined by continuity. In particular the trace from above and below ω coincide, cf. [SS04, Satz 2.6.8].

To prove the jump condition of the normal derivative we proceed as follows: Let $\varphi \in \tilde{H}^{-1/2}$ and define $u = S\varphi$. Then, $\Delta u = 0$ in $\mathbb{R}^3 \setminus \bar{\omega}$ and therefore $u \in H_L^1(\mathbb{R}^3 \setminus \bar{\omega})$. The second Green's formula with $v \in \mathcal{D}(\mathbb{R}^3)$ yields

$$(u, \Delta v)_{L^2(G_{\pm})} = \mp(\langle t_0 u, t_1 v \rangle - \langle t_1 u, t_0 v \rangle).$$

We add the equations for G_- and G_+ and obtain

$$(u, \Delta v)_{L^2(\mathbb{R}^3)} = -\langle [u], t_1 v \rangle + \langle [t_1 u], t_0 v \rangle,$$

where we have used $[v] = [t_1 v] = 0$ and $u \in L_{loc}^2(\mathbb{R}^3)$. We have already proven $[u] = [S\varphi] = 0$. Therefore, we obtain

$$\langle [t_1 u], t_0 v \rangle = (u, \Delta v)_{L^2(\mathbb{R}^3)}.$$

Plugging in the definition of S we obtain

$$(u, \Delta v)_{L^2(\mathbb{R}^3)} = (\Delta u, v)_{L^2(\mathbb{R}^3)} = (\Delta \mathcal{N}t'\varphi, v)_{L^2(\mathbb{R}^3)} = -\langle t'\varphi, v \rangle = -\langle \varphi, tv \rangle.$$

Altogether, we have proven

$$\langle [t_1 S\varphi], t_0 v \rangle = -\langle \varphi, tv \rangle.$$

Since $t_0(\mathcal{D}(\mathbb{R}^3))$ is dense in $H^{1/2}(\omega)$ and therefore $t(\mathcal{D}(\mathbb{R}^3))$ is dense in $H^{1/2}(\omega)$ and $S\varphi \in \mathcal{C}^\infty(\mathbb{R}^3 \setminus \bar{\omega})$ for $\varphi \in L^2$, the jump condition

$$\left[\frac{\partial S\varphi}{\partial x_3} \right] = -\varphi \in \tilde{H}^{-1/2}(\omega)$$

follows with density arguments. ■

Corollary 2.9. *With the simple-layer potential S from (2.11) and given $\nabla \cdot \mathbf{m} \in \tilde{H}^{-1/2}(\omega)$, the function $S(-\nabla \cdot \mathbf{m})$ is a solution of the Maxwell equation (2.3). ■*

2.5 The energy spaces \mathcal{H} and $B_1^2(\mathbb{R}^3)$

Since $S : \tilde{H}^{-1/2}(\omega) \rightarrow H_{loc}^1(\mathbb{R}^3)$, we have to ensure $\nabla \cdot \mathbf{m} \in \tilde{H}^{-1/2}(\omega)$ in the definition of the function space for \mathbf{m} . To that end, we must explain the meaning of $(\nabla \cdot) : L^2(\omega)^2 \rightarrow \tilde{H}^{-1/2}(\omega)$.

We summarize some observations:

- The magnetostatic potential u may be represented as the simple-layer potential of $\nabla \cdot \mathbf{m}$.
- The simple-layer potential is a linear, and continuous mapping $S : \tilde{H}^{-1/2}(\omega) \rightarrow H_{loc}^1(\mathbb{R}^3)$.
- If we prescribe $\nabla \cdot \mathbf{m} \in \tilde{H}^{-1/2}(\omega)$, then existence of u is ensured. On the other hand the precise meaning of $\nabla \cdot : L^2(\omega)^2 \rightarrow \tilde{H}^{-1/2}(\omega)$ is not trivially clear.

- Suppose $\mathbf{m} \cdot \mathbf{n} \neq 0$ on γ , then a solution of the reduced Maxwell equation (2.3) does not in general satisfy $u \in H_{loc}^1(\mathbb{R}^3)$. Assume e.g. $\mathbf{m} = (1, 0)^T$ constant. Then $\mathbf{m} \cdot \mathbf{n} \neq 0 \in L^2(\gamma)$ and $\nabla \cdot \mathbf{m} = 0 \in L^2(\omega)$. Choose some smooth and bounded domain $G \subseteq \mathbb{R}^3$ with $\bar{\omega} \subseteq G$. Integration by parts in the Maxwell equation (2.3) yields for all $v \in \mathcal{D}(G)$

$$\int_G \nabla u \nabla v \, dX = \int_{\omega} \mathbf{m} \cdot \nabla v \, dx = - \int_{\omega} \nabla \cdot \mathbf{m} v \, dx + \int_{\gamma} (\mathbf{m} \cdot \mathbf{n}) v|_{\gamma} \, ds \quad (2.14)$$

In the right-hand side of (2.14), the integral on ω vanishes due to the choice of \mathbf{m} . The term $\int_{\gamma} (\mathbf{m} \cdot \mathbf{n}) v \, ds$ cannot be extended to define a linear functional for $v \in H^1(G)$, since the restriction to γ is not well-defined in $H^1(G)$. We conclude that $v \mapsto (\nabla u, \nabla v)_{\omega}$ does not define a continuous functional in $H^1(G)$, which means $\nabla u \notin L^2(G)$ and, therefore, $u \notin H^1(G)$.

2.5.1 The energy space \mathcal{H} for the magnetization

Our goal is to first explain the weak divergence $\nabla \cdot \mathbf{m}$ of an L^2 -vector field and then construct an appropriate energy space \mathcal{H} for the magnetization, which also takes the constraint $\mathbf{m} \cdot \mathbf{n} = 0$ into account. We define $\mathcal{D}(\bar{X}) := \{\varphi|_X \mid \varphi \in \mathcal{D}(\mathbb{R}^n)\}$.

In a first step, we consider the weak divergence and the spaces $H^1(\nabla \cdot; \omega)$ and $H_0^1(\nabla \cdot; \omega)$. Then, we introduce the energy space \mathcal{H} with an appropriate norm. We stress that $\mathcal{D}(\omega)^2 \subseteq L^2(\omega)^2$ is a dense subspace. For $\mathbf{m} \in \mathcal{C}^1(\bar{\omega})^2$, there holds the well-known Gauss divergence theorem

$$\int_{\omega} \mathbf{m} \cdot \nabla \varphi \, dx + \int_{\omega} (\nabla \cdot \mathbf{m}) \varphi \, dx = \int_{\gamma} (\mathbf{m} \cdot \mathbf{n}) \varphi \, ds \quad \text{for all } \varphi \in \mathcal{D}(\omega). \quad (2.15)$$

Since all functions in $\mathcal{D}(\omega)$ are zero on the boundary γ the right hand-side in (2.15) vanishes.

Definition. A function $v \in L^2(\omega)$ is called weak divergence of $\mathbf{m} \in L^2(\omega)^2$ if it satisfies

$$\int_{\omega} v \varphi \, dx = - \int_{\omega} \mathbf{m} \cdot \nabla \varphi \, dx \quad \text{for all } \varphi \in \mathcal{D}(\omega). \quad (2.16)$$

In this case and according to the fundamental theorem of calculus of variations, also known as du Bois-Reymond Lemma [dBR79], the weak divergence of \mathbf{m} is unique, and we simply write $\nabla \cdot \mathbf{m} := v$. \square

Definition. We define the space

$$H^1(\nabla \cdot; \omega) := \{\mathbf{m} \in L^2(\omega)^2 \mid \nabla \cdot \mathbf{m} \in L^2(\omega)\} \quad (2.17)$$

with the canonical norm $\|\mathbf{m}\|_{H^1(\nabla \cdot; \omega)} := (\|\mathbf{m}\|_{L^2(\omega)^2}^2 + \|\nabla \cdot \mathbf{m}\|_{L^2(\omega)}^2)^{1/2}$. We further define $H_0^1(\nabla \cdot; \omega) := \overline{\mathcal{D}(\omega)^2}^{\|\cdot\|_{H^1(\nabla \cdot; \omega)}}$. \square

Lemma 2.10. $H^1(\nabla \cdot; \omega)$ is a Hilbert space.

Proof. Let $(\mathbf{m}_n) \subseteq H^1(\nabla \cdot; \omega)$ be a Cauchy sequence. Then from

$$\|\mathbf{m}_n - \mathbf{m}_m\|_{H^1(\nabla \cdot; \omega)}^2 \leq \varepsilon$$

we deduce that $(\mathbf{m}_n) \subseteq L^2(\omega)^2$ and $(\nabla \cdot \mathbf{m}_n) \subseteq L^2(\omega)$ both are Cauchy sequences as well. In particular, it follows that $\mathbf{m}_n \rightarrow \mathbf{m} \in L^2(\omega)^2$ and $\nabla \cdot \mathbf{m}_n \rightarrow g \in L^2(\omega)$. Finally

$$(g, v)_{L^2} = \lim_{n \rightarrow \infty} (\nabla \cdot \mathbf{m}_n, v)_{L^2} = - \lim_{n \rightarrow \infty} (\mathbf{m}_n, \nabla v)_{L^2} = -(\mathbf{m}, \nabla v)_{L^2} \quad \forall v \in \mathcal{D}(\omega)$$

follows by continuity of the L^2 scalar product. \blacksquare

Proposition 2.11 ([GR86, Theorem 2.4]). *Let ω be a Lipschitz domain in \mathbb{R}^2 . The space $\mathcal{D}(\bar{\omega})^2$ is dense in $H^1(\nabla\cdot;\omega)$. ■*

Proposition 2.12 ([GR86, Theorem 2.5]). *The mapping $f_n : \mathbf{v} \mapsto \mathbf{v} \cdot \mathbf{n}|_\gamma$ defined on $\mathcal{D}(\bar{\omega})^2$ can be extended by continuity to a linear and continuous mapping from $H^1(\nabla\cdot;\omega)$ into $H^{-1/2}(\gamma)$. ■*

We are particularly interested in functions that satisfy $\mathbf{m} \cdot \mathbf{n} = 0$.

Proposition 2.13 ([GR86, Theorem 2.6]).

There holds $H_0^1(\nabla\cdot;\omega) = \ker(f_n) = \{\mathbf{m} \in H^1(\nabla\cdot;\omega) \mid \mathbf{m} \cdot \mathbf{n} = 0\}$. ■

One last step has to be done to define the appropriate space for the magnetization. Namely we have to allow $\nabla \cdot \mathbf{m} \in \tilde{H}^{-1/2}(\omega)$.

Definition. The energy space for the magnetization is defined by

$$\mathcal{H} = \overline{H_0^1(\nabla\cdot;\omega)}^{\|\cdot\|} \quad (2.18)$$

with

$$\|\mathbf{m}\|^2 := \|\mathbf{m}\|_{L^2(\omega)}^2 + \|\nabla \cdot \mathbf{m}\|_{\tilde{H}^{-1/2}(\omega)}^2. \quad (2.19)$$

By construction \mathcal{H} is a Hilbert space. □

Lemma 2.14. $\mathcal{D}(\omega)^2 \subseteq \mathcal{H}$ is a dense subspace.

Proof. Let $\mathbf{m} \in \mathcal{H}$ and $\varepsilon > 0$ be given. Then by definition of \mathcal{H} , we may choose $\mathbf{m}_1 \in H_0^1(\nabla\cdot;\omega)$ with $\|\mathbf{m} - \mathbf{m}_1\| \leq \frac{\varepsilon}{2}$. Since $\mathcal{D}(\omega)^2$ is dense in $H_0^1(\nabla\cdot;\omega)$, we may furthermore choose $\mathbf{m}_\mathcal{D} \in \mathcal{D}(\omega)$ such that $\|\mathbf{m}_1 - \mathbf{m}_\mathcal{D}\|_{H_0^1} \leq \frac{\varepsilon}{2}$. Then, the triangle inequality shows

$$\|\mathbf{m} - \mathbf{m}_\mathcal{D}\| \leq \|\mathbf{m} - \mathbf{m}_1\| + \|\mathbf{m}_1 - \mathbf{m}_\mathcal{D}\|.$$

Recall that for $v \in L^2$, it holds that

$$\begin{aligned} \|v\|_{\tilde{H}^{-1/2}} &= \sup_{w \in H^{1/2} \setminus \{0\}} \frac{\langle v, w \rangle}{\|w\|_{H^{1/2}}} \\ &\leq \sup_{w \in L^2 \setminus \{0\}} \frac{\langle v, w \rangle}{\|w\|_{L^2}} \leq \|v\|_{L^2}. \end{aligned}$$

Hence,

$$\begin{aligned} \|\mathbf{m} - \mathbf{m}_\mathcal{D}\| &\leq \|\mathbf{m} - \mathbf{m}_1\| + \left(\|\mathbf{m}_1 - \mathbf{m}_\mathcal{D}\|_{L^2(\omega)}^2 + \|\nabla \cdot (\mathbf{m}_1 - \mathbf{m}_\mathcal{D})\|_{\tilde{H}^{-1/2}(\omega)}^2 \right)^{1/2} \\ &\leq \|\mathbf{m} - \mathbf{m}_1\| + \|\mathbf{m}_1 - \mathbf{m}_\mathcal{D}\|_{H^1(\nabla\cdot;\omega)} \leq \varepsilon, \end{aligned}$$

which proves that any function in \mathcal{H} may be approximated up to some arbitrary tolerance $\varepsilon > 0$ by smooth functions. ■

Finally, we make a last remark on \mathcal{H} before analyzing the existence of minimizers \mathbf{m}^* in our function setting.

Lemma 2.15. *For all functions $\mathbf{m} \in \mathcal{H}$, it holds that $\langle \nabla \cdot \mathbf{m}, 1 \rangle_{\tilde{H}^{-1/2}(\omega) \times H^{1/2}(\omega)} = 0$.*

Proof. According to the Gauss divergence theorem there holds

$$\int_{\omega} \nabla \cdot \mathbf{m} \, dx = \int_{\gamma} \mathbf{m} \cdot \mathbf{n} \, ds$$

for $\mathbf{m} \in \mathcal{D}(\omega)^2$, where the right-hand side vanishes in our case. This, however, implies

$$(\nabla \cdot \mathbf{m}, 1)_{L^2(\omega)} = \langle \nabla \cdot \mathbf{m}, 1 \rangle_{\tilde{H}^{-1/2}(\omega) \times H^{1/2}(\omega)} = 0$$

for $\mathbf{m} \in \mathcal{D}(\omega)^2$. Due to the density of $\mathcal{D}(\omega)^2$ in \mathcal{H} , the statement immediately follows by continuity. ■

2.5.2 The Beppo-Levi class $B_1^2(\mathbb{R}^3)$ for the magnetostatic potential u

Given $\mathbf{m} \in \mathcal{H}$, we may represent u as the simple-layer potential of $\nabla \cdot \mathbf{m} \in \tilde{H}^{-1/2}(\omega)$. From Proposition 2.5, we immediately conclude $u \in H_{loc}^1(\mathbb{R}^3) \supseteq \text{range}(S)$. However, $H_{loc}^1(\mathbb{R}^3)$ is not a normed space. Also, the fact $\nabla u \in L^2(\mathbb{R}^3)^3$ implies further regularity and is necessary for the energy $e(\mathbf{m})$ defined in (2.2) to be finite. We now establish an appropriate Hilbert space for the magnetostatic potential u .

Definition. We define the set

$$\tilde{B}_1^2(\mathbb{R}^3) := \{u \in H_{loc}^1(\mathbb{R}^3) \mid \nabla u \in L^2(\mathbb{R}^3)^3\} \quad (2.20)$$

associated with the seminorm $\|u\|_{B_1^2(\mathbb{R}^3)} := \|\nabla u\|_{L^2(\mathbb{R}^3)^3}$. Further, we define the Beppo-Levi space

$$B_1^2(\mathbb{R}^3) := \tilde{B}_1^2(\mathbb{R}^3) / \mathbb{R} \quad (2.21)$$

by factoring out the constant functions. Note that $\|\cdot\|_{B_1^2(\mathbb{R}^3)}$ now in fact is norm. □

Proposition 2.16 ([DL54, Corollaire 1.1, Theoreme 2.1]). B_1^2 is a Hilbert space. $\mathcal{D}(\mathbb{R}^3)$ is a dense subspace of $B_1^2(\mathbb{R}^3)$.

Lemma 2.17. *There is a continuous linear lifting operator $L : H^{1/2}(\omega) \rightarrow B_1^2(\mathbb{R}^3)$, i.e. for $v \in H^{1/2}(\omega)$ with $v \neq 0$ it holds that*

$$v = (Lv)|_{\omega} \quad \text{and} \quad \|\nabla(Lv)\|_{L^2(\mathbb{R}^3)^3} \leq C \|v\|_{H^{1/2}(\omega)}$$

with $C > 0$ the operator norm of L .

Proof. First, choose some bounded Lipschitz domain $G \subseteq (\mathbb{R}^3)$ with $\omega \subseteq \partial G$. Let $v \in H^{1/2}(\omega)$ be given. Then, from [McL00, Theorem 3.37], we conclude existence of some extension $\tilde{v} \in H^1(G)$ with $\|\tilde{v}\|_{H^1(G)} \lesssim \|v\|_{H^{1/2}(\omega)}$. From [DL54, Theoreme 8.1], we finally obtain the full continuous extension to $B_1^2(\mathbb{R}^3)$. ■

2.6 Well posedness of the thin-film minimization problem

2.6.1 Existence and uniqueness of the magnetostatic potential

Definition. We define the simple-layer operator

$$V = tS, \quad V\varphi(x) = \frac{1}{4\pi} \int_{\omega} \frac{\varphi(y)}{\|x - y\|} ds_y \quad \forall x \in \omega. \quad (2.22)$$

Due to the mapping properties of S and t , it holds that $V \in L(\tilde{H}^{-1/2}(\omega); H^{1/2}(\omega))$. \square

We recall the variational formulation (2.4) of the magnetostatic Maxwell equation. As discussed above, $\mathbf{m} \in \mathcal{H}$ satisfies all constraints and the necessary regularity. Therefore, (2.4) may be stated as

$$(\nabla u, \nabla v)_{L^2(\mathbb{R}^3)^3} = -\langle \nabla \cdot \mathbf{m}, v \rangle_{\tilde{H}^{-1/2}(\omega) \times H^{1/2}(\omega)} \quad \text{for all } v \in \mathcal{D}(\mathbb{R}^3) \quad (2.23)$$

in our functional setting. First, we prove the unique existence of a magnetostatic potential u .

Theorem 2.18.

- (i) Given $\mathbf{m} \in \mathcal{H}$, there is a uniquely determined $u \in B_1^2(\mathbb{R}^3)$ with (2.23).
- (ii) Equation (2.23) holds with $\mathcal{D}(\mathbb{R}^3)$ replaced by the full space $B_1^2(\mathbb{R}^3)$.
- (iii) The mapping $\mathcal{P} : \mathcal{H} \rightarrow L^2(\mathbb{R}^3)^3$, which maps \mathbf{m} onto the corresponding stray field $\mathcal{P}(\mathbf{m}) := -\nabla u$, is a linear and continuous operator.
- (iv) For $\mathbf{m}, \tilde{\mathbf{m}} \in \mathcal{H}$ there holds $\langle \mathcal{P}\mathbf{m}, \mathcal{P}\tilde{\mathbf{m}} \rangle_{L^2(\mathbb{R}^3)^3} = \langle \nabla \cdot \mathbf{m}, V(\nabla \cdot \tilde{\mathbf{m}}) \rangle_{\tilde{H}^{-1/2}(\omega) \times H^{1/2}(\omega)}$.
- (v) In particular, there holds $\|\mathcal{P}\mathbf{m}\|_{L^2(\mathbb{R}^3)^3}^2 = \|\nabla \cdot \mathbf{m}\|_V^2 = \langle \nabla \cdot \mathbf{m}, V(\nabla \cdot \mathbf{m}) \rangle \sim \|\nabla \cdot \mathbf{m}\|_{\tilde{H}^{-1/2}(\omega)}^2$.

Proof. Let $\mathbf{m} \in \mathcal{H}$ be fixed. We first consider $F_{\mathbf{m}}(v) := \langle \nabla \cdot \mathbf{m}, v \rangle_{\tilde{H}^{-1/2}(\omega) \times H^{1/2}(\omega)}$ for arbitrary $v \in \mathcal{D}(\mathbb{R}^3)$. According to Lemma 2.15, it holds that

$$\langle \nabla \cdot \mathbf{m}, v \rangle_{\tilde{H}^{-1/2}(\omega) \times H^{1/2}(\omega)} = \langle \nabla \cdot \mathbf{m}, v - \lambda \rangle \quad \text{for all constants } \lambda \in \mathbb{R}.$$

We consider the enriched domain $\hat{\omega} := \omega \times [0, 1]$. With $v \in \mathcal{D}(\mathbb{R}^3)$ and $\lambda := (1/|\hat{\omega}|) \int_{\hat{\omega}} v dx$, the continuity of the trace operator stated in Proposition 2.1 and a Poincaré inequality (2.5) show

$$\|v - \lambda\|_{H^{1/2}(\omega)} \lesssim \|v - \lambda\|_{H^1(\hat{\omega})} \lesssim \|\nabla v\|_{L^2(\hat{\omega})^3} \leq \|\nabla v\|_{L^2(\mathbb{R}^3)^3}.$$

This proves that $F_{\mathbf{m}}$ defines a linear and continuous functional $F_{\mathbf{m}} : \mathcal{D}(\mathbb{R}^3) \rightarrow \mathbb{R}$ with respect to $\|\cdot\|_{B_1^2(\mathbb{R}^3)}$ and operator norm $\|F_{\mathbf{m}}\| \leq \|\nabla \cdot \mathbf{m}\|_{\tilde{H}^{-1/2}(\omega)}$. Since $\mathcal{D}(\mathbb{R}^3)$ is dense in $B_1^2(\mathbb{R}^3)$ the functional $F_{\mathbf{m}}$ may be continuously extended, conserving the upper bound for the operator norm, to the entire Beppo-Levi space. Since the left-hand side of (2.23) is the scalar product of $B_1^2(\mathbb{R}^3)$ the variational formulation may be extended to the full space $B_1^2(\mathbb{R}^3)$. This proves (ii).

We stress that the Riesz theorem provides the unique existence of a solution $u \in B_1^2(\mathbb{R}^3)$ since (2.23) may be written as

$$(u, v)_{B_1^2(\mathbb{R}^3)} = F_{\mathbf{m}}(v) \quad \text{for all } v \in B_1^2(\mathbb{R}^3),$$

which yields statement (i).

The Riesz theorem furthermore implies

$$\|\mathcal{P}\mathbf{m}\|_{L^2(\mathbb{R}^3)^3} = \|u\|_{B_1^2(\mathbb{R}^3)} = \|F_{\mathbf{m}}\| \lesssim \|\nabla \cdot \mathbf{m}\|_{\tilde{H}^{-1/2}(\omega)}.$$

In particular the mapping $\mathcal{P} : \mathcal{H} \rightarrow L^2(\mathbb{R}^3)^3$ is well defined and continuous. Linearity follows from the composition $\mathcal{P} : \mathbf{m} \mapsto \nabla \cdot \mathbf{m} \mapsto u \mapsto -\nabla u$, which finally proves (iii).

Now, we prove the converse estimate $\|\nabla \cdot \mathbf{m}\|_{\tilde{H}^{-1/2}(\omega)} \lesssim \|F_{\mathbf{m}}\|$. Lemma 2.17 and (2.23) imply

$$\frac{|\langle \nabla \cdot \mathbf{m}, v \rangle_{\tilde{H}^{-1/2}(\omega) \times H^{1/2}(\omega)}|}{\|v\|_{H^{1/2}(\omega)}} \leq C \frac{|\langle \mathcal{P}\mathbf{m}, \nabla(Lv) \rangle_{L^2(\mathbb{R}^3)^3}|}{\|\nabla(Lv)\|_{L^2(\mathbb{R}^3)^3}} \leq C \|\mathcal{P}\mathbf{m}\|_{L^2(\mathbb{R}^3)^3}$$

for arbitrary $v \in H^{1/2}(\omega) \setminus \{0\}$. Taking the supremum over all $v \in H^{1/2}(\omega) \setminus \{0\}$ we obtain

$$\|\nabla \cdot \mathbf{m}\|_{\tilde{H}^{-1/2}(\omega)} \lesssim \|\mathcal{P}\mathbf{m}\|_{L^2(\mathbb{R}^3)^3}. \quad (2.24)$$

For $\mathcal{P}\tilde{\mathbf{m}} = -\nabla\tilde{u}$, the representation $\tilde{u} = S(-\nabla \cdot \tilde{\mathbf{m}})$ and the variational equality (2.23) for $\mathcal{P}\mathbf{m} = -\nabla u$ imply

$$(\mathcal{P}\mathbf{m}, \mathcal{P}\tilde{\mathbf{m}})_{L^2(\mathbb{R}^3)^3} = -\langle \nabla \cdot \mathbf{m}, \tilde{u} \rangle_{\tilde{H}^{-1/2}(\omega) \times H^{1/2}(\omega)} = -\langle \nabla \cdot \mathbf{m}, V(-\nabla \cdot \tilde{\mathbf{m}}) \rangle_{\tilde{H}^{-1/2}(\omega) \times H^{1/2}(\omega)}.$$

The choice of $\tilde{\mathbf{m}} = \mathbf{m}$ yields $(\mathcal{P}\mathbf{m}, \mathcal{P}\tilde{\mathbf{m}})_{L^2(\mathbb{R}^3)^3} = \|\nabla \cdot \mathbf{m}\|_V^2$ which finally concludes the proof. \blacksquare

2.6.2 Existence and uniqueness of minimizer $\mathbf{m}^* \in \mathcal{H}$

We now use our functional setting to state the considered problem in static thin-film micromagnetics.

Definition. We define the set of admissible magnetizations

$$\mathcal{A} := \{\mathbf{m} \in \mathcal{H} \mid |\mathbf{m}| \leq 1\}. \quad (2.25)$$

\square

Reduced Thin-Film Problem (M). We seek to find a minimizer $\mathbf{m}^* \in \mathcal{A}$ of the energy

$$e(\mathbf{m}) = \frac{1}{2} \|\nabla \cdot \mathbf{m}\|_V^2 + \frac{q}{2} \|\mathbf{m}_2\|_{L^2(\omega)}^2 - (\mathbf{f}, \mathbf{m})_{L^2(\omega)^2}. \quad (2.26)$$

The following observation follows immediately from the representation (2.26).

Corollary 2.19. *The energy functional $e : \mathcal{H} \rightarrow \mathbb{R}$ is convex and continuous.* \blacksquare

Our aim is to prove the existence of a minimizer $\mathbf{m}^* \in \mathcal{A}$ and therefore the solvability of our problem. To that end, we use well-known techniques from the theory of calculus of variations. The direct method is, e.g., described in depth in [Dac89]. Furthermore, we would like to provide uniqueness of \mathbf{m}^* . This, however, is non-trivial since the energy $e(\mathbf{m})$ does not depend on the first in-plane component \mathbf{m}_1 of the magnetization. The density of $\mathcal{D}(\omega)^2 \subseteq \mathcal{H}$, i.e. our convenient function space setting, and mollifier techniques even allow us to finally conclude uniqueness, provided $q > 0$.

Note that the magnetostatic potential u is uniquely determined by $\nabla \cdot \mathbf{m}$. We state the well-posedness of our problem first in the following sense: All minimizers $\mathbf{m}^* \in \mathcal{A}$ induce the same uniquely determined magnetostatic potential u^* . Provided $q > 0$, they share the same in-plane component \mathbf{m}_2^* .

Lemma 2.20. *The set \mathcal{A} is non-empty, convex and closed with respect to \mathcal{H} .*

Proof. We only need to prove the closedness of \mathcal{A} since convexity and non-emptiness is clear by definition. Let $(\mathbf{m}_k)_{k \in \mathbb{N}} \subseteq \mathcal{A}$ be a convergent sequence with limit $\mathbf{m} \in \mathcal{H}$. In particular, convergence in \mathcal{H} implies convergence in the L^2 -sense, i.e. $\mathbf{m}_k \rightarrow \mathbf{m} \in L^2(\omega)^2$. Therefore, at least a subsequence converges pointwise almost everywhere towards \mathbf{m} . This proves $|\mathbf{m}| \leq 1$ almost everywhere, i.e. $\mathbf{m} \in \mathcal{A}$. \blacksquare

Theorem 2.21. *There is a minimizer $\mathbf{m}^* \in \mathcal{A}$ of (2.26). Moreover, $\mathcal{P}\mathbf{m}$ and $q\mathbf{m}_2$ are uniquely determined, i.e. for any minimizers $\mathbf{m}^*, \tilde{\mathbf{m}}^* \in \mathcal{A}$ of (2.26) there holds $\mathcal{P}\mathbf{m}^* = \mathcal{P}\tilde{\mathbf{m}}^*$ as well as $q\mathbf{m}_2^* = q\tilde{\mathbf{m}}_2^*$.*

Proof. Let $(\mathbf{m}_k)_{k \in \mathbb{N}} \subseteq \mathcal{A}$ be a sequence with

$$\lim_{k \rightarrow \infty} e(\mathbf{m}_k) = \inf_{\mathbf{m} \in \mathcal{A}} e(\mathbf{m}) =: M.$$

Plugging in $\mathbf{m} = 0 \in \mathcal{A}$, we obtain $M \leq 0$. This implies for arbitrary $\varepsilon > 0$

$$e(\mathbf{m}_k) = \frac{1}{2} \|\mathcal{P}\mathbf{m}_k\|_{L^2(\mathbb{R}^3)}^2 + \frac{q}{2} \|\mathbf{m}_{k,2}\|_{L^2(\omega)}^2 - \langle \mathbf{f}, \mathbf{m}_k \rangle_{L^2(\omega)} \leq \frac{\varepsilon}{2}$$

for k sufficiently large. Using the Cauchy-Schwarz inequality and dropping the anisotropy energy one obtains

$$\|\mathcal{P}\mathbf{m}_k\|_{L^2(\mathbb{R}^3)}^2 \leq 2 \|\mathbf{f}\|_{L^2(\omega)^2} \|\mathbf{m}_k\|_{L^2(\omega)^2} + \varepsilon.$$

From the definition of \mathcal{A} , we derive $\|\mathbf{m}_k\|_{L^2(\omega)^2} \leq |\omega|^{1/2}$. The equivalence $\|\mathcal{P}\mathbf{m}\|_{L^2(\mathbb{R}^3)} \sim \|\nabla \cdot \mathbf{m}\|_{\tilde{H}^{-1/2}}$ therefore implies that $(\mathbf{m}_k)_{k \in \mathbb{N}}$ is bounded with respect to the \mathcal{H} -norm, namely,

$$\|\mathbf{m}_k\|^2 \lesssim \|\mathbf{m}_k\|_{L^2(\omega)^2}^2 + \|\mathcal{P}\mathbf{m}_k\|_{L^2(\mathbb{R}^3)}^2 \leq |\omega| + 2 \|\mathbf{f}\|_{L^2(\omega)^2} |\omega|^{1/2} + \varepsilon.$$

Since the sequence $(\mathbf{m}_k)_{k \in \mathbb{N}}$ is bounded we may assume that it has a weak limit $\mathbf{m} \in \mathcal{H}$. The set \mathcal{A} is convex and closed, hence it is closed with respect to the weak topology in \mathcal{H} . This implies $\mathbf{m} \in \mathcal{A}$. Moreover, a convex and continuous functional is also weakly lower semicontinuous [Dac89, Chapter 3 Theorem 1.2] so that

$$e(\mathbf{m}) \leq \liminf_{k \rightarrow \infty} e(\mathbf{m}_k). \quad (2.27)$$

Altogether $\mathbf{m} \in \mathcal{A}$ is a minimizer of $e(\cdot)$ in \mathcal{A} . Finally, we observe from the representation (2.26) that the energy functional is strictly convex in $\mathcal{P}\mathbf{m}$ and $q\mathbf{m}_2$. Therefore, the stray field and the second component of a minimizing sequence are uniquely determined. \blacksquare

Remark. Note that the proof of Theorem 2.21 only used some properties of the functional $e(\cdot)$ as well as the closedness and convexity of \mathcal{A} . Any finite dimensional subspace $X_h \leq \mathcal{H}$ of the full energy space is obviously closed and convex. In particular the set $\mathcal{A}_h := (\mathcal{A} \cap X_h)$ is closed and convex as well. This, however, implies the existence of a minimizer \mathbf{m}_h^* within the discrete space X_h that satisfies the convex constraint $|\mathbf{m}| \leq 1$. Furthermore, the stray field $\mathcal{P}\mathbf{m}_h$ and the second component $\mathbf{m}_{h,2}$ of two discrete minimizers $\mathbf{m}_h^{*,1}, \mathbf{m}_h^{*,2} \in \mathcal{A}_h$ must coincide. \square

We now state that $\|\mathbf{m}\|^2 := \|\mathbf{m}_2\|_{L^2(\omega)}^2 + \|\nabla \cdot \mathbf{m}\|_{\tilde{H}^{-1/2}(\omega)}^2$ is definite and therefore a norm on \mathcal{H} . This is the main theorem of the present section as it provides uniqueness of $\mathbf{m}^* \in \mathcal{A}$ as immediate consequence.

Theorem 2.22. *Let $\mathbf{m} \in \mathcal{H}$ with $\nabla \cdot \mathbf{m} = 0 \in \tilde{H}^{-1/2}$ and $\mathbf{m}_2 = 0 \in L^2$, then $\mathbf{m} = 0 \in \mathcal{H}$.*

Proof. First we show $\nabla \cdot \mathbf{m} = 0 \in L^2$. This follows immediately from $\langle \nabla \cdot \mathbf{m}, v \rangle = 0$ for all $v \in H^{1/2}(\omega) \supseteq \mathcal{D}(\omega)$ and the fundamental theorem of calculus of variations. Therefore \mathbf{m} is an element of the space $H^1(\nabla \cdot; \omega)$. Furthermore the extension

$$\hat{\mathbf{m}} = \begin{cases} \mathbf{m}(x) & x \in \omega \\ 0 & \text{otherwise} \end{cases}$$

is element of $H^1(\nabla\cdot; \mathbb{R}^2)$ with $\nabla \cdot \widehat{\mathbf{m}} = 0$ since

$$\begin{aligned} \int_{\mathbb{R}^2} \widehat{\mathbf{m}} \cdot \nabla \varphi \, dx &= \int_{\omega} \widehat{\mathbf{m}} \cdot \nabla \varphi \, dx + \int_{\mathbb{R}^2 \setminus \omega} \widehat{\mathbf{m}} \cdot \nabla \varphi \, dx \\ &= - \int_{\omega} \nabla \cdot \mathbf{m} \varphi \, dx + \int_{\gamma} (\mathbf{m} \cdot \mathbf{n}) \varphi \, dx - \int_{\mathbb{R}^2 \setminus \omega} \nabla \cdot \widehat{\mathbf{m}} \varphi \, dx - \int_{\gamma} (\widehat{\mathbf{m}} \cdot \mathbf{n}) \varphi \, dx = 0. \end{aligned}$$

Next, we will show that $\widehat{\mathbf{m}} = 0 \in L^2(\mathbb{R}^2)$ which in turn implies $\mathbf{m} = 0 \in \mathcal{H}$. For this, we define the mollifier

$$\begin{aligned} \psi_1 &= \begin{cases} \frac{e^{\frac{1}{|x|^2-1}}}{\int_{[-1,1]} e^{\frac{1}{|x|^2-1}} \, dx}, & |x| < 1 \\ 0, & \text{otherwise} \end{cases} \\ \psi_\varepsilon &= \varepsilon^{-2} \psi_1\left(\frac{x}{\varepsilon}\right) \end{aligned}$$

which satisfies $\psi \in \mathcal{D}(\mathbb{R}^2)$ with $\text{supp}(\psi) \subseteq [-\varepsilon, \varepsilon]^2$. Moreover, it holds that [McL00, Theorem 3.4]

$$\psi_\varepsilon \star \widehat{\mathbf{m}} \longrightarrow \widehat{\mathbf{m}} \in L^2 \text{ as } \varepsilon \rightarrow 0 \quad (2.28)$$

componentwise, where \star denotes the convolution of functions. From [Pra03, Lemma 2.13], we know

$$\begin{aligned} \psi_\varepsilon \star \widehat{\mathbf{m}} &\in H^1(\nabla\cdot; \mathbb{R}^2) \cap \mathcal{D}(\mathbb{R}^2)^2, \\ \nabla \cdot (\psi_\varepsilon \star \widehat{\mathbf{m}}) &= \psi_\varepsilon \star (\nabla \cdot \widehat{\mathbf{m}}) = 0. \end{aligned}$$

From $\widehat{\mathbf{m}}_2 = 0 \in L^2(\mathbb{R}^2)$, we additionally know

$$\psi_\varepsilon \star \widehat{\mathbf{m}} = \begin{pmatrix} \psi_\varepsilon \star \widehat{\mathbf{m}}_1 \\ 0 \end{pmatrix}$$

which trivially implies $\frac{\partial(\psi_\varepsilon \star \widehat{\mathbf{m}}_2)}{\partial x_2} = 0$. Together with $\nabla \cdot (\psi_\varepsilon \star \widehat{\mathbf{m}}) = 0$ this means

$$\frac{\partial(\psi_\varepsilon \star \widehat{\mathbf{m}}_1)}{\partial x_1} = 0.$$

Since $(\psi_\varepsilon \star \widehat{\mathbf{m}}_1) \in \mathcal{D}(\mathbb{R}^2)$ is a smooth function with compact support, $\frac{\partial(\psi_\varepsilon \star \widehat{\mathbf{m}}_1)}{\partial x_1} = 0$ already implies $\psi_\varepsilon \star \widehat{\mathbf{m}}_1 = 0$. We conclude $0 = \psi_\varepsilon \star \widehat{\mathbf{m}}_1$, whence $\widehat{\mathbf{m}}_1 = 0$ according to (2.28). ■

Corollary 2.23 (Well posedness of the reduced thin-film problem (M)). *The model Problem (M) has a uniquely determined minimizer $\mathbf{m}^* \in \mathcal{H}$, provided $q > 0$.*

Proof. According to Theorem 2.21 there exists a minimizer \mathbf{m}^* , and the quantities $\nabla \cdot \mathbf{m}^*$ and \mathbf{m}_2^* are uniquely determined. Let \mathbf{w}^* be a minimizer. Then $\mathbf{m}^* - \mathbf{w}^* \in \mathcal{H}$ satisfies $\nabla \cdot (\mathbf{m}^* - \mathbf{w}^*) = 0$ as well as $\mathbf{m}_2^* - \mathbf{w}_2^* = 0$. From Theorem 2.22, we therefore conclude $(\mathbf{m}^* - \mathbf{w}^*) = 0 \in \mathcal{H}$ which proves $\mathbf{w}^* = \mathbf{m}^*$. ■

Definition. We define the energy (semi-)norm

$$\|\mathbf{m}\| := \sqrt{q \|\mathbf{m}_2\|_{L^2}^2 + \|\nabla \cdot \mathbf{m}\|_V^2}$$

on \mathcal{H} . We stress that this is indeed a norm according to Theorem 2.22, provided $q > 0$. Moreover there obviously holds $\|\mathbf{m}\| \lesssim \|\mathbf{m}\|$. In the following chapter where we provide numerical algorithms to compute \mathbf{m}^* approximately, all error estimates are given in this energy norm. □

Chapter 3

Discretization, numerical solution, and error analysis

In order to solve the thin-film minimization problem (M) numerically, two steps have to be performed: First, the energy space of the magnetization needs to be discretized. Second, the resulting discrete minimization problem (M_h) needs to be solved.

Most of the literature addresses the problem of solving finite dimensional minimization problems, e.g. [NW99], whereas the monograph [IK08], for instance, collects the state of the art of Lagrangian multiplier methods in a continuous context. For quadratic programming, some publications such as [DS05] are available, where super-linear convergence of an active set algorithm is proven. However, the latter work considers box constraints. In contrast, we are dealing with a 2-dimensional vector field under a 1-dimensional non-linear constraint, which seems to be a significant enhancement of the problem's complexity. Therefore, only penalty or interior point methods, see [NW99], seem to be suitable in our case. These approaches include some parameter $\varepsilon > 0$ which introduces an additional error of the minimization algorithm. Since we are also faced with a discretization error, two non-trivial questions arise: First, it is not trivially clear that one may expect convergence of the penalized and discretized scheme as $(h, \varepsilon) \rightarrow (0, 0)$. Second, the question how to balance the penalty and the discretization error arises. The choice of $\varepsilon \ll 1$ usually yields high computational cost due to ill-conditioning and slow convergence of the underlying Newton-method. In this chapter, we address this problem and provide a mathematically justified choice of ε and h by establishing a priori convergence results.

In Section 3.1, we propose our discretization scheme which leads to a discrete minimization problem (M_h) . Existence and uniqueness of solutions of (M_h) is established in the same way as for the continuous problem (M) . Since the energy space \mathcal{H} of the magnetization demands $\nabla \cdot \mathbf{m} \in \tilde{H}^{-1/2}(\omega)$, a lowest-order discretization has to provide $\nabla \cdot \mathbf{m}_h \in \mathcal{P}^0(\mathcal{T}_h)$. Here, $\mathcal{P}^0(\mathcal{T}_h)$ denotes the space of piecewise constant functions over some mesh \mathcal{T}_h . The natural choice is that of Raviart-Thomas finite elements $RT^0(\mathcal{T}_h)$ [RT77], which are often used, e.g., in the context of mixed formulation of the Poisson equation [Bra97]. This discretization scheme has also been proposed in the dissertation [Drw08], where the thin-film minimization problem (M) is solved by first choosing the discrete ansatz space and then applying an interior point method. However, neither the choice of the barrier parameter with respect to the discretization nor the well-posedness of the discrete system are discussed in [Drw08]. Moreover, the work [Drw08] is essentially dedicated to the case of soft ferromagnetic films only, i.e. $q = 0$.

Even though we use the same discrete ansatz space for the magnetization, our approach for solving the problem (M) numerically is quite different from that in [Drw08]: In Section 3.1.3, we propose a penalization scheme. Instead of satisfying the constraint $|\mathbf{m}| \leq 1$, we introduce a new energy contribution $\frac{1}{2\varepsilon} \|(|\mathbf{m}| - 1)_+\|_{L^2(\omega)}^2$ where $(\cdot)_+$ denotes the positive part, i.e. $(|\mathbf{m}(x)| - 1)_+ = 0$ at points $x \in \omega$ where $|\mathbf{m}(x)| \leq 1$. This leads to a discrete penalized, but unconstrained minimization problem (M_h^ε) . Convergence of this kind of penalty methods is well established in the literature, see e.g. [NW99]. Let \mathbf{m}_h^ε denote the minimizer of (M_h^ε) and \mathbf{m}_h^0 denote the minimizer of (M_h) . Then there holds convergence $\mathbf{m}_h^\varepsilon \rightarrow \mathbf{m}_h^0$ in a weak sense for any fixed $h \geq 0$. However, from this it is not clear that convergence $\mathbf{m}_h^\varepsilon \rightarrow \mathbf{m}^*$ with \mathbf{m}^* the minimizer of (M) follows as $(h, \varepsilon) \rightarrow (0, 0)$. This result, which we failed to find in the literature, is established in Section 3.2 in an abstract framework and contributes to the understanding of the numerical solution of infinite dimensional minimization problems.

Finally, after studying convergence of our proposed solution scheme, some statement on the rate of convergence still remains open. In Section 3.3 we close this gap. First, we state the Euler-Lagrange equations for the continuous model problem (M) . We stress that standard Lagrange multiplier techniques, e.g. as described in [IK08], are not sufficient for our purposes, since we need further geometrical and smoothness properties of the Lagrangian multiplier. In [De 93], the Euler-Lagrange equations for the large-body limit, a related problem in an L^2 setting, are established. With Theorem 3.18, we prove a similar result. On the one hand, we follow the main idea of [De 93], on the other hand, the proof in our case involves some different techniques. This is due to the fact that our dual space \mathcal{H}^* contains distributions, whereas the dual space in the case of the large-body limit is simply $L^2(\omega)$. Having established the Euler-Lagrange equations, we proceed to prove an a priori convergence order $\|\mathbf{m}^* - \mathbf{m}_h^\varepsilon\| = \mathcal{O}(\sqrt{h} + \sqrt{\varepsilon})$ under suitable regularity conditions. This finally suggests the choice of $\varepsilon = h$ for uniform meshes.

From the well-known generic singular behavior of solutions of the weakly singular boundary integral equation associated with the Laplacian in 3D [ESAES90], we expect to observe strong edge and corner singularities of $\nabla \cdot \mathbf{m}^*$. Also, geometric effects and non-smooth applied field \mathbf{f} may cause singularities of \mathbf{m}^* . In Section 3.4.2, we propose a heuristic adaptive algorithm to steer local mesh-refinements as well as a local penalty scheme. For the estimation of the discretization error, we use the easy-to-implement, however computationally intense, $h - h/2$ based error estimation strategy. For steering the local penalty parameter, we use the penalty energy, which is proven to vanish as $\varepsilon \rightarrow 0$ in Lemma 3.9. Hence, we refine, i.e. reduce by a fraction, ε locally on elements $T \in \mathcal{T}$, where $\|(|\mathbf{m}| - 1)_+\|_{L^2(\omega)}^2$ is large.

3.1 Discretization and penalization

The lowest-order Raviart-Thomas finite elements were first introduced in [RT77]. Having defined regular triangulations of our domain, we follow the lines of [RT77] and [BC05] to define the Raviart-Thomas finite elements and collect their crucial properties. Finally, we give an explicit basis suitable for an implementation.

In the second subsection, we pose the discrete minimization problem (M_h) . First, we note that existence and uniqueness of minimizers for the discrete problem follows by the same arguments as for the continuous problem. Then, using standard finite element techniques, we give an a priori error analysis. We stress, however, that this analysis is somewhat artificial, as the minimizer \mathbf{m}_h^* of the problem (M_h) , in general, cannot be computed analytically.

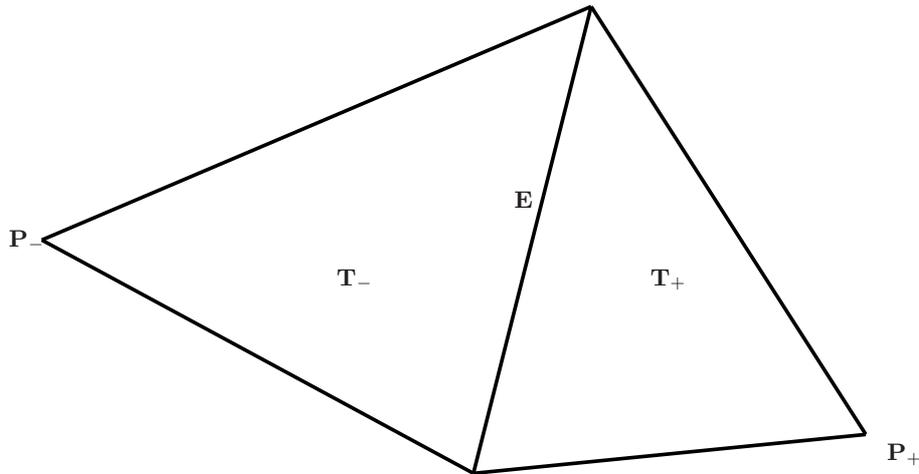


Figure 3.1: Each interior edge E belongs to precisely two triangles T_+ and T_- . The points opposite of E are denoted by P_+ and P_- , respectively.

Finally, in the third subsection, we propose a penalty method which leads to a numerically solvable unconstrained (discrete) penalized problem (M_h^ε) . Well-posedness follows from direct method of calculus of variations due to the coercivity of the modified energy functional $e_\varepsilon(\cdot)$.

3.1.1 The space of lowest-order Raviart-Thomas finite elements

Definition. We call a partition $\mathcal{T} = \{T_1, \dots, T_N\}$ of the domain ω a regular triangulation if it satisfies the following conditions:

- Each element T is a non-degenerate and closed triangle,
- \mathcal{T} covers ω , i.e. $\bar{\omega} = \bigcup_{T \in \mathcal{T}} T$,
- The intersection $T_i \cap T_j$, for $i \neq j$, is either empty, a common vertex, or a common edge.

The global mesh-size h is defined by $h = \max_{T \in \mathcal{T}} \text{diam}(T)$. Moreover, the set of all edges of a triangulation is denoted by \mathcal{E} and \mathcal{E}_ω is the set of all interior edges. \square

Definition. For a given regular triangulation \mathcal{T} of ω , we define the space of lowest order Raviart-Thomas finite elements by

$$RT^0(\mathcal{T}) = \{\mathbf{m}_h \in \mathcal{P}^1(\mathcal{T}) \mid [\mathbf{m}_h \cdot \mathbf{n}_E]_E = 0 \forall E \in \mathcal{E}_\omega \text{ and } \mathbf{m}_h \cdot \mathbf{n} = 0 \text{ on } \gamma\},$$

where $\mathcal{P}^1(\mathcal{T})$ denotes the space of piecewise linear and discontinuous functions, \mathbf{n}_E denotes a normal vector on the edge E , and $[\cdot]_E$ denotes the jump across an edge of the triangulation. \square

We stress that the crucial property $[\mathbf{m}_h \cdot \mathbf{n}_E]_E$ ensures the $H_0^1(\nabla \cdot; \omega)$ conformity of the discrete space $RT^0(\mathcal{T})$. Since $H_0^1(\nabla \cdot; \omega) \subseteq \mathcal{H}$, the set

$$\mathcal{A}_h := \{\mathbf{m}_h \in RT^0(\mathcal{T}) \mid |\mathbf{m}_h| \leq 1 \text{ a.e.}\}$$

yields a conforming discretization of our admissible set \mathcal{A} .

The given definition of $RT^0(\mathcal{T})$ describes the properties of the functions involved. However, in order to implement numerical algorithms using Raviart-Thomas finite elements, it is necessary to operate on a set of basis functions. Each interior edge $E \in \mathcal{E}_\omega$ belongs to precisely two elements T_+ and T_- . Let P_+ and P_- denote the vertices of T_+ and T_- , respectively, opposite E , i.e. $T_\pm = \text{conv}\{E \cup \{P_\pm\}\}$, see Figure 3.1. For each $E \in \mathcal{E}_\omega$, we define

$$\psi_E = \begin{cases} \pm \frac{|E|}{2|T_\pm|} (x - P_\pm), & \text{for } x \in T_\pm \\ 0, & \text{elsewhere} \end{cases} \quad (3.1)$$

and notice that the jump $[\psi_E \cdot \mathbf{n}]$ across any edge vanishes. This implies $\psi_E \in RT^0(\mathcal{T})$. Moreover, it can be shown that the set

$$\mathcal{B} = \{\psi_E \mid E \in \mathcal{E}_\omega\}$$

is a basis of $RT^0(\mathcal{T})$, cf. [GR86].

Obviously $RT^0(\mathcal{T})$ satisfies $(\nabla \cdot) : RT^0(\mathcal{T}) \rightarrow \mathcal{P}^0(\mathcal{T})$, where $\mathcal{P}^0(\mathcal{T})$ denotes the space of piecewise constant functions over the mesh \mathcal{T} . Moreover it even holds that the mapping $(\nabla \cdot) : RT^0(\mathcal{T}) \rightarrow \mathcal{P}_*^0(\mathcal{T})$ is onto [Bra97, Hilfssatz 5.4], where $\mathcal{P}_*^0(\mathcal{T})$ denotes the space of piecewise constants with vanishing integral mean.

In order to obtain a priori convergence rates, we will need some interpolation operator $\Pi_h : H^1(\omega) \rightarrow RT^0(\mathcal{T}_h)$ with some approximation property with respect to the mesh-size h .

Theorem 3.1. *The operator $\Pi_h : H^1(\omega)^2 \rightarrow RT^0(\mathcal{T})$ characterized by*

$$\Pi_h \mathbf{m}|_{T \in \mathcal{T}_h} \text{ satisfies } \int_E (\mathbf{m} - \Pi_h \mathbf{m}) \cdot \mathbf{n} \, ds = 0 \text{ for all } E \in (\mathcal{E} \cap \partial T)$$

is a projection from $H^1(\omega)^2$ onto $RT^0(\mathcal{T}_h)$. Let $\Pi_h^{L^2}$ denote the L^2 orthogonal projection onto $\mathcal{P}^0(\mathcal{T}_h)$. Then, it holds that

$$\nabla \cdot (\Pi_h \mathbf{m}) = \Pi_h^{L^2} (\nabla \cdot \mathbf{m}). \quad (3.2)$$

Furthermore, given $\mathbf{m} \in H^1(\omega)$ with $\nabla \cdot \mathbf{m} \in H^1(\omega)$, there hold the approximation properties

$$\|\mathbf{m} - \Pi_h \mathbf{m}\|_{L^2(\omega)} \leq C_1 h \|\mathbf{m}\|_{H^1(\omega)}, \quad (3.3)$$

$$\|\nabla \cdot (\mathbf{m} - \Pi_h \mathbf{m})\|_{\tilde{H}^{-1/2}(\omega)} \leq C_2 h^{3/2} \|\nabla \cdot \mathbf{m}\|_{H^1(\omega)}. \quad (3.4)$$

Proof. The first estimate (3.3) follows from [Bra97, Hilfssatz 5.5], where the approximation property of the interpolation operator Π_h is stated for the space $H^1(\nabla \cdot; \omega)$. The constant C_1 stems from an application of the Bramble-Hilbert lemma. For the second estimate (3.4) we recall the approximation property

$$\left\| u - \Pi_h^{L^2} u \right\|_{\tilde{H}^{-1/2}} \leq \tilde{C}_2 h^{1/2} \|u\|_{L^2}$$

proven e.g. in [CP06, Theorem 4.1]. An Application with $v := u - \Pi_h^{L^2} u$ yields

$$\left\| u - \Pi_h^{L^2} u \right\|_{\tilde{H}^{-1/2}} = \left\| v - \Pi_h^{L^2} v \right\|_{\tilde{H}^{-1/2}} \leq \tilde{C}_2 h^{1/2} \|v\|_{L^2} = \tilde{C}_2 h^{1/2} \left\| u - \Pi_h^{L^2} u \right\|_{L^2}.$$

Note that the L^2 -orthogonal projection onto piecewise constant function is the elementwise integral mean $\Pi_h^{L^2} u|_T = \frac{1}{|T|} \int_T u \, dx$. Therefore a scaling argument and an elementwise application of the Poincaré inequality yield

$$\left\| u - \Pi_h^{L^2} u \right\|_{L^2}^2 \leq C_2^2 h^2 |u|_{H^1}^2$$

This estimate combined with the identity (3.2) completes the proof. ■

3.1.2 The Discrete Minimization Problem (M_h)

Definition. Given a regular triangulation \mathcal{T}_h of ω , we define the set of discrete admissible magnetizations

$$\mathcal{A}_h := \{\mathbf{m}_h \in RT^0(\mathcal{T}_h) \mid |\mathbf{m}_h| \leq 1\} = \mathcal{A} \cap RT^0(\mathcal{T}_h). \quad (3.5)$$

□

Discrete minimization problem (M_h). Find a minimizer $\mathbf{m}_h^* \in \mathcal{A}_h$ of the discrete energy

$$e_h(\mathbf{m}_h) = e(\mathbf{m}_h) = \frac{1}{2} \|\nabla \cdot \mathbf{m}_h\|_V^2 + \frac{q}{2} \|\mathbf{m}_{h,2}\|_{L^2(\omega)}^2 - (\mathbf{f}, \mathbf{m}_h)_{L^2(\omega)}. \quad (3.6)$$

First, we stress that the conformity of the chosen discretization ensures that our discrete problem inherits the well-posedness of the continuous problem.

Corollary 3.2. *The discrete minimization problem (M_h) has a minimizer \mathbf{m}_h^* with uniquely determined divergence $\nabla \cdot \mathbf{m}_h^*$. If $q > 0$, the minimizer $\mathbf{m}_h^* \in \mathcal{A}_h$ is uniquely determined.*

Proof. The space $RT^0(\mathcal{T})$ is a finite dimensional and therefore closed subspace of the energy space \mathcal{H} . Since the intersection of two closed and convex sets is itself a closed and convex set, we may apply the arguments of the proof of Theorem 2.21 and use Theorem 2.22 to conclude the existence and uniqueness of a minimizer \mathbf{m}_h^* as in Corollary 2.23. ■

In order to establish an a priori analysis for the discrete minimization problem (M_h), we first state the well known equivalence of certain minimization problems and variational inequalities. This will provide some sort of generalization of C ea's Lemma which enables us to derive a priori convergence results by straight forward analysis.

Lemma 3.3. *Let H be a Hilbert space with (semi) scalar product $\langle \cdot, \cdot \rangle_H$. Furthermore, let $\Phi \in L(H; \mathbb{R})$ and let $K \subseteq H$ denote a closed and convex subset. Given the energy functional*

$$E(x) = \frac{1}{2} \langle x, x \rangle_H - \Phi(x),$$

an element $x^ \in K$ is a minimizer, i.e.*

$$E(x^*) \leq E(y) \quad \text{for all } y \in K, \quad (3.7)$$

if and only if x^ satisfies the variational inequality*

$$\langle x^*, x^* - y \rangle_H \leq \Phi(x^* - y) \quad \text{for all } y \in K. \quad (3.8)$$

Proof. We first show that (3.8) implies (3.7). Let therefore x^* be a solution of the variational inequality (3.8). Then, it holds that

$$\begin{aligned} E(y) - E(x^*) &= \frac{1}{2} \langle y, y \rangle_H - \frac{1}{2} \langle x^*, x^* \rangle_H - \Phi(y) + \Phi(x^*), \\ &= -\frac{1}{2} \langle x^* + y, x^* - y \rangle_H + \Phi(x^* - y), \\ &= \frac{1}{2} \langle x^* - y, x^* - y \rangle_H - \langle x^*, x^* - y \rangle_H + \Phi(x^* - y). \end{aligned}$$

Since $\langle x^* - y, x^* - y \rangle_H = \|x^* - y\|_H^2 \geq 0$ and x^* is a solution of (3.8), the last term in the computation has to be non-negative. Since $y \in K$ is arbitrary, this shows that x^* also solves the minimization problem (3.7).

Next, we show that each solution of the minimization problem (3.7) solves the variational inequality (3.8). To that end, assume $x^* \in K$ to be a solution of (3.7) and let $y \in K$. Due to the convexity of K , there holds $\tilde{x} := tx^* + (1-t)y \in K$ for all $0 < t < 1$. We compute

$$\begin{aligned} 0 &\leq E(\tilde{x}) - E(x^*) = E((1-t)y + tx^*) - E(y), \\ &= \frac{1}{2} \|-ty + tx^*\|_H^2 - \langle x^*, tx^* - ty \rangle_H + \Phi(tx^* - ty), \\ &= \frac{1}{2} t^2 \|-x^* + y\|_H^2 - t \langle x^*, x^* - y \rangle_H + t \Phi(x^* - y). \end{aligned}$$

In particular, this means

$$0 \leq \frac{1}{2} t \|-x^* + y\|_H^2 - \langle x^*, x^* - y \rangle_H + \Phi(x^* - y)$$

for all $t \in (0, 1)$. Considering the limit $t \rightarrow 0$ and recalling that $y \in K$ was arbitrary, we have therefore proven x^* to solve (3.8). \blacksquare

Remark. In the case of $K = H$ in Lemma 3.3, the corresponding variational inequality would in fact be an equality

$$\langle x, y \rangle_H = \Phi(x)$$

which is the Euler-Lagrange equation of the unconstrained minimization problem. From that point of view, Equation (3.8) may be seen as the Euler-Lagrange inequality of (M) . \square

We recall that our model problem (M) induces a natural energy (semi-)norm $\|\mathbf{m}\|^2 = q \|\mathbf{m}_2\|_{L^2}^2 + \|\nabla \cdot \mathbf{m}\|_V^2$ which obviously stems from the (semi) scalar product

$$\langle \mathbf{m}, \mathbf{w} \rangle = q(\mathbf{m}_2, \mathbf{w}_2)_{L^2} + \langle \nabla \cdot \mathbf{m}, V(\nabla \cdot \mathbf{w}) \rangle_{H^{1/2} \times \tilde{H}^{-1/2}}.$$

Since the mapping $\mathbf{m} \mapsto (\mathbf{f}, \mathbf{m})_{L^2}$ is a continuous and linear functional on \mathcal{H} , our model problem perfectly fits into the general setting of Lemma 3.3.

In Chapter 2, we applied the direct method of calculus of variations to establish existence and uniqueness of solutions of (M) . After reformulating (M) in terms of its variational inequality, by standard arguments [JLJ98, KS80], we finally obtain also continuous dependence of \mathbf{m}^* of the applied field \mathbf{f} .

Corollary 3.4. *Let \mathbf{m}_1 and \mathbf{m}_2 be solutions of (M) for applied fields \mathbf{f}_1 and \mathbf{f}_2 , then it holds that*

$$\|\mathbf{m}_1 - \mathbf{m}_2\| \leq \sqrt{2} |\omega|^{1/2} \|\mathbf{f}_1 - \mathbf{f}_2\|_{L^2(\omega)}^{1/2}. \quad (3.9)$$

Proof. The variational inequality

$$\langle \mathbf{m}_k, \mathbf{m}_k - \mathbf{m} \rangle \leq (\mathbf{f}_k, \mathbf{m})_{L^2}$$

with $k \in \{1, 2\}$ holds true for all $\mathbf{m} \in \mathcal{A}$. From that we conclude

$$\begin{aligned} \|\mathbf{m}_1 - \mathbf{m}_2\|^2 &= \langle \mathbf{m}_1, \mathbf{m}_1 - \mathbf{m}_2 \rangle + \langle \mathbf{m}_2, \mathbf{m}_2 - \mathbf{m}_1 \rangle \leq (\mathbf{f}_1, \mathbf{m}_1 - \mathbf{m}_2)_{L^2} + (\mathbf{f}_2, \mathbf{m}_2 - \mathbf{m}_1)_{L^2} \\ &= (\mathbf{f}_1 - \mathbf{f}_2, \mathbf{m}_1 - \mathbf{m}_2)_{L^2} \\ &\leq \|\mathbf{f}_1 - \mathbf{f}_2\|_{L^2(\omega)} \|\mathbf{m}_1 - \mathbf{m}_2\|_{L^2(\omega)}. \end{aligned}$$

Since $|\mathbf{m}_1(x) - \mathbf{m}_2(x)| \leq 2$, we have that $\|\mathbf{m}_1 - \mathbf{m}_2\|_{L^2(\omega)}^2 \leq 2|\omega|$, which concludes the proof. \blacksquare

The variational inequality, furthermore, allows us to establish an a priori error estimate.

Theorem 3.5. *Let $\mathbf{m}^* \in \mathcal{A}$ and $\mathbf{m}_h^* \in \mathcal{A}_h$ be solutions of (M) and (M_h) , respectively. Then it holds that*

$$\|\mathbf{m}^* - \mathbf{m}_h^*\|^2 \leq \inf_{\mathbf{w}_h \in \mathcal{A}_h} (\|\mathbf{m}^* - \mathbf{w}_h\|^2 - 2\langle \mathbf{m}^*, \mathbf{m}^* - \mathbf{w}_h \rangle + 2(f, \mathbf{m}^* - \mathbf{w}_h)_{L^2}). \quad (3.10)$$

Proof. First, we observe the inequality

$$\langle \mathbf{m}^*, \mathbf{m}^* \rangle = \langle \mathbf{m}^*, \mathbf{m}^* - \mathbf{m}_h^* \rangle + \langle \mathbf{m}^*, \mathbf{m}_h^* \rangle \leq \langle \mathbf{m}^*, \mathbf{m}_h^* \rangle + (f, \mathbf{m}^* - \mathbf{m}_h^*)_{L^2}$$

that follows from Lemma 3.3 applied to the minimization problem (M) . Using this simple observation and the variational inequality for the discrete problem (M_h) , we compute for arbitrary $\mathbf{w}_h \in \mathcal{A}_h$

$$\begin{aligned} \|\mathbf{m}^* - \mathbf{m}_h^*\|^2 &= \langle \mathbf{m}^* - \mathbf{m}_h^*, \mathbf{m}^* - \mathbf{m}_h^* \rangle \\ &= \langle \mathbf{m}^*, \mathbf{m}^* \rangle - 2\langle \mathbf{m}^*, \mathbf{m}_h^* \rangle + \langle \mathbf{m}_h^*, \mathbf{m}_h^* \rangle \\ &\leq \langle \mathbf{m}^*, \mathbf{m}_h^* \rangle + (f, \mathbf{m}^* - \mathbf{m}_h^*)_{L^2} - 2\langle \mathbf{m}^*, \mathbf{m}_h^* \rangle + \langle \mathbf{m}_h^*, \mathbf{w}_h^* \rangle + (f, \mathbf{m}_h^* - \mathbf{w}_h^*)_{L^2} \\ &= -\langle \mathbf{m}^*, \mathbf{m}_h^* \rangle + (f, \mathbf{m}^* - \mathbf{w}_h)_{L^2} + \langle \mathbf{m}_h^*, \mathbf{w}_h^* \rangle \\ &= \langle \mathbf{m}_h^*, \mathbf{w}_h - \mathbf{m}^* \rangle + (f, \mathbf{m}^* - \mathbf{w}_h)_{L^2} \\ &= \langle \mathbf{m}^* - \mathbf{m}_h^*, \mathbf{m}^* - \mathbf{w}_h \rangle - \langle \mathbf{m}^*, \mathbf{m}^* - \mathbf{w}_h \rangle + (f, \mathbf{m}^* - \mathbf{w}_h)_{L^2}. \end{aligned}$$

Finally, applying the trivial inequality $2ab \leq a^2 + b^2$ to the term $\langle \mathbf{m}^* - \mathbf{m}_h^*, \mathbf{m}^* - \mathbf{w}_h \rangle$, we see

$$\begin{aligned} \|\mathbf{m}^* - \mathbf{m}_h^*\|^2 &\leq \frac{1}{2}\|\mathbf{m}^* - \mathbf{m}_h^*\|^2 + \frac{1}{2}\|\mathbf{m}^* - \mathbf{w}_h\|^2 - \langle \mathbf{m}^*, \mathbf{m}_h^* - \mathbf{w}_h \rangle + (f, \mathbf{m}^* - \mathbf{w}_h)_{L^2} \\ \|\mathbf{m}^* - \mathbf{m}_h^*\|^2 &\leq \|\mathbf{m}^* - \mathbf{w}_h\|^2 - 2\langle \mathbf{m}^*, \mathbf{m}^* - \mathbf{w}_h \rangle + 2(f, \mathbf{m}^* - \mathbf{w}_h)_{L^2}. \end{aligned}$$

Since $\mathbf{w}_h \in \mathcal{A}_h$ was arbitrary the last inequality still holds when taking the infimum over all $\mathbf{w}_h \in \mathcal{A}_h$ on the right-hand side. This concludes the proof. \blacksquare

Corollary 3.6. *Let $\mathbf{m}^* \in \mathcal{A}$ and $\mathbf{m}_h^* \in \mathcal{A}_h$ be the solutions of the continuous model problem (M) and the discrete minimization problem (M_h) , respectively. Under the regularity assumption $\mathbf{m} \in H^1(\omega)^2$ and $\nabla \cdot \mathbf{m} \in H^{1/2}(\omega)$, they satisfy the a priori convergence result*

$$\|\mathbf{m}^* - \mathbf{m}_h^*\| = \mathcal{O}(\sqrt{h}). \quad (3.11)$$

Let $q = 0$. Then under the assumption $\nabla \cdot \mathbf{m} \in H^1(\omega)$, we obtain

$$\|\mathbf{m}^* - \mathbf{m}_h^*\| = \|\nabla \cdot (\mathbf{m}^* - \mathbf{m}_h^*)\|_V = \mathcal{O}(h^{3/4}).$$

Proof. We recall the interpolation operator $\Pi_h : H^1(\omega)^2 \rightarrow RT^0(\mathcal{T}_h)$ with its approximation properties from Theorem 3.1. Choosing $\mathbf{w}_h = \Pi_h \mathbf{m}^*$ in Theorem 3.5, the inequality (3.10) implies

$$\|\mathbf{m}^* - \mathbf{m}_h^*\|^2 \leq \|\mathbf{m}^* - \Pi_h \mathbf{m}^*\|^2 + 2\|\mathbf{m}^*\| \|\mathbf{m}^* - \Pi_h \mathbf{m}^*\| + 2\|f\|_{L^2} \|\mathbf{m}\|_{L^2}$$

which immediately yields the statement according to Theorem 3.1. \blacksquare

3.1.3 The (discrete) penalized minimization problem (M_h^ε)

We still need to provide some strategy to deal with the non-linear side constraint. For any $\varepsilon > 0$, we introduce an additional energy term

$$\frac{1}{2\varepsilon} \|(|\mathbf{m}| - 1)_+\|_{L^2(\omega)}^2. \quad (3.12)$$

The function $(\cdot)_+$ is given by

$$(u(x))_+ = \begin{cases} u(x) & \text{for } u(x) > 0 \\ 0 & \text{else.} \end{cases}$$

Discrete penalized minimization problem (M_h^ε). We seek to find a minimizer $\mathbf{m}_h^\varepsilon \in X_h$ of the penalized energy

$$e_\varepsilon(\mathbf{m}_h) = \frac{1}{2} \|\nabla \cdot \mathbf{m}_h\|_V^2 + \frac{q}{2} \|\mathbf{m}_{h,2}\|_{L^2(\omega)}^2 + \frac{1}{2\varepsilon} \|(|\mathbf{m}_h| - 1)_+\|_{L^2(\omega)}^2 - (\mathbf{f}, \mathbf{m}_h)_{L^2(\omega)}. \quad (3.13)$$

For $h > 0$, we set $X_h = RT^0(\mathcal{T}_h)$ with some mesh \mathcal{T}_h with mesh-size h . Formally we also allow $h = 0$, where we seek a continuous minimizer $\mathbf{m}_0^\varepsilon \in X_0 = \mathcal{H}$.

In a first step on our way to prove convergence $\mathbf{m}_h^\varepsilon \rightarrow \mathbf{m}_0^0$ as $(h, \varepsilon) \rightarrow (0, 0)$ and $\mathbf{m}_0^0 = \mathbf{m}^*$, we state and prove coercivity of the energy functional $e_\varepsilon(\cdot)$. To that end, we modify the proof of [Pra03, Satz 2.18] to fit our setting.

Lemma 3.7 (Coercivity of $e_\varepsilon(\cdot)$). *Let $\varepsilon > 0$ be given. Then, for any sequence of magnetizations $(\mathbf{m}_n)_{n \in \mathbb{N}} \subseteq \mathcal{H}$ with $\|\mathbf{m}_n\| \rightarrow \infty$, it holds that $e(\mathbf{m}_n) \rightarrow \infty$. In particular, any sequence $(\mathbf{m}_k)_{k \in \mathbb{N}}$ with bounded energy $\sup_{k \in \mathbb{N}} e(\mathbf{m}_k) < \infty$ has bounded norm, i.e. $\sup_{k \in \mathbb{N}} \|\mathbf{m}_k\| < \infty$.*

Proof. Recall the definition of the energy

$$e_\varepsilon(\mathbf{m}) = \frac{1}{2} \|\nabla \cdot \mathbf{m}\|_V^2 + \frac{q}{2} \|\mathbf{m}_2\|_{L^2(\omega)}^2 + \frac{1}{2\varepsilon} \|(|\mathbf{m}| - 1)_+\|_{L^2(\omega)}^2 - (\mathbf{f}, \mathbf{m})_{L^2(\omega)}.$$

From equivalence of norms $\|\nabla \cdot \mathbf{m}\|_V \sim \|\nabla \cdot \mathbf{m}\|_{\tilde{H}^{-1/2}(\omega)}$, more precisely the upper bound (2.24), we conclude the existence of a constant $C_1 > 0$ such that

$$e_\varepsilon(\mathbf{m}) \geq C_1 \|\nabla \cdot \mathbf{m}\|_{\tilde{H}^{-1/2}(\omega)}^2 + \frac{q}{2} \|\mathbf{m}_2\|_{L^2(\omega)}^2 + \frac{1}{2\varepsilon} \|(|\mathbf{m}| - 1)_+\|_{L^2(\omega)}^2 - (\mathbf{f}, \mathbf{m})_{L^2(\omega)}.$$

Let ω_\geq denote the set, where $|\mathbf{m}| \geq 1$, and $\omega_<$ the complement, i.e. $|\mathbf{m}(x)| < 1$ for $x \in \omega_<$. To estimate the Zeeman energy contribution, we use Hölder's inequality to see

$$\begin{aligned} \int_\omega \mathbf{f} \cdot \mathbf{m} \, dx &= \int_{\omega_\geq} \mathbf{f} \cdot \mathbf{m} \, dx + \int_{\omega_<} \mathbf{f} \cdot \mathbf{m} \, dx \\ &\leq \|\mathbf{f}\|_{L^2(\omega)} \|\mathbf{m}\|_{L^2(\omega_\geq)} + \int_{\omega_<} |\mathbf{f}| \, dx \\ &\leq \|\mathbf{f}\|_{L^2(\omega)} \|\mathbf{m}\|_{L^2(\omega_\geq)} + \|\mathbf{f}\|_{L^1(\omega)}. \end{aligned} \quad (3.14)$$

Next, for the penalty energy contribution it holds that

$$\begin{aligned} \int_\omega (|\mathbf{m}| - 1)_+^2 \, dx &= \int_{\omega_\geq} |\mathbf{m}|^2 - 2|\mathbf{m}| + 1 \, dx \\ &\geq \|\mathbf{m}\|_{L^2(\omega_\geq)}^2 - 2\|\mathbf{m}\|_{L^2(\omega_\geq)} |\omega_\geq|^{1/2} + |\omega_\geq| \\ &\geq \|\mathbf{m}\|_{L^2(\omega_\geq)}^2 - 2\|\mathbf{m}\|_{L^2(\omega_\geq)} |\omega|^{1/2}, \end{aligned} \quad (3.15)$$

where we again used the Hölder inequality to obtain the upper bound. Dropping the anisotropy energy and applying the inequalities (3.14)–(3.15), we obtain

$$\begin{aligned} e_\varepsilon(\mathbf{m}) &\geq \frac{1}{2\varepsilon} \left(\|\mathbf{m}\|_{L^2(\omega)}^2 - (\mathbf{f}, \mathbf{m})_{L^2(\omega)} + C_1 \|\nabla \cdot \mathbf{m}\|_{\tilde{H}^{-1/2}(\omega)}^2 \right) \\ &\geq \frac{1}{2\varepsilon} \left(\|\mathbf{m}\|_{L^2(\omega_\geq)}^2 - 2|\omega|^{1/2} \|\mathbf{m}\|_{L^2(\omega_\geq)} \right) - \|\mathbf{f}\|_{L^2(\omega)} \|\mathbf{m}\|_{L^2(\omega_\geq)} - \|\mathbf{f}\|_{L^1(\omega)} + C_1 \|\nabla \cdot \mathbf{m}\|_{\tilde{H}^{-1/2}(\omega)}^2 \\ &= \frac{1}{2\varepsilon} \left(\|\mathbf{m}\|_{L^2(\omega_\geq)}^2 - \left(2|\omega|^{1/2} + 2\varepsilon \|\mathbf{f}\|_{L^2(\omega)} \right) \|\mathbf{m}\|_{L^2(\omega_\geq)} - 2\varepsilon \|\mathbf{f}\|_{L^1(\omega)} \right) + C_1 \|\nabla \cdot \mathbf{m}\|_{\tilde{H}^{-1/2}(\omega)}^2. \end{aligned}$$

Defining the constants $C_2 = 2|\omega|^{1/2} + 2\varepsilon \|\mathbf{f}\|_{L^2(\omega)}$ and $C_3 = 2\varepsilon \|\mathbf{f}\|_{L^1(\omega)}$, we conclude

$$e_\varepsilon(\mathbf{m}) \geq \frac{1}{2\varepsilon} \left(\|\mathbf{m}\|_{L^2(\omega_\geq)}^2 - C_2 \|\mathbf{m}\|_{L^2(\omega_\geq)} - C_3 \right) + C_1 \|\nabla \cdot \mathbf{m}\|_{\tilde{H}^{-1/2}(\omega)}^2. \quad (3.16)$$

From

$$\|\mathbf{m}\|_{L^2(\omega)} \leq \|\mathbf{m}\|_{L^2(\omega_\geq)} + \|\mathbf{m}\|_{L^2(\omega_<)} \leq \|\mathbf{m}\|_{L^2(\omega_\geq)} + |\omega|^{1/2}, \quad (3.17)$$

we conclude the proof with the following observations: Let $(\mathbf{m}_n) \subseteq \mathcal{H}$ be a sequence of magnetizations with $\limsup_n \|\mathbf{m}_n\| = \infty$. Then either $\limsup_n \|\nabla \cdot \mathbf{m}_n\|_{\tilde{H}^{-1/2}(\omega)} = \infty$ or $\limsup_n \|\mathbf{m}_n\|_{L^2(\omega)} = \infty$, or both. Suppose $\limsup_n \|\nabla \cdot \mathbf{m}_n\|_{\tilde{H}^{-1/2}} = \infty$, then we immediately conclude $\limsup_n e_\varepsilon(\mathbf{m}_n) = \infty$ from (3.16). On the other hand if $\limsup_n \|\mathbf{m}_n\|_{L^2(\omega)} = \infty$, the combination of (3.17) and (3.16) again yields $\limsup_n e_\varepsilon(\mathbf{m}_n) = \infty$. ■

Well-posedness of the model problem (M_h^ε) now follows by the direct method of calculus of variations, cf. [Dac89].

Corollary 3.8. *For $h > 0$, let X_h denote some closed subspace of \mathcal{H} . For $h = 0$, we identify $X_0 = \mathcal{H}$. Then, there is a minimizer $\mathbf{m}_h^\varepsilon \in X_h$ of the penalized energy $e_\varepsilon(\cdot)$. Moreover, for $q > 0$ the minimizer \mathbf{m}_h^ε is uniquely determined.*

Proof. Let $(\mathbf{m}_n)_{n \in \mathbb{N}}$ be a sequence with $\inf_{\mathbf{m} \in X_h} e_\varepsilon(\mathbf{m}) = \lim_{n \rightarrow \infty} e_\varepsilon(\mathbf{m}_n)$. From $e_\varepsilon(0) = 0$, we conclude that the sequence has bounded energy. The coercivity of $e_\varepsilon(\cdot)$ implies that \mathbf{m}_n is bounded in norm, and therefore there exists (w.l.o.g.) some weak limit \mathbf{m}_h^ε . Since X_h is closed and convex, it is also closed with respect to the weak topology, and thus $\mathbf{m}_h^\varepsilon \in X_h$. With the weak lower semi-continuity of $e_\varepsilon(\cdot)$, which follows from continuity and convexity, we finally conclude that \mathbf{m}_h^ε is a minimizer. If $q > 0$, then by Theorem 2.22, we obtain uniqueness. ■

3.2 Convergence

We establish weak convergence of minimizers as $\varepsilon \rightarrow 0$. We stress that convergence for $\varepsilon \rightarrow 0$ with fixed h is well-known in the literature, see e.g. [NW99]. However our proof differs from the one included in the cited monograph. Since we use similar arguments in the proof of convergence with fixed ε for $h \rightarrow 0$, we state the result as a theorem and prove it.

Lemma 3.9. *Let $\varepsilon_n \rightarrow 0$ be a non-negative zero sequence and let $(\mathbf{m}_n)_{n \in \mathbb{N}} \subseteq \mathcal{H}$ be a sequence of magnetizations such that*

$$\sup_{n \in \mathbb{N}} e_{\varepsilon_n}(\mathbf{m}_n) < \infty \quad (3.18)$$

and

$$\mathbf{m}_n \rightharpoonup \mathbf{m}_\infty$$

for some $\mathbf{m}_\infty \in \mathcal{H}$. Then, it holds that $\mathbf{m}_\infty \in \mathcal{A}$, i.e. the weak limit \mathbf{m} satisfies the constraint

$$|\mathbf{m}_\infty| \leq 1.$$

Proof. Note that the term

$$\|(|\mathbf{m}| - 1)_+\|_{L^2(\omega)}^2$$

is convex and continuous. Hence, it is also weakly lower semi-continuous. From the weak convergence $\mathbf{m}_n \rightharpoonup \mathbf{m}_\infty$, we therefore obtain

$$\|(|\mathbf{m}_\infty| - 1)_+\|_{L^2(\omega)}^2 \leq \liminf_{n \in \mathbb{N}} \|(|\mathbf{m}_n| - 1)_+\|_{L^2(\omega)}^2. \quad (3.19)$$

By definition it holds that $e_{\varepsilon_n}(\mathbf{m}_n) = e(\mathbf{m}_n) + \frac{1}{2\varepsilon} \|(|\mathbf{m}_n| - 1)_+\|_{L^2}^2$, which means

$$2\varepsilon_n(e_{\varepsilon_n}(\mathbf{m}_n) - e(\mathbf{m}_n)) = \|(|\mathbf{m}_n| - 1)_+\|_{L^2}^2.$$

From (3.18), we obtain an upper bound

$$\limsup_{n \in \mathbb{N}} e(\mathbf{m}_n) \leq \limsup_{n \in \mathbb{N}} e_{\varepsilon_n}(\mathbf{m}_n) < \infty.$$

The weak lower semicontinuity of $e(\cdot)$ together with $\mathbf{m}_n \rightharpoonup \mathbf{m}_\infty$ yields the lower bound

$$e(\mathbf{m}_\infty) \leq \liminf_{n \in \mathbb{N}} e(\mathbf{m}_n) \leq \liminf_{n \in \mathbb{N}} e_{\varepsilon_n}(\mathbf{m}_n).$$

In particular, both sequences $e(\mathbf{m}_n)$ and $e_{\varepsilon_n}(\mathbf{m}_n)$ are bounded. This, together with (3.19) and $\varepsilon_n \rightarrow 0$, proves

$$\|(|\mathbf{m}_\infty| - 1)_+\|_{L^2}^2 \leq \liminf_{n \in \mathbb{N}} \|(|\mathbf{m}_n| - 1)_+\|_{L^2}^2 = \liminf_{n \in \mathbb{N}} 2\varepsilon_n(e_{\varepsilon_n}(\mathbf{m}_n) - e(\mathbf{m}_n)) = 0,$$

i.e. $\mathbf{m}_\infty \in \mathcal{A}$. ■

Theorem 3.10 (Convergence as $\varepsilon \rightarrow 0$). *For any sequence $(\varepsilon_n)_{n \in \mathbb{N}} \subseteq \mathbb{R}_{>0}$ with $\varepsilon_n \rightarrow 0$, there holds weak convergence with respect to the norm topology of \mathcal{H} of the minimizers $\mathbf{m}_h^{\varepsilon_n}$ of $(M_h^{\varepsilon_n})$ in the following sense: Any subsequence $\mathbf{m}_h^{\varepsilon_{n_k}}$ contains a weakly convergent subsequence $\mathbf{m}_h^{\varepsilon_{n_{k_\ell}}}$ whose limit is a minimizer \mathbf{m}_h^0 of the constrained problem (M_h) . For $q > 0$, there holds weak convergence of the full sequence*

$$\mathbf{m}_h^{\varepsilon_n} \rightharpoonup \mathbf{m}_h^0.$$

Proof. Since \mathbf{m}_h^ε is the minimizer of the unconstrained problem (M_h^ε) , it holds that

$$e_\varepsilon(\mathbf{m}_h^\varepsilon) \leq e_\varepsilon(\mathbf{m}_h^0) = e(\mathbf{m}_h^0). \quad (3.20)$$

Therefore, the sequence of minimizers $\mathbf{m}_h^{\varepsilon_n}$ has bounded energy. From Lemma 3.7, we obtain boundedness of $\mathbf{m}_h^{\varepsilon_n}$ in the norm. This yields that any subsequence $\mathbf{m}_h^{\varepsilon_{n_k}}$ is bounded and must have a weakly convergent subsequence $\mathbf{m}_h^{\varepsilon_{n_{k_\ell}}} \rightharpoonup \mathbf{m}_h^*$. It remains to prove that first $\mathbf{m}_h^* \in \mathcal{A}_h$ and second it is indeed a minimizer of $e(\cdot)$.

The first statement, i.e. $\mathbf{m}_h^* \in \mathcal{A}_h$ follows immediately from Lemma 3.9. From

$$e(\mathbf{m}_h^{\varepsilon_{n_{k_\ell}}}) \leq e_{\varepsilon_{n_{k_\ell}}}(\mathbf{m}_h^{\varepsilon_{n_{k_\ell}}}) \leq e(\mathbf{m}_h^0)$$

and weakly lower semicontinuity of $e(\cdot)$ (2.27), we conclude

$$e(\mathbf{m}_h^*) \leq \liminf_{\ell \in \mathbb{N}} e(\mathbf{m}_h^{\varepsilon_{n_{k_\ell}}}) \leq e(\mathbf{m}_h^0),$$

and since $\mathbf{m}_h^0 \in \mathcal{A}_h$ is a minimizer of $e(\cdot)$ there must hold $e(\mathbf{m}_h^*) = e(\mathbf{m}_h^0)$, i.e. $\mathbf{m}_h^* \in \mathcal{A}_h$ is a minimizer of (M_h) as well.

In a final step, we assume $q > 0$. By Corollary 2.23 the minimizer $\mathbf{m}_h^0 \in \mathcal{A}_h$ of (M_h) is uniquely determined and there must thus hold $\mathbf{m}_h^* = \mathbf{m}_h^0$. This means that every subsequence of $\mathbf{m}_h^{\varepsilon_n}$ has a subsequence with the same weak limit \mathbf{m}_h^0 . Therefore, there already holds convergence

$$\mathbf{m}_h^{\varepsilon_n} \rightharpoonup \mathbf{m}_h^0$$

of the sequence itself. ■

Next, we prove convergence of solutions with respect to the discretization parameter h without any regularity assumptions. To that end, in the case of $\varepsilon > 0$ we refer to the penalized problem (M_h^ε) , whereas in the case $\varepsilon = 0$ we refer to the constrained problem $(M_h) = (M_h^0)$.

Theorem 3.11 (Convergence as $h \rightarrow 0$). *Let $X_h \subseteq \mathcal{H}$ be a monotone family of finite dimensional subspaces of \mathcal{H} with $\lim_{h \rightarrow 0} X_h = \mathcal{H}$. By this we mean that $h_1 \leq h_2$ implies $X_{h_2} \subseteq X_{h_1}$ and $\overline{\bigcup_{h>0} X_h} = \mathcal{H}$. Then, for any sequence $(h_n)_{n \in \mathbb{N}} \subseteq \mathbb{R}_{>0}$ with $h_n \rightarrow 0$, there holds weak convergence of minimizers $\mathbf{m}_{h_n}^\varepsilon$ of $(M_{h_n}^\varepsilon)$ with respect to the norm topology of \mathcal{H} in the following sense: Any subsequence $\mathbf{m}_{h_{n_k}}^\varepsilon$ contains a weakly convergent subsequence $\mathbf{m}_{h_{n_{k_\ell}}}^\varepsilon$ whose limit is a minimizer \mathbf{m}_0^ε of the continuous penalized problem (M_0^ε) . For $q > 0$, there holds weak convergence of the full sequence*

$$\mathbf{m}_{h_n}^\varepsilon \rightharpoonup \mathbf{m}_0^\varepsilon.$$

Proof. Let h_n be a zero sequence and assume w.l.o.g. $h_n \leq 1$. Then, from monotonicity of spaces we conclude $X_1 \subseteq X_{h_n}$ and hence

$$e_\varepsilon(\mathbf{m}_{h_n}^\varepsilon) \leq e_\varepsilon(\mathbf{m}_1^\varepsilon). \quad (3.21)$$

Put differently, the sequence $(\mathbf{m}_{h_n}^\varepsilon)_{n \in \mathbb{N}}$ has bounded energy. From Lemma 3.7, we deduce boundedness of the sequence in \mathcal{H} . In particular, each subsequence $\mathbf{m}_{h_{n_k}}^\varepsilon$ is itself bounded and has therefore a weakly convergent subsequence

$$\mathbf{m}_{h_{n_{k_\ell}}}^\varepsilon \rightharpoonup \mathbf{m}_*^\varepsilon.$$

It remains to prove that \mathbf{m}_*^ε is a minimizer.

Let \mathbf{m}_0^ε denote a minimizer of the continuous problem (M_0^ε) . Since $e_\varepsilon(\cdot)$ is continuous, for any $\eta > 0$ there is a $\delta > 0$ such that

$$\|\mathbf{w} - \tilde{\mathbf{w}}\| \leq \delta \Rightarrow |e_\varepsilon(\mathbf{w}) - e_\varepsilon(\tilde{\mathbf{w}})| \leq \eta.$$

From $\lim_{h \rightarrow 0} X_h = \mathcal{H}$ we know that for each $\delta > 0$ there is an integer $L \in \mathbb{N}$ such that for all $\ell \geq L$ there exists some $\tilde{\mathbf{m}}_{h_{n_{k_\ell}}}^\varepsilon \in X_{h_{n_{k_\ell}}}$ such that

$$\left\| \mathbf{m}_0^\varepsilon - \tilde{\mathbf{m}}_{h_{n_{k_\ell}}}^\varepsilon \right\| \leq \delta.$$

Altogether, we know that for arbitrary $\eta > 0$ there exists some index $L \in \mathbb{N}$ such that for all $\ell \geq L$ we may choose $\tilde{\mathbf{m}}_{h_{n_{k_\ell}}}^\varepsilon \in X_{h_{n_{k_\ell}}}$ with

$$e_\varepsilon(\tilde{\mathbf{m}}_{h_{n_{k_\ell}}}^\varepsilon) \leq e_\varepsilon(\mathbf{m}_0^\varepsilon) + \eta.$$

Recall that $\mathbf{m}_{h_{n_{k_\ell}}}^\varepsilon$ is a minimizer and therefore $e_\varepsilon(\mathbf{m}_{h_{n_{k_\ell}}}^\varepsilon) \leq e_\varepsilon(\tilde{\mathbf{m}}_{h_{n_{k_\ell}}}^\varepsilon)$. Since this holds for any $\eta > 0$ and from (weakly lower semi-)continuity of $e_\varepsilon(\cdot)$, we conclude

$$e_\varepsilon(\mathbf{m}_*^\varepsilon) \leq \liminf_{\ell \rightarrow \infty} e_\varepsilon(\mathbf{m}_{h_{n_{k_\ell}}}^\varepsilon) \leq e_\varepsilon(\mathbf{m}_0^\varepsilon),$$

which means that \mathbf{m}_*^ε is in fact a minimizer.

For $q > 0$ the minimizer \mathbf{m}_0^ε is uniquely determined from which weak convergence

$$\mathbf{m}_{h_n}^\varepsilon \rightharpoonup \mathbf{m}_0^\varepsilon$$

follows. ■

Now that we have proven

$$\lim_{h \rightarrow 0} \lim_{\varepsilon \rightarrow 0} \mathbf{m}_h^\varepsilon = \mathbf{m}^* = \lim_{\varepsilon \rightarrow 0} \lim_{h \rightarrow 0} \mathbf{m}_h^\varepsilon$$

in a weak sense, we finally establish that any choice of discretization and penalization parameters yields convergence, as long as $(h, \varepsilon) \rightarrow (0, 0)$. This finally ensures unconditional convergence of our proposed scheme for the numerical solution of (M) . In particular no assumptions on the regularity of the analytical solution are necessary.

Corollary 3.12 (Convergence as $(h, \varepsilon) \rightarrow (0, 0)$). *With the assumptions of Theorem 3.10 and Theorem 3.11, there holds weak convergence with respect to the norm topology of \mathcal{H} of minimizers $\mathbf{m}_{h_n}^{\varepsilon_n}$ of $(M_{h_n}^{\varepsilon_n})$ for any zero sequence $(h_n, \varepsilon_n) \rightarrow (0, 0)$ with $h_n, \varepsilon_n > 0$ in the following sense: Any subsequence $\mathbf{m}_{h_{n_k}}^{\varepsilon_{n_k}}$ contains a weakly convergent subsequence $\mathbf{m}_{h_{n_{k_\ell}}}^{\varepsilon_{n_{k_\ell}}}$ whose limit is a minimizer \mathbf{m}^* of the constrained and continuous problem (M) . For $q > 0$, there holds weak convergence of the full sequence*

$$\mathbf{m}_{h_n}^{\varepsilon_n} \rightharpoonup \mathbf{m}^*.$$

Proof. Let $\mathbf{m}_0^{\varepsilon_n}$ denote a minimizer of the continuous and penalized problem $(M_0^{\varepsilon_n})$ and let $\mathbf{m}_{h_n}^0$ denote a minimizer of the discrete constrained problem (M_{h_n}) , respectively. We observe

$$e(\mathbf{m}_0^{\varepsilon_n}) \leq e(\mathbf{m}_{h_n}^{\varepsilon_n}) \leq e(\mathbf{m}_{h_n}^0) \leq e(0) = 0. \quad (3.22)$$

Therefore, the sequence $\mathbf{m}_{h_n}^{\varepsilon_n}$ is bounded and every subsequence $\mathbf{m}_{h_{n_k}}^{\varepsilon_{n_k}}$ has a weakly convergent subsequence $\mathbf{m}_{h_{n_{k_\ell}}}^{\varepsilon_{n_{k_\ell}}} \rightharpoonup \tilde{\mathbf{m}}^*$. From Theorem 3.11, we conclude existence of a subsequence which we write

for simplicity as $\mathbf{m}_{h_s}^{\varepsilon_s}$ such that the associated sequence $\mathbf{m}_{h_s}^0$ converges weakly to a minimizer \mathbf{m}^* and hence

$$e(\mathbf{m}_{h_s}^{\varepsilon_s}) \leq e(\mathbf{m}_{h_s}^0) \rightarrow e(\mathbf{m}^*).$$

From this it follows, again by weak lower semicontinuity of $e(\cdot)$, that $e(\tilde{\mathbf{m}}^*) \leq e(\mathbf{m}^*)$. Recall that $e_{\varepsilon_s}(\mathbf{m}_{h_s}^{\varepsilon_s}) \leq e(\mathbf{m}_{h_0}^0)$. From that we see that $\mathbf{m}_{h_s}^{\varepsilon_s}$ satisfies the assumption of Lemma 3.9, hence $\tilde{\mathbf{m}}^* \in \mathcal{A}$. Finally we obtain that $\tilde{\mathbf{m}}^*$ is a minimizer, i.e. the desired convergence result. ■

3.3 A priori error estimates

We have already established an a priori estimate for $\|\mathbf{m} - \mathbf{m}_h\|$. However, in practice, we compute in fact the solution \mathbf{m}_h^ε of (M_h^ε) . We are therefore interested in some estimate for $\|\mathbf{m} - \mathbf{m}_h^\varepsilon\|$. We proceed in two steps. First, we estimate the error $\|\mathbf{m} - \mathbf{m}_0^\varepsilon\|$ introduced by the penalization. To that end we need to establish the Euler-Lagrange equations for the constrained problem (M) . Second, we prove that, under suitable regularity assumptions, the discretization of the penalized system leads to the same convergence behavior as the discretization of the constrained problem.

3.3.1 The Euler-Lagrange equations (KKT conditions)

In the first subsection, we aim at understanding the Euler-Lagrange equations of our minimization problem. Our further numerical analysis is built upon this result. Based on a regularity result from [DKMO02], we prove the existence of a Lagrange multiplier $\lambda \geq 0$ such that the minimization problem (M) may be stated *equivalently* in terms of its Euler-Lagrange equations

$$q \begin{pmatrix} 0 \\ \mathbf{m}_2 \end{pmatrix} + \nabla u - f + \lambda \mathbf{m} = 0, \quad (3.23)$$

$$\lambda(|\mathbf{m}| - 1) = 0, \quad (3.24)$$

$$\lambda \geq 0, \quad (3.25)$$

$$|\mathbf{m}| \leq 1, \quad (3.26)$$

often also referred to as KKT-conditions.

One may be tempted to use standard techniques from infinite or semi-infinite optimization. However, black-box techniques do not make use of the very concrete structure of our model problem and therefore may not provide a good framework for numerical analysis. Instead, we follow the lines of the proof of equivalence of the Euler-Lagrange equation and the model problem in case of the large body limit that was performed in [De 93]. The proof cannot be transferred easily to our thin-film problem. The underlying idea to understand the geometry of the first derivative of the energy, however, is preserved.

Before proving the equivalence of the Euler-Lagrange equation and the reduced thin-film model problem, we first need some definitions and statements, well known and established in the literature, to form the basis for our analysis.

Definition. Let X be a Banach space and $F : X \rightarrow \mathbb{R}$. If the limit

$$dF(x, y) := \lim_{\varepsilon \rightarrow 0^+} \frac{F(x + \varepsilon y) - F(x)}{\varepsilon} \quad (3.27)$$

exists we shall call it the derivative at the point $x \in X$ in the direction $y \in X$. Furthermore, the function F is said to be Gateaux differentiable at the point x with Gateaux derivative $DF(x) \in X^*$, if (3.27) exists for all $y \in X$ and

$$dF(x, y) = \langle y, DF(x) \rangle \quad (3.28)$$

is satisfied. In the following we also use the notation

$$(DF(x))(y) = \langle y, DF(x) \rangle.$$

□

We now compute the Gateaux derivative of our energy functional e as defined in (2.26). We first prove a lemma which in fact covers two of the three terms of which the energy is composed.

Lemma 3.13. *Let X, Y be Hilbert spaces and let $F : H \rightarrow \mathbb{R}$ be defined through $F(x) = (Tx, Tx)_Y$ with some linear and continuous mapping $T \in L(X; Y)$. Then the Gateaux derivative of F at the point $x \in X$ in the direction $\tilde{x} \in X$ reads*

$$(DF(x))(\tilde{x}) = 2(Tx, T\tilde{x}).$$

Proof. We plug in the definition (3.27) to obtain

$$\begin{aligned} dF(x, y) &= \frac{F(x + \varepsilon y) - F(x)}{\varepsilon} = \frac{(T(x + \varepsilon y), T(x + \varepsilon y)) - (Tx, Tx)}{\varepsilon} \\ &= \frac{(Tx, Tx) + 2\varepsilon(Tx, Ty) + \varepsilon^2(Ty, Ty) - (Tx, Tx)}{\varepsilon} \\ &= 2(Tx, Ty) + \varepsilon(Ty, Ty). \end{aligned}$$

Hence the limit for $\varepsilon \rightarrow 0$ is precisely as stated. ■

Theorem 3.14. *The Gateaux derivative of the energy $e : \mathcal{H} \rightarrow \mathbb{R}$ reads*

$$(De(\mathbf{m}))(\tilde{\mathbf{m}}) = (\mathcal{P}\mathbf{m}, \mathcal{P}\tilde{\mathbf{m}})_{L^2(\mathbb{R}^3)^3} + q(\mathbf{m}_2, \tilde{\mathbf{m}}_2)_{L^2(\omega)} - (\mathbf{f}, \tilde{\mathbf{m}})_{L^2(\omega)^2}. \quad (3.29)$$

Proof. The energy e may be written in the form

$$e(\mathbf{m}) = \frac{1}{2}(\mathcal{P}\mathbf{m}, \mathcal{P}\mathbf{m})_{L^2(\mathbb{R}^3)^3} + \frac{q}{2}(\mathbf{m}_2, \mathbf{m}_2)_{L^2(\omega)} - (\mathbf{f}, \mathbf{m})_{L^2(\omega)^2}. \quad (3.30)$$

We simply apply Lemma 3.13 to the first two terms of the energy and compute the derivative of the last term trivially to prove the statement. ■

Our aim is to find an explicit L^2 -representation of the derivative $De(\mathbf{m})(\cdot)$. To that end we consider the algebraic-topological dual \mathcal{H}^* of \mathcal{H} with respect to the extended L^2 scalar product. The following Lemma from [Pra] provides the mathematical context necessary to define \mathcal{H}^* in the intended way properly.

Lemma 3.15. *Let $X \subseteq Y$ be continuously embedded Hilbert spaces. Then the Riesz mapping*

$$J_Y : Y \rightarrow Y^*, \quad J_Y y := (y; \cdot)_Y$$

is well-defined as linear operator $J_Y \in L(Y, X^)$ and the range $J_Y(Y)$ is a dense subspace of X^* .*

Proof. Due to the continuous embedding, there is a constant $C > 0$ such that $\|x\|_Y \leq C \|x\|_X$ for any $x \in X$. By Cauchy-Schwarz' inequality, we infer

$$|(y; x)_Y| \leq \|y\|_Y \|x\|_Y \leq C \|y\|_Y \|x\|_X.$$

Therefore, $J_Y \in L(Y, X^*)$ is well-defined. Let J_X denote the Riesz mapping with respect to X . Then $J_Y(Y)$ is dense in X^* if and only if $V := J_X^{-1}(J_Y(Y))$ is dense in $X = \bar{V} \oplus \bar{V}^\perp$. We conclude the proof by verifying $\bar{V}^\perp = \{0\}$. Let $x \in \bar{V}^\perp$ and $y \in Y$ then it holds at

$$0 = (x; J_X^{-1}(J_Y(y)))_X = (J_Y(y))(x) = (y; x)_Y.$$

The admissible choice of $y = x \in \bar{V}^\perp$ implies $x = 0 \in Y \supseteq X$. ■

We may apply Lemma 3.15 to the spaces $X = \mathcal{H}$ and $Y = L^2(\omega)$ which yields a representation of \mathcal{H}^* with respect to the extended L^2 scalar product.

We need to state some regularity result concerning the gradient of the magnetostatic potential $\nabla u = \mathcal{P}\mathbf{m}$. Finally, based on this regularity result, the explicit representation of $De(\mathbf{m})(\cdot)$ may be derived.

Proposition 3.16 ([DKMO02, Proposition 3.3]). *Assume that ω is simply connected and that \mathbf{f} has all derivatives up to second order in $L^1(\omega)$. Then the magnetostatic potential u^* corresponding to the minimizer $\mathbf{m}^* \in \mathcal{A}$ of (2.26) satisfies*

$$\nabla u \in L_{loc}^4(\omega) \subseteq L_{loc}^2(\omega).$$
■

Lemma 3.17. *Assume that ω is simply connected and that \mathbf{f} has all derivatives up to second order in $L^1(\omega)$. Let $\mathbf{m}^* \in \mathcal{A}$ denote a minimizer of (2.26) with associated magnetostatic potential u^* . Then it holds that*

$$De(\mathbf{m}^*) = q \begin{pmatrix} 0 \\ \mathbf{m}_2^* \end{pmatrix} + \nabla u^* - f \tag{3.31}$$

almost everywhere on ω .

Proof. Let r denote the right-hand side in (3.31). Since both, r and $De(\mathbf{m}^*)$ are distributions, i.e. $r, De(\mathbf{m}^*) \in \mathcal{D}^*(\omega)^2$, we prove

$$De(\mathbf{m}^*)(\varphi) = \int_{\omega} r \cdot \varphi \quad \text{for all } \varphi \in \mathcal{D}(\omega)^2.$$

This, together with Proposition 3.16 and the fundamental theorem of the calculus of variations, yields the representation stated in (3.31). Let $\varphi \in \mathcal{D}(\omega)^2$ be given and let μ denote the corresponding magnetostatic potential. According to the Maxwell equation (2.3), it holds that

$$\int_{\mathbb{R}^3} \nabla \mu \cdot \nabla \psi \, dx = \int_{\omega} \varphi \cdot \nabla \psi(x, 0) \, dx \quad \text{for all } \psi \in \mathcal{D}(\mathbb{R}^3). \tag{3.32}$$

As argued in the proof of [DKMO02, Corollary 3.4], the regularity of u allows to replace $\psi \in \mathcal{D}(\mathbb{R}^3)$ in (3.32) by the magnetostatic potential, i.e.

$$\int_{\mathbb{R}^3} \nabla \mu \cdot \nabla u \, dx = \int_{\omega} \varphi \cdot \nabla u(x, 0) \, dx.$$

Using (3.29), we obtain

$$\begin{aligned} De(\mathbf{m}^*)(\varphi) &= (q \begin{pmatrix} 0 \\ \mathbf{m}_2^* \end{pmatrix}, \varphi)_{L^2(\omega)} + (\nabla u, \nabla \mu)_{L^2(\mathbb{R}^3)^3} - (f, \varphi)_{L^2(\omega)} \\ &= (q \begin{pmatrix} 0 \\ \mathbf{m}_2^* \end{pmatrix}, \varphi)_{L^2(\omega)} + (\nabla u, \varphi)_{L^2(\omega)} - (f, \varphi)_{L^2(\omega)} = \int_{\omega} r \cdot \varphi \end{aligned}$$

which concludes the proof. \blacksquare

Now, we have collected all preliminary results necessary to state the Euler-Lagrange equations and prove the equivalence to our model problem (M) properly.

Theorem 3.18. *Assume that ω is simply connected and \mathbf{f} has all derivatives up to second order in $L^1(\omega)$. Then, any solution $(\mathbf{m}^*, \lambda^*) \in \mathcal{A} \times L^2_{loc}(\omega)$ of the Euler-Lagrange equation*

$$\begin{aligned} q \begin{pmatrix} 0 \\ \mathbf{m}_2 \end{pmatrix} + \nabla u - f + \lambda \mathbf{m} &= 0 \quad a. e., \\ \lambda(|\mathbf{m}| - 1) &= 0 \quad a. e., \\ \lambda &\geq 0 \quad a. e., \\ |\mathbf{m}| &\leq 1 \quad a. e., \end{aligned} \tag{3.33}$$

provides a minimizer $\mathbf{m}^* \in \mathcal{A}$ of the energy

$$e(\mathbf{m}) = \frac{q}{2}(\mathbf{m}_2, \mathbf{m}_2)_{L^2(\omega)} + \frac{1}{2}(\nabla \cdot \mathbf{m}, \nabla \cdot \mathbf{m})_V - (f, \mathbf{m})_{L^2(\omega)}. \tag{3.34}$$

Conversely, if $\mathbf{m}^* \in \mathcal{A}$ is a solution of (M), there exists $\lambda^* \in L^2_{loc}(\omega)$ such that $(\mathbf{m}^*, \lambda^*)$ solves (3.33).

Proof. We recall that solving our minimization problem is, according to Lemma 3.3, equivalent to solving the variational inequality

$$De(\mathbf{m})(\mathbf{m} - \mathbf{w}) \leq 0 \quad \text{for all } \mathbf{w} \in \mathcal{A}. \tag{3.35}$$

This fact will be used throughout the entire proof.

First we prove that any solution of (3.33) provides a minimizer of (2.26): From

$$\lambda(|\mathbf{m}| - 1) = 0$$

we infer $\lambda(x) = 0$ for all $x \in \omega$ with $|\mathbf{m}(x)| < 1$. Let $(\mathbf{m}^*, \lambda^*)$ be a solution of (3.33) and choose some arbitrary $\mathbf{m} \in \mathcal{A}$. Then with $\omega_1 := \{x \in \omega \mid |\mathbf{m}^*| = 1\}$ it holds that

$$\begin{aligned} De(\mathbf{m}^*)(\mathbf{m}^* - \mathbf{m}) &= \langle q \begin{pmatrix} 0 \\ \mathbf{m}_2^* \end{pmatrix} + \nabla u^* - f, \mathbf{m}^* - \mathbf{m} \rangle \\ &= -(\lambda^* \mathbf{m}^*, \mathbf{m}^*)_{L^2(\omega_1)} + (\lambda^* \mathbf{m}^*, \mathbf{m})_{L^2(\omega_1)}. \end{aligned}$$

Since $|\mathbf{m}| \leq |\mathbf{m}^*|$ almost everywhere on ω_1 , it holds that $\mathbf{m}^* \cdot \mathbf{m}^* = 1$ is a pointwise upper bound for $\mathbf{m}^* \cdot \mathbf{m}$ and we conclude from $\lambda^* \geq 0$ the variational inequality

$$De(\mathbf{m}^*)(\mathbf{m}^* - \mathbf{m}) \leq 0 \quad \text{for all } \mathbf{m} \in \mathcal{A},$$

which is equivalent to $\mathbf{m}^* \in \mathcal{A}$ being a minimizer of (2.26).

Second we prove that for a minimizer $\mathbf{m}^* \in \mathcal{A}$ of (2.26) there exists a Lagrange multiplier $\lambda^* \geq 0$ such that $(\mathbf{m}^*, \lambda^*)$ solves (3.33): Let r denote the L^2 representation of $De(\mathbf{m}^*) \in \mathcal{H}^*$, i.e.

$$r = q \begin{pmatrix} 0 \\ \mathbf{m}_2^* \end{pmatrix} + \nabla u^* - f.$$

We define $\omega_{<} := \{x \in \omega \mid |\mathbf{m}^*(x)| < 1\}$. We analyze r on the sub-domains $\omega_{<}$ and ω_1 separately and prove $r|_{\omega_{<}} = 0$ and $r|_{\omega_1} \parallel \mathbf{m}^*$ in two steps.

First we prove $r = 0$ on $\omega_{<}$: Let $x_0 \in \omega_{<}$ be arbitrary. Then there exist a neighborhood of $\omega_{x_0} \subseteq \omega$ and some $\varepsilon_{x_0} > 0$ such that $\overline{\omega_{x_0} + B_\varepsilon(0)} \subseteq \omega$ for all $\varepsilon \leq \varepsilon_{x_0}$. We stress that $\overline{\omega_{x_0} + B_\varepsilon(0)}$ is bounded and therefore compact. We further define the set $\omega_k := \{x \in \omega \mid |\mathbf{m}^*(x)| \leq 1 - \frac{1}{k}\}$. Let $v \in L^\infty(\omega)^2$, then we define $\delta_k := \frac{1}{k\|v\|_\infty}$. We observe that the variation

$$\mathbf{m}_\kappa := \mathbf{m}^* + \kappa \chi_{\omega_k} \chi_{\omega_{x_0}} v \in L^\infty(\omega)^2$$

satisfies the constraint $|\mathbf{m}_\kappa| \leq 1$ for all $\kappa \in [-\delta_k, \delta_k]$. We seek a sequence of admissible functions which converges to \mathbf{m}_κ in $L^2(\omega)^2$. To this end we choose a sequence $(\varphi_n)_{n \in \mathbb{N}} \subseteq \mathcal{D}(\omega_{x_0})^2 \cap \mathcal{A}$ of admissible test functions with $\varphi_n \xrightarrow{L^2} \mathbf{m}_\kappa|_{\omega_{x_0}}$. Furthermore we choose a sequence of positive numbers $\varepsilon_n \rightarrow 0$ with $\varepsilon_n \leq \varepsilon_{x_0}$ for all $n \in \mathbb{N}$. For each ε_n , let $\psi_n \in \mathcal{C}^\infty(\omega)$ with $0 \leq \psi_n \leq 1$, $\psi = 1$ on $\omega \setminus (\omega_{x_0} + B_{\varepsilon_n}(0))$, and $\psi = 0$ on ω_{x_0} be a smooth cut-off function. Then it holds that

$$\mathbf{m}_n := (\psi_n \mathbf{m}^* + \varphi_n) \xrightarrow{L^2} \mathbf{m}_\kappa \quad (3.36)$$

and each \mathbf{m}_n is admissible, i.e. $\mathbf{m}_n \in \mathcal{A}$. The variational inequality (3.35) yields

$$0 \geq De(\mathbf{m}^*)(\mathbf{m}^* - \mathbf{m}_n) = \int_\omega r \cdot (\mathbf{m}^* - \mathbf{m}_n) dx.$$

From the continuity of the L^2 scalar product we finally infer

$$\int_\omega r \cdot (\mathbf{m}^* - \mathbf{m}_n) \longrightarrow -\kappa \int_{\omega_{x_0} \cap \omega_k} r \cdot v \leq 0 \quad \text{for all } \kappa \in [-\delta_k, \delta_k]$$

which in turn proves $\int_{\omega_{x_0} \cap \omega_k} r \cdot v dx = 0$ by choosing $\kappa = \pm \delta_k$. This statement holds for all $k \in \mathbb{N}$ from which we conclude $\int_{\omega_{x_0} \cap \omega_{<}} r \cdot v dx = 0$. Finally since $v \in (L^\infty)^2$ and $x_0 \in \omega_{<}$ were arbitrary it follows that $r = 0$ on $\omega_{<}$.

Second we prove $r = -\lambda^* \mathbf{m}^*$ on ω_1 for some $\lambda^* \in L^2_{loc}(\omega)$ with $\lambda^* \geq 0$: Let $x_0 \in \omega_1$ be arbitrary and choose a neighborhood ω_{x_0} and a positive number ε_{x_0} as above. Let $v \in L^\infty(\omega)^2$ with $\mathbf{m}^*(x) \cdot v(x) < 0$ almost everywhere on ω_1 and let $\omega' \subseteq \omega$ be an arbitrary measurable subset. We define $\omega_k := \{x \in \omega_1 \mid \mathbf{m}^*(x) \cdot \frac{v(x)}{|v(x)|} \leq -\frac{1}{k}\}$ and $\delta_k := \frac{2}{k\|v\|_\infty}$. Then,

$$\mathbf{m}_\kappa := \mathbf{m}^* + \kappa \chi_{\omega_{x_0}} \chi_{\omega_k} \chi_{\omega'} v \in L^\infty(\omega)^2$$

satisfies the constraint $|\mathbf{m}_\kappa| \leq 1$ for all $\kappa \in [0, \delta_k]$. As above we construct a sequence of admissible functions

$$\mathbf{m}_n := \psi_n \mathbf{m}^* + \varphi_n \xrightarrow{L^2} \mathbf{m}_\kappa$$

and use the variational inequality (3.35) to prove

$$\int_{\omega} r \cdot (\mathbf{m}^* - \mathbf{m}_{\kappa}) dx \leq 0.$$

The continuity of the L^2 scalar product yields

$$- \int_{\omega_{x_0} \cap \omega_k \cap \omega'} r \cdot v dx \leq 0.$$

Since this holds for all measurable subsets ω' of ω we obtain $r \cdot v \geq 0$ almost everywhere on $\omega_{x_0} \cap \omega_k$. Again this holds for all $k \in \mathbb{N}$ and we therefore obtain $r \cdot v \geq 0$ almost everywhere on $\omega_{x_0} \cap \omega_1$ and in particular $r(x_0) \cdot v(x_0) \geq 0$. Since $x_0 \in \omega_1$ was arbitrary we finally conclude $r \cdot v \geq 0$ almost everywhere on ω_1 . Let R^α denote the pointwise rotation by angle α . The choice of $v = R^\alpha \mathbf{m}^*$ and $r \cdot R^\alpha \mathbf{m}^* \geq 0$ for all $\alpha \in (\pi/2, 3\pi/2)$ proves $r = -\lambda^* \mathbf{m}^*$ on ω_1 for some non-negative function λ^* . Recall that due to $|\mathbf{m}^*| = 1$ on ω_1 , the function \mathbf{m}^* can have no regularizing effect and λ^* must therefore be element of the same function space as r , i.e. $\lambda^* \in L^2_{loc}$ due to $\nabla u \in L^2_{loc}$.

We conclude the proof by summarizing the results. The L^2 representation r of $De(\mathbf{m}^*)$ satisfies $r = 0$ on $\omega_{<}$. Therefore, the choice of $\lambda^* = 0$ on $\omega_{<}$ and the choice of $\lambda^* \geq 0$ according to the second step of the proof yields

$$\begin{aligned} r &= -\lambda^* \mathbf{m}^* \\ \lambda^*(|\mathbf{m}^*| - 1) &= 0, \end{aligned}$$

which concludes the proof. We stress that the second equation in (3.33) enforces uniqueness of λ^* on the entire domain ω . \blacksquare

Since the penalized problem is unconstrained, the Euler-Lagrange equation simply reads $De_\varepsilon(\mathbf{m}_h^\varepsilon) = 0$. We have already computed the derivatives of the energy contributions that are also part of $e(\cdot)$. The computation of the derivative of $\|(|\mathbf{m}| - 1)_+\|_{L^2}^2$ can be found, e.g., in [Pra03].

Proposition 3.19 ([Pra03, Satz 2.18(iv)]). *Given the energy contribution $e^{(4)}(\mathbf{m}) = \|(|\mathbf{m}| - 1)_+\|_{L^2}^2$, its Gateaux derivative reads*

$$De^{(4)}(\mathbf{m})(\mathbf{w}) = \left(\frac{(|\mathbf{m}| - 1)_+}{|\mathbf{m}|} \mathbf{m}, \mathbf{w} \right)_{L^2}. \quad (3.37)$$

From this, we deduce the Euler-Lagrange equations for (M_h^ε) : Any minimizer $\mathbf{m}_h^\varepsilon \in X_h$ of the penalized energy $e_\varepsilon(\cdot)$ satisfies

$$\begin{aligned} (\nabla \cdot \mathbf{m}_h^\varepsilon, \nabla \cdot \mathbf{w}_h)_V + q(\mathbf{m}_{h,2}, \mathbf{w}_{h,2})_{L^2} - (\mathbf{f}, \mathbf{w}_h)_{L^2} + (\lambda_h^\varepsilon \mathbf{m}_h^\varepsilon, \mathbf{w}_h)_{L^2} &= 0, \\ \lambda_h^\varepsilon &= \frac{1}{\varepsilon} \frac{(|\mathbf{m}_h^\varepsilon| - 1)_+}{|\mathbf{m}_h^\varepsilon|} \end{aligned} \quad (3.38)$$

for all $\mathbf{w}_h \in X_h$. Moreover, we stress the relation

$$\langle \mathbf{m}^* - \mathbf{m}_h^\varepsilon, \mathbf{w}_h \rangle = -\langle \lambda^* \mathbf{m}^* - \lambda_h^\varepsilon \mathbf{m}_h^\varepsilon, \mathbf{w}_h \rangle \quad (3.39)$$

for all $\mathbf{w}_h \in X_h$.

3.3.2 A priori analysis

We aim at establishing an a priori error analysis for $\|\mathbf{m}^* - \mathbf{m}_h^\varepsilon\|$. To that end, we proceed in two steps. First, we estimate the error $\|\mathbf{m}^* - \mathbf{m}_0^\varepsilon\|$, i.e. the error introduced by the penalization in the continuous case. Second, we provide an estimate for $\|\mathbf{m}_0^\varepsilon - \mathbf{m}_h^\varepsilon\|$. In order to obtain convergence rates for any of those estimates, we need some additional regularity assumptions.

Regularity assumption for the magnetostatic potential u :

$$\begin{aligned} u &\in H^{3/2}(B_+) \text{ for all bounded } C^1\text{-domains } B_+ \subseteq (\mathbb{R}^2 \times \mathbb{R}_{\geq 0}) \text{ with } \omega \subseteq \partial B_+ \\ u &\in H^{3/2}(B_-) \text{ for all bounded } C^1\text{-domains } B_- \subseteq (\mathbb{R}^2 \times \mathbb{R}_{\leq 0}) \text{ with } \omega \subseteq \partial B_- \end{aligned} \quad (\text{U})$$

Lemma 3.20. *Let \mathbf{m}^* be a minimizer of (M) and assume that the magnetostatic potential u satisfies the regularity assumption (U). Then, the Lagrange multiplier λ^* of (3.33) satisfies $\lambda^* \in L^2(\omega)^2$, and in particular $\|\lambda^* \mathbf{m}^*\|_{L^2} < \infty$.*

Proof. The mapping properties of the trace operator from Proposition 2.1 yield $tu \in H^1(\omega)$, and hence $\nabla u \in L^2(\omega)$. Then in the Euler-Lagrange equation (3.33) all quantities are L^2 -functions and hence from $|\mathbf{m}^*(x)| = 1$ at all points where $\lambda^* \neq 0$, we conclude $\lambda^* \in L^2(\omega)$. \blacksquare

In [CP01], the penalty method for the large-body limit is studied. The resulting Euler-Lagrange equations for the constrained and the penalized model are quite similar to our results. In particular some estimates for the non-linear contributions of $\lambda^* \mathbf{m}^*$ and $\lambda_h^\varepsilon \mathbf{m}_h^\varepsilon$ are given that can be used in our context as well.

Lemma 3.21. *Let $\mathbf{m}, \tilde{\mathbf{m}} \in \mathcal{H}$ be two given magnetizations. Then, there holds the pointwise estimate*

$$\left(\frac{(|\mathbf{m}| - 1)_+}{|\mathbf{m}|} \mathbf{m} - \frac{(|\tilde{\mathbf{m}}| - 1)_+}{|\tilde{\mathbf{m}}|} \tilde{\mathbf{m}} \right) \cdot (\mathbf{m} - \tilde{\mathbf{m}}) \geq 0. \quad (3.40)$$

Let furthermore \mathbf{m} be such that $|\mathbf{m}| \leq 1$ and $\mu \in L^2$ with $\mu(|\mathbf{m}| - 1) = 0$ and let \mathbf{m}_h^ε be a solution of (M_h^ε) for $\varepsilon > 0$. Then, with the quantity λ_h^ε defined in (3.38), the inequality

$$-\langle \mu \mathbf{m} - \lambda_h^\varepsilon \mathbf{m}_h^\varepsilon, \mathbf{m} - \mathbf{m}_h^\varepsilon \rangle \leq \frac{\varepsilon}{2} \|\mu \mathbf{m}\|_{L^2}^2 - \frac{\varepsilon}{2} \|\lambda_h^\varepsilon \mathbf{m}_h^\varepsilon\|_{L^2}^2 \quad (3.41)$$

holds true, where ω_ε denotes the subset on which $\lambda_h^\varepsilon > 0$.

Proof. In [CP01, Proof of Theorem 3.1] the pointwise estimate

$$0 \leq ((|b| - 1)_+ b / |b| - (|a| - 1)_+ a / |a|) \cdot (b - a) \quad (3.42)$$

is established, which is a more general formulation of (3.40). The inequality (3.41) is stated pointwise in [CP01, Proof of Theorem 4.3] and our formulation follows by integration over the domain ω . \blacksquare

We are now ready to establish the first a priori error estimate.

Theorem 3.22. *The continuous solutions \mathbf{m}^* of (M) and \mathbf{m}_0^ε of (M_0^ε) satisfy the a priori error estimate*

$$\|\mathbf{m}^* - \mathbf{m}_0^\varepsilon\|^2 + \frac{\varepsilon}{2} \|\lambda_0^\varepsilon \mathbf{m}_0^\varepsilon\|_{L^2}^2 \leq \frac{\varepsilon}{2} \|\lambda^* \mathbf{m}^*\|_{L^2}^2 \quad (3.43)$$

with the Lagrange multiplier λ^* of (3.33). Assume that the magnetostatic potential u of the solution \mathbf{m}^* of (M) satisfies the regularity assumption (U), then

$$\|\mathbf{m}^* - \mathbf{m}_0^\varepsilon\| = \mathcal{O}(\sqrt{\varepsilon}). \quad (3.44)$$

Proof. Recall the relation (3.39)

$$\langle \mathbf{m}^* - \mathbf{m}_0^\varepsilon, \mathbf{m}^* - \mathbf{m}_0^\varepsilon \rangle = -\langle \lambda^* \mathbf{m}^* - \lambda_0^\varepsilon \mathbf{m}_0^\varepsilon, \mathbf{m}^* - \mathbf{m}_0^\varepsilon \rangle.$$

If u satisfies the additional regularity assumption, then by Lemma 3.20 it holds that $\lambda^* \in L^2$. With (3.41), we conclude

$$\|\mathbf{m}^* - \mathbf{m}_0^\varepsilon\|^2 \leq \frac{\varepsilon}{2} \|\lambda^* \mathbf{m}^*\|_{L^2}^2 - \frac{\varepsilon}{2} \|\lambda_0^\varepsilon \mathbf{m}_0^\varepsilon\|_{L^2}^2.$$

Note that $\|\lambda_0^\varepsilon \mathbf{m}_0^\varepsilon\|_{L^2} \geq 0$. Hence, dropping this term and taking the square root concludes the proof. \blacksquare

Lemma 3.23. *Assume strong L^2 convergence $\|\mathbf{m}_0^\varepsilon - \mathbf{m}_h^\varepsilon\|_{L^2} \rightarrow 0$ as $h \rightarrow 0$. Then it holds that*

$$\|\lambda_h^\varepsilon \mathbf{m}_h^\varepsilon - \lambda_0^\varepsilon \mathbf{m}_0^\varepsilon\|_{L^2} \rightarrow 0 \text{ as } h \rightarrow 0.$$

Proof. As mentioned in the proof of [Pra03, Satz 2.18], the mapping $g \mapsto g_+ = (g + |g|)/2$ is continuous in L^2 . From that we obtain that the sequence of mappings

$$\mathbf{m} \mapsto |\mathbf{m}| \mapsto |\mathbf{m}| - 1 \mapsto \frac{1}{\varepsilon} (|\mathbf{m}| - 1)_+ \mapsto \frac{(|\mathbf{m}| - 1)_+}{\varepsilon ((|\mathbf{m}| - 1)_+ + 1)}$$

is continuous in L^2 since the denominator is bounded away from 0. Note that $(|\mathbf{m}(x)| - 1)_+ + 1 = 1$ for $|\mathbf{m}(x)| \leq 1$ and $(|\mathbf{m}(x)| - 1)_+ + 1 = |\mathbf{m}(x)|$ for $|\mathbf{m}(x)| \geq 1$. Multiplication with \mathbf{m} , thus shows

$$\mathbf{m} \mapsto \frac{(|\mathbf{m}| - 1)_+}{\varepsilon ((|\mathbf{m}| - 1)_+ + 1)} \mathbf{m} = \lambda^\varepsilon \mathbf{m}$$

to be continuous. From $\mathbf{m}_h^\varepsilon \rightarrow \mathbf{m}_0^\varepsilon \in L^2$ we obtain the desired result. \blacksquare

As a second — and more involved — step, we provide an a priori error estimate for the discretization of the penalized minimization problem.

Theorem 3.24. *Let $\varepsilon, h > 0$. Then, the solutions \mathbf{m}_0^ε of (M_0^ε) and \mathbf{m}_h^ε of (M_h^ε) satisfy the a priori error estimate*

$$\frac{1}{2} \|\mathbf{m}_0^\varepsilon - \mathbf{m}_h^\varepsilon\|^2 + (\lambda_0^\varepsilon \mathbf{m}_0^\varepsilon - \lambda_h^\varepsilon \mathbf{m}_h^\varepsilon, \mathbf{m}_0^\varepsilon - \mathbf{m}_h^\varepsilon)_{L^2} \leq \inf_{\mathbf{w}_h \in X_h} \left(\frac{1}{2} \|\mathbf{m}_0^\varepsilon - \mathbf{w}_h\|^2 + \|\lambda_0^\varepsilon \mathbf{m}_0^\varepsilon - \lambda_h^\varepsilon \mathbf{m}_h^\varepsilon\|_{L^2} \|\mathbf{m}_0^\varepsilon - \mathbf{w}_h\| \right). \quad (3.45)$$

If \mathbf{m}_0^ε satisfies the additional regularity assumptions $\mathbf{m}_0^\varepsilon \in H^1(\omega)^2$ and $\nabla \cdot \mathbf{m}_0^\varepsilon \in H^{1/2}(\omega)$, we have

$$\|\mathbf{m}_0^\varepsilon - \mathbf{m}_h^\varepsilon\|^2 \leq C(h^2 + h \|\lambda_0^\varepsilon \mathbf{m}_0^\varepsilon - \lambda_h^\varepsilon \mathbf{m}_h^\varepsilon\|_{L^2}) (\|\nabla \cdot \mathbf{m}_0^\varepsilon\|_{H^{1/2}} + \|\mathbf{m}_0^\varepsilon\|_{H^1}). \quad (3.46)$$

Assume $\sup_h \|\lambda_0^\varepsilon \mathbf{m}_0^\varepsilon - \lambda_h^\varepsilon \mathbf{m}_h^\varepsilon\|_{L^2} < \infty$, then we obtain

$$\|\mathbf{m}_0^\varepsilon - \mathbf{m}_h^\varepsilon\| = \mathcal{O}(\sqrt{h}).$$

Proof. Proof of (3.45):

First, we use Cauchy-Schwarz' as well as Young's inequality to obtain

$$\begin{aligned} \|\mathbf{m}_0^\varepsilon - \mathbf{m}_h^\varepsilon\|^2 &= \langle \mathbf{m}_0^\varepsilon - \mathbf{m}_h^\varepsilon, \mathbf{m}_0^\varepsilon - \mathbf{m}_h^\varepsilon \rangle \\ &= \langle \mathbf{m}_0^\varepsilon - \mathbf{m}_h^\varepsilon, \mathbf{m}_0^\varepsilon - \mathbf{w}_h \rangle + \langle \mathbf{m}_0^\varepsilon - \mathbf{m}_h^\varepsilon, \mathbf{w}_h - \mathbf{m}_h^\varepsilon \rangle \\ &\leq \|\mathbf{m}_0^\varepsilon - \mathbf{m}_h^\varepsilon\| \cdot \|\mathbf{m}_0^\varepsilon - \mathbf{w}_h\| + \langle \mathbf{m}_0^\varepsilon - \mathbf{m}_h^\varepsilon, \mathbf{w}_h - \mathbf{m}_h^\varepsilon \rangle \\ &\leq \frac{1}{2} \|\mathbf{m}_0^\varepsilon - \mathbf{m}_h^\varepsilon\|^2 + \frac{1}{2} \|\mathbf{m}_0^\varepsilon - \mathbf{w}_h\|^2 + \langle \mathbf{m}_0^\varepsilon - \mathbf{m}_h^\varepsilon, \mathbf{w}_h - \mathbf{m}_h^\varepsilon \rangle \end{aligned}$$

for arbitrary $\mathbf{w}_h \in X_h$. Next, we stress that both, the continuous solution \mathbf{m}_0^ε as well as the discrete solution \mathbf{m}_h^ε , satisfy the discrete Euler-Lagrange equation $De_\varepsilon(\mathbf{m})(\tilde{\mathbf{w}}_h) = 0$ for all $\tilde{\mathbf{w}}_h \in X_h$. In particular, it holds that

$$\begin{aligned} \langle \mathbf{m}_0^\varepsilon - \mathbf{m}_h^\varepsilon, \mathbf{w}_h - \mathbf{m}_h^\varepsilon \rangle &= -(\lambda_0^\varepsilon \mathbf{m}_0^\varepsilon - \lambda_h^\varepsilon \mathbf{m}_h^\varepsilon, \mathbf{w}_h - \mathbf{m}_h^\varepsilon)_{L^2} \\ &= -(\lambda_0^\varepsilon \mathbf{m}_0^\varepsilon - \lambda_h^\varepsilon \mathbf{m}_h^\varepsilon, \mathbf{m}_0^\varepsilon - \mathbf{m}_h^\varepsilon)_{L^2} + (\lambda_0^\varepsilon \mathbf{m}_0^\varepsilon - \lambda_h^\varepsilon \mathbf{m}_h^\varepsilon, \mathbf{m}_0^\varepsilon - \mathbf{w}_h)_{L^2} \end{aligned}$$

since $\mathbf{w}_h - \mathbf{m}_h^\varepsilon \in X_h$. Using again Cauchy-Schwarz' inequality, altogether, we have

$$\frac{1}{2} \|\mathbf{m}_0^\varepsilon - \mathbf{m}_h^\varepsilon\|^2 + (\lambda_0^\varepsilon \mathbf{m}_0^\varepsilon - \lambda_h^\varepsilon \mathbf{m}_h^\varepsilon, \mathbf{m}_0^\varepsilon - \mathbf{m}_h^\varepsilon)_{L^2} \leq \frac{1}{2} \|\mathbf{m}_0^\varepsilon - \mathbf{w}_h\|^2 + \|\lambda_0^\varepsilon \mathbf{m}_0^\varepsilon - \lambda_h^\varepsilon \mathbf{m}_h^\varepsilon\|_{L^2} \|\mathbf{m}_0^\varepsilon - \mathbf{w}_h\|_{L^2}$$

for arbitrary $\mathbf{w}_h \in X_h$. Taking the infimum over all $\mathbf{w}_h \in X_h$ yields the statement.

Rate of convergence:

To obtain the desired rate of convergence, we assume $\mathbf{m}_h^\varepsilon \in H^1(\omega)$ as well as $\nabla \cdot \mathbf{m}_h^\varepsilon \in H^{1/2}(\omega)$. Recall the interpolation operator $\Pi : H^1(\omega)^2 \rightarrow RT^0(\mathcal{T}_h)$ of Theorem 3.1. The choice of $\mathbf{w}_h = \Pi \mathbf{m}_0^\varepsilon$ as well as the observation that $(\lambda_0^\varepsilon \mathbf{m}_0^\varepsilon - \lambda_h^\varepsilon \mathbf{m}_h^\varepsilon, \mathbf{m}_0^\varepsilon - \mathbf{m}_h^\varepsilon)_{L^2} \geq 0$ according to (3.40) conclude the proof. ■

Remark. If \mathbf{m}_0^ε is smooth, then Lemma 3.23 provides a sufficient condition to obtain the a priori estimate $\|\mathbf{m}_0^\varepsilon - \mathbf{m}_h^\varepsilon\| = \mathcal{O}(\sqrt{h})$. We stress that we observe in all of our numerical experiments not only convergence in the energy norm $\|\cdot\|$, but also – at a possibly lower rate – convergence in the full space norm $\|\cdot\|$. □

Corollary 3.25. *Assume $\mathbf{m}_0^\varepsilon \in H^1$ with $\sup_\varepsilon \|\mathbf{m}_0^\varepsilon\|_{H^1} < \infty$ and $\nabla \cdot \mathbf{m}_0^\varepsilon \in H^{1/2}$ with $\sup_\varepsilon \|\nabla \cdot \mathbf{m}_0^\varepsilon\|_{H^{1/2}} < \infty$. Let the magnetostatic potential u of the solution \mathbf{m}^* of (M) satisfy the regularity assumption (U). Moreover, assume $\sup_{h,\varepsilon} \|\lambda_0^\varepsilon \mathbf{m}_0^\varepsilon - \lambda_h^\varepsilon \mathbf{m}_h^\varepsilon\|_{L^2} < \infty$. Then, we obtain the a priori error estimate*

$$\|\mathbf{m}^* - \mathbf{m}_h^\varepsilon\| = \mathcal{O}(\sqrt{h} + \sqrt{\varepsilon}). \quad (3.47)$$

Proof. The triangle inequality yields

$$\|\mathbf{m}^* - \mathbf{m}_h^\varepsilon\| \leq \|\mathbf{m}^* - \mathbf{m}_0^\varepsilon\| + \|\mathbf{m}_0^\varepsilon - \mathbf{m}_h^\varepsilon\|.$$

Theorems 3.22 and 3.24 yields the stated rate of convergence, since we assumed uniform bounds for all quantities that depend on h or ε . ■

Remark. The regularity assumptions of Corollary 3.25 with respect to \mathbf{m}_0^ε and $\nabla \cdot \mathbf{m}_0^\varepsilon$ are satisfied if

- $\mathbf{m}_0^\varepsilon \rightarrow \mathbf{m}^* \in H^1$ as $\varepsilon \rightarrow 0$,
- $\nabla \cdot \mathbf{m}_0^\varepsilon \rightarrow \nabla \cdot \mathbf{m}^* \in H^{1/2}$ as $\varepsilon \rightarrow 0$,

hold true. □

Remark. The a priori estimate (3.47) indicates that $\varepsilon = h$ is a reasonable choice when dealing with uniform meshes. This is because, asymptotically, the error contributions are of the same order. □

Remark. The result of Corollary 3.25 is somewhat unsatisfactory. First, the rate is only obtained under strong regularity and uniformity assumptions. In particular, we have no proof for convergence in the energy norm, even without rates. □

3.4 Adaptive algorithm

In this section, we propose an adaptive algorithm for the efficient simulation of the model problem. A uniform mesh may be suboptimal for several reasons. For example, we expect to encounter generic edge singularities of $\nabla \cdot \mathbf{m}$. To explain this, consider the case of soft material with $q = 0$ and weak constant applied field $|\mathbf{f}| \ll 1$. With these restrictions the side-constraint is not active. As observed by [Drw08], the fact that \mathbf{f} is constant means

$$\nabla(\mathbf{f} \cdot x) = \mathbf{f}.$$

Hence, integration by parts implies

$$\int_{\omega} \mathbf{f} \cdot \mathbf{m} \, dx = \int_{\omega} \nabla(\mathbf{f} \cdot x) \cdot \mathbf{m} \, dx = \int_{\omega} (\mathbf{f} \cdot x) \nabla \cdot \mathbf{m} \, dx - \int_{\omega} (\mathbf{f} \cdot x)(\mathbf{m} \cdot \mathbf{n}) \, dx.$$

Recall the side constraint $\mathbf{m} \cdot \mathbf{n} = 0$, then for soft thin films with $q = 0$ it holds that

$$e(\mathbf{m}) = \frac{1}{2} \|\nabla \cdot \mathbf{m}\|_V^2 - \int_{\omega} (\mathbf{f} \cdot x)(\nabla \cdot \mathbf{m}) \, dx$$

and the Euler-Lagrange equation reads

$$V(\nabla \cdot \mathbf{m}) = \mathbf{f} \cdot x, \tag{3.48}$$

which, with the right-hand side $\mathbf{f} \cdot x$, is the weakly singular integral equation associated with the Laplacian for a screen in 3D. The generic singularities of the solution $\nabla \cdot \mathbf{m}$ have been studied, e.g., in [ESAES90]. The author of [Drw08] concludes that the generic singular behavior of $\nabla \cdot \mathbf{m}$ suggests an a priori refinement of the mesh towards the edges of the simulation domain ω .

This a priori refinement seems to be very efficient in many soft thin-film simulations, even when the applied field is large and the constraint $|\mathbf{m}| \leq 1$ is active. However, we treat the minimization problem (M) as a prototype for a large class of energy functionals. In a more general case, such as for non-constant \mathbf{f} or complicated geometries, an automatic h -refinement strategy that is not based on a priori knowledge seems to be preferable. Moreover, a refinement towards all edges might lead to higher computational effort than necessary. Possible additional regularity of $\nabla \cdot \mathbf{m}$ is ignored. Finally, we also treat $q > 0$. The energy norm includes an L^2 -component which also must be considered.

In [CP01] a residual error estimator for the large-body limit is derived. The construction is based on pointwise KKT-conditions for the discrete penalized problem. The pointwise nature of the equations is a consequence of the L^2 setting, i.e. the absence of weak negative order Sobolev norms. It doesn't seem possible to use similar ideas to obtain a simple residual based error estimator for our model problem (M). For the estimation of the error contribution caused by the discretization, we suggest heuristic error estimation by space enrichment, i.e. the simple $h - h/2$ strategy.

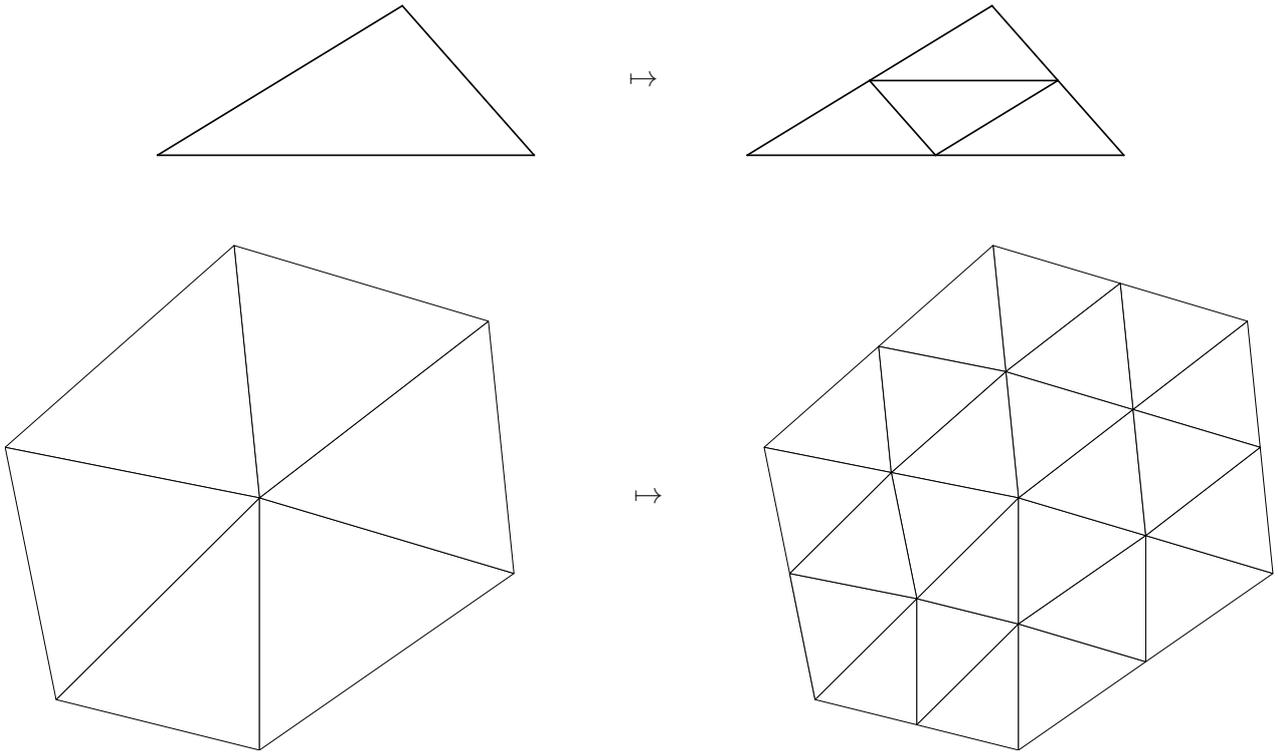


Figure 3.2: Red (uniform) refinement of a triangle (top). Uniform refinement of a sample mesh (bottom).

3.4.1 Error estimation by space enrichment

Let \mathcal{T}_ℓ denote some mesh of the simulation domain. We refer to the uniform refinement of \mathcal{T}_ℓ by $\widehat{\mathcal{T}}_\ell$, i.e. $\widehat{\mathcal{T}}_\ell$ is obtained by splitting each triangle of \mathcal{T}_ℓ into four similar ones, see Figure 3.2. Let \mathbf{m}_ℓ and $\widehat{\mathbf{m}}_\ell$ be the corresponding discrete minimizers for some fixed penalty parameter $\varepsilon > 0$. We estimate the error by

$$\|\mathbf{m}_0^\varepsilon - \mathbf{m}_\ell\| \approx \|\widehat{\mathbf{m}}_\ell - \mathbf{m}_\ell\| =: \eta_\ell^H. \quad (3.49)$$

This strategy is natural and quite popular, e.g., in the context of ordinary differential equations. Recently, it has been successfully applied to linear elliptic partial differential equations in the context of finite element [FLOP10] and boundary element methods [FLP08].

The analysis of the $h - h/2$ error estimator is based upon either some best approximation property or the so-called *saturation assumption*

$$\|\mathbf{m}_0^\varepsilon - \widehat{\mathbf{m}}_\ell\| \leq C_{sat} \|\mathbf{m}_0^\varepsilon - \mathbf{m}_\ell\| \text{ with } \ell\text{-independent } C_{sat} \in (0, 1). \quad (S)$$

Theorem 3.26. *Under the saturation assumption (S), it holds that*

$$C_{eff}^{-1} \eta_\ell^H \leq \|\mathbf{m}_0^\varepsilon - \mathbf{m}_\ell\| \leq C_{rel} \eta_\ell^H \quad (3.50)$$

with the ℓ -independent constants $C_{eff} = (1 + C_{sat})$ and $C_{rel} = (1 - C_{sat})^{-1}$. The lower bound is referred to as efficiency and the upper bound as reliability of the error estimator.

Proof. An application of the triangle inequality and the saturation assumption yield

$$\begin{aligned}\eta_\ell^H &= \|\widehat{\mathbf{m}}_\ell - \mathbf{m}_\ell\| \leq \|\mathbf{m}_0^\varepsilon - \widehat{\mathbf{m}}_\ell\| + \|\mathbf{m}_0^\varepsilon - \mathbf{m}_\ell\| \\ &\leq (1 + C_{sat})\|\mathbf{m}_0^\varepsilon - \mathbf{m}_\ell\|.\end{aligned}$$

The reliability estimate is obtained by

$$\begin{aligned}\|\mathbf{m} - \mathbf{m}_\ell\| &\leq \|\mathbf{m}_0^\varepsilon - \widehat{\mathbf{m}}_\ell\| + \|\widehat{\mathbf{m}}_\ell - \mathbf{m}_\ell\| \\ &\leq C_{sat}\|\mathbf{m}_0^\varepsilon - \mathbf{m}_\ell\| + \eta_\ell^H,\end{aligned}$$

which after reordering the terms yields the statement. \blacksquare

Remark. Note that the efficiency with $C_{eff} = 1$ holds if the approximation property

$$\|\mathbf{m}_0^\varepsilon - \widehat{\mathbf{m}}_\ell\| \leq \|\mathbf{m}_0^\varepsilon - \mathbf{m}_\ell\|,$$

is true. \square

Remark. The saturation assumption was proven for some P^1 finite element method for the Dirichlet problem with the Laplace operator in [DN02], up to data oscillations. In the context of boundary element methods it is completely open, which is a drawback for our model problem. The major difficulty is the treatment of the non-local operator V . We stress, however, that it is observed empirically for Symm's integral equation (3.48), cf. [EFLFP09]. \square

The estimator η_ℓ^H cannot be used to steer an adaptive mesh-refinement since it includes the non-local V -norm. Recall that

$$\|\phi\|_V^2 = \frac{1}{4\pi} \int_\omega \int_\omega \frac{\phi(x)\phi(y)}{|x-y|} dy dx,$$

which is not straight forwardly written a sum of local contributions. This is different, e.g., for the L^2 norm where

$$\|\phi\|_{L^2(\omega)}^2 = \sum_{T \in \mathcal{T}_\ell} \|\phi\|_{L^2(T)}^2.$$

The localization of the $\widetilde{H}^{-1/2}$ -norm is discussed for certain discrete functions in [Fae02]. We follow here a different and more easy to implement approach based on a local inverse estimate.

Definition. We define the local mesh-width functions $h_\ell, \varrho_\ell \in L^\infty(\omega)$: The function h_ℓ is defined by

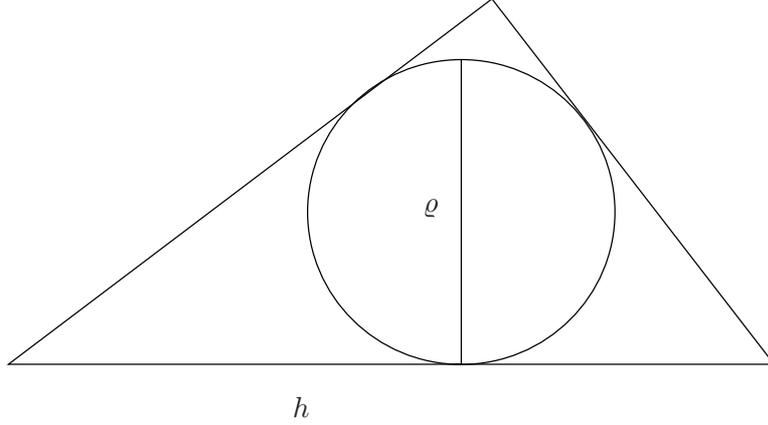
$$h_\ell|_T = \text{diam}(T).$$

The function $\varrho_\ell|_T$ is the maximal diameter of an inscribed ball in T , see Figure 3.3. \square

Definition. We call a sequence of meshes *isotropic* if the shape-regularity $\sigma(\mathcal{T}_\ell) := \max_{T \in \mathcal{T}_\ell} \{h_\ell|_T / \varrho_\ell|_T\}$ satisfies a uniform bound

$$\sup_{\ell \in \mathbb{N}} \sigma(\mathcal{T}_\ell) \leq C_\sigma < \infty.$$

\square

Figure 3.3: Local mesh-width functions h_ℓ and ϱ_ℓ .

Proposition 3.27 ([GHS05, Theorem 3.6]). *For any $v_\ell \in \mathcal{P}^0(\mathcal{T}_\ell)$ it holds that*

$$\left\| h_\ell^{1/2} v_\ell \right\|_{L^2(\omega)} \leq C_{inv} \|v_\ell\|_V. \quad (3.51)$$

The constant C_{inv} depends on the shape regularity constant $\sigma(\mathcal{T}_\ell)$. In particular one can find an ℓ independent constant C_{inv} for an isotropic sequence of meshes.

Corollary 3.28. *Let $(\mathcal{T}_\ell)_{\ell \in \mathbb{N}}$ be an isotropic sequence of meshes. Then the local error indicator*

$$\mu_\ell^H(T)^2 := \left\| h_\ell^{1/2} \nabla \cdot (\widehat{\mathbf{m}}_\ell - \mathbf{m}_\ell) \right\|_{L^2(T)}^2 + \|\widehat{\mathbf{m}}_{\ell,2} - \mathbf{m}_{\ell,2}\|_{L^2(T)}^2 \quad (3.52)$$

satisfies the efficiency estimate

$$(\mu_\ell^H)^2 := \sum_{T \in \mathcal{T}_\ell} \mu_\ell^H(T)^2 \lesssim (\eta_\ell^H)^2.$$

Remark. We use the local error indicator μ_ℓ^H to decide which triangles should be refined, i.e. refine the elements $T \in \mathcal{T}_\ell$ where $\mu_\ell^H(T)$ is large. The local contribution $\mu_{\ell,V}^H := \|h^{1/2} \nabla \cdot (\widehat{\mathbf{m}}_\ell - \mathbf{m}_\ell)\|_{L^2}$ only gives a lower bound $\mu_{\ell,V}^H \lesssim \eta_{\ell,V}^H := \|\nabla \cdot (\widehat{\mathbf{m}}_\ell - \mathbf{m}_\ell)\|_V$ up to some in general unknown constant. In contrast, the L^2 component $\mu_{\ell,L^2}^H := \|\widehat{\mathbf{m}}_{\ell,2} - \mathbf{m}_{\ell,2}\|_{L^2} =: \eta_{\ell,L^2}^H$ does not rescale the norm. In our experiments we observe that the contribution $\mu_{\ell,V}^H$ also gives an upper bound so that it is equivalent to $\eta_{\ell,V}^H$. However, in all our simulations $\eta_{\ell,V}^H < \mu_{\ell,V}^H$ so that the V -norm contribution of the total error in the energy norm is overestimated. We observed an improvement in the behavior of the mesh-refinement when we balance the contributions in the following sense: Define the rescaled local error indicator

$$\widetilde{\mu}_{\ell,V}^H(T)^2 := \frac{(\eta_{\ell,V}^H)^2}{(\mu_{\ell,V}^H)^2} \mu_{\ell,V}^H(T)^2,$$

then $\widetilde{\mu}_\ell^H := ((\widetilde{\mu}_{\ell,V}^H)^2 + (\mu_{\ell,L^2}^H)^2)^{1/2}$ satisfies $\widetilde{\mu}_\ell^H = \eta_\ell^H$. This provides a proper scaling of the V -norm and the L^2 -norm contributions of the total error. \square

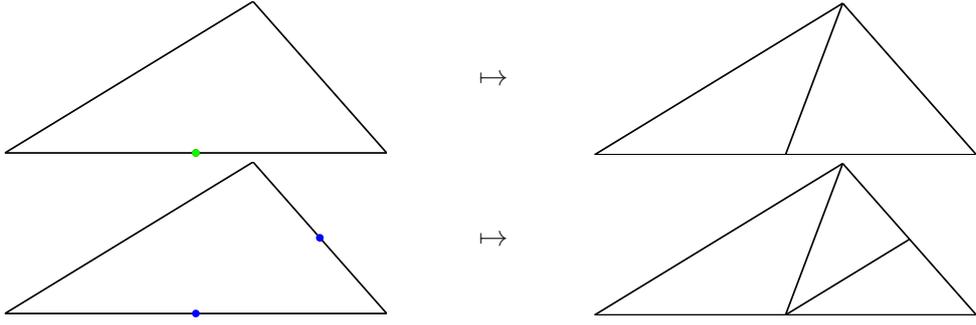


Figure 3.4: Green (top) and blue (bottom) refinement rules. In any case the longest edge is split first.

3.4.2 A heuristic adaptive algorithm

The inverse estimate (3.51) requires the use of an isotropic sequence of meshes, i.e. $\sup_{\ell \in \mathbb{N}} \sigma(\mathcal{T}_\ell) \leq C_\sigma < \infty$. The space $RT^0(\mathcal{T}_\ell)$ requires a regular mesh, i.e. no hanging nodes are allowed.

To ensure these restrictions throughout our adaptive loop, we use the red-green-blue refinement strategy. It is discussed in [Ver96] in detail, and we present the strategy here only briefly.

Adaptive Loop: Input: initial mesh \mathcal{T}_0 , \mathbf{f} , adaptivity parameter $\theta \in (0, 1)$, and some tolerance τ . Set $\ell = 0$ and do

1. compute uniform refinement $\widehat{\mathcal{T}}_\ell$
2. compute solutions $\widehat{\mathbf{m}}_\ell$ and \mathbf{m}_ℓ
3. compute error estimator η_ℓ^H as well as local indicators $\tilde{\mu}_\ell^H(T)$.
4. if $\eta_\ell^H \leq \tau$ stop, otherwise find a (minimal) set of marked elements $\mathcal{M}_\ell \subseteq \mathcal{T}_\ell$ such that

$$\theta \sum_{T \in \mathcal{T}_\ell} \tilde{\mu}_\ell^H(T)^2 \leq \sum_{T \in \mathcal{M}_\ell} \tilde{\mu}_\ell^H(T)^2. \quad (3.53)$$

5. refine (at least) marked elements and construct a new (regular) mesh $\mathcal{T}_{\ell+1}$
6. $\ell \mapsto \ell + 1$ and goto (1)

The red-green-blue refinement strategy specifies how the new mesh $\mathcal{T}_{\ell+1}$ is constructed. A marked element $T \in \mathcal{M}_\ell$ is refined *red*. All edges are split at their midpoints and four similar triangles T_1, \dots, T_4 are created, cf. Figure 3.2. In general $\mathcal{M}_\ell \subsetneq \mathcal{T}_\ell$ so that hanging nodes would be created. To avoid this, we mark further edges for refinement in a second step: In each triangle where some hanging node would be created, we first mark its longest edge and all the edges with hanging nodes for refinement. Then, the triangles are split either red, green, or blue (cf. Figure 3.4), depending on how many edges are marked. This recursive marking and the refinement is visualized in Figure 3.5.

The fact that we always split the longest edge ensures that the minimal interior angle $\alpha(T) := \min\{\sphericalangle CAB, \sphericalangle ABC, \sphericalangle BCA\}$ is bounded from below. There exists some ℓ -independent constant C_α such that

$$C_\alpha \leq \inf_{\ell \in \mathbb{N}} \min_{T \in \mathcal{T}_\ell} \alpha(T).$$

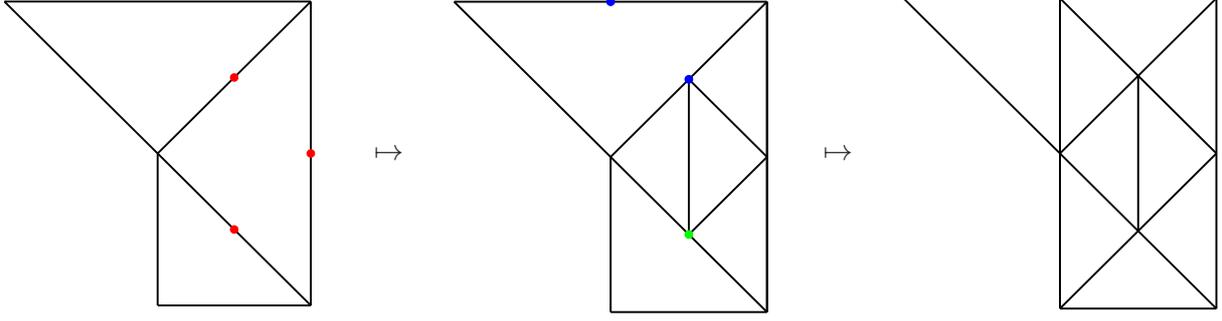


Figure 3.5: Illustration of red-green-blue refinement: Assume the right central triangle was marked for refinement. Then all of its edges are marked, hence the triangle is refined red. The hanging node on the top triangle effects that additionally its longest edge is marked as well. This leads to blue refinement. The hanging node on the bottom triangle is already on the longest edge. Therefore, it is refined green.

This in turn ensures that the sequence of generated meshes is isotropic, cf. [Ver96].

Remark. The marking criterion (3.53) was introduced in [Dör96] to prove convergence for some P^1 finite element method for the Poisson equation. Basically, we mark enough elements to dominate the total error up to some given fraction θ . In [CKNS08] it was observed that this marking criterion is necessary for convergence of the underlying method. If some marking criterion leads to convergence, then essentially it already implies the Dörfler criterion (3.53) for some $\theta \in (0, 1)$. \square

Remark. The adaptive loop as presented here was proven recently to be convergent for the simplified case (3.48), cf. [AFLP10]. Convergence is understood there in the sense that the adaptive algorithm ensures estimator convergence $\eta_\ell \rightarrow 0$. Hence, under the saturation assumption (S), convergence of the error $\|\mathbf{m}_0^\varepsilon - \mathbf{m}_\ell\| \rightarrow 0$ follows. In the work [AFLP10], the authors use a different mesh-refinement strategy, i.e. newest vertex bisection, which is necessary in their convergence proof. However, the red-green-blue refinement seems more natural and is more popular among the engineering community. \square

We have discussed in depth a strategy for the estimation of the discretization error and the steering of an adaptive mesh-refinement. All arguments apply to our discretized and penalized model problem (M_h^ε) for some fixed penalty parameter $\varepsilon > 0$. However, the penalty-scheme introduces itself an error that has not been taken into account, yet. We stress that according to Lemma 3.9 the sequence of energies $e_{\varepsilon_n}(\mathbf{m}_h^{\varepsilon_n})$ satisfies

$$\lim_{n \rightarrow \infty} \frac{1}{\varepsilon_n^\alpha} \left\| (|\mathbf{m}_h^{\varepsilon_n}| - 1)_+ \right\|_{L^2(\omega)}^2 = 0 \quad (3.54)$$

for any choice of $\alpha \in [0, 1)$. This is a consequence of the weak convergence of the sequence. In our numerical experiment, we observe that (3.54) even holds for the choice of $\alpha = 1$.

In the works [CP01, Pra03], an h -adaptive algorithm for the penalty-method applied to the large-body limit in micromagnetics is proposed. There, the authors succeeded to derive reliable a posteriori error estimates for the error $\|\mathbf{m} - \mathbf{m}_h^\varepsilon\|$. However, the error estimator does not reveal the contributions of the total error stemming from the discretization and from the penalization. From an a priori result of the kind

$$\|\mathbf{m} - \mathbf{m}_h^\varepsilon\| \leq \mathcal{O}(h^\alpha + \varepsilon^\beta)$$

they conclude that in the adaptive algorithm the penalty parameter should be chosen locally as $\varepsilon|_T = h_\ell|_T^{\alpha/\beta}$. However, we experienced that this strategy is not optimal in all cases. In all numerical experiments studied in [CP01, Pra03], singularities of the magnetization \mathbf{m} and the potential u appear only where the constraint is active. In this case it is clear that local refinement of the mesh with simultaneous local refinement of ε leads to good convergence behavior. However, we experienced that, at least for our thin-film model, it is easy to construct examples where the penalization is active in a very large region of ω , but the solution has singularities only where $|\mathbf{m}|$ is small. Numerical experiments covering this setting are provided in Chapter 5.

Therefore, it is necessary to find some h -independent strategy to steer the local refinement of ε . In light of (3.54), and observing that the statement also holds for $\alpha = 1$ empirically, we propose to refine ε locally, where $\frac{1}{\varepsilon} \|(|\mathbf{m}_h^\varepsilon| - 1)_+\|_{L^2(T)}^2$ is large. We introduce the error indicator

$$\mu_\ell^\varepsilon(T)^2 = \frac{1}{2\varepsilon} \|(|\mathbf{m}_\ell| - 1)_+\|_{L^2(T)}^2 \quad (3.55)$$

and define

$$\eta_\ell^\varepsilon := \left(\sum_{T \in \mathcal{T}_\ell} \mu_\ell^\varepsilon \right)^{1/2}. \quad (3.56)$$

We extend our adaptive algorithm to steer also local ε -refinement: Start with some initial penalty parameter $\varepsilon_0|_T = \varepsilon$. In step (3), we additionally compute the estimator η_ℓ^ε . In step (4), the stopping criterion now reads

$$\eta_\ell := \eta_\ell^H + \eta_\ell^\varepsilon < \tau. \quad (3.57)$$

As error indicator we use the combined quantity

$$\mu_\ell(T)^2 := \tilde{\mu}_\ell^H(T)^2 + \mu_\ell^\varepsilon(T)^2.$$

Let $T \in \mathcal{M}_\ell$ be a marked element. Choose some constant $C_{\text{refine}} > 1$. If

$$\tilde{\mu}_\ell^H(T)^2 > C_{\text{refine}} \mu_\ell^\varepsilon(T)^2, \quad (3.58)$$

refine the triangle (red). If

$$\mu_\ell^\varepsilon(T)^2 > C_{\text{refine}} \tilde{\mu}_\ell^H(T)^2, \quad (3.59)$$

don't refine the triangle, but set $\varepsilon_{\ell+1}|_T = \frac{\varepsilon_\ell|_T}{2}$. Otherwise refine both, the triangle (red) and the local penalty parameter. In our experiments we use the choice $C_{\text{refine}} = 2$ throughout.

The proposed algorithm may steer the mesh and penalty parameter in an intelligent manner. It is not at all clear the error contributions stemming from the discretization and the penalization are independent. We are estimating these error contributions separately in our algorithm. Therefore, the algorithm can only perform well if local refinement of ε does not increase the local discretization error and vice versa. The weak convergence result of Corollary 3.12 indicates that we may expect some uniformity in h and ε . Recall that any choice of $(h_n, \varepsilon_n) \rightarrow (0, 0)$ provides weak convergence.

Chapter 4

Implementation

Scientific computing involves several tasks, all of which are necessary to provide contributions to the understanding of phenomena in scientific problems. This work is concerned with the numerical solution of quadratic minimization problems involving non-linear constraints and non-local operators. The model problem (M) describes the behavior of thin-film micromagnetic devices and it serves as a prototype.

After analyzing the given equations and stating the well-posedness of the model problem, we provided a general strategy for the numerical solution of quadratic non-linear constrained problems with non-local norms. Finally, in the present chapter we dedicate ourselves to the not less challenging and important task of providing an efficient and easy accessible strategy for the implementation of the proposed method.

Nowadays, high-level programming and scripting languages such as `Matlab` and `python` are becoming more and more important even for the implementation of highly complex numerical algorithms. This increase of usage in the scientific community is driven by several causes: Firstly, the increase in computer power over the last decades has made high-level languages competitive for many applications. Secondly, languages such as `Matlab` decrease the development time when compared to low-level languages such as C. This is because `Matlab` provides a very large mathematical library and because high-level languages usually make the manipulation of simple data structures easy.

However, when it comes to mere power in terms of both, execution time and control over the algorithms, low-level languages such as C or C++ are still superior and probably will not be deprecated for quite some time in their field of excellence. There were mainly five reasons that led to the decision to stick to C++ as the tool of choice for the implementation of our algorithms. First, the author feels comfortable with the style of C++. From a scientific point of view this might not be a good reason, however from a practical point of view it is of high relevance since development times for codes are kept low. Second, good C++ compilers are available on almost all platforms. Third, we are convinced that object-oriented programming should be the paradigm of choice when it comes to tasks such as mesh administration. Fourth, C++ is naturally compatible with C, which makes the use of existing numerical libraries such as `HLib` comfortable. Finally, C++ provides a mighty standard library that is used extensively in our codes.

4.1 An object oriented implementation of meshes

We stress that our strategies and codes are not optimized with respect to system resources. Instead, we aim at providing a general approach that is easily extendable to 3D triangulations and meshes designed for other purposes. This, however, implies that the codes should be easy to understand and that the possibilities for the introduction of abstract classes or template arguments are clear from the data structure. Moreover, in most numerical simulations the mesh administration is neither the bottleneck with respect to memory consumption nor with respect to computation time.

4.1.1 Geometric base classes

Before introducing the actual base classes, we discuss in short the `Property` class that will be used for low-level data structures. The aim is to provide a natural interface to primitive members without violating the object oriented style of cascading data representation from data access.

Listing 4.1: The `Property` class

```

1 template <class prop_t, class obj_t,
2           const prop_t& (obj_t::*get)() const, void (obj_t::*set)(prop_t ref)>
3 class Property {
4     private:
5         obj_t& instance;
6     public:
7         Property(obj_t& refInstance) : instance(refInstance) {}
8         prop_t operator = (const prop_t& ref) {
9             (instance.*set)(ref);
10            return (instance.*get)();
11        }
12        operator prop_t const & () const {
13            return (instance.*get)();
14        }
15        bool operator ==(Property<prop_t, obj_t, get, set>& prop2) const {
16            prop_t myprop = instance.get();
17            prop_t hisprop = prop2();
18            return myprop == hisprop;
19        }
20 };

```

A class using a `Property` needs to have a private variable of type `prop_t` as well as a get-function of type `const prop_t&` and a set-function of type `void`. Besides access to get- and set-functions the `Property` also provides an overloaded equality operator.

Now, we introduce `Point`, `Edge`, and `Triangle` classes that represent the basic geometric data for our simulations.

Listing 4.2: The `Point` class

```

1 class Point {
2     private:
3         double m_x;
4         double m_y;
5     public:
6         Point() : m_x(0.), m_y(0.), x(*this), y(*this) {}
7         Point(const double refX, const double refY) :
8             m_x(refX), m_y(refY), x(*this), y(*this) {}
9         Point(const Point& point) : m_x(point.x), m_y(point.y),

```

```

10         x(*this), y(*this){}
11
12     ~Point() {}
13
14     const double& getX() const {return m_x;}
15     void setX(double refX) {m_x = refX;}
16     const double& getY() const {return m_y;}
17     void setY(double refY) {m_y = refY;}
18
19     Property<double, Point, &Point::getX, &Point::setX> x;
20     Property<double, Point, &Point::getY, &Point::setY> y;
21
22     const Point& operator=(const Point& point);
23     const Point& operator+=(const Point& point);
24     const Point& operator-=(const Point& point);
25     const Point& operator*=(double scalar);
26     const Point& operator/=(double scalar);
27 };
28
29 typedef Point Vector2D;
30
31 bool operator==(const Point& point1, const Point& point2);
32 bool operator!=(const Point& point1, const Point& point2);
33
34 bool operator<(const Point& point1, const Point& point2);
35 bool operator>(const Point& point1, const Point& point2);
36 bool operator<=(const Point& point1, const Point& point2);
37 bool operator>=(const Point& point1, const Point& point2);
38
39 const Point operator-(const Point& point);
40 const Point operator+(const Point& point1, const Point& point2);
41 const Point operator-(const Point& point1, const Point& point2);
42 const Point operator*(const Point& point, double scalar);
43 const Point operator/(const Point& point, double scalar);
44 const Point operator*(double scalar, const Point& point);
45
46 double operator*(const Vector2D& v1, const Vector2D& v2);
47 const Vector2D operator!(const Vector2D& vector);
48 double norm(const Vector2D& vector); //euclidian norm

```

The `Point` class basically consists of two coordinates that may be accessed as `Properties`. Besides the usual member functions such as constructors, the `Point` class also provides a large set of operators. Equality `==` and inequality `!=` are interpreted up to rounding error tolerance. The ordering operators `<`, `<=`, `>`, `>=` represent lexicographical ordering, i.e. `A<B` holds true if either `A.x<B.x` or `(A.x==B.x && A.y<B.y)`, where the identity here, again, is meant up to rounding errors.

Finally, since two-dimensional vectors and points in the Cartesian plane may be identified, we introduce a `typedef` `Vector2D` and provide the scalar product as well as computation of Euclidian norm and counter clockwise rotation via the operator `!`.

Listing 4.3: The Edge class

```

1 class Edge{
2     private:
3         Point m_A;
4         Point m_B;
5     public:
6         Edge(const Point& refA, const Point& refB) :
7             m_A(refA), m_B(refB), A(*this), B(*this) {}

```

```

8      Edge(const double AX, const double AY, const double BX, const double BY):
9          m_A(Point(AX, AY)), m_B(Point(BX, BY)), A(*this), B(*this){}
10     Edge(const double AX, const double AY, const Point& refB) :
11         m_A(Point(AX, AY)), m_B(refB), A(*this), B(*this){}
12     Edge(const Point& A, const double BX, const double BY):
13         m_A(A), m_B(Point(BX, BY)), A(*this), B(*this){}
14     Edge(const Edge& edge) : m_A(edge.A), m_B(edge.B), A(*this), B(*this){}
15
16     ~Edge(){}
17
18     const Point& getA() const {return m_A;}
19     void setA(Point refA) {m_A = refA;}
20     const Point& getB() const {return m_B;}
21     void setB(Point refB) {m_B = refB;}
22
23     Property<Point, Edge, &Edge::getA, &Edge::setA> A;
24     Property<Point, Edge, &Edge::getB, &Edge::setB> B;
25
26     const Edge& operator=(const Edge& edge);
27
28     double Length() const;
29 };
30
31 bool operator==(const Edge& edge1, const Edge& edge2);
32 bool operator!=(const Edge& edge1, const Edge& edge2);
33 bool operator<(const Edge& edge1, const Edge& edge2);
34 bool operator>(const Edge& edge1, const Edge& edge2);
35 bool operator<=(const Edge& edge1, const Edge& edge2);
36 bool operator>=(const Edge& edge1, const Edge& edge2);
37
38 Edge operator-(const Edge& edge);
39
40 Vector2D NormalVector(const Edge& edge);
41
42 bool AreParallel(const Edge& edge1, const Edge& edge2);
43 bool DoIntersect(const Edge& edge1, const Edge& edge2);

```

Edges consist of two **Point** objects that may be accessed as **Properties**. The **Length()** method returns the Euclidean length of the **Edge**. All operators, as in the case of the **Point** class, are interpreted up to rounding errors. The ordering operators again represent lexicographical ordering. All operators ignore the orientation of the **Edge**, i.e. **Edge(A,B) == Edge(B,A)** holds true. The auxiliary functions are quite self explanatory.

Finally, the **Triangle** class provides the data structure for the representation of a triangle in the plane. It consists of three **Points** and provides a get-method that returns its **Edges** that, however, are not stored explicitly. The equality operator, as in the case of **Edges**, is independent of the orientation and interpreted up to rounding errors.

Listing 4.4: The Triangle class

```

1  class Triangle{
2      private:
3          Point m_A;
4          Point m_B;
5          Point m_C;
6      public:
7          Triangle(const Point& refA, const Point& refB, const Point& refC) :
8              m_A(refA), m_B(refB), m_C(refC), A(*this), B(*this), C(*this){}
9          Triangle(const Triangle& triangle) :

```

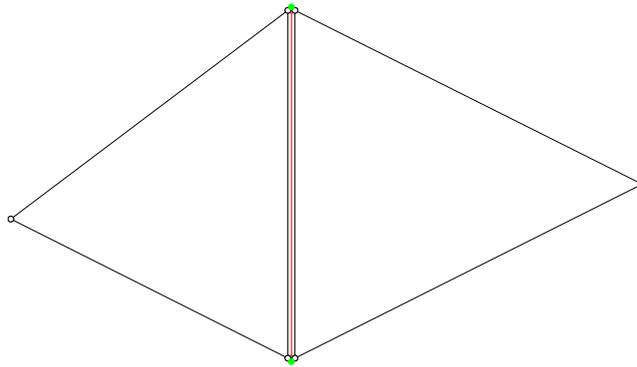


Figure 4.1: In a `Mesh` object, all triangles, edges and vertices are stored in a `list` container. Consider two triangles stored as `MeshTriangle` objects. If they share an edge, the geometric relation between them is provided with the help of containers of `MeshEdge*`, of which one is set to store the address of the global `MeshEdge` object (red). Also pointers to the global `MeshPoint` objects (green) are stored in the `MeshTriangle` objects.

```

10         m_A(triangle.A), m_B(triangle.B), m_C(triangle.C),
11         A(*this), B(*this), C(*this) {}
12
13     ~Triangle() {}
14
15     const Point& getA() const {return m_A;}
16     void setA(Point refA) {m_A = refA;}
17     const Point& getB() const {return m_B;}
18     void setB(Point refB) {m_B = refB;}
19     const Point& getC() const {return m_C;}
20     void setC(Point refC) {m_C = refC;}
21
22     Property<Point, Triangle, &Triangle::getA, &Triangle::setA> A;
23     Property<Point, Triangle, &Triangle::getB, &Triangle::setB> B;
24     Property<Point, Triangle, &Triangle::getC, &Triangle::setC> C;
25
26     const Triangle& operator=(const Triangle& triangle);
27     void operator==(const Point& point);
28
29     const vector<Edge> Edges() const;
30     double Area() const;
31 };
32
33 bool operator==(const Triangle& triangle1, const Triangle& triangle2);

```

4.1.2 The Mesh class

In a mesh, the geometric objects need to be related to each other somehow. To serve this purpose efficiently, we introduce derived classes `MeshPoint`, `MeshEdge`, and `MeshTriangle`. They simply extend the base classes, by adding containers of pointers to other related geometric objects. Additionally, each of these classes adopts the `Mesh` class as a friend, allowing private data access. The mutable data `m_copy` is used in the adaptive refinement of the mesh. It allows for efficient recalculation of the geometric relations.

Listing 4.5: The `MeshPoint` `MeshEdge` and `MeshTriangle` classes

```

1  class MeshPoint : public Point
2  {
3      private:
4          list<MeshTriangle*> triangles;
5          list<MeshEdge*> edges;
6
7          friend class Mesh;
8          mutable MeshPoint* m_copy;
9      public:
10         MeshPoint();
11         MeshPoint(const Point& p);
12         MeshPoint(const MeshPoint& p);
13
14         const MeshPoint& operator=(const MeshPoint& point);
15
16         list<MeshTriangle*> MeshTriangles() const{return triangles;}
17         list<MeshEdge*> MeshEdges() const{return edges;}
18         void setEdges(list<MeshEdge*> e){edges = e;}
19         void setTriangles(list<MeshTriangle*> t){triangles = t;}
20 };
21
22 class MeshEdge : public Edge
23 {
24     private:
25         bool marked;
26         list<MeshTriangle*> triangles;
27         list<MeshPoint*> points;
28
29         friend class Mesh;
30         friend class POSpace;
31         friend class RTOSpace;
32         mutable MeshEdge* m_copy;
33
34         list<MeshEdge*> m_sons;
35     public:
36         MeshEdge(const Edge& e);
37         MeshEdge(const MeshEdge& e);
38
39         const MeshEdge& operator=(const MeshEdge& edge);
40
41         void Mark(){marked = true;}
42         void Unmark(){marked = false;}
43         bool IsMarked() const{return marked;}
44         list<MeshTriangle*> MeshTriangles() const{return triangles;}
45         list<MeshPoint*> MeshPoints() const{return points;}
46         void setPoints(list<MeshPoint*> p){points = p;}
47         void setTriangles(list<MeshTriangle*> t){triangles = t;}
48 };
49
50 class MeshTriangle : public Triangle
51 {
52     private:
53         list<MeshPoint*> points;
54         list<MeshEdge*> edges;
55
56         friend class Mesh;
57         mutable MeshTriangle* m_copy;
58     public:
59         MeshTriangle(const Triangle& t);
60         MeshTriangle(const MeshTriangle& t);
61

```

```

62     const MeshTriangle& operator=(const MeshTriangle& triangle);
63
64     list<MeshEdge*> MeshEdges() const{return edges;}
65     list<MeshPoint*> MeshPoints() const{return points;}
66     void setPoints(list<MeshPoint*> p){points = p;}
67     void setEdges(list<MeshEdge*> e){edges = e;}
68
69     double MeshWidth() const;
70 };

```

A `Mesh` is in general nothing but a set of `MeshTriangle` objects. However in finite element applications it proves useful to have explicit access to vertices and edges as well. This is because degrees of freedom are often associated, e.g., with vertices instead of elements. The private data of the mesh, therefore, has three containers. One for global `MeshPoint` objects, one for global `MeshEdge` objects, and one for the `MeshTriangle` objects. With global we mean here that actually each `MeshTriangle` object has its own set of private `Points`. In particular `MeshTriangle` objects that represent neighboring triangles might share a point in the plane, cf. Figure 4.1. In this case the point is stored twice in the local data of the `MeshTriangle` objects, and additionally once as global vertex of the mesh, i.e. a `MeshPoint` object. As the container class for storing the global data we chose the `std::list<T>`, which is a double linked list. Unfortunately, the C++ standard does not guarantee that the elements of a `list` container may not change their location in memory. However, the standard does guarantee constant time for inserting elements and there is no actual reason for elements to really change their location in memory. In all of our tests, we observed that pointers to elements of a `list` stay valid even for large and dynamically growing container, at least when compiled with the popular `g++` on various Linux platforms.

In order to provide the geometric information as to which global edges correspond to which global vertices and triangles, we implement interface functions such as the `TrianglesOfVertex` method. Note that the access to the global geometric objects is granted by providing iterators to the first and the last element of the container.

Listing 4.6: The Mesh class

```

1  class Mesh
2  {
3      protected:
4          list<MeshPoint> vertices;
5          list<MeshTriangle> triangles;
6          list<MeshEdge> edges;
7
8          void deep_copy(const Mesh&);
9      public:
10         Mesh(){};
11         Mesh(string coordinatesFilename, string elementsFilename);
12         Mesh(const Mesh& mesh);
13
14         const Mesh& operator=(const Mesh&);
15
16         list<MeshPoint>::iterator VerticesBegin()
17             {return vertices.begin();};
18         list<MeshPoint>::iterator VerticesEnd()
19             {return vertices.end();};
20         list<MeshEdge>::iterator EdgesBegin()
21             {return edges.begin();};
22         list<MeshEdge>::iterator EdgesEnd()
23             {return edges.end();};
24         list<MeshTriangle>::iterator TrianglesBegin()

```

```

25     {return triangles.begin();};
26 list<MeshTriangle>::iterator TrianglesEnd()
27     {return triangles.end();};
28
29 list<MeshPoint>::const_iterator VerticesBegin() const
30     {return vertices.begin();};
31 list<MeshPoint>::const_iterator VerticesEnd() const
32     {return vertices.end();};
33 list<MeshEdge>::const_iterator EdgesBegin() const
34     {return edges.begin();};
35 list<MeshEdge>::const_iterator EdgesEnd() const
36     {return edges.end();};
37 list<MeshTriangle>::const_iterator TrianglesBegin() const
38     {return triangles.begin();};
39 list<MeshTriangle>::const_iterator TrianglesEnd() const
40     {return triangles.end();};
41
42 long NrOfTriangles() const {return triangles.size();};
43 long NrOfEdges() const {return edges.size();};
44 long NrOfVertices() const {return vertices.size();};
45
46 list<const MeshTriangle*> TrianglesOfVertex(const MeshPoint&) const;
47 list<const MeshTriangle*> TrianglesOfEdge(const MeshEdge&) const;
48 list<const MeshEdge*> EdgesOfVertex(const MeshPoint&) const;
49 list<const MeshEdge*> EdgesOfTriangle(const MeshTriangle&) const;
50 list<const MeshPoint*> VerticesOfTriangle(const MeshTriangle&) const;
51 list<const MeshPoint*> VerticesOfTEdge(const MeshEdge&) const;
52
53 void Mark(MeshEdge&);
54 void Mark(MeshPoint&);
55 void Mark(MeshTriangle&);
56
57 void Refine(bool uniform = false);
58
59 void ExportToMatlabFormat(string coordinatesFilename,
60                          string edgesFilename,
61                          string elementsFilename) const;
62
63 void PlotPS(string filename);
64 };

```

Since the plots for visualization of our numerical results are done in **Matlab**, we provide an interface to the usual matrix and index based mesh-format in **Matlab**. Here, the vertices of the mesh are stored in a $V \times 2$ -matrix of their coordinates, where V denotes the number of vertices. The triangles are stored in a $T \times 3$ matrix, where T denotes the number of triangles. Each row of the matrix consist of three integer values denoting the vertices (indexed starting with 1) specified in the vertices-matrix. This data is sufficient to describe the geometry of a mesh. Therefore the constructor expects two ASCII-files with data in this format. For a simple mesh as depicted in Figure 4.2, the corresponding *.dat files read

```

vertices.dat
-0.5 -0.5
0.5 -0.5
0.5 0.5
-0.5 0.5
0.0 0.0

```

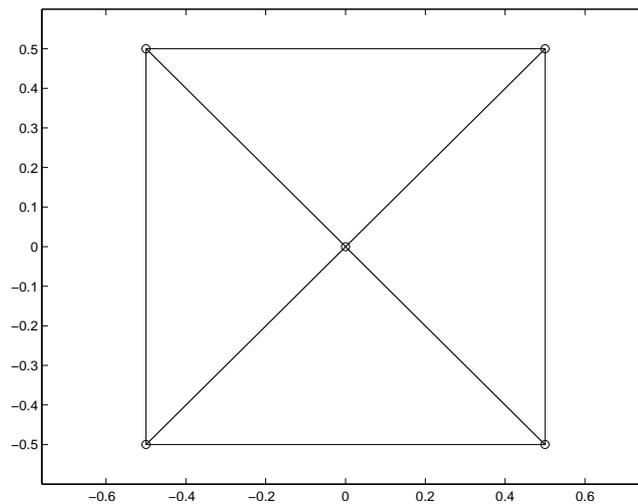


Figure 4.2: A simple mesh with 5 vertices, 4 triangles, and 8 edges.

and

```
triangles.dat
1 2 5
2 3 5
3 4 5
4 1 5
```

The basis functions of the Raviart-Thomas space $RT^0(\mathcal{T})$ are linked geometrically to the interior edges of the underlying mesh. We therefore provide moreover an $E \times 4$ -matrix with E the number of edges. Each row contains the following information: The first two entries are the indices of the vertices that are the endpoints of the edge. The third integer is the index of the triangle T_+ . The fourth integer is either zero if the edge is a boundary edge or the index of the triangle T_- adjacent to the edge. For the simple mesh of Figure 4.2 the corresponding *.dat file could read

```
edges.dat
1 2 1 0
2 5 1 2
5 1 4 1
2 3 2 0
3 5 2 3
3 4 3 0
4 5 3 4
4 1 4 0
```

We stress that the edge-matrix is not uniquely determined and fixes the orientation for the basis functions of $RT^0(\mathcal{T})$ by setting T_+ and T_- for each interior edge. Since this orientation is arbitrary, the `ExportToMatlabFormat` routine also stores an `edge.dat` file.

The adaptive (or uniform) refinement is performed on an edge based logic. First, the user may `Mark` edge, triangles, or vertices. In principle we always mark `MeshEdge` objects for refinement. The `Mark(MeshEdge&)` method, recursively ensures that for each triangle where an edge is marked, also the

longest edge of the triangle is marked. The `Mark(MeshPoint&)` and `Mark(MeshTriangle&)` methods simply call the `Mark(MeshEdge&)` version for all edges corresponding to the point or the triangle. Finally the `Refine()` method performs the adaptive mesh-refinement following the red-green-blue strategy as described in Section 3.4.2.

4.2 Finite element spaces

A finite element space is a set of discrete functions. Usually the basis functions are associated with some geometric entity of a mesh. In this spirit, our implementation first provides classes to define basis functions. Then the finite element space is derived from the mesh class and extended by operations with basis functions. Since we only focus on lowest-order methods, we only need an implementation for the discrete spaces $P^0(\mathcal{T})$ and $RT^0(\mathcal{T})$. In principle the concept of basis functions is extendable and can be generalized with the help of templates easily. However, since we aim at a specific application, we only present concrete implementations for our experiments.

4.2.1 Basis functions

A basis function expects upon creation a reference to a `Mesh` object as well as an index and some information on its geometric link to the mesh. Other than providing the support of itself and its index, the basis function and its derivative may be evaluated. Both evaluation methods expect a `Point` object that represents a point on the reference triangle $T_{\text{ref}} = \text{conv}\{(0,0), (1,0), (0,1)\}$ as well as the reference to an existing `MeshTriangle` object. Note that in contrast to the geometric elements of a `Mesh`, the basis functions are indexed (starting with index 0 as usual in C++).

Listing 4.7: P0 basis function

```

1 class P0Element {
2     private:
3         long m_index;
4         const MeshTriangle* m_triangle;
5
6     public:
7         P0Element(long index, const MeshTriangle& geometry, const Mesh& mesh);
8         ~P0Element(){};
9         const P0Element& operator= (const P0Element& element);
10
11        long Index() const {return m_index;}
12        void Index(long index) {m_index = index;}
13
14        list<const MeshTriangle*> Support() const;
15        const MeshTriangle& GeometricHook() const {return *m_triangle;}
16
17        double Evaluate(const Point& refP, const MeshTriangle& triangle) const;
18        Vector2D EvaluateDiff(const Point& refP, const MeshTriangle& triangle)
19                               const {return Vector2D(0.,0.);}
20 };

```

Listing 4.8: RT0 basis function

```

1 class RT0Element {
2     private:
3         long m_index;
4         const MeshEdge* m_edge;

```

```

5     const MeshTriangle* m_TPlus;
6     const MeshTriangle* m_TMinus;
7
8     public:
9         RT0Element(long index, const MeshEdge& geometry, const Mesh& mesh);
10        ~RT0Element(){};
11        const RT0Element& operator= (const RT0Element& element);
12
13        long Index() const {return m_index;}
14        void Index(long index) {m_index = index;}
15
16        list<const MeshTriangle*> Support() const;
17        const MeshEdge& GeometricHook() const {return *m_edge;}
18
19        Vector2D Evaluate(const Point& refP, const MeshTriangle& triangle)
20                                const;
21        double EvaluateDiff(const Point& refP, const MeshTriangle& triangle)
22                                const;
23
24 };

```

4.2.2 Finite element spaces

In contrast to the private data of the `Mesh` class, the basis-functions of a finite element space are stored in a `std::vector<T>` container. This has the advantage that basis functions can be accessed by their index in constant time. Moreover, sometimes it is useful to access basis functions by their geometric link. To that end the parentheses-operator provides read access. Finally, given some C-vector of coefficients, i.e. a `double*`, the corresponding finite element function and its derivative may be evaluated. In principle one could export the underlying mesh and the coefficient vector for plotting in `Matlab`. However, evaluation of the finite element function is a lot faster in C++ than in a `Matlab` environment. Therefore the `ExportFunctionToMatlab` method also exports the function and its derivative evaluated at the center of mass of each triangle. Note that this export routine is slow, due to the massive number of evaluations and the file manipulations necessary for large spaces.

Listing 4.9: Finite element space P^0

```

1 class P0Space : public Mesh {
2     private:
3         vector<P0Element> m_basis;
4         static bool m_compare(const pair<const MeshTriangle*, long>& left,
5                               const pair<const MeshTriangle*, long>& right)
6                               {return left.first < right.first;}
7         vector<pair<const MeshTriangle*, long>> geometry2basis;
8
9     public:
10        P0Space(){};
11        P0Space(string coordinatesFilename, string elementsFilename);
12        P0Space(Mesh& mesh);
13
14        long NumberOfElements() const {return m_basis.size();};
15
16        const P0Element& operator [] (long index) const {return m_basis[index];};
17        const P0Element& operator () (const MeshTriangle& geometricHook) const;
18
19        double Evaluate(const Point& refP, const MeshTriangle& T, double* x)
20        Vector2D EvaluateDiff(const Point& refP, const MeshTriangle& T, double*x)
21                                const;

```

```

22
23     RefinementData* Refine(bool uniform = false);
24
25     void ExportFunctionToMatlab( string coordinatesFilename ,
26                               string edgesFilename ,
27                               string elementsFilename ,
28                               string P0CoeffFilename ,
29                               string P0EvalFilename ,
30                               string P0EvalDiffFilename, double* x);
31
32     void buildBasisFunctions();
33 };

```

Listing 4.10: Finite element space RT^0

```

1 class RT0Space : public Mesh {
2     private:
3         vector<RT0Element> m_basis;
4
5         vector<P0Element> m_diffbasis;
6
7         static bool m_compare(const pair<const MeshEdge*, long>& lhs ,
8                             const pair<const MeshEdge*, long>& rhs)
9             {return lhs.first < rhs.first;}
10
11        static bool m_diffcompare(const pair<const MeshTriangle*, long>& lhs ,
12                                const pair<const MeshTriangle*, long>& rhs)
13                {return lhs.first < rhs.first;}
14        vector<pair<const MeshEdge*, long>> geometry2basis;
15        vector<pair<const MeshTriangle*, long>> geometry2diffbasis;
16
17    public:
18        RT0Space(){};
19        RT0Space(string coordinatesFilename , string elementsFilename);
20        RT0Space(const Mesh& mesh);
21        RT0Space(const RT0Space& space);
22
23        long NumberOfElements() const {return m_basis.size();};
24
25        const RT0Element& operator [](long index) const{return m_basis[index];};
26        const RT0Element& operator ()(const MeshEdge& geometricHook) const;
27        const P0Element& operator ()(const MeshTriangle& geometricDiffHook)
28                                const;
29
30        Vector2D Evaluate(const Point& refP , const MeshTriangle& T, double* mH)
31                                const;
32        double EvaluateDiff(const Point& refP , const MeshTriangle& T, double* mH)
33                                const;
34
35        RefinementData* Refine(bool uniform = false);
36
37        void ExportFunctionToMatlab( string coordinatesFilename ,
38                                  string edgesFilename ,
39                                  string elementsFilename ,
40                                  string mHCoeffFilename ,
41                                  string mHEvalFilename ,
42                                  string mHEvalDiffFilename ,
43                                  double* mH);
44
45        void buildBasisFunctions();
46 };

```

The basis functions of the space need to be evaluated often. In many cases, however, we iterate in our algorithms naturally over the triangles of the mesh. This is, e.g., the case in the computation of error estimators. With the method `EdgesOfTriangle`, we first seek the corresponding global `MeshEdges`. Then the `operator()` provides access to the basis functions associated with these edges. We therefore need fast access ($\mathcal{O}(\log N)$) via the geometric hook. The naive approach to implementing this access would be the use of a `std::map<MeshEdge*, long>` container associating each `MeshEdge` with the index of a basis function. However, we found that this solution is not very efficient. This is because the insertion of pairs into the `map` container is very expensive. This effects that the `buildBasisFunctions()` routine shows quadratic runtime behavior.

We avoided this problem by simply using a `std::vector<pair<MeshEdge*, long>>` instead. In `buildBasisFunctions`, we first fill the container before finally sorting it. The `operator()`, then, uses the `lower_bound` function from the `<algorithm>` library to find the correct pair. The `lower_bound` function implements a bisection search algorithm, such that in the end the access to basis function has an overall cost of $\mathcal{O}(\log N)$ as desired.

Listing 4.11: Efficient implementation of link between basis functions and geometry

```

1 void RT0Space::buildBasisFunctions(){
2     long index = 0;
3
4     m_basis.clear();
5     m_basis.reserve(NrOfEdges());
6     geometry2basis.clear();
7     geometry2basis.reserve(NrOfEdges());
8
9     for(list<MeshEdge>::const_iterator it = EdgesBegin(); it != EdgesEnd();
10         ++it){
11         if (TrianglesOfEdge(*it).size() == 2) {//this is an interior edge
12             m_basis.push_back(RT0Element(index, *it, *this));
13             geometry2basis.push_back(pair<const
14                                     MeshEdge*, long>(&(*it), index++));
15         }
16     }
17     sort(geometry2basis.begin(), geometry2basis.end(), m_compare);
18
19     index = 0;
20     m_diffbasis.clear();
21     m_diffbasis.reserve(NrOfTriangles());
22     geometry2diffbasis.clear();
23     geometry2diffbasis.reserve(NrOfTriangles());
24     for(list<MeshTriangle>::const_iterator it = TrianglesBegin(); it !=
25         TrianglesEnd(); ++it){
26         m_diffbasis.push_back(P0Element(index, *it, *this));
27         geometry2diffbasis.push_back(pair<const MeshTriangle*,
28                                     long>(&(*it), index++));
29     }
30     sort(geometry2diffbasis.begin(), geometry2diffbasis.end(), m_diffcompare);
31 }
32
33 const RT0Element& RT0Space::operator()(const MeshEdge& geometricHook) const{
34     vector<pair<const MeshEdge*, long>>::const_iterator findIt =
35         lower_bound(geometry2basis.begin(), geometry2basis.end(),
36                     pair<const MeshEdge*, long>(&geometricHook, 0), m_compare);
37
38     const RT0Element* element = 0;
39     if (findIt != geometry2basis.end())
40         element = &m_basis[findIt->second];
41

```

```

42     return *element ;
43 }

```

For the implementation of our $h - h/2$ based error estimators, we need to provide some information on how to map functions over a coarser mesh onto some refinement of the mesh, and vice versa. To that end, the refinement procedure of a finite element space provides a (sparse) matrix that represents the prolongation operator. Let $\mathbf{m}_h \in RT^0(\mathcal{T}_1)$ and assume \mathcal{T}_2 is some refinement of \mathcal{T}_1 . Then it holds that $\mathbf{m}_h \in RT^0(\mathcal{T}_2)$ as well. Let \mathbf{x} be the coefficient vector of \mathbf{m}_h with respect to the coarse mesh \mathcal{T}_1 . Then, the prolongation operator \mathbf{P} is defined by

$$\mathbf{m}_h = \sum_{E \in \mathcal{E}_{\omega,2}} (\mathbf{P}\mathbf{x})_E \psi_E$$

with $\mathcal{E}_{\omega,2}$ the set of interior edges of the finer mesh \mathcal{T}_2 and ψ_E the associated basis function as defined in (3.1). This means that the prolongation matrix maps the coefficient vector of a function over the coarse mesh to the vector of coefficients that represent the same function over the finer mesh.

Listing 4.12: Data sturcture for the prolongation operators

```

1 typedef struct _RefinementData {
2     sparsematrix* Prolongation ;
3     sparsematrix* ProlongationDiff ;
4 } RefinementData ;

```

Recall the definition of the basis function ψ_E associated with some interior edge $E \in \mathcal{E}_{\omega}$,

$$\psi_E = \begin{cases} \pm \frac{|E|}{2|T_{\pm}|} (x - P_{\pm}), & \text{for } x \in T_{\pm} \\ 0, & \text{elsewhere.} \end{cases} \quad (4.1)$$

The basis function is normalized in the sense that $\psi_E \cdot \mathbf{n} = 1$ with \mathbf{n} the unit normal vector pointing to T_+ . Hence, the coefficient determines the normal flux through the edge. The prolongation matrix may be thus computed in the following way:

- Each `MeshEdge` E_i that is refined into two new `MeshEdge` objects E_{i_1} and E_{i_2} effects two entries with value 1 that map the coefficient \mathbf{x}_i to the new coefficients $\mathbf{x}_{i_1} = \mathbf{x}_{i_2} = \mathbf{x}_i$.
- Let E_j be an interior edge that is newly generated by the mesh refinement. Then, there is a uniquely determined triangle T of the original mesh, such that $E_j \subseteq T$. For each interior edge E of T , we generate a matrix entry by computing $\psi_E \cdot \mathbf{n}$ with \mathbf{n} the normal vector at the new edge E_j . The only challenging practical aspect is to ensure the correct orientation of \mathbf{n} .

4.3 A damped Newton algorithm

After discretizing the penalized model problem, we need to solve the minimization problem (M_h^ε) , i.e. equivalently the Euler-Lagrange equation (3.38). We apply Newton's algorithm to find the root of the derivative $De_\varepsilon(\mathbf{m}_h^\varepsilon)$. Note that this derivative of the penalized energy functional is not continuously differentiable due to the penalty energy. Hence no classical result on convergence of Newton's algorithm may be applied.

Let \mathcal{T} denote some regular triangulation with $\#\mathcal{E}_{\omega} =: N_D$ interior edges and recall the space of Raviart-Thomas functions $RT^0(\mathcal{T})$ defined in Chapter 3.1. Given some coefficient vector $\mathbf{x} \in \mathbb{R}^{N_D}$ the

discrete Euler-Lagrange equation reads

$$\left(\nabla \cdot \sum_{i=1}^{N_D} \mathbf{x}_i \psi_i, \nabla \cdot \mathbf{w}_h\right)_V + q\left(\sum_{i=1}^{N_D} \mathbf{x}_i \psi_{i,2}, \mathbf{w}_{h,2}\right)_{L^2} - (f, \mathbf{w}_h)_{L^2} + \frac{1}{\varepsilon} (\lambda_h^\varepsilon \sum_{i=1}^{N_D} \mathbf{x}_i \psi_i, \mathbf{w}_h) = 0$$

for all $\mathbf{w}_h \in RT^0(\mathcal{T})$. Obviously this equations holds for all $\mathbf{w}_h \in RT^0(\mathcal{T})$ if and only if it holds for all basis functions $(\psi_i)_{i=1}^{N_D}$ of $RT^0(\mathcal{T})$. Hence we seek to find the root of the discrete function $F : \mathbb{R}^{N_D} \rightarrow \mathbb{R}^{N_D}$ defined through

$$F(\mathbf{x})_j = (\nabla \cdot \mathbf{m}_h^\varepsilon(\mathbf{x}), \nabla \cdot \psi_j)_V + q(\mathbf{m}_{h,2}^\varepsilon(\mathbf{x}), \psi_{j,2})_{L^2} + \frac{1}{\varepsilon} (\lambda_h^\varepsilon \mathbf{m}_h^\varepsilon(\mathbf{x}), \psi_j)_{L^2} - (f, \psi_j)_{L^2}. \quad (4.2)$$

The notation $\mathbf{m}_h^\varepsilon(\mathbf{x})$ indicates that the discrete magnetization depends on the given coefficient vector.

Newton's algorithm:

Let $\mathbf{x}^{(0)} \in \mathbb{R}^{N_D}$ denote some initial value and set $\ell = 0$.

- (i) Evaluate $F(\mathbf{x}^{(\ell)})$ and compute the derivative $DF^{(\ell)}$.
- (ii) Compute the Newton update $\delta \in \mathbb{R}^{N_D}$ by solving the linear system $DF^{(\ell)}\delta = -F(\mathbf{x}^{(\ell)})$.
- (iii) Define $\mathbf{x}^{(\ell+1)} := \mathbf{x}^{(\ell)} + \delta$
- (iv) Either stop or $\ell \mapsto \ell + 1$ and goto (i)

Output: An approximation \mathbf{x} to some root of the function F .

One crucial step is the computation of the derivative of the function F . In equation (4.2), the derivative of the first two scalar products can be computed easily. The fourth contribution vanishes, since it is just a constant. The third term, however, is a bit more involved.

We aim at computing the derivative of $(\lambda_h^\varepsilon \mathbf{m}_h^\varepsilon(\mathbf{x}), \psi_j)_{L^2}$. Recall that this is not a differentiable function. The derivative of λ_h^ε is not defined classically at the points where $|\mathbf{m}_h^\varepsilon(\mathbf{x})| = 1$. However, since the scalar product is computed by use of numerical quadrature, this precise condition is not expected to be encountered numerically at any quadrature point. We therefore proceed to compute the derivative for $|\mathbf{m}_h^\varepsilon(\mathbf{x})| < 1$ (which vanishes trivially) and for $|\mathbf{m}_h^\varepsilon(\mathbf{x})| > 1$.

Lemma 4.1. *The derivative $\mathbf{DF}_{\text{NL}} \in \mathbb{R}^{N_D \times N_D}$ of the function $F_{\text{NL}} : \mathbb{R}^{N_D} \rightarrow \mathbb{R}^{N_D}$ defined through*

$$\mathbf{x} \mapsto (g(\mathbf{x}), \psi_j)_{L^2} \quad \text{with} \quad g(\mathbf{x}) := \frac{(|\mathbf{m}_h^\varepsilon(\mathbf{x})| - 1)_+}{|\mathbf{m}_h^\varepsilon(\mathbf{x})|} \mathbf{m}_h^\varepsilon(\mathbf{x})$$

reads

$$(\mathbf{DF}_{\text{NL}})(\mathbf{x})_{i,j} = \left(\frac{\partial g(\mathbf{x})}{\partial \mathbf{x}_i}, \psi_j\right)_{L^2}.$$

At some point $x \in \omega$ the derivative of g either reads $\frac{\partial g(\mathbf{x})}{\partial \mathbf{x}_i}(x) = 0$ for $|\mathbf{m}_h^\varepsilon(\mathbf{x})(x)| < 1$ or

$$\frac{\partial g(\mathbf{x})}{\partial \mathbf{x}_i}(x) = \psi_i - \begin{pmatrix} \frac{1}{|\mathbf{m}_h^\varepsilon|} - \frac{\mathbf{m}_{h,1}^\varepsilon}{|\mathbf{m}_h^\varepsilon|^3} & -\frac{\mathbf{m}_{h,1}^\varepsilon \mathbf{m}_{h,2}^\varepsilon}{|\mathbf{m}_h^\varepsilon|^3} \\ -\frac{\mathbf{m}_{h,1}^\varepsilon \mathbf{m}_{h,2}^\varepsilon}{|\mathbf{m}_h^\varepsilon|^3} & \frac{1}{|\mathbf{m}_h^\varepsilon|} - \frac{\mathbf{m}_{h,2}^\varepsilon}{|\mathbf{m}_h^\varepsilon|^3} \end{pmatrix} \cdot \psi_i.$$

if $|\mathbf{m}_h^\varepsilon(\mathbf{x})(x)| > 1$.

Proof. The first statement $(DF_{NL})(\mathbf{x})_{i,j} = (\frac{\partial g(\mathbf{x})}{\partial \mathbf{x}_i}, \psi_j)_{L^2}$ is obvious. Also since $g(\mathbf{x})(x) = 0$ for all $x \in \omega$ with $|\mathbf{m}_h^\varepsilon(\mathbf{x})(x)| < 1$ and $\mathbf{m}_h^\varepsilon \in \mathcal{C}^\infty(\mathcal{T})$, i.e. piecewise smooth, the derivative of g vanishes at all such points. (Note that the set $\omega_{<} := \{x \in \omega \mid |\mathbf{m}_h^\varepsilon(\mathbf{x})(x)| < 1\}$ is open).

We therefore aim at computing the derivative of g at points $x \in \omega$ with $|\mathbf{m}_h^\varepsilon(\mathbf{x})(x)| > 1$. For each such point there is a neighborhood where g simply adopts the form

$$g(\mathbf{x}) = \frac{|\mathbf{m}_h^\varepsilon(\mathbf{x})| - 1}{|\mathbf{m}_h^\varepsilon(\mathbf{x})|} \mathbf{m}_h^\varepsilon(\mathbf{x}).$$

Hence the derivative reads

$$\frac{\partial g(\mathbf{x})}{\partial \mathbf{x}_i}(x) = \frac{\partial}{\partial \mathbf{x}_i} \left(\mathbf{m}_h^\varepsilon(\mathbf{x}) - \frac{\mathbf{m}_h^\varepsilon(\mathbf{x})}{|\mathbf{m}_h^\varepsilon(\mathbf{x})|} \right).$$

Since $\frac{\partial \mathbf{m}_h^\varepsilon(\mathbf{x})}{\partial \mathbf{x}_i} = \psi_i$ and from the chain rule $\frac{\partial}{\partial \mathbf{x}_i} = \frac{\partial}{\partial \mathbf{m}_h^\varepsilon} \frac{\partial \mathbf{m}_h^\varepsilon}{\partial \mathbf{x}_i}$ we obtain

$$\frac{\partial g(\mathbf{x})}{\partial \mathbf{x}_i}(x) = \psi_i - \frac{\partial}{\partial \mathbf{m}_h^\varepsilon} \frac{\mathbf{m}_h^\varepsilon(\mathbf{x})}{|\mathbf{m}_h^\varepsilon(\mathbf{x})|} \cdot \psi_i.$$

The derivative $\frac{\partial}{\partial \mathbf{m}_h^\varepsilon} \frac{\mathbf{m}_h^\varepsilon}{|\mathbf{m}_h^\varepsilon|}$ can be computed straight forwardly. First we define

$$f := \frac{\mathbf{m}_h^\varepsilon}{|\mathbf{m}_h^\varepsilon|} = \begin{pmatrix} \mathbf{m}_{h,1}^\varepsilon ((\mathbf{m}_{h,1}^\varepsilon)^2 + (\mathbf{m}_{h,2}^\varepsilon)^2)^{-1/2} \\ \mathbf{m}_{h,2}^\varepsilon ((\mathbf{m}_{h,1}^\varepsilon)^2 + (\mathbf{m}_{h,2}^\varepsilon)^2)^{-1/2} \end{pmatrix}$$

Then all derivatives may be computed to be

$$\begin{aligned} \frac{\partial f_1}{\partial \mathbf{m}_{h,1}^\varepsilon} &= \frac{1}{|\mathbf{m}_h^\varepsilon|} - \frac{\mathbf{m}_{h,1}^\varepsilon}{|\mathbf{m}_h^\varepsilon|^3} \\ \frac{\partial f_1}{\partial \mathbf{m}_{h,2}^\varepsilon} &= \frac{\partial f_2}{\partial \mathbf{m}_{h,1}^\varepsilon} = -\frac{\mathbf{m}_{h,1}^\varepsilon \mathbf{m}_{h,2}^\varepsilon}{|\mathbf{m}_h^\varepsilon|^3} \\ \frac{\partial f_2}{\partial \mathbf{m}_{h,2}^\varepsilon} &= \frac{1}{|\mathbf{m}_h^\varepsilon|} - \frac{\mathbf{m}_{h,2}^\varepsilon}{|\mathbf{m}_h^\varepsilon|^3}. \end{aligned}$$

Collecting all terms finally concludes the proof. ■

Remark. The derivative of $g(\mathbf{x})$ is not defined at points where $|\mathbf{m}(x)| = 1$. Numerically, we do not expect to encounter this situation. In our implementation, points where $|\mathbf{m}(x)| = 1$ are treated in the same way as points where $|\mathbf{m}(x)| < 1$. □

In our simulation runs, we found that the Newton algorithm did not converge in all cases. It is well known, see e.g. [SB02], that the Newton algorithm is only guaranteed to converge for sufficiently smooth functions if the initial value chosen is sufficiently close to some root. For smooth function the following modified algorithm, often referred to as relaxed Newton algorithm or damped Newton method, can be proven to converge globally. However this is at the cost of decreased order of convergence.

Damped Newton algorithm:

Let $\mathbf{x}^{(0)} \in \mathbb{R}^{N_D}$ denote some initial value and set $\ell = 0$.

- (i) Evaluate $F(\mathbf{x}^{(\ell)})$ and compute the derivative $DF^{(\ell)}$.

Point	x	y	Weight
0	0.1666666666666667	0.1666666666666667	0.3333333333333333
1	0.1666666666666667	0.6666666666666667	0.3333333333333333
2	0.6666666666666667	0.1666666666666667	0.3333333333333333

Table 4.1: A 3 point Gauss quadrature rule on the reference triangle $T_{\text{ref}} = \text{conv}\{(0,0), (1,0), (0,1)\}$.

- (ii) Compute the search direction $\delta \in \mathbb{R}^{N_D}$ by solving the linear system $DF^{(\ell)}\delta = -F(\mathbf{x}^{(\ell)})$.
- (iii) Find minimal k such that $|F(\mathbf{x}^{(\ell)} + 0.5^k\delta)| < |F(\mathbf{x}^{(\ell)})|$
- (iv) Define $\mathbf{x}^{(\ell+1)} := \mathbf{x}^{(\ell)} + 0.5^k\delta$
- (v) Either stop or $\ell \mapsto \ell + 1$ and goto (i)

Output: An approximation \mathbf{x} to some root of the function F .

Note the modification in step (iii) to ensure a reduction of the residual in each step. As a stopping criterion we simply check $|F(\mathbf{x})| < 1e - 8$ which seems to be sufficient in our simulations. More sophisticated stopping criteria that, e.g., also deal with the possibility of not being able to find a root up to rounding errors could be chosen. This, however, was, as mentioned before, not necessary in any of our simulation runs. That is why we stick to the simpler criterion.

Let \mathcal{T} be some given mesh with triangles T_1, \dots, T_{N_T} and $N_D = \#\mathcal{E}_\omega$ interior edges. Let χ_{T_j} denote the characteristic function of a triangle T_j . Note that $\{\chi_{T_1}, \dots, \chi_{T_{N_T}}\}$ is a basis of the space $\mathcal{P}^0(\mathcal{T})$ of piecewise constant functions over the mesh. Then, we define the system matrices

$$\begin{aligned}
\mathbf{V} &\in \mathbb{R}^{N_T \times N_T} & \mathbf{V}_{i,j} &:= (\chi_{T_j}, V\chi_{T_i})_{L^2}, \\
\mathbf{Q} &\in \mathbb{R}^{N_D \times N_D} & \mathbf{Q}_{i,j} &:= (\psi_{j,2}, \psi_{i,2})_{L^2}, \\
\mathbf{D} &\in \mathbb{R}^{N_T \times N_D} & \mathbf{D}_{i,j} &:= (\nabla \cdot \psi_j)|_{T_i}.
\end{aligned} \tag{4.3}$$

We define the vectors $\mathbf{b}_j := (\mathbf{f}, \psi_j)_{L^2}$ and $\mathbf{a}_j(\mathbf{x}) := \left(\frac{(|\mathbf{m}_h^\varepsilon(\mathbf{x})| - 1)_+}{|\mathbf{m}_h^\varepsilon(\mathbf{x})|} |\mathbf{m}_h^\varepsilon(\mathbf{x})|, \psi_j \right)$. Finally the evaluation of the function $F(\mathbf{x})$ reads

$$F(\mathbf{x}) = \mathbf{D}^T \mathbf{V} \mathbf{D} \mathbf{x} + q \mathbf{Q} \mathbf{x} + \mathbf{a}(\mathbf{x}) - \mathbf{b}.$$

The derivative matrix $\mathbf{D}\mathbf{F}$ reads

$$\mathbf{D}\mathbf{F}(\mathbf{x}) = \mathbf{D}^T \mathbf{V} \mathbf{D} + q \mathbf{Q} + \mathbf{D}\mathbf{F}_{\text{NL}}(\mathbf{x}).$$

4.3.1 Quadrature rules

We use two different sets of quadrature rules on triangles. First, we use Gauss quadrature rules with 3 and 7 quadrature points which can be found e.g. in [Str71], see also Figure 4.3. The 3 point formula (see Table 4.1) is of order 2 and serves for the computation of the system matrix \mathbf{Q} . Note that since each basis function ψ_i is piecewise linear, i.e. the product $\psi_{i,2}\psi_{j,2}$ is an elementwise quadratic function and is integrated exactly by this rule. The 7 point Gauss rule (see Table 4.2) is used in the computation of the entries for the simple-layer matrix \mathbf{V} . To compute $\mathbf{V}_{i,j}$ we must integrate

$$\frac{1}{4\pi} \int_{T_j} \int_{T_i} \frac{1}{|x - y|} dy dx.$$

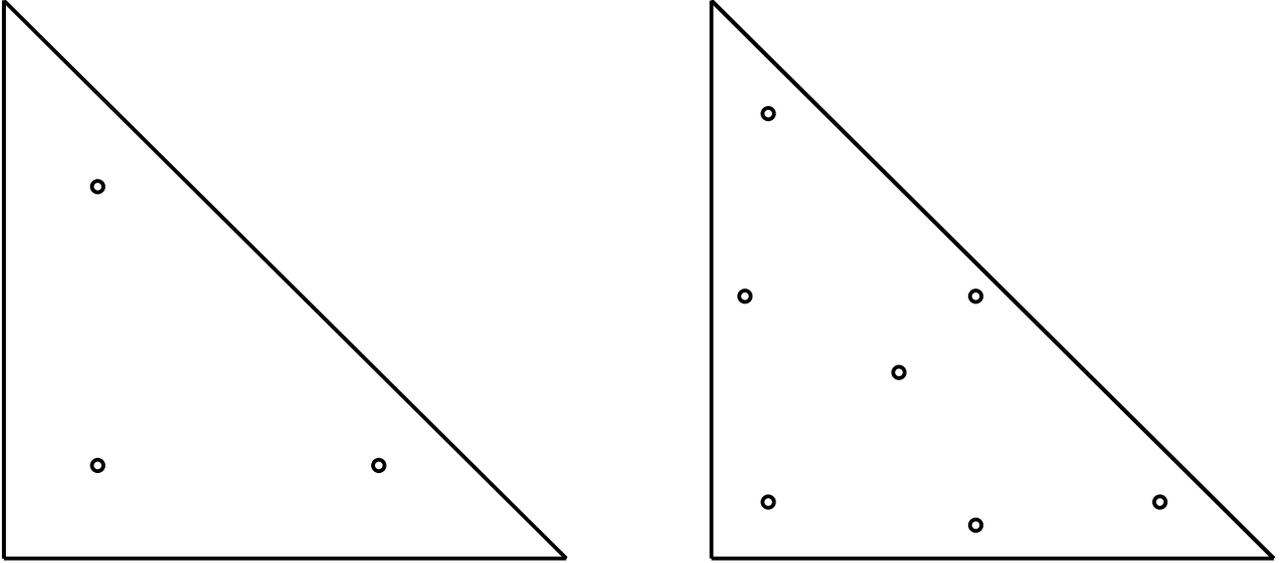


Figure 4.3: 3 point (left) and 7 point (right) Gauss quadrature rules on the reference triangle.

Point	x	y	Weight
0	0.101286507323456	0.101286507323456	0.125939180544827
1	0.797426985353087	0.101286507323456	0.125939180544827
2	0.101286507323456	0.797426985353087	0.125939180544827
3	0.470142064105115	0.059715871789770	0.132394152788506
4	0.470142064105115	0.470142064105115	0.132394152788506
5	0.059715871789770	0.470142064105115	0.132394152788506
6	0.333333333333333	0.333333333333333	0.225000000000000

Table 4.2: A 7 point Gauss quadrature rule on the reference triangle $T_{\text{ref}} = \text{conv}\{(0, 0), (1, 0), (0, 1)\}$.

We do so by replacing the outer integration with the 7 point Gauss rule and computing for each quadrature point the inner integral analytically. The formulae for the analytical computation of the inner integral can be found in [Hac95].

We stress that in the so-called near field, i.e.

$$\min\{\text{diam}T_j, \text{diam}T_i\} \geq \eta \text{dist}(T_j, T_i)$$

with some suitable constant $\eta > 0$, we are not aware of any error analysis for the introduced quadrature error. In contrast, in the far field, i.e. for

$$\min\{\text{diam}T_j, \text{diam}T_i\} \leq \eta \text{dist}(T_j, T_i), \quad (4.4)$$

the quadrature error is known to decay exponentially with the order of integration, cf. [Hac09]. Finally, we want to remark that in our experiments we observe that, at least for the error level we are interested in, this quadrature strategy is sufficient in the sense that the quadrature error does not contribute significantly.

The second quadrature rule we use in our implementation is a Newton-Cotes formula from [Sil70]. To be precise, we use the 10 point rule specified in Table 4.3, see also Figure 4.4. Whereas Gauss rules are known to be a very good choice when dealing with smooth functions, the terms associated

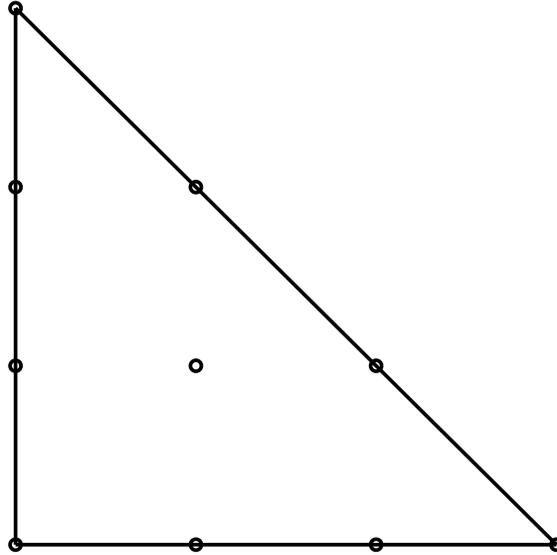


Figure 4.4: 10 point Newton-Cotes quadrature rule on the reference triangle.

Point	x	y	Weight
0	1	0	0.0333333333333333
1	0	0	0.0333333333333333
2	0	1	0.0333333333333333
3	2/3	1/3	0.075
4	1/3	0	0.075
5	0	2/3	0.075
6	1/3	2/3	0.075
7	0	1/3	0.075
8	2/3	0	0.075
9	1/3	1/3	0.45

Table 4.3: A 10 point Newton-Cotes quadrature rule on the reference triangle $T_{\text{ref}} = \text{conv}\{(0,0), (1,0), (0,1)\}$.

with the penalty energy involve the quadrature of the non-smooth function $(|\mathbf{m}_h| - 1)_+$. Since the Raviart-Thomas functions \mathbf{m}_h are piecewise linear, their maximal length is always attained at a corner point of the triangle. In our experiments we found that the Gauss quadrature rules are therefore not the best choice for the computation of the non-linear term F_{NL} or of its derivative. Some elements that should be penalized are ignored by the Gauss rule because of $|\mathbf{m}_h| < 1$ at all Gauss points. In contrast, the Newton-Cotes formula seems to be accurate enough in the sense that quadrature errors don't seem to contribute significantly to the overall error.

4.3.2 Sparse matrix representation and HLib

All of the system matrices defined in (4.3) allow for a sparse data representation. In the case of \mathbf{D} , \mathbf{DF}_{NL} , and \mathbf{Q} this is done straight forwardly by storing the matrices in a sparse compressed column format. In our software we use the implementation provided by the HLib library. The library does not only provide a data structure for the storage of sparse matrices but also functions for the

efficient computation of matrix vector products.

The most challenging one of the system matrices is the matrix \mathbf{V} associated with the simple-layer potential. Recall

$$\mathbf{V}_{ij} = \frac{1}{4\pi} \int_{T_j} \int_{T_i} \frac{1}{|x-y|} dy dx$$

and observe that the matrix is fully populated. This is a problem that arises in most calculations involving non-local boundary integral operators and it has been treated by the scientific community working on boundary element methods in several ways. Most of the methods are based on the observation that whereas the kernel function

$$\kappa(x, y) = \frac{1}{|x-y|}$$

has a singularity at the diagonal $x = y$, it is \mathcal{C}^∞ for $x \neq y$. This allows for efficient approximation of submatrices $\mathbf{V}_{\mathcal{I}, \mathcal{J}}$ with \mathcal{I} and \mathcal{J} subsets of indices such that $\text{dist}(\bigcup_{i \in \mathcal{I}} T_i, \bigcup_{j \in \mathcal{J}} T_j)$ sufficiently large.

Definition. A subset $\mathcal{I} \times \mathcal{J}$ of $\{1, \dots, n\} \times \{1, \dots, n\}$ is called admissible, if for given $\eta > 0$

$$\min\{\text{diam} \bigcup_{i \in \mathcal{I}} T_i, \text{diam} \bigcup_{j \in \mathcal{J}} T_j\} \leq \eta \text{dist}(\bigcup_{i \in \mathcal{I}} T_i, \bigcup_{j \in \mathcal{J}} T_j).$$

□

For the sparse data representation of \mathbf{V} , we use the `supermatrix` data structure provided by the HLib library. The HLib package was developed at the Max-Planck-Institute for Mathematics in the Sciences and is available at www.hlib.org free for academic purposes. It is based on the implementation of \mathcal{H} -matrix arithmetics of the dissertation [Gra01]. The `supermatrix` structure implements the storage of hierarchical matrices [Hac99]. The idea is to store inadmissible blocks of \mathbf{V} as full matrix, but use a low rank approximation for the storage of admissible blocks of \mathbf{V} . Figure 4.5 shows the typical structure of a hierarchical matrix. The inadmissible blocks are colored in red, admissible blocks are colored green. Instead of storing an admissible block $\mathbf{V}_{\mathcal{I} \times \mathcal{J}}$ elementwise, an approximation of the form

$$\mathbf{V}_{\mathcal{I} \times \mathcal{J}} \approx UV^T$$

with $U \in \mathbb{R}^{\#\mathcal{I} \times k}$ and $V \in \mathbb{R}^{k \times \#\mathcal{J}}$ where $k \ll \min\{\#\mathcal{I}, \#\mathcal{J}\}$. It can be shown that the storage requirements and the number of arithmetic operations for the matrix vector multiplication have almost linear complexity $\mathcal{O}(n \log n)$ [Hac09].

Many possibilities to compute suitable low-rank approximations for admissible blocks have been proposed in the literature. For a good overview on hierarchical matrices in general and various approximation strategies, we refer to [Hac09]. In [Beb00], a black-box algorithm for the computation of low-rank approximations, called adaptive-cross-approximation (ACA), is presented. The proof of the error analysis only works for collocation matrices, however in [Beb08] also an error analysis for Galerkin matrices, such as is our matrix \mathbf{V} , is provided. For the computation of our low-rank approximations, we use the implementation of the ACA algorithm provided by the HLib library. We only need to implement a function for the evaluation of a matrix entry \mathbf{V}_{ij} from which the admissible low-rank approximations with local error control are built. As local error tolerance for the admissible blocks we use $1e-8$ in our implementation.

We stress that this strategy is not the most efficient one with respect to storage requirements or fast evaluation. A large part of the dissertation [Drw08] is concerned with the efficient computation of the system matrix \mathbf{V} and the solution of linear soft thin-film problems with $q = 0$ and $|\mathbf{f}| \ll 1$.

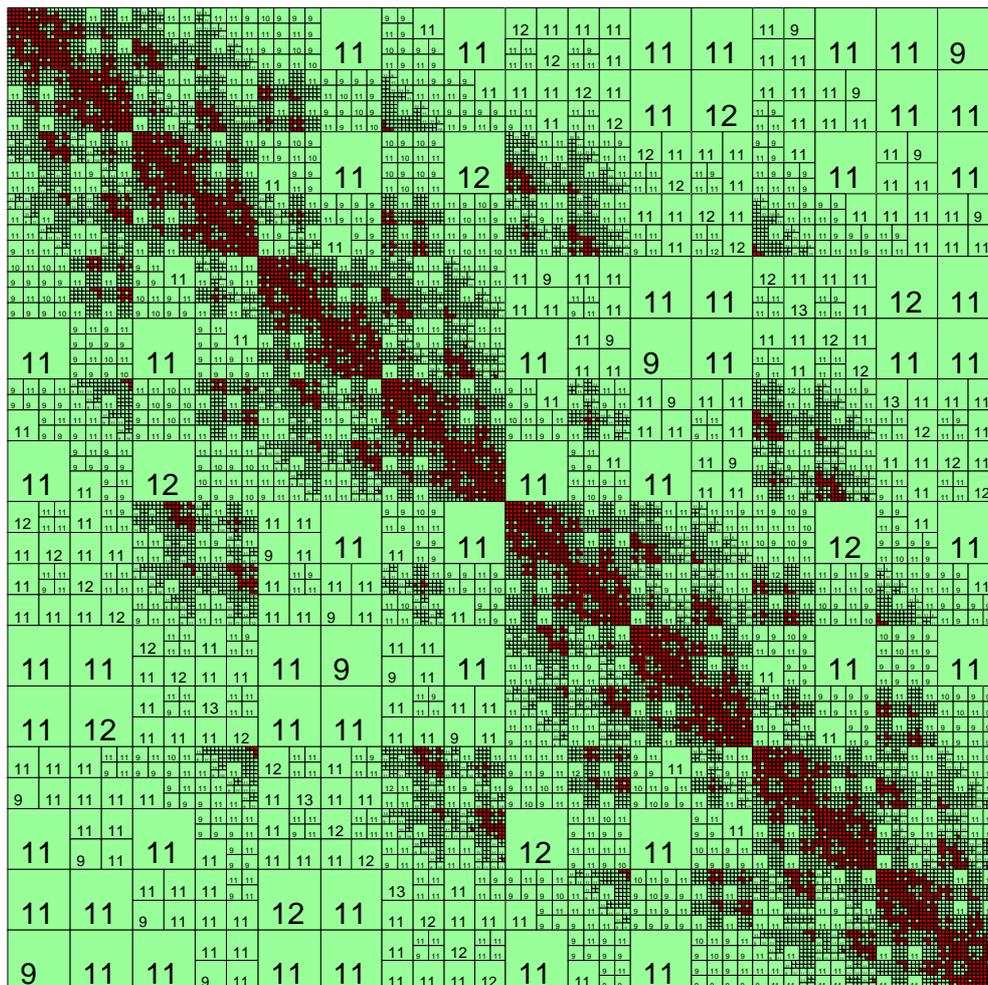


Figure 4.5: Typical structure of a hierarchical matrix. Inadmissible blocks (red) are stored uncompressed, admissible blocks (green) are approximated by low-rank matrices.

4.3.3 Code listing

Listing 4.13: The Newton algorithm with uniform mesh refinements

```

1 Vector2D f(Point x);
2
3 //Compute  $D^T * V * D * x$ 
4 double* evalDTVD(SimpleLayerPotential& V, sparsematrix* D, const double* x){
5     double* Dx = new double[D->rows];
6     eval_sparsematrix(D,x,Dx);
7     double* VDx = V * Dx;
8     delete Dx;
9     double* DTVDx = new double[D->cols];
10    evaltrans_sparsematrix(D,VDx,DTVDx);
11    delete VDx;
12    return DTVDx;
13 }
14
15 int main()
16 {
17     cout<<"STARTING_COMPUTATION\n"
18         <<"===== \n\n";
19     RT0Space femSpace = RT0Space("coordinates.dat","elements.dat");
20     double q = 1;
21     MAX_ELEMENTS = 50000
22
23     double* mH;
24     double* epsilon;
25     int mesh_refinements = 0;
26
27     while(femSpace.NumberOfElements() < MAX_ELEMENTS) {
28         if(mesh_refinements == 0) {
29             mH = new double[femSpace.NumberOfElements()];
30             for(int i = 0; i < femSpace.NumberOfElements(); ++i)
31                 mH[i] = 0.;
32             epsilon = new double[femSpace.NrOfTriangles()];
33             double eps = femSpace.TrianglesBegin()->MeshWidth();
34             for(int i = 0; i < femSpace.NrOfTriangles(); ++i)
35                 epsilon[i] = eps;
36         }
37         else {
38             double* oldmH = new double[femSpace.NumberOfElements()];
39             for(int ell = 0; ell < femSpace.NumberOfElements(); ++ell)
40                 oldmH[ell] = mH[ell];
41
42             RefinementData* refinedata = 0;
43             refinedata = femSpace.Refine(true);
44             cout<<"done.\n"<<flush;
45
46             delete mH;
47             mH = new double[femSpace.NumberOfElements()];
48             eval_sparsematrix(refinedata->Prolongation,oldmH,mH);
49             delete oldmH;
50
51             double* epsilon_new = new double[femSpace.NrOfTriangles()];
52             eval_sparsematrix(refinedata->ProlongationDiff,epsilon,epsilon_new);
53             delete epsilon;
54             epsilon = epsilon_new;
55             for(int i = 0; i < femSpace.NrOfTriangles(); ++i)
56                 epsilon[i] *= 0.5;
57

```

```

58     del_sparsematrix(refineData->Prolongation);
59     del_sparsematrix(refineData->ProlongationDiff);
60     delete refineData;
61 }
62
63 cout<<"Build_Q_..._ "<<flush;
64 sparsematrix* Q = stimaQ(femSpace);
65 cout<<"done.\n";
66
67 cout<<"Build_V_..._ "<<flush;
68 SimpleLayerPotential V = SimpleLayerPotential(P0Space(femSpace), 1e-8);
69 cout<<"done.\n";
70
71 cout<<"Build_Div_..._ "<<flush;
72 sparsematrix* D = DivergenceOperator(femSpace);
73 cout<<"done.\n";
74
75 double* Fx = new double[femSpace.NumberOfElements()];
76 for(int i = 0; i < femSpace.NumberOfElements(); ++i)
77     Fx[i] = 1.;
78
79 //Build RHS
80 double* a = buildRHS(f, femSpace);
81
82 double normFx = 0.;
83 for(int i = 0; i < femSpace.NumberOfElements(); ++i)
84     normFx += Fx[i]*Fx[i];
85 normFx = sqrt(normFx);
86 sparsematrix* DF_NL = 0;
87
88 while(normFx > 1e-8) {
89     //compute nonlinear term
90     double* f_NL = computeNonlinearTermAdaptive(femSpace, mH, epsilon);
91
92     //Compute F(x)
93     double* DTVDmH = evalDTVD(V,D,mH);
94     double* QmH = new double[femSpace.NumberOfElements()];
95     eval_sparsematrix(Q,mH,QmH);
96     for(int i = 0; i < femSpace.NumberOfElements(); ++i)
97         Fx[i] = DTVDmH[i] + q * QmH[i] - a[i] + f_NL[i];
98
99     //Compute derivative DF_NL
100    if(iter > 1)
101        del_sparsematrix(DF_NL);
102    cout<<"Compute_DF_NL_..._ "<<flush;
103    DF_NL = computeNonlinearTermDerivativeAdaptive(femSpace, mH,
104                                                    epsilon);
105    cout<<"done.\n";
106
107    //compute damping-parameter and perform Newton-step
108    double damp = 0.5;
109    normFx = 0.;
110    for(int i = 0; i < femSpace.NumberOfElements(); ++i)
111        normFx += Fx[i]*Fx[i];
112    normFx = sqrt(normFx);
113
114    double* delta = solveGMRES(Fx, V, D, q, Q, DF_NL);
115
116    double* x1 = new double[femSpace.NumberOfElements()];
117    for(int i = 0; i < femSpace.NumberOfElements(); ++i)
118        x1[i] = mH[i] - delta[i];

```

```

119
120     double* Fx1 = new double[femSpace.NumberOfElements()];
121
122     double* DTVDx1 = evalDTVD(V,D,x1);
123
124     double* Qx1 = new double[femSpace.NumberOfElements()];
125     eval_sparsematrix(Q,x1,Qx1);
126     double* nl = computeNonlinearTermAdaptive(femSpace, x1, epsilon);
127
128     for(int i = 0; i < femSpace.NumberOfElements(); ++i)
129         Fx1[i] = DTVDx1[i] + q * Qx1[i] - a[i] + nl[i];
130
131     double normFx1 = 0.;
132     for(int i = 0; i < femSpace.NumberOfElements(); ++i)
133         normFx1 += Fx1[i]*Fx1[i];
134     normFx1 = sqrt(normFx1);
135
136     delete Fx1;
137     delete DTVDx1;
138     delete Qx1;
139
140     while(normFx1>normFx) {
141         damp = damp*0.5;
142         for(int i = 0; i < femSpace.NumberOfElements(); ++i)
143             x1[i] = mH[i] - damp*delta[i];
144
145         Fx1 = new double[femSpace.NumberOfElements()];
146
147         double* DTVDx1 = evalDTVD(V,D,x1);
148
149         Qx1 = new double[femSpace.NumberOfElements()];
150         eval_sparsematrix(Q,x1,Qx1);
151         nl = computeNonlinearTermAdaptive(femSpace, x1, epsilon);
152
153         for(int i = 0; i < femSpace.NumberOfElements(); ++i)
154             Fx1[i] = DTVDx1[i] + q * Qx1[i] - a[i] + nl[i];
155
156         normFx1 = 0.;
157         for(int i = 0; i < femSpace.NumberOfElements(); ++i)
158             normFx1 += Fx1[i]*Fx1[i];
159         normFx1 = sqrt(normFx1);
160
161         delete Fx1;
162         delete DTVDx1;
163         delete Qx1;
164     }
165     for(int i = 0; i < femSpace.NumberOfElements(); ++i)
166         mH[i] = x1[i];
167
168     delete f_NL;
169     delete DTVDmH;
170     delete QmH;
171     delete x1;
172     delete nl;
173 }
174 del_sparsematrix(DF_NL);
175
176 cout<<"Computing_energy_..._ "<<flush;
177 double* DTVDmH = evalDTVD(V,D,mH);
178 double* QmH = new double[femSpace.NumberOfElements()];
179 eval_sparsematrix(Q, mH, QmH);

```

```
180     double energy = 0.;
181     for (int i = 0; i < femSpace.NumberOfElements(); ++i)
182         energy += 0.5 * (mH[i]*DTVDmH[i]) + 0.5 * (mH[i]* q * QmH[i]) -
183             (a[i]*mH[i]);
184     cout<<"_E=__"<<energy<<"\n";
185
186     //Cleanup memory
187     del_sparsematrix(Q);
188     del_sparsematrix(QFull);
189     del_sparsematrix(D);
190     delete Fx;
191     delete a;
192     ++mesh_refinements;
193 }
194 return 0;
195 }
```

Chapter 5

Numerical experiments

5.1 Overview and general remarks

In this chapter, we study several aspects of our analysis by providing extensive numerical experiments.

Discretization error: As a first topic in Section 5.2, we study the behavior of the discretization error

$$\|\mathbf{m}_0^\varepsilon - \mathbf{m}_h^\varepsilon\|.$$

First, we perform an experiment for the linear case with weak applied field $|\mathbf{f}| \ll 1$. The constraint $|\mathbf{m}| \leq 1$ is not active and the penalty scheme does not contribute to the error. For smooth data, we observe a rate of

$$\|\mathbf{m}_0^\varepsilon - \mathbf{m}_\ell\| = \mathcal{O}(h^{1/2})$$

for a sequence of discrete solutions \mathbf{m}_ℓ computed on uniform meshes. Note that $h \sim N_T^{-1/2}$ for N_T the number of triangles. This means the rate $h^{1/2}$ corresponds to $N_T^{-1/4}$.

In the second experiment in Section 5.2.2, we apply a stronger field \mathbf{f} such that the penalty scheme is active in some regions of the simulation domain $\omega = (-0.5, 0.5)^2$. To ensure that the error stemming from the penalization is of higher order, we choose a relatively small parameter of $\varepsilon = 5e - 3$. Again, a sequence of uniform meshes reveals a rate of $\mathcal{O}(N_T^{-1/4})$.

In the first two experiments with uniform mesh-refinements, we observe that the L^2 -contribution

$$\|\mathbf{m}_{0,2}^\varepsilon - \mathbf{m}_{\ell,2}\|_{L^2}$$

to the total error is of higher order

$$\|\mathbf{m}_{0,2}^\varepsilon - \mathbf{m}_{\ell,2}\|_{L^2} = \mathcal{O}(h) = \mathcal{O}(N_T^{-1/2}).$$

The divergence $\nabla \cdot \mathbf{m}_0^\varepsilon$, however, has edge and corner singularities. Uniform meshes allow for resolution of this quantity only at the inferior rate of $\mathcal{O}(h^{1/2})$, which, therefore, dominates the overall error. In the simplified linear case of soft films with $q = 0$ and $|\mathbf{f}| \ll 1$, it is well known, see e.g. [FLP08], that the divergence $\nabla \cdot \mathbf{m}$ can be resolved at almost linear rate $\mathcal{O}(N_T^{-1/2})$ using adaptively generated isotropic sequences of meshes.

We apply our h -adaptive algorithm of Section 3.4 to the first two experiments, i.e. we use $\tilde{\mu}_\ell^H(T)$ as an indicator for the mesh-refinement. Apparently, the optimal resolution of the L^2 -component $\mathbf{m}_{0,2}^\varepsilon$ of the magnetization demands uniform meshes. The divergence, however, demands strong edge refinements, which in turn yields $\|\mathbf{m}_{0,2}^\varepsilon - \mathbf{m}_\ell\|_{L^2} = \mathcal{O}(N_T^{-\alpha})$ with $\alpha < 1/4$. The two goals seem to be incompatible, and we observe that the adaptive algorithm does improve the asymptotics of the error only slightly. The use of anisotropic sequences of meshes might provide a solution to this problem. It is well known that the divergence can be resolved more efficiently with anisotropic elements along the boundary, cf. [FLP08]. Even though the adaptive computations do not improve the asymptotic behavior significantly, the error level, and specially the error in the energy

$$|e(\mathbf{m}_0^\varepsilon) - e(\mathbf{m}_\ell)|,$$

is reduced when compared to the uniform approach. This is because the dominant error contribution $\|\nabla \cdot (\mathbf{m}_0^\varepsilon - \mathbf{m}_\ell)\|_V$ is first resolved at a higher rate, until both error components are at the same level.

Micromagnetic devices are usually studied at small scales and it is a common modeling assumption that the applied field is (almost) constant. Our interest goes beyond the concrete application of the model problem (M) as we wish to study the properties of the penalty method in general. In Section 5.2.3, we construct an example where \mathbf{f} is discontinuous and has a weak singularity at the origin. Note that our a priori analysis is based on Proposition 3.16 and the statement demands \mathbf{f} to be smooth. Hence, we are testing the performance of our numerical scheme even when necessary regularity assumptions of the analysis are violated. The discontinuity lines are chosen such that they cannot be resolved precisely by any refinement of the initial mesh. The uniform algorithm performs at a decreased rate $\mathcal{O}(N^{-\alpha})$ with $\alpha < 1/4$. The adaptive algorithm seems to resolve the singularities of \mathbf{m}_0^ε more efficiently and recovers a rate of at least $\mathcal{O}(N_T^{-1/4})$.

Finally, Section 5.2.4 is concerned with the influence of the penalty parameter ε on the discretization error. We compute numerical solutions for the same data and different values of ε . Empirically, we observe the discretization error to be almost independent of the penalty scheme. Only for examples where the penalty energy is very large, we observe a slight positive influence in the sense that the discretization error is reduced for smaller penalty parameter.

Conclusions: Theorem 3.24 proves a rate of

$$\|\mathbf{m}_0^\varepsilon - \mathbf{m}_h^\varepsilon\| = \mathcal{O}(h^{1/2})$$

for smooth $\mathbf{m}_0^\varepsilon \in H^1(\omega)$ with $\nabla \cdot \mathbf{m}_0^\varepsilon \in H^{1/2}(\omega)$. In practice, this seems to be the rate that may be expected for uniform meshes if the given data is smooth. However, apparently the statement of Theorem 3.24 holds even under weaker regularity assumptions on \mathbf{m}_0^ε . The error estimator η_ℓ^H is empirically efficient and reliable. Moreover, the local indicator $\mu_{\ell,V}^H$ not only gives a lower bound as stated in Corollary 3.28, but empirically also gives an upper bound,

$$\eta_{\ell,V}^H \lesssim \mu_{\ell,V}^H,$$

as is known for soft films ($q = 0$) in the linear case $|\mathbf{f}| \ll 1$. The isotropic adaptive algorithm increases the accuracy, specially of the energy, in all of our computations and recovers an asymptotic order of at least $\mathcal{O}(N_T^{-1/4})$ even in presence of singularities caused by non-smooth data.

Penalization error: Section 5.3 is concerned with the experimental analysis of the error introduced by the penalty scheme

$$\|\mathbf{m}_h^0 - \mathbf{m}_h^\varepsilon\|.$$

Recall the statement

$$\|\mathbf{m}_h^0 - \mathbf{m}_h^\varepsilon\| = \mathcal{O}(\varepsilon^{1/2})$$

of Theorem 3.22. In [CP01], the authors succeeded to improve this a priori result for the large-body limit. The proof is based on L^2 -orthogonalities of the discrete solution. In our case, we do not have such orthogonalities which is why we did not succeed to provide a similar improvement of the a priori result. One might suspect, though, that some other techniques allow for a proof of linear convergence. We therefore perform some numerical experiments to empirically verify our analysis. In the experiments in Section 5.3.1, we compute solutions with fixed meshes and varying penalty parameter for smooth data. We do observe in this uniform case of ε -refinements a rate of $\mathcal{O}(\varepsilon^{1/2})$, which indicates that the statement of Theorem 3.22 cannot be improved in general. Note that in contrast to the results concerning the convergence rates with respect to h , the rate with respect to the penalty error does not depend heavily on the regularity of \mathbf{m}_h^0 .

In our first series of experiments, we observe that the heuristic error estimator η_ℓ^ε is empirically reliable and gives a true upper bound of the error. The penalty error is not independent of the discretization. A reduction of the mesh size leads to an improvement of the penalty error. This, however, does not cause problems, as it means that simultaneous h - and ε -refinements may lead to an improved rate of the penalty error.

In Section 5.3.2, we give empirical evidence that the choice of $\varepsilon = h^\alpha$ with $\alpha = 1$ is optimal for uniform mesh refinements. Choice of $\alpha > 1$ does not improve the asymptotic behavior of the error whereas choice of $\alpha < 1$ reveals a reduced order of convergence when compared to $\alpha = 1$. One interesting fact is that the order of convergence for simultaneous h - and ε -refinements is increased, at least for examples, where the penalization is active on large regions of ω . This may be due to the slight decrease of discretization error for smaller ε and the large decrease of the penalty error for smaller discretization parameter h .

Conclusions: The penalty error behaves as predicted by our analysis. It seems that the a priori estimate cannot be improved in general. The penalty error is not independent of the discretization. A finer mesh leads to a smaller penalty error. Finally, the choice of $\varepsilon = h$ in uniform computations appears to be optimal. We observe an increased order of convergence with respect to N_T , when the mesh and the penalty parameter are refined simultaneously. It is not fully clear whether this is only a pre-asymptotic effect or the rate is increased permanently.

Further simulations: In Section 5.4, we apply a field \mathbf{f} with strong singularities along the diagonal $x = y$. In contrast to the example from Section 5.2.3, here the discontinuity line of \mathbf{f} is resolved by the mesh exactly. As a consequence we observe that the L^2 component of the error still behaves well. The error introduced by the penalty scheme and the error stemming from the divergence, however, converge at a slow rate. The choice of \mathbf{f} effects a singularity of $\nabla \cdot \mathbf{m}$ along $x = y$. The h - ε -adaptive algorithm resolves the singularity of $\nabla \cdot \mathbf{m}$ and the ε -refinements improve the penalty error when compared to the uniform case. The penalty error and the discretization error are balanced by the algorithm, the order of convergence of the error estimator is improved when compared to the use of uniform refinements.

In the last section, we perform experiments for three examples where the anisotropy parameter is large, i.e. $q \gg 1$, or vanishes, i.e. $q = 0$. The sample $\omega = (-0.5, 0.5) \times (-0.1, 0.1)$ is rectangular and in the first experiment we choose $q = 1e4$, so that the anisotropy energy contribution is dominant. We apply the constant field $\mathbf{f} = 10(0.5, 2.0)^T$. Even though the applied field is very large, and almost orthogonal to the easy axis, we observe that the magnetization is still almost aligned with the first

in-plane axis. The penalization term is active on almost the entire domain ω . We observe for η_ℓ a rate of convergence $N_T^{-1/4}$ when using uniform refinements. The h - ε -adaptive algorithm improves this and leads to a rate of $N_T^{-1/3}$ for the estimator η_ℓ .

The two last examples are dedicated to the case of soft films with $q = 0$. We stress that the energy norm, then, only measures the divergence of \mathbf{m} in the V -norm, i.e. $\|\mathbf{m}\| = \|\nabla \cdot \mathbf{m}\|_V$. In the case of soft applied field $\mathbf{f} = 0.15(0.5, 2.0)^T$ the constraint is not active. We observe that uniform mesh refinements lead to a convergence order of $N_T^{-1/4}$. The h -adaptive algorithm improves the convergence up to $N_T^{-1/2}$. When the stronger field $\mathbf{f} = 2(0.5, 2.0)^T$ is applied the constraint is active. Uniform mesh refinements with $\varepsilon = h$ reveal convergence with order $N_T^{-1/4}$. The h - ε -adaptive algorithm, again, leads to a higher order of convergence of almost $N_T^{-1/2}$. In particular, the ε -refinements are aggressive and the algorithm seeks to balance the error contributions.

Representation of results: All results of our experiments are provided in tables. Additionally, we plot the error and error estimators in a double logarithmic scale. The quantities are plotted usually on the number of triangles of the mesh. Recall the link $h \sim N_T^{-1/2}$ for uniform meshes. From that we read that $h^{1/2}$ corresponds to a rate of $N_T^{-1/4}$ with respect to the number of triangles. Only in the experiments where we study the ε -convergence, we plot all quantities naturally on the choice of the parameter ε .

In order to get a feeling for the behavior of the numerical solutions, we often provide quiver-plots of some discrete magnetization. Usually, we use some relatively coarse solution on a uniform mesh for good visibility.

The divergence of the magnetization is of great importance as it effects the leading error contribution in many examples. Plotting piecewise constant functions on meshes with very small mesh-size and large number of elements can lead to figures that are hard to interpret. Therefore, instead we plot a piecewise linear and globally continuous function obtained by averaging the values of the piecewise constant function at the nodes. Let z be a node and let $\{T_1, \dots, T_n\}$ be the set of triangles that share z as a vertex. Given the piecewise constant function $X \in \mathcal{P}^0(\mathcal{T})$, we define the nodal value of a piecewise linear and globally continuous approximation $\tilde{X} \in \mathcal{S}^1(\mathcal{T})$ through

$$\tilde{X}(z) := \frac{1}{n} \sum_{i=1}^n X|_{T_i}.$$

This kind of interpolation plot is used throughout to visualize the divergence of discrete functions $\mathbf{m}_h \in RT^0(\mathcal{T})$ as well as other piecewise constant functions.

The effects of the constraint $|\mathbf{m}| \leq 1$ are visualized in the following way: First, we provide plots of the regions where the penalty-scheme contributes to the energy. Triangles of some mesh where $\|(|\mathbf{m}_\ell| - 1)_+\|_{L^2(T)} > 0$ are colored red, those where $|\mathbf{m}_\ell|_T < 1$ are colored blue. Second, we plot the piecewise constant function of the maximal length $L(\mathbf{m}_\ell) \in \mathcal{P}^0(\mathcal{T}_\ell)$ on each triangle, i.e.

$$L(\mathbf{m}_\ell)|_T = \max_{x \in T} |\mathbf{m}_\ell(x)|.$$

In those experiments where we use the fully adaptive algorithm to steer h and ε refinements, we also provide plots of the adaptive generated piecewise constant function $\varepsilon_\ell \in \mathcal{P}^0(\mathcal{T}_\ell)$.

Recall that the model problem under consideration is non-linear. The only known analytical solution that we are aware of is, unfortunately, the trivial solution $\mathbf{m} = 0$ in absence of an applied field. In

order to measure the error in various norms in our uniform simulation runs, we use some solution computed on a very fine mesh as reference solution. The discrete solutions obtained on coarser meshes are prolonged to the fine mesh. The system matrices $\mathbf{V}, \mathbf{D}, \mathbf{Q}$, and \mathbf{Q}_F are given with respect to the finest mesh. Let X_ℓ denote the coefficient vector of some prolonged discrete solution and X_n the coefficient vector of some reference solution. Then, the error in the energy norm may be computed by

$$(X_n - X_\ell)\mathbf{D}^T\mathbf{V}\mathbf{D}(X_n - X_\ell) + (X_n - X_\ell)\mathbf{Q}(X_n - X_\ell).$$

Similarly, the error in the full space norm may be computed by

$$(X_n - X_\ell)\mathbf{D}^T\mathbf{V}\mathbf{D}(X_n - X_\ell) + (X_n - X_\ell)\mathbf{Q}_F(X_n - X_\ell),$$

where \mathbf{Q}_F denotes the full stiffness matrix

$$(\mathbf{Q}_F)_{i,j} = (\psi_j, \psi_i)_{L^2}.$$

In case of adaptive calculations, it is hard to obtain a reference solution. First, convergence of the adaptive sequence does not imply convergence of the discrete solutions to the analytical solution. Second and more important, the red-green-blue refinement strategy employed does in general not allow to embed a mesh \mathcal{T}_ℓ into some uniform refinement of the initial mesh \mathcal{T}_0 easily. We therefore restrict to plotting the error estimators in the adaptive simulations. We stress that for uniform refinements, the estimator η_ℓ is observed to give reliable information on the overall simulation error.

5.2 Experimental analysis of discretization error

As pointed out in Section 3.4, the uniformity of h - and ε -convergence seems crucial for the heuristic adaptive algorithm. We therefore run some simulations with the aim of empirically studying the h -convergence of our proposed method. In contrast to the extended algorithm including the estimation of the penalty error, we restrict here to the pure h -adaptive strategy where ε is fixed and only the rescaled estimator $\tilde{\mu}_\ell^H$ is used as refinement indicator.

In this set of experiments, we choose the smoothest set-up that we could think of. The simulation domain is the unit square centered in the plane, i.e. $\omega = (-0.5, 0.5)^2$. The applied field $\mathbf{f} = (x, x)^T$ with $x > 0$ is constant and the anisotropy parameter is $q = 1$ to exclude effects possibly arising from very large or very small values of q . The penalty parameter is fixed at $\varepsilon = 5e - 3$. The applied field varies from soft to strong. Soft applied field means that $|\mathbf{f}|$ is sufficiently small such that the penalization is not active. The initial discretization of ω is depicted in Figure 5.1.

5.2.1 Soft applied field $\mathbf{f} = (0.1, 0.1)^T$

In our first experiment, we chose $f = (0.1, 0.1)^T$. In this case, the constraint is not active and we are computing the solution of a linear problem. Hence, the penalty parameter has no effect on the discretization error. Figure 5.2 shows a representative solution on a uniform mesh with $N_T = 1024$ triangles and $N_D = 1504$ degrees of freedom. One can see the effects of the anisotropy at the top and bottom edge of ω . We observe alignment of \mathbf{m}_ℓ along the axis of the applied field $(1, 1)$. However, the anisotropy seems to be relevant in the sense that the magnetization tends to spin a little bit more in direction of the first in-plane axis, i.e. the easy axis of the ferromagnetic sample. The divergence of the adaptively generated discrete solution \mathbf{m}_{14} is plotted in Figure 5.3. We observe that the solution has strong corner and edge singularities.

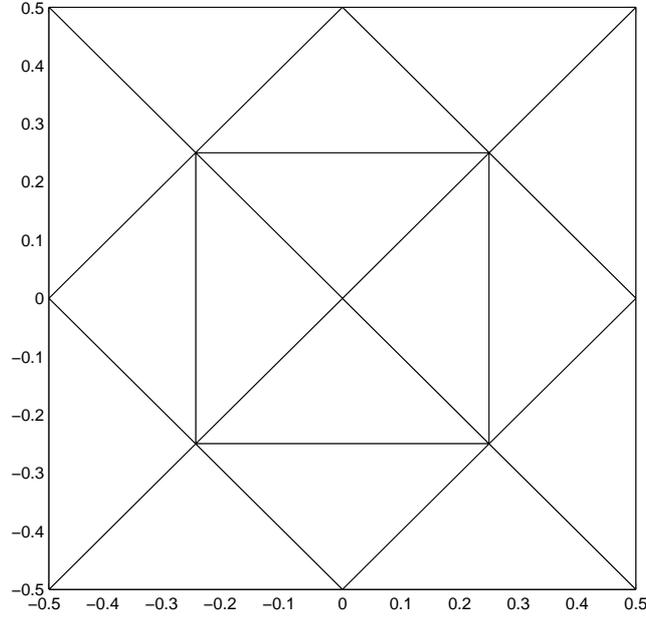


Figure 5.1: Initial mesh with 16 triangles and 20 interior edges for simulations where the domain is the unit square $\omega = (-0.5, 0.5)^2$ centered in the plane.

Iteration	N_T	N_D	Energy	$\eta_{\ell,V}^H$	$\eta_{\ell,Q}^H$	η_{ℓ}^H	$\mu_{\ell,V}^H$	N_{Newton}
0	16	20	-0.006413	0.03276	0.017083	0.03695	0.1486	2
1	64	88	-0.007103	0.02208	0.009776	0.02415	0.1004	2
2	256	368	-0.007399	0.01523	0.005400	0.01616	0.0696	2
3	1024	1504	-0.007532	0.01064	0.002921	0.01104	0.0489	2
4	4096	6080	-0.007594	0.00748	0.001559	0.00764	0.0345	2
5	16384	24448	-0.007624	0.00527	0.000825	0.00534	0.0244	2
6	65536	98048	-0.007638	-	-	-	-	2

Table 5.1: Results of the calculations with $\mathbf{f} = (0.1, 0.1)^T$, $q = 1$, $\omega = (-0.5, 0.5)^2$, and uniform mesh-refinements: mesh size N_T , space size N_D , energy, error estimators, and number of Newton iterations N_{Newton} .

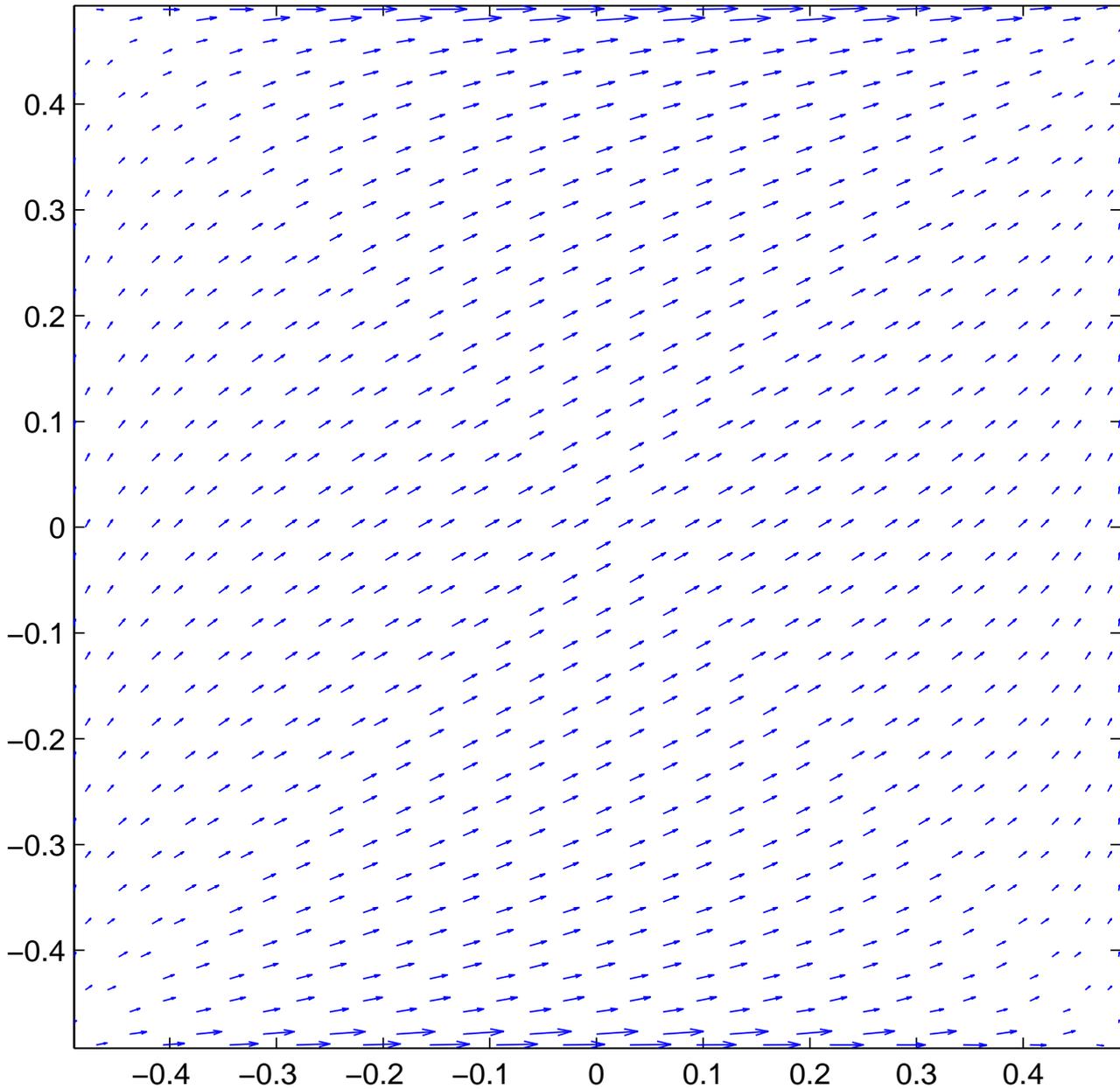


Figure 5.2: Discrete magnetization for $\mathbf{f} = (0.1, 0.1)^T$ and $q = 1$ on a uniform mesh of the domain $\omega = (-0.5, 0.5)^2$ with $N_T = 1024$ triangles and $N_D = 1504$ degrees of freedom.

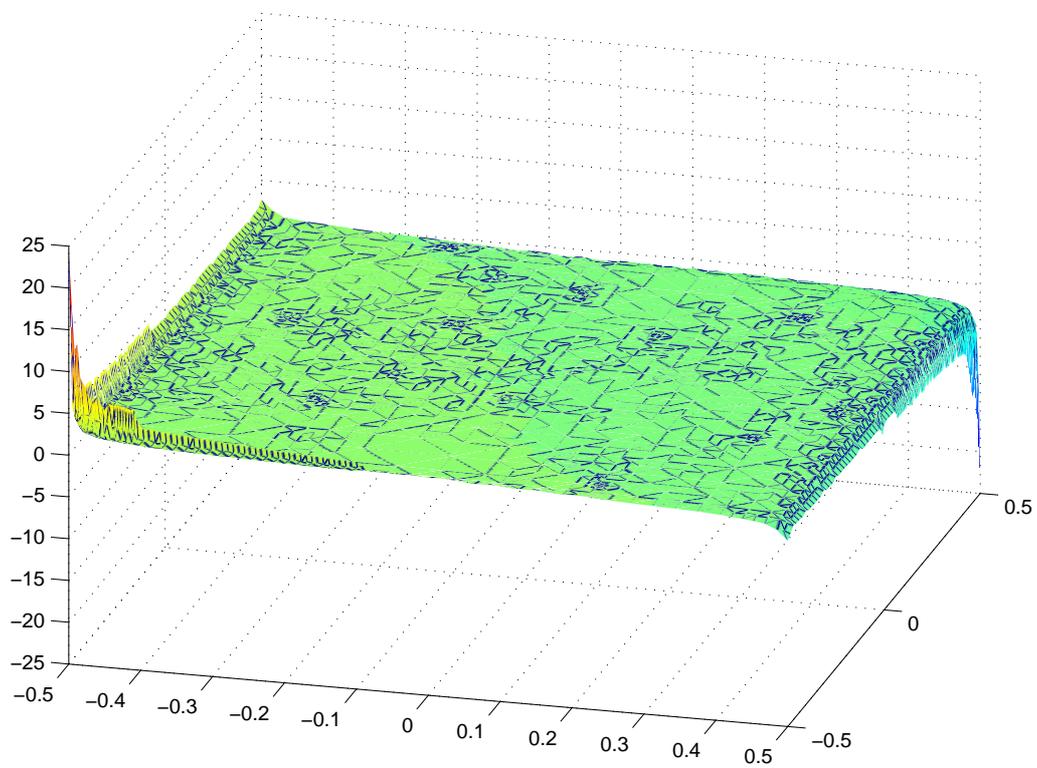


Figure 5.3: Divergence of the solution \mathbf{m}_{14} for $\mathbf{f} = (0.1, 0.1)^T$ and $q = 1$ on an h -adaptively generated mesh of the simulation domain $\omega = (-0.5, 0.5)^2$ with $N_T = 2058$ triangles and $N_D = 2976$ degrees of freedom.

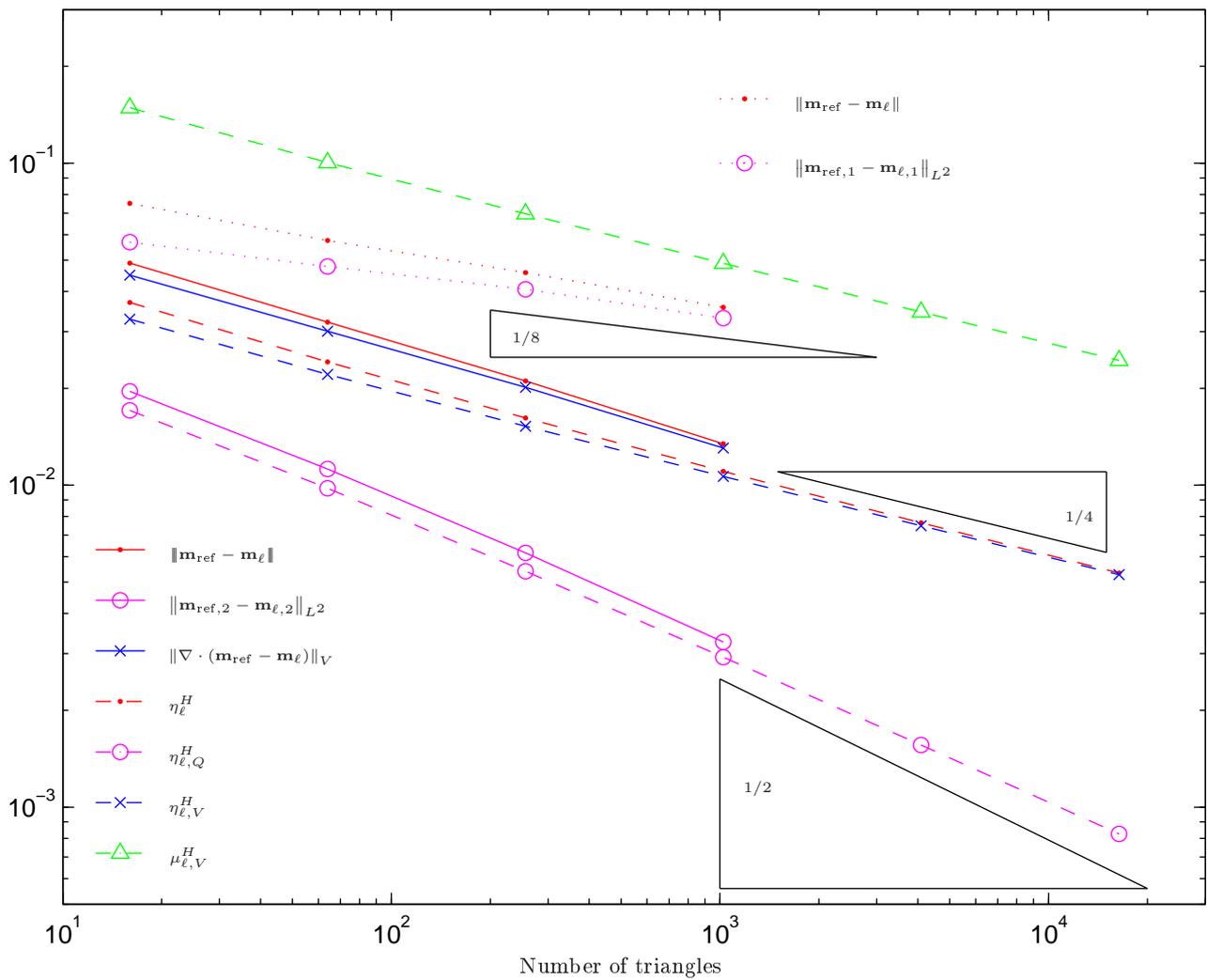


Figure 5.4: Error and error estimators for a sequence of discrete solutions \mathbf{m}_ℓ on uniform meshes with $\omega = (-0.5, 0.5)^2$, $\mathbf{f} = (0.1, 0.1)^T$, and $q = 1$.

Table 5.1 shows the experimental results for uniform mesh-refinements. Figure 5.4 shows the error and error estimators plotted on the number of triangles. We observe that the error in the energy norm decays approximately with order $N_T^{-1/4}$, which corresponds to $h^{1/2}$. The L^2 -contribution $\|\mathbf{m}_2^* - \mathbf{m}_{\ell,2}\|_{L^2(\omega)}$ is of higher order, i.e. almost $N_T^{-1/2}$ which corresponds to h^1 . We observe that the estimator η_ℓ^H gives a good approximation to the true error. Moreover, we empirically observe efficiency and reliability of η_ℓ^H and all of its components. The local error indicator $\mu_{\ell,V}^H$, which is proven to give a lower bound for $\eta_{\ell,V}^H$ up to some constant, also gives empirically an upper bound, we observe $\eta_{\ell,V}^H \sim \mu_{\ell,V}^H$. Recall that the energy functional does not measure the easy-axis component of \mathbf{m}_ℓ . As a consequence, we observe a dramatically reduced order of convergence for the first component \mathbf{m}_1 when compared to the order of the error in the second component. The L^2 -error $\|\mathbf{m}_1^* - \mathbf{m}_{\ell,1}\|_{L^2(\omega)}$ seems to dominate the full space norm error $\|\mathbf{m}^* - \mathbf{m}_\ell\|$. Both decay at a rate of approximately $N_T^{-1/8}$.

Figure 5.5 shows the sequence of adaptively generated meshes. At first, we observe strong refinements towards the lower left and top right corners as well as some refinements towards the edges. In the last few refinement steps, however, also some triangles in the interior of ω are refined. Figure 5.6 shows the error estimators in the adaptive computation. For reference, we also plotted the quantities obtained with uniform refinements. At first, the L^2 -contribution of the energy error is significantly smaller than the contribution of the error in the divergence. However, in the last few refinement steps it seems that some elements were also marked due to the L^2 -norm contributed error estimator $\eta_{\ell,Q}^H$. It seems that it is not possible to resolve both, L^2 and V -norm contributions of the error efficiently with the same mesh.

We observe that the overall error decays asymptotically at least at the same rate $N_T^{-1/4}$ as in the uniform case. Apparently, the use of uniform meshes is optimal to resolve the magnetization \mathbf{m} . However, the divergence $\nabla \cdot \mathbf{m}$ demands adaptive meshes with strong refinements towards the edges and corners. These goals are competing and impossible to satisfy simultaneously. Hence, after some preasymptotic phase, where the error in the divergence is reduced to the level of the L^2 -error, the asymptotic behavior again reveals almost the same rate of $N_T^{-1/4}$ as in the uniform case. The results of the adaptive computation are listed in Table 5.2.

The convergence of the error $|e(\mathbf{m}_0^{5e-3}) - e(\mathbf{m}_h^{5e-3})|$ in the energy is estimated by extrapolating the sequence of discrete energies obtained with uniform meshes. For our error plots we use the estimate $e(\mathbf{m}^*) \approx -0.007652721009050$. Figure 5.7 shows the estimated error in the energy for both, uniform and adaptive computations. In the first steps, the adaptive algorithm clearly computes more accurate values than the uniform approach. This is probably because the error level of the divergence is reduced drastically. However, in the last few steps, where the L^2 -component of the error is at the same level as the error of the divergence in the V -norm, we observe a slow-down of the convergence.

We stress that computations on finer meshes beyond iteration 14 showed numerical instabilities. Considering the very low level of the error in Figure 5.7, this is not surprising. Recall that we use an adaptive cross approximation for the sparse storage of the \mathbf{V} -matrix with local error tolerance of $1e-8$. Moreover, the use of numerical quadrature for the computation of the matrix entries might be relevant in this specific experiment. The search direction in the Newton algorithm is computed up to an absolute error if $1e-10$.

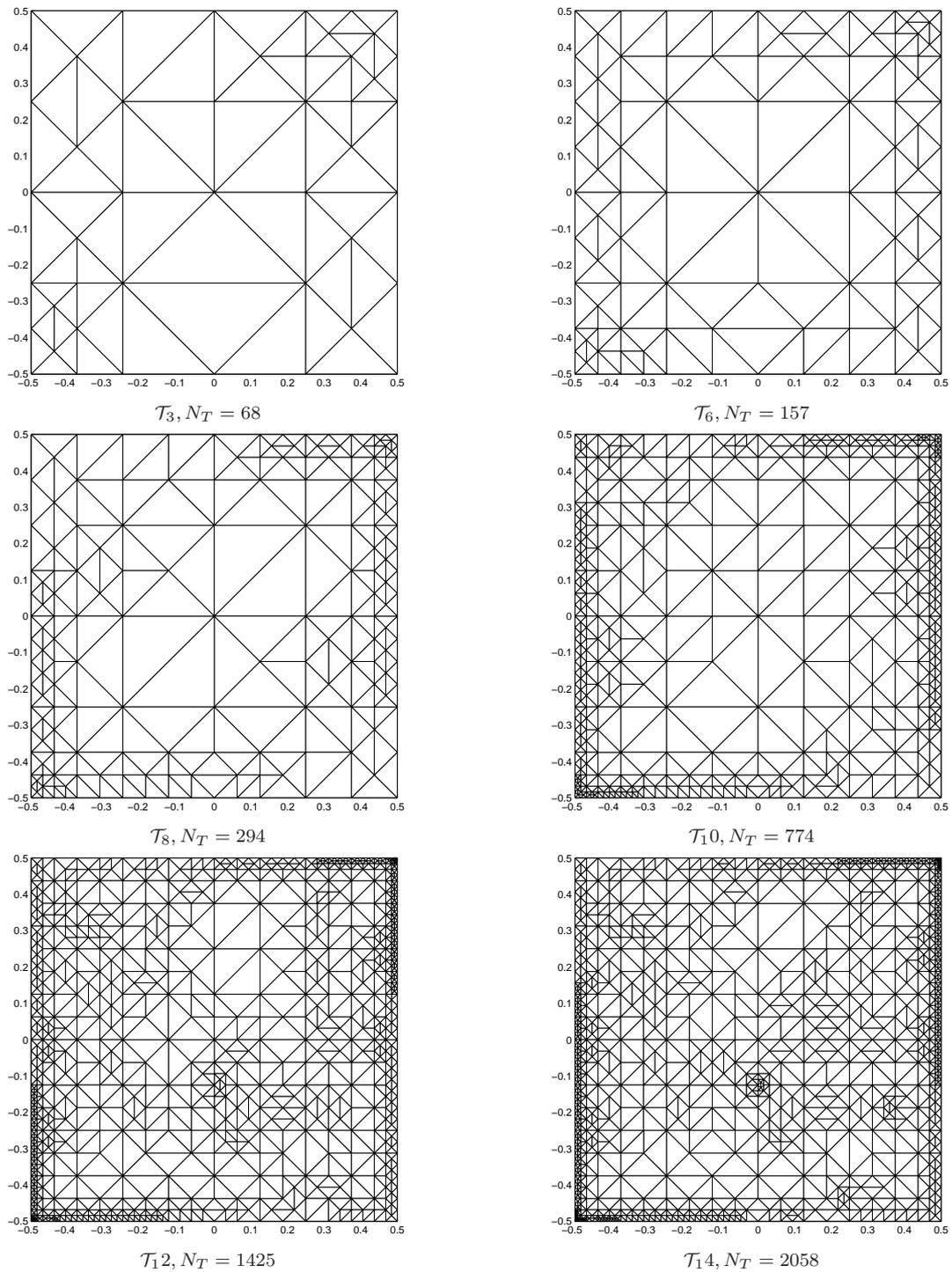


Figure 5.5: Sequence of h -adaptively generated meshes with $\omega = (-0.5, 0.5)^2$, $\mathbf{f} = (0.1, 0.1)^T$, and $q = 1$.

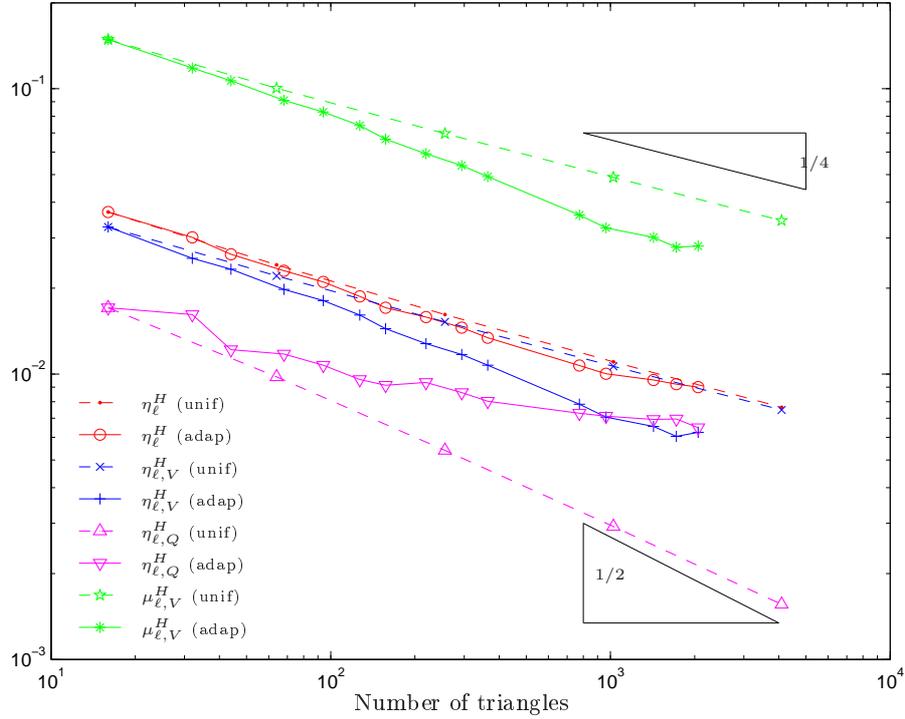


Figure 5.6: Error estimators for h -adaptive and uniform mesh-refinements. The field $\mathbf{f} = (0.1, 0.1)^T$ is applied to the sample $\omega = (0.5, 0.5)^2$ with anisotropy parameter $q = 1$.

Iteration	N_T	N_D	Energy	$\eta_{\ell,V}^H$	$\eta_{\ell,Q}^H$	η_ℓ^H	$\mu_{\ell,V}^H$	N_{Newton}
0	16	20	-0.006413	0.032765	0.017083	0.036951	0.14869	2
1	32	42	-0.006756	0.025432	0.016162	0.030133	0.11812	2
2	44	58	-0.006932	0.023311	0.012177	0.026300	0.10671	2
3	68	92	-0.007187	0.019804	0.011749	0.023027	0.09102	2
4	94	129	-0.007316	0.018082	0.010734	0.021028	0.08288	2
5	127	175	-0.007369	0.016093	0.009573	0.018725	0.07428	2
6	157	217	-0.007443	0.014422	0.009131	0.017070	0.06658	2
7	219	305	-0.007502	0.012789	0.009349	0.015842	0.05913	2
8	294	413	-0.007546	0.011714	0.008616	0.014542	0.05380	2
9	364	511	-0.007564	0.010733	0.008033	0.013407	0.04923	2
10	774	1101	-0.007565	0.007844	0.007294	0.010712	0.03611	2
11	964	1374	-0.007583	0.007063	0.007119	0.010028	0.03254	2
12	1425	2053	-0.007621	0.006550	0.006931	0.009536	0.03012	2
13	1720	2480	-0.007626	0.006050	0.006959	0.009222	0.02780	2
14	2058	2976	-0.007629	0.006246	0.006485	0.009004	0.02808	2

Table 5.2: Results of calculations with $\omega = (-0.5, 0.5)^2$, $\mathbf{f} = (0.1, 0.1)^T$, $q = 1$, and h -adaptive mesh-refinements: mesh size N_T , space size N_D , energy, error estimators, and number of Newton iterations N_{Newton} .

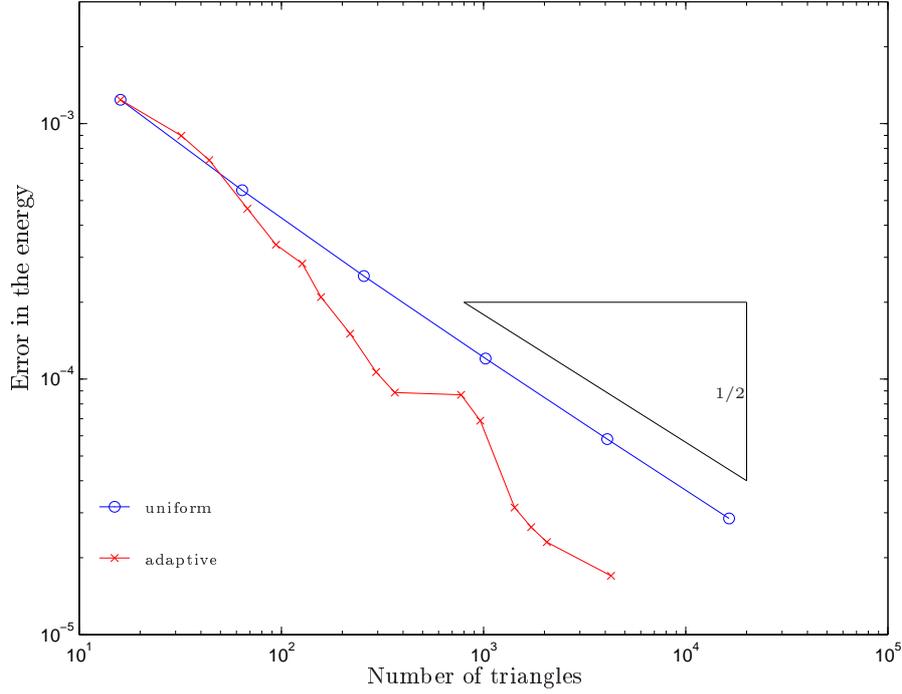


Figure 5.7: Error in the energy $|e(\mathbf{m}^*) - e(\mathbf{m}_\ell)|$ for a sequence of discrete solutions on uniform and h -adaptive meshes with $\omega = (-0.5, 0.5)^2$, $\mathbf{f} = (0.1, 0.1)^T$, and $q = 1$.

Iteration	N_T	N_D	Energy	$\eta_{\ell,V}^H$	$\eta_{\ell,Q}^H$	η_ℓ^H	η_ℓ^ε	$\mu_{\ell,V}^H$	N_{Newton}
0	16	20	-0.2306	0.19320	0.10679	0.22075	0.006584	0.8766	3
1	64	88	-0.2553	0.12913	0.06009	0.14243	0.007179	0.5871	2
2	256	368	-0.2658	0.08951	0.03315	0.09545	0.006242	0.4090	12
3	1024	1504	-0.2706	0.06269	0.01783	0.06518	0.005316	0.2879	12
4	4096	6080	-0.2728	0.04412	0.00948	0.04513	0.004326	0.2034	15
5	16384	24448	-0.2739	-	-	-	-	-	16

Table 5.3: Results of calculations with $\omega = (-0.5, 0.5)^2$, $\mathbf{f} = (0.6, 0.6)^T$, $q = 1$, and uniform mesh-refinements: mesh size N_T , space size N_D , energy, error estimators, and number of Newton iterations N_{Newton} .

5.2.2 Active penalization with $\mathbf{f} = (0.6, 0.6)^T$

In this second simulation, we strengthen the applied field and choose $\mathbf{f} = (0.6, 0.6)^T$. Now, the solution is penalized in some regions of ω , see Figure 5.10. A representative solution on a uniform mesh is shown in Figure 5.8 and the divergence of the h -adaptively computed discrete solution \mathbf{m}_8 with $N_D = 16856$ degrees of freedom can be seen in Figure 5.9. We observe again some effects caused by the anisotropy. Note that the penalized regions are those, where the magnetization is well aligned with the easy-axis e_1 . Moreover, we again observe the characteristic edge and corner singularities in the divergence.

The detailed results of the computation with uniform meshes are given in Table 5.3. Figure 5.11, shows error and error estimators for a sequence of uniform meshes. Note that in contrast to the first experiment with soft applied field, we now also plot the square root of the penalty energy $\eta_\ell^\varepsilon = \left(\frac{1}{2\varepsilon} \|(|\mathbf{m}_\ell| - 1)_+\|_{L^2(\omega)}^2\right)^{1/2}$. One major difference to the first experiment is that in the present case

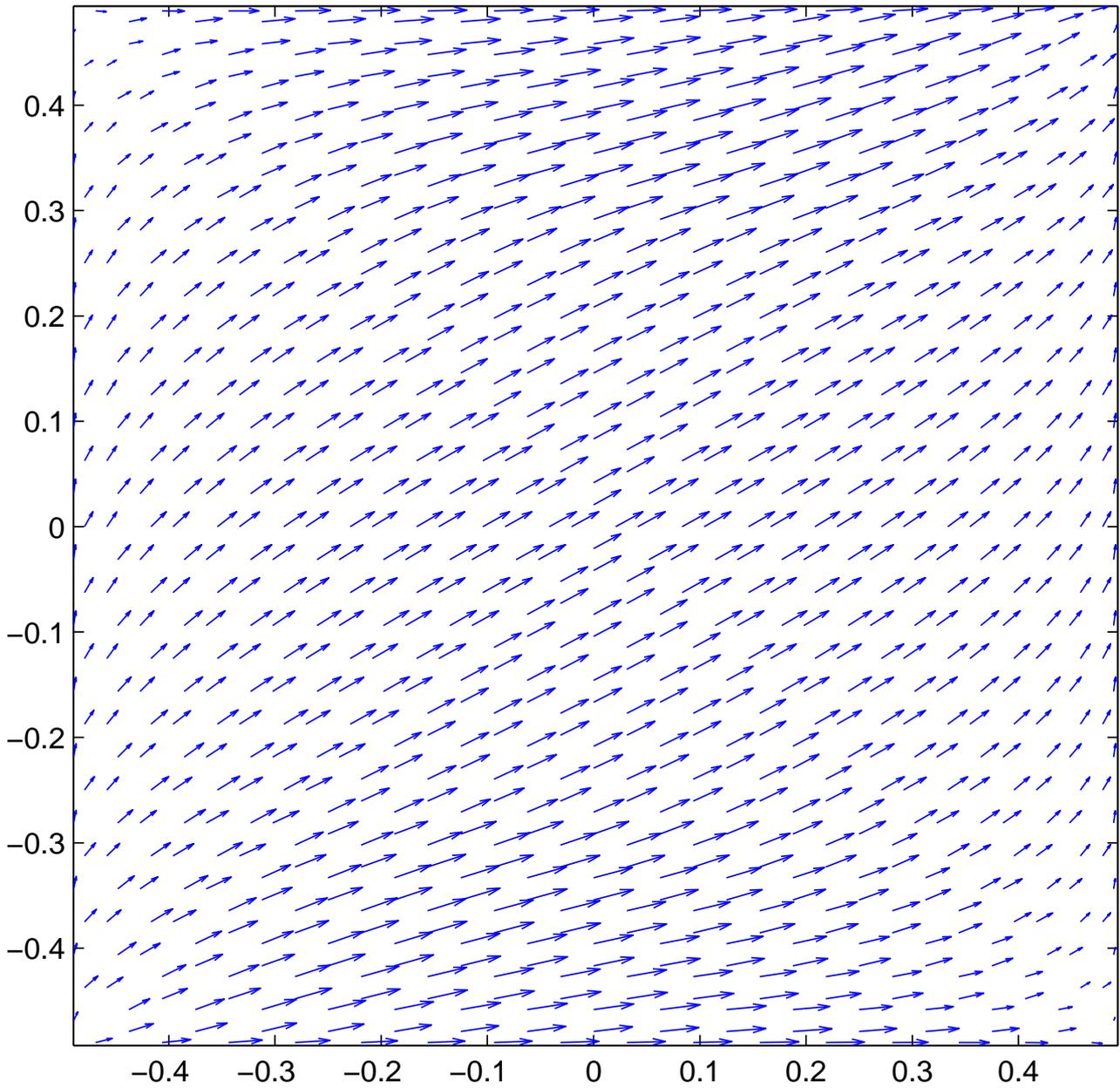


Figure 5.8: Discrete magnetization for $\omega = (-0.5, 0.5)^2$, $\mathbf{f} = (0.6, 0.6)^T$, and $q = 1$ on a mesh with $N_T = 1024$ triangles and $N_D = 1504$ degrees of freedom.

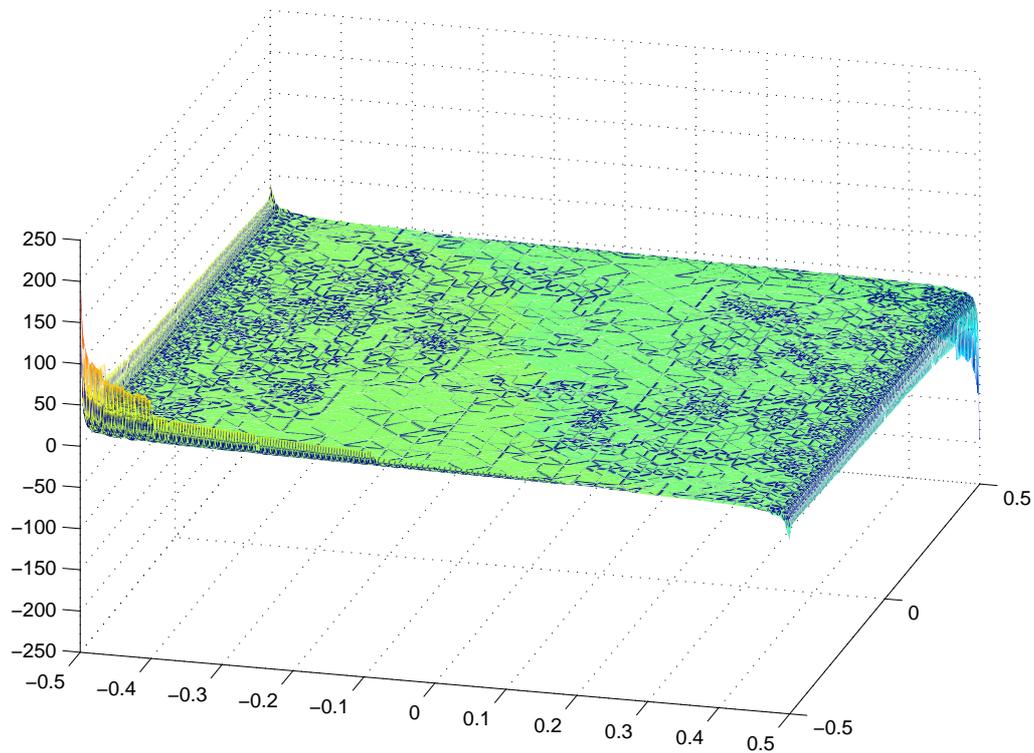


Figure 5.9: Divergence of the solution \mathbf{m}_g for $\omega = (-0.5, 0.5)^2$, $\mathbf{f} = (0.6, 0.6)^T$, and $q = 1$ on an h -adaptively generated mesh with $N_T = 11555$ triangles and $N_D = 16856$ degrees of freedom.

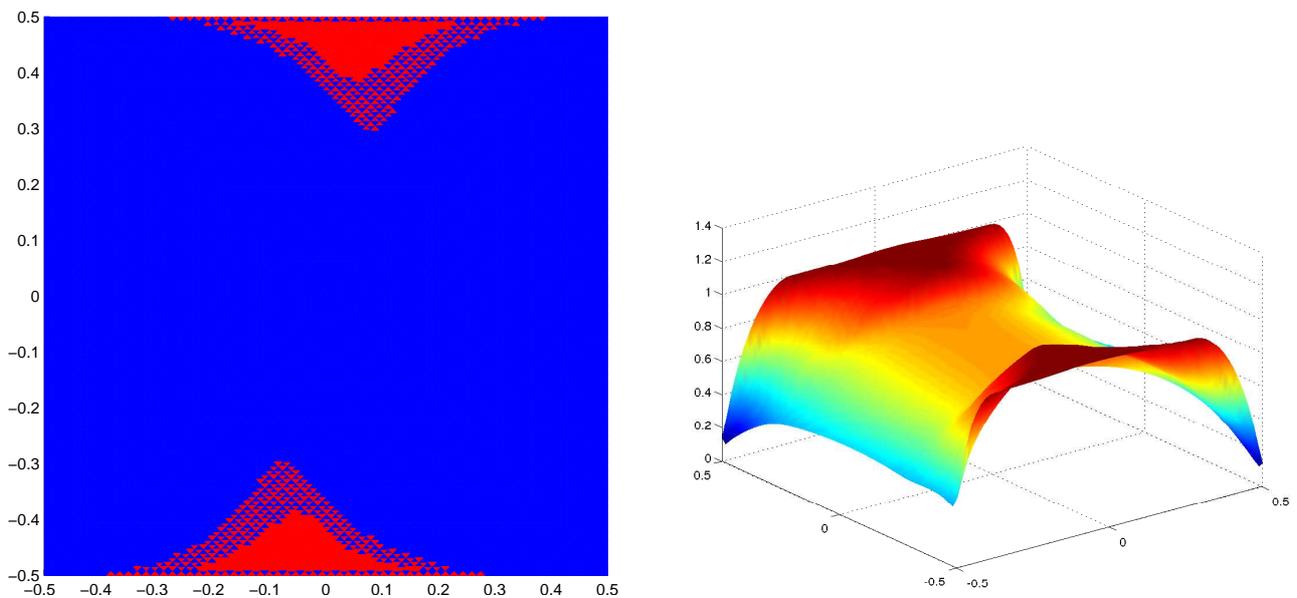


Figure 5.10: Left: Penalized (red) and not penalized (blue) elements of the solution \mathbf{m}_4 for $\omega = (-0.5, 0.5)^2$, $\mathbf{f} = (0.6, 0.6)^T$, and $q = 1$. The uniform mesh has $N_T = 16384$ triangles. Right: Elementwise maximal length of \mathbf{m}_4 .

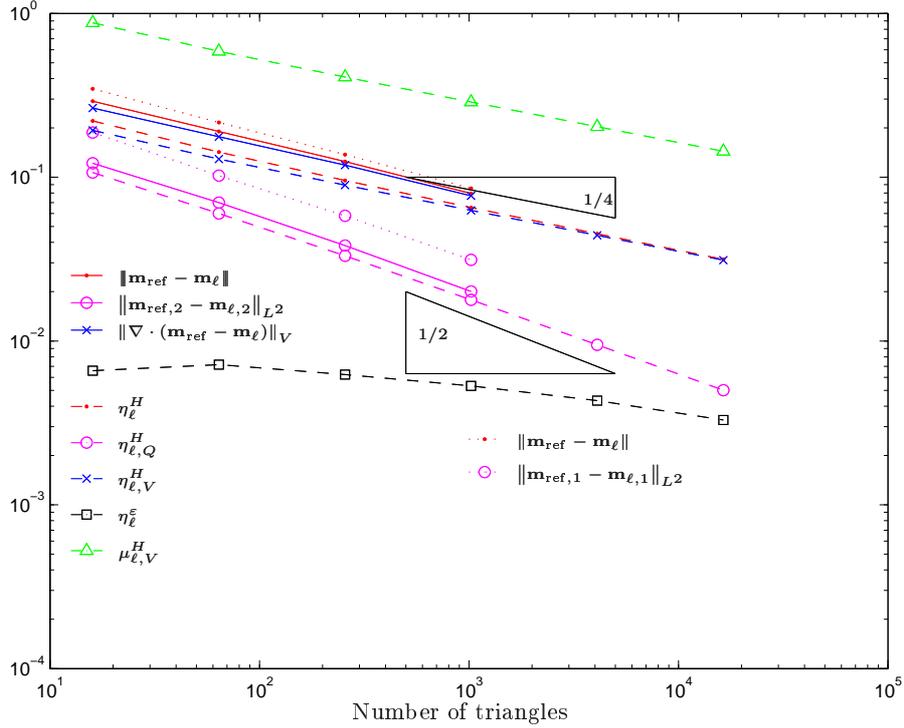


Figure 5.11: Error and error estimators for a sequence of discrete solutions \mathbf{m}_ℓ on uniform meshes for $\omega = (-0.5, 0.5)^2$, $\mathbf{f} = (0.6, 0.6)^T$, and $q = 1$.

we observe convergence in the full space norm $\|\cdot\|$ at the same rate as in the energy norm $\|\!\| \cdot \!\|$. One possible explanation is that in the regions where the magnetization is almost aligned with the easy axis e_1 , the penalty scheme is active and hence the length of the magnetization is restricted there. The second component $\mathbf{m}_{\ell,2}$ is controlled by the energy functional. Therefore, also $\mathbf{m}_{\ell,1} \approx (1 - \mathbf{m}_{\ell,2}^2)^{1/2}$ is determined by quantities that are controlled through the equations directly.

Figure 5.12 shows the sequence of h -adaptively generated meshes. We observe a similar behavior as in the first experiment. Figure 5.13 shows the error estimators in the adaptive computation. For reference, we also plotted the quantities obtained with uniform refinements. At first, the L^2 -contribution of the energy error is significantly smaller than the contribution of the error in the divergence. However, in the last few refinement steps, it becomes relevant and effort is put into refining elements in the interior of ω to reduce the L^2 contribution of the error. We observe that the overall error, again, decays asymptotically at the rate of at least $N_T^{-1/4}$. The results of the adaptive computation are listed in Table 5.4.

The true energy is estimated by extrapolating the sequence of discrete energies obtained with uniform meshes. For our error plots we use the estimate $e(\mathbf{m}^*) \approx -0.274951722507912$. Figure 5.14 shows the estimated error in the energy. Apparently, the adaptive algorithm improves the asymptotic behavior slightly.

5.2.3 Non-smooth solution and h -adaptive algorithm

In a last experiment we demonstrate that the adaptive algorithm is able to improve the order of convergence, even for some non-smooth magnetization \mathbf{m} . To that end, we apply a discontinuous field

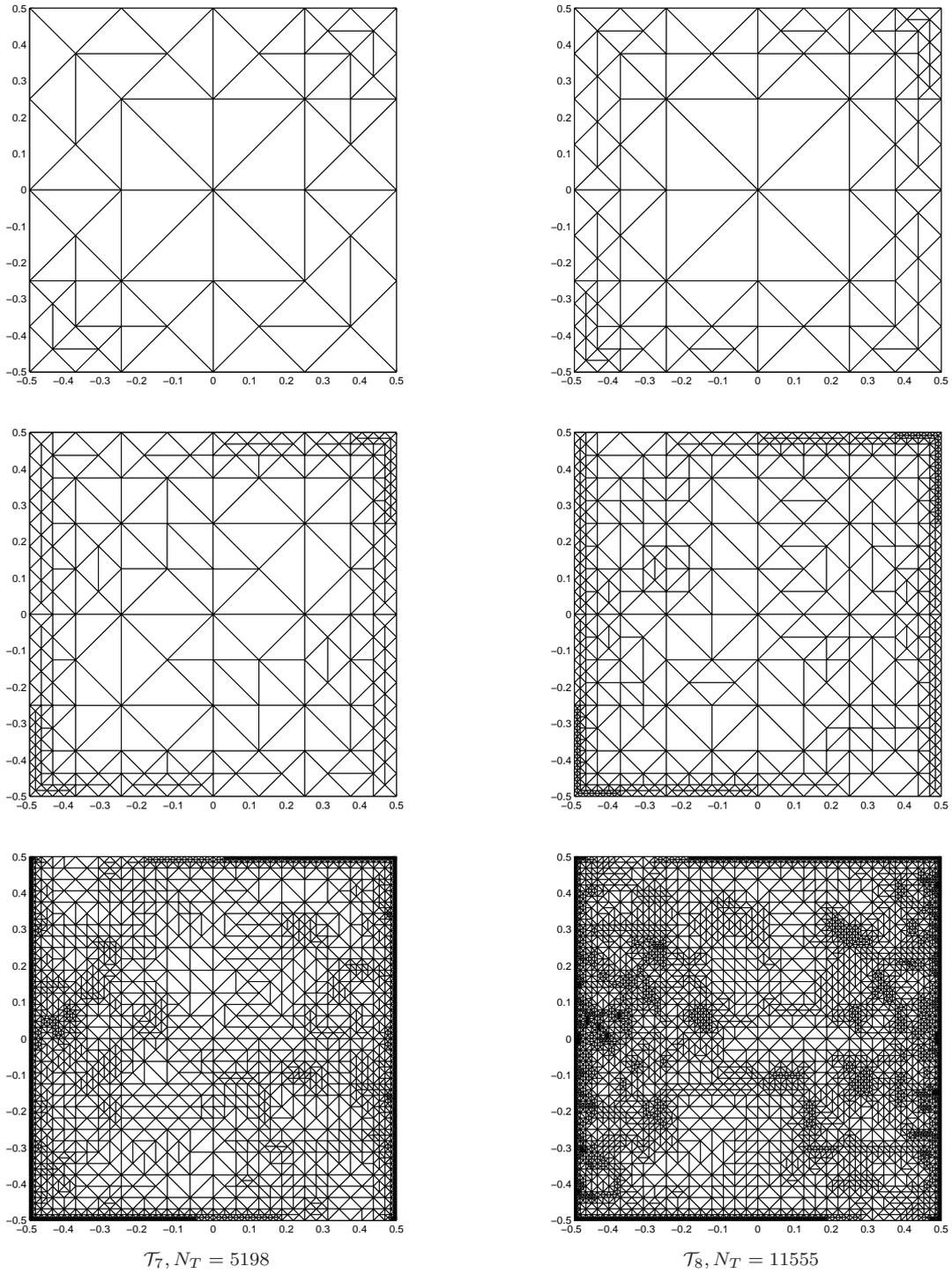


Figure 5.12: Sequence of h -adaptively generated meshes for $\omega = (-0.5, 0.5)^2$, $\mathbf{f} = (0.6, 0.6)^T$, and $q = 1$.

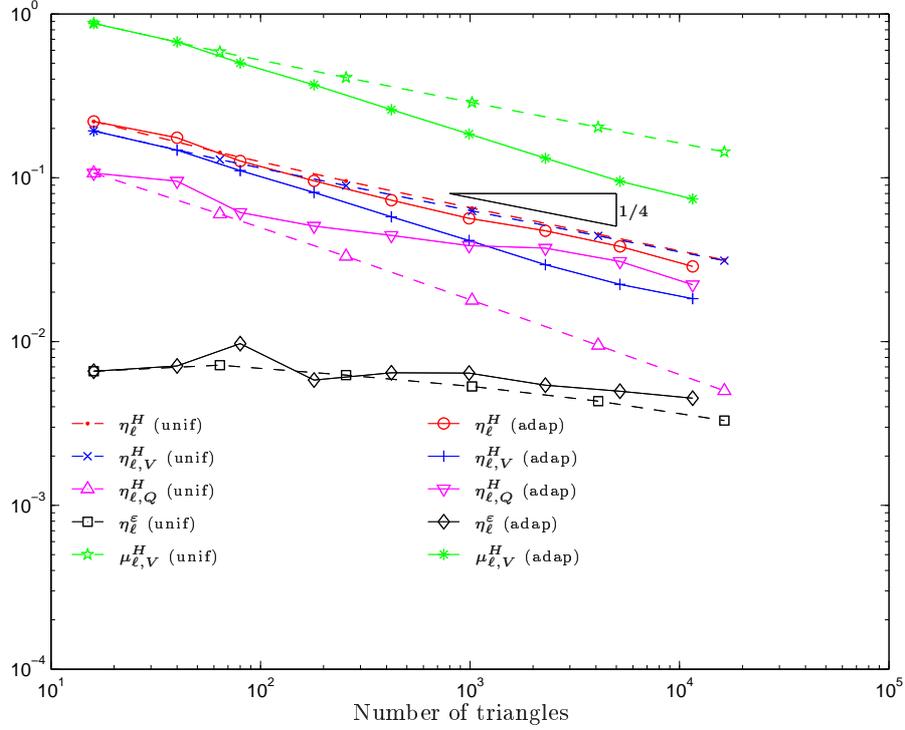


Figure 5.13: Error estimators for h -adaptive and uniform mesh-refinements. The field $\mathbf{f} = (0.6, 0.6)^T$ is applied to the sample $\omega = (0.5, 0.5)^2$ with anisotropy parameter $q = 1$.

Iteration	N_T	N_D	Energy	$\eta_{\ell,V}^H$	$\eta_{\ell,Q}^H$	η_{ℓ}^H	$\eta_{\ell}^{\varepsilon}$	$\mu_{\ell,V}^H$	N_{Newton}
0	16	20	-0.2306	0.19320	0.10679	0.22075	6.58e-3	0.8743	3
1	40	54	-0.2470	0.14747	0.09524	0.17555	7.09e-3	0.6752	4
2	80	110	-0.2599	0.11066	0.06135	0.12653	9.71e-3	0.5015	11
3	180	252	-0.2671	0.08127	0.05064	0.09576	5.81e-3	0.3690	10
4	422	598	-0.2711	0.05776	0.04449	0.07291	6.44e-3	0.2606	12
5	990	1417	-0.2726	0.04132	0.03849	0.05647	6.41e-3	0.1849	12
6	2292	3303	-0.2735	0.02942	0.03717	0.04741	5.40e-3	0.1314	17
7	5198	7540	-0.2743	0.02230	0.03085	0.03807	4.96e-3	0.0950	22
8	11555	16856	-0.2744	0.01825	0.02220	0.02874	4.50e-3	0.0741	31

Table 5.4: Results of calculations with $\omega = (-0.5, 0.5)^2$, $\mathbf{f} = (0.6, 0.6)^T$, and $q = 1$: mesh size N_T , space size N_D , energy, error estimators, and number of Newton iterations N_{Newton} .

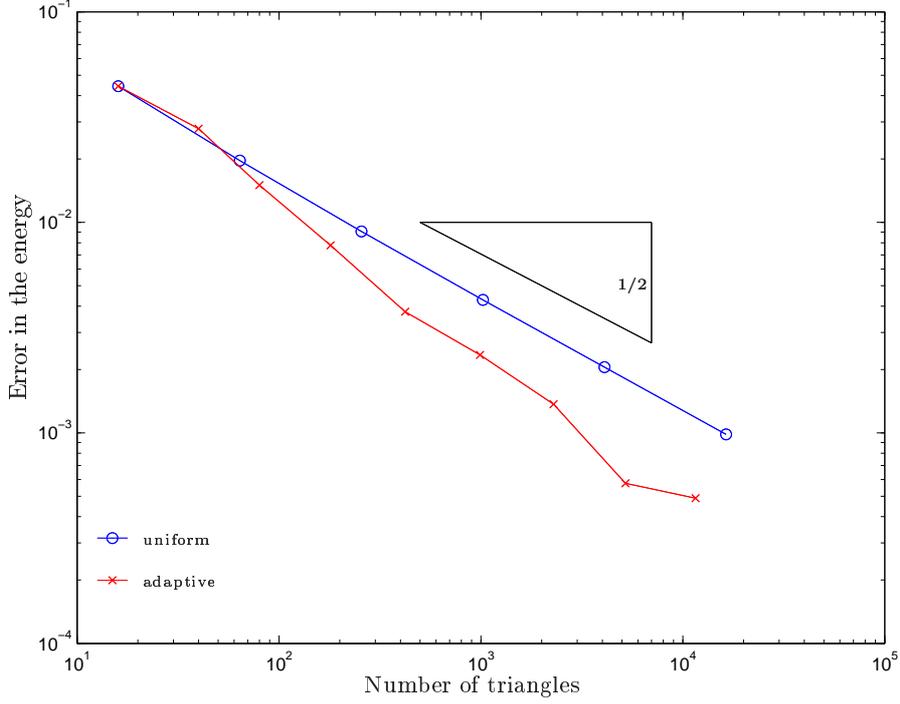


Figure 5.14: Error in the energy $|e(\mathbf{m}^*) - e(\mathbf{m}_\ell)|$ for a sequence of discrete solutions on uniform and h -adaptive meshes for $\omega = (-0.5, 0.5)^2$, $\mathbf{f} = (0.6, 0.6)^T$, and $q = 1$.

\mathbf{f} defined by

$$\mathbf{f}(x, y) = \begin{cases} (0, 1/|x|^{0.1})^T & \text{for } |y| > \sqrt{|x|}, \\ (0, -1) & \text{else,} \end{cases}$$

to our sample $\omega = (-0.5, 0.5)^2$. The anisotropy with $q = 1$ seems to have little effect as the applied field is always orthogonal to the easy axis. The singularity of \mathbf{f} at the origin $(0, 0)$ is very weak. However, the discontinuous behavior seems to be relevant. A solution computed on a uniform mesh with $N_T = 4096$ triangles and a solution computed on an adaptively generated mesh with $N_T = 1760$ triangles can be seen in Figure 5.15 and Figure 5.16, respectively. The divergence of the adaptively computed solution \mathbf{m}_{20} is shown in Figure 5.17. The detailed results of our computation are listed in Table 5.5 and Table 5.6, respectively.

Note that even though we use a relatively small penalty parameter of $\varepsilon = 5e - 3$, we observe some elements at the top and bottom edge, where the magnetization seems to be significantly longer than 1, see Figure 5.18 and Figure 5.19. It seems that the penalization is not equally good everywhere on ω .

The convergence in the energy for uniform meshes seems to be affected by the singular behavior of \mathbf{m} . As an estimate for the true energy, we used the extrapolated value $e(\mathbf{m}_0^\varepsilon) \approx -0.400251416014251$. In the uniform case, we observe a rate of approximately $N_T^{-3/7}$. In contrast, the adaptive algorithm recovers a rate of at least $N_T^{-1/2}$, see Figure 5.20. Estimates in the energy norm reveal a similar behavior as can be seen in Figure 5.21. The error seems to decay at a rate of approximately $N_T^{-3/14}$. This seems to be mainly an effect due to the bad resolution of the L^2 -component of the error.

The h -adaptive algorithm with fixed penalty parameter $\varepsilon = 5e - 3$ effects strong refinements towards

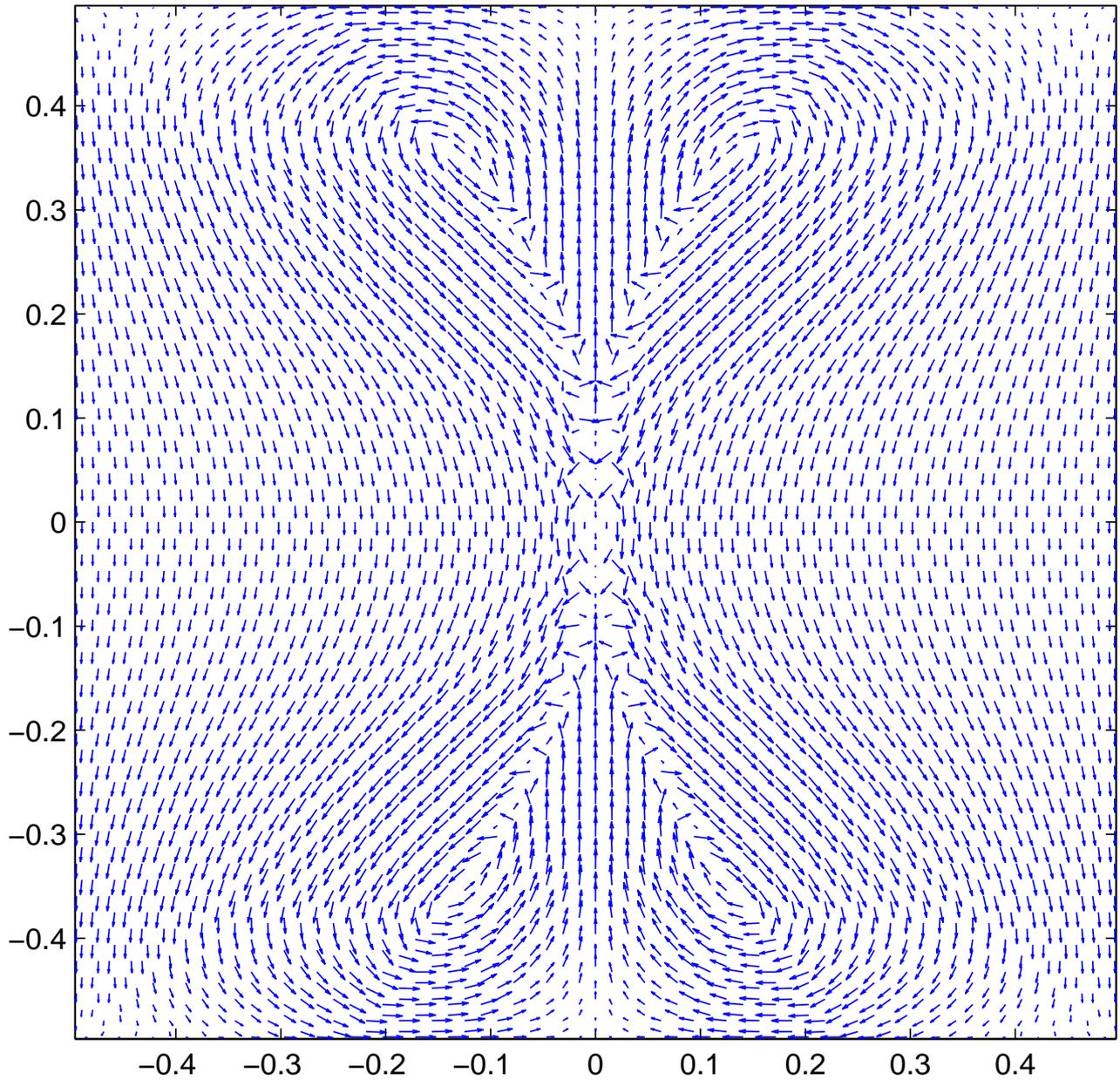


Figure 5.15: Discrete magnetization for $\omega = (-0.5, 0.5)^2$, discontinuous \mathbf{f} , and $q = 1$ on a mesh with $N_T = 4096$ triangles and $N_D = 6080$ degrees of freedom.

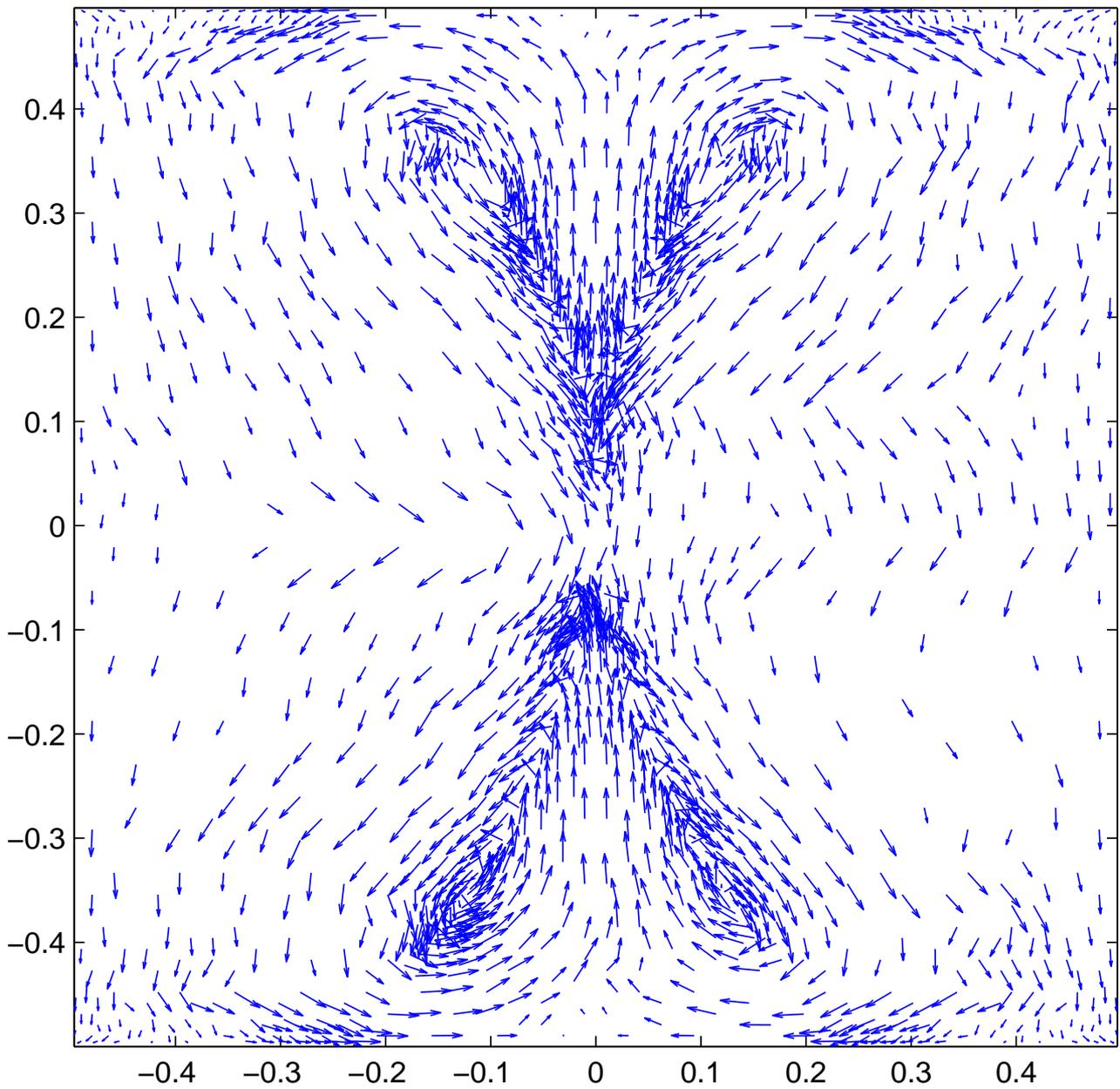


Figure 5.16: Discrete magnetization for $\omega = (-0.5, 0.5)^2$, discontinuous \mathbf{f} , and $q = 1$ on an h -adaptively generated mesh with $N_T = 1760$ triangles and $N_D = 2589$ degrees of freedom.

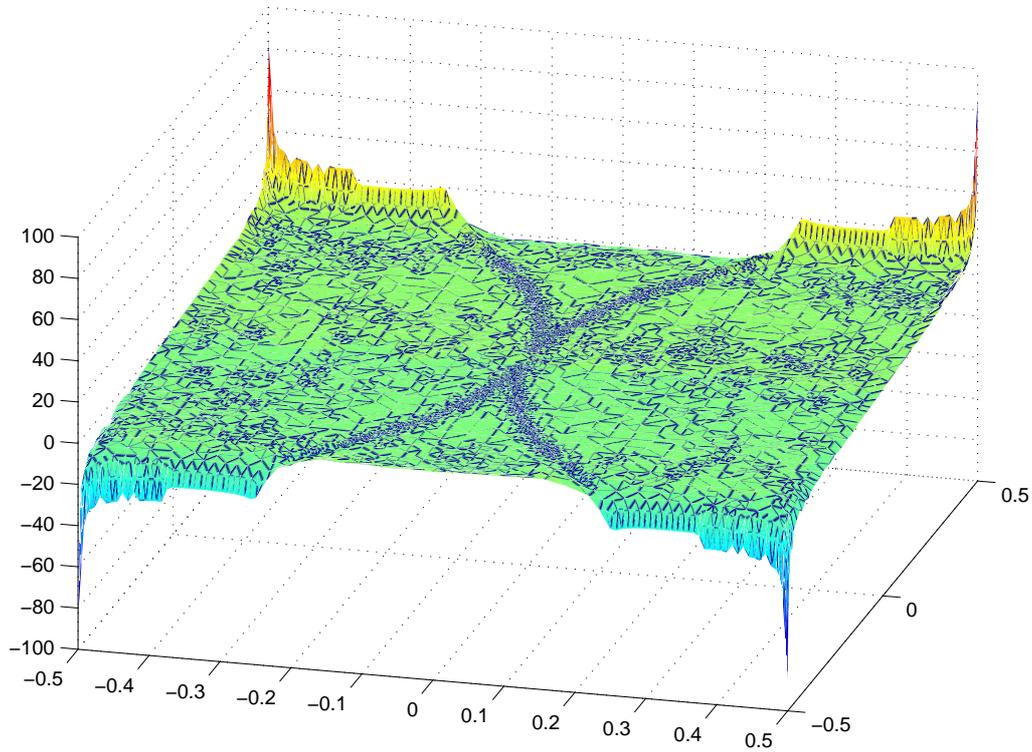


Figure 5.17: Divergence of the discrete solution \mathbf{m}_{20} for $\omega = (-0.5, 0.5)^2$, discontinuous \mathbf{f} , and $q = 1$ on an h -adaptively generated mesh with $N_T = 10485$ triangles and $N_D = 15588$ degrees of freedom.

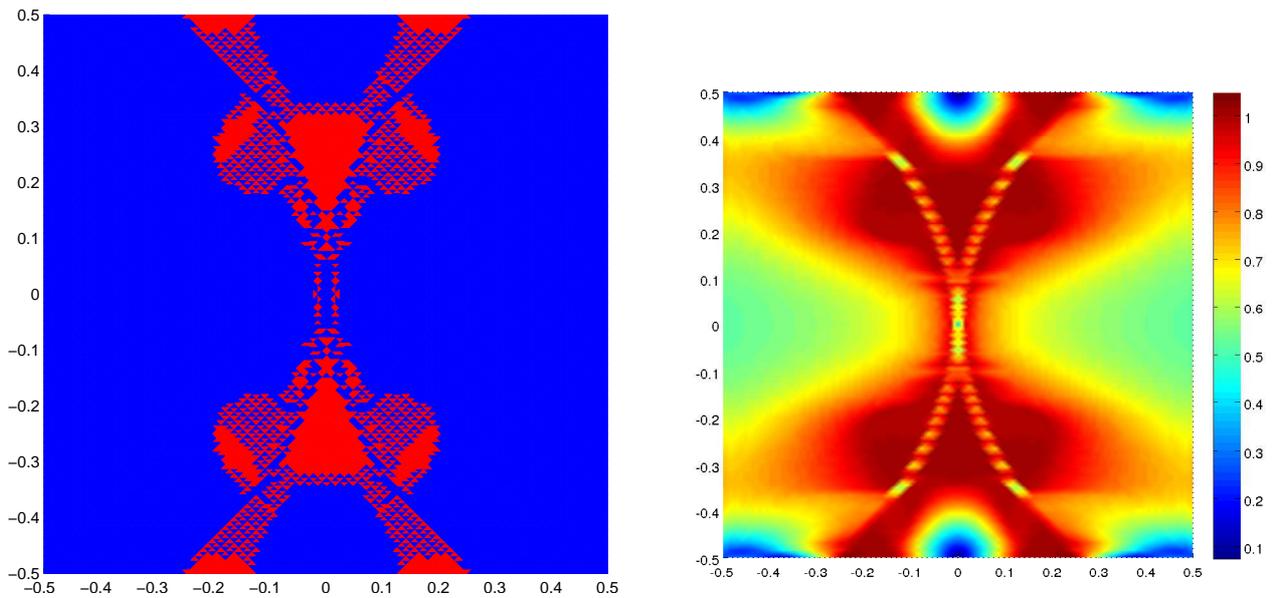


Figure 5.18: Left: Penalized (red) and not penalized (blue) elements of the solution \mathbf{m}_5 for $\omega = (-0.5, 0.5)^2$, discontinuous \mathbf{f} , and $q = 1$. Right: Maximal length of the magnetization \mathbf{m}_5 on each element. The uniform mesh has $N_T = 16384$ triangles and $N_D = 24448$ degrees of freedom.

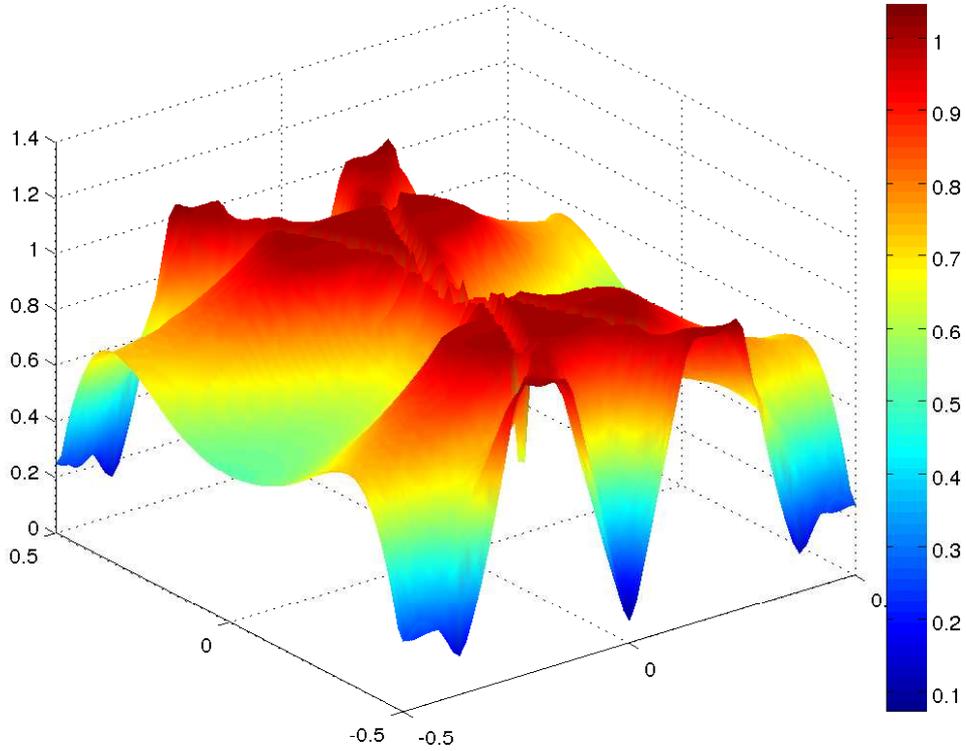


Figure 5.19: Maximal length of the magnetization \mathbf{m}_5 on each element for $\omega = (-0.5, 0.5)^2$, discontinuous \mathbf{f} , and $q = 1$. The uniform mesh has $N_T = 16384$ triangles and $N_D = 24448$ degrees of freedom.

Iteration	N_T	N_D	Energy	$\eta_{\ell,V}^H$	$\eta_{\ell,Q}^H$	η_{ℓ}^H	$\eta_{\ell}^{\varepsilon}$	$\mu_{\ell,V}^H$	N_{Newton}
0	16	20	-0.1915	0.19132	0.3599	0.4076	0	0.7922	2
1	64	88	-0.2913	0.11621	0.2348	0.2620	0.02198	0.5094	11
2	256	368	-0.3271	0.07541	0.1907	0.2051	0.02121	0.3361	17
3	1024	1504	-0.3624	0.05286	0.1427	0.1521	0.02243	0.2312	24
4	4096	6080	-0.3797	0.03675	0.1080	0.1141	0.02271	0.1579	37
5	16384	24448	-0.3891	0.02425	0.0797	0.0833	0.02256	0.1082	43
6	65536	98048	-0.3937	-	-	-	-	-	42

Table 5.5: Results of calculations with $\omega = (-0.5, 0.5)^2$, discontinuous \mathbf{f} , $q = 1$, and uniform mesh-refinements: mesh size N_T , space size N_D , energy, error estimators, and number of Newton iterations N_{Newton} .

Iteration	N_T	N_D	Energy	$\eta_{\ell,V}^H$	$\eta_{\ell,Q}^H$	η_{ℓ}^H	$\eta_{\ell}^{\varepsilon}$	$\mu_{\ell,V}^H$	N_{Newton}
0	16	20	-0.1915	0.19139	0.35996	0.40768	0	0.7922	2
1	32	42	-0.2801	0.15583	0.26168	0.30457	0.01620	0.7134	16
2	56	74	-0.3035	0.11218	0.23590	0.26122	0.01808	0.5094	13
3	80	110	-0.3350	0.1147	0.20386	0.23395	0.02115	0.4978	20
4	116	164	-0.3369	0.10650	0.17557	0.20535	0.01772	0.4786	14
5	184	258	-0.3416	0.08312	0.16490	0.18466	0.01872	0.3612	20
6	297	426	-0.3434	0.07866	0.16235	0.18040	0.01930	0.3463	20
7	403	585	-0.3529	0.07344	0.14630	0.16370	0.01994	0.3312	15
8	587	860	-0.3659	0.07157	0.12440	0.14352	0.02021	0.3249	16
9	774	1132	-0.3725	0.05923	0.11984	0.13368	0.02254	0.2628	14
10	1006	1477	-0.3784	0.05392	0.10384	0.11701	0.02318	0.2420	17
11	1327	1950	-0.3814	0.04674	0.09484	0.10574	0.02368	0.2060	28
12	1760	2589	-0.3839	0.04278	0.09367	0.10298	0.02416	0.1853	18
13	2219	3274	-0.3849	0.03954	0.08850	0.09693	0.02374	0.1750	27
14	2910	4303	-0.3866	0.03714	0.08227	0.09026	0.02391	0.1614	17
15	3687	5460	-0.3888	0.03405	0.07550	0.08282	0.02383	0.1478	20
16	4668	6923	-0.3916	0.03203	0.07352	0.08019	0.02357	0.1402	39
17	5499	8147	-0.3919	0.02863	0.06961	0.07527	0.02358	0.1235	21
18	7073	10498	-0.3937	0.02675	0.06773	0.07282	0.02269	0.1162	25
19	8441	12538	-0.3944	0.02530	0.06576	0.07046	0.02263	0.1100	30
20	10485	15588	-0.3949	0.02379	0.06143	0.06587	0.02241	0.1028	18

Table 5.6: Results of calculations with $\omega = (-0.5, 0.5)^2$, discontinuous \mathbf{f} , $q = 1$, and h -adaptive mesh-refinements: mesh size N_T , space size N_D , energy, error estimators, and number of Newton iterations N_{Newton} .

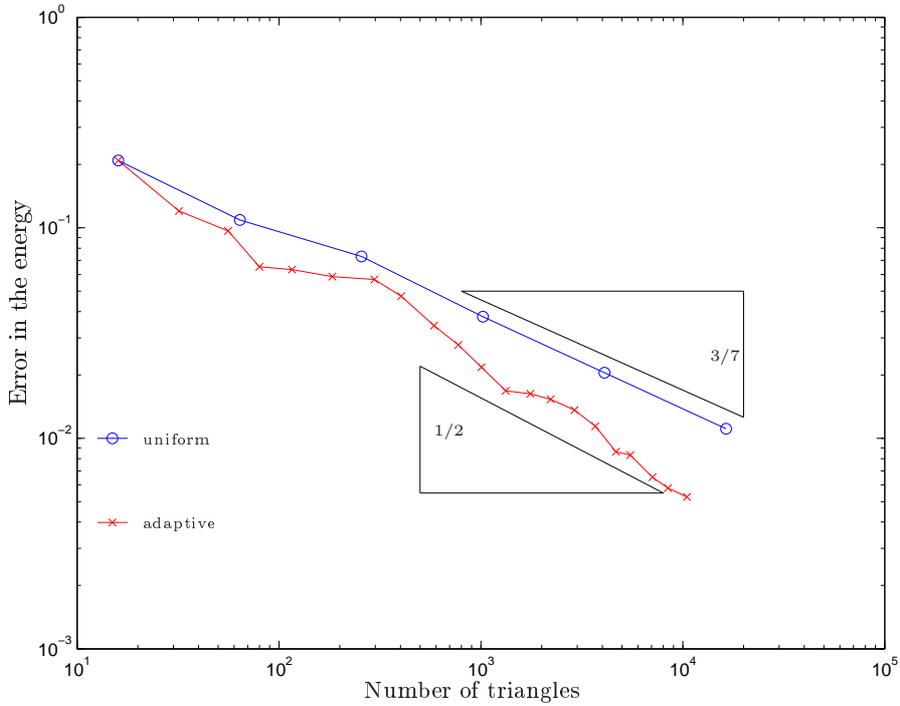


Figure 5.20: Error in the energy $|e(\mathbf{m}^*) - e(\mathbf{m}_{\ell})|$ for a sequence of discrete solutions on uniform and h -adaptive meshes for $\omega = (-0.5, 0.5)^2$, discontinuous \mathbf{f} , and $q = 1$.

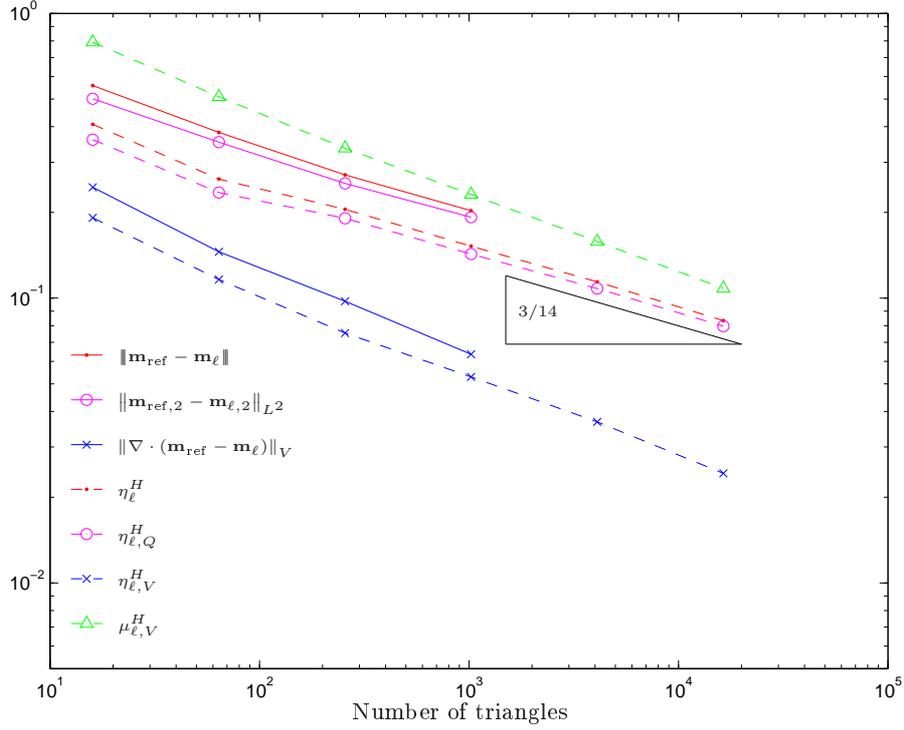


Figure 5.21: Error and error estimators for a sequence of discrete solutions \mathbf{m}_ℓ on uniform meshes for $\omega = (-0.5, 0.5)^2$, discontinuous \mathbf{f} , and $q = 1$.

the discontinuity lines of \mathbf{f} . The singularity of \mathbf{f} at the origin $(0, 0)$ seems to be less significant. A possible explanation is that it is a weak singularity that only effects a very small region of ω . A selection of adaptively generated meshes can be seen in Figure 5.22. The error estimators plotted in Figure 5.23 reveal that the rate of $N_T^{-1/4}$ is recovered with the help of the adaptive mesh-refinements.

5.2.4 Influence of ε

It remains to study experimentally the dependence of the discretization error from the penalty parameter ε . We choose the applied field $\mathbf{f} = (0.8, 0.8)^T$. The penalty region of a solution computed with $\varepsilon = 0.005$ on a uniform mesh with $N_T = 65536$ triangles is shown together with the elementwise maximal length of the magnetization in Figure 5.24. We can clearly see that the penalization term is active. Moreover, in the last refinement step, the error estimators η_ℓ^H and η_ℓ^ε have values close to each other for coarse ε .

We run four h -uniform simulations with penalty parameter values of $\varepsilon \in \{0.1, 0.05, 0.01, 0.005\}$. For each simulation run, we compute a reference solution $\mathbf{m}_{ref}^\varepsilon$ on a mesh with $N_T = 65536$ triangles. Figure 5.25 shows the error $\|\mathbf{m}_{ref}^\varepsilon - \mathbf{m}_\ell^\varepsilon\|$ for various choices of ε . The plot is difficult to read because the lines almost coincide. The Tables 5.7–5.9 show the detailed results of the simulations. It seems that the discretization error is basically independent of the penalty error in this case.

In a second simulation run, we chose a very strong applied field $\mathbf{f} = (3, 3)^T$ such that the error from the penalization is dominant. Figure 5.26 shows the error for the parameter values of $\varepsilon \in \{0.1, 0.01\}$. The reduction of the penalty parameter by an order of magnitude leads to a slight improvement of the discretization error in this example. The results of the simulation are documented in Tables 5.11–5.12.

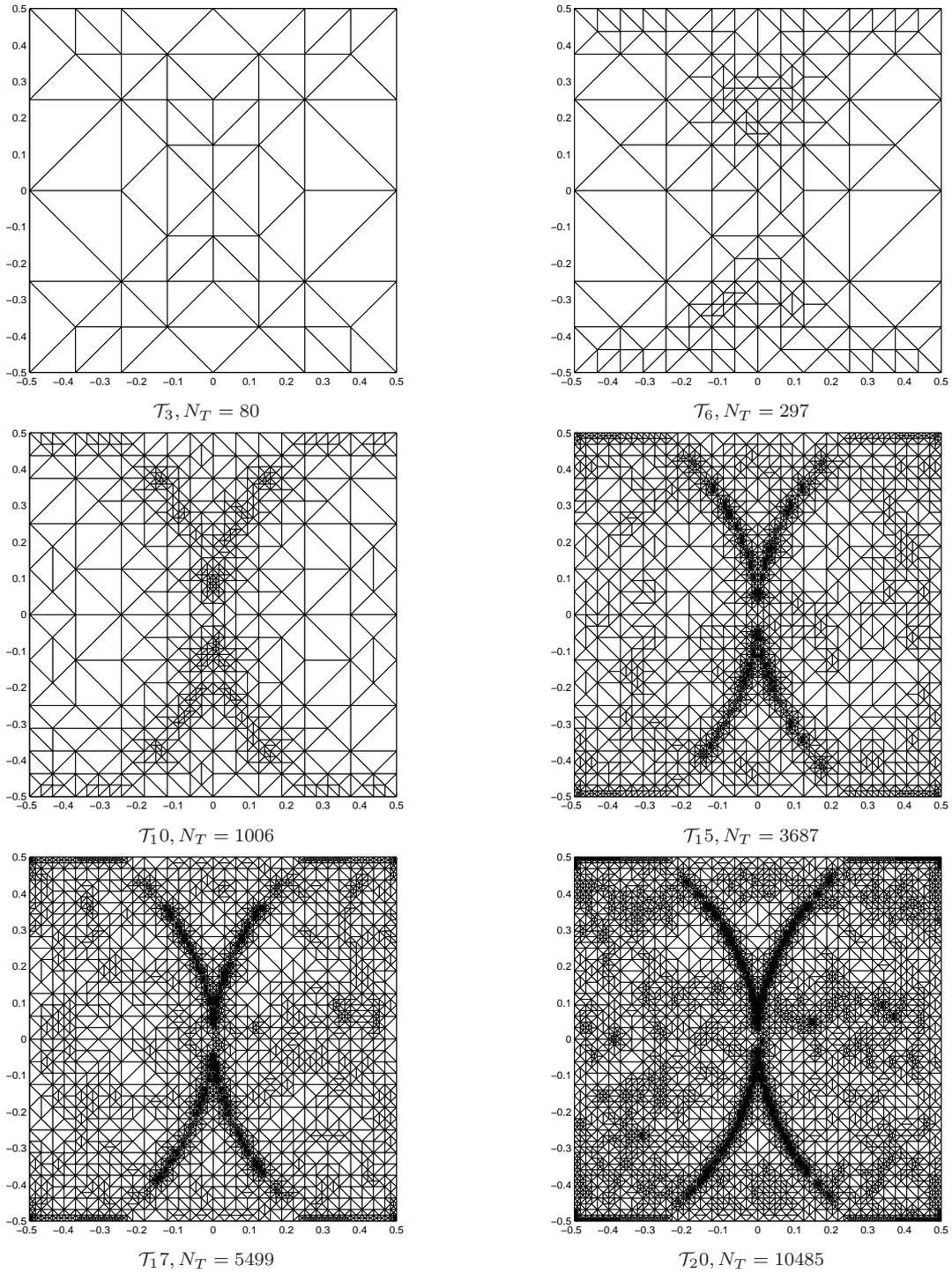


Figure 5.22: Sequence of h -adaptively generated meshes for $\omega = (-0.5, 0.5)^2$, discontinuous \mathbf{f} , and $q = 1$.

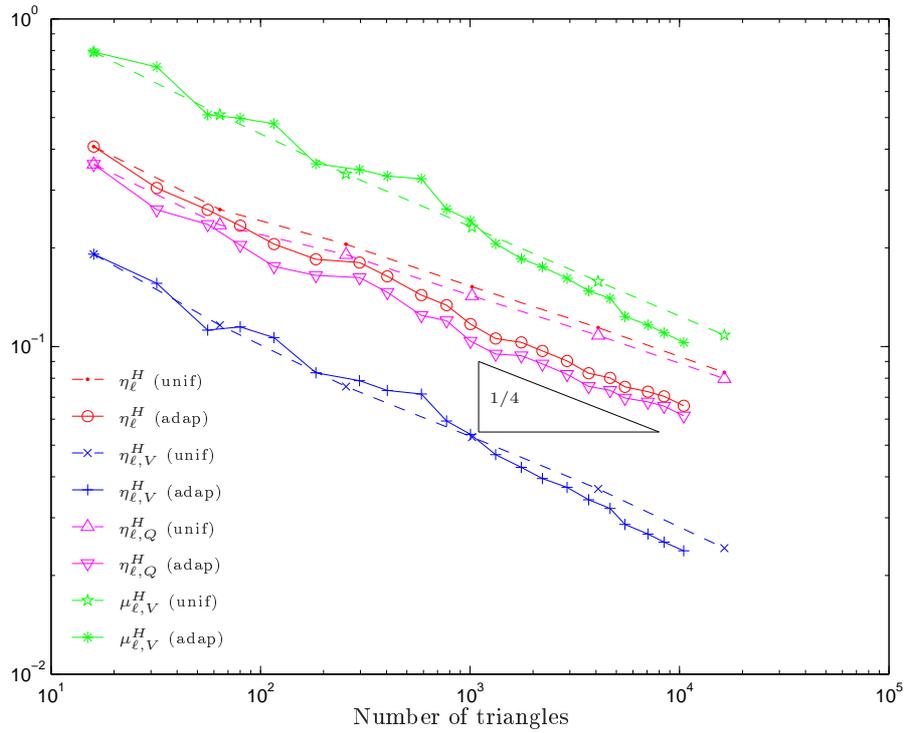


Figure 5.23: Error estimators for h -adaptive and uniform mesh-refinements. A discontinuous field \mathbf{f} is applied to the sample $\omega = (0.5, 0.5)^2$ with anisotropy parameter $q = 1$.

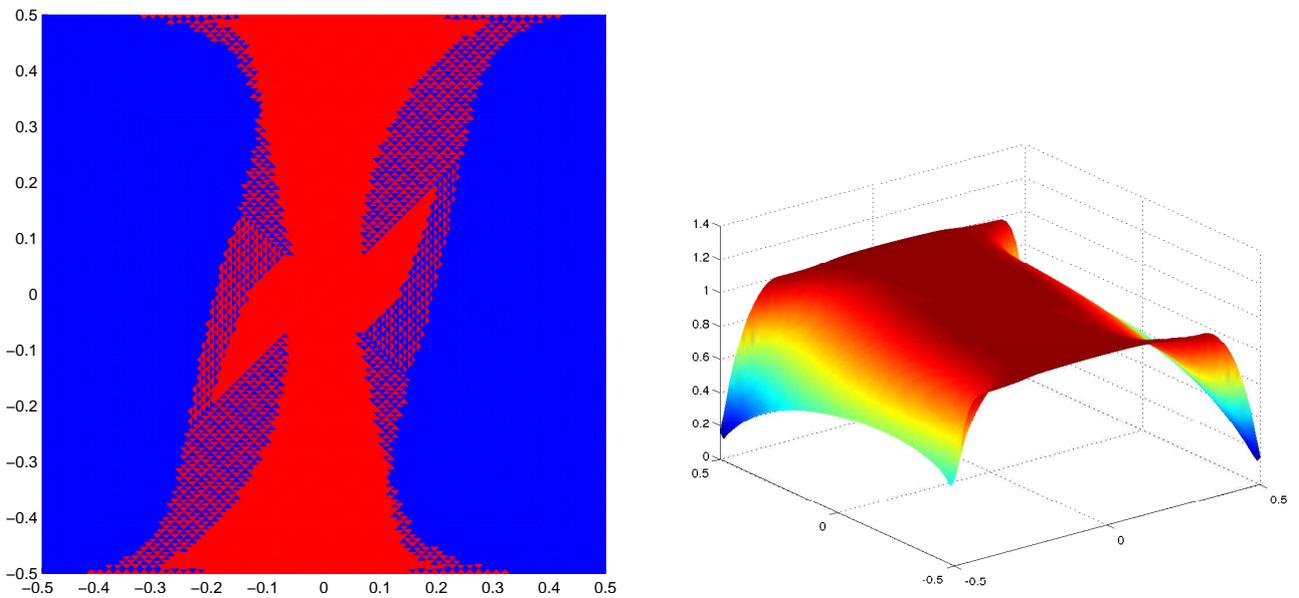


Figure 5.24: Left: penalized (red) and not penalized (blue) elements of a uniform mesh of $\omega = (-0.5, 0.5)^2$ with $N_T = 65536$ triangles. The penalty parameter is $\varepsilon = 0.005$, the applied field is $\mathbf{f} = (0.8, 0.8)^T$. Right: elementwise maximal length of the discrete magnetization.

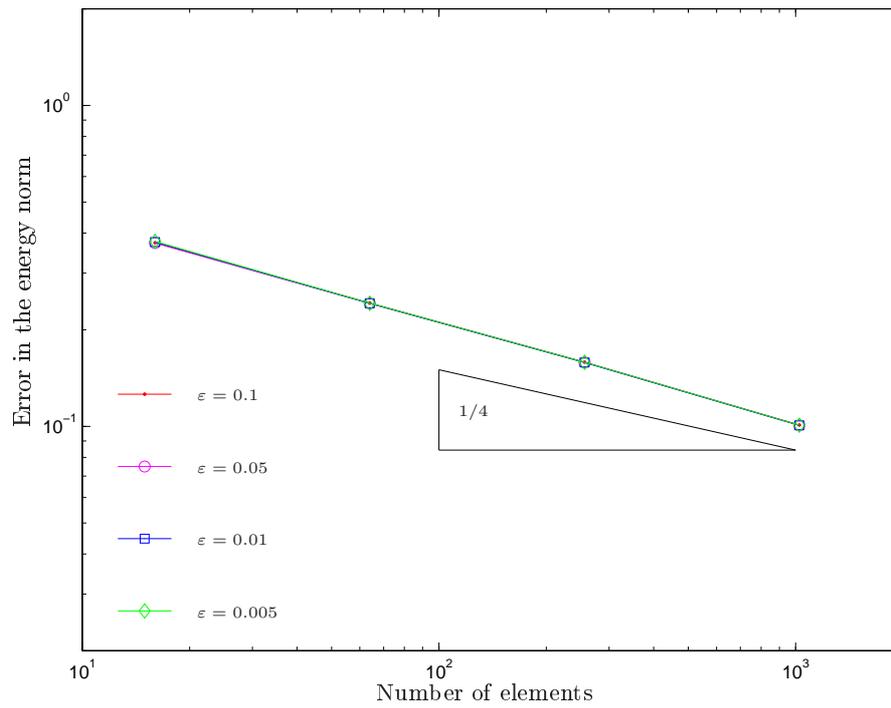


Figure 5.25: Error in the energy norm for a series of experiments with uniform meshes of $\omega = (-0.5, 0.5)^2$, applied field $\mathbf{f} = (0.8, 0.8)^T$, $q = 1$ and fixed values of ε .

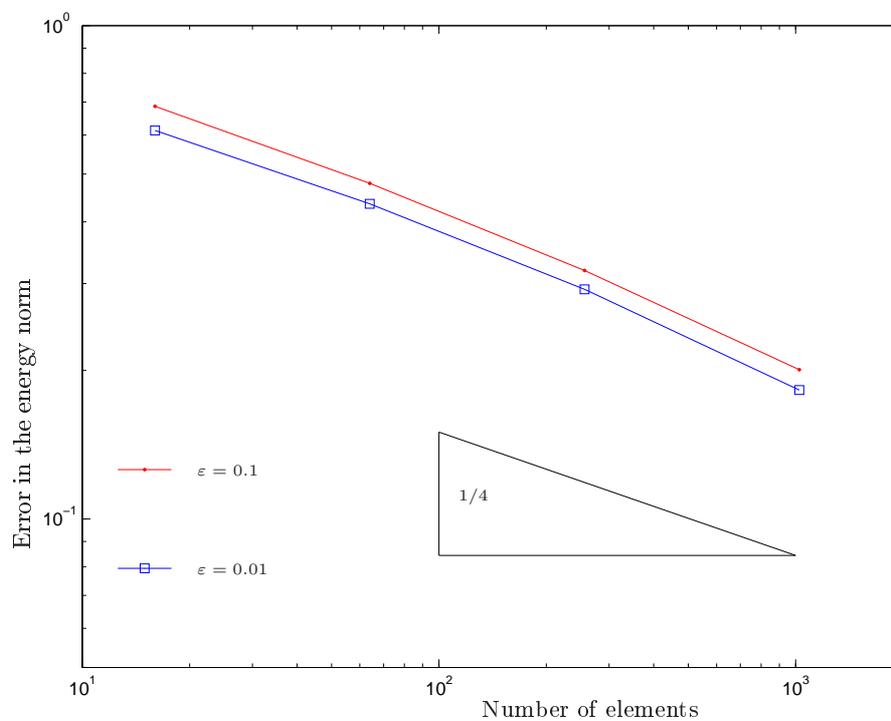


Figure 5.26: Error in the energy norm for a series of experiments with uniform meshes of $\omega = (-0.5, 0.5)^2$, applied field $\mathbf{f} = (3, 3)^T$, $q = 1$ and fixed values of ε .

Iteration	N_T	N_D	Energy	η_ℓ^H	η_ℓ^ε	$\ \mathbf{m}_{ref}^\varepsilon - \mathbf{m}_\ell^\varepsilon\ $	N_{Newton}
0	16	20	-0.4081	0.2824	0.0546	0.3732	6
1	64	88	-0.4512	0.1825	0.0388	0.2420	6
2	256	368	-0.4691	0.1215	0.0341	0.1584	6
3	1024	1504	-0.4770	0.0828	0.0308	0.1009	6
4	4096	6080	-0.4806	0.0572	0.0288	0.0572	7
5	16384	24448	-0.4823	0.0399	0.0279	-	6

Table 5.7: Results of calculations with $\omega = (-0.5, 0.5)^2$, $\mathbf{f} = (0.8, 0.8)^T$, $q = 1$, uniform mesh-refinements, and $\varepsilon = 0.1$: mesh size N_T , space size N_D , energy, error estimators, and number of Newton iterations N_{Newton} .

Iteration	N_T	N_D	Energy	η_ℓ^H	η_ℓ^ε	$\ \mathbf{m}_{ref}^\varepsilon - \mathbf{m}_\ell^\varepsilon\ $	N_{Newton}
0	16	20	-0.4071	0.2806	0.0624	0.3728	11
1	64	88	-0.4505	0.1819	0.0400	0.2416	6
2	256	368	-0.4686	0.1212	0.0319	0.1581	7
3	1024	1504	-0.4766	0.0825	0.0272	0.1007	7
4	4096	6080	-0.4802	0.0571	0.0238	0.0571	8
5	16384	24448	-0.4819	0.0398	0.0219	-	7

Table 5.8: Results of calculations with $\omega = (-0.5, 0.5)^2$, $\mathbf{f} = (0.8, 0.8)^T$, $q = 1$, uniform mesh-refinements, and $\varepsilon = 0.05$: mesh size N_T , space size N_D , energy, error estimators, and number of Newton iterations N_{Newton} .

Iteration	N_T	N_D	Energy	η_ℓ^H	η_ℓ^ε	$\ \mathbf{m}_{ref}^\varepsilon - \mathbf{m}_\ell^\varepsilon\ $	N_{Newton}
0	16	20	-0.4034	0.2785	0.0577	0.3752	11
1	64	88	-0.4490	0.1814	0.0380	0.2416	9
2	256	368	-0.4675	0.1209	0.0278	0.1581	9
3	1024	1504	-0.4758	0.0824	0.0213	0.1006	13
4	4096	6080	-0.4797	0.0570	0.0172	0.0570	10
5	16384	24448	-0.4815	0.0397	0.0142	-	10

Table 5.9: Results of calculations with $\omega = (-0.5, 0.5)^2$, $\mathbf{f} = (0.8, 0.8)^T$, $q = 1$, uniform mesh-refinements, and $\varepsilon = 0.01$: mesh size N_T , space size N_D , energy, error estimators, and number of Newton iterations N_{Newton} .

Iteration	N_T	N_D	Energy	η_ℓ^H	η_ℓ^ε	$\ \mathbf{m}_{ref}^\varepsilon - \mathbf{m}_\ell^\varepsilon\ $	N_{Newton}
0	16	20	-0.4019	0.2791	0.0461	0.3769	12
1	64	88	-0.4483	0.1814	0.0310	0.2419	11
2	256	368	-0.4672	0.1211	0.0260	0.1583	10
3	1024	1504	-0.4756	0.0825	0.0196	0.1008	11
4	4096	6080	-0.4796	0.0570	0.0153	0.0570	13
5	16384	24448	-0.4814	0.0398	0.0124	-	12

Table 5.10: Results of calculations with $\omega = (-0.5, 0.5)^2$, $\mathbf{f} = (0.8, 0.8)^T$, $q = 1$, uniform mesh-refinements, and $\varepsilon = 0.005$: mesh size N_T , space size N_D , energy, error estimators, and number of Newton iterations N_{Newton} .

Iteration	N_T	N_D	Energy	η_ℓ^H	η_ℓ^ε	$\ \mathbf{m}_{ref}^\varepsilon - \mathbf{m}_\ell^\varepsilon\ $	N_{Newton}
0	16	20	-3.5290	0.4756	0.7434	0.686183	7
1	64	88	-3.8293	0.3439	0.7459	0.478808	6
2	256	368	-3.9509	0.2405	0.7445	0.318782	5
3	1024	1504	-3.9993	0.1639	0.7441	0.200693	5
4	4096	6080	-4.0191	0.1124	0.7445	0.112477	5
5	16384	24448	-4.0277	0.0781	0.7450	-	4

Table 5.11: Results of calculations with $\omega = (-0.5, 0.5)^2$, $\mathbf{f} = (3, 3)^T$, $q = 1$, uniform mesh-refinements, and $\varepsilon = 0.1$: mesh size N_T , space size N_D , energy, error estimators, and number of Newton iterations N_{Newton} .

Iteration	N_T	N_D	Energy	η_ℓ^H	η_ℓ^ε	$\ \mathbf{m}_{ref}^\varepsilon - \mathbf{m}_\ell^\varepsilon\ $	N_{Newton}
0	16	20	-2.9089	0.4025	0.4519	0.613074	9
1	64	88	-3.2395	0.2986	0.3951	0.435184	8
2	256	368	-3.3772	0.2122	0.3398	0.292067	7
3	1024	1504	-3.4338	0.1463	0.3007	0.182514	7
4	4096	6080	-3.4565	0.1008	0.2790	0.100889	7
5	16384	24448	-3.4654	0.0701	0.2681	-	7

Table 5.12: Results of calculations with $\omega = (-0.5, 0.5)^2$, $\mathbf{f} = (3, 3)^T$, $q = 1$, uniform mesh-refinements, and $\varepsilon = 0.01$: mesh size N_T , space size N_D , energy, error estimators, and number of Newton iterations N_{Newton} .

5.3 Experimental analysis of penalty error

The error introduced by the penalty scheme is of a different quality than the error due to the discretization. We expect that the discretization error depends on the smoothness of the analytical solution. In Theorem 3.22, we assumed $\nabla \cdot \mathbf{m}^* \in L^2$ to prove the convergence rate of

$$\|\mathbf{m}^* - \mathbf{m}_0^\varepsilon\| = \mathcal{O}(\sqrt{\varepsilon}).$$

Note that the proof only needs the regularity of $\nabla \cdot \mathbf{m}^*$ because we used pointwise estimates that yield $\|\nabla \cdot \mathbf{m}\|_{L^2}$ in the upper bound of the error estimation. There is, however, no necessity to apply any interpolation operator to obtain the result. One may therefore suspect that a different strategy in the proof might yield the same result with weaker regularity assumptions.

5.3.1 Convergence and influence of h

Nonetheless, we restrict ourselves to a simple and smooth setting. The simulation domain is, again, the unit square centered in the plane, i.e. $\omega = (-0.5, 0.5)^2$. We apply the constant field $\mathbf{f} = (1, 1)^T$. In the preceding experiments concerning the h -convergence we also chose the rather weak applied fields because we wanted the penalty error to be relatively small when compared to the discretization error. Now, we are particularly interested in the penalty error and the behavior of our estimator η_ℓ^ε . We therefore stick to the choice of strong \mathbf{f} . Figure 5.27 shows the penalized region for a uniform mesh with $N_T = 4096$ triangles and the penalty parameter of $\varepsilon = 1e - 4$. The maximal length of the magnetization on each triangle is plotted in Figure 5.28. The anisotropy parameter is set to $q = 1$.

We choose fixed meshes \mathcal{T}_1 with $N_T = 256$, \mathcal{T}_2 with $N_T = 1024$, and \mathcal{T}_3 with $N_T = 4096$ triangles.

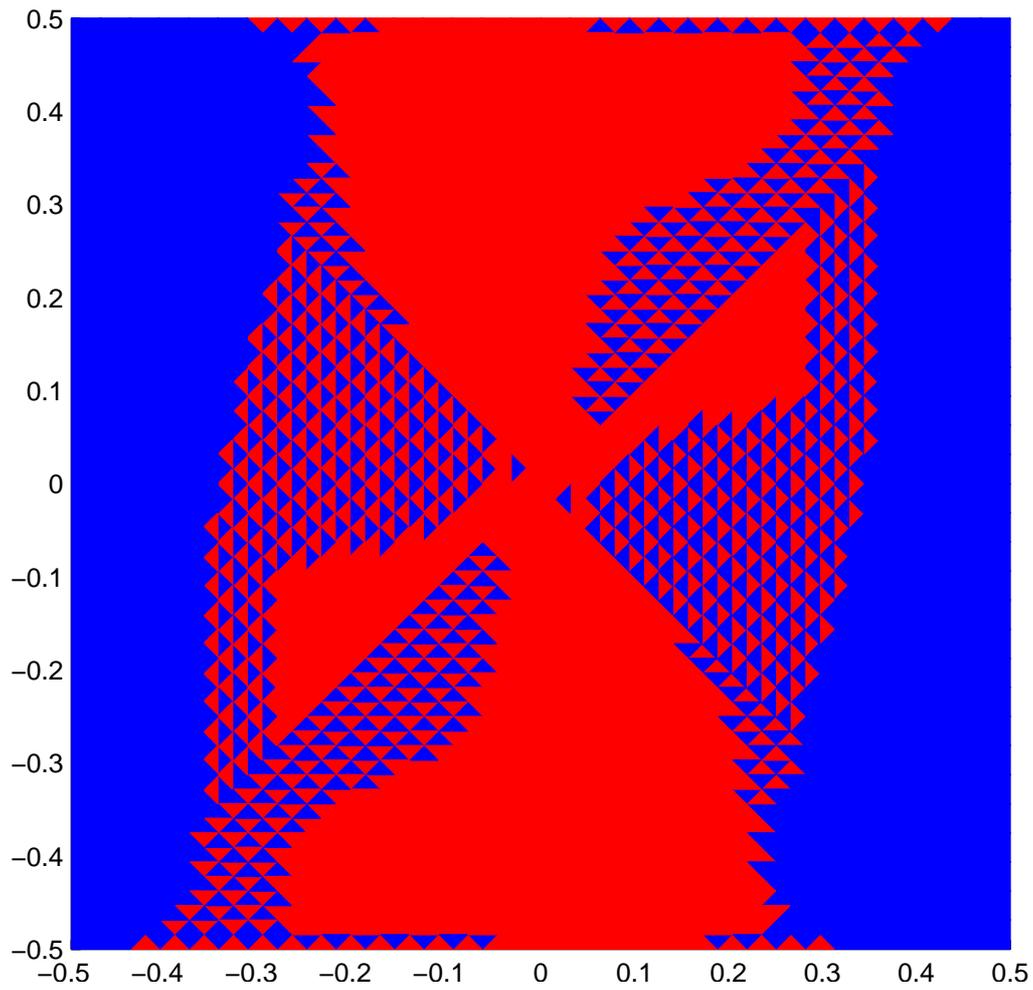


Figure 5.27: Penalized (red) and not penalized (blue) elements of the solution \mathbf{m}_3 for $\omega = (-0.5, 0.5)^2$, $\mathbf{f} = (1, 1)^T$, $\varepsilon = 1e - 4$, and $q = 1$. The uniform mesh has $N_T = 4096$ triangles.

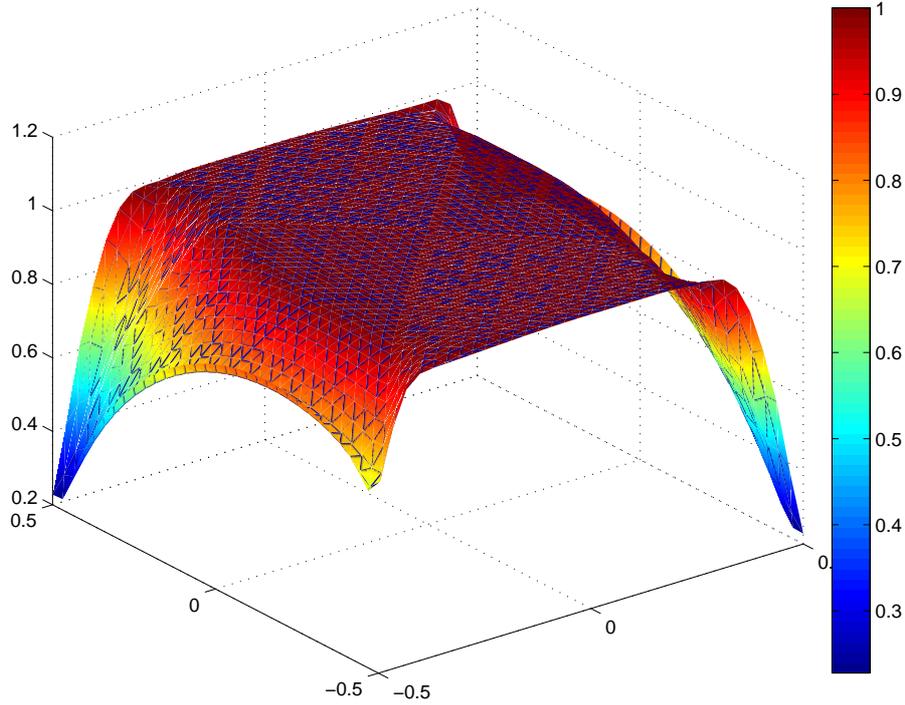


Figure 5.28: Length of the discrete solution \mathbf{m}_3 on a uniform mesh of $\omega = (-0.5, 0.5)^2$ with $N_T = 4096$ triangles, $\mathbf{f} = (1, 1)^T$, $q = 1$, and $\varepsilon = 1e - 4$.

Then, we compute the discrete solutions with $\varepsilon = 0.5^k \cdot 0.0625$ for $k = 0, 1, \dots$. Since we are interested in the error caused by the penalty scheme, we compute the reference solution $\mathbf{m}_{ref,\ell}$ on each mesh \mathcal{T}_ℓ with $\varepsilon = 1e - 4$. This reference solution is used to estimate the error

$$\|\mathbf{m}_h^0 - \mathbf{m}_h^\varepsilon\| \approx \|\mathbf{m}_{ref,\ell} - \mathbf{m}_h^\varepsilon\|.$$

Figure 5.29 shows the estimated error along with the error estimator

$$\eta_\ell^\varepsilon = \left(\frac{1}{2\varepsilon} \|(|\mathbf{m}_\ell| - 1)_+\|_{L^2}^2\right)^{1/2}.$$

We observe that the error decays at a rate of approximately $\varepsilon^{1/2}$. The estimator η_ℓ^ε appears to be reliable, but not efficient. It gives an upper bound and converges with an order of approximately $\varepsilon^{1/4}$. This, however, means that the original quadratic penalty energy

$$\frac{1}{2\varepsilon} \|(|\mathbf{m}_\ell| - 1)_+\|_{L^2}^2 = (\eta_\ell^\varepsilon)^2,$$

plotted for reference in the figure, appears to be also efficient.

Moreover, the penalty error seems to influence the energy in a different manner than the discretization error. Figure 5.30 shows the error in the energy, where we used the extrapolated values

$$\begin{aligned} e(\mathbf{m}_{\mathcal{T}_1}^0) &\approx -0.6941157823, \\ e(\mathbf{m}_{\mathcal{T}_2}^0) &\approx -0.7075830949, \\ e(\mathbf{m}_{\mathcal{T}_3}^0) &\approx -0.7142900317, \end{aligned}$$

for estimation of the error. We observe convergence of the order $\varepsilon^{1/2}$.

Iteration	ε	Energy	η_ℓ^ε	η_ℓ^H	$\ \mathbf{m}_{ref, \mathcal{T}_1} - \mathbf{m}_\ell\ $	N_{Newton}
0	6.25e-2	-0.70940	0.09072	0.1399	0.064066	12
1	3.12e-2	-0.70539	0.07643	0.1388	0.050271	4
2	1.56e-2	-0.70254	0.06601	0.1382	0.040217	4
3	7.81e-3	-0.70039	0.05860	0.1380	0.031730	4
4	3.90e-3	-0.69868	0.05249	0.1382	0.023629	4
5	1.95e-3	-0.69745	0.04876	0.1389	0.017111	4
6	9.77e-4	-0.69636	0.04159	0.1394	0.011788	4
7	4.88e-4	-0.69555	0.03310	0.1401	0.007200	4

Table 5.13: Results of calculations with $\omega = (-0.5, 0.5)^2$, $\mathbf{f} = (1, 1)^T$, $q = 1$, and $N_T = 256$: penalty parameter ε , energy, error estimators, and number of Newton iterations N_{Newton} .

Iteration	ε	Energy	η_ℓ^ε	η_ℓ^H	$\ \mathbf{m}_{ref, \mathcal{T}_1} - \mathbf{m}_\ell\ $	N_{Newton}
0	6.25e-2	-0.72012	0.08864	0.09496	0.050189	16
1	3.12e-2	-0.71618	0.07076	0.09425	0.037144	4
2	1.56e-2	-0.71370	0.05793	0.09387	0.029373	4
3	7.81e-3	-0.71205	0.04901	0.09371	0.023281	4
4	3.90e-3	-0.71087	0.04233	0.09385	0.017904	5
5	1.95e-3	-0.71001	0.03735	0.09424	0.013073	5
6	9.77e-4	-0.70938	0.03333	0.09465	0.009647	4
7	4.88e-4	-0.70887	0.02794	0.09514	0.006076	4

Table 5.14: Results of calculations with $\omega = (-0.5, 0.5)^2$, $\mathbf{f} = (1, 1)^T$, $q = 1$, and $N_T = 1024$: penalty parameter ε , energy, error estimators, and number of Newton iterations N_{Newton} .

Iteration	ε	Energy	η_ℓ^ε	$\ \mathbf{m}_{ref, \mathcal{T}_1} - \mathbf{m}_\ell\ $	N_{Newton}
0	6.25e-2	-0.72498	0.08795	0.043307	26
1	3.12e-2	-0.72101	0.06796	0.029822	5
2	1.56e-2	-0.71866	0.05270	0.022990	4
3	7.81e-3	-0.71727	0.04247	0.018834	4
4	3.90e-3	-0.71639	0.03552	0.015173	4
5	1.95e-3	-0.71577	0.03034	0.011776	5
6	9.77e-4	-0.71533	0.02639	0.009107	5
7	4.88e-4	-0.71502	0.02305	0.006896	5

Table 5.15: Results of calculations with $\omega = (-0.5, 0.5)^2$, $\mathbf{f} = (1, 1)^T$, $q = 1$, and $N_T = 4096$: penalty parameter ε , energy, error estimators, and number of Newton iterations N_{Newton} .

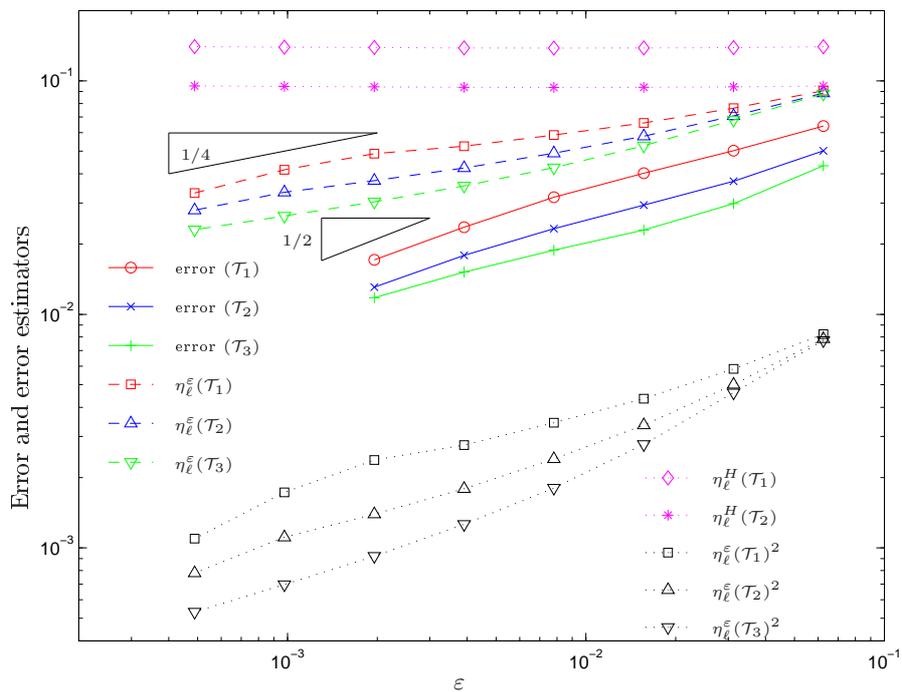


Figure 5.29: Error and error estimators for three different uniform meshes and varying values of ε . We apply the field $\mathbf{f} = (1, 1)^T$ to the sample $\omega = (-0.5, 0.5)^2$ with $q = 1$.

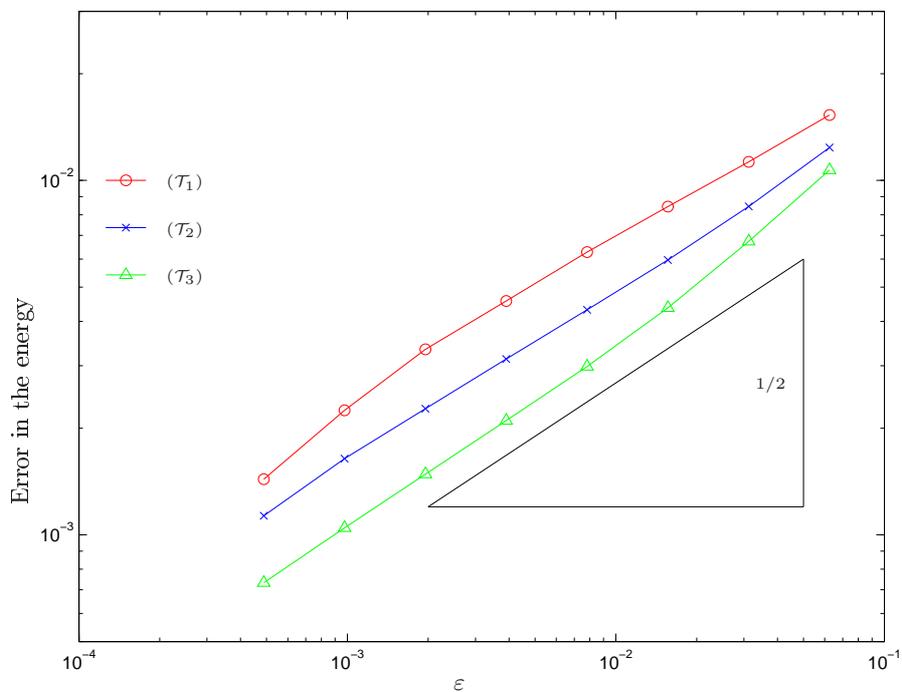


Figure 5.30: Error in the energy for three different uniform meshes and varying values of ε . We apply the field $\mathbf{f} = (1, 1)^T$ to the sample $\omega = (-0.5, 0.5)^2$ with $q = 1$.

The detailed results of the experiments are listed in Tables 5.13–5.15.

The discretization parameter seems to have some positive effect on the penalty error. As the elements shrink in their size, the penalty error is reduced slightly. One reason may be that on coarser meshes the magnetization cannot vary its length efficiently. As the mesh gets finer, some elements that would be linked to a penalized element on a coarse mesh, might not be penalized at all on the finer one, i.e. the penalty region is resolved better.

On the other hand, we observe that the discretization error seems to be almost independent of the penalty parameter. Corollary 3.25 establishes rates of convergence. We stress that the penalty error is only considered for the continuous problem in the proof. Hence, the empirical evidence at hand supports our analytical results. However, the regularity assumptions seem to be too strict and it may be possible to prove the same statement under weaker assumptions.

5.3.2 Empirical choice of $\varepsilon = h^\alpha$

Our analysis predicts the convergence behavior

$$\|\mathbf{m}^* - \mathbf{m}_\ell\| = \mathcal{O}(\sqrt{\varepsilon} + \sqrt{h}).$$

From this, the choice of $\varepsilon = h$ is natural. In particular, the preceding experiments suggest that the individual rates of the discretization error and the penalty error established in Theorem 3.22 and Theorem 3.24 cannot be improved in general with uniform sequences of meshes.

Note, however, that the discretization error and the penalty error are apparently not completely independent. A harsher penalization improves the discretization error and vice versa. Therefore, we need to study the correct choice of $\varepsilon = h^\alpha$ also empirically.

We stick to the simple and smooth setting of $\omega = (-0.5, 0.5)^2$ and constant applied field. The field is chosen $\mathbf{f} = (2, 2)^T$ to ensure that not only the discretization error but also the error introduced by the penalty scheme are significant at the beginning of the simulation. The barrier is active on almost the whole domain. To estimate the error, we computed a reference solution on a mesh with $N_T = 65536$ triangles and $N_D = 98048$ degrees of freedom and a penalty parameter of $\varepsilon = h^{1.5} = 6.52444e - 4$. We compare the solutions computed on a sequence of uniform meshes and penalty parameter of $\varepsilon = h^\alpha$ with $\alpha \in \{0.5, 0.8, 1.0, 1.2, 1.5\}$ with the reference solution. Figure 5.31 shows the error in the energy norm. The choice of $\alpha = 0.5$ and $\alpha = 0.8$ lead to a reduced order of convergence. On the other hand, choice of $\alpha \geq 1$ does not improve the convergence beyond a rate of $N_T^{-1/3}$. This suggests the choice of $\varepsilon = h$ is optimal.

Figure 5.32 shows the total error estimator $\eta_\ell = \eta_\ell^H + \eta_\ell^\varepsilon$. Here, the rate seems to improve up to the choice of $\varepsilon = h^{1.5}$. Recall that the estimator η_ℓ^ε was only observed to be reliable but not efficient. The figure also shows the estimator η_ℓ^ε for values of $\alpha = 0.5$, $\alpha = 1.0$, and $\alpha = 1.5$. Only in the case of $\alpha = 1.5$, the true order of convergence $N_T^{-1/3}$ is reflected by the estimator, since the penalty error is of higher order. This effects that the empirically reliable estimator η_ℓ^ε adopts the order of the overall error. Sticking to the empirical optimal choice of $\varepsilon = h$, the estimator shows a reliable behavior and reveals an order of $N_T^{-1/4}$. The detailed result of the simulations are given in Table 5.16–5.20.

We stress the increased order of convergence $N_T^{-1/3}$ that seems to be caused by the dependence of discretization and penalty error. However, it is not entirely clear whether this is an asymptotic behavior or a preasymptotic effect only. To give further evidence, we perform a second simulation run, where we choose an applied field of $\mathbf{f} = (0.8, 0.8)^T$. In this case, the penalization is not active on large

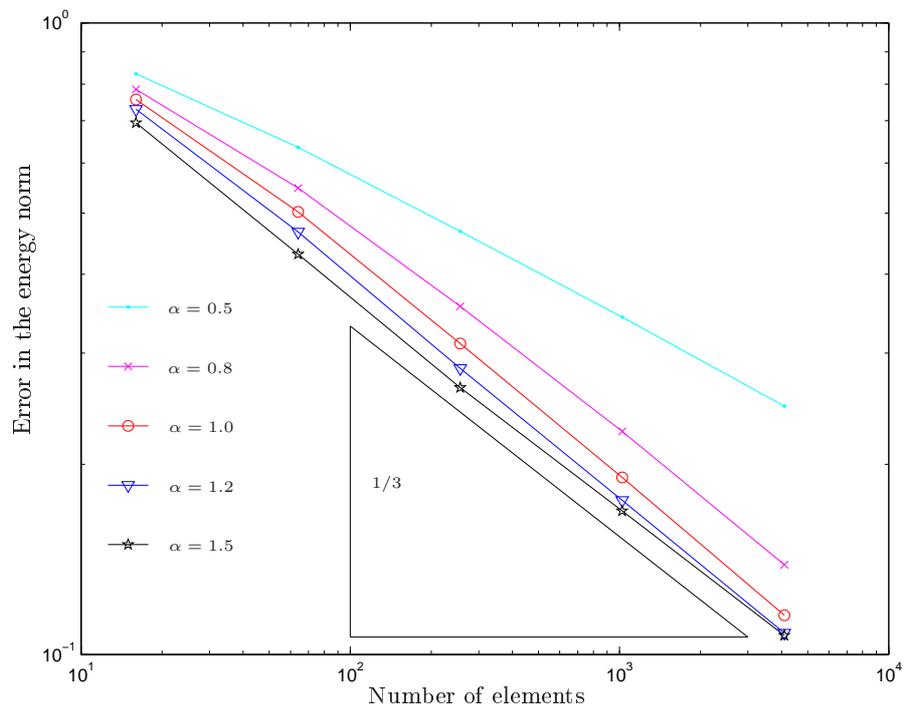


Figure 5.31: Error for different choices of penalty parameter $\varepsilon = h^\alpha$. We apply the field $\mathbf{f} = (2, 2)^T$ to the sample $\omega = (-0.5, 0.5)$ with $q = 1$.

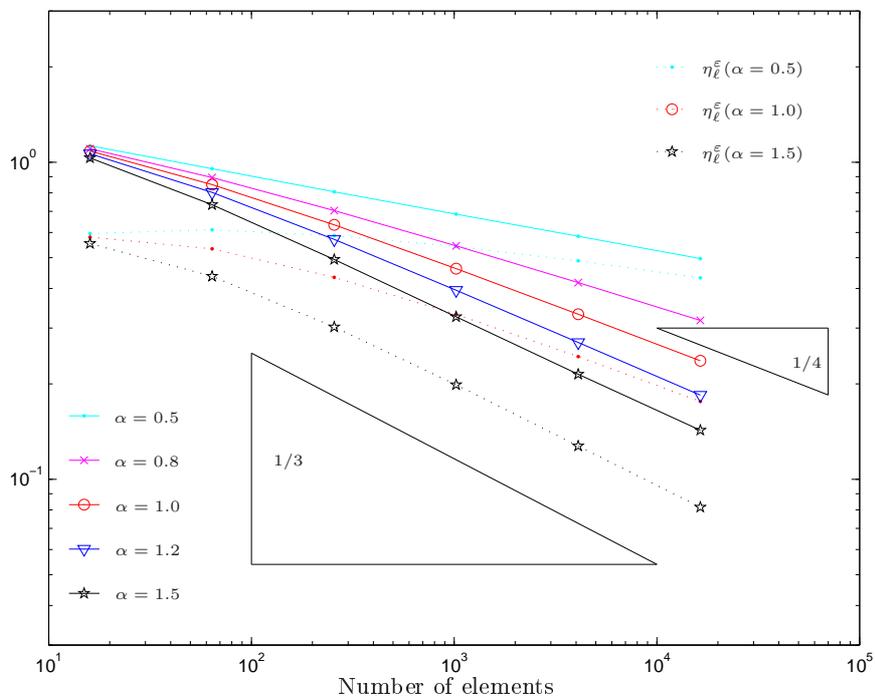


Figure 5.32: Error estimators for different choices of penalty parameter $\varepsilon = h^\alpha$. We apply the field $\mathbf{f} = (2, 2)^T$ to the sample $\omega = (-0.5, 0.5)$ with $q = 1$.

h	ε	$\eta_{\ell,V}^H$	$\eta_{\ell,Q}^H$	$\eta_{\ell}^{\varepsilon}$	$\ \mathbf{m}_{ref} - \mathbf{m}_{\ell}\ $
2.5e-1	5.0e-1	0.4734	0.2420	0.5957	0.8305
1.25e-1	3.5e-1	0.3144	0.1352	0.6119	0.6353
6.25e-2	2.5e-1	0.2079	0.0710	0.5866	0.4676
3.125e-2	1.8e-1	0.1383	0.0369	0.5421	0.3421
1.5625e-2	1.2e-1	0.0935	0.0190	0.4884	0.2472
7.5225e-3	8.5e-2	0.0639	0.0098	0.4320	-

Table 5.16: Results of calculations with $\omega = (-0.5, 0.5)^T$, $\mathbf{f} = (2, 2)^T$, $q = 1$, and $\varepsilon = h^{0.5}$: mesh-width h , penalty parameter ε , error estimators, and error in the energy norm for a sequence of solutions on uniform meshes.

h	ε	$\eta_{\ell,V}^H$	$\eta_{\ell,Q}^H$	$\eta_{\ell}^{\varepsilon}$	$\ \mathbf{m}_{ref} - \mathbf{m}_{\ell}\ $
2.5e-1	3.2e-1	0.4592	0.2326	0.5881	0.7847
1.25e-1	1.8e-1	0.2994	0.1282	0.5689	0.5477
6.25e-2	1.1e-1	0.1960	0.0665	0.4967	0.3557
3.125e-2	6.3e-2	0.1296	0.0347	0.4104	0.2255
1.5625e-2	3.6e-2	0.0878	0.0180	0.3272	0.1387
7.5225e-3	1.9e-2	0.0606	0.0095	0.2554	-

Table 5.17: Results of calculations with $\omega = (-0.5, 0.5)^T$, $\mathbf{f} = (2, 2)^T$, $q = 1$, and $\varepsilon = h^{0.8}$: mesh-width h , penalty parameter ε , error estimators, and error in the energy norm for a sequence of solutions on uniform meshes.

h	ε	$\eta_{\ell,V}^H$	$\eta_{\ell,Q}^H$	$\eta_{\ell}^{\varepsilon}$	$\ \mathbf{m}_{ref} - \mathbf{m}_{\ell}\ $
2.5e-1	2.5e-1	0.4500	0.2266	0.5804	0.7563
1.25e-1	1.3e-1	0.2908	0.1241	0.5331	0.5020
6.25e-2	6.3e-2	0.1901	0.0645	0.4334	0.3107
3.125e-2	3.1e-2	0.1260	0.0340	0.3309	0.1907
1.5625e-2	1.6e-2	0.0859	0.0179	0.2436	0.1153
7.5225e-3	7.2e-3	0.0596	0.0096	0.1759	-

Table 5.18: Results of calculations with $\omega = (-0.5, 0.5)^T$, $\mathbf{f} = (2, 2)^T$, $q = 1$, and $\varepsilon = h^1$: mesh-width h , penalty parameter ε , error estimators, and error in the energy norm for a sequence of solutions on uniform meshes.

h	ε	$\eta_{\ell,V}^H$	$\eta_{\ell,Q}^H$	$\eta_{\ell}^{\varepsilon}$	$\ \mathbf{m}_{ref} - \mathbf{m}_{\ell}\ $
2.5e-1	1.9e-1	0.4410	0.2209	0.5708	0.7301
1.25e-1	8.2e-2	0.2835	0.1205	0.4944	0.4666
6.25e-2	3.6e-2	0.1857	0.0632	0.3744	0.2838
3.125e-2	1.6e-2	0.1237	0.0338	0.2662	0.1754
1.5625e-2	6.8e-3	0.0849	0.0182	0.1831	0.1081
7.5225e-3	2.7e-3	0.0592	0.0100	0.1242	-

Table 5.19: Results of calculations with $\omega = (-0.5, 0.5)^T$, $\mathbf{f} = (2, 2)^T$, $q = 1$, and $\varepsilon = h^{1.2}$: mesh-width h , penalty parameter ε , error estimators, and error in the energy norm for a sequence of solutions on uniform meshes.

h	ε	$\eta_{\ell,V}^H$	$\eta_{\ell,Q}^H$	$\eta_{\ell}^{\varepsilon}$	$\ \mathbf{m}_{ref} - \mathbf{m}_{\ell}\ $
2.5e-1	1.3e-1	0.4283	0.2130	0.5535	0.6948
1.25e-1	4.4e-2	0.2745	0.1158	0.4366	0.4304
6.25e-2	1.6e-2	0.1809	0.0621	0.3019	0.2646
3.125e-2	5.5e-3	0.1218	0.0342	0.1987	0.1687
1.5625e-2	1.9e-3	0.0847	0.0194	0.1273	0.1072
7.5225e-3	6.1e-4	0.0601	0.0109	0.0816	-

Table 5.20: Results of calculations with $\omega = (-0.5, 0.5)^T$, $\mathbf{f} = (2, 2)^T$, $q = 1$, and $\varepsilon = h^{1.5}$: mesh-width h , penalty parameter ε , error estimators, and error in the energy norm for a sequence of solutions on uniform meshes.

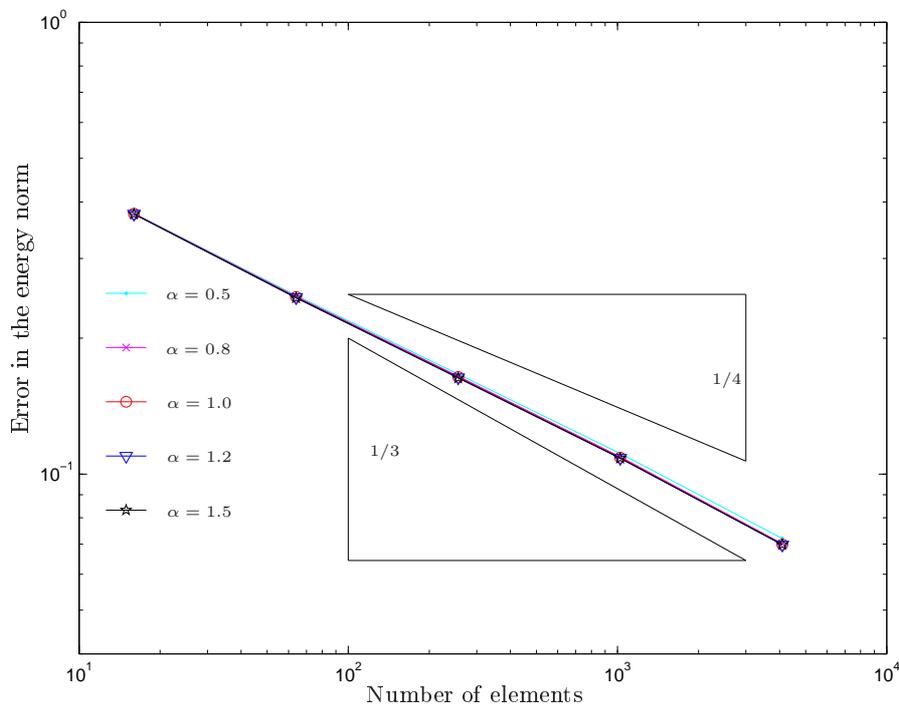


Figure 5.33: Error for different choices of penalty parameter $\varepsilon = h^\alpha$. The field $\mathbf{f} = (0.8, 0.8)^T$ is applied to the sample $\omega = (-0.5, 0.5)^2$ with $q = 1$.

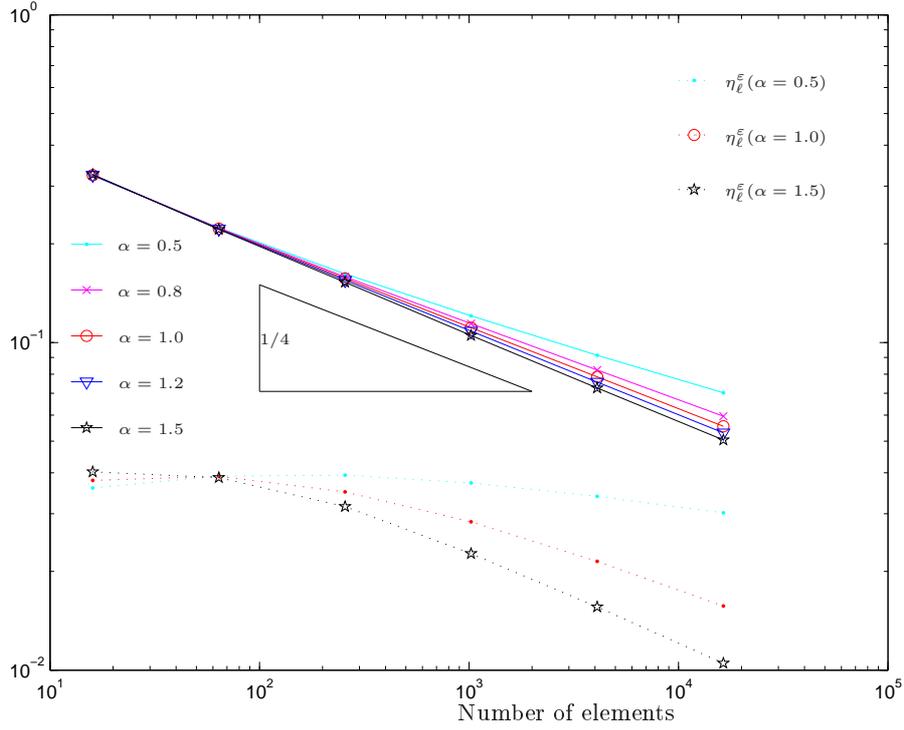


Figure 5.34: Error estimators for different choices of penalty parameter $\varepsilon = h^\alpha$. The field $\mathbf{f} = (0.8, 0.8)^T$ is applied to the sample $\omega = (-0.5, 0.5)^2$ with $q = 1$.

regions of ω but still contributes to the total error. Figure 5.33 shows the error in the energy norm for various choices of $\varepsilon = h^\alpha$. Since the error introduced by the penalty scheme is significantly smaller than the discretization error, the decreased rate of convergence for $\alpha < 1$ when compared to the choice of $\alpha \geq 1$ is not as clearly visible as in the first experiment. We still see, however, a significant decrease for the choice of $\alpha = 0.5$. In contrast to the first experiment with stronger applied field, the choice of $\mathbf{f} = (0.8, 0.8)^T$, now yields a rate of convergence $N_T^{-\alpha}$ with $1/4 < \alpha < 1/3$. Figure 5.34 shows the error estimators in this experiment. Here, the observed reliability of η_ℓ^ε causes that the decreased order of convergence for choices of $\alpha < 1$ is more clearly visible. The results of this simulation are given in Tables 5.21–5.25.

h	ε	$\eta_{\ell,V}^H$	$\eta_{\ell,Q}^H$	η_ℓ^ε	$\ \mathbf{m}_{ref} - \mathbf{m}_\ell\ $
2.5e-1	5.0e-1	0.2513	0.1386	0.03594	0.3773
1.25e-1	3.5e-1	0.1676	0.07772	0.03912	0.2483
6.25e-2	2.5e-1	0.1153	0.04185	0.03934	0.1663
3.125e-2	1.8e-1	0.0802	0.02235	0.03726	0.1110
1.5625e-2	1.2e-1	0.0561	0.01184	0.03392	0.0720
7.5225e-3	8.5e-2	0.0395	0.00623	0.03019	-

Table 5.21: Results of calculations with $\omega = (-0.5, 0.5)^2$, $\mathbf{f} = (0.8, 0.8)^T$, $q = 1$, and $\varepsilon = h^{0.5}$: mesh-width h , penalty parameter ε , error estimators, and error in the energy norm for a sequence of solutions on uniform meshes.

h	ε	$\eta_{\ell,V}^H$	$\eta_{\ell,Q}^H$	$\eta_{\ell}^{\varepsilon}$	$\ \mathbf{m}_{ref} - \mathbf{m}_{\ell}\ $
2.5e-1	3.2e-1	0.2509	0.13849	0.03710	0.3769
1.25e-1	1.8e-1	0.1670	0.07760	0.03914	0.2471
6.25e-2	1.1e-1	0.1146	0.04173	0.03664	0.1646
3.125e-2	6.3e-2	0.0797	0.02235	0.03136	0.1092
1.5625e-2	3.6e-2	0.0558	0.01186	0.02527	0.0701
7.5225e-3	1.9e-2	0.0393	0.00625	0.01969	-

Table 5.22: Results of calculations with $\omega = (-0.5, 0.5)^T$, $\mathbf{f} = (0.8, 0.8)^T$, $q = 1$, and $\varepsilon = h^{0.8}$: mesh-width h , penalty parameter ε , error estimators, and error in the energy norm for a sequence of solutions on uniform meshes.

h	ε	$\eta_{\ell,V}^H$	$\eta_{\ell,Q}^H$	$\eta_{\ell}^{\varepsilon}$	$\ \mathbf{m}_{ref} - \mathbf{m}_{\ell}\ $
2.5e-1	2.5e-1	0.2506	0.13835	0.03792	0.3767
1.25e-1	1.3e-1	0.1665	0.07748	0.03901	0.2464
6.25e-2	6.3e-2	0.1143	0.04171	0.03500	0.1639
3.125e-2	3.1e-2	0.0795	0.02241	0.02834	0.1086
1.5625e-2	1.6e-2	0.0557	0.01192	0.02148	0.0698
7.5225e-3	7.2e-3	0.0392	0.00629	0.01569	-

Table 5.23: Results of calculations with $\omega = (-0.5, 0.5)^T$, $\mathbf{f} = (0.8, 0.8)^T$, $q = 1$, and $\varepsilon = h^1$: mesh-width h , penalty parameter ε , error estimators, and error in the energy norm for a sequence of solutions on uniform meshes.

h	ε	$\eta_{\ell,V}^H$	$\eta_{\ell,Q}^H$	$\eta_{\ell}^{\varepsilon}$	$\ \mathbf{m}_{ref} - \mathbf{m}_{\ell}\ $
2.5e-1	1.9e-1	0.2503	0.13822	0.03880	0.3765
1.25e-1	8.2e-2	0.1661	0.07743	0.03879	0.2459
6.25e-2	3.6e-2	0.1140	0.04171	0.03345	0.1635
3.125e-2	1.6e-2	0.0794	0.02248	0.02585	0.1084
1.5625e-2	6.8e-3	0.0557	0.01199	0.01870	0.0697
7.5225e-3	2.7e-3	0.0392	0.00634	0.01321	-

Table 5.24: Results of calculations with $\omega = (-0.5, 0.5)^T$, $\mathbf{f} = (0.8, 0.8)^T$, $q = 1$, and $\varepsilon = h^{1.2}$: mesh-width h , penalty parameter ε , error estimators, and error in the energy norm for a sequence of solutions on uniform meshes.

h	ε	$\eta_{\ell,V}^H$	$\eta_{\ell,Q}^H$	$\eta_{\ell}^{\varepsilon}$	$\ \mathbf{m}_{ref} - \mathbf{m}_{\ell}\ $
2.5e-1	1.3e-1	0.2499	0.13803	0.04027	0.3762
1.25e-1	4.4e-2	0.1655	0.07740	0.03868	0.2453
6.25e-2	1.6e-2	0.1137	0.04166	0.03154	0.1631
3.125e-2	5.5e-3	0.0793	0.02252	0.02268	0.1084
1.5625e-2	1.9e-3	0.0557	0.01201	0.01559	0.0698
7.5225e-3	6.1e-4	0.0393	0.00635	0.01049	-

Table 5.25: Results of calculations with $\omega = (-0.5, 0.5)^T$, $\mathbf{f} = (0.8, 0.8)^T$, $q = 1$, and $\varepsilon = h^{1.5}$: mesh-width h , penalty parameter ε , error estimators, and error in the energy norm for a sequence of solutions on uniform meshes.

Iteration	N_T	N_D	Energy	$\eta_{\ell,V}^H$	$\eta_{\ell,Q}^H$	η_{ℓ}^H	$\eta_{\ell}^{\varepsilon}$	$\mu_{\ell,V}^H$	N_{Newton}
0	16	20	-2.6723	0.8237	0.5115	0.9696	0.7045	3.7237	7
1	64	88	-3.3119	0.6025	0.2926	0.6698	0.7290	2.7832	5
2	256	368	-3.5269	0.4383	0.1606	0.4668	0.6606	2.0516	5
3	1024	1504	-3.5658	0.3188	0.0867	0.3304	0.5516	1.5011	6
4	4096	6080	-3.5525	0.2314	0.0473	0.2362	0.4382	1.0920	6
5	16384	24448	-3.5331	0.1676	0.0263	0.1697	0.3381	0.7914	8

Table 5.26: Results of calculations with $\omega = (-0.5, 0.5)^2$, singular \mathbf{f} , $q = 1$, and h - ε -uniform refinements: mesh size N_T , space size N_D , energy, error estimators, and number of Newton iterations N_{Newton} .

5.4 A simulation with non-smooth data

In this experiment we perform a simulation with strong singularities of the applied field \mathbf{f} . The aim is to study the performance of the h - ε -adaptive algorithm when compared to the uniform approach with the choice of $\varepsilon = h$. As simulation domain we choose the unit square $\omega = (-0.5, 0.5)^2$ and the anisotropy parameter is $q = 1$. The applied field reads

$$\mathbf{f}(x, y) = \begin{cases} \frac{1}{|x-y|^{1/2}}(1, -2)^T, & \text{for } x < y \\ \frac{1}{|x-y|^{1/2}}(-1, 2)^T, & \text{else} \end{cases}$$

for $(x, y) \in \omega$. The function \mathbf{f} has a singularity along the diagonal. Moreover, it switches orientation at the discontinuity line. From that we expect to observe the magnetization to show singular behavior along $x = y$. We stress that the diagonal is resolved by the mesh exactly. We expect that due to the varying strength of \mathbf{f} the algorithm should lead to an adapted penalization scheme.

Figure 5.35 shows the discrete solution \mathbf{m}_4 computed on a uniform mesh with $N_T = 4096$ triangles and $N_D = 6080$ degrees of freedom and $\varepsilon = h$. The singularities along the diagonal are visible. For reference, Figure 5.36 shows the h - ε -adaptively computed solution \mathbf{m}_{20} on a mesh with $N_T = 1811$ triangles and $N_D = 2671$ degrees of freedom. Figure 5.38 shows the maximal length of the uniform magnetization \mathbf{m}_5 and the penalized and not penalized elements of the underlying mesh with $N_T = 16384$ triangles and $N_D = 24448$ degrees of freedom. Since the applied field is strong close to the diagonal, the penalty scheme allows for a large magnetization there if it is well aligned with \mathbf{f} . Figure 5.37 shows the divergence of the solution \mathbf{m}_{34} computed on an adaptively generated mesh with $N_T = 9132$ triangles and $N_D = 13559$ degrees of freedom. One can clearly see that not only the edges and corners of ω but also the singularity of \mathbf{f} along the diagonal $x = y$ causes non-smooth behavior.

The detailed results of the uniform and the h - ε -adaptive simulations are given in the Tables 5.26 and 5.27. One can see that the energy cannot be extrapolated from the sequence of uniform meshes since the competing discretization and penalty error do not reach an asymptotic behavior throughout the simulation. We therefore only provide plots of the error estimators. The quantities in the uniform case are shown in Figure 5.39. As reliable error estimator, we use the combined quantity $\eta_{\ell}^H + \eta_{\ell}^{\varepsilon}$. We observe a rate of approximately $N_T^{-3/14}$. In particular both, the discretization error and the penalty error adopt this rate asymptotically. The error estimators in the adaptive computation are depicted in Figure 5.41. Here, we observe that both, the penalty error and the discretization error, are improved in the sense that the estimators exhibit a rate of at least $N_T^{-1/4}$. Figure 5.42 shows the length of the h - ε -adaptively computed solution \mathbf{m}_{34} as well as the quantity $1/\varepsilon_{34}$ on the adaptively generated mesh \mathcal{T}_{34} with $N_T = 9132$ triangles and $N_D = 13559$ degrees of freedom. A closer look at

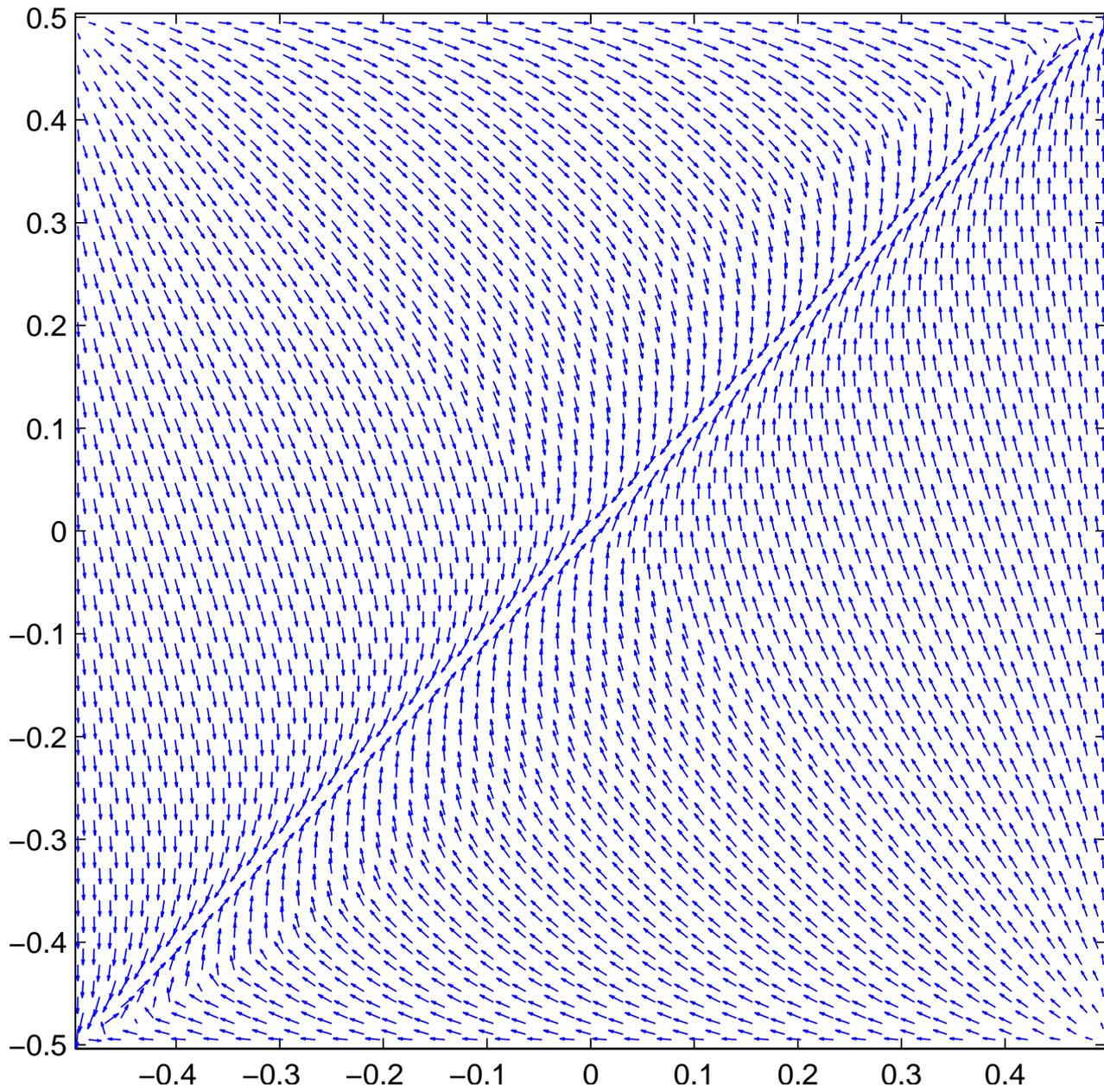


Figure 5.35: Discrete magnetization for singular \mathbf{f} and $q = 1$ on a mesh of $\omega = (-0.5, 0.5)^2$ with $N_T = 4096$ triangles and $N_D = 6080$ degrees of freedom.

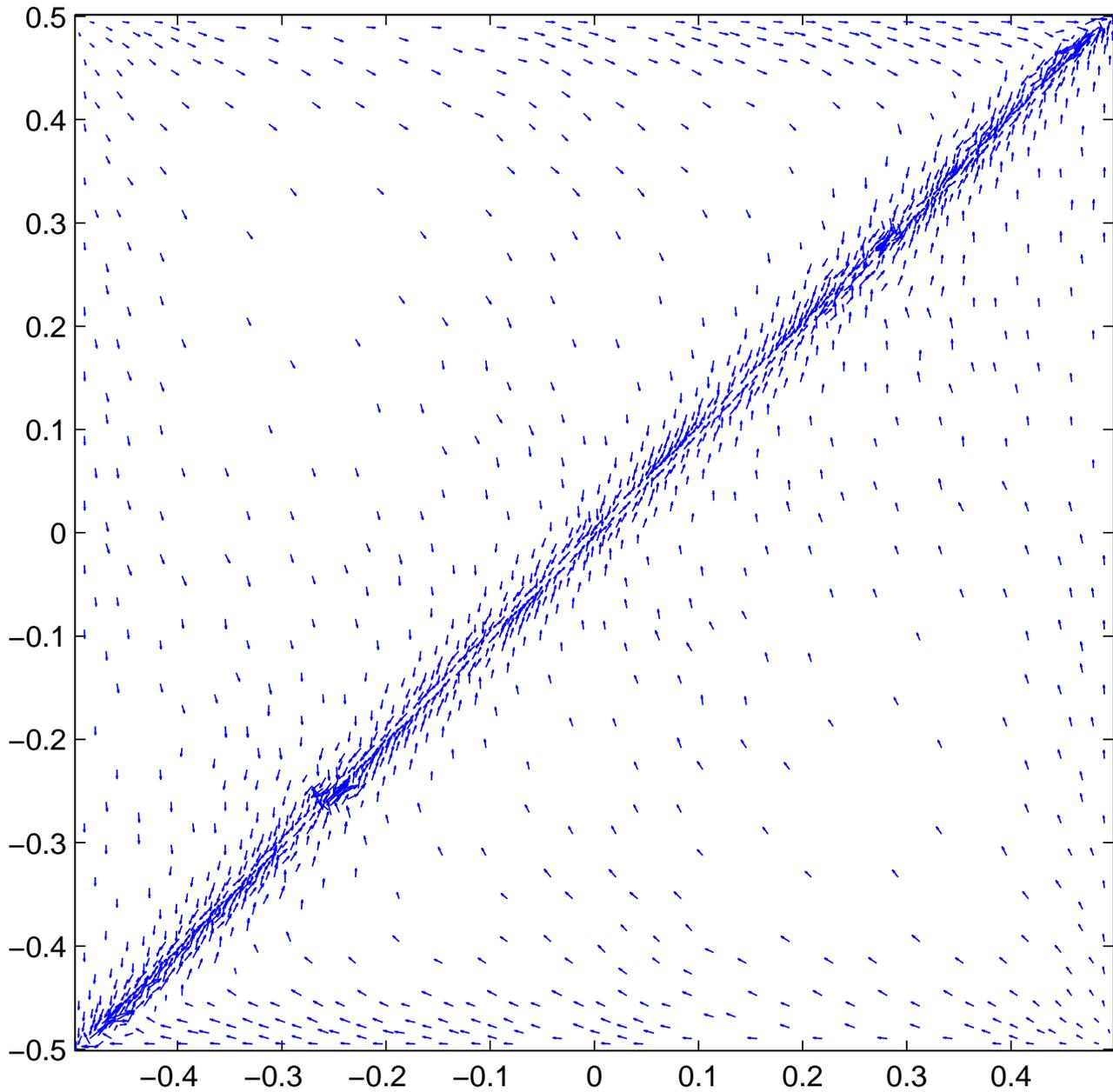


Figure 5.36: Discrete magnetization \mathbf{m}_{20} for singular \mathbf{f} and $q = 1$ on an h - ε -adaptively generated mesh of $\omega = (-0.5, 0.5)^2$ with $N_T = 1811$ triangles and $N_D = 2671$ degrees of freedom.

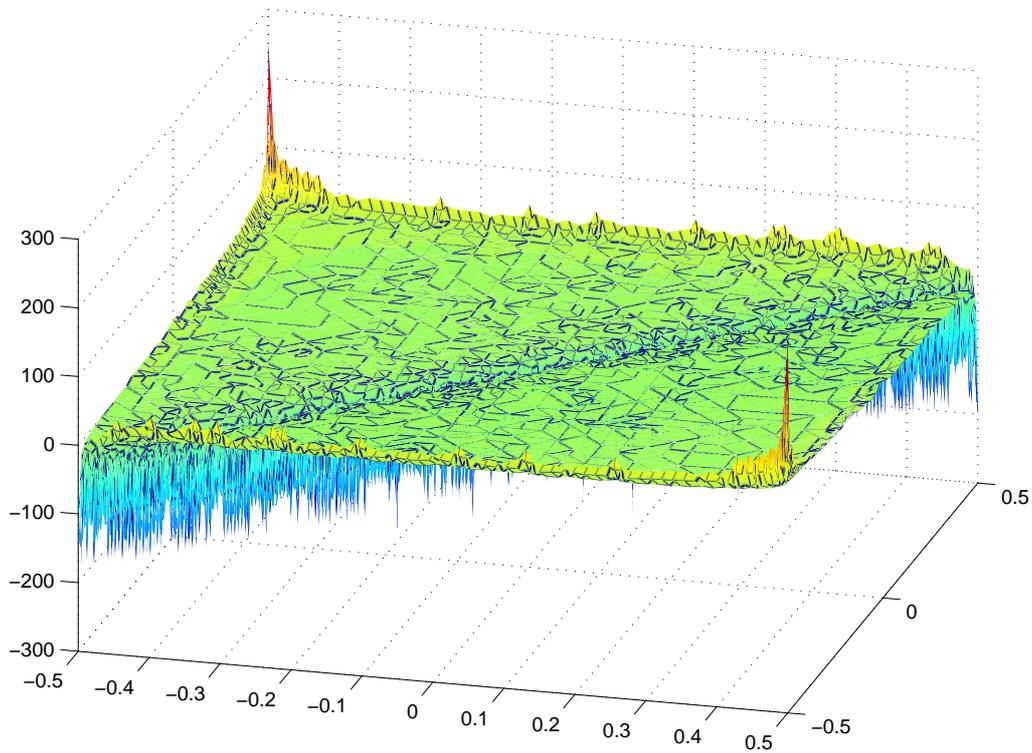


Figure 5.37: Divergence of the h - ε -adaptively generated discrete solution \mathbf{m}_{34} for singular \mathbf{f} and $q = 1$ on a mesh of $\omega = (-0.5, 0.5)^2$ with $N_T = 9132$ triangles and $N_D = 13559$ degrees of freedom.

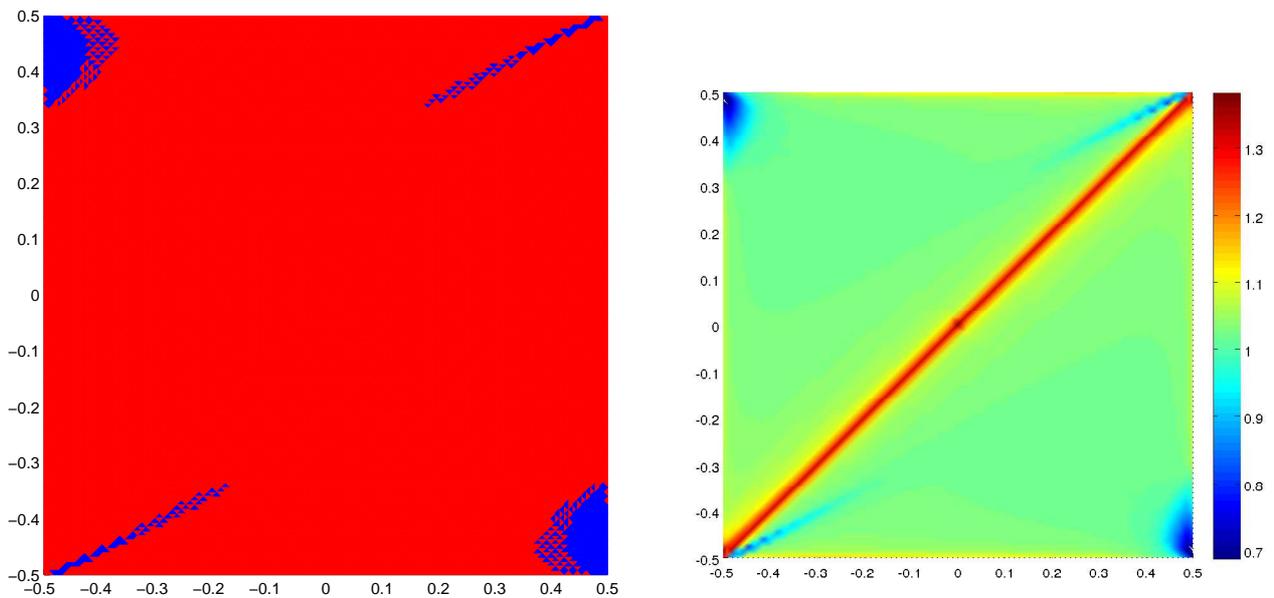


Figure 5.38: Left: Penalized (red) and not penalized (blue) elements of the solution \mathbf{m}_5 for singular applied field \mathbf{f} and $q = 1$. Right: Maximal length of the magnetization \mathbf{m}_5 on each element. The uniform mesh of $\omega = (-0.5, 0.5)^2$ has $N_T = 16384$ triangles and $N_D = 24448$ degrees of freedom.

Iteration	N_T	N_D	Energy	$\eta_{\ell,V}^H$	$\eta_{\ell,Q}^H$	η_{ℓ}^H	$\eta_{\ell}^{\varepsilon}$	$\mu_{\ell,V}^H$	N_{Newton}
0	16	20	-2.6723	0.82372	0.51158	0.96966	0.70455	3.7237	7
1	32	42	-3.3341	0.69361	0.39611	0.79875	0.78708	3.1820	5
2	52	72	-3.4038	0.64831	0.34492	0.73436	0.74582	2.9451	5
3	82	115	-3.6417	0.57528	0.28641	0.64264	0.80866	2.5969	5
4	138	197	-3.7978	0.48939	0.23664	0.54360	0.78390	2.1957	5
5	178	256	-3.8379	0.44948	0.20697	0.49484	0.77150	2.0476	5
6	284	414	-3.9027	0.40628	0.17703	0.44317	0.75368	1.8390	5
7	354	515	-3.8918	0.35982	0.15549	0.39198	0.73026	1.6232	5
8	400	584	-3.8492	0.34790	0.15295	0.38004	0.70001	1.5630	4
9	484	710	-3.8347	0.33100	0.14505	0.36139	0.67998	1.4857	4
10	580	849	-3.8344	0.30368	0.13834	0.33371	0.66287	1.3598	5
11	685	1004	-3.8138	0.28995	0.13047	0.31796	0.64026	1.2956	5
12	725	1062	-3.7647	0.28835	0.12801	0.31549	0.60869	1.2820	5
13	790	1157	-3.7361	0.28384	0.12557	0.31038	0.57926	1.2557	5
14	848	1244	-3.6958	0.28207	0.12561	0.30878	0.54618	1.2486	5
15	952	1400	-3.6759	0.27210	0.12267	0.29847	0.52082	1.2072	7
16	1063	1563	-3.6537	0.26302	0.12335	0.29053	0.49565	1.1659	5
17	1287	1896	-3.6551	0.24900	0.12054	0.27664	0.48189	1.1064	6
18	1464	2160	-3.6358	0.24263	0.11584	0.26887	0.45831	1.0787	7
19	1674	2469	-3.6336	0.23271	0.10894	0.25695	0.44349	1.0308	7
20	1811	2671	-3.6214	0.22760	0.10239	0.24957	0.42533	1.0082	12
21	2104	3106	-3.6184	0.21739	0.09568	0.23752	0.41189	0.9615	9
22	2308	3408	-3.6074	0.21093	0.09276	0.23043	0.39449	0.9335	7
23	2528	3737	-3.5944	0.20782	0.09147	0.22706	0.37515	0.9191	7
24	3057	4527	-3.5957	0.19584	0.08988	0.21548	0.36296	0.8713	8
25	3359	4976	-3.5869	0.19051	0.08777	0.20976	0.34821	0.8471	8
26	3720	5507	-3.5799	0.18495	0.08212	0.20236	0.33569	0.8199	9
27	4260	6310	-3.5760	0.17750	0.07887	0.19424	0.32686	0.7867	9
28	4773	7067	-3.5645	0.17135	0.07649	0.18765	0.31439	0.7592	7
29	5275	7812	-3.5581	0.16612	0.07433	0.18199	0.30185	0.73417	7
30	5865	8692	-3.5502	0.16184	0.07095	0.17671	0.29039	0.7165	10
31	6817	10119	-3.5496	0.15550	0.06842	0.16989	0.28024	0.6897	12
32	7715	11460	-3.5506	0.14915	0.06565	0.16296	0.27181	0.6636	10
33	8425	12512	-3.5472	0.14536	0.06445	0.15901	0.26238	0.6455	12
34	9132	13559	-3.5428	0.14156	0.06292	0.15491	0.25274	0.6274	9

Table 5.27: Results of calculations for $\omega = (-0.5, 0.5)^2$, singular \mathbf{f} , $q = 1$, and h - ε -adaptive refinements: mesh size N_T , space size N_D , energy, error estimators, and number of Newton iterations N_{Newton} .

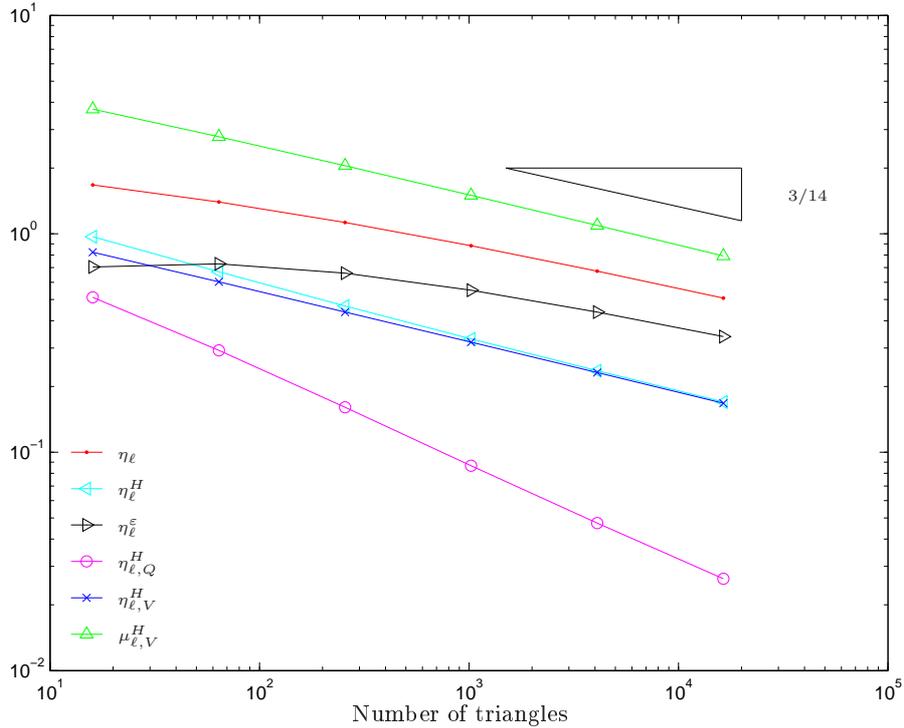


Figure 5.39: Error estimators for $\omega = (-0.5, 0.5)^2$, singular \mathbf{f} , $q = 1$ and h - ε -uniform refinements.

the penalty parameter reveals the effects of the anisotropy and shows that along the diagonal $x = y$ the penalization parameter is very small. A sequence of adaptively generated meshes is shown in Figure 5.40. As expected, we observe resolution of the discontinuity line.

5.5 Hard and soft material

In the present section, we perform simulations for the choice of $\omega = (-0.5, 0.5) \times (-0.1, 0.1)$. As before, in the uniform computations we choose $\varepsilon = h$.

First, we study the effect of hard material with anisotropy parameter of $q = 1e4$. We apply the constant field $\mathbf{f} = 10(0.5, 2.0)^T$. Figure 5.43 shows the solution \mathbf{m}_3 computed on a uniform mesh with $N_T = 1280$ triangles and $N_D = 1872$ degrees of freedom. We observe that the magnetization is almost aligned with the easy axis, although the applied field is very strong when compared to the prior examples with constant applied field. The divergence of the h - ε -adaptively computed solution \mathbf{m}_{16} on a mesh with $N_T = 14662$ triangles and $N_D = 21675$ degrees of freedom is shown in Figure 5.44. We observe edge singularities at the left and right end of ω . The penalized elements of the uniform mesh \mathcal{T}_5 with $N_T = 20480$ triangles are shown in Figure 5.45 along with the length of the corresponding discrete magnetization \mathbf{m}_5 with $N_D = 30528$ degrees of freedom. We observe that the penalty scheme is active only where the solution is smooth.

Estimating the value of the energy by extrapolation of the values obtained by the sequence of uniform meshes yields $e(\mathbf{m}^*) \approx -0.897945894609717$. Figure 5.46 shows the error plot, where we used this extrapolated value. The slow convergence of the energy in case of uniform meshes convinced us that the extrapolated value is not very accurate. Possibly this is caused due to the values of the energy

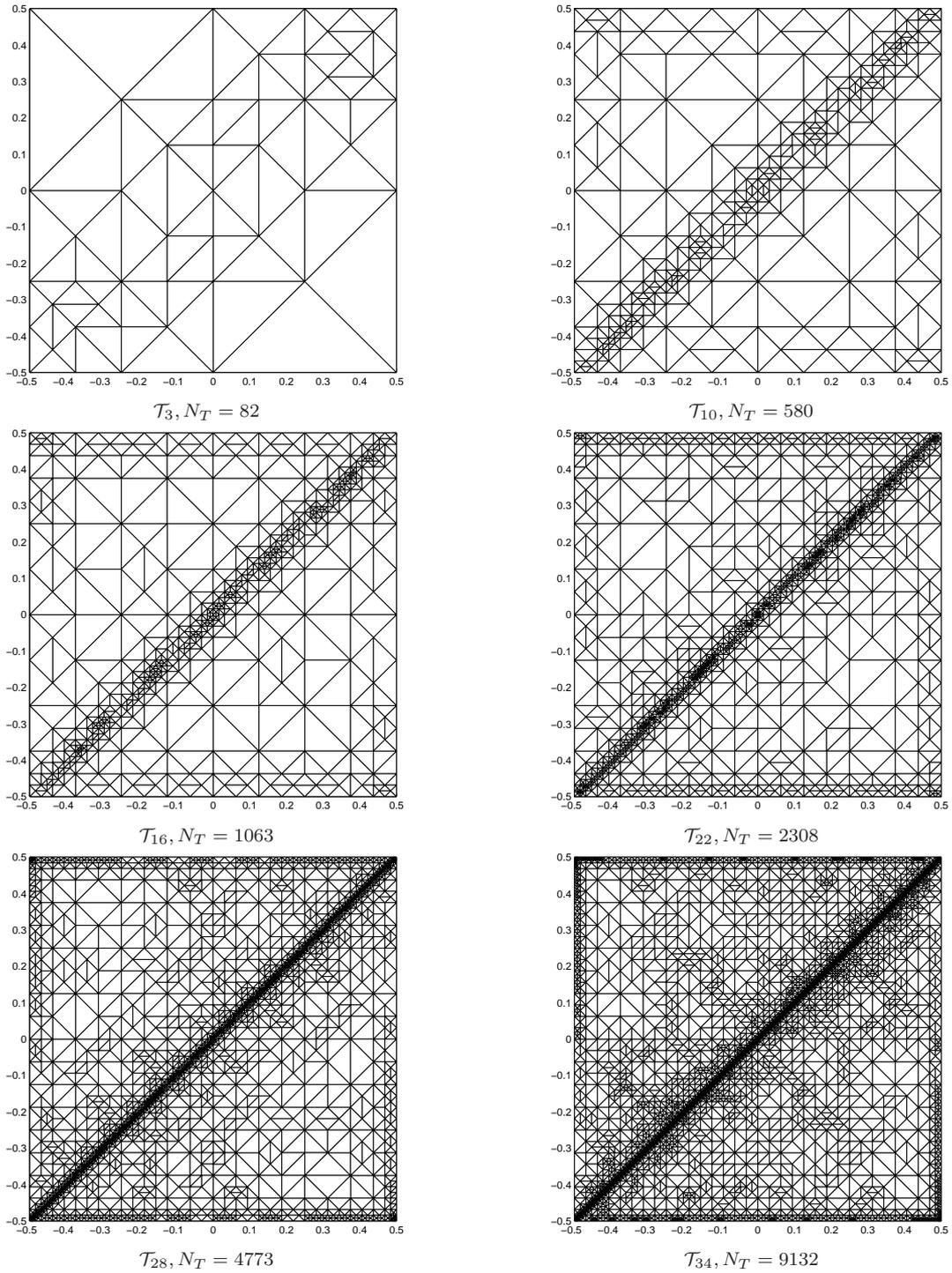


Figure 5.40: Sequence of h - ε -adaptively generated meshes of $\omega = (-0.5, 0.5)^2$ with singular applied field \mathbf{f} and $q = 1$.

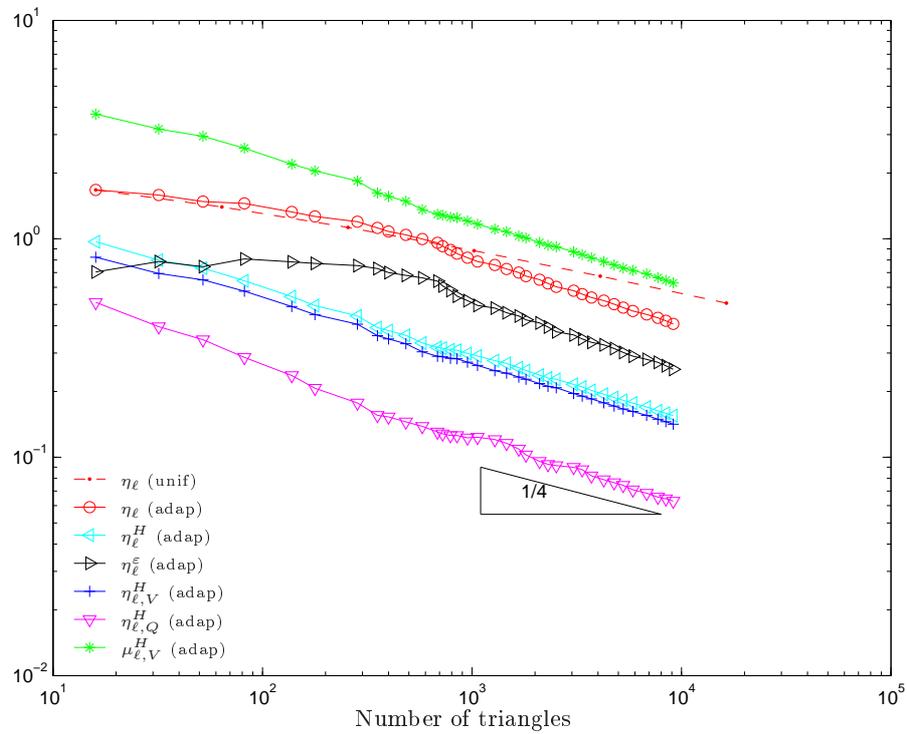


Figure 5.41: Error estimators for h - ε -adaptive refinements with $\omega = (-0.5, 0.5)^2$, singular applied field \mathbf{f} and $q = 1$.

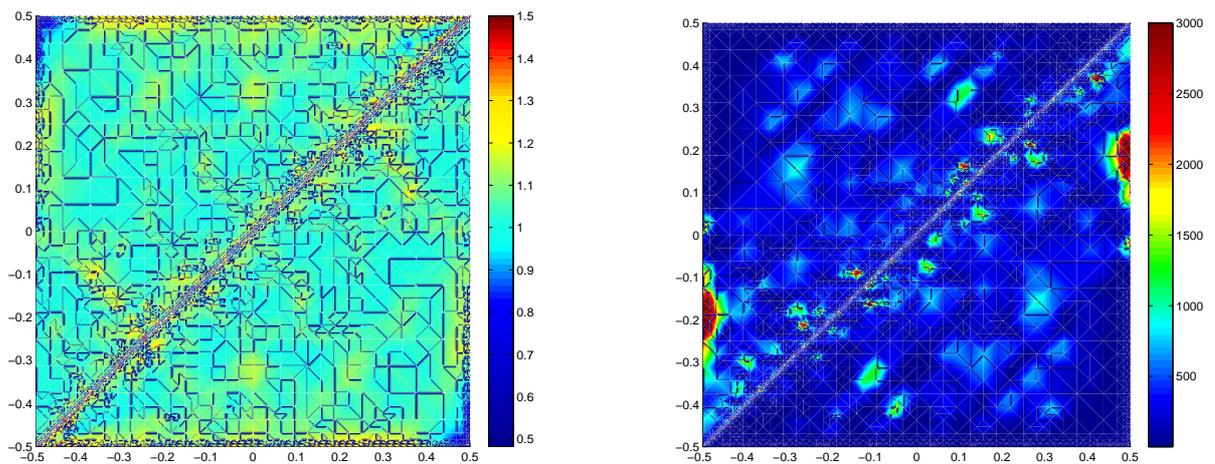


Figure 5.42: Left: Maximal length of the h - ε -adaptively generated solution \mathbf{m}_{34} for $\omega = (-0.5, 0.5)^2$, singular applied field \mathbf{f} , and anisotropy parameter $q = 1$. Right: Inverse of the adaptively generated penalty parameter $1/\varepsilon_\ell$.

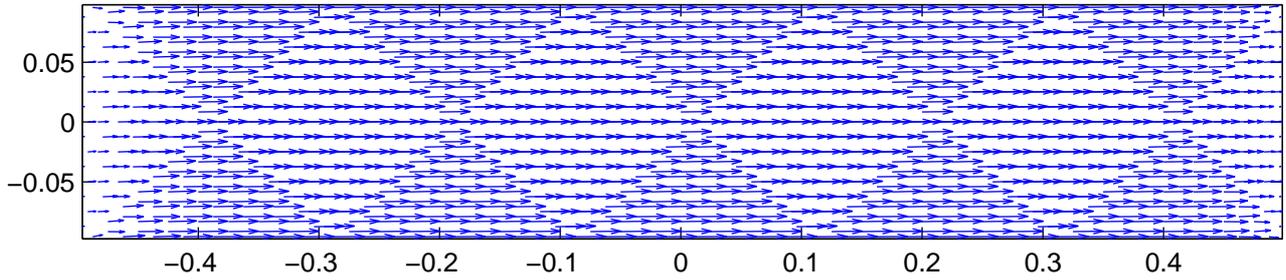


Figure 5.43: Discrete magnetization for $\omega = (-0.5, 0.5) \times (-0.1, 0.1)$, $\mathbf{f} = 10(0.5, 2.0)^T$ and $q = 1e4$ on a uniform mesh with $N_T = 1280$ triangles and $N_D = 1872$ degrees of freedom.

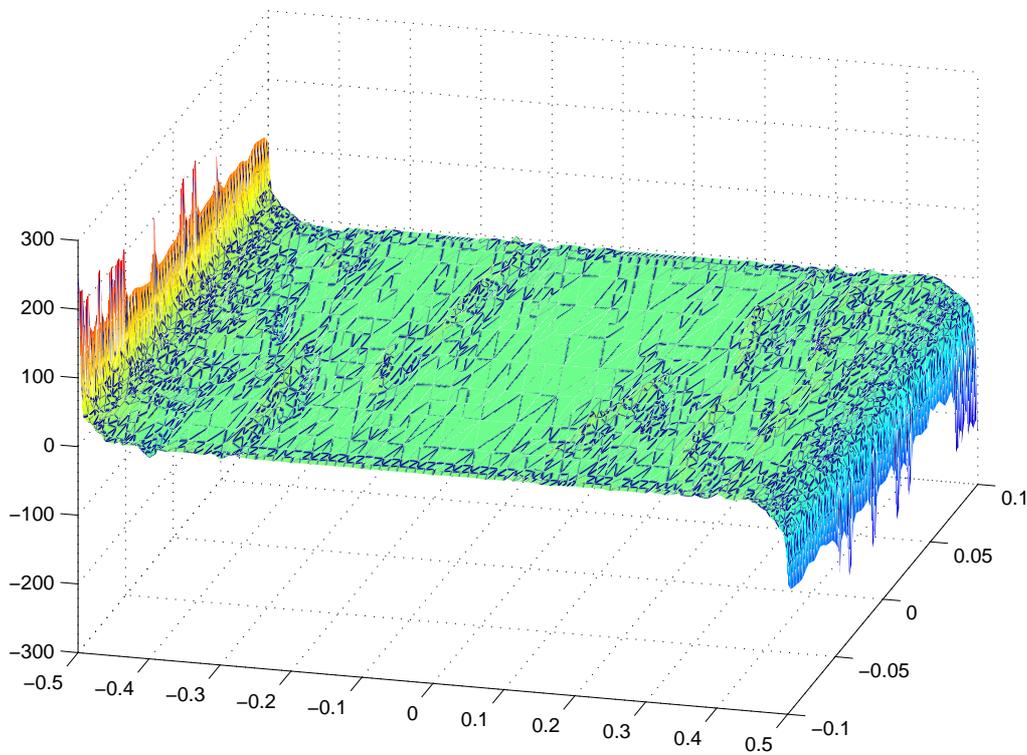


Figure 5.44: Divergence of the h - ε -adaptively generated discrete solution \mathbf{m}_{16} for $\omega = (-0.5, 0.5) \times (-0.1, 0.1)$, $\mathbf{f} = 10(0.5, 2.0)^T$, and $q = 1e4$ on a mesh with $N_T = 14662$ triangles and $N_D = 21675$ degrees of freedom.

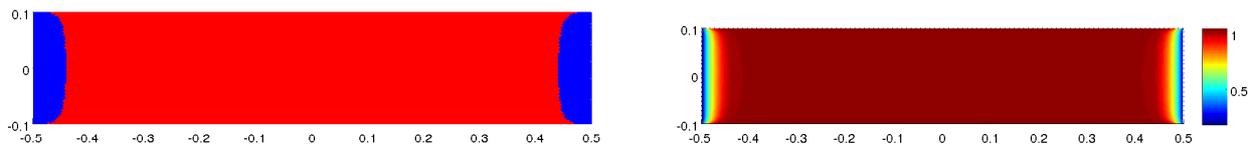


Figure 5.45: Left: Penalized (red) and not penalized (blue) elements of the solution \mathbf{m}_5 for $\omega = (-0.5, 0.5) \times (-0.1, 0.1)$, $\mathbf{f} = 10(0.5, 2.0)^T$, and $q = 1e4$. Right: Maximal length of the magnetization \mathbf{m}_5 on each element. The uniform mesh has $N_T = 20480$ triangles and $N_D = 30528$ degrees of freedom.

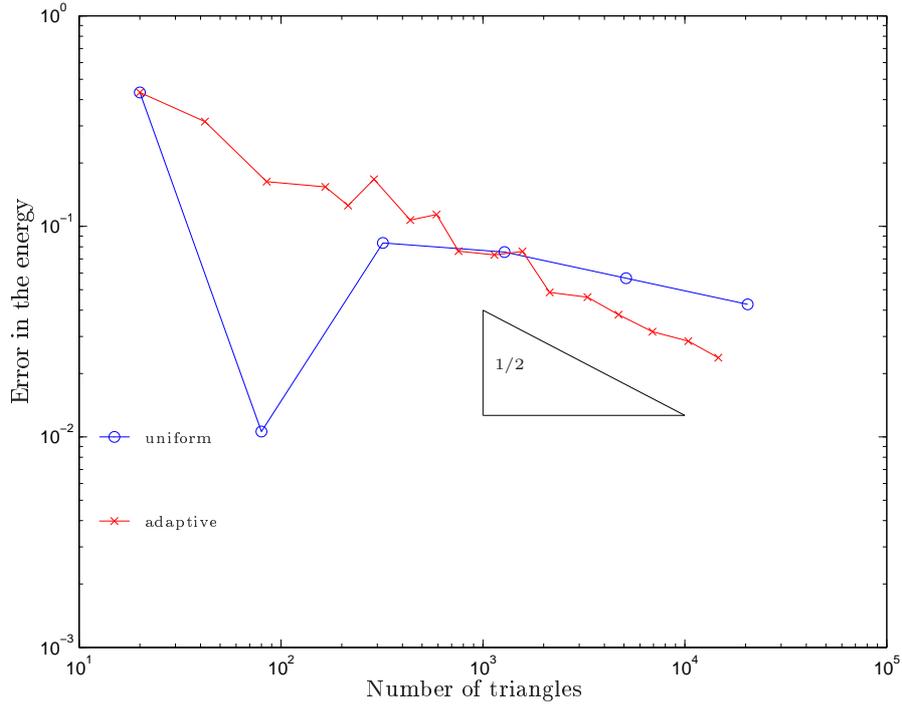


Figure 5.46: Error in the energy $|e(\mathbf{m}^*) - e(\mathbf{m}_\ell)|$ for a sequence of discrete solutions on h - ε -uniform and h - ε -adaptive meshes of $\omega = (-0.5, 0.5) \times (-0.1, 0.1)$ with $\mathbf{f} = 10(0.5, 2.0)^T$, and $q = 1e4$.

still being in a pre-asymptotic regime.

The results of the uniform and h - ε -adaptive computation are given in Table 5.28 and Table 5.29. Figure 5.47 shows the error in the energy norm as well as all error estimators for uniform mesh refinements. We observe an order of convergence of approximately $N_T^{-1/3}$. However, the empirically reliable estimator η_ℓ reflects an order of $N_T^{-1/4}$. This is caused by the contribution η_ℓ^ε . The results of the h - ε -adaptive computation are depicted in Figure 5.48. We observe that the error estimator η_ℓ decreases at a rate of $N_T^{-1/3}$. The contribution of the penalty error η_ℓ^ε is of higher order than in the uniform case. A selection of meshes generated by the h - ε -adaptive algorithm is shown in Figure 5.49. Figure 5.50 shows the adaptive penalty parameter ε_ℓ on the mesh \mathcal{T}_{16} with $N_T = 14662$ triangles. We observe that the penalty parameter is refined only where the length of the magnetization is large.

Iteration	N_T	N_D	Energy	$\eta_{\ell,V}^H$	$\eta_{\ell,Q}^H$	η_ℓ^H	η_ℓ^ε	$\mu_{\ell,V}^H$	N_{Newton}
0	20	24	-0.4653	0.17531	0.77221	0.79186	0.08910	0.4280	4
1	80	108	-0.8873	0.12883	0.53677	0.55201	0.38312	0.3203	5
2	320	456	-0.9813	0.10172	0.32762	0.34305	0.36231	0.2508	5
3	1280	1872	-0.9735	0.06956	0.19877	0.21059	0.27769	0.1851	5
4	5120	7584	-0.9546	0.04224	0.11734	0.12471	0.20180	0.1362	5
5	20480	30528	-0.9405	0.02619	0.06682	0.07177	0.14415	0.1023	5
6	81920	122496	-0.9318	-	-	-	-	-	4

Table 5.28: Results of calculations with $\omega = (-0.5, 0.5) \times (-0.1, 0.1)$, $\mathbf{f} = 10(0.5, 2.0)^T$, $q = 1e4$, and h - ε -uniform refinements: mesh size N_T , space size N_D , energy, error estimators, and number of Newton iterations N_{Newton} .

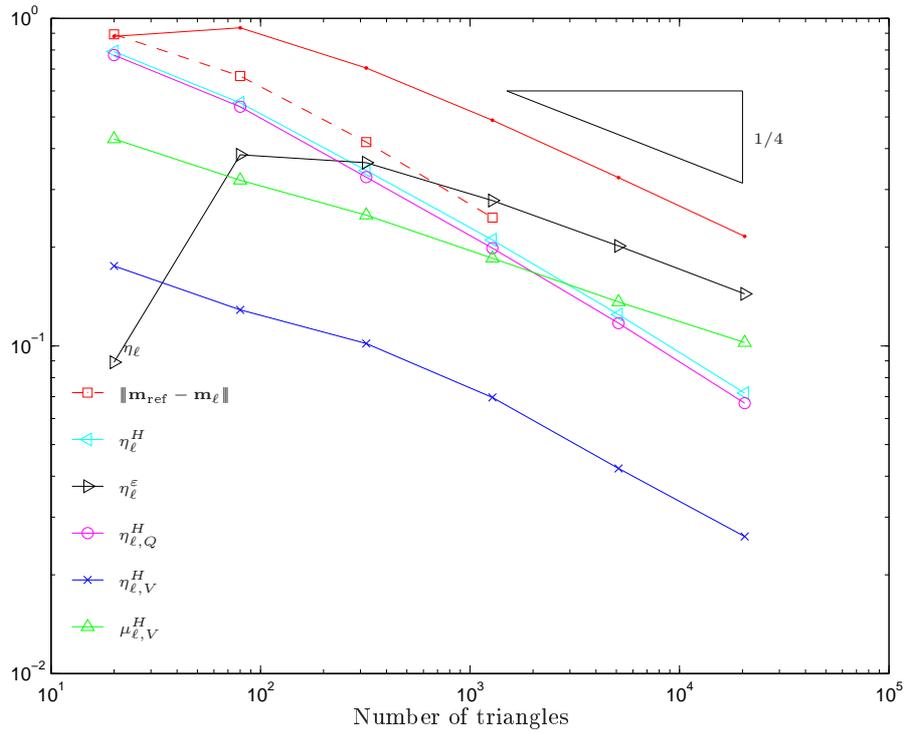


Figure 5.47: Error and error estimators for a sequence of discrete solutions \mathbf{m}_ℓ on h - ε -uniform meshes of $\omega = (-0.5, 0.5) \times (-0.1, 0.1)$ with $\mathbf{f} = 10(0.5, 2.0)^T$ and $q = 1e4$.

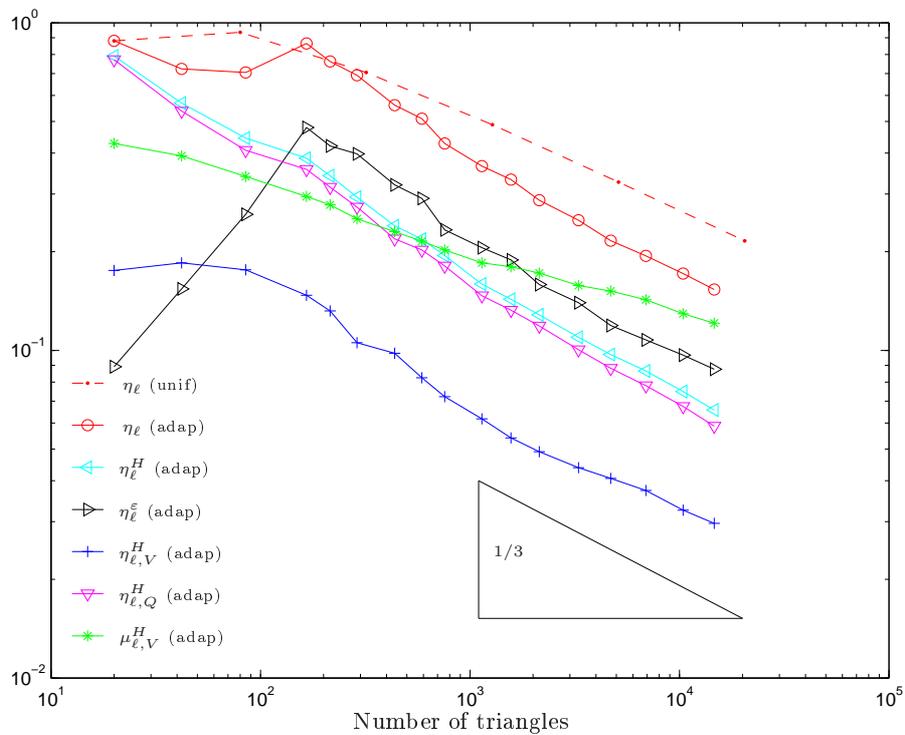


Figure 5.48: Error estimators for a sequence of discrete solutions \mathbf{m}_ℓ on h - ε -adaptive meshes of $\omega = (-0.5, 0.5) \times (-0.1, 0.1)$ with $\mathbf{f} = 10(0.5, 2.0)^T$ and $q = 1e4$.

Iteration	N_T	N_D	Energy	$\eta_{\ell,V}^H$	$\eta_{\ell,Q}^H$	η_{ℓ}^H	$\eta_{\ell}^{\varepsilon}$	$\mu_{\ell,V}^H$	N_{Newton}
0	20	24	-0.4653	0.17531	0.77221	0.79186	0.08910	0.4280	4
1	42	54	-0.5833	0.18501	0.53768	0.56862	0.15410	0.3925	4
2	85	114	-0.7349	0.17615	0.40820	0.44459	0.26031	0.3395	4
3	166	228	-1.0520	0.14728	0.35671	0.38592	0.47874	0.2948	4
4	215	299	-1.0234	0.13198	0.31481	0.34136	0.42063	0.2781	4
5	289	408	-1.0652	0.10554	0.27393	0.29357	0.39815	0.2520	4
6	437	619	-1.0050	0.09805	0.21895	0.23990	0.32020	0.2302	4
7	590	844	-1.0118	0.08250	0.20249	0.21865	0.29101	0.2154	4
8	758	1092	-0.9741	0.07221	0.18093	0.19480	0.23333	0.2026	4
9	1140	1644	-0.9711	0.06179	0.14684	0.15931	0.20588	0.1849	4
10	1569	2277	-0.9740	0.05408	0.13267	0.14327	0.18890	0.1800	4
11	2142	3127	-0.9465	0.04905	0.11867	0.12841	0.15890	0.1721	5
12	3300	4815	-0.9440	0.04387	0.10052	0.10968	0.13984	0.1578	5
13	4702	6897	-0.9360	0.04065	0.08820	0.09712	0.11914	0.1517	5
14	6914	10173	-0.9295	0.03734	0.07807	0.08654	0.10766	0.1427	7
15	10421	15340	-0.9264	0.03251	0.06745	0.07488	0.09670	0.1293	7
16	14662	21675	-0.9217	0.02964	0.05874	0.06579	0.08762	0.1211	7

Table 5.29: Results of calculations with $\omega = (-0.5, 0.5) \times (-0.1, 0.1)$, $\mathbf{f} = 10(0.5, 2.0)^T$, $q = 1e4$, and h - ε -adaptive refinements: mesh size N_T , space size N_D , energy, error estimators, and number of Newton iterations N_{Newton} .

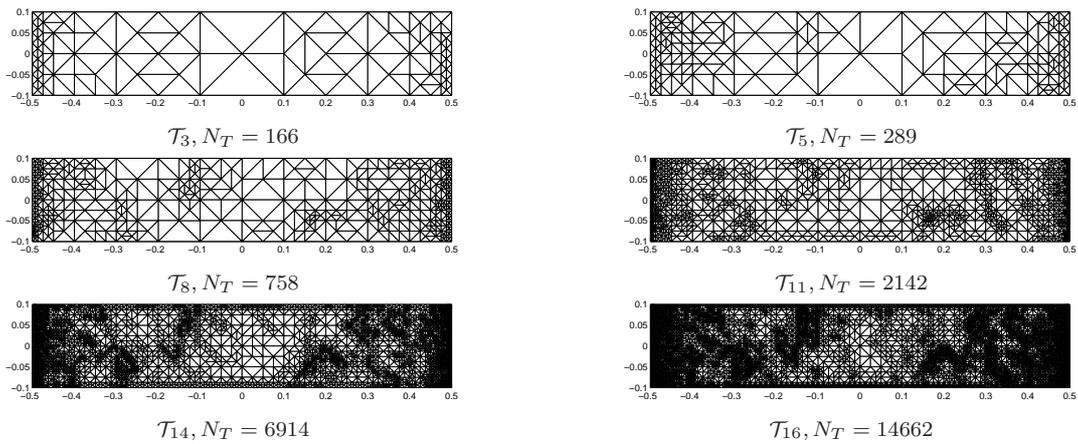


Figure 5.49: Sequence of h - ε -adaptively generated meshes of $\omega = (-0.5, 0.5) \times (-0.1, 0.1)$ with $\mathbf{f} = 10(0.5, 2.0)^T$ and $q = 1e4$.

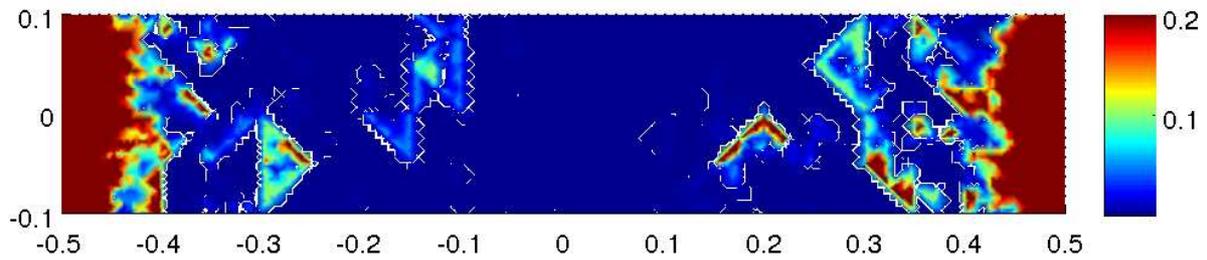


Figure 5.50: Adaptive penalty parameter on the mesh \mathcal{T}_{16} of $\omega = (-0.5, 0.5) \times (-0.1, 0.1)$ with $N_T = 14662$ triangles, $\mathbf{f} = 10(0.5, 2.0)^T$, and $q = 1e4$.

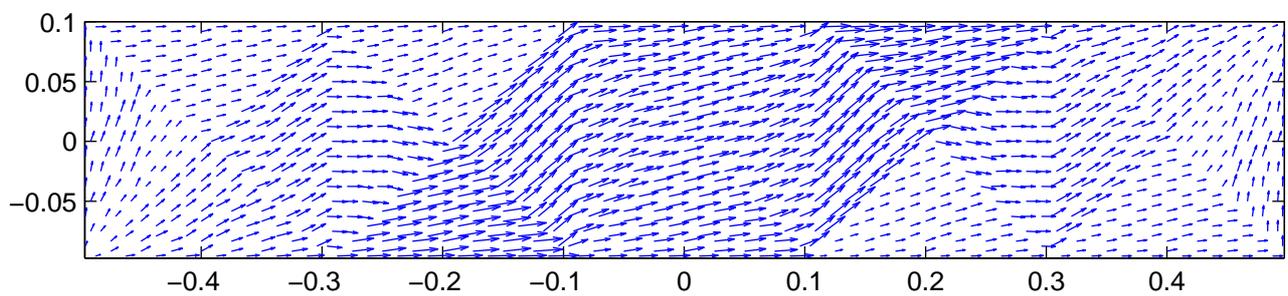


Figure 5.51: Discrete magnetization for $\omega = (-0.5, 0.5) \times (-0.1, 0.1)$, $\mathbf{f} = 0.15(0.5, 2.0)^T$, and $q = 0$ on a uniform mesh with $N_T = 1280$ triangles and $N_D = 1872$ degrees of freedom.

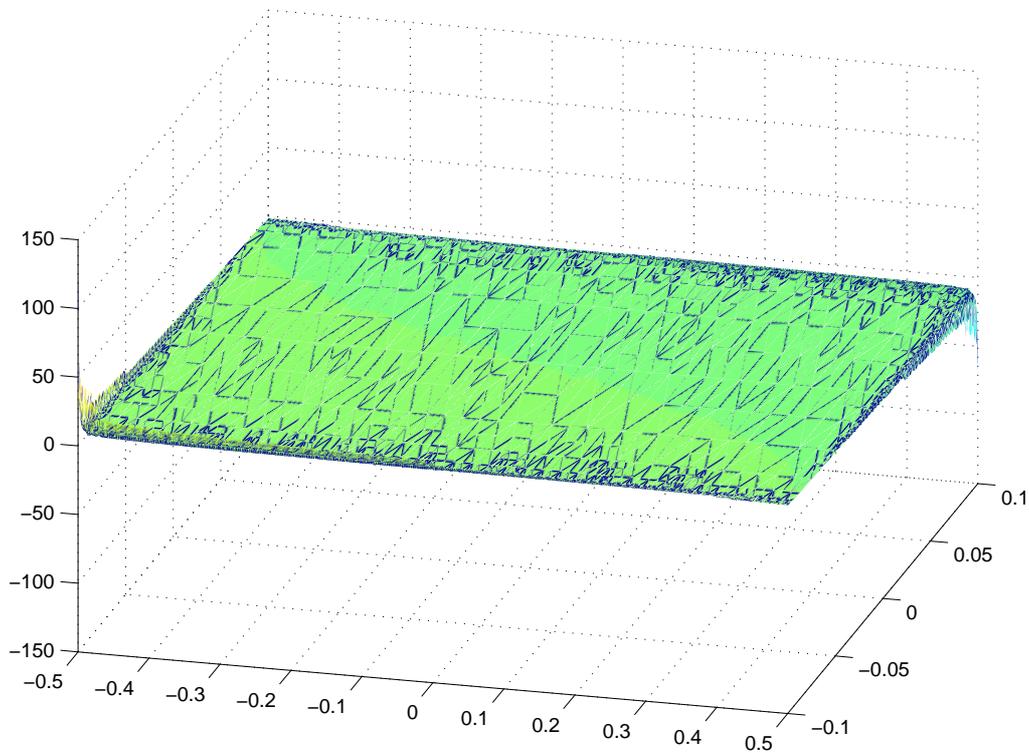


Figure 5.52: Divergence of the h - ε -adaptively generated discrete solution \mathbf{m}_{41} for $\omega = (-0.5, 0.5) \times (-0.1, 0.1)$, $\mathbf{f} = 0.15(0.5, 2.0)^T$, and $q = 0$ on a mesh with $N_T = 9070$ triangles and $N_D = 13031$ degrees of freedom.

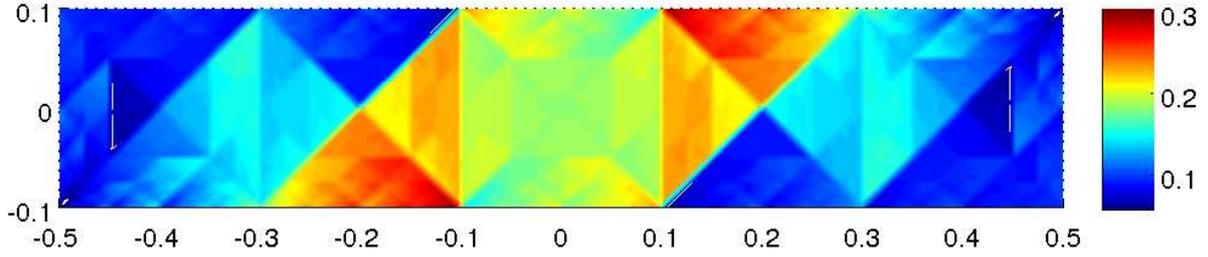


Figure 5.53: Maximal length of the magnetization \mathbf{m}_5 on each element. The uniform mesh of $\omega = (-0.5, 0.5) \times (-0.1, 0.1)$ has $N_T = 20480$ triangles and $N_D = 30528$ degrees of freedom. The field $\mathbf{f} = 0.15(0.5, 2.0)^T$ is applied to the soft sample with $q = 0$.

Iteration	N_T	N_D	Energy	η_ℓ	$\mu_{\ell,V}^H$	N_{Newton}
0	20	24	-0.002015	0.024069	0.11008	2
1	80	108	-0.002310	0.015260	0.06969	2
2	320	456	-0.002429	0.010265	0.04714	2
3	1280	1872	-0.002482	0.007094	0.03271	2
4	5120	7584	-0.002508	0.004964	0.02295	2
5	20480	30528	-0.002521	0.003494	0.01618	2
6	81920	122496	-0.002527	-	-	2

Table 5.30: Results of calculations with $\omega = (-0.5, 0.5) \times (-0.1, 0.1)$, $\mathbf{f} = 0.15(0.5, 2.0)^T$, $q = 0$, and uniform mesh-refinements: mesh size N_T , space size N_D , energy, error estimators, and number of Newton iterations N_{Newton} .

After analyzing the effects of strong crystalline anisotropy, we study the behavior of soft material. First, we apply the field $\mathbf{f} = 0.15(0.5, 2.0)^T$ to the sample $\omega = (-0.5, 0.5) \times (-0.1, 0.1)$ with $q = 0$. Figure 5.51 shows a representative solution \mathbf{m}_3 on a mesh with $N_T = 1280$ triangles. We observe that the magnetization tends to align with the first in-plane axis even though there is no crystalline anisotropy. This is caused by the shape anisotropy of the sample. The energy contribution of the divergence favors alignment along the first in-plane axis. The maximal length of the discrete solution \mathbf{m}_5 on a uniform mesh with $N_T = 20480$ triangles is shown in Figure 5.53. The constraint is not active and the effect of the applied field is strongest at the center of ω . The divergence of the solution \mathbf{m}_{41} on an adaptive mesh with $N_T = 9070$ triangles is shown in Figure 5.52. As expected, we see the characteristic edge and corner singularities.

We estimate the energy $e(\mathbf{m}^*) \approx -0.002533235568891$ by extrapolation of the values obtained on a sequence of uniform meshes. Figure 5.54 shows the error of the energy for uniform and h -adaptive computations. In the uniform case, the convergence is of order $N_T^{-1/2}$. In the adaptive case, we observe an asymptotic behavior of N_T^{-1} , at least up to the point where numerical instabilities arise. Note that this happens at an error level below $1e - 5$. The results of the simulations are given in Table 5.30 and Table 5.31.

Figure 5.55 shows the error in the energy norm, i.e. the error of the divergence in the V -norm, and the error in the L^2 -norm for uniform mesh-refinements. The L^2 error is of higher order. The convergence in the energy norm is of order $N_T^{-1/4}$. Figure 5.56 shows the error estimators η_ℓ^H and $\mu_{\ell,V}^H$ of computation with the h -adaptive algorithm. The mesh refinement apparently resolves the singularities of \mathbf{m}^* , cf.

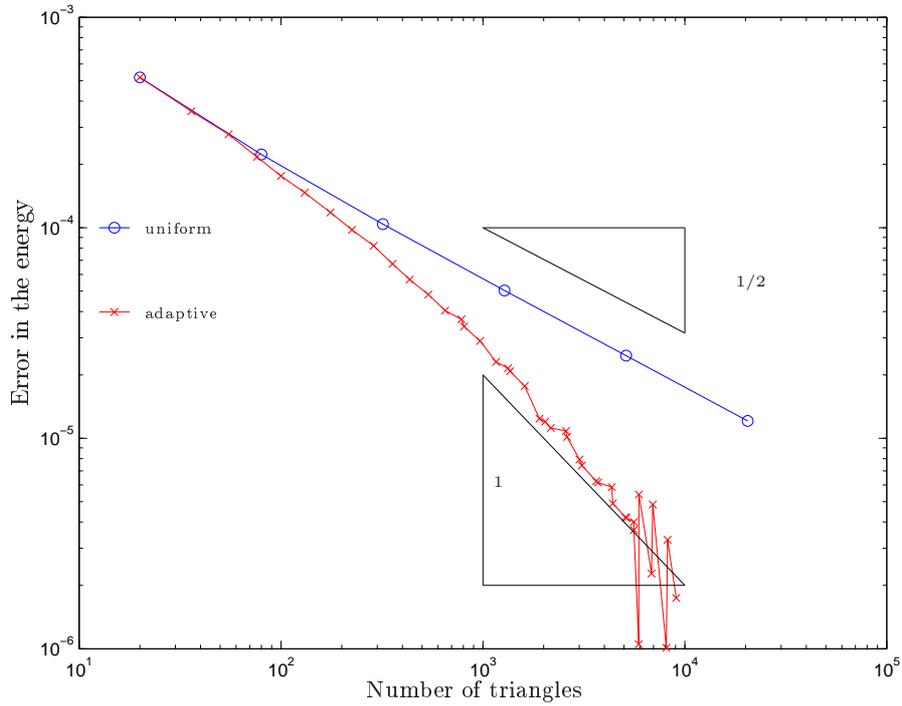


Figure 5.54: Error in the energy $|e(\mathbf{m}^*) - e(\mathbf{m}_\ell)|$ for a sequence of discrete solutions on uniform and h - ε -adaptive meshes for $\omega = (-0.5, 0.5) \times (-0.1, 0.1)$, $\mathbf{f} = 0.15(0.5, 2.0)^T$, and $q = 0$.

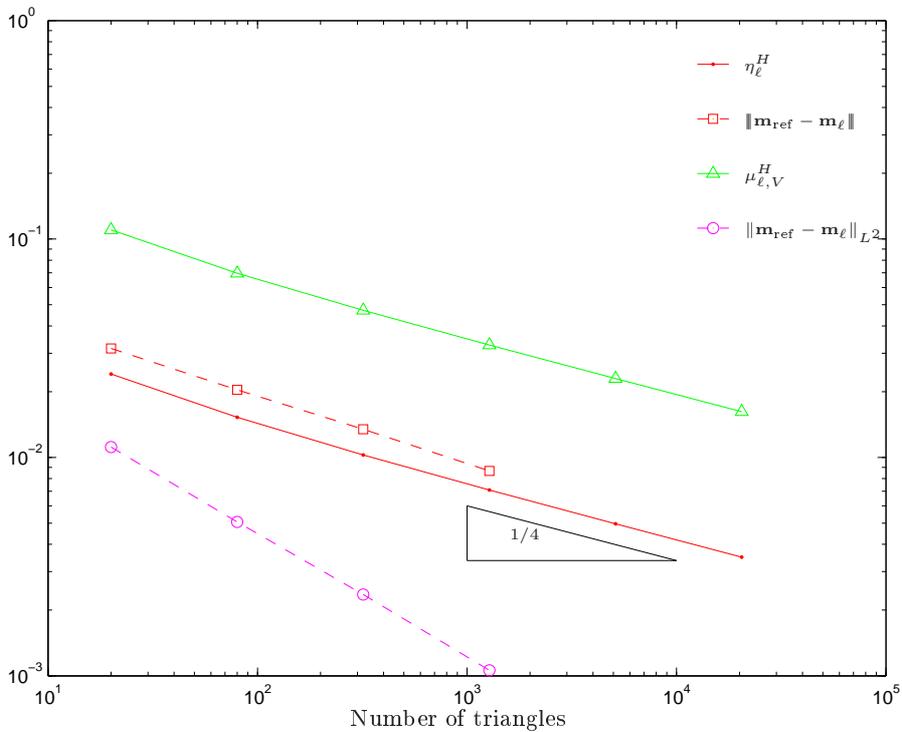


Figure 5.55: Error and error estimators for a sequence of discrete solutions \mathbf{m}_ℓ on uniform meshes of $\omega = (-0.5, 0.5) \times (-0.1, 0.1)$ with $\mathbf{f} = 0.15(0.5, 2.0)^T$ and $q = 0$.

Iteration	N_T	N_D	Energy	η_ℓ	$\mu_{\ell,V}^H$	N_{Newton}
0	20	24	-0.002015	0.024069	0.11008	2
1	36	46	-0.002175	0.019785	0.09042	2
2	55	73	-0.002255	0.017256	0.07847	2
3	76	102	-0.002316	0.015240	0.06994	2
4	100	136	-0.002357	0.013602	0.06307	2
5	131	180	-0.002386	0.012368	0.05687	2
6	176	244	-0.002415	0.011112	0.05113	2
7	225	314	-0.002435	0.010069	0.04648	2
8	288	404	-0.002451	0.009020	0.04258	2
9	357	501	-0.002465	0.008130	0.03835	2
10	434	611	-0.002476	0.007345	0.03527	2
11	537	758	-0.002485	0.006653	0.03253	2
12	650	919	-0.002492	0.006064	0.02990	2
13	783	1109	-0.002496	0.005374	0.02848	2
14	807	1145	-0.002499	0.005129	0.02741	2
15	968	1375	-0.002504	0.004988	0.02537	2
16	1163	1652	-0.002510	0.004612	0.02499	2
17	1331	1894	-0.002511	0.004401	0.02335	2
18	1363	1942	-0.002512	0.004070	0.02202	2
19	1611	2296	-0.002515	0.003811	0.02070	2
20	1907	2720	-0.002520	0.003656	0.02031	2
21	2036	2906	-0.002521	0.003438	0.01960	2
22	2169	3098	-0.002522	0.003200	0.01899	2
23	2579	3684	-0.002522	0.003130	0.01843	2
24	2611	3732	-0.002523	0.003087	0.01727	2
25	3011	4306	-0.002525	0.002910	0.01743	2
26	3095	4432	-0.002525	0.002797	0.01651	2
27	3643	5214	-0.002526	0.002681	0.01557	2
28	3719	5328	-0.002527	0.002591	0.01424	2
29	4347	6228	-0.002527	0.002475	0.01409	2
30	4405	6315	-0.002528	0.002311	0.01333	2
31	5082	7283	-0.002529	0.002309	0.01308	2
32	5142	7373	-0.002529	0.002293	0.01338	2
33	5595	8026	-0.002529	0.005036	0.02123	2
34	5603	8038	-0.002529	0.003001	0.01329	2
35	5913	8488	-0.002532	0.004571	0.01914	2
36	5923	8503	-0.002527	0.002561	0.01174	2
37	6846	9822	-0.002530	0.003014	0.01317	2
38	6944	9969	-0.002528	0.002330	0.01073	2
39	8102	11626	-0.002532	0.003030	0.01295	3
40	8216	11797	-0.002529	0.002365	0.01064	2
41	9070	13031	-0.002531	0.002621	0.01146	3

Table 5.31: Results of calculations with $\omega = (-0.5, 0.5) \times (-0.1, 0.1)$, $\mathbf{f} = 0.15(0.5, 2.0)^T$, $q = 0$, and h -adaptive mesh-refinements: mesh size N_T , space size N_D , energy, error estimators, and number of Newton iterations N_{Newton} .

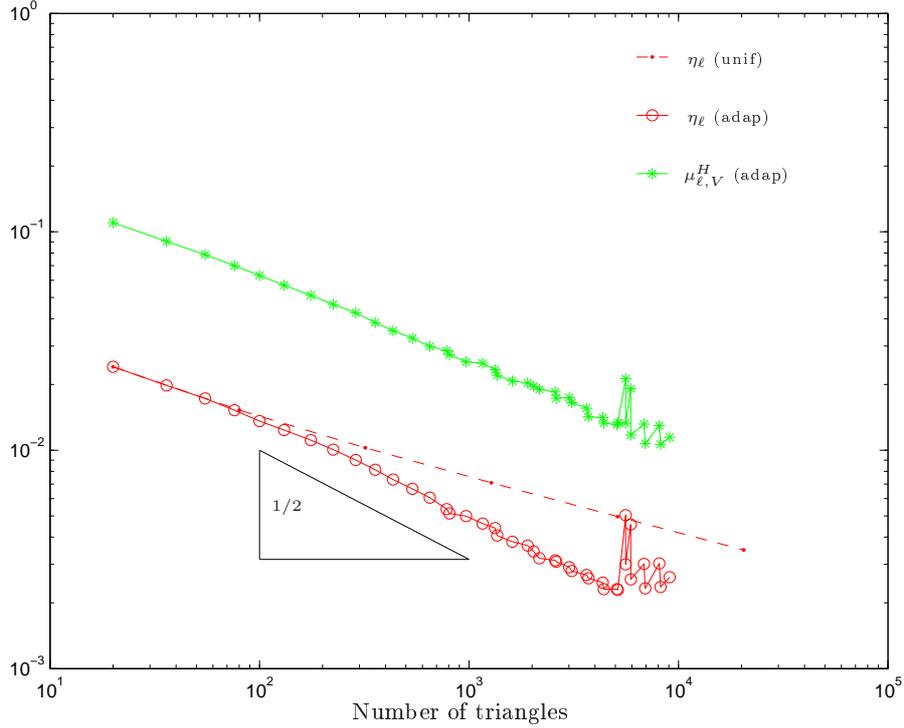


Figure 5.56: Error estimators for a sequence of discrete solutions \mathbf{m}_ℓ on h -adaptive meshes for $\omega = (-0.5, 0.5) \times (-0.1, 0.1)$, $\mathbf{f} = 0.15(0.5, 2.0)^T$, and $q = 0$.

Figure 5.57, which leads to an improved order of convergence of $N_T^{-1/2}$.

In the last experiment, we stick to the same setting as before, i.e. $\omega = (-0.5, 0.5) \times (-0.1, 0.1)$ and $q = 0$, but apply a stronger field $\mathbf{f} = 2(0.5, 2.0)^T$ which effects that the constraint is active, see Figure 5.60. Figure 5.58 shows the discrete solution \mathbf{m}_3 on a uniform mesh with $N_T = 1280$ triangles and $N_D = 1872$ degrees of freedom, the divergence of the solution \mathbf{m}_{29} on an h - ε -adaptive mesh with $N_T = 5668$ triangles is shown in Figure 5.59.

The performance of the uniform algorithm with $\varepsilon = h$ can be seen in Figure 5.61. We observe convergence with order $N_T^{-1/4}$. The error in the L^2 norm is of higher order. The h - ε -adaptive algorithm performs better and resolves the error component of the divergence at almost linear rate of $N_T^{-1/2}$. Also the resolution of the penalty error is very good, the empirically reliable error estimator η_ℓ^ε shows also almost convergence with order $N_T^{-1/2}$. The results of the uniform and the h - ε -adaptive computations are given in Table 5.32 and Table 5.33. As can be seen in Figure 5.63, the adaptive algorithm leads to strong refinements of the mesh to the edges, see Figure 5.63. The penalty parameter is refined stronger at the center of ω , where the magnetization \mathbf{m} is smooth but well aligned with the applied field, see Figure 5.64.

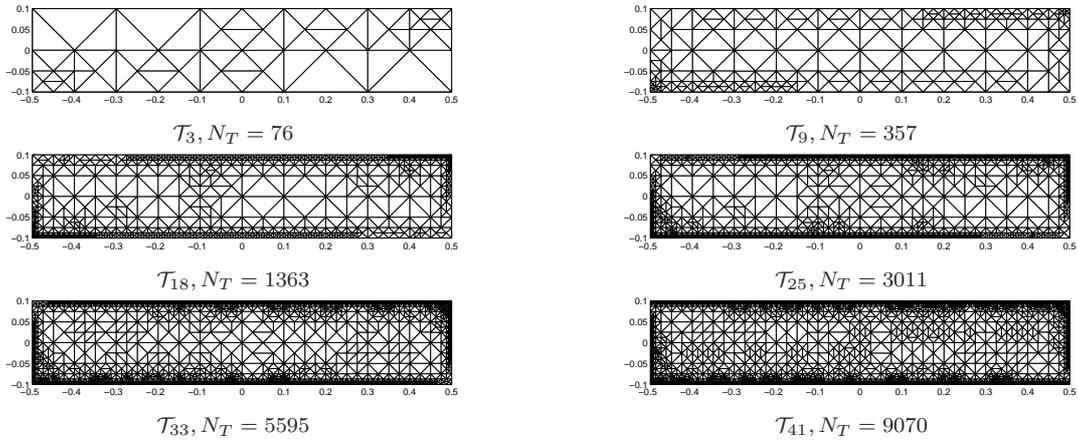


Figure 5.57: Sequence of h -adaptively generated meshes of $\omega = (-0.5, 0.5) \times (-0.1, 0.1)$ with $\mathbf{f} = 0.15(0.5, 2.0)^T$ and $q = 0$.

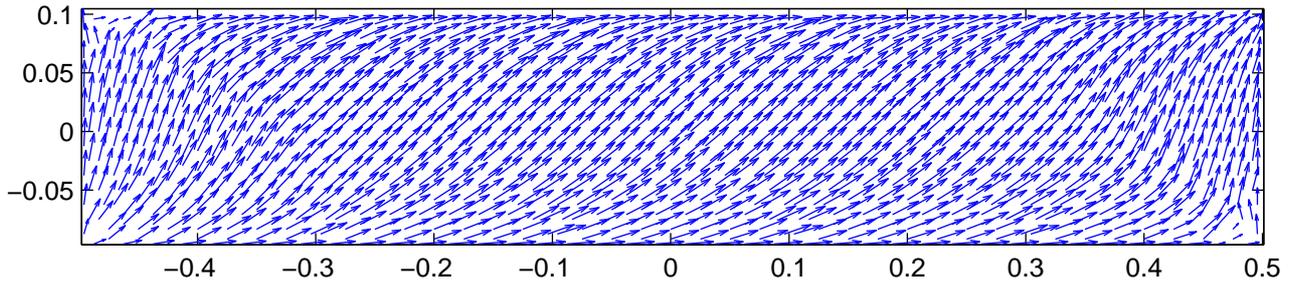


Figure 5.58: Discrete magnetization for $\omega = (-0.5, 0.5) \times (-0.1, 0.1)$, $\mathbf{f} = 2(0.5, 2.0)^T$ and $q = 0$ on a mesh with $N_T = 1280$ triangles and $N_D = 1872$ degrees of freedom.

Iteration	N_T	N_D	Energy	$\eta_{\ell, V}^H$	$\eta_{\ell}^{\varepsilon}$	$\mu_{\ell, V}^H$	N_{Newton}
0	20	24	-0.3134	0.2803	0.12543	1.2819	7
1	80	108	-0.3537	0.1786	0.10373	0.8136	7
2	320	456	-0.3680	0.1183	0.07999	0.5419	7
3	1280	1872	-0.3737	0.0810	0.05997	0.3725	12
4	5120	7584	-0.3762	0.0564	0.04391	0.2600	11
5	20480	30528	-0.3774	0.0396	0.03167	0.1830	14
6	81920	122496	-0.3784	-	-	-	14

Table 5.32: Results of calculations with $\omega = (-0.5, 0.5) \times (-0.1, 0.1)$, $\mathbf{f} = 2(0.5, 2.0)^T$, $q = 0$, and h - ε -uniform refinements: mesh size N_T , space size N_D , energy, error estimators, and number of Newton iterations N_{Newton} .

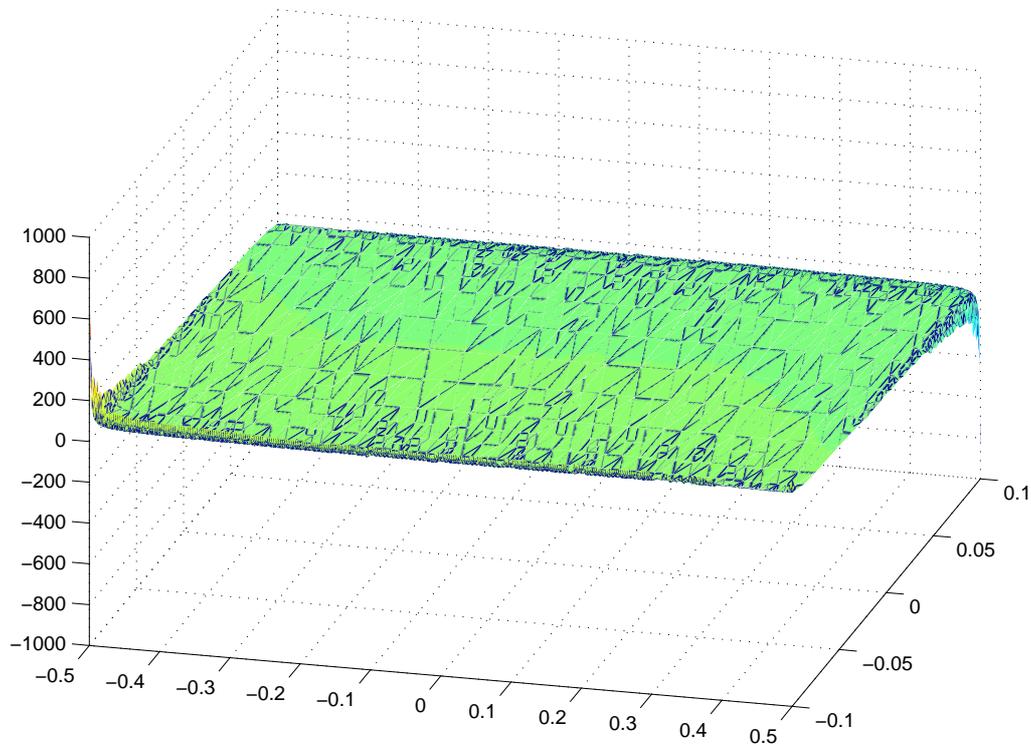


Figure 5.59: Divergence of the adaptively generated discrete solution \mathbf{m}_{29} for $\omega = (-0.5, 0.5) \times (-0.1, 0.1)$, $\mathbf{f} = 2(0.5, 2.0)^T$, and $q = 0$ on an h - ε -adaptive mesh with $N_T = 5668$ triangles and $N_D = 8153$ degrees of freedom.

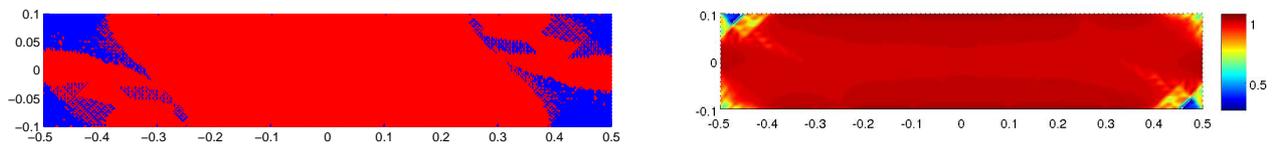


Figure 5.60: Left: Penalized (red) and not penalized (blue) elements of the solution \mathbf{m}_5 for $\omega = (-0.5, 0.5) \times (-0.1, 0.1)$, $\mathbf{f} = 2(0.5, 2.0)^T$, and $q = 0$. Right: Maximal length of the magnetization \mathbf{m}_5 on each element. The uniform mesh has $N_T = 20480$ triangles and $N_D = 30528$ degrees of freedom.

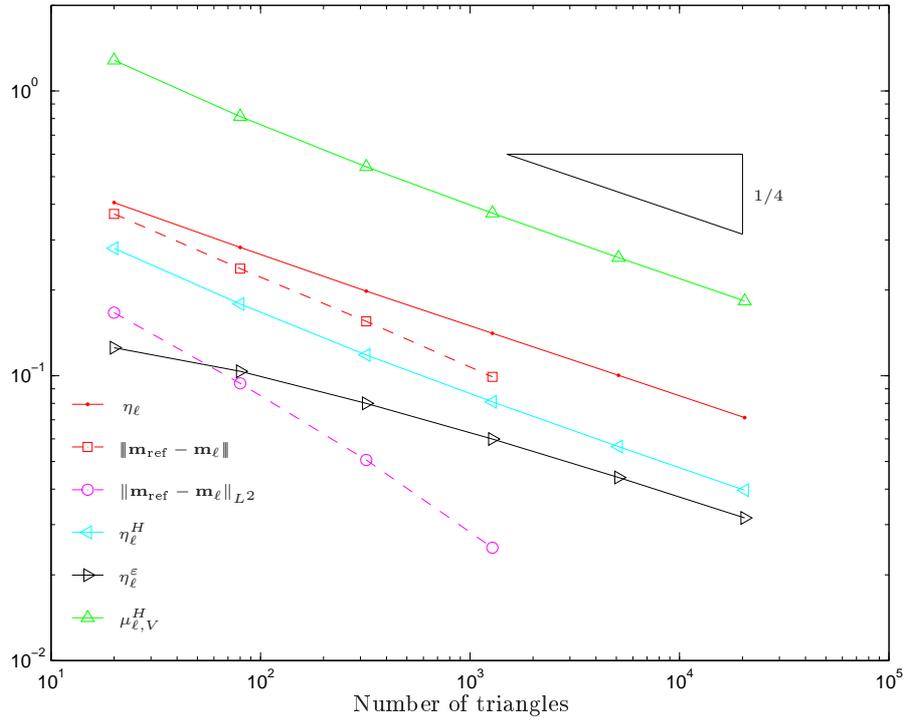


Figure 5.61: Error and error estimators for a sequence of discrete solutions \mathbf{m}_ℓ on h - ε -uniform meshes of $\omega = (-0.5, 0.5) \times (-0.1, 0.1)$ with $\mathbf{f} = 2(0.5, 2.0)^T$ and $q = 0$.

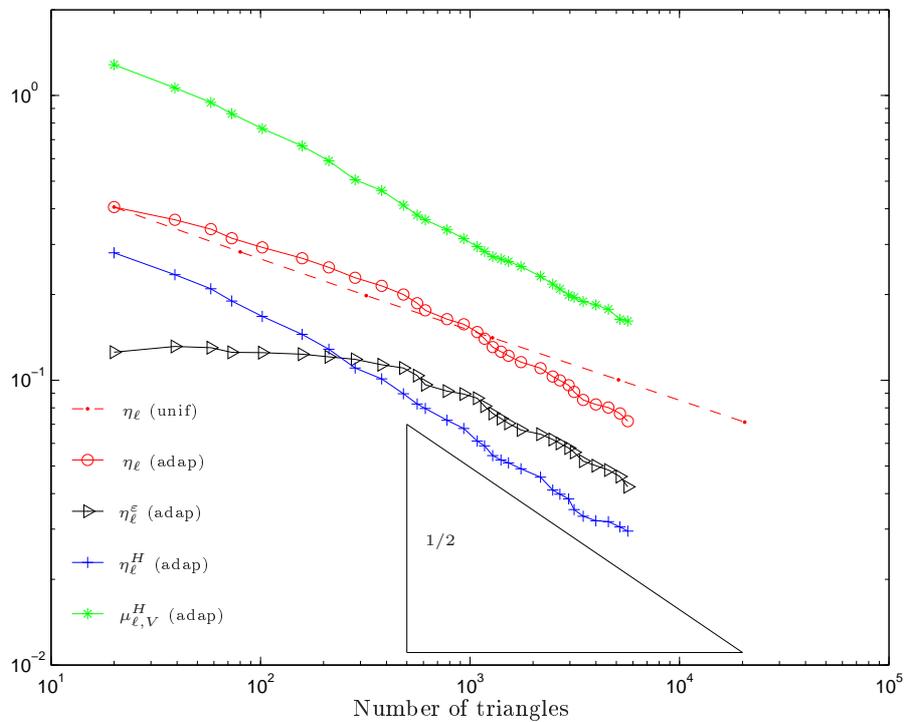


Figure 5.62: Error estimators for a sequence of discrete solutions \mathbf{m}_ℓ on h - ε -adaptive meshes of $\omega = (-0.5, 0.5) \times (-0.1, 0.1)$ with $\mathbf{f} = 2(0.5, 2.0)^T$ and $q = 0$.

Iteration	N_T	N_D	Energy	$\eta_{\ell,V}^H$	$\eta_{\ell}^{\varepsilon}$	$\mu_{\ell,V}^H$	N_{Newton}
0	20	24	-0.3134	0.28031	0.12543	1.2819	7
1	39	50	-0.3364	0.23480	0.13142	1.0633	6
2	58	77	-0.3483	0.20954	0.13013	0.9451	5
3	73	98	-0.3574	0.18977	0.12537	0.8628	5
4	102	139	-0.3654	0.16766	0.12503	0.7649	5
5	158	219	-0.3749	0.14484	0.12347	0.6646	5
6	212	296	-0.3798	0.12841	0.12078	0.5895	5
7	283	396	-0.3815	0.11045	0.11842	0.5059	6
8	379	534	-0.3821	0.10108	0.11335	0.4636	5
9	481	678	-0.3838	0.08964	0.11036	0.4116	4
10	560	790	-0.3833	0.08254	0.10381	0.3799	6
11	612	864	-0.3820	0.07957	0.09642	0.3660	5
12	777	1101	-0.3818	0.07234	0.09155	0.3373	5
13	935	1327	-0.3834	0.06771	0.08945	0.3140	4
14	1083	1540	-0.3820	0.06117	0.08643	0.2944	5
15	1174	1670	-0.3812	0.05880	0.08107	0.2833	9
16	1284	1826	-0.3805	0.05435	0.07645	0.2717	4
17	1407	2001	-0.3813	0.05251	0.07337	0.2666	4
18	1527	2177	-0.3796	0.05127	0.07056	0.2612	5
19	1751	2503	-0.3796	0.04885	0.06687	0.2508	8
20	2173	3108	-0.3798	0.04568	0.06460	0.2317	8
21	2484	3555	-0.3806	0.04117	0.06191	0.2175	9
22	2686	3847	-0.3799	0.04480	0.06000	0.2093	6
23	2961	4242	-0.3804	0.03830	0.05779	0.1990	7
24	3144	4510	-0.3795	0.03513	0.05584	0.1954	7
25	3471	4983	-0.3792	0.03330	0.05195	0.1890	7
26	3992	5737	-0.3793	0.03211	0.05009	0.1839	8
27	4579	6584	-0.3792	0.03186	0.04837	0.1775	8
28	5199	7473	-0.3792	0.03053	0.04592	0.1636	8
29	5668	8153	-0.3793	0.02950	0.04227	0.1614	10

Table 5.33: Results of calculations with $\omega = (-0.5, 0.5) \times (-0.1, 0.1)$, $\mathbf{f} = 2(0.5, 2.0)^T$, $q = 0$, and h - ε -adaptive refinements: mesh size N_T , space size N_D , energy, error estimators, and number of Newton iterations N_{Newton} .

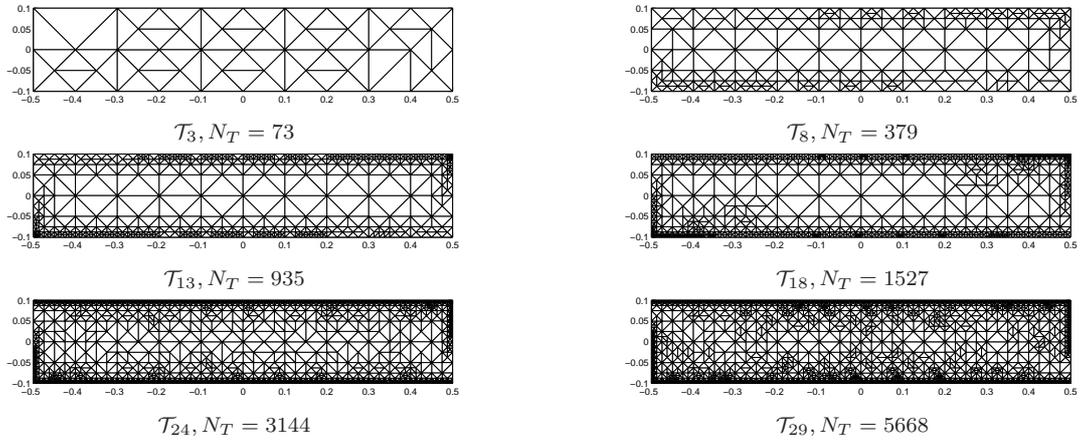


Figure 5.63: Sequence of h - ε -adaptively generated meshes of $\omega = (-0.5, 0.5) \times (-0.1, 0.1)$ with $\mathbf{f} = 2(0.5, 2.0)^T$ and $q = 0$.

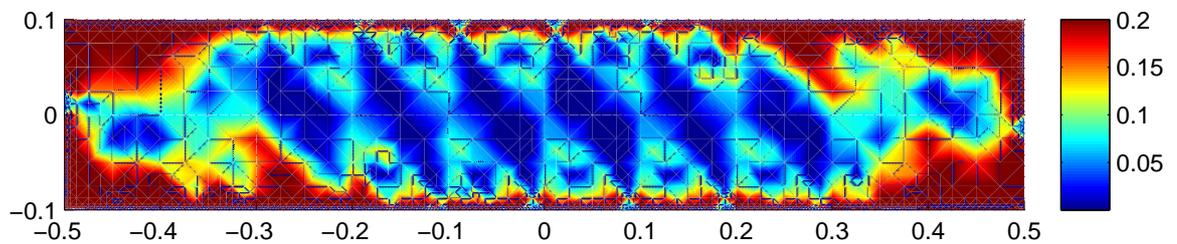


Figure 5.64: Adaptive penalty parameter on the mesh \mathcal{T}_{29} of $\omega = (-0.5, 0.5) \times (-0.1, 0.1)$ with $N_T = 5668$ triangles, $\mathbf{f} = 2(0.5, 2.0)^T$, and $q = 0$.

List of Figures

1.1	Typical solution of a problem with smooth data; left: Magnetization, right: Divergence	6
1.2	Estimated error for non-smooth data. The adaptive algorithm leads to convergence at a higher order than the uniform approach.	7
2.1	Landau state of a soft ferromagnetic film with diameter of several μm . The magnetization (blue) is nearly constant on the domains. The domain walls depicted here as lines are continuous transitions of the magnetization at nm scale.	11
3.1	Each interior edge E belongs to precisely two triangles T_+ and T_- . The points opposite of E are denoted by P_+ and P_- , respectively.	29
3.2	Red (uniform) refinement of a triangle (top). Uniform refinement of a sample mesh (bottom).	49
3.3	Local mesh-width functions h_ℓ and ϱ_ℓ	51
3.4	Green (top) and blue (bottom) refinement rules. In any case the longest edge is split first.	52
3.5	Illustration of red-green-blue refinement: Assume the right central triangle was marked for refinement. Then all of its edges are marked, hence the triangle is refined red. The hanging node on the top triangle effects that additionally its longest edge is marked as well. This leads to blue refinement. The hanging node on the bottom triangle is already on the longest edge. Therefore, it is refined green.	53
4.1	In a <code>Mesh</code> object, all triangles, edges and vertices are stored in a <code>list</code> container. Consider two triangles stored as <code>MeshTriangle</code> objects. If they share an edge, the geometric relation between them is provided with the help of containers of <code>MeshEdge*</code> , of which one is set to store the address of the global <code>MeshEdge</code> object (red). Also pointers to the global <code>MeshPoint</code> objects (green) are stored in the <code>MeshTriangle</code> objects.	59
4.2	A simple mesh with 5 vertices, 4 triangles, and 8 edges.	63
4.3	3 point (left) and 7 point (right) Gauss quadrature rules on the reference triangle.	72
4.4	10 point Newton-Cotes quadrature rule on the reference triangle.	73
4.5	Typical structure of a hierarchical matrix. Inadmissible blocks (red) are stored uncompressed, admissible blocks (green) are approximated by low-rank matrices.	75

5.1	Initial mesh with 16 triangles and 20 interior edges for simulations where the domain is the unit square $\omega = (-0.5, 0.5)^2$ centered in the plane.	86
5.2	Discrete magnetization for $\mathbf{f} = (0.1, 0.1)^T$ and $q = 1$ on a uniform mesh of the domain $\omega = (-0.5, 0.5)^2$ with $N_T = 1024$ triangles and $N_D = 1504$ degrees of freedom.	87
5.3	Divergence of the solution \mathbf{m}_{14} for $\mathbf{f} = (0.1, 0.1)^T$ and $q = 1$ on an h -adaptively generated mesh of the simulation domain $\omega = (-0.5, 0.5)^2$ with $N_T = 2058$ triangles and $N_D = 2976$ degrees of freedom.	88
5.4	Error and error estimators for a sequence of discrete solutions \mathbf{m}_ℓ on uniform meshes with $\omega = (-0.5, 0.5)^2$, $\mathbf{f} = (0.1, 0.1)^T$, and $q = 1$	89
5.5	Sequence of h -adaptively generated meshes with $\omega = (-0.5, 0.5)^2$, $\mathbf{f} = (0.1, 0.1)^T$, and $q = 1$	91
5.6	Error estimators for h -adaptive and uniform mesh-refinements. The field $\mathbf{f} = (0.1, 0.1)^T$ is applied to the sample $\omega = (0.5, 0.5)^2$ with anisotropy parameter $q = 1$	92
5.7	Error in the energy $ e(\mathbf{m}^*) - e(\mathbf{m}_\ell) $ for a sequence of discrete solutions on uniform and h -adaptive meshes with $\omega = (-0.5, 0.5)^2$, $\mathbf{f} = (0.1, 0.1)^T$, and $q = 1$	93
5.8	Discrete magnetization for $\omega = (-0.5, 0.5)^2$, $\mathbf{f} = (0.6, 0.6)^T$, and $q = 1$ on a mesh with $N_T = 1024$ triangles and $N_D = 1504$ degrees of freedom.	94
5.9	Divergence of the solution \mathbf{m}_8 for $\omega = (-0.5, 0.5)^2$, $\mathbf{f} = (0.6, 0.6)^T$, and $q = 1$ on an h -adaptively generated mesh with $N_T = 11555$ triangles and $N_D = 16856$ degrees of freedom.	95
5.10	Left: Penalized (red) and not penalized (blue) elements of the solution \mathbf{m}_4 for $\omega = (-0.5, 0.5)^2$, $f = (0.6, 0.6)^T$, and $q = 1$. The uniform mesh has $N_T = 16384$ triangles. Right: Elementwise maximal length of \mathbf{m}_4	95
5.11	Error and error estimators for a sequence of discrete solutions \mathbf{m}_ℓ on uniform meshes for $\omega = (-0.5, 0.5)^2$, $\mathbf{f} = (0.6, 0.6)^T$, and $q = 1$	96
5.12	Sequence of h -adaptively generated meshes for $\omega = (-0.5, 0.5)^2$, $\mathbf{f} = (0.6, 0.6)^T$, and $q = 1$	97
5.13	Error estimators for h -adaptive and uniform mesh-refinements. The field $\mathbf{f} = (0.6, 0.6)^T$ is applied to the sample $\omega = (0.5, 0.5)^2$ with anisotropy parameter $q = 1$	98
5.14	Error in the energy $ e(\mathbf{m}^*) - e(\mathbf{m}_\ell) $ for a sequence of discrete solutions on uniform and h -adaptive meshes for $\omega = (-0.5, 0.5)^2$, $\mathbf{f} = (0.6, 0.6)^T$, and $q = 1$	99
5.15	Discrete magnetization for $\omega = (-0.5, 0.5)^2$, discontinuous \mathbf{f} , and $q = 1$ on a mesh with $N_T = 4096$ triangles and $N_D = 6080$ degrees of freedom.	100
5.16	Discrete magnetization for $\omega = (-0.5, 0.5)^2$, discontinuous \mathbf{f} , and $q = 1$ on an h -adaptively generated mesh with $N_T = 1760$ triangles and $N_D = 2589$ degrees of freedom.	101
5.17	Divergence of the discrete solution \mathbf{m}_{20} for $\omega = (-0.5, 0.5)^2$, discontinuous \mathbf{f} , and $q = 1$ on an h -adaptively generated mesh with $N_T = 10485$ triangles and $N_D = 15588$ degrees of freedom.	102

5.18	Left: Penalized (red) and not penalized (blue) elements of the solution \mathbf{m}_5 for $\omega = (-0.5, 0.5)^2$, discontinuous \mathbf{f} , and $q = 1$. Right: Maximal length of the magnetization \mathbf{m}_5 on each element. The uniform mesh has $N_T = 16384$ triangles and $N_D = 24448$ degrees of freedom.	102
5.19	Maximal length of the magnetization \mathbf{m}_5 on each element for $\omega = (-0.5, 0.5)^2$, discontinuous \mathbf{f} , and $q = 1$. The uniform mesh has $N_T = 16384$ triangles and $N_D = 24448$ degrees of freedom.	103
5.20	Error in the energy $ e(\mathbf{m}^*) - e(\mathbf{m}_\ell) $ for a sequence of discrete solutions on uniform and h -adaptive meshes for $\omega = (-0.5, 0.5)^2$, discontinuous \mathbf{f} , and $q = 1$	104
5.21	Error and error estimators for a sequence of discrete solutions \mathbf{m}_ℓ on uniform meshes for $\omega = (-0.5, 0.5)^2$, discontinuous \mathbf{f} , and $q = 1$	105
5.22	Sequence of h -adaptively generated meshes for $\omega = (-0.5, 0.5)^2$, discontinuous \mathbf{f} , and $q = 1$	106
5.23	Error estimators for h -adaptive and uniform mesh-refinements. A discontinuous field \mathbf{f} is applied to the sample $\omega = (0.5, 0.5)^2$ with anisotropy parameter $q = 1$	107
5.24	Left: penalized (red) and not penalized (blue) elements of a uniform mesh of $\omega = (-0.5, 0.5)^2$ with $N_T = 65536$ triangles. The penalty parameter is $\varepsilon = 0.005$, the applied field is $\mathbf{f} = (0.8, 0.8)^T$. Right: elementwise maximal length of the discrete magnetization.	107
5.25	Error in the energy norm for a series of experiments with uniform meshes of $\omega = (-0.5, 0.5)^2$, applied field $\mathbf{f} = (0.8, 0.8)^T$, $q = 1$ and fixed values of ε	108
5.26	Error in the energy norm for a series of experiments with uniform meshes of $\omega = (-0.5, 0.5)^2$, applied field $\mathbf{f} = (3, 3)^T$, $q = 1$ and fixed values of ε	108
5.27	Penalized (red) and not penalized (blue) elements of the solution \mathbf{m}_3 for $\omega = (-0.5, 0.5)^2$, $\mathbf{f} = (1, 1)^T$, $\varepsilon = 1e - 4$, and $q = 1$. The uniform mesh has $N_T = 4096$ triangles.	111
5.28	Length of the discrete solution \mathbf{m}_3 on a uniform mesh of $\omega = (-0.5, 0.5)^2$ with $N_T = 4096$ triangles, $\mathbf{f} = (1, 1)^T$, $q = 1$, and $\varepsilon = 1e - 4$	112
5.29	Error and error estimators for three different uniform meshes and varying values of ε . We apply the field $\mathbf{f} = (1, 1)^T$ to the sample $\omega = (-0.5, 0.5)^2$ with $q = 1$	114
5.30	Error in the energy for three different uniform meshes and varying values of ε . We apply the field $\mathbf{f} = (1, 1)^T$ to the sample $\omega = (-0.5, 0.5)^2$ with $q = 1$	114
5.31	Error for different choices of penalty parameter $\varepsilon = h^\alpha$. We apply the field $\mathbf{f} = (2, 2)^T$ to the sample $\omega = (-0.5, 0.5)$ with $q = 1$	116
5.32	Error estimators for different choices of penalty parameter $\varepsilon = h^\alpha$. We apply the field $\mathbf{f} = (2, 2)^T$ to the sample $\omega = (-0.5, 0.5)$ with $q = 1$	116
5.33	Error for different choices of penalty parameter $\varepsilon = h^\alpha$. The field $\mathbf{f} = (0.8, 0.8)^T$ is applied to the sample $\omega = (-0.5, 0.5)^2$ with $q = 1$	118
5.34	Error estimators for different choices of penalty parameter $\varepsilon = h^\alpha$. The field $\mathbf{f} = (0.8, 0.8)^T$ is applied to the sample $\omega = (-0.5, 0.5)^2$ with $q = 1$	119

5.35	Discrete magnetization for singular \mathbf{f} and $q = 1$ on a mesh of $\omega = (-0.5, 0.5)^2$ with $N_T = 4096$ triangles and $N_D = 6080$ degrees of freedom.	122
5.36	Discrete magnetization \mathbf{m}_{20} for singular \mathbf{f} and $q = 1$ on an h - ε -adaptively generated mesh of $\omega = (-0.5, 0.5)^2$ with $N_T = 1811$ triangles and $N_D = 2671$ degrees of freedom.	123
5.37	Divergence of the h - ε -adaptively generated discrete solution \mathbf{m}_{34} for singular \mathbf{f} and $q = 1$ on a mesh of $\omega = (-0.5, 0.5)^2$ with $N_T = 9132$ triangles and $N_D = 13559$ degrees of freedom.	124
5.38	Left: Penalized (red) and not penalized (blue) elements of the solution \mathbf{m}_5 for singular applied field \mathbf{f} and $q = 1$. Right: Maximal length of the magnetization \mathbf{m}_5 on each element. The uniform mesh of $\omega = (-0.5, 0.5)^2$ has $N_T = 16384$ triangles and $N_D = 24448$ degrees of freedom.	124
5.39	Error estimators for $\omega = (-0.5, 0.5)^2$, singular \mathbf{f} , $q = 1$ and h - ε -uniform refinements.	126
5.40	Sequence of h - ε -adaptively generated meshes of $\omega = (-0.5, 0.5)^2$ with singular applied field \mathbf{f} and $q = 1$	127
5.41	Error estimators for h - ε -adaptive refinements with $\omega = (-0.5, 0.5)^2$, singular applied field \mathbf{f} and $q = 1$	128
5.42	Left: Maximal length of the h - ε -adaptively generated solution \mathbf{m}_{34} for $\omega = (-0.5, 0.5)^2$, singular applied field \mathbf{f} , and anisotropy parameter $q = 1$. Right: Inverse of the adaptively generated penalty parameter $1/\varepsilon_\ell$	128
5.43	Discrete magnetization for $\omega = (-0.5, 0.5) \times (-0.1, 0.1)$, $\mathbf{f} = 10(0.5, 2.0)^T$ and $q = 1e4$ on a uniform mesh with $N_T = 1280$ triangles and $N_D = 1872$ degrees of freedom.	129
5.44	Divergence of the h - ε -adaptively generated discrete solution \mathbf{m}_{16} for $\omega = (-0.5, 0.5) \times (-0.1, 0.1)$, $\mathbf{f} = 10(0.5, 2.0)^T$, and $q = 1e4$ on a mesh with $N_T = 14662$ triangles and $N_D = 21675$ degrees of freedom.	129
5.45	Left: Penalized (red) and not penalized (blue) elements of the solution \mathbf{m}_5 for $\omega = (-0.5, 0.5) \times (-0.1, 0.1)$, $\mathbf{f} = 10(0.5, 2.0)^T$, and $q = 1e4$. Right: Maximal length of the magnetization \mathbf{m}_5 on each element. The uniform mesh has $N_T = 20480$ triangles and $N_D = 30528$ degrees of freedom.	129
5.46	Error in the energy $ e(\mathbf{m}^*) - e(\mathbf{m}_\ell) $ for a sequence of discrete solutions on h - ε -uniform and h - ε -adaptive meshes of $\omega = (-0.5, 0.5) \times (-0.1, 0.1)$ with $\mathbf{f} = 10(0.5, 2.0)^T$, and $q = 1e4$	130
5.47	Error and error estimators for a sequence of discrete solutions \mathbf{m}_ℓ on h - ε -uniform meshes of $\omega = (-0.5, 0.5) \times (-0.1, 0.1)$ with $\mathbf{f} = 10(0.5, 2.0)^T$ and $q = 1e4$	131
5.48	Error estimators for a sequence of discrete solutions \mathbf{m}_ℓ on h - ε -adaptive meshes of $\omega = (-0.5, 0.5) \times (-0.1, 0.1)$ with $\mathbf{f} = 10(0.5, 2.0)^T$ and $q = 1e4$	131
5.49	Sequence of h - ε -adaptively generated meshes of $\omega = (-0.5, 0.5) \times (-0.1, 0.1)$ with $\mathbf{f} = 10(0.5, 2.0)^T$ and $q = 1e4$	132
5.50	Adaptive penalty parameter on the mesh \mathcal{T}_{16} of $\omega = (-0.5, 0.5) \times (-0.1, 0.1)$ with $N_T = 14662$ triangles, $\mathbf{f} = 10(0.5, 2.0)^T$, and $q = 1e4$	133

5.51	Discrete magnetization for $\omega = (-0.5, 0.5) \times (-0.1, 0.1)$, $\mathbf{f} = 0.15(0.5, 2.0)^T$, and $q = 0$ on a uniform mesh with $N_T = 1280$ triangles and $N_D = 1872$ degrees of freedom.	133
5.52	Divergence of the h - ε -adaptively generated discrete solution \mathbf{m}_{41} for $\omega = (-0.5, 0.5) \times (-0.1, 0.1)$, $\mathbf{f} = 0.15(0.5, 2.0)^T$, and $q = 0$ on a mesh with $N_T = 9070$ triangles and $N_D = 13031$ degrees of freedom.	133
5.53	Maximal length of the magnetization \mathbf{m}_5 on each element. The uniform mesh of $\omega = (-0.5, 0.5) \times (-0.1, 0.1)$ has $N_T = 20480$ triangles and $N_D = 30528$ degrees of freedom. The field $\mathbf{f} = 0.15(0.5, 2.0)^T$ is applied to the soft sample with $q = 0$	134
5.54	Error in the energy $ e(\mathbf{m}^*) - e(\mathbf{m}_\ell) $ for a sequence of discrete solutions on uniform and h - ε -adaptive meshes for $\omega = (-0.5, 0.5) \times (-0.1, 0.1)$, $\mathbf{f} = 0.15(0.5, 2.0)^T$, and $q = 0$	135
5.55	Error and error estimators for a sequence of discrete solutions \mathbf{m}_ℓ on uniform meshes of $\omega = (-0.5, 0.5) \times (-0.1, 0.1)$ with $\mathbf{f} = 0.15(0.5, 2.0)^T$ and $q = 0$	135
5.56	Error estimators for a sequence of discrete solutions \mathbf{m}_ℓ on h -adaptive meshes for $\omega = (-0.5, 0.5) \times (-0.1, 0.1)$, $\mathbf{f} = 0.15(0.5, 2.0)^T$, and $q = 0$	137
5.57	Sequence of h -adaptively generated meshes of $\omega = (-0.5, 0.5) \times (-0.1, 0.1)$ with $\mathbf{f} = 0.15(0.5, 2.0)^T$ and $q = 0$	138
5.58	Discrete magnetization for $\omega = (-0.5, 0.5) \times (-0.1, 0.1)$, $\mathbf{f} = 2(0.5, 2.0)^T$ and $q = 0$ on a mesh with $N_T = 1280$ triangles and $N_D = 1872$ degrees of freedom.	138
5.59	Divergence of the adaptively generated discrete solution \mathbf{m}_{29} for $\omega = (-0.5, 0.5) \times (-0.1, 0.1)$, $\mathbf{f} = 2(0.5, 2.0)^T$, and $q = 0$ on an h - ε -adaptive mesh with $N_T = 5668$ triangles and $N_D = 8153$ degrees of freedom.	139
5.60	Left: Penalized (red) and not penalized (blue) elements of the solution \mathbf{m}_5 for $\omega = (-0.5, 0.5) \times (-0.1, 0.1)$, $\mathbf{f} = 2(0.5, 2.0)^T$, and $q = 0$. Right: Maximal length of the magnetization \mathbf{m}_5 on each element. The uniform mesh has $N_T = 20480$ triangles and $N_D = 30528$ degrees of freedom.	139
5.61	Error and error estimators for a sequence of discrete solutions \mathbf{m}_ℓ on h - ε -uniform meshes of $\omega = (-0.5, 0.5) \times (-0.1, 0.1)$ with $\mathbf{f} = 2(0.5, 2.0)^T$ and $q = 0$	140
5.62	Error estimators for a sequence of discrete solutions \mathbf{m}_ℓ on h - ε -adaptive meshes of $\omega = (-0.5, 0.5) \times (-0.1, 0.1)$ with $\mathbf{f} = 2(0.5, 2.0)^T$ and $q = 0$	140
5.63	Sequence of h - ε -adaptively generated meshes of $\omega = (-0.5, 0.5) \times (-0.1, 0.1)$ with $\mathbf{f} = 2(0.5, 2.0)^T$ and $q = 0$	142
5.64	Adaptive penalty parameter on the mesh \mathcal{T}_{29} of $\omega = (-0.5, 0.5) \times (-0.1, 0.1)$ with $N_T = 5668$ triangles, $\mathbf{f} = 2(0.5, 2.0)^T$, and $q = 0$	142

List of Tables

4.1	A 3 point Gauss quadrature rule on the reference triangle $T_{\text{ref}} = \text{conv}\{(0, 0), (1, 0), (0, 1)\}$.	71
4.2	A 7 point Gauss quadrature rule on the reference triangle $T_{\text{ref}} = \text{conv}\{(0, 0), (1, 0), (0, 1)\}$.	72
4.3	A 10 point Newton-Cotes quadrature rule on the reference triangle $T_{\text{ref}} = \text{conv}\{(0, 0), (1, 0), (0, 1)\}$.	73
5.1	Results of the calculations with $\mathbf{f} = (0.1, 0.1)^T$, $q = 1$, $\omega = (-0.5, 0.5)^T$, and uniform mesh-refinements: mesh size N_T , space size N_D , energy, error estimators, and number of Newton iterations N_{Newton} .	86
5.2	Results of calculations with $\omega = (-0.5, 0.5)^2$, $\mathbf{f} = (0.1, 0.1)^T$, $q = 1$, and h -adaptive mesh-refinements: mesh size N_T , space size N_D , energy, error estimators, and number of Newton iterations N_{Newton} .	92
5.3	Results of calculations with $\omega = (-0.5, 0.5)^2$, $\mathbf{f} = (0.6, 0.6)^T$, $q = 1$, and uniform mesh-refinements: mesh size N_T , space size N_D , energy, error estimators, and number of Newton iterations N_{Newton} .	93
5.4	Results of calculations with $\omega = (-0.5, 0.5)^2$, $\mathbf{f} = (0.6, 0.6)^T$, and $q = 1$: mesh size N_T , space size N_D , energy, error estimators, and number of Newton iterations N_{Newton} .	98
5.5	Results of calculations with $\omega = (-0.5, 0.5)^2$, discontinuous \mathbf{f} , $q = 1$, and uniform mesh-refinements: mesh size N_T , space size N_D , energy, error estimators, and number of Newton iterations N_{Newton} .	103
5.6	Results of calculations with $\omega = (-0.5, 0.5)^2$, discontinuous \mathbf{f} , $q = 1$, and h -adaptive mesh-refinements: mesh size N_T , space size N_D , energy, error estimators, and number of Newton iterations N_{Newton} .	104
5.7	Results of calculations with $\omega = (-0.5, 0.5)^2$, $\mathbf{f} = (0.8, 0.8)^T$, $q = 1$, uniform mesh-refinements, and $\varepsilon = 0.1$: mesh size N_T , space size N_D , energy, error estimators, and number of Newton iterations N_{Newton} .	109
5.8	Results of calculations with $\omega = (-0.5, 0.5)^2$, $\mathbf{f} = (0.8, 0.8)^T$, $q = 1$, uniform mesh-refinements, and $\varepsilon = 0.05$: mesh size N_T , space size N_D , energy, error estimators, and number of Newton iterations N_{Newton} .	109
5.9	Results of calculations with $\omega = (-0.5, 0.5)^2$, $\mathbf{f} = (0.8, 0.8)^T$, $q = 1$, uniform mesh-refinements, and $\varepsilon = 0.01$: mesh size N_T , space size N_D , energy, error estimators, and number of Newton iterations N_{Newton} .	109

5.10	Results of calculations with $\omega = (-0.5, 0.5)^2$, $\mathbf{f} = (0.8, 0.8)^T$, $q = 1$, uniform mesh-refinements, and $\varepsilon = 0.005$: mesh size N_T , space size N_D , energy, error estimators, and number of Newton iterations N_{Newton}	109
5.11	Results of calculations with $\omega = (-0.5, 0.5)^2$, $\mathbf{f} = (3, 3)^T$, $q = 1$, uniform mesh-refinements, and $\varepsilon = 0.1$: mesh size N_T , space size N_D , energy, error estimators, and number of Newton iterations N_{Newton}	110
5.12	Results of calculations with $\omega = (-0.5, 0.5)^2$, $\mathbf{f} = (3, 3)^T$, $q = 1$, uniform mesh-refinements, and $\varepsilon = 0.01$: mesh size N_T , space size N_D , energy, error estimators, and number of Newton iterations N_{Newton}	110
5.13	Results of calculations with $\omega = (-0.5, 0.5)^2$, $\mathbf{f} = (1, 1)^T$, $q = 1$, and $N_T = 256$: penalty parameter ε , energy, error estimators, and number of Newton iterations N_{Newton}	113
5.14	Results of calculations with $\omega = (-0.5, 0.5)^2$, $\mathbf{f} = (1, 1)^T$, $q = 1$, and $N_T = 1024$: penalty parameter ε , energy, error estimators, and number of Newton iterations N_{Newton}	113
5.15	Results of calculations with $\omega = (-0.5, 0.5)^2$, $\mathbf{f} = (1, 1)^T$, $q = 1$, and $N_T = 4096$: penalty parameter ε , energy, error estimators, and number of Newton iterations N_{Newton}	113
5.16	Results of calculations with $\omega = (-0.5, 0.5)^T$, $\mathbf{f} = (2, 2)^T$, $q = 1$, and $\varepsilon = h^{0.5}$: mesh-width h , penalty parameter ε , error estimators, and error in the energy norm for a sequence of solutions on uniform meshes.	117
5.17	Results of calculations with $\omega = (-0.5, 0.5)^T$, $\mathbf{f} = (2, 2)^T$, $q = 1$, and $\varepsilon = h^{0.8}$: mesh-width h , penalty parameter ε , error estimators, and error in the energy norm for a sequence of solutions on uniform meshes.	117
5.18	Results of calculations with $\omega = (-0.5, 0.5)^T$, $\mathbf{f} = (2, 2)^T$, $q = 1$, and $\varepsilon = h^1$: mesh-width h , penalty parameter ε , error estimators, and error in the energy norm for a sequence of solutions on uniform meshes.	117
5.19	Results of calculations with $\omega = (-0.5, 0.5)^T$, $\mathbf{f} = (2, 2)^T$, $q = 1$, and $\varepsilon = h^{1.2}$: mesh-width h , penalty parameter ε , error estimators, and error in the energy norm for a sequence of solutions on uniform meshes.	117
5.20	Results of calculations with $\omega = (-0.5, 0.5)^T$, $\mathbf{f} = (2, 2)^T$, $q = 1$, and $\varepsilon = h^{1.5}$: mesh-width h , penalty parameter ε , error estimators, and error in the energy norm for a sequence of solutions on uniform meshes.	118
5.21	Results of calculations with $\omega = (-0.5, 0.5)^T$, $\mathbf{f} = (0.8, 0.8)^T$, $q = 1$, and $\varepsilon = h^{0.5}$: mesh-width h , penalty parameter ε , error estimators, and error in the energy norm for a sequence of solutions on uniform meshes.	119
5.22	Results of calculations with $\omega = (-0.5, 0.5)^T$, $\mathbf{f} = (0.8, 0.8)^T$, $q = 1$, and $\varepsilon = h^{0.8}$: mesh-width h , penalty parameter ε , error estimators, and error in the energy norm for a sequence of solutions on uniform meshes.	120
5.23	Results of calculations with $\omega = (-0.5, 0.5)^T$, $\mathbf{f} = (0.8, 0.8)^T$, $q = 1$, and $\varepsilon = h^1$: mesh-width h , penalty parameter ε , error estimators, and error in the energy norm for a sequence of solutions on uniform meshes.	120

5.24	Results of calculations with $\omega = (-0.5, 0.5)^T$, $\mathbf{f} = (0.8, 0.8)^T$, $q = 1$, and $\varepsilon = h^{1.2}$: mesh-width h , penalty parameter ε , error estimators, and error in the energy norm for a sequence of solutions on uniform meshes.	120
5.25	Results of calculations with $\omega = (-0.5, 0.5)^T$, $\mathbf{f} = (0.8, 0.8)^T$, $q = 1$, and $\varepsilon = h^{1.5}$: mesh-width h , penalty parameter ε , error estimators, and error in the energy norm for a sequence of solutions on uniform meshes.	120
5.26	Results of calculations with $\omega = (-0.5, 0.5)^2$, singular \mathbf{f} , $q = 1$, and h - ε -uniform refinements: mesh size N_T , space size N_D , energy, error estimators, and number of Newton iterations N_{Newton}	121
5.27	Results of calculations for $\omega = (-0.5, 0.5)^2$, singular \mathbf{f} , $q = 1$, and h - ε -adaptive refinements: mesh size N_T , space size N_D , energy, error estimators, and number of Newton iterations N_{Newton}	125
5.28	Results of calculations with $\omega = (-0.5, 0.5) \times (-0.1, 0.1)$, $\mathbf{f} = 10(0.5, 2.0)^T$, $q = 1e4$, and h - ε -uniform refinements: mesh size N_T , space size N_D , energy, error estimators, and number of Newton iterations N_{Newton}	130
5.29	Results of calculations with $\omega = (-0.5, 0.5) \times (-0.1, 0.1)$, $\mathbf{f} = 10(0.5, 2.0)^T$, $q = 1e4$, and h - ε -adaptive refinements: mesh size N_T , space size N_D , energy, error estimators, and number of Newton iterations N_{Newton}	132
5.30	Results of calculations with $\omega = (-0.5, 0.5) \times (-0.1, 0.1)$, $\mathbf{f} = 0.15(0.5, 2.0)^T$, $q = 0$, and uniform mesh-refinements: mesh size N_T , space size N_D , energy, error estimators, and number of Newton iterations N_{Newton}	134
5.31	Results of calculations with $\omega = (-0.5, 0.5) \times (-0.1, 0.1)$, $\mathbf{f} = 0.15(0.5, 2.0)^T$, $q = 0$, and h -adaptive mesh-refinements: mesh size N_T , space size N_D , energy, error estimators, and number of Newton iterations N_{Newton}	136
5.32	Results of calculations with $\omega = (-0.5, 0.5) \times (-0.1, 0.1)$, $\mathbf{f} = 2(0.5, 2.0)^T$, $q = 0$, and h - ε -uniform refinements: mesh size N_T , space size N_D , energy, error estimators, and number of Newton iterations N_{Newton}	138
5.33	Results of calculations with $\omega = (-0.5, 0.5) \times (-0.1, 0.1)$, $\mathbf{f} = 2(0.5, 2.0)^T$, $q = 0$, and h - ε -adaptive refinements: mesh size N_T , space size N_D , energy, error estimators, and number of Newton iterations N_{Newton}	141

Bibliography

- [Ada75] Robert A. Adams. *Sobolev spaces*. Academic Press [A subsidiary of Harcourt Brace Jovanovich, Publishers], New York-London, 1975. Pure and Applied Mathematics, Vol. 65.
- [AFLP10] Markus Aurada, Samuel Ferraz-Leite, and Dirk Praetorius. Estimator reduction and convergence of adaptive bem. Technical Report ASC Report 09/2010, Institute for Analysis and Scientific Computing, Vienna University of Technology, Vienna, 2010.
- [BC05] C. Bahriawati and C. Carstensen. Three MATLAB implementations of the lowest-order Raviart-Thomas MFEM with a posteriori error control. *Comput. Methods Appl. Math.*, 5(4):333–361 (electronic), 2005.
- [Beb00] Mario Bebendorf. Approximation of boundary element matrices. *Numer. Math.*, 86(4):565–589, 2000.
- [Beb08] Mario Bebendorf. *Hierarchical matrices*, volume 63 of *Lecture Notes in Computational Science and Engineering*. Springer-Verlag, Berlin, 2008. A means to efficiently solve elliptic boundary value problems.
- [Bra97] Dietrich Braess. *Finite elements*. Cambridge University Press, Cambridge, 1997. Theory, fast solvers, and applications in solid mechanics, Translated from the 1992 German original by Larry L. Schumaker.
- [Bra02] Andrea Braides. Γ -convergence for beginners, volume 22 of *Oxford Lecture Series in Mathematics and its Applications*. Oxford University Press, Oxford, 2002.
- [Brw63] William Fuller Brwon. *Micromagnetics*. Interscience Tracts on Physics and Astronomy. John Wiley & Sons, 1963.
- [CKNS08] J. Manuel Cascon, Christian Kreuzer, Ricardo H. Nochetto, and Kunibert G. Siebert. Quasi-optimal convergence rate for an adaptive finite element method. *SIAM J. Numer. Anal.*, 46(5):2524–2550, 2008.
- [CP01] Carsten Carstensen and Andreas Prohl. Numerical analysis of relaxed micromagnetics by penalised finite elements. *Numer. Math.*, 90(1):65–99, 2001.
- [CP06] Carsten Carstensen and Dirk Praetorius. Averaging techniques for the effective numerical solution of Symm’s integral equation of the first kind. *SIAM J. Sci. Comput.*, 27(4):1226–1260 (electronic), 2006.
- [Dac89] Bernard Dacorogna. *Direct methods in the calculus of variations*, volume 78 of *Applied Mathematical Sciences*. Springer-Verlag, Berlin, 1989.

- [dBR79] Paul du Bois-Reymond. Erläuterungen zu der Anfangsgründen der Variationsrechnung. *Math. Ann.*, 15:283–314, 1879.
- [De 93] Antonio De Simone. Energy minimizers for large ferromagnetic bodies. *Arch. Rational Mech. Anal.*, 125(2):99–143, 1993.
- [DKM⁺01] Antonio DeSimone, Robert V. Kohn, Stefan Müller, Felix Otto, and Rudolf Schäfer. Two-dimensional modelling of soft ferromagnetic films. *R. Soc. Lond. Proc. Ser. A Math. Phys. Eng. Sci.*, 457(2016):2983–2991, 2001.
- [DKMO02] Antonio Desimone, Robert V. Kohn, Stefan Müller, and Felix Otto. A reduced theory for thin-film micromagnetics. *Comm. Pure Appl. Math.*, 55(11):1408–1460, 2002.
- [DKMO06] Antonio Desimone, Robert V. Kohn, Stefan Müller, and Felix Otto. Recent analytical developments in micromagnetics. In Giorgio Bertotti and Isaak D. Mayergoyz, editors, *The science of hysteresis. Vol. II*, pages 269–381. Elsevier/Academic Press, Amsterdam, 2006. Physical modeling, micromagnetics, and magnetization dynamics.
- [DL54] J. Deny and J. L. Lions. Les espaces du type de Beppo Levi. *Ann. Inst. Fourier, Grenoble*, 5:305–370 (1955), 1953–54.
- [DN02] Willy Dörfler and Ricardo H. Nochetto. Small data oscillation implies the saturation assumption. *Numer. Math.*, 91(1):1–12, 2002.
- [DO04] Jörg Drwenski and Felix Otto. \mathcal{H}^2 -matrix methods vs. FFT in thin-film stray-field computations. *SFB Preprint*, 161, 2004.
- [Dör96] Willy Dörfler. A convergent adaptive algorithm for Poisson’s equation. *SIAM J. Numer. Anal.*, 33(3):1106–1124, 1996.
- [Drw08] Jörg Drwenski. *Numerical Methods for a Reduced Model in Thin-Film Micromagnetics*. Dissertation, Rheinische Friedrichs-Wilhelms-Universität Bonn, Bonn, 2008.
- [DS05] Zdeněk Dostál and Joachim Schöberl. Minimizing quadratic functions subject to bound constraints with the rate of convergence and finite termination. *Comput. Optim. Appl.*, 30(1):23–43, 2005.
- [EFLFP09] C. Erath, S. Ferraz-Leite, S. Funken, and D. Praetorius. Energy norm based a posteriori error estimation for boundary element methods in two dimensions. *Appl. Numer. Math.*, 59(11):2713–2734, 2009.
- [ESAES90] V. J. Ervin, E. P. Stephan, and S. Abou El-Seoud. An improved boundary element method for the charge density of a thin electrified plate in \mathbf{R}^3 . *Math. Methods Appl. Sci.*, 13(4):291–303, 1990.
- [Eva98] Lawrence C. Evans. *Partial differential equations*, volume 19 of *Graduate Studies in Mathematics*. American Mathematical Society, Providence, RI, 1998.
- [Fae02] Birgit Faermann. Localization of the Aronszajn-Slobodeckij norm and application to adaptive boundary element methods. II. The three-dimensional case. *Numer. Math.*, 92(3):467–499, 2002.
- [FLOP10] S. Ferraz-Leite, C. Ortner, and D. Praetorius. Convergence of simple adaptive Galerkin schemes based on $h - h/2$ error estimators. *Numer. Math.*, 116(2):291–316, 2010.

- [FLP08] S. Ferraz-Leite and D. Praetorius. Simple a posteriori error estimators for the h -version of the boundary element method. *Computing*, 83(4):135–162, 2008.
- [GHS05] I. G. Graham, W. Hackbusch, and S. A. Sauter. Finite elements on degenerate meshes: inverse-type inequalities and applications. *IMA J. Numer. Anal.*, 25(2):379–407, 2005.
- [GR86] Vivette Girault and Pierre-Arnaud Raviart. *Finite element methods for Navier-Stokes equations*, volume 5 of *Springer Series in Computational Mathematics*. Springer-Verlag, Berlin, 1986. Theory and algorithms.
- [Gra01] Lars Grasedyck. *Theorie und Anwendungen Hierarchischer Matrizen*. Dissertation, Christian-Albrechts-Universität zu Kiel, Kiel, 2001.
- [Hac95] Wolfgang Hackbusch. *Integral equations*, volume 120 of *International Series of Numerical Mathematics*. Birkhäuser Verlag, Basel, 1995. Theory and numerical treatment, Translated and revised by the author from the 1989 German original.
- [Hac99] W. Hackbusch. A sparse matrix arithmetic based on \mathcal{H} -matrices. I. Introduction to \mathcal{H} -matrices. *Computing*, 62(2):89–108, 1999.
- [Hac09] Wolfgang Hackbusch. *Hierarchische Matrizen: Algorithmen und Analysis*. Springer-Verlag, Berlin Heidelberg, 2009.
- [HB02] Wolfgang Hackbusch and Steffen Börm. \mathcal{H}^2 -matrix approximation of integral operators by interpolation. *Appl. Numer. Math.*, 43(1-2):129–143, 2002. 19th Dundee Biennial Conference on Numerical Analysis (2001).
- [IK08] Kazufumi Ito and Karl Kunisch. *Lagrange multiplier approach to variational problems and applications*, volume 15 of *Advances in Design and Control*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2008.
- [JLJ98] Jürgen Jost and Xianqing Li-Jost. *Calculus of variations*, volume 64 of *Cambridge Studies in Advanced Mathematics*. Cambridge University Press, Cambridge, 1998.
- [KS80] David Kinderlehrer and Guido Stampacchia. *An introduction to variational inequalities and their applications*, volume 88 of *Pure and Applied Mathematics*. Academic Press Inc. [Harcourt Brace Jovanovich Publishers], New York, 1980.
- [LL35] Lev Davidovich Landau and Eugenii Mykhailovich Lifshits. On the theory of the dispersion of magnetic permeability in ferromagnetic bodies. *Phys. Zeitsch. der Sow.*, 8:153–169, 1935.
- [McL00] William McLean. *Strongly elliptic systems and boundary integral equations*. Cambridge University Press, Cambridge, 2000.
- [NW99] Jorge Nocedal and Stephen J. Wright. *Numerical optimization*. Springer Series in Operations Research. Springer-Verlag, New York, 1999.
- [Pra] Dirk Praetorius. Integral Equations and Boundary Element Methods. lecture notes, Institute for Analysis and Scientific Computing, Vienna University of Technology.
- [Pra03] Dirk Praetorius. *Analysis, Numerik und Simulation eines relaxierten Modellproblems zum Mikromagnetismus*. Dissertation, Technische Universität Wien, Wien, 2003.
- [RT77] P.-A. Raviart and J. M. Thomas. Primal hybrid finite element methods for 2nd order elliptic equations. *Math. Comp.*, 31(138):391–413, 1977.

- [SB89] H. Suhl and P. Bryant. Thin-film magnetic patterns in an external field. *Appl. Phys. Lett.*, 54(22):2224–2226, 1989.
- [SB02] J. Stoer and R. Bulirsch. *Introduction to numerical analysis*, volume 12 of *Texts in Applied Mathematics*. Springer-Verlag, New York, third edition, 2002. Translated from the German by R. Bartels, W. Gautschi and C. Witzgall.
- [SH98] Rudolf Schäfer and Alex Hubert. *Magnetic domains: the analysis of magnetic microstructures*. Springer-Verlag, 1998.
- [Sha09] Yair Shapira. *Mathematical Objects in C++: Computational Tools in A Unified Object-Oriented Approach*. CRC Press, Inc., Boca Raton, FL, USA, 1st edition, 2009.
- [Sil70] P. Silvester. Symmetric quadrature formulae for simplexes. *Math. Comp.*, 24:95–100, 1970.
- [SS04] Stephan Sauter and Christoph Schwab. *Randelementmethoden*. B. G. Teubner, Wiesbaden, 2004. Analyse, Numerik und Implementierung schneller Algorithmen.
- [Ste87] Ernst P. Stephan. Boundary integral equations for screen problems in \mathbf{R}^3 . *Integral Equations Operator Theory*, 10(2):236–257, 1987.
- [Ste03] Olaf Steinbach. *Numerische Näherungsverfahren für elliptische Randwertprobleme*. B. G. Teubner, Wiesbaden, 2003. Finite Elemente und Randelemente.
- [Str71] A. H. Stroud. *Approximate calculation of multiple integrals*. Prentice-Hall Inc., Englewood Cliffs, N.J., 1971. Prentice-Hall Series in Automatic Computation.
- [vdB86] H.A.M. van den Berg. Self-consistent domain theory in soft ferromagnetic media. ii, basic domain structures in thin film objects. *J. Appl. Phys.*, 60:1104–1113, 1986.
- [Ver96] Rüdiger Verfürth. *A review of a posteriori error estimation and adaptive mesh-refinement techniques*. Wiley-Teubner series in advances in numerical mathematics. Wiley-Teubner, 1996.
- [Yos65] Kôsaku Yosida. *Functional analysis*. Die Grundlehren der Mathematischen Wissenschaften, Band 123. Academic Press Inc., New York, 1965.

Curriculum vitae Samuel Ferraz-Leite

Institute for Analysis and Scientific Computing
Vienna University of Technology
Wiedner Hauptstr. 8-10
1040 Vienna, Austria

Fon: +43 1 58801-10167
Fax: +43 1 58801-10196
Mail: samuel.ferraz-leite@tuwien.ac.at
www.asc.tuwien.ac.at/~samuel

PERSONAL

Date of birth: 16.01.1982
Place of birth: Montevideo, Uruguay
Citizenship: Austrian
Marital status: married to Bianca (07/2004),
children: Joel (09/2007), Jan Simon (01/2010)

EDUCATION

since 01/2008 **PhD student**
Vienna University of Technology
01/2008–12/2010 PhD grant from graduate school
"Differential Equations" (funded by Austrian Science Fund)
01/2011–02/2011 Funded by the FWF project "Adaptive BEM" (P21732)

10/2001–11/2007 **Studies of Technical Mathematics**
Vienna University of Technology
11/2007 Diploma in Technical Mathematics (with distinction)
10/2007 Diploma thesis: "A posteriori Fehlerschätzer für die
Symmsche Integralgleichung in 3D"

09/1992–06/2000 **High school**
Theresianische Akademie Wien
06/2000 Matura

ENGINEERING EXPERIENCE

04/2010–12/2010 **Consultant: risk management**
Best Gaming Technology GmbH
Design and implementation of strategies for
risk monitoring and counter measures (C#)

08/2009–07/2010 **Senior developer – HILBERT**
Vienna University of Technology
Senior member of the HILBERT Team
Teamleader in the implementation of BEM operators in C

08/2000–11/2001 **Softwareengineer**
FSC KG
Development, installation and maintenance
of software in C, Delphi and Java

LIST OF PUBLICATIONS

- S. FERRAZ-LEITE, C. ORTNER, D. PRAETORIUS:
Convergence of simple adaptive Galerkin schemes based on H - $H/2$ error estimators,
Numer. Math., 116, 291-316, 2010
- M. MAYR (SUPERVISOR: S. FERRAZ-LEITE, D. PRAETORIUS):
Stabile Implementierung der Randelementmethode auf stark adaptierten Gittern,
Bachelorthesis, Institute for Analysis and Scientific Computing,
Vienna University of Technology, 2010
- C. ERATH, S. FERRAZ-LEITE, S. FUNKEN, D. PRAETORIUS:
Energy Norm Based A Posteriori Error Estimation for Boundary Element Methods in Two Dimensions,
Appl. Numer. Math., 59, 2713-2734, 2009
- S. FERRAZ-LEITE, J. MELENK, D. PRAETORIUS:
Reduced Model in Thin-Film Micromagnetics,
Proceedings MATHMOD 09 Vienna, I. Troch, F. Breitenecker (Hrsg.),
ARGESIM Report no. 35, 2009
- M. AURADA, S. FERRAZ-LEITE, D. PRAETORIUS:
Convergence of Adaptive Boundary Element Methods,
Short Papers of 18th International Conference on Computer Methods in Mechanics,
Zielona Gora, 2009
- S. FERRAZ-LEITE, D. PRAETORIUS:
Simple A Posteriori Error Estimators for the h -Version of the Boundary Element Method,
Computing, 83, 135-162, 2008
- S. FERRAZ-LEITE, C. ORTNER, D. PRAETORIUS:
Adaptive Boundary Element Method: Simple Error Estimators and Convergence,
Analysis of Boundary Element Methods, EMS Publishing House,
Oberwolfach Report 19/2008, 2008
- S. FERRAZ-LEITE, D. PRAETORIUS:
Adaptive Boundary Element Methods Based on Accurate A Posteriori Error Estimation,
Proceedings of the Junior Scientist Conference, Technische Universität Wien, 2008
- S. FERRAZ-LEITE
A posteriori Fehlerschätzer für die Symmsche Integralgleichung in 3D,
Diplomathesis in German, Institute for Analysis and Scientific Computing,
Vienna University of Technology, 2007