



DISSERTATION

Multiscale Simulation of Magnetic Nanostructures

zur Erlangung des akademischen Grades

Doktor der technischen Wissenschaften

eingereicht von

Dipl.-Ing. Florian Bruckner

Matrikelnummer 0425375

Eslarngasse 2/18, 1030 Wien

an der

Fakultät für Physik der Technischen Universität Wien

Betreuung:

Privatdoz. Dipl.-Ing. Dr.techn. Dieter Süss,

Institut für Festkörperphysik – E138

Diese Dissertation haben begutachtet:

(Privatdoz. Dipl.-Ing.
Dr.techn. Dieter Süss)

(Ao. Univ. Prof. Dipl. Math.
Dr.techn. Dirk Praetorius)

Wien, 31.03.2013

(Dipl.-Ing. Florian Bruckner)

DISSERTATION

Multiscale Simulation of Magnetic Nanostructures

submitted in partial fulfillment of the requirements for the degree of

Doktor der technischen Wissenschaften

by

Dipl.-Ing. Florian Bruckner

Registration Number 0425375

Eslarngasse 2/18, 1030 Wien

to the Faculty of Physics
at the Vienna University of Technology

Advisor:

Privatdoz. Dipl.-Ing. Dr.techn. Dieter Süss

Institute of Solid State Physics – E138

The dissertation has been reviewed by:

(Privatdoz. Dipl.-Ing.
Dr.techn. Dieter Süss)

(Ao. Univ. Prof. Dipl. Math.
Dr.techn. Dirk Praetorius)

Wien, 31.03.2013

(Dipl.-Ing. Florian Bruckner)

Erklärung zur Verfassung der Arbeit

Dipl.-Ing. Florian Bruckner
Eslarngasse 2/18, 1030 Wien

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit - einschließlich Tabellen, Karten und Abbildungen -, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

(Ort, Datum)

(Unterschrift Dipl.-Ing. Florian
Bruckner)

Abstract

Due to the ongoing miniaturization of modern magnetic devices, ranging from GMR sensors, magnetic write heads, to spintronic devices, micromagnetic simulations gain more and more importance, since they are an essential tool to understand the behavior of magnetic materials in the nanometer scale. The use of numerical simulations allows to optimize the micro-structure of such devices or to test new concepts prior to performing expensive experimental tests.

The purpose of this work is to extend the micromagnetic model by additional macroscopic parts which are described in an averaged sense. In contrast to the microscopic parts which are described by Landau-Lifshitz-Gilbert (LLG) equations, these macroscopic parts are based on classical magnetostatic Maxwell equations, which could be extended to a full Maxwell description in a straight-forward way. The averaged description using Maxwell equations allows to overcome the upper bound for the discrete element sizes, which is intrinsic to the micromagnetic models, since the detailed domain structure of the ferromagnetic material needs to be resolved.

Combining microscopic and macroscopic models and solving the corresponding equations simultaneously provides a multiscale method, which allows to handle problems of a dimension, which would otherwise be far out of reach. A basic prerequisite for the application of the method, is that microscopic and macroscopic parts can be separated into disjoint regions. The performance of the implemented algorithm is demonstrated by the simulation of the transfer curve of a magnetic recording read head, as it is used in current hard drives.

Independent of the former approach the use of parallelized algorithms al-

lows to handle larger problems. This work especially deals with the shared-memory-parallelization of hierarchical matrices, since these require a large amount of the storage consumption as well as of the computation time for typical simulations. For the setup of the matrices a nearly perfect parallelization could be reached, whereas for the matrix-vector-multiplication the computation time stagnates at a few computation cores, due to the restricted main memory bandwidth.

Kurzfassung

Durch die fortlaufende Miniaturisierung moderner magnetischer Bauelemente, wie GMR Sensoren, magnetischen Schreibköpfen, Spintronik Bauteilen, usw., erlangen mikromagnetische Simulationen mehr und mehr an Bedeutung. Sie stellen ein wichtiges Hilfsmittel dar, um das Verhalten magnetischer Materialien im Nanometer-Bereich besser verstehen zu können. Die Verwendung numerischer Simulationen erlaubt es die Mikrostruktur solcher Bauelemente zu optimieren bzw. neue Konzepte kostengünstig zu überprüfen, bevor diese im Laborexperiment getestet werden.

Ziel dieser Arbeit ist es mikromagnetische Modelle, um makroskopische Komponenten zu erweitern, deren Verhalten nur im gemittelten Sinn beschrieben wird. Im Unterschied zu den mikroskopischen Komponenten, welche durch die Landau-Lifshitz-Gilbert (LLG) Gleichungen beschrieben werden, basieren diese makroskopischen Komponenten auf den klassischen magnetostatischen Maxwell - Gleichungen, welche auch durch die vollen Maxwell - Gleichungen ersetzt werden könnten. Diese gemittelte Beschreibung mittels Maxwell - Gleichungen erlaubt es, beliebig große Elemente zur Diskretisierung zu verwenden, was bei mikromagnetischen Modellen nicht möglich ist, weil die detaillierte Domänen-Struktur des ferromagnetischen Materials abgebildet werden muss.

Die Kopplung von mikroskopischen und makroskopischen Modellen und gleichzeitige Lösung der zugrundeliegenden Gleichungen führt zu einer Multi-Skalen-Methode, die es erlaubt Probleme von zuvor unerreichbarer Größe zu verarbeiten. Eine Grundvoraussetzung für die Anwendbarkeit der Methode ist, dass mikroskopische und makroskopische Komponenten in nicht-überlappende

Bereiche aufgeteilt werden können. Die Leistungsfähigkeit des implementierten Algorithmus wird anhand der Simulation der Transfer-Kurve eines magnetischen Lesekopfes, wie er in modernen Festplatten zum Einsatz kommt, demonstriert.

Unabhängig von diesem Ansatz erlauben parallelisierte Algorithmen die Verarbeitung größerer Modelle. Im Zuge dieser Arbeit wurde speziell auf die Shared-Memory-Parallelisierung von hierarchischen Matrizen eingegangen, da diese in typischen Simulationen einen Großteil der Rechenzeit und des Speicherbedarfs benötigen. Für das Aufstellen der Matrizen konnte eine annähernd perfekte Parallelisierung erreicht werden, während man bei der Matrix-Vektor-Multiplikation aufgrund der Bandbreitenlimitierung des Hauptspeichers bereits mit einigen wenigen Rechenkernen eine Stagnation der Verarbeitungsgeschwindigkeit beobachten kann.

Contents

1	Introduction	1
1.1	State of the Art	1
1.2	Advances over the State of the Art	3
1.3	Structure of the Dissertation	4
2	Numerical Methods	6
2.1	Finite Elements	6
2.2	Boundary Elements	11
2.3	Hierarchical Matrices	14
2.4	Newton-Method	19
3	Micromagnetism	22
3.1	Introduction	22
3.1.1	Exchange energy	23
3.1.2	Magnetocrystalline anisotropy energy	24
3.1.3	Strayfield Energy	25
3.1.4	The effective field and Brown's equations	25
3.2	The Landau-Lifshitz-Gilbert equation	26
3.2.1	Standard form	27
3.2.2	Gilbert form	28
3.2.3	Alternative form	29
3.2.4	Weak Formulation	29
3.3	Strayfield Calculation	31
3.3.1	Integral Methods	31

3.3.2	Differential Methods	33
3.3.3	Mixed Methods	35
4	Maxwell Equations	40
4.1	Introduction	40
4.2	Scalar Potential Formulations	41
4.2.1	Reduced Scalar Potential u	42
4.2.2	Total Scalar Potential \tilde{u} [1]	43
4.2.3	Mayergoyz Potential \hat{u} [2]	43
4.3	FEM-BEM coupling	45
4.3.1	Linear Reduced Scalar Potential	45
4.3.2	Linear Total Scalar Potential	47
4.3.3	Nonlinear Reduced Scalar Potential	48
4.4	Calculation of BEM matrices	49
4.5	Nonlinear Materials	52
4.6	Solution Algorithms	54
4.6.1	Globalization strategies	56
4.7	Preconditioning	58
4.8	Test Cases	59
4.8.1	Magnetic sphere in homogeneous external field	61
4.8.2	Field within air-gap of a magnetic yoke	61
4.8.3	Artificial analytical solutions for the stator of an electric motor	64
4.8.4	Team Workshop Problem 13	67
4.9	Conclusion	67
5	Coupling	70
5.1	Introduction	70
5.2	Fundamentals	71
5.2.1	Discretization	73
5.3	Interactions	73
5.3.1	Coil-Model Interaction	74

5.3.2	LLG-Model Interaction	75
5.3.3	Maxwell-Model Interaction	75
5.4	Coupling Method	76
5.4.1	Optimization	77
5.5	Test Cases	77
5.5.1	Unidirectional interaction	78
5.5.2	Bidirectional interaction	81
5.5.3	GMR Read Head Transfer Curve	83
5.6	Conclusion	87
6	Parallelization of \mathcal{H}-matrices	91
6.1	Introduction	91
6.2	Implementation	92
6.2.1	Matrix Setup	93
6.2.2	Matrix-Vector Multiplication	94
6.3	Benchmark	95
7	Conclusion and Outlook	98
7.1	Conclusion	98
7.2	Outlook	99
	Bibliography	101

CHAPTER 1

Introduction

1.1 State of the Art

As a motivation for the following work some typical real-world problems will be outlined. The main field of interest lays in understanding the behavior of magnetic structures in a microscopic scale. Due to the small scales classical magnetism, which is described by Maxwell's equations, needs to be extended by quantum-mechanical interactions. This short range interaction is called exchange interaction and often plays an important role for the magnetic behavior in small dimensions. Micromagnetism provides one way to include these quantum-mechanical effects into the model description, via a continuum model of individual spins. The dynamic of these spins is described by the classical precession as well as a phenomenological damping contribution. By this way Micromagnetism makes it possible to describe the formation of magnetic domains within ferromagnetic media including dynamic processes and diverse interactions.

One of the original fields of application of these micromagnetic equations are magnetic storage devices. Due to the ongoing miniaturization and optimization of magnetic storage devices, more and more side-effects need to be considered for the design of these structures. Computer simulations could pro-

vide a cheap way to pre-assess new concepts or to optimize the final design as well. Additionally simulations can offer a better understanding of the underlying principles, since they allow to break down the contributions of different parts of the system, or of different interactions, which would otherwise not be individually measurable within a real experiment.

Current state of the art within this field are simulations which allow to describe problems with up to one billion unknowns. The effective field typically contains at least contributions of the quantum-mechanical exchange, the magneto-crystalline anisotropy for anisotropic materials, the magnetostatic demagnetization field as well as an external strayfield created by coils or other magnetic components. Depending on the problem-details additional contributions can be considered, like thermal fluctuations, eddy currents, or spin-torque. On the one hand including more of these interactions allows to describe the microscopic magnetic behavior more accurately, but on the other hand real magnetic devices also contain macroscopic parts, which have a strong influence on the overall performance. Describing these macroscopic parts using the same underlying model is normally not applicable, since the resolution of the total domain-structure would result in a huge number of elements for the discretization. Today most micromagnetic simulations therefore strictly separate these two scales.

For example the write field created by a magnetic write head for a certain current is pre-computed once. It is often sufficient to use classical Maxwell equations for this description since the medium is not included within this pre-computation. For the simulation of a magnetic medium this field is then scaled according to the actual current. By this way the scales can be separated, but this also means that there cannot be any back-interaction of the magnetic medium onto the write head. Improving this situation is exactly the intention of this work. The first step into this direction is to utilize some of the methods, which are already successfully applied to the micromagnetic equations, also for the solution of the classical Maxwell equations (see Chapter 4). This leads to a standalone Maxwell solver, which can be used as aforementioned, but it now

fits into the framework of the micromagnetic solver. The most important feature that the Maxwell solver provide is some sort of FEM-BEM coupling, which avoids the need to mesh regions outside of the magnetic parts. Finally the original micromagnetic equations can be coupled with the classical Maxwell equations via their strayfield and both systems of equations can be solved simultaneously (see Chapter 5). By doing this it is possible to also include very large components into the problem description, without ending up with a tremendous number of unknowns which cannot be handled adequately.

Another research field where such a multiscale method could be applicable, is the fabrication of permanent magnets with a high energy product. Proper intrinsic material properties, like a high magneto-crystalline anisotropy as well as a high saturation polarization, are the foundations for a strong permanent magnet. But nevertheless also the structure of the magnetic grains has an important influence on the achievable field strengths. The magnet itself has macroscopic dimensions, whereas its grains are in the nanometer scale. Using classical methods only allows to simulate a limited number of magnetic grains, which admits finite size effects as well as large systematic errors due to the reduced complexity. Using a multiscale method allows to adequately describe some of the magnetic grains by means of their full micromagnetic model, whereas all the rest is described in an averaged manner by means of classical Maxwell equations.

1.2 Advances over the State of the Art

Building upon an existing micromagnetic code, the first step towards a multiscale method was the implementation of an efficient stand-alone solver for Maxwell's equations, which fits into the framework of the micromagnetic code. In the course of this thesis several solvers, with different underlying potential formulations, have been implemented and tested for various test cases. The use of matrix-free Krylov methods results in very efficient algorithms which are well suited for large scale problems. All of the implemented methods have in common that they are based on FEM-BEM coupling, in order to avoid mesh-

ing of the other region. This is essential for the possible coupling with micro-magnetic models, since otherwise the meshing of the combined problem would be much more complicated. A further innovation was the use of hierarchical matrices for the BEM part of the developed algorithms as well as a proper preconditioner for the combined system with leads to a further optimization of the code [3].

The largest advance that could be archived during this thesis was the coupling of the newly developed Maxwell solver with the original micromagnetic code. The resulting multiscale method could be validated by several test cases and its application to the simulation of magnetic recoding devices shows a huge performance benefit [4]. Different coupling strategies have been tested and validated. Finally an optimized sequential coupling method proved most stable and efficient and convergence could even be proved analytically [5].

The consideration of FEM-BEM codes shows that typically the most storage as well as computation time is needed for the handling of the dense BEM matrices. Although the use of hierarchical matrices allows to archive the same asymptotic performance as for the FEM part, in practice there is still a large performance difference between both parts. During this thesis parts of the hierarchical matrix code could be parallelized in order to speed up evaluation and setup on shared memory machines. Nearly perfect scaling could be reached for the matrix setup, whereas for the evaluation of the matrix-vector product a different performance bottleneck due to the main memory bandwidth became visible. Nevertheless it was possible to significantly speed up the handling of \mathcal{H} -matrices especially on systems with fast memory.

1.3 Structure of the Dissertation

The following chapter will give a short overview of the numerical methods, which are the basis for the methods presented in the later chapters. Chapter 3 summarizes the main concepts behind Micromagnetism and gives an overview over the methods used to solve open boundary problems such as the micro-

magnetic strayfield computation. Classical Maxwell equations and their representation via scalar potentials are introduced in Chapter 4. Additionally the solution algorithm as well as the FEM-BEM method used to deal with the open boundary problem are presented in detail. Simulation results of some test cases are listed in order to validate and benchmark the implemented algorithms. Finally Chapter 5 shows how the former two methods can be linked together in order to create a multiscale solver. In addition to some artificial test cases the simulation of a magnetic read head demonstrates the usability of the multiscale code. Parallelization of the code which became more and more important in recent years is addressed in Chapter 6, where the parallel setup and evaluation of \mathcal{H} -matrices is demonstrated.

Numerical Methods

2.1 Finite Elements

Magnetization is a long range interaction, whose basic equations can be formulated in integral or differential form, leading to partial differential equations (PDE) or integral equations (IE). Typically computational implementation of the differential form is preferred since it leads to sparse matrices which can be handled very efficiently. In either case finite elements can be used to discretize the spacial part of the equations.

Based on the definition of Ciarlet [6] a finite element consists of the following parts:

- Geometric Domain Ω_{FE} : Describes the finite size region in which the element basis functions should differ from 0. The region can be of any dimension (1D, 2D, 3D).
- Local Element Space V_{FE} : This vector space spans the space of functions which should be representable by the finite element. Typically this vector space is a subset of the polynomial space of a certain order. The dimension of the vector space N is equal to the number of unknowns per element.

- Set of linearly independent functionals $\psi_1, \psi_2, \dots, \psi_N$ on V_{FE} : These functionals are called degrees of freedom of the finite element. Given all values of the functionals uniquely defines the finite element function. Given these functionals allows to derive the corresponding element basis functions ϕ_i with $i = 1 \dots N$ via the following orthogonality relation:

$$\psi_i(\phi_j) = \delta_{ij} \quad \delta_{ij} = \begin{cases} 0 & i \neq j \\ 1 & i = j \end{cases} \quad (2.1)$$

In FEM literature the basis functions are often defined directly without the underlying functionals. The advantage of starting from the functionals is that they directly lead to an interpolation operator which allows to project arbitrary functions into the finite element space:

$$u_h = I_h u = \sum_{i=1}^N \psi_i(u) \phi_i \quad (2.2)$$

In Fig. 2.1 the projection into the finite element space is presented for a linear Lagrange element in one dimension.

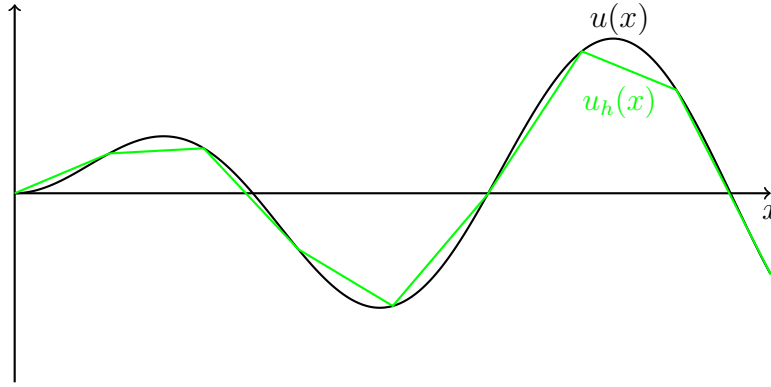


Figure 2.1: Visualization of the FEM interpolation for the 1D linear Lagrange element. The degrees of freedom are the function-values of the continuous function $u(x)$ at some sampling point x_i , and the basis functions are hat-functions, which results in a linear interpolation between the sampling points.

In order to utilize finite elements the PDE first has to be transformed into its weak formulation by multiplying it with a test function, integrating over the

whole space and performing an integration by parts in order to end up with a symmetric bilinear form. As an example the weak form of the Laplace equation is derived in the following

$$\begin{aligned}
 -\Delta u &= f \\
 \int_{\Omega} -\Delta u \phi \, d\Omega &= \int_{\Omega} f \phi \, d\Omega \\
 \int_{\Omega} \nabla u \cdot \nabla \phi \, d\Omega - \int_{\Gamma} \underbrace{\nabla u \cdot \mathbf{n}}_{=g} \phi \, d\Gamma &= \int_{\Omega} f \phi \, d\Omega
 \end{aligned} \tag{2.3}$$

where Γ is the boundary of the integration domain Ω , ϕ are arbitrary test functions and g are so-called Neumann boundary conditions, which have to be set in order to get a unique solution. Alternatively one can define Dirichlet boundary conditions by fixing u at the boundary, which is equivalent to testing only with the volume test functions and omitting the surface test functions.

In order to obtain the discretized system of equations one can use the Galerkin-approach, where a certain basis is selected and the discretized solution u_h is described within this basis.

$$u_h(x) = \sum_{i=1}^N u_i \phi_i(x) \tag{2.4}$$

Each basis-function ϕ_j is used as test function, which leads to a quadratic system of equations for the coefficients u_i . For the upper example this looks like

$$u_i \int_{\Omega} \nabla \phi_i \cdot \nabla \phi_j \, d\Omega - \int_{\Gamma} g \phi_j \, d\Gamma = \int_{\Omega} f \phi_j \, d\Omega \tag{2.5}$$

As an example the lowest order nodal Lagrange element can be defined for this problem. The geometric domain should be a line since we are dealing with a 1D problem. The lowest order local element space is the $P1$, which is the space of polynomials of degree 1. The resulting basis function will therefore be linear. For Lagrange elements the linear independent functionals are integrals over delta functions defined on the sampling points of the element multiplied with the input function. The application of one of these delta functions on an arbitrary function just evaluates its value on the sampling points. For the 1D example the orthogonality relation (2.1) can be used to derive the basis functions.

The first one has to be 0 at the left node and 1 at the right on and the second one vice versa. For a reference element defined on the interval $(0, 1)$ one therefore obtains the basis functions x and $1 - x$.

The weak formulation of PDEs is very comfortable because it allows to prove the existence and uniqueness of a solution under certain conditions, e.g. using the Lax-Milgram lemma.

Lemma 2.1 (Lax-Milgram) *Let V be a Hilbert-Space. Let $B(., .) : V \times V \rightarrow \mathbb{R}$ be a bilinear form, which is coercive*

$$B(u, u) \geq c_1 \|u\|_V^2 \quad \forall u \in V$$

and continuous

$$B(u, v) \leq c_2 \|u\|_V \|v\|_V \quad \forall u, v \in V$$

and let $f(.)$ be a continuous linear form, then there exists a unique $u \in V$ such that

$$B(u, v) = f(v) \quad \forall v \in V$$

There holds the stability estimate

$$\|u\|_V \leq \frac{1}{c_1} \|f\|_{V^*}$$

Additionally to existence and uniqueness the upper lemma shows that the solution u of the problem is stable, which means that it does not change significantly if the right-hand-side is varied slightly.

The proper vector space V for solutions of the Laplace equations defined in the weak formulation is the so-called H^1 Sobolev spaces:

Definition 2.1 (H^1 Sobolev space) *We call Sobolev space of order 1 on Ω the space*

$$H^1(\Omega) := \{u \in L^2(\Omega) | u' \in L^2(\Omega)\} \quad (2.6)$$

where L^2 is the space of all square-integrable functions and the corresponding H^1 -norm is defined as

$$\|u\|_{H^1} = \sqrt{\|u\|_{L^2}^2 + \|u'\|_{L^2}^2} \quad (2.7)$$

The use of this H^1 -norm makes sure that the function u as well as its derivative u' is considered.

Another important question is how much the discretized solution u_h may differ from the continuous one. One first notices that the difference between the discretized solution and the continuous one is always B -orthogonal to functions within V_{FE} , which directly follows from the definition of the discretized solution

$$B(u_h, v_h) = f(v_h) \quad \forall v_h \in V_{FE} \quad (2.8)$$

and from the fact that $V_{FE} \subset V$. Therefore one can choose v_h as test functions of the continuous equations and subtraction finally leads to

$$B(u - u_h, v_h) = 0 \quad (2.9)$$

which is also called Galerkin-orthogonality. From this orthogonality relation it is only a small step to Céa's-Lemma, which holds under the same assumptions as the Lax-Milgram-Lemma:

$$\begin{aligned} c_1 \|u - u_h\|_V^2 &\leq B(u - u_h, u - u_h) = B(u - u_h, u - v_h) + \underbrace{B(u - u_h, v_h - u_h)}_{=0} \leq \\ &\leq c_2 \|u - u_h\|_V \|u - v_h\|_V \\ \Rightarrow \|u - u_h\|_V &\leq \frac{c_2}{c_1} \|u - v_h\|_V \quad \forall v_h \in V_{FE} \end{aligned} \quad (2.10)$$

The meaning of Céa's Lemma is that the error-norm of the discretized solution is smaller than the error norm of every possible function v_h , within the given discrete space, up to a certain constant, which depends on the used norm. Using the so-called energy norm, which is deduced from the bilinear form, the constant becomes 1, which means that the discrete solution is the best approximation with respect to the energy norm.

Although the former result tells that the discretized solution is a best approximation, it does not give any absolute bound for the achieved discretization error. This is where the Bramble-Hilbert lemma comes into play, which

gives an absolute bound for the error of the approximation of a function u by a polynomial of order at most $m - 1$ in terms of the derivatives of u of order m . The solution of typical problems that we are dealing with are located within the H^1 vector space, so using the H^1 -norm for the error estimation seems natural. For the one-dimensional case on the interval $(0, h)$, the lemma can be stated as follows

$$\inf_{v \in \mathbf{P}_{m-1}} \|u - v\|_{H^1} \leq C(m) h^{m-1} \|u^{(m)}\|_{L^2} \quad (2.11)$$

where $u^{(m)}$ is the m -th derivative of u . This is a so-called a-priori error estimation. It derives bounds for the error of the discretized solution u_h only assuming some properties of the exact solution u . In general such methods do not allow to explicitly calculate the constants of the estimation. It therefore only makes it possible to determine how the error behaves for varying mesh sizes h . Alternatively one can use so-called a-posteriori error estimators which estimate the error based on the actual discretized solution and therefore allow to calculate absolute bounds.

2.2 Boundary Elements

The boundary element method is based on some sort of representation formulas, which allow to represent the solution of the underlying PDE within the domain, only by its values on the boundary. Examples for such formulas are Green's third identity for the Laplace equation, Betti's formula for elasticity theory, or the Stratton-Chu formula for electrodynamics. As an example Green's third identity for the Laplace equation shall be presented here, since it is also used in the subsequent chapters to solve magnetostatic Maxwell equations, by introducing a scalar potential:

$$\alpha u(\mathbf{x}) = \underbrace{\int_{\Gamma} u(\mathbf{x}') \nabla G(\mathbf{x}, \mathbf{x}') \mathbf{n} d\Gamma'}_{=Ku} - \underbrace{\int_{\Gamma} \nabla u(\mathbf{x}') G(\mathbf{x}, \mathbf{x}') \mathbf{n} d\Gamma'}_{=V\partial_n u} \quad (2.12)$$

with $\Gamma = \partial\Omega$ and

$$\alpha = \begin{cases} 0 & \mathbf{x} \notin \Omega \\ \frac{1}{2} & \mathbf{x} \in \Gamma \\ 1 & \mathbf{x} \in \Omega \setminus \Gamma \end{cases} \quad (2.13)$$

G is the Green's function of the Laplacian which is

$$G(\mathbf{x}, \mathbf{x}') = \begin{cases} -\frac{1}{2\pi} \ln |\mathbf{x} - \mathbf{x}'| & \text{for } d = 2 \\ \frac{1}{4\pi} \frac{1}{|\mathbf{x} - \mathbf{x}'|} & \text{for } d = 3 \end{cases} \quad (2.14)$$

Using these representation formulas allow to reduce the dimension of the problem by one, which simplifies the discretization. Additionally using boundary elements allows to handle problems with an open boundary, if they fulfill some radiation condition like $u \rightarrow 0$ for $r \rightarrow \infty$, since only the finite boundaries need to be considered in that case. The drawback of the method is that it is based on integral equations, which lead to dense matrices. Furthermore the description of inhomogeneous and non-linear problems becomes more complicated. Finally the evaluation of the integrals of Green's functions may be very difficult.

If Eqn. (2.12) is evaluated at a single location this method is called collocation-BEM. If the integration domain can be split into simple elements like triangles or squares there exists some analytical formulas for the occurring integrals. In general or especially when Galerkin-BEM is used, numerical integration methods are needed. For a Galerkin-BEM method Eqn. (2.12) is evaluated in a weak sense, by multiplying it with a test function and integration over the whole space. The advantage of the Galerkin-BEM is that it leads to symmetric matrices and it is easier to analyze.

The following integral operators are important for the definition of a boundary integral equation:

- the **Newton potential** \tilde{N}

$$\tilde{N}f(x) := \int_{\Omega} G(\mathbf{x}, \mathbf{x}') f(\mathbf{x}') d\Omega' \quad \text{for } x \in \Omega \quad (2.15)$$

describes the influence of source terms f on the right-hand-side of the underlying PDE. Since the integration domain is the whole volume Ω , evaluation of this term is very time consuming.

- the **Single-Layer potential** \tilde{V}

$$\tilde{V}\phi(x) := \int_{\Gamma} G(\mathbf{x}, \mathbf{x}') \phi(\mathbf{x}') d\Gamma' \quad \text{for } x \in \Omega \quad (2.16)$$

is a convolution of Green's functions with the potential ϕ given on the boundary of the problem. Therefore it describes the field created by a single layer of charges sitting on the surface.

- the **Double-Layer potential** \tilde{K}

$$\tilde{K}\phi(x) := \int_{\Gamma} \partial_{\mathbf{x}'} G(\mathbf{x}, \mathbf{x}') \phi(\mathbf{x}') \mathbf{n} d\Gamma' \quad \text{for } x \in \Omega \quad (2.17)$$

is a convolution of the gradient of Green's functions with the potential ϕ given on the boundary of the problem. Due to the gradient it describes the field created by a layer of dipoles or a double layer of charges with opposite sign sitting on the surface.

- the **adjoint Double-Layer potential** \tilde{K}'

$$\tilde{K}'\phi(x) := \int_{\Gamma} \partial_{\mathbf{x}} G(\mathbf{x}, \mathbf{x}') \phi(\mathbf{x}') \mathbf{n} d\Gamma' \quad \text{for } x \in \Omega \quad (2.18)$$

is the adjoint operator of the double-layer potential, which means that $\langle K'\phi, \psi \rangle_{\Gamma} = \langle \phi, K\psi \rangle_{\Gamma}$.

- the **Hyper-Singular operator** \tilde{W}

$$\tilde{W}\phi(x) := - \int_{\Gamma} \partial_{\mathbf{x}} \partial_{\mathbf{x}'} G(\mathbf{x}, \mathbf{x}') \phi(\mathbf{x}') \mathbf{n} d\Gamma' \quad \text{for } x \in \Omega \quad (2.19)$$

is strongly singular and thus not an integrable function if evaluated on the boundary Γ . Regularization techniques have to be used to evaluate its value.

The upper equations hold for evaluation points \mathbf{x} located inside of the domain Ω . For the boundary element method the integral operators need to be evaluated directly at the surface, where singularities of the Green's function occur. Using the trace operator γ , which means calculating the limit of a series of points closer and closer to the surface, allows to deduce the following behavior of the integral operators at the surface

$$\begin{aligned}
 N_0 &:= \gamma \tilde{N} \\
 N_1 &:= \partial_n \tilde{N} \\
 V &:= \gamma \tilde{V} \\
 K &:= \gamma \tilde{K} + \frac{1}{2} \\
 K' &:= \gamma \tilde{K}' - \frac{1}{2} \\
 W &:= \gamma \tilde{W}
 \end{aligned} \tag{2.20}$$

which shows that the double-layer-potential jumps at the surface, which is the reason for the α -term in equation (2.12), when it is evaluated directly at the boundary. Note that $\alpha = \frac{1}{2}$ is only correct for smooth surfaces. In general it must be $\alpha = \frac{\omega(\mathbf{x})}{4\pi}$, where $\omega(\mathbf{x})$ is the solid angle the body subtends at \mathbf{x} .

A representation equation for the Cauchy data $(\gamma u, \partial_n u)$ which contains all the upper integral operators is provided by the Calderón system

$$\begin{pmatrix} \gamma u \\ \partial_n u \end{pmatrix} = \begin{pmatrix} \frac{1}{2} - K & V \\ W & \frac{1}{2} + K' \end{pmatrix} \begin{pmatrix} \gamma u \\ \partial_n u \end{pmatrix} + \begin{pmatrix} N_0 f \\ N_1 f \end{pmatrix} \tag{2.21}$$

which is for example important to define the so-called symmetric FEM-BEM coupling method (see 3.3.3), which is the “best” known FEM-BEM coupling and provides symmetric matrices.

2.3 Hierarchical Matrices

For the solution of open-boundary problems as presented in the following chapters finite elements as well as boundary elements or a combination of both

can be utilized. Therefore it is interesting how the different methods perform asymptotically. These considerations can be done in 2D or 3D, which makes a slight difference in the relative performance. One starts with the assumption that the number of elements of a certain dimension behaves like $N_{3D} \propto L^3$, $N_{2D} \propto L^2$, and $N_{1D} \propto L$, where L is the characteristic length of the problem. Further assuming that iterative solvers for FEM and BEM case show similar performance, which means that they need approximately the same number of iterations to reach the desired accuracy, allows to compare only the matrix-vector-product costs of FEM and BEM to assess to total performance.

Since FEM matrices are sparse their matrix-vector-product scales like $O(N)$, whereas the dense matrix-vector-product used by the BEM scales like $O(N^2)$. In the 2D case this leads to a $O(L^2)$ -behavior for the FEM as well as for the BEM, but in the 3D case one gets $O(L^3)$ for the FEM, but $O(L^4)$ for the BEM. This means that for large scale systems the BEM is the limiting part of coupled FEM-BEM methods if dense matrices are used. In order to improve the performance of the boundary element method, the dense matrices could be compressed in some way in order to reduce storage as well as computational costs.

One method to compress these dense matrices are hierarchical matrices (sometimes also called supermatrices), which provide a matrix-vector product of order $O(N \log N)$. This results in a total order of $O(L^3 \log L)$ for the 3D BEM, which makes the BEM asymptotically more efficient than the FEM. The main idea behind hierarchical matrices is to represent dense interaction matrices, by low-rank-approximations. It can be shown that the whole interaction matrix cannot be represented in this way. Only blocks which correspond to long-range interactions are compressible [7]. Therefor the creation of a \mathcal{H} -matrix can be split into three main steps:

- **Create a 1D cluster-tree T_I :** A cluster-tree is a hierarchical structure containing index sets. The toplevel node is equal to an indexset containing all nodes of the problem. For the creation of a 1D cluster-tree the indexset is then divided into parts, in a way that nodes of equal subsets are physically located near to each other, whereas nodes from different subsets should be

well separated. This so-called partitioning can be based on geometrical information or on some sort of connectivity matrix, as it is the case for FEM matrices. This partitioning is recursively executed until the leaves of the tree are reached, which means that all the indexsets only contain less than a certain minimal number of elements. An example 1D cluster tree for 8 elements is shown in Fig. 2.2.

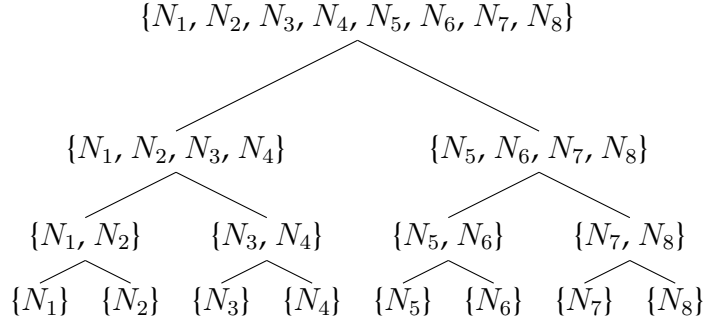


Figure 2.2: Example of a 1D-cluster-tree for 8 nodes. The main work which has to be done to create such a cluster tree is to renumber the nodes N_i in a way that nodes with similar indices are located near to each other. This is a prerequisite for the compressibility of the corresponding matrix blocks in the final stage.

- **Create a 2D block-cluster-tree $T_{I \times I}$:** An interaction matrix is represented by a so-called block-cluster-tree which is a tensor-product of two 1D cluster-trees. Within this stage the separation into near-field and far-field-interaction is performed by evaluating an admissibility-criterion for each node of the tree. If a node of the tree and thus the corresponding block of the interaction matrix is admissible, this means that it corresponds with a long-range-interaction and is therefore compressible by low-rank RK-matrices. A typical admissibility criterion looks like

$$\max(\text{diam}(I_A), \text{diam}(I_B)) \leq \eta \text{dist}(I_A, I_B) \quad (2.22)$$

where I_A and I_B are the target- as well as destination-indexsets and the parameter η allows to determine where the border between near- and far-field-interaction is located.

- **Create Low-Rank-Approximations:** The last step is to calculate the matrix elements of the hierarchical matrix. To that end the block-cluster-tree is recursively scanned until an admissible block is found. In order to calculate a low-rank approximation in form of an RK-matrix algorithms like Adaptive Cross Approximation(ACA) [8], or Hybrid Cross Approximation(HCA) [9] can be utilized. Those algorithms allow to set up the approximate matrix without the need to evaluate the whole original matrix. Only a certain number of crosses of the original matrix elements need to be evaluated in order to get an approximation of the matrix. If the algorithm ends up with a leave which is not admissible, the corresponding matrix block is not compressed and stored as an ordinary dense matrix. By this way the total interaction matrix is separated into different sized blocks as visualized in Fig. 2.3

Using the compressed RK-matrix-format as described above directly reduces the storage size of an $N \times N$ -block from $O(N^2)$ to $O(Nk)$, where k is the used rank, which is typically much smaller than N . Since not the whole matrix is representable by a single RK-matrix, but only individual block can be compressed, the storage size of the whole $N \times N$ -matrix can be reduced from $O(N^2)$ to $O(k N \log N)$. In order to utilize matrices within this compressed format, all usual matrix operations need to be performed directly within this format. As a useful tool the truncation operator T_k can be defined to truncate an arbitrary matrix to a certain rank. Alternatively the truncation operator T_ϵ can be used which truncates an arbitrary matrix to a low-rank-approximation with a maximum error ϵ .

Definition 2.2 (Rank Truncation T_k) Let $M \in \mathbb{R}^{n \times m}$ and $k \in \mathbb{N}$. We define the truncation operator $T_k : \mathbb{R}^{n \times m} \rightarrow R(k, n, m)$, $M \rightarrow \tilde{M}$, where \tilde{M} is the best approximation of M in the set of rank k matrices $R(k, n, m)$. The matrix \tilde{M} is called “truncation of M to rank k ”.

The truncation operator can be extended to \mathcal{H} -matrices by applying it block-wise to all admissible leaves of the matrix. Doing this allows to define basic

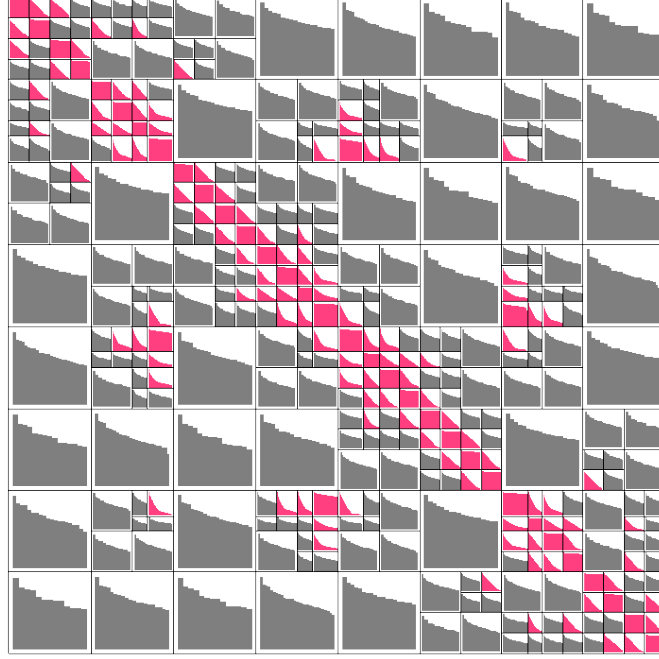


Figure 2.3: Example \mathcal{H} -matrix of a single-layer-potential which shows the separation into different sized admissible blocks (gray). If a block is not admissible but the block-size would fall below a certain minimal size, if further split, this block is stored as an ordinary dense matrix (magenta). The bars inside of each blocks visualizes the first singular values of the corresponding block. If the singular values decrease rapidly this is an indication that the corresponding block is well compressible, since only few singular values are required in order to gain the desired accuracy.

matrix operations for \mathcal{H} -matrices in a straight-forward way. The main problem is that the addition as well as the multiplication of RK-matrices does not preserve the rank of the matrices. Therefore the result needs to be truncated afterwards:

$$A \oplus B := T_k(A + B) \quad (2.23)$$

$$A \otimes B := T_k(AB) \quad (2.24)$$

Since it is not guaranteed that the rank of the result is equal to those of the operands, it is often preferred to use the error dependent truncation operator T_ϵ . Detailed bounds for the memory consumption as well as computational costs of matrix operations depend on the sparsity and/or the idempotency of the

underlying block-cluster-tree (see [7] for some detailed results). As a very crude approximation one can say that \mathcal{H} -matrices provide $O(k N \log(N))$ operations as well as storage consumption, whereas H^2 -matrices further reduce the costs to $O(k^2 N)$.

A very comprehensive introduction to \mathcal{H} -matrices and their applications can be found at [10]

2.4 Newton-Method

Both micromagnetic equations and Maxwell's equations for non-linear materials lead to systems of nonlinear equations. The most common methods to solve a system of non-linear equations are variants of the Newton method. Alternatively one could use some simpler methods like bisection or fixpoint iteration, but compared with the Newton method these provide less than quadratic convergence rates. The Newton method is based on a Taylor expansion of the system function at the current bias point

$$f_i(\mathbf{x} + \Delta\mathbf{x}) = f_i(\mathbf{x}) + \underbrace{\frac{\partial f_i}{\partial x_j} \Big|_{\mathbf{x}}}_{=\mathbf{J}} \Delta x_j + \underbrace{\frac{\partial^2 f_i}{\partial x_j \partial x_k} \Big|_{\mathbf{x}}}_{=\mathbf{H}} \Delta x_j \Delta x_k + \dots \quad (2.25)$$

where \mathbf{J} is the system Jacobian and \mathbf{H} is the Hessian matrix. Truncating this equation after the linear term allows to get an explicit formula for the approximated root, where the function is equal to 0:

$$\mathbf{J}_{ij}(\mathbf{x}) \Delta x_j = -f_i(\mathbf{x}) \quad (2.26)$$

Typically this linearized system of equations is solved within each non-linear step of the Newton method, instead of explicitly calculating the matrix inverse, because it is more efficient. The Newton method provides quadratic convergence, if the initial state is already near to the root. The initial value is therefore a crucial point when using a Newton-type method. Quadratic convergence can be proven by simply inserting the exact solution \mathbf{x}^* into Eqn. (2.25) and using the Lagrange form of the Taylor series expansion remainder, which means that

all higher order contributions are considered by evaluating the quadratic term at a certain unknown location ξ between x^* and x_n . For sake of simplicity the derivation is shown for a single variable x :

$$0 = f(x_n) + f'(x_n)(x^* - x_n) + \frac{1}{2}f''(\xi_n)(x^* - x_n)^2 \quad (2.27)$$

Dividing by $f'(x_n)$ and considering $x_{n+1} = x_n - \frac{f(x)}{f'(x)}$ directly leads to the mentioned convergence rate of the error ϵ :

$$\underbrace{x^* - x_{n+1}}_{\epsilon_{n+1}} = -\frac{f''(\xi_n)}{2f'(x_n)} \underbrace{(x^* - x_n)^2}_{\epsilon_n^2} \quad (2.28)$$

Calculating the Jacobian of a multivariate system of equations can sometimes be a very time consuming task. If this is the case so-called Quasi-Newton methods can be beneficial. Instead of analytically calculating the Jacobian, it is approximated by the function values of previous iteration steps. Examples for such Quasi-Newton methods are the secant method in the one-dimensional case, as well as Broyden's Method for the multivariate case.

Another class of Newton-like methods are Inexact-Newton methods [11], where parts of the Newton system are only solved approximately. Typically this is the case for large problems, where the linear system of equations within each Newton step can only be solved approximately by means of some iterative methods. Analyzing such methods allows to estimate the influence of the error bound of the iterative solver on the convergence rate of the Newton iteration.

Extending Newton's method to higher order terms within the Taylor expansion results in even better convergence rates. Naive implementations would need to evaluate the Hessian or even higher order derivatives and the system which has to be solved within each step would become nonlinear, which again requires some sort of Newton-method to be solved. Nevertheless it is possible to design higher order Newton methods, so-called Householder methods, which avoid the necessity of solving a non-linear system within every iteration. Furthermore there even exist methods which only require evaluation of the function and its Jacobian at certain locations - no higher order derivatives are needed (see e.g. [12]).

As already mentioned the initial state of the Newton iteration is very crucial. If it is chosen too far away from the root, the method may converge very slowly, or it may even diverge. In order to overcome this limitation some globalization strategies have been proposed, which result in a convergent iteration for arbitrary initial states. Two of these methods, the homotopy method as well as the linesearch method will be introduced in Chapter 4 since they produce stable and reliable methods for the solution of nonlinear systems.

Micromagnetism

3.1 Introduction

Micromagnetism is a continuum theory, which allows to investigate magnetic phenomena in the nanometer scale, like the domain formation in ferromagnetic materials. One of the key ideas of Micromagnetism is that ferromagnetic materials consist of individual spins, which can be described by a magnetic moment density with constant absolute value. In order to derive the basic micromagnetic equations, we start from the Gibbs free energy G of the micromagnetic system, which consists of the internal energy U as well as the interaction energy of magnetic moments \mathbf{J} with the external field \mathbf{H}_{ext} , also called Zeeman energy:

$$G = U - \underbrace{\int_{\Omega} \mathbf{H}_{\text{ext}} \cdot \mathbf{J} d\Omega}_{E_{\text{zeeman}}} \quad U = E_{\text{ex}} + E_{\text{ani}} + E_{\text{stray}} + \dots \quad (3.1)$$

The internal energy U itself consists of several contributions. The most common ones are the quantum-mechanical exchange energy E_{ex} , the magnetocrystalline anisotropy energy E_{ani} , and the magnetic strayfield energy E_{stray} . Additional terms like the magneto-elastic energy E_{elastic} , or the stochastic thermal energy E_{thermal} may be considered for some special problems.

In the following subsection the different energy contributions will be described in detail.

3.1.1 Exchange energy

Handling the quantum-mechanical exchange interaction is one of the key features of Micromagnetism, since this allows to use a continuum theory to describe effects, which otherwise would require one spin per atom such as an Ising model or something similar. Using a continuum theory therefore extends the usable system sizes by several orders of magnitude. In order to derive the continuum version of the exchange energy, one starts from the Heisenberg Hamiltonian

$$\mathcal{H} = - \sum_{i,j \neq i} J_{ij} \mathbf{S}_i \cdot \mathbf{S}_j \quad (3.2)$$

where \mathbf{S}_i are the individual spins of the system and J_{ij} is the exchange integral. Since the exchange integral decays strongly with the distance of the corresponding spins only the nearest neighbors need to be considered in the upper sum. Treating the spins classically, performing a Taylor expansion of the scalar product and omitting the constant terms one gets (see also [13])

$$E_{\text{heisenberg}} = JS^2 \sum_i \sum_{j=NN} |\hat{\mathbf{S}}_i - \hat{\mathbf{S}}_j|^2 \quad (3.3)$$

where $\hat{\mathbf{S}}_i$ is unit vector in direction of the spin \mathbf{S}_i . The continuum approximation can now be derived by transforming this sum in an integral and additionally use the approximation $|\hat{\mathbf{S}}_i - \hat{\mathbf{S}}_j| \approx |(r_{ij} \cdot \nabla) \hat{\mathbf{S}}|$, where r_{ij} is the distance of the spins i and j

$$E_{ex} = \frac{JS^2 z}{a} \int_{\Omega} (\nabla \hat{\mathbf{S}}_x)^2 + (\nabla \hat{\mathbf{S}}_y)^2 + (\nabla \hat{\mathbf{S}}_z)^2 d\Omega \quad (3.4)$$

or

$$E_{ex} = \frac{A}{J_s^2} \int_{\Omega} (\nabla \mathbf{J}_x)^2 + (\nabla \mathbf{J}_y)^2 + (\nabla \mathbf{J}_z)^2 d\Omega \quad (3.5)$$

where z is the number of next neighbors and a is their distance. In the latter version the spin density is replaced by the magnetic polarization density with the saturation polarization J_s and all constants are collected in the exchange constant A . The value of A depends on the used material and can be determined by experiments or by ab-initio simulations. For ferromagnetic materials the exchange constant is positive and thus the minimal energy $E_{ex} = 0$ corresponds with a constant polarization where all spins are parallel aligned.

3.1.2 Magnetocrystalline anisotropy energy

The crystalline structure of a material influences the orbits of the electron and due to the spin-orbit coupling this causes spins to preferably align in certain directions. The actual shape of the anisotropy energy therefore depends on the crystal system. Crystal systems with one axis of high symmetry lead to a single axis in which spins are easier to align - the so-called *easy axis*. This kind of anisotropy is called *uniaxial anisotropy* and obeys the following formula

$$e_{ani}^{uni} = K_1^{uni} \left[1 - \left(\frac{\mathbf{J} \cdot \mathbf{k}}{J_s} \right)^2 \right] + K_2^{uni} \left[1 - \left(\frac{\mathbf{J} \cdot \mathbf{k}}{J_s} \right)^4 \right] \quad E_{ani}^{uni} = \int_{\Omega} e_{ani}^{uni} d\Omega \quad (3.6)$$

where \mathbf{k} denotes to direction of the easy axis and K_1^{uni} and K_2^{uni} are the uniaxial anisotropy constants. Examples for crystal system which show uniaxial anisotropy are hexagonal, tetragonal or rhombohedral systems.

For cubic crystals the anisotropy energy looks like the following

$$e_{ani}^{cubic} = K_1^{cubic} (\alpha^2 \beta^2 + \beta^2 \gamma^2 + \gamma^2 \alpha^2) + K_2^{cubic} \alpha^2 \beta^2 \gamma^2 \quad E_{ani}^{cubic} = \int_{\Omega} e_{ani}^{cubic} d\Omega \quad (3.7)$$

where $\alpha = \mathbf{J} \cdot \mathbf{e}_{x'}$, $\beta = \mathbf{J} \cdot \mathbf{e}_{y'}$, $\gamma = \mathbf{J} \cdot \mathbf{e}_{z'}$ and $\mathbf{e}_{x'}$, $\mathbf{e}_{y'}$, $\mathbf{e}_{z'}$ are the basis vectors of a rotated coordinate system. For $K_2^{cubic} = 0$ and $K_1^{cubic} > 0$ this relation leads to easy axes which are equal to the basis axes $\mathbf{e}_{x'}$, $\mathbf{e}_{y'}$, $\mathbf{e}_{z'}$, whereas for $K_1^{cubic} < 0$ the easy axes are $\mathbf{e}_{x'} \pm \mathbf{e}_{y'} \pm \mathbf{e}_{z'}$.

3.1.3 Strayfield Energy

Interaction between the dipoles within the ferromagnetic material gives rise to the magnetostatic strayfield energy

$$E_{stray} = -\frac{1}{2} \int_{\Omega} \mathbf{J} \cdot \mathbf{H}_d d\Omega \quad (3.8)$$

where \mathbf{H}_d is the magnetic field created by the polarization \mathbf{J} . The infinite range of the electro-magnetic interaction leads to an open-boundary-problem, which requires the use of sophisticated numerical methods to be solved. Direct integration via a discretized version of

$$\mathbf{H}_d(\mathbf{x}) = -\frac{1}{4\pi} \iiint_{\Omega} \frac{\nabla \cdot \mathbf{M}(\mathbf{x}') (\mathbf{x} - \mathbf{x}')}{|\mathbf{x} - \mathbf{x}'|^3} d^3\mathbf{x}' + \frac{1}{4\pi} \iint_{\partial\Omega} \frac{\mathbf{x} - \mathbf{x}'}{|\mathbf{x} - \mathbf{x}'|^3} \mathbf{M}(\mathbf{x}') \cdot \mathbf{n} d^2\mathbf{x}' \quad (3.9)$$

results in dense matrices and is therefore only possible for relatively small problems. Section 3.3 or [14] describes some recent methods that can be used to handle also large scale systems.

3.1.4 The effective field and Brown's equations

Summarizing all contributions to the total Gibbs free energy (3.1) gives

$$G = \int_{\Omega} \left[\frac{A}{J_s^2} (\nabla \mathbf{J})^2 + e_{ani} - \frac{1}{2} \mathbf{J} \cdot \mathbf{H}_d - \mathbf{J} \cdot \mathbf{H}_{ext} \right] d\Omega \quad (3.10)$$

The functional derivative of this energy with respect to the polarization directly gives the negative effective field \mathbf{H}_{eff} , which acts on the polarizations:

$$\mathbf{H}_{eff} = -\frac{\delta G}{\delta \mathbf{J}} = \frac{2A}{J_s^2} \Delta \mathbf{J} - \frac{\partial e_{ani}}{\partial \mathbf{J}} + \mathbf{H}_d + \mathbf{H}_{ext} \quad (3.11)$$

There exists a close relation between the calculation of the effective field and the calculation of the ground state of the micromagnetic system. The ground state is given by the minimum of the Gibbs free energy, which can be calculated by setting the variation of the energy equal to zero. The result is then integrated

by parts in order to factor out $\delta \mathbf{J}$, which gives rise to an additional boundary condition for the polarization \mathbf{J} . Since the absolute value of the polarization is constrained \mathbf{H}_{eff} does not need to be identical to zero at the minimum of G , which can be seen by applying the constraint by only allowing variations of the form $\delta \mathbf{J} = \mathbf{J} \times \delta \phi$:

$$\begin{aligned}
\delta G &= \iiint_{\Omega} \left[\frac{A}{J_s^2} 2(\nabla \mathbf{J}) \right] \delta(\nabla \mathbf{J}) + \left[\frac{\partial e_{\text{ani}}}{\partial \mathbf{J}} - \frac{1}{2} \mathbf{H}_{\text{d}} - \mathbf{H}_{\text{ext}} \right] \cdot \delta \mathbf{J} d^3 \mathbf{x}' \\
&= \iiint_{\Omega} \left[-\frac{2A}{J_s^2} (\Delta \mathbf{J}) + \frac{\partial e_{\text{ani}}}{\partial \mathbf{J}} - \frac{1}{2} \mathbf{H}_{\text{d}} - \mathbf{H}_{\text{ext}} \right] \cdot \delta \mathbf{J} d^3 \mathbf{x}' + \iint_{\partial \Omega} \frac{2A}{J_s^2} \frac{\partial \mathbf{J}}{\partial \mathbf{n}} \cdot \delta \mathbf{J} d^2 \mathbf{x}' \\
&= \iiint_{\Omega} \left[-\frac{2A}{J_s^2} (\Delta \mathbf{J}) + \frac{\partial e_{\text{ani}}}{\partial \mathbf{J}} - \frac{1}{2} \mathbf{H}_{\text{d}} - \mathbf{H}_{\text{ext}} \right] \times \mathbf{J} \cdot \delta \phi d^3 \mathbf{x}' + \\
&\quad + \iint_{\partial \Omega} \frac{2A}{J_s^2} \frac{\partial \mathbf{J}}{\partial \mathbf{n}} \times \mathbf{J} \cdot \delta \phi d^2 \mathbf{x}' \\
&= 0
\end{aligned} \tag{3.12}$$

Since variations are arbitrary the whole integral can only be identical to zero, if the integrands themselves are zero. Notice that $\frac{\partial \mathbf{J}}{\partial \mathbf{n}} \times \mathbf{J} = 0$ means that $\frac{\partial \mathbf{J}}{\partial \mathbf{n}} = 0$, because \mathbf{J} and $\frac{\partial \mathbf{J}}{\partial \mathbf{n}}$ are always orthogonal to each other. Finally inserting the actual form of the effective field gives the so-called Brown equations [15]:

$$\Rightarrow \begin{cases} \mathbf{H}_{\text{eff}} \times \mathbf{J} = 0 & \text{on } \Omega \\ \frac{\partial \mathbf{J}}{\partial \mathbf{n}} = 0 & \text{on } \partial \Omega \end{cases} \tag{3.13}$$

The meaning of these equations is that due to the polarization constraint the effective field at the minimum Gibbs free energy must not be equal to zero, but only orthogonal to the actual polarization.

3.2 The Landau-Lifshitz-Gilbert equation

Solving Brown's equations or minimizing the Gibbs free energy directly is often not sufficient, because the energy landscape of micromagnetic simulations may

consist of several local minima, which correspond with stable states of the system. Using energy minimization one can only determine the starting point of the minimization, but there is almost no possibility to predict which of the local minima will be reached by the numerical minimization algorithm. One way to overcome this limitation is to switch to a dynamic description of the polarization. By this way theoretical minima which will never be reached in practice can be avoided and additionally the possibility to describe magnetization dynamics and not only the ground-state is important for various applications.

3.2.1 Standard form

The dynamics of the magnetic polarization can be described by means of the Landau-Lifshitz-Gilbert equation

$$\dot{\mathbf{J}} = -\frac{|\gamma|}{1+\alpha^2} \mathbf{J} \times \mathbf{H}_{\text{eff}} - \frac{\alpha|\gamma|}{1+\alpha^2} \frac{1}{J_s} \mathbf{J} \times \mathbf{J} \times \mathbf{H}_{\text{eff}} \quad (3.14)$$

where α is the Gilbert damping constant, J_s is the saturation polarization and $|\gamma| = \mu_0 |\gamma_e| = 2.210175 \cdot 10^5 \text{ m/As}$ is the reduced gyromagnetic ratio (with μ_0 the permeability of the free space and γ_e the gyromagnetic ratio of the electron).

The first term describes the simple precessional motion of the magnetic polarization around the effective field, whereas the second phenomenological term describes damping, which leads to a rotation into the direction of the effective field (see Fig. 3.1 for a visualization of the precessional and damping components). Without damping the magnetic polarization would perform a stationary rotation with the Larmor frequency $\omega = |\gamma| H_{\text{eff}}$, which is typically in the GHz range. The physical reason for the damping in ferromagnetic materials are for example eddy currents, magnon-magnon-, and magnon-phonon-interactions. In contrast to the original Landau-Lifshitz equation the damping coefficients in the Landau-Lifshitz-Gilbert equation (3.14) are chosen in a way that $\dot{\mathbf{J}} = 0$ if $\alpha \rightarrow \infty$.

Multiplying the equation with \mathbf{J} shows gives $\mathbf{J} \cdot \dot{\mathbf{J}} = 0$, which can also be written as $\frac{d}{dt} \left(\frac{1}{2} \mathbf{J} \cdot \mathbf{J} \right) = 0$. So one can deduce that for the continuous version

of the LLG the absolute value of the polarization \mathbf{J} is constant over the time. Notice that this must no longer be true for the time-discretized equations.

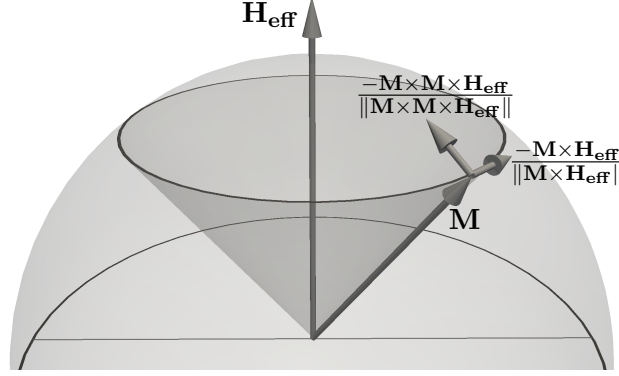


Figure 3.1: Visualization of the directions of the precessional- and damping-term of the LLG equation for a given magnetization \mathbf{M} within an effective field \mathbf{H}_{eff} .

3.2.2 Gilbert form

A different, but equivalent, formulation of the LLG has been proposed by Gilbert [16, 17] and is therefore often called the Gilbert form of the LLG. This form can be derived by taking $\mathbf{J} \times$ of Eqn. (3.14), which transforms the second term into $\mathbf{J} \times \mathbf{J} \times \mathbf{H}_{\text{eff}}$ and the third one becomes proportional to $\mathbf{J} \times \mathbf{J} \times \mathbf{J} \times \mathbf{H}_{\text{eff}} = -J_s^2 \mathbf{J} \times \mathbf{H}_{\text{eff}}$. Using this relation allows to replace the damping term of the original LLG with a term depending on $\dot{\mathbf{J}}$:

$$\dot{\mathbf{J}} = -|\gamma| \mathbf{J} \times \mathbf{H}_{\text{eff}} + \frac{\alpha}{J_s} \mathbf{J} \times \dot{\mathbf{J}} \quad (3.15)$$

Since we only used equivalence transformations to derive this version of the LLG the conservation of the absolute value of \mathbf{J} still holds. One application of this form of the LLG is the prove of existence and non-uniqueness of weak solutions [18].

3.2.3 Alternative form

Another form of the LLG can be derived by means of the vector identity $\mathbf{J} \times \mathbf{J} \times \mathbf{H}_{\text{eff}} = \mathbf{J} (\mathbf{J} \cdot \mathbf{H}_{\text{eff}}) - \mathbf{H}_{\text{eff}} J_s^2$. After applying this relation to (3.14) one again takes $\mathbf{J} \times$ and eliminates the $\mathbf{J} \times \mathbf{H}_{\text{eff}}$ term by a linear combination with the original equation, which finally leads to

$$\alpha J_s \dot{\mathbf{J}} + \mathbf{J} \times \dot{\mathbf{J}} = |\gamma| J_s^2 \mathbf{H}_{\text{eff}} - |\gamma| (\mathbf{J} \cdot \mathbf{H}_{\text{eff}}) \mathbf{J} \quad (3.16)$$

The advantage of this version is that used in its weak form it allows to define a time-splitting scheme by defining $\mathbf{v} = \dot{\mathbf{J}}$, which if test functions are only defined in the orthogonal space of the actual \mathbf{J} is linear in \mathbf{J} also if the time-integration is done implicitly [19–22].

3.2.4 Weak Formulation

Transforming the LLG into its weak formulation allows to prove the existence and some characteristic features of the solutions [18]. As for the strong form of the LLG (see Section 3.2) also the weak form allows different formulations. Basically one starts from the strong form, multiplies the equation with a test function and integrates over the support of the test function. Then all terms containing second derivatives are partially integrated, which for the Gilbert form of the LLG leads to

$$\begin{aligned} \int_{\Omega} \left(\mathbf{v} - \frac{\alpha}{J_s} \mathbf{J} \times \mathbf{v} \right) \cdot \phi \, d\Omega &= - \int_{\Omega} |\gamma| (\mathbf{J} \times \mathbf{H}_{\text{eff}}) \cdot \phi \, d\Omega = \\ &= \int_{\Omega} \frac{2A|\gamma|}{J_s^2} \mathbf{J} \times \nabla \mathbf{J} \cdot \nabla \phi \, d\Omega - \int_{\Omega} |\gamma| \mathbf{J} \times (\mathbf{H}_{\text{stray}} + \mathbf{H}_{\text{eff}} + \mathbf{H}_{\text{eff}}) \cdot \phi \, d\Omega \end{aligned} \quad (3.17)$$

with $\mathbf{v} = \dot{\mathbf{J}}$. The surface integral, which arises due to the partial integration, vanishes due to the Brown condition $\frac{\partial \mathbf{J}}{\partial \mathbf{n}} = 0$ on Γ (see second part of 3.13, or [23] for a derivation in a micromagnetic context). One way to simplify the solution of this system of equations is to utilize the fact that $\mathbf{v} \cdot \mathbf{J} = 0$, which means that one searches for \mathbf{v} only in the orthogonal space of the actual \mathbf{J} . Therefore it is sufficient to use only test functions ϕ lying in this orthogonal space and

setting the parallel component of \mathbf{v} equal to 0. This method was proposed by Alouges [19] for the exchange-only case and independently generalized [21,22] to linear contributions of the LLG. In [5] this approach is also generalized to nonlinear field contributions. An additional advantage of this method is that arising system matrices are now linear in \mathbf{J} . The easiest way to see this, is to remember the alternative form of the LLG (3.16) and transform this into its weak representation. Then it is obvious that due to testing with functions orthogonal to \mathbf{J} the last term vanishes and the remaining equations are linear.

Another derivation which is closer to the practical implementation of such a scheme directly uses (3.17), where the test functions are expressed as $\phi = \mathbf{J} \times \omega$. This means that although ω is defined in the whole \mathbb{R}^3 testing is effectively limited to the tangential plane. Since one then has more equations than unknowns it is possible to treat one of the non-linear terms as if it was an additional unknown λ .

$$\int_{\Omega} \left(\mathbf{v} - \frac{\alpha}{J_s} \mathbf{J} \times \mathbf{v} \right) \cdot (\mathbf{J} \times \omega) \, d\Omega = - \int_{\Omega} |\gamma| (\mathbf{J} \times \mathbf{H}_{\text{eff}}) \cdot (\mathbf{J} \times \omega) \, d\Omega \quad (3.18)$$

$$\begin{aligned} \int_{\Omega} \left(-\mathbf{J} \times \mathbf{v} - \frac{\alpha}{J_s} \left(J_s^2 \mathbf{v} - \underbrace{(\mathbf{J} \cdot \mathbf{v})}_{=0} \mathbf{J} \right) \right) \cdot \omega \, d\Omega = \\ - \int_{\Omega} |\gamma| \left(J_s^2 \mathbf{H}_{\text{eff}} - \underbrace{(\mathbf{J} \cdot \mathbf{H}_{\text{eff}})}_{=\lambda} \mathbf{J} \right) \cdot \omega \, d\Omega \end{aligned} \quad (3.19)$$

As before the parallel component needs to be set to 0 as additional weak equation with test functions μ . By this way one ends up with a square system of equations of the unknowns \mathbf{v} and λ with a unique solution.

$$\int_{\Omega} (\mathbf{J} \times \mathbf{v} + \alpha J_s \mathbf{v}) \cdot \omega \, d\Omega + \int_{\Omega} |\gamma| \lambda \mathbf{J} \cdot \omega \, d\Omega = \int_{\Omega} |\gamma| J_s^2 \mathbf{H}_{\text{eff}} \cdot \omega \, d\Omega \quad (3.20)$$

$$\int_{\Omega} \mu (\mathbf{J} \cdot \mathbf{v}) \, d\Omega = 0 \quad (3.21)$$

3.3 Strayfield Calculation

As already mentioned shortly the most time consuming part of micromagnetic codes is the calculation of the magnetic strayfield. Numerical methods for this problem can be split into three parts: integral methods, differential methods and combinations of both. A short summary of some recently developed algorithms can be found in [14].

3.3.1 Integral Methods

Since one only needs to evaluate the strayfield within the ferromagnetic material directly evaluating Eqn. (3.8) would be an easy method to cope with the open-boundary problem. Alternatively to directly calculating the strayfield $\mathbf{H}_{\text{stray}}$ it is often preferred to calculate the correlated potential field u and then derive $\mathbf{H}_{\text{stray}} = -\nabla u$ within a post-processing step.

$$u(\mathbf{x}) = -\frac{1}{4\pi} \iiint_{\Omega} \frac{\nabla \cdot \mathbf{M}(\mathbf{x}')}{|\mathbf{x} - \mathbf{x}'|} d^3\mathbf{x}' + \frac{1}{4\pi} \iint_{\partial\Omega} \frac{\mathbf{M}(\mathbf{x}') \cdot \mathbf{n}}{|\mathbf{x} - \mathbf{x}'|} d^2\mathbf{x}' \quad (3.22)$$

Doing this allows to reduce the number of degrees of freedom by a factor of 3, and allows to use a single-layer- instead of a double-layer-potential. The most efficient methods to solve the integral equations are summarized in the following:

- **Fast Fourier Transform Method [24–28]:** The most efficient way to handle this integral is to use an FFT and utilize the convolution theorem. By this way the convolution in real-space can be transformed into a multiplication in the Fourier-space, which can be evaluated by $N \log(N)$ operations (where N is the number of unknowns and is typically in the order of 10^6). Various highly optimized FFT libraries are available which allow to utilize nearly every hardware in an optimal way. One drawback of the method is, that it is restricted to equidistant rectangular grids.
- **Tensor Grid Methods [29,30]:** For tensor grid methods the Green's function is represented as an integral of a Gaussian function, which allows

to factorize the volume integration in a very efficient way. Using a special tensor-grid-representation of the magnetization allows to achieve sub-linear computational costs. But if the magnetization first has to be transformed into this representation the sub-linear performance is lost and unfortunately calculating directly within the tensor-grid-representation is very complicated since most of the common algorithms cannot directly be applied to tensor grids without modifications.

- **Fast Multipole Method [31,32]:** The FMM utilizes the fact that the far-field of a given source distribution can be described very accurately by the first few terms of the multipole expansion of the distribution. Introducing a hierarchical grid structure and approximating far-field interaction by multipole expansions leads to the so-called tree-code which allows to reduce computational costs of the strayfield calculation to $O(Np \log(N))$, where p is the order of the multipole expansion and the $\log(N)$ is the approximate number of hierarchy levels needed to reach the required accuracy. FMM uses an additional optimization, where the coefficients of the far-field interactions are transformed to local near-field coefficients in the target area. This trick allows to further reduce the computational costs to $O(p^2 N)$.
- **Hierarchical Matrix Method [10,33–35]:** This method is very similar to the FMM, since it is also based on a hierarchical problem description. As before only blocks which correspond with far-field interaction are stored in a compressed format. In contrast to the FMM low-rank matrices instead of multipole moments are used. These low-rank approximations reproduce the largest singular values of the exact matrix, but only need a fraction of the storage size. An additional benefit of using this method is that several matrix operations can also be defined for hierarchical matrices and many of them can be calculated very efficiently. For example one can calculate the approximate inverse or LU decomposition of a matrix and use it as an efficient preconditioner for iterative methods. The order of these operations using \mathcal{H} -matrices is $O(N \log^2 N)$. Another example are matrix

equations such as the Lyapunov, Sylvester and Riccati equation for which the order of $O(N^3)$ when using standard linear algebra can be reduced to $O(N \log^4(N))$ by means of \mathcal{H} -matrices. The parallelization of \mathcal{H} -matrices is presented in Chapter 6.

3.3.2 Differential Methods

Differential methods for strayfield calculation are based on the differential form of Maxwell's equations

$$\text{rot } \mathbf{H} = \mathbf{j} \quad \text{div } \mathbf{B} = 0 \quad (3.23a)$$

$$\mathbf{B} = \mu_0 \mathbf{H} + \mathbf{J} = \mu_0 \mu \mathbf{H} \quad (3.23b)$$

where \mathbf{H} is the magnetic field, \mathbf{J} is the magnetic polarization, \mathbf{j} is the current density, $\mu_0 = 4\pi 10^{-7} \text{Vs/Am}$ is the vacuum permeability and μ is the relative permeability.

The only source of the strayfield is the magnetic polarization \mathbf{J} within the magnetic material. Therefore one sets the current density $\mathbf{j} = 0$, which allows to describe the magnetic field as the gradient of a potential field u , which then directly solves the first Maxwell equation. Together with the second Maxwell equation the following differential equation for the potential u can be derived:

$$\Delta u = \frac{1}{\mu_0} \text{div } \mathbf{J} \quad \mathbf{H} = -\nabla u \quad (3.24)$$

In order to get a unique solution one additionally has to apply proper boundary conditions. For an open-boundary problem the only boundary-condition which is known is that the potential u should be 0 at infinity. A very easy approach to this problem is to simply include a large bounding box around the magnetic region and set the potential equal to 0 at the surface of this bounding box. Although this method is very easy to implement, it leads to systematic errors. Some more comprehensive methods to deal with this problem are shortly summarized in the following. All of them have in common that they try to approximate the correct behavior of the potential in the exterior region and therefore include intrinsic systematic errors. An additional practical complication of

those methods is that at least parts to the exterior region need to be discretized, which often requires special capabilities of the used meshing algorithm.

- **Shell Transformation [36]:** This method defines a bounding box of a certain geometry around the magnetic region and then adds an additional layer in which a coordinate transformation is applied. This transformation is defined in a way that the region of this additional layer is transformed to the whole exterior region and it also depends on the shape of the additional layer. As a consequence the integrals which arise from the weak formulation, become more complicated and can in general only be solved numerically.
- **Infinite Elements [37]:** Very similar to the previous method are infinite elements. Instead of transforming the coordinates, the elements are scaled and special basis-functions are defined, which assures the correct potential decay.
- **Ballooning Elements [38]:** The basic idea of this method is to recursively add additional surface layers around the original geometry and then eliminate all nodes between the added layers, by means of the Schur-complement. After a certain number of recursions the outermost nodes are set to 0 which results in a system matrix which is of the same size as the original matrix. In order to reduce the computational cost of the matrix setup, the added surface nodes are created by radially scaling of the original surface points. By this way the system matrices of the additional layers are always scaled versions of one and the same matrix.
- **Asymptotic Boundary Conditions [39]:** Since there is no magnetic polarization in the exterior region the potential equation becomes an ordinary Laplace equation. The solutions in spherical coordinates can be expressed in terms of spherical harmonics Y_{lm} :

$$u(r, \phi, \theta) = \sum_{l=0}^{\infty} \sum_{m=-l}^l B_{lm} \frac{1}{r^{l+1}} Y_{lm}(\phi, \theta) \quad (3.25)$$

Using this relation asymptotic boundary operators can be constructed, which make the highest order $1/r$ -terms vanish, if applied to a solution of the Laplace equation. For the lowest order operator B_1 one gets

$$B_1 u = \frac{\partial u}{\partial r} - \frac{u}{r} = O\left(\frac{1}{r^3}\right) \quad (3.26)$$

Inserting this additional relation into a FEM formulation allows to fix the boundary condition and therefore yields a unique result.

3.3.3 Mixed Methods

The advantages of integral- and differential-methods can be combined by using mixed-methods (as first proposed in [40]), which describe the inhomogeneous magnetic region using a differential-method (FEM) and open-boundary conditions are handled by an integral formulation (BEM). This leads to sparse system matrices corresponding to the magnetic volume as well as dense matrices, which are only defined on the boundary for a reduced number of degrees of freedom. Mixed-methods can be divided into direct, indirect and hybrid coupling-methods. Both direct and indirect FEM-BEM coupling use an integral representation of the exterior solution, which then forms a combined system together with the interior FEM equations. The difference between the two categories is that for direct coupling methods the exterior solution is represented by means of some physical quantities (such as the normal component of the magnetic field), whereas indirect methods may use some artificial mathematical quantities. In contrast to these methods hybrid coupling schemes do not solve a combined FEM-BEM system, but FEM and BEM part are handled sequentially, which may lead to numerical problems in some cases.

Since the FEM- and BEM-representation need to be connected at the surface, one needs to know the jump conditions of the magnetic properties. In the case of the magnetic field, one can derive that the normal component of the magnetic flux density \mathbf{B} as well as the tangential component of the magnetic field \mathbf{H} is continuous across the boundary, whereas the other components may contain

jumps.

$$\mathbf{n} \times (\mathbf{H}^+ - \mathbf{H}^-) = 0 \quad (3.27a)$$

$$\mathbf{n} \cdot (\mathbf{B}^+ - \mathbf{B}^-) = 0 \quad (3.27b)$$

The superscript $+$ or $-$ denote internal or external quantities. These conditions for the vector fields are equivalent to the following conditions for the scalar potential u :

$$u^+ = u^- \quad \frac{\partial u^+}{\partial \mathbf{n}} - \frac{\partial u^-}{\partial \mathbf{n}} = -\frac{1}{\mu_0} \mathbf{J} \cdot \mathbf{n} \quad (3.28)$$

The following enumeration will present some of the most important FEM-BEM coupling methods and shortly explain the principles which they are based on:

- **Hybrid Fredkin-Koehler FEM-BEM Coupling [41]:** This hybrid method makes use of the double-layer potential

$$u_{DL}(\mathbf{x}) = \frac{1}{4\pi} \iint_{\Gamma} d(\mathbf{x}') \frac{\mathbf{x} - \mathbf{x}'}{|\mathbf{x} - \mathbf{x}'|^3} d^2 \mathbf{x}' \quad (3.29)$$

with the dipole density d defined on a surface Γ , which is a solution of the homogeneous Laplace equation. At the surface the normal derivative of the potential is continuous whereas the potential itself shows a jump by d . One can now define u_1 as the solution of the Poisson equation (3.24) with Neumann boundary conditions at the surface of the magnetic material. Outside of the magnet u_1 is set to zero, which directly leads to $\frac{\partial u_1^+}{\partial \mathbf{n}} = \frac{1}{\mu_0} \mathbf{J} \cdot \mathbf{n}$ as proper boundary condition. The potential u_1 calculated this way shows the correct behavior inside of the magnet and also the correct jump-condition of the normal derivative at the surface, but unfortunately this potential alone is not continuous at the boundary of the magnet, but it jumps from u_1 inside to 0 outside. By adding a double-layer-potential with $d(\mathbf{x}) = u_1(\mathbf{x})$ it is possible to correct this jump while preserving the other boundary condition as well as the behavior inside of the magnet. A

further optimization of this method evaluates the double-layer potential u_2 only at the surface of the magnet and solves a homogeneous Laplace equation, with these surface-values as Dirichlet boundary conditions, in order to calculate the potential inside of the magnet. The equations to be solved can be summarized as:

$$\Delta u_1 = \frac{1}{\mu_0} \operatorname{div} \mathbf{J} \quad \text{on } \Omega \quad (3.30a)$$

$$\frac{\partial u_1}{\partial \mathbf{n}} = -\frac{1}{\mu_0} \mathbf{J} \cdot \mathbf{n} \quad \text{on } \Gamma \quad (3.30b)$$

$$\Delta u_2 = 0 \quad \text{on } \Omega \quad (3.30c)$$

$$u_2(\mathbf{x}) = \frac{1}{4\pi} \iint_{\Gamma} u_1(\mathbf{x}') \frac{\mathbf{x} - \mathbf{x}'}{|\mathbf{x} - \mathbf{x}'|^3} d^2 \mathbf{x}' \quad \text{on } \Gamma \quad (3.30d)$$

$$u = u_1 + u_2 \quad (3.30e)$$

- **Hybrid García-Cervera / Roma FEM-BEM Coupling [42]:** This coupling method is very similar to the Fredkin-Koehler method, but instead of the double-layer potential a single-layer potential is used:

$$u_{SL}(\mathbf{x}) = \frac{1}{4\pi} \iint_{\Gamma} s(x) \frac{1}{|\mathbf{x} - \mathbf{x}'|} d^2 x' \quad (3.31)$$

This potential is also a solution of the homogeneous Laplace equations, but in contrast to the double-layer-potential the potential u_{SL} is continuous at the surface Γ , whereas its normal derivative jumps by $s(x)$. Using these properties again allows to first solve an Poisson problem - now with Dirichlet boundary condition - and then correct the remaining jump-

condition using the single-layer potential.

$$\Delta u_1 = \frac{1}{\mu_0} \operatorname{div} \mathbf{J} \quad \text{on } \Omega \quad (3.32a)$$

$$u_1 = 0 \quad \text{on } \Gamma \quad (3.32b)$$

$$\Delta u_2 = 0 \quad \text{on } \Omega \quad (3.32c)$$

$$u_2(\mathbf{x}) = \frac{1}{4\pi} \iint_{\Gamma} \left(\frac{\partial u_1}{\partial \mathbf{n}} - \frac{1}{\mu_0} \mathbf{J} \cdot \mathbf{n} \right) \frac{1}{|\mathbf{x} - \mathbf{x}'|} d^2 \mathbf{x}' \quad \text{on } \Gamma \quad (3.32d)$$

$$u = u_1 + u_2 \quad (3.32e)$$

- **Direct Johnson-Nédélec FEM-BEM Coupling [43]:** Applying Green's second identity on the potential u and the Green's function G , choosing the whole exterior region as integration domain, which gives $\Delta u = 0$, and evaluating this equation at the surface of the magnet leads to

$$\iiint_{\mathbb{R}^3 \setminus \Omega} \underbrace{\Delta u}_{=0} G - u \underbrace{\Delta G}_{\delta(\mathbf{x}-\mathbf{x}')} d^3 \mathbf{x}' = \iint_{\Gamma} \nabla u^- G - u^- \nabla G d^2 \mathbf{x}' = -\frac{1}{2} u^- \quad (3.33)$$

where the last step uses the fact the the delta function is located directly at the surface which gives the $\frac{1}{2}$ factor. Since one integrates over the exterior region, u^- needs to be considered within the surface integrals. Using this equation yields a relation between the potential u^- and its normal derivative $\frac{\partial u^-}{\partial \mathbf{n}}$ at the boundary, which can be used as boundary conditions of an internal FEM description. This can be achieved by introducing a new variable $\phi = \frac{\partial u^-}{\partial \mathbf{n}}$ and solving the combined equation system for u and ϕ . Using this approach allows to handle arbitrary jump condition of the potential as well as its normal derivative in a very flexible way. The drawback of the method is that one has to deal with larger non-symmetric system matrices, due to the additional unknowns. Especially the dense BEM matrices lead to high computational costs, although these can be reduced by means of matrix compression techniques as presented in Section 3.3.1. Inserting the jump conditions (3.28) the equations that have to be solved

can be summarized as follows:

$$\Delta u = \frac{1}{\mu_0} \operatorname{div} \mathbf{J} \quad \text{on } \Omega \quad (3.34a)$$

$$\frac{\partial u}{\partial \mathbf{n}} = \frac{1}{\mu_0} \mathbf{J} \cdot \mathbf{n} + \phi \quad \text{on } \Gamma \quad (3.34b)$$

$$-\frac{1}{2}u = \iint_{\Gamma} \phi G - u \frac{\partial G}{\partial \mathbf{n}} d^2x' \quad \text{on } \Gamma \quad (3.34c)$$

- **Direct Symmetric FEM-BEM Coupling [44]:** From the mathematical view this is the best known FEM-BEM coupling method. It makes use of two additional integral operators - the adjoint double-layer potential and the hyper-singular integral operator. The main difference to the Johnson-Nédélec coupling is how the interior and exterior solutions are coupled. Depending of the sign of the BEM part the resulting bilinear forms are either symmetric or elliptic, which is advantageous for the efficient iterative solution of the system. On the other hand the numerical calculation of the hyper-singular integrals requires advanced integration methods, which makes them slightly more complicated in use.

Maxwell Equations

Portions of this chapter were previously published as [3] and have been reproduced with permission of the coauthors and in accordance with the publisher's policy. Content which was not generated by the author of this thesis is explicitly denoted. Copyright is held by Elsevier.

4.1 Introduction

This chapter deals with the standalone solution of Maxwell's equations, which is a key component of the multiscale method presented in Chapter 5. We focus on the solution of magnetostatic Maxwell's equations which arise from the full Maxwell's equations by assuming slowly changing fields. Therefore the time derivatives of the field variables can be neglected and the system of equations decouples into magnetic and electric equations. As the strayfield calculation presented in Section 3.3, magnetostatic Maxwell's equations represent an open boundary problem. Most of the methods presented for the solution of the strayfield problem can also be applied to the magnetostatic Maxwell's problem. In contrast to the strayfield calculation, where the polarization of the material was known, only the material behavior, which is a relation between the local effective field and the magnetic polarization, is given. The main problem is now

to find a polarization \mathbf{J} for a given geometry and certain external fields, which create a strayfield in a way that the total field at every location is related to the polarization via the specified constitutive law.

One starts from magnetostatic Maxwell's equations which are defined in the entire space. The full space is then divided into two partitions, since different representations may be used within these regions. The region Ω^+ contains all the magnetic parts of the problem which are described by the FEM, whereas Ω^- contains the surrounding non-magnetic volume described by the BEM. The superscripts $+$ and $-$ are therefore used for physical quantities within Ω^+ or Ω^- , respectively (see Fig. 5.1). Maxwell's equations, the material laws in both regions, and the jump conditions at the boundary of the magnetic parts are given by

$$\text{rot } \mathbf{H} = \mathbf{j} \qquad \mathbf{B}^+ = \mu^+ \mathbf{H}^+ \qquad (4.1a)$$

$$\text{div } \mathbf{B} = 0 \qquad \mathbf{B}^- = \mu^- \mathbf{H}^- \qquad (4.1b)$$

$$\mathbf{n} \times (\mathbf{H}^+ - \mathbf{H}^-) = 0 \qquad (4.1c)$$

$$\mathbf{n} \cdot (\mathbf{B}^+ - \mathbf{B}^-) = 0 \qquad (4.1d)$$

where the current density \mathbf{j} is the source of the magnetic field strength \mathbf{H} which is related to the magnetic flux \mathbf{B} via the permeability μ . The normal vector at the boundary \mathbf{n} allows to distinguish jump conditions for the normal as well as for the parallel component of the magnetic field.

4.2 Scalar Potential Formulations

In order to reduce the number of degrees of freedom one can introduce scalar potentials. In the following subsections some of the most important scalar potential formulations for the magnetostatic Maxwell problem are listed. A very good overview which also contains the eddy current case can be found in [45].

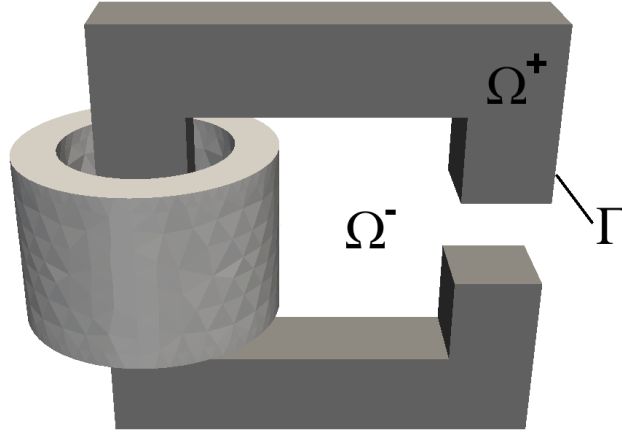


Figure 4.1: Example geometry for a yoke containing a magnetic region Ω^+ and the surrounding area Ω^- . In this example the external field is created by a coil located in Ω^- .

4.2.1 Reduced Scalar Potential u

The easiest way to define a scalar potential is to split the total magnetic field into an external part \mathbf{H}_{ext} (which is created by currents outside of Ω^+) and the curl-free induced magnetic field which can be expressed as the gradient of a scalar potential u . Inserting this definition of the magnetic field into the Eqn. (4.1a) - (4.1d) leads to the following system of equations for the scalar potential u . We introduce the normal derivative of the potential at the surface of the magnetic parts as $\phi = \frac{\partial u}{\partial \mathbf{n}}$:

$$\mathbf{H} = \mathbf{H}_{\text{ext}} - \nabla u \quad (4.2a)$$

$$\nabla \cdot (\mu^+ \nabla u^+) = \nabla \cdot (\mu^+ \mathbf{H}_{\text{ext}}) \quad (4.2b)$$

$$\nabla^2 u^- = 0 \quad (4.2c)$$

$$u^+ - u^- = 0 \quad (4.2d)$$

$$\mu^+ \phi^+ - \mu^- \phi^- = (\mu^+ - \mu^-) \mathbf{n} \cdot \mathbf{H}_{\text{ext}} \quad (4.2e)$$

This so-called reduced scalar potential always exists and is continuous. One problem of this formulation occurs for materials with high permeability, where

normally the external field and the induced demagnetization field nearly cancel each other, so that the remaining total field is very small. Finite precision arithmetic leads to cancellation errors which can cause instabilities in the solving method.

4.2.2 Total Scalar Potential \tilde{u} [1]

If there are no currents within Ω^+ and Ω^+ is simply connected, an alternative formulation of the scalar potential can be used. In this case, the internal field H^+ is curl-free and can therefore be directly expressed as the gradient of a total scalar potential, which leads to some slightly modified equations:

$$\mathbf{H}^+ = -\nabla\tilde{u}^+ \quad \mathbf{H}^- = \mathbf{H}_{\text{ext}} - \nabla u^- \quad (4.3a)$$

$$\nabla \cdot (\mu^+ \nabla \tilde{u}^+) = 0 \quad (4.3b)$$

$$\nabla^2 u^- = 0 \quad (4.3c)$$

$$\tilde{u}^+ - u^- = u_{\text{ext}} \quad (4.3d)$$

$$\mu^+ \tilde{\phi}^+ - \mu^- \phi^- = -\mu^- \mathbf{n} \cdot \mathbf{H}_{\text{ext}} \quad (4.3e)$$

where u_{ext} is the potential related to the external field H_{ext} . Since the total magnetic field can be calculated directly, numerical errors due to the subtraction of the external field are avoided. The drawback of this method is that due to the different descriptions in Ω^+ and Ω^- the potential is discontinuous at the surface of the magnet and additional complications occur if the region Ω^+ is not simply connected.

4.2.3 Mayergoyz Potential \hat{u} [2]

The method proposed by Mayergoyz is intended to combine the advantages of both aforementioned methods, namely it is continuous and defined in the whole space for arbitrary geometries, and no cancellation effect does occur. The method is split into two consecutive steps.

In the first step a linear material with (nearly) infinite permeability is assumed, and an ordinary reduced scalar potential formulation is used to calculate field \mathbf{H}_0 created by the material. Due to the use to the linear material the right-hand-side of (4.2b) becomes equal to 0 and the left-hand-side becomes an ordinary Laplace operator. Since permeability is infinite (or very large for practical implementations) one gets $\mathbf{H}_0^+ = 0$ inside of the magnetic material.

$$\mathbf{H}_0 = \mathbf{H}_{\text{ext}} - \nabla u_0 \quad (4.4a)$$

$$\nabla^2 u_0^+ = 0 \quad (4.4b)$$

$$\nabla^2 u_0^- = 0 \quad (4.4c)$$

$$u_0^+ - u_0^- = 0 \quad (4.4d)$$

$$\mu^+ \phi_0^+ - \mu^- \phi_0^- = (\mu^+ - \mu^-) \mathbf{n} \cdot \mathbf{H}_{\text{ext}} \quad (4.4e)$$

Once this linear response has been calculated a difference field $\mathbf{h} = \mathbf{H} - \mathbf{H}_0$ can be introduced, which only describes the difference from the linear behavior. Inside of the magnetic material this field is equal to the total magnetic field. Additionally it is defined in the whole space and since it is curl-free it can be expressed as the gradient of a potential \hat{u} . The defining equations of the potential field can be summarized as follows:

$$\mathbf{h} = -\nabla \hat{u} \quad \mathbf{H} = \mathbf{h} + \mathbf{H}_0 \quad (4.5a)$$

$$\nabla \cdot (\mu^+ \nabla \hat{u}^+) = 0 \quad (4.5b)$$

$$\nabla^2 \hat{u}^- = 0 \quad (4.5c)$$

$$\hat{u}^+ - \hat{u}^- = 0 \quad (4.5d)$$

$$\mu^+ \hat{\phi}^+ - \mu^- \hat{\phi}^- = -\mu^- \mathbf{n} \cdot \mathbf{H}_0^- \quad (4.5e)$$

Despite all of the positive properties of this algorithm the fact that it is necessary to solve two systems of equations consecutively, complicates the practical use of this algorithm, although the systems are very similar (the only difference

are the used permeabilities). Due to this reason and due to the longer solving time the use of the total scalar potential is preferred for most of the simulations in the following sections. Only for non-simply-connected models with moderate susceptibilities the reduced scalar potential is used.

4.3 FEM-BEM coupling

For the solution of the scalar potential formulations as given by Eqn. (4.2), (4.3), (4.4), or (4.5) some of the open boundary methods listed in 3.3 can be used. In contrast to the strayfield calculation one now has to deal with more complex jump conditions at the medium boundary, which prevents the use of the hybrid Fredkin-Koehler FEM-BEM coupling, which is the standard method for the strayfield calculation in the used FEMME software. This section therefore focuses on the direct Johnson-Nédélec FEM-BEM coupling, which offers a very general and flexible way to handle arbitrary jump conditions.

4.3.1 Linear Reduced Scalar Potential

For sake of simplicity only linear systems using a reduced scalar potential are considered at first. Since μ is constant the divergence operator on the right-hand side of the internal Poisson Eqn. (4.2b) only acts on \mathbf{H}_{ext} , which gives 0.

Starting with the FEM equations for the internal problem, we use a Galerkin approach with polynomial test and shape function $\Lambda_i(\mathbf{x})$ for each node i of the problem. Transforming Eqn. (4.2b) into the weak formulation, performing an integration by parts, and considering that for linear systems the source term on the right-hand side vanishes, leads to the following system of equations:

$$\int_{\Omega} \mu^+ \nabla \Lambda_i \cdot \nabla u^+ d\Omega - \int_{\Gamma} \mu^+ \Lambda_i \nabla u^+ \cdot \mathbf{n} d\Gamma = 0 \quad (4.6)$$

Using a FEM-only approach would require to define either Dirichlet or Neumann boundary conditions in order to get a solvable system. Since we do not know these boundary conditions, we introduce the normal derivative of the

potential at the boundary as new independent variables ϕ and add additional BEM equations to the system, which then has a unique solution.

For these additional equations, one recognizes that the outer potential u^- fulfills the Laplace equation. With the help of the fundamental solution $G = \frac{1}{4\pi} \frac{1}{|\mathbf{x}-\mathbf{y}|}$, u^- can be expressed by the third Green's identity in terms of its Cauchy data (the potential and its normal derivative) on the boundary Γ :

$$\frac{1}{2}u^- = \int_{\Gamma} (u^- \nabla G - \nabla u^- G) \mathbf{n} d\Gamma \quad (4.7)$$

Mathematically, the coupling method resulting from this approach, was first analyzed by Johnson and Nédélec [43] (the $\frac{1}{2}$ factor is only valid for smooth surfaces and has to be modified at edges [46]). Convergence and stability could be proven for linear problems [47] as well as for non-linear ones [48]. The terms on the right-hand side are called double-layer and single-layer potential. For a general Galerkin-BEM, formula (4.7) is multiplied by a test function $\Psi_m(\mathbf{x})$ and integrated over the boundary. In our implementation, we use delta distribution as test functions for the BEM equations $\Psi_m(\mathbf{x}) = \delta(\mathbf{x} - \mathbf{x}_m)$, which leads to a point-matching approach that only evaluates the equations at the centers of each boundary triangle. The main advantage of this approach is that the computation of the occurring boundary integrals is easier and less time-consuming.

Finally, we discretize our formulas by using linear shape functions Λ_j for the potential u and piecewise-constant shape functions $\mathbb{1}_n$ for its normal derivative ϕ . Furthermore, u^- and ϕ^+ can be eliminated by use of the jump conditions (4.2d) and (4.2e). Our discretized variables as well as the test function used within the Galerkin formalism look like the following:

$$u^+(\mathbf{x}) = \sum_j u_j \Lambda_j(\mathbf{x}) \quad \phi^-(\mathbf{x}) = \sum_n \phi_n \mathbb{1}_n(\mathbf{x}) \quad (4.8)$$

Combining FEM and BEM equations to one total system of equations we end up with a quadratic system M which fully defines our problem:

$$\begin{pmatrix} M_{ij}^{11} & M_{in}^{12} \\ M_{mj}^{21} & M_{mn}^{22} \end{pmatrix} \begin{pmatrix} u_j \\ \phi_n \end{pmatrix} = \begin{pmatrix} RHS_i^1 \\ \mathbf{0} \end{pmatrix} \quad (4.9a)$$

The individual elements of the matrices can be calculated as follows. Remember that for the FEM part these matrices are sparse whereas the BEM formalism leads to dense matrices:

$$M_{ij}^{11} = \int_{\Omega} \mu^+ \nabla \Lambda_i \cdot \nabla \Lambda_j d\Omega \quad (4.9b)$$

$$M_{in}^{12} = - \int_{\Gamma} \mu^+ \Lambda_i \mathbb{1}_n d\Gamma \quad (4.9c)$$

$$M_{mj}^{21} = \frac{1}{2} \Lambda_j(\mathbf{x}_m) - \frac{1}{4\pi} \int_{\Gamma} \Lambda_j(\mathbf{y}) \frac{\mathbf{x}_m - \mathbf{y}}{|\mathbf{x}_m - \mathbf{y}|^3} \mathbf{n} d\Gamma_y \quad (4.9d)$$

$$M_{mn}^{22} = \frac{1}{4\pi} \int_{\Gamma} \frac{\mathbb{1}_n(\mathbf{y})}{|\mathbf{x}_m - \mathbf{y}|} d\Gamma_y \quad (4.9e)$$

$$RHS_i^1 = \int_{\Gamma} (\mu^+ - \mu^-) \mathbf{n} \cdot \mathbf{H}^{\text{ext}} \Lambda_i d\Gamma \quad (4.9f)$$

The indices i, j run from 1 to the number of nodes of the geometry, whereas n, m run from 1 to the number of boundary elements. Since only boundary nodes contribute to the boundary integrals in M^{12} and M^{21} all matrix elements which correspond to nodes within the volume are equal to 0.

4.3.2 Linear Total Scalar Potential

The same calculations as for the reduced scalar potential can now be done to the total scalar potential. Since the right-hand side of the internal Poisson equation is always 0 for the total scalar potential, nothing changes there. The only difference is the jump condition. The jump of the normal derivative (4.3e) simply has another pre-factor which can easily be accounted for, but the jump condition of the potential itself (4.3d) is more problematic. Inserting this condition into (4.7) in order to replace u^- leads to an additional integral term on the right-hand side.

$$\begin{pmatrix} M_{ij}^{11} & M_{in}^{12} \\ M_{mj}^{21} & M_{mn}^{22} \end{pmatrix} \begin{pmatrix} u_j \\ \phi_n \end{pmatrix} = \begin{pmatrix} RHS_i^1 \\ RHS_m^2 \end{pmatrix} \quad (4.10a)$$

$$M_{ij}^{11} = \int_{\Omega} \mu^+ \nabla \Lambda_i \cdot \nabla \Lambda_j d\Omega \quad (4.10b)$$

$$M_{in}^{12} = - \int_{\Gamma} \mu^+ \Lambda_i \mathbb{1}_n d\Gamma \quad (4.10c)$$

$$M_{mj}^{21} = \frac{1}{2} \Lambda_j(\mathbf{x}_m) - \frac{1}{4\pi} \int_{\Gamma} \Lambda_j(\mathbf{y}) \frac{\mathbf{x}_m - \mathbf{y}}{|\mathbf{x}_m - \mathbf{y}|^3} \mathbf{n} d\Gamma_y \quad (4.10d)$$

$$M_{mn}^{22} = \frac{1}{4\pi} \int_{\Gamma} \frac{\mathbb{1}_n(\mathbf{y})}{|\mathbf{x}_m - \mathbf{y}|} d\Gamma_y \quad (4.10e)$$

$$RHS_i^1 = \int_{\Gamma} -\mu^- \mathbf{n} \cdot \mathbf{H}^{\text{ext}} \Lambda_i d\Gamma \quad (4.10f)$$

$$RHS_m^2 = M_{mj}^{21} \cdot u_j^{\text{ext}} \quad (4.10g)$$

4.3.3 Nonlinear Reduced Scalar Potential

If the material property μ depends on the location, as it is true for inhomogeneous or nonlinear materials, the right-hand side of the internal Poisson equations does not vanish. Partial integration of this term gives a surface term which is proportional to the right-hand side of the linear equations as well as an additional volume integral. It is remarkable that if the two surface terms are combined, there occurs the same pre-factor as for the total scalar potential:

$$\begin{aligned} RHS_i^1 &= \int_{\Gamma} (\mu^+ - \mu^-) \mathbf{n} \cdot \mathbf{H}^{\text{ext}} \Lambda_i d\Gamma - \int_{\Omega} \nabla \cdot (\mu^+ \mathbf{H}^{\text{ext}}) \Lambda_i d\Omega = \\ &= \int_{\Gamma} -\mu^- \mathbf{n} \cdot \mathbf{H}^{\text{ext}} \Lambda_i d\Gamma + \int_{\Omega} \mu^+ \mathbf{H}^{\text{ext}} \cdot \nabla \Lambda_i d\Omega \end{aligned} \quad (4.11)$$

The additional volume term containing the external field, complicates the handling of external sources, like coils, permanent magnets and so on, since their strayfield needs to be known everywhere within the volume of the Maxwell model and not only on its surface. Fortunately this problem can be tackled by considering the solution of an additional Laplace equation, which allows to derive the behavior of the external field inside of the Maxwell model from its value on the boundary. By this way all interactions only need to be evaluated at

the boundary, which reduces memory consumption as well as the computation time of these interactions. The field-values calculated at the boundary are then taken as Neumann boundary conditions of a Laplace equation for the potential within the Maxwell model. From the calculated potential one can then derive the interaction field at each location inside of the model. A detailed description of how interactions are implemented can be found in Section 5.3.

4.4 Calculation of BEM matrices

The occurring BEM matrices are calculated analytically as given in Ref. [49]. An alternative but more special solution was given by Lindholm [46]. Furthermore a fully discrete integration could be used [50]. The single and double-layer potential (E and D) at the point \mathbf{x} created by a triangle Δ using the Green's function of the 3 dimensional Laplacian are given by

$$E(\mathbf{x}) = \frac{1}{4\pi} \int_{\Delta} \frac{f(\mathbf{y})}{|\mathbf{x} - \mathbf{y}|} d\Gamma_y \quad (4.12)$$

$$D(\mathbf{x}) = \frac{1}{4\pi} \int_{\Delta} f(\mathbf{y}) \frac{\mathbf{x} - \mathbf{y}}{|\mathbf{x} - \mathbf{y}|^3} \mathbf{n} d\Gamma_y \quad (4.13)$$

where $f(\mathbf{y})$ stands for the shape function of the used elements. For our implementation we confine ourselves to piecewise constant shape function for the single-layer potential and linear shape functions for the double-layer potential. Since these integrals are translation- and rotation-invariant, we use a transformation Φ to rotate the triangle into the x_1 - x_2 -plane and to move the reference point \mathbf{x} onto the x_3 axis (see Fig. 4.2). Doing this, we can write a general polynomial shape function as $f(\mathbf{y}) = \sum_{jk} f_{jk} y_1^j y_2^k$ and we can break the problem down to the calculation of the integrals for the individual monomials that occur

within the sum:

$$E_{jk}(\mathbf{x}) = \frac{1}{4\pi} \int_{\Delta} \frac{y_1^j y_2^k}{\sqrt{y_1^2 + y_2^2 + x_3^2}} dy_1 dy_2 \quad (4.14)$$

$$D_{jk}(\mathbf{x}) = \frac{x_3}{4\pi} \int_{\Delta} \frac{y_1^j y_2^k}{\sqrt{y_1^2 + y_2^2 + x_3^2}^3} dy_1 dy_2 \quad (4.15)$$

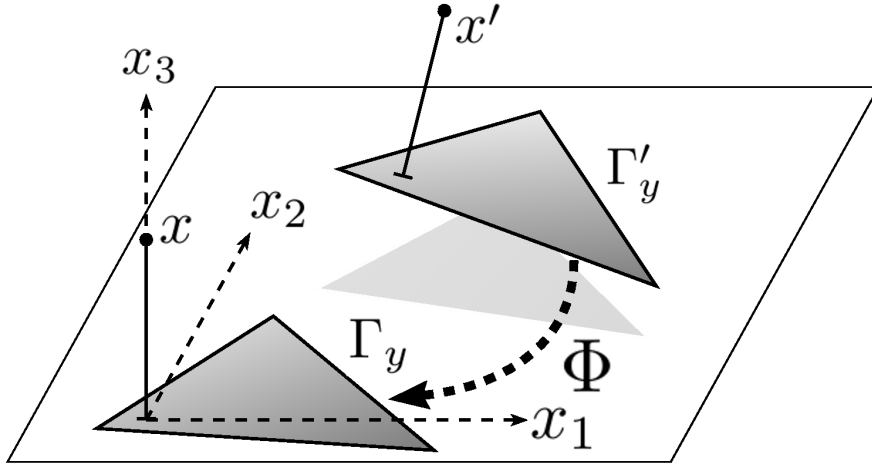


Figure 4.2: Diagram schematizing the triangle area used for the integration process.

The divergence theorem $\int_{\Delta} \text{div } \mathbf{G} d\Gamma = \int_{\gamma} \mathbf{G} \cdot \mathbf{n} d\gamma$ can be used to further simplify the calculation by transforming the surface integrals to line integrals [51]. One can finally derive a recursive formula for the higher order surface integrals that only depends on lowest order line integrals which can be solved analytically as well as on the lowest order double-layer term D_{00} . Furthermore D_{00} can be expressed as the surface angle corresponding to \mathbf{x} and Δ which allows a simple geometric interpretation of these terms. For the construction of the surface angle one connects \mathbf{x} with the three corners of the triangle. The intersection points of these connection lines with the unit sphere define the corners of a spherical triangle on the unit sphere. The area of this triangle finally gives us the value of D_{00} (see Fig. 4.3).

Using the recursive formula [51] allows to calculate the occurring integrals for shape functions of any order. For sake of clarity we now demonstrate how

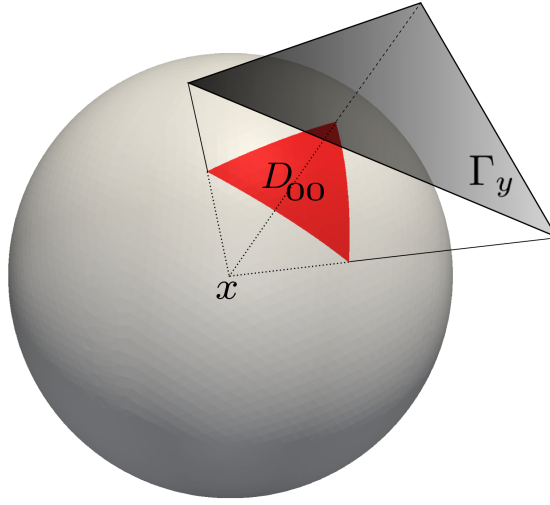


Figure 4.3: D_{00} can be interpreted as the surface of the triangle on the unit sphere.

this recursion is derived for the lowest order single-layer potential that is used in our algorithm. If we set $\mathbf{G} = \left(\frac{y_1}{r}, \frac{y_2}{r}\right)$ with $\text{div } \mathbf{G} = \frac{1}{r} + \frac{x_3^2}{r^3}$ and $r = \sqrt{y_1^2 + y_2^2 + x_3^2}$ the divergence theorem gives:

$$\underbrace{\int_{\Delta} \frac{1}{r}}_{4\pi E_{00}} + \underbrace{x_3^2 \int_{\Delta} \frac{1}{r^3}}_{4\pi x_3 D_{00}} = \int_{\gamma} \left(\frac{y_1}{r}, \frac{y_2}{r}\right) \mathbf{n} d\gamma \quad (4.16)$$

We can now calculate the interaction matrix elements of the piecewise constant single-layer shape functions E_{00} , by means of the line integral on the right-hand side of equation (4.16) as well as of the surface angle D_{00} .

The main numerical problem when dealing with BEM matrices is that they are dense, and thus extensive computation time and storage capacity is needed to handle them. This problem can be overcome by use of \mathcal{H} -matrices (see Section 2.3), where some blocks of the dense matrices (\mathbf{M}^{21} , \mathbf{M}^{22}) are replaced by low rank approximations. If a block can be simplified or not depends on the admissibility criterion which tells whether the size of the corresponding pointsets is smaller than the distance between source and destination pointset. If the admissibility criterion is fulfilled, the corresponding block describes far-field interaction and thus can be approximated without introducing a significant er-

ror. On the other hand if source and destination pointsets are very close to each other, then the corresponding matrix is stored as a full matrix in order to obtain a sufficient accuracy.

We use the ACA+ algorithm which allows to obtain low-rank approximations without having to evaluate all elements of the original block [52]. Therefore the approximated system matrix of rank k is expressed as $\tilde{\mathbf{M}}^{n \times m} = \mathbf{U}^{n \times k} \mathbf{V}^{k \times m}$. \mathbf{U} and \mathbf{V} are created by sequentially extracting lines and columns of the original matrix $\mathbf{M}^{m \times n}$. Lines and columns of the matrix are chosen in a way that the largest elements of the error matrix $\mathbf{R} = \mathbf{M} - \tilde{\mathbf{M}}$ are eliminated.

4.5 Nonlinear Materials

Dealing with nonlinear materials does not only lead to additional terms in the FEM-BEM equations (as described in Sec. 4.3), but it also means that one has to perform a nonlinear iteration on top of the linearized FEM-BEM equations. Typically some sort of Newton iteration is applied. Within every nonlinear step of this method the linearized system matrix needs to be updated. In the case of the FEM-BEM coupling method, fortunately only the M_{ij}^{11} block depends on the solution vector. Since our implementation uses linear basis functions for the potential u this leads to a constant magnetic field inside of each element. Therefore updating the system matrix can be done by simply scaling the contribution of each element with the actual magnetic field inside of this element.

The current implementation assumes isotropic and soft magnetic materials. Therefore the following dependence holds

$$\mathbf{B} = \mu_0(\mathbf{H} + \mathbf{M}) = \mu_0(1 + \chi) \mathbf{H} \quad \chi = \frac{M}{H} = \frac{J}{\mu_0 H} \quad (4.17)$$

where M , J and H are the absolute values of the magnetization, the magnetic polarization or the magnetic field strength, respectively. Practically a material is specified by defining J and $\mu_0 H$ at certain sampling points and choose a proper interpolation or fitting method. The following methods have been implemented:

- **Linear Interpolation** If the specified H lies between two given sampling points H_1 and H_2 (with the corresponding magnetization M_1 and M_2) the interpolated M can simply be expressed as

$$M(H) = M_1 + \frac{M_2 - M_1}{H_2 - H_1} (H - H_1) \quad (4.18)$$

If the specified magnetic field lies outside of all the sampling points the proper magnetization value is extrapolated. The disadvantage of this problem is that jumps in the derivative of the interpolated curve may lead to numerical problems when solving nonlinear equations.

- **Cubic Spline Interpolation** During the setup phase a cubic polynomial is defined between each pair of sampling points. This is done in a way that for each interval the two sampling points are reproduced exactly and that the derivative of the curve does not jump from one interval to the other. By this way one ends up with a smooth curve which reproduces all sampling points exactly. Although this method provides smooth interpolations it may suffer from unnatural fluctuations between the sampling points and especially it is not possible to reproduce the horizontal asymptote, which nearly all materials show for high fields.
- **tanh Interpolation** The material law for many simple materials looks very similar to the \tanh . For testing reasons using this interpolation provides the advantage that it is strictly monotone and infinitely often differentiable. Therefore analytical expressions can be given even for the derivatives of the material law, which may help to avoid numerical errors. In order to make this interpolation easy to use, it is only necessary to define three sample points. The first two are used to determine the initial derivative of the \tanh for small values of H , whereas the last one is used to define the saturation magnetization M . If further sampling points are specified these will be ignored. In contrast to the former methods, no real interpolation is performed since the specified data points are not reconstructed. Instead of interpolation the method fits a \tanh -function to the data points

in a way that the initial derivative as well as the saturation magnetization can be reproduced.

A comparison of the implemented methods is shown in Fig. 4.4.

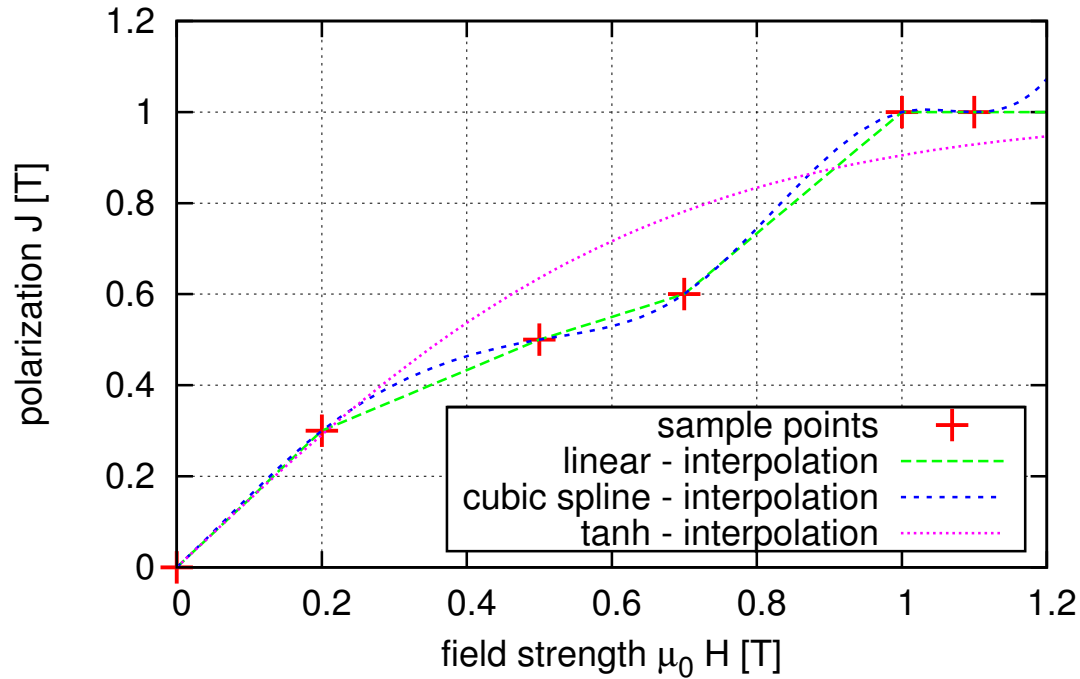


Figure 4.4: Comparison of the implemented interpolation / fitting methods for given sample points. The initial χ of the described material is equal to 1, and the material is saturated at $1T$. One can clearly see the bad behavior of the spline interpolation at the end of the interval, as well as the edges in the linear interpolation. The tanh interpolation does not show these problems but therefore it is not able to correctly reproduce the internal sample points.

4.6 Solution Algorithms

The solution of the coupled FEM-BEM system needs to use matrix-free algorithms, since FEM and BEM parts are stored in different formats. In this work we are using the SUNDIALS solver suite which provides matrix-free methods

for differential, algebraic and differential-algebraic equations. In the case of magnetostatic Maxwell equations the algebraic solver named KINSOL [53] is the proper choice. It allows to solve a nonlinear, quadratic system of equations of the form

$$\mathbf{F}(\mathbf{x}) = 0 \quad (4.19)$$

where \mathbf{x} are the unknowns of the system and \mathbf{F} are the nonlinear functions. KINSOL utilizes a Newton method, which is the default method to deal with nonlinear equations. This leads to the top-level iteration shown in Alg. (4.1), where \mathbf{J} is the system Jacobian and λ is an optional parameter, which can be determined by a Linesearch algorithm, which then leads to more robust algorithms. For the standard Newton method λ is constant and equal to 1. KINSOL provides several methods to solve these systems of linear equations, which differ in the information that is provided. If the system Jacobian is known, and can be stored in a certain matrix format a direct linear solver can be used. KINSOL includes solvers for dense and sparse matrices or it may also use some external solver routines as those provided by the BLAS/LAPACK library.

```

1 while not converged do
2   | Solve  $\mathbf{J}(\mathbf{x}_n) \delta_n = -\mathbf{F}(\mathbf{x}_n)$ 
3   | Set  $\mathbf{x}_{n+1} = \mathbf{x}_n + \lambda \delta_n, 0 < \lambda \leq 1$ 
4 end
```

Algorithm 4.1: Top-level Newton Iteration

Unfortunately storing the full Jacobian of the FEM-BEM system would result in too much memory consumption as well as computational costs in the general case. In order to handle large scale problems, one needs to choose iterative solvers, where not the full Jacobian is needed, but only the matrix-vector-product of the Jacobian with a given direction vector \mathbf{v} needs to be calculated. By default KINSOL uses a finite difference approximation $\mathbf{J}(\mathbf{x}) \cdot \mathbf{v} \approx (F(\mathbf{x} + \sigma \mathbf{v}) - F(\mathbf{x}))/\sigma$, with $\sigma \rightarrow 0$, but it is also possible to hand over a user-defined function, which calculates this matrix-vector product.

KINSOL provides the following iterative Krylov-subspace methods which are suited to iteratively solve the FEM-BEM system

- Scaled Preconditioned GMRES (Generalized Minimal Residual method)
- Scaled Preconditioned Bi-CGStab (Bi-Conjugate Gradient Stable method)
- Scaled Preconditioned TFQMR (Transpose-Free Quasi-Minimal Residual method)

Typically GMRES is the most stable method, but depending on the equation system, Bi-CGStab or TFQMR could offer a better performance. Since all of the three methods require the same input data, it is possible to easily switch between the methods and compare their results. Since these iterative solvers only approximately solve the linear system the resulting methods are called Inexact Newton Methods [11]. Pseudo-codes for all of these methods can be found e.g. in [54].

4.6.1 Globalization strategies

Since an ordinary Newton iteration provides only local convergence the following globalization strategies have been considered:

- Goldstein-Armijo Linesearch [55,56]: This method is provided by the KINSOL solver itself. As already mentioned it uses the direction derived by an ordinary Newton method, but the step length is adapted in a way that global convergence can be reached. In practice there exist two main cases which lead to a non-converging results. The first problem is an overshooting if the calculated Newton step is too large (as it is the case in the simple example of $|x|^a$, with $0 < a < 0.5$). A back-tracking algorithm can be used to avoid this problem. This is done by successive reduction of the used step length λ until the so-called α -condition

$$\|\mathbf{F}(\mathbf{x}_n + \lambda \delta_n)\| \leq \|\mathbf{F}(\mathbf{x}_n) + \alpha \nabla \mathbf{F}(\mathbf{x}_n) \lambda \delta_n\| \quad (4.20)$$

is satisfied. This α -condition makes sure that the step length is small enough that the exact function can still be well approximated by its linearization at \mathbf{x}_n . The parameter α can be set by the user and is typically

chosen between 0 and 0.5. The second problem which can influence convergence occurs if the calculated Newton step is too small and it would require a large number of iteration to obtain a certain accuracy. In order to prevent this problem a second condition is checked as soon as the α -condition is fulfilled. This so-called β -condition

$$\|\nabla \mathbf{F}(\mathbf{x}_n + \lambda \delta_n)\| \leq \beta \|\nabla \mathbf{F}(\mathbf{x}_n)\| \quad (4.21a)$$

or as it is implemented within KINSOL

$$\|\mathbf{F}(\mathbf{x}_n + \lambda \delta_n)\| \geq \|\mathbf{F}(\mathbf{x}_n) + \beta \nabla \mathbf{F}(\mathbf{x}_n) \lambda \delta_n\| \quad (\text{test})$$

makes sure that the chosen step length is not too small. In practice the use of a back-tracking method also avoids excessively small steps, and the β -condition is only checked as an additional cancellation condition. Again this parameter can be set by the user and typical values are between 0.5 and 1.

- **Homotopy Method:** This method is implemented on top of KINSOL and it is based on finding a good initial guess for the problem to solve. This can be reached by starting from a similar, but easier problem with a known solution. Then this easy problem is continuously transformed into the original problem and within each step of this transformation a Newton system is solved. By this way one can assure that the solution of the last step provides a good initial guess for the next step and one finally ends up with a solution of the original problem.

In the case of the magnetostatic Maxwell equations such a homotopy can be created by continuously turn the applied current up from 0 to its final value. In this case the initial solution is equal to 0. The first steps will result in a linear response since the applied field is too small to saturate the magnetic material. Finally the material will get more and more saturated until the final result is reached. In the current implementation a fixed step size of the homotopy parameter λ is assumed. Of course this leads to a relatively large number of steps where the material behaves nearly linear.

This drawback can be overcome with some sort of step size control for the λ parameter.

Alternatively one could choose a different kind of homotopy. For example one could start from a linear material behavior and transform this continuously into the correct non-linear material law.

4.7 Preconditioning

For linear system one can define a condition number of the system matrix M which is directly related to the convergence speed of the iterative method and can be expressed as $\text{cond}(M) = \|M\| \|M^{-1}\|$ (with a proper matrix norm). We empirically found that the condition number is directly proportional to the susceptibility of the simulated material for various models (see Fig. 4.7). Therefore, especially for systems with high permeability, computation time can be decreased significantly by use of a proper preconditioner.

In our current implementation, we use the preconditioning feature of the solver suite, that does a right-preconditioning, where one rewrites the original matrix equation as $(M P^{-1})(P x) = \text{RHS}$, with a preconditioner matrix P and the unknowns $x = (u_j, \phi_n)$. This leads to a new system matrix $M P^{-1}$ for the iterative solver with a smaller condition number.

The preconditioner matrix P is a good approximation of the system matrix M , but it is easier to solve. The most time consuming part of solving the system of equations is due to the dense boundary matrices. For BEM matrices it can be proven that simple diagonal scaling reasonably improves conditioning [57], if the used element sizes vary strongly.

For the linear coupled system (4.9) numerical studies showed that using only the diagonal elements of the single-layer matrix as well as only the elements contributing to the $\frac{1}{2}$ -factor within the double-layer matrix leads to a sparse approximation of the system matrix M which can be used as an efficient pre-

conditioner:

$$\mathbf{P} = \begin{pmatrix} M_{ij}^{11} & M_{in}^{12} \\ \frac{1}{2}\Lambda_j(\mathbf{x}_m) & \text{diag}(M_{mn}^{22}) \end{pmatrix} \quad (4.22)$$

Omitting parts of the BEM matrices which correspond with a long range interaction results in a good approximation of the original system matrix, but now the whole preconditioner matrix is sparse which makes it much more efficient to evaluate. In the nonlinear case where the system matrix depends on the actual solution an additional sparse term arises from the FEM formulation. Since it depends on the derivative of the susceptibility χ' , it complicates the definition of the nonlinear material. A comparison of both methods in practice shows nearly no benefit of nonlinear version in comparison to the linear one, therefore the default method is using the linear preconditioning matrix for linear as well as nonlinear problems. For all our test geometries, computation times could be decreased dramatically, and additionally it does no longer depend on the used susceptibility. Results are presented in the following section.

4.8 Test Cases

In this section, the algorithm is applied to calculate the induced magnetization for certain significant test geometries. The purpose of these test cases is to verify the solutions of the solver. Three different approaches are followed when choosing the test cases:

- **Physical Analytical Solutions:** If there exists certain physical analytical solutions for a problem, this is the best way to verify a numerical algorithm. By this way it is finally possible to specify an absolute error using a desired norm. If the numerical algorithm consists of multiple consecutive calculations, it is also possible to check each of the intermediate results. Unfortunately for the magnetostatic Maxwell equations there exists very few exact analytical solutions. In fact the magnetic sphere in a homogeneous external field is the only available exact analytical solution which

can be verified with this method. Nevertheless it is also possible to utilize approximated analytical results like the magnetic field in the air-gap of a joke.

- **Artificial Analytical Solutions:** A very similar approach is based on mathematical solutions of the equation system in consideration. In contrast to physical analytical solutions, where the jump conditions and the internal solution needs to be consistent with some external field sources, artificial analytical solutions can be chosen arbitrarily and the corresponding jump conditions can then be derived. The other solution u^- has to fulfill the Laplace equation and needs to be $O(1/r)$ for $r \rightarrow \infty$, whereas the internal solution u^+ needs to fulfill the internal Poisson equation, which also reduces to a Laplace equation in the linear case. After the solutions have been defined the jump conditions for the potential as well as for its derivative can be calculated. Finally one can solve the equation system with these artificial jump conditions and gets a numerical solution of the problem, which can be compared easily with the artificial analytical solution. The advantage of this method is that it is applicable to general geometries and it provides the absolute error of the calculated solution.
- **Comparison:** Alternatively it is also always possible to compare simulation results among different numerical algorithms or with an experiment. Since such experiments are very time consuming and expensive to implement and using various numerical simulation tools also requires a lot of preparation, a group of scientists published a number of test cases called TEAM Workshop problems. These problems are well defined and realized as experiment and up to now various groups tested their algorithms against them. Therefore it is perfectly suited for the verification of the FEM-BEM algorithm.

The following subsections present a subset of the most significant tests performed in order to validate the FEM-BEM method to solve magnetostatic Maxwell equations.

4.8.1 Magnetic sphere in homogeneous external field

Since the solution for the magnetic sphere in an external field can be calculated analytically, it is well suited to check the accuracy of our algorithm. The analytical calculation of the homogeneously magnetized sphere (with a magnetization M) shows that the induced field H_d is also homogeneous within the sphere and its value is $-\frac{1}{3}M$. Adding the material law $M = \chi H$, where H is the total field $H = H_d + H_{ext}$ and χ is the magnetic susceptibility of the material, directly leads to the analytical result:

$$M = \frac{3\chi}{3 + \chi} H_{ext} \quad (4.23)$$

The comparison of the numerical results with Eqn. (4.23) shows that the mean magnetization (averaged over the volume of the sphere) differs by around 0.1% using 10535 elements. In order to demonstrate the performance of the presented algorithm we measured runtime and maximal memory consumption for different number of elements (see Table 4.1). For high susceptibilities the problem becomes more difficult to solve due to the large condition number. For practical applications the largest relevant susceptibility is around $\chi = 10^5$, which we therefore use for our studies. The calculated magnetization for a sphere with 1074 elements is shown in Fig. 4.5.

No. elements	1074	10535	337505	1546384
runtime [s]	1.470	26.45	1590.3	25324.1
memory [MB]	194	248	1499	5503

Table 4.1: Performance values of the sphere example for different grid sizes.

4.8.2 Field within air-gap of a magnetic yoke

This more complex example shows a cylindrical coil around a magnetic yoke. The external field H_{ext} is created by the coil and is directly calculated by the

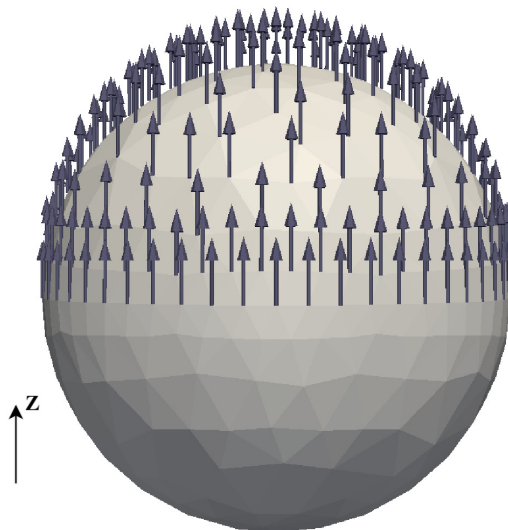


Figure 4.5: Simulation results for the magnetization of the magnetic sphere in a homogeneous external field. The Magnetization is nearly perfectly aligned to the external field which is applied in z direction.

Biot-Savart law:

$$\mathbf{H}(\mathbf{x}) = \frac{1}{4\pi} \int_{\Omega_{coil}} \frac{\mathbf{j} \times (\mathbf{x} - \mathbf{x}')}{|\mathbf{x} - \mathbf{x}'|^3} d^3\mathbf{x}' \quad (4.24)$$

We again measured solver runtime as well as maximum memory consumption for different discretizations (see Table 4.2). The calculated magnetization for $\chi = 10^6$ can be seen in Fig. 4.6.

The calculated condition numbers as well as the computation time of the equation solver with and without preconditioning is illustrated for the yoke example in Fig. 4.7.

No. elements	1077	10205	109384	1180305
runtime [s]	5.027	28.33	734.3	14035.4
memory [MB]	218	374	1025	5782

Table 4.2: Performance values of the yoke example for different grid sizes.

Unfortunately for this model there exist no analytical results. Therefore one needs to introduce a simplified model in order to deduce some analytical pre-

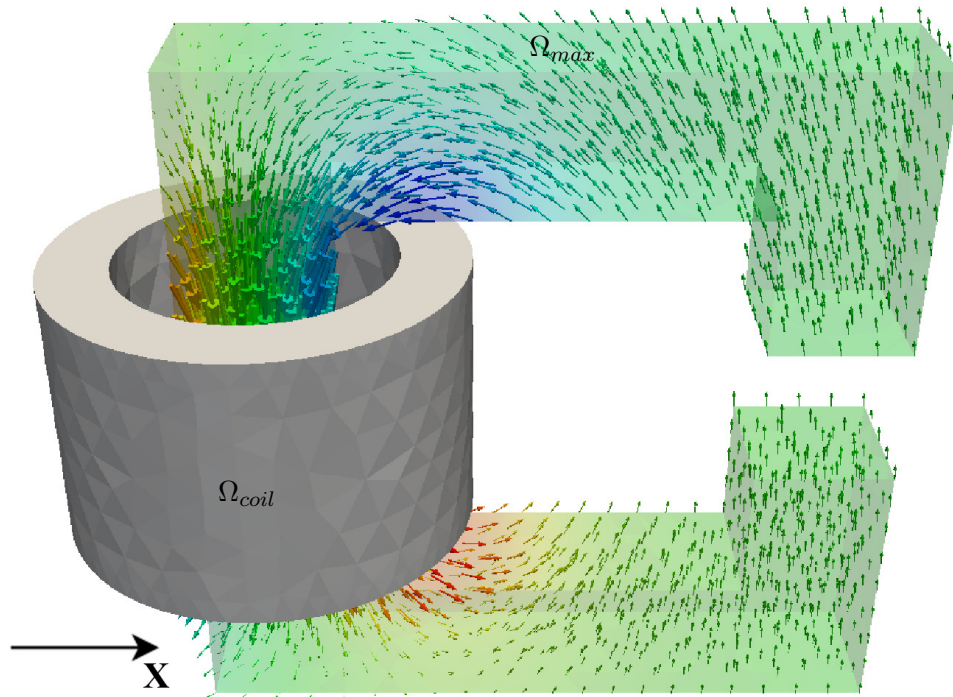


Figure 4.6: Simulation results for a magnetic yoke magnetized by an electric coil. 10205 elements were used for the discretization of the yoke. The color of the arrows shows the x component of the magnetization.

dictions, which the simulations could be compared with. In the case of the yoke with an air-gap one can assume that the whole magnetic flux is guided through the magnetic parts. Since the magnetic flux needs to be conserved within the whole circuit, assuming a constant cross section A leads to a constant B everywhere within the yoke and also inside of the air-gap. Due to the different permeabilities this leads to different magnetic fields \mathbf{H} inside yoke and air-gap. Writing down Ampere's law along the average path of the magnetic flux and considering that the permeability within the yoke $\mu_y \gg \mu_0$ therefore simplifies to:

$$\int_{\gamma} \mathbf{H} d\mathbf{s} = H_y l_y + H_a a = \frac{B A}{\mu_y} l_y + \frac{B A}{\mu_0} a \approx \frac{B A}{\mu_0} a = \int_{\Gamma} \mathbf{j} d\Gamma = I \quad (4.25)$$

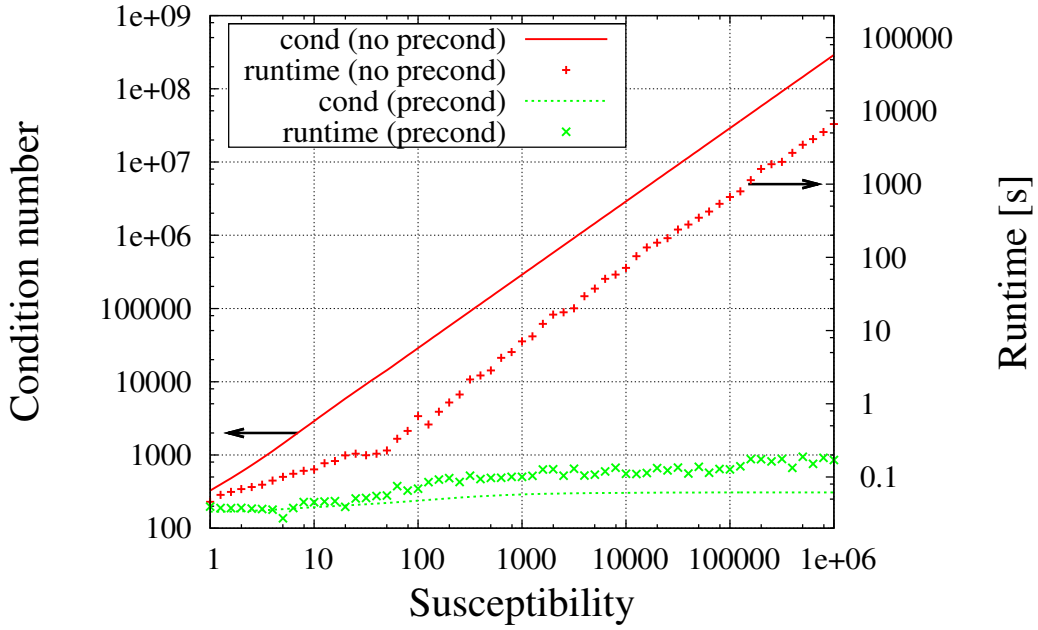


Figure 4.7: Condition number and solver runtime of the Yoke example (description see section 4.8.2) with and without preconditioner. The condition number gives a good estimation of the solver runtime. One can see that the performance is improved significantly, especially for high susceptibilities. This behavior could also be confirmed for different models.

$$\Rightarrow H_a = \frac{I}{a} \quad (4.26)$$

One now deduced the absolute value of the magnetic field and also its behavior for changing gap sizes for this very simplified model. Those values can now be easily compared with the simulation results as it is shown in Fig. 4.8.

4.8.3 Artificial analytical solutions for the stator of an electric motor

This example should demonstrate the creation of artificial analytical solutions for arbitrary geometries. One starts with selecting proper analytical solutions for the potential inside and outside of the magnetic region. Of course these potentials need to fulfill the internal or external Poisson equation respectively, and the external solution needs to decay faster than $1/r$ for $r \rightarrow \infty$. For this

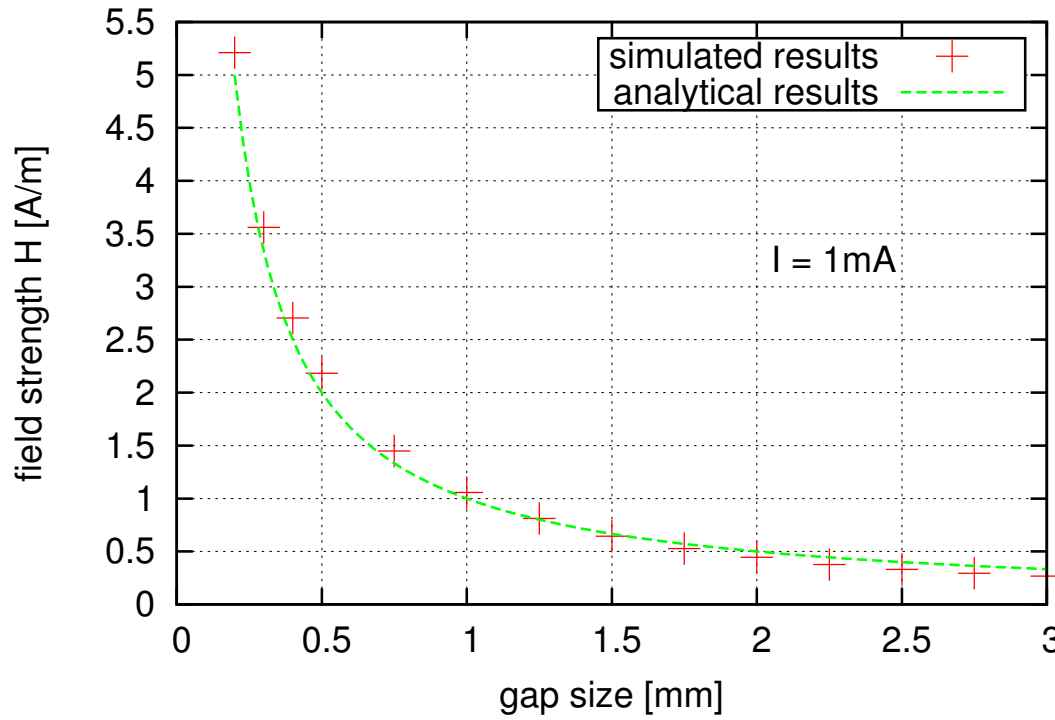


Figure 4.8: Comparison of simulation results of the averaged magnetic field within the air-gap for different gap sizes with an analytical approximation. For the simulation results the average of the magnetic field within cubic box located at the center of the air-gap is taken. The permeability of the yoke $\mu_y = 1000$, and a current of $I = 1\text{mA}$ is applied.

example we decided to use:

$$u_{analytic}^+ = z \qquad u_{analytic}^- = \frac{z}{r^3} \quad (4.27)$$

The next step is to derive the gradient of these potentials which in turn leads to the jump conditions of the normal derivative:

$$\nabla u_{analytic}^+ = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \qquad \nabla u_{analytic}^- = \frac{1}{r^3} \left[\begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} - \frac{3z}{r^2} \begin{pmatrix} x \\ y \\ z \end{pmatrix} \right] \quad (4.28)$$

From these analytical equations one can deduce the artificial jump conditions of the potential as well as of its normal derivative at the surface of the

magnetic region. Setting this in into Eqn. (4.3) one ends up with the following artificial right-hand sides for the discretized equations:

$$RHS_i^1 = M_{in}^{12} \cdot ((\nabla u_{analytic}^+ - \nabla u_{analytic}^-) \cdot \mathbf{n})_n \quad (4.29)$$

$$RHS_m^2 = M_{mj}^{21} \cdot (u_{analytic}^+ - u_{analytic}^-)_j \quad (4.30)$$

Using these right-hand sides the potential Eqn. (4.3) can be solved and the simulation results can be compared with the analytical formula. In Fig. 4.9 this has been done for the simulation of the stator of an electric motor. Additionally to the validation of the implemented algorithm the absolute error derived from this simulation allows to optimize solver parameters or to visualize problems with the discretization of the model.

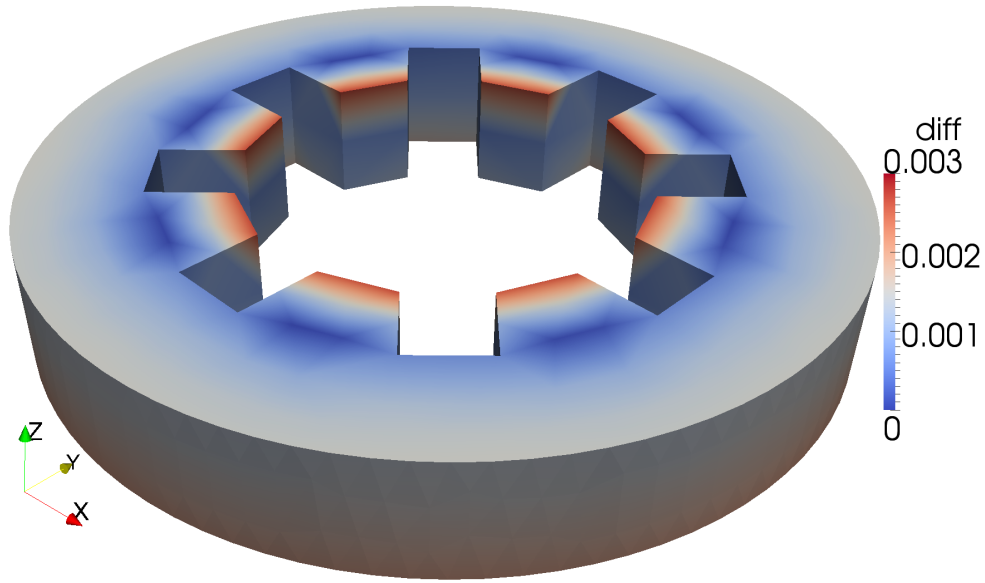


Figure 4.9: Absolute error of the (unoptimized) solution of the potential equation using an artificial right-hand side calculated from some analytical potentials. The potential varies from 0 to 1, which gives an relative error in the per mill range.

4.8.4 Team Workshop Problem 13

The last test case that is presented is based on a comparison of the results of a well defined problem against several other algorithms. Fortunately there exists the TEAM Workshop problem No. 13 [58], which deals with the solutions of magnetostatic Maxwell equations with nonlinear materials. This example is perfectly suited to demonstrate the performance of the implemented algorithm.

The problem consists of a magnetic source in form of a cylindrical coil, driven by a constant current. The magnetomotive force $\Phi = N I$, where I is the current through the coil and N is the number turns, is chosen to be $1000A$ turns. It is chosen in a way that only parts of the magnetic material around the coil become saturated. The magnetic part consists of several thin plates which are positioned around the coil. The result of the simulation should be the averaged magnetization in the cross section of the magnetic sheets at different locations.

The exact definition of the used geometries as well as of the material law and an instruction how the measurements have been realized in experiment can be found in [59]. For the solution of the problem a total scalar potential (4.3) formulation is used. Since the geometry is simply connected and one tries to avoid cancellation errors this is the preferred method. The geometries and the resulting magnetization for the whole model is visualized in Fig. 4.10, whereas Fig. 4.11 shows the averaged magnetization within the cross section of the horizontal plates for certain positions x . It can be seen that from all considered algorithms the FEM-BEM method is able to reproduce the measured data most accurately.

4.9 Conclusion

We have introduced a FEM / BEM coupling method which combines the advantages of both methods. The algorithm is applied to a magnetostatic problem with an open boundary. The arising dense boundary matrices are approximated by \mathcal{H} -matrices in order to speed up computation and to decrease memory consumption. Furthermore an effective preconditioner has been proposed in order

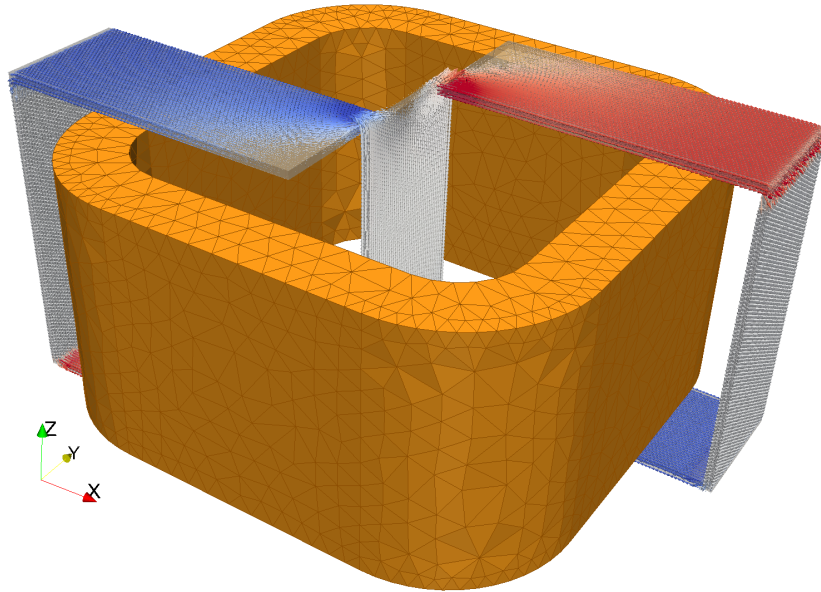


Figure 4.10: TEAM Workshop problem No. 13 geometry as well as the resulting magnetization within the magnetic sheets are shown. The magnetic field is created by a cylindrical coil (yellow), which creates an magnetomotive force of $1000A$ turns. Additional to the magnetization vectors, the x -component of the magnetization is visualized by color.

to improve the convergence of the iterative solver. Finally the performance of the algorithm is demonstrated by several test cases.

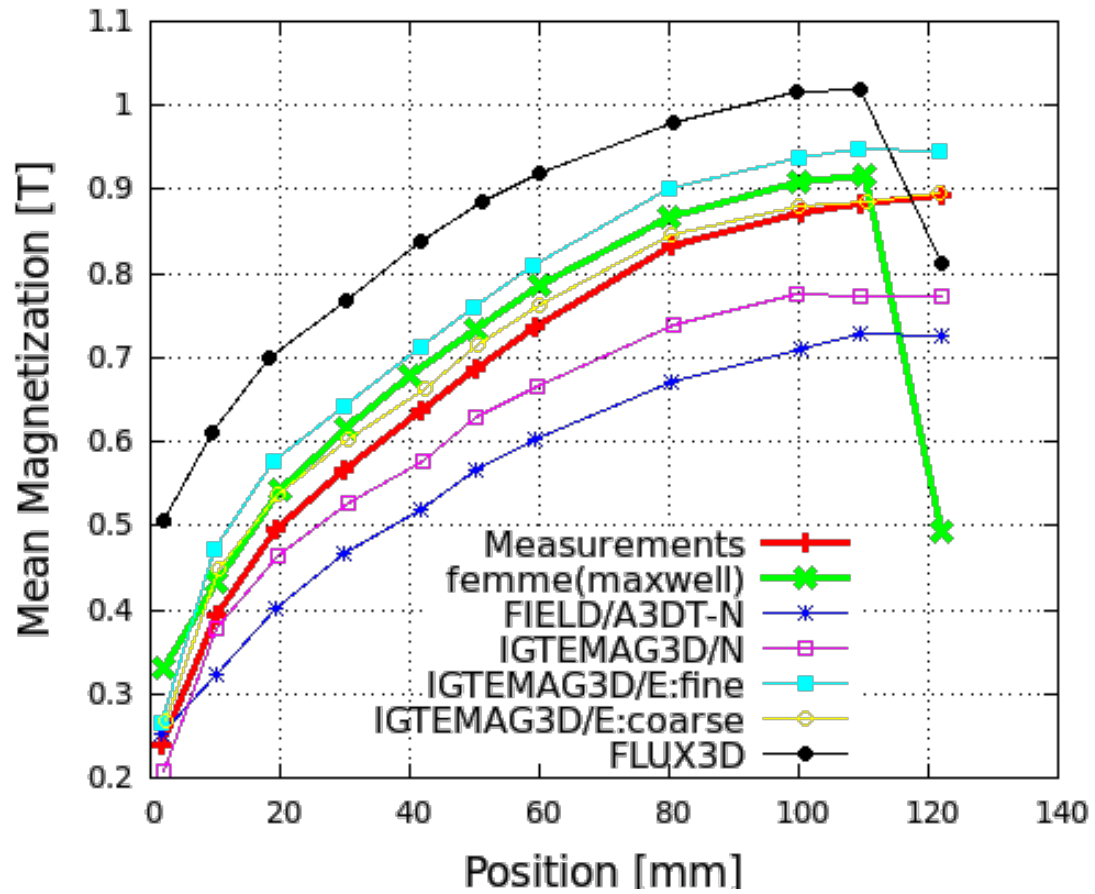


Figure 4.11: Comparison of different algorithms by means of the TEAM Workshop problem No. 13. The averaged magnetization within the cross section of the horizontal magnetic sheet is plotted for different positions x . Results of the FEM-BEM method are labeled `femme(maxwell)`. All other results are taken from the benchmark summary [59].

CHAPTER 5

Coupling

Portions of this chapter were previously published as [4] and have been reproduced with permission of the coauthors and in accordance with the publisher's policy. Content which was not generated by the author of this thesis is explicitly denoted. Copyright is held by Elsevier.

5.1 Introduction

Micromagnetic simulations are utilized in a wide range of applications ranging from magnetic storage devices, permanent magnets to spintronic devices. With increasing complexity of the devices more properties have to be included in the simulations in order to predict the functional behavior of the structures accurately. State of the art micromagnetic simulations can handle systems with several millions of unknowns. In order to tackle these large scale problems both (i) new hardware architectures [60,61] as well as (ii) advanced numerical methods are required. A good overview of the available numerical methods can be found in Ref. [62–65].

Newly developed numerical methods focus on speeding up the two most time consuming parts in micromagnetic simulations, which are the calculation of the strayfield and the time integration of the LLG equation. Advanced time

integrations schemes can be found in Ref. [21, 22, 66–70]. For the calculation of the strayfield advanced FFT algorithms [24–26, 28, 71], fast multipole methods [31, 32], nonuniform grid methods [72], FEM/BEM coupling approaches including compression of the boundary matrix [35, 41, 73], and tensor grid methods [29, 30] have been developed.

Aside from new algorithms solving the micromagnetic model efficiently for systems with many degrees of freedom, it is often possible to choose a simplified physical model to describe at least some parts of the total problem. By this way the number of degrees of freedom can be reduced dramatically without loosing accuracy in regions where it is desired. One can utilize the fact that models described by the LLG equation require very fine grained discretizations which can lead to impractically large system sizes. We propose using the LLG equation to describe only those regions of the problem where detailed information about the domain structure such as domain walls and vortex structures are required. For the rest of the model a macroscopic description via magnetostatic Maxwell equations is chosen. Since it does not resolve the detailed domain structure it allows to use much coarser discretization.

In contrast to the multiscale method presented in this chapter there exist methods which solve combined LLG-Maxwell equations within the whole problem region [70, 74–78]. These methods extend the ordinary LLG model, by allowing to describe eddy currents or other dynamic effects like magnetostriction [79, 80], but they do not address the discretization size constraint and are therefore not suitable for large scale problems.

5.2 Fundamentals

For the coupling of Micromagnetism and magnetostatic Maxwell equations the full model is divided into two separated regions (see Fig. 5.1). The LLG equation is used to describe the first region Ω_{llg} , where domain structure, short range interactions or the magnetization dynamics of the magnetic parts is of great interest. The second region Ω_{max} is described by magnetostatic Maxwell equa-

tions, which describe the magnetic state in a spatially averaged sense and without dynamics. Since both models contain the external field as a source term, coupling via the strayfield can be achieved in a straightforward way. The strayfield created from the LLG model can be considered as an external field of the Maxwell model and vice versa. An additional region Ω_{coil} allows to define currents in a nonmagnetic medium, which in turn creates the source field for the magnetic model. The solution of the open-boundary problem requires the definition of the boundaries of the LLG-region (Γ_{llg}) as well as of the Maxwell-region (Γ_{max}).

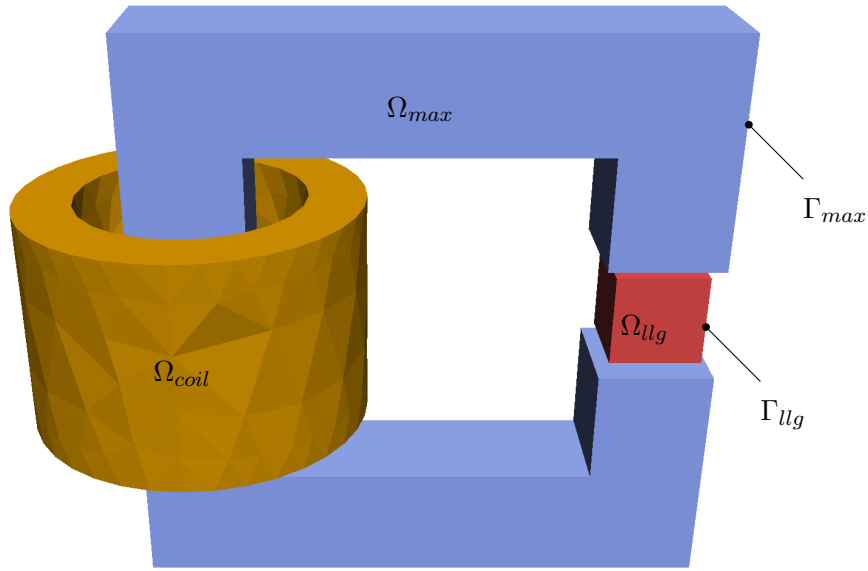


Figure 5.1: Example geometry which demonstrates model separation into LLG region Ω_{llg} and Maxwell region Ω_{max} (and in this case in an electric coil region Ω_{coil}). The boundaries of the regions are called Γ_{llg} and Γ_{max} respectively.

The individual solution of the two sub-problems has already been described in Chapter 3 for the LLG part or in Chapter 4 for the Maxwell part. The standard form of the LLG is applied in its strong form within the micromagnetic part and the strayfield contribution is calculated using the Fredkin-Koebler-Method. A detailed description of how the LLG equation is actually solved as well as a proper preconditioning method to speed up the calculation of large problems

can also be found in Ref. [67].

For the solution of the Maxwell part different scalar potential formulations can be selected. The Johnson-Nédélec FEM-BEM coupling is used to solve the open boundary problem. A detailed description of the methods used to solve the magnetostatic Maxwell equations can also be found in Ref. [3].

5.2.1 Discretization

The inhomogeneities within the LLG- as well as within the Maxwell-domain are discretized by means of finite elements. Within the LLG domain the element size is constrained by the exchange length of the used material. Typical values are in the range of $10nm$. Choosing larger elements would lead to unphysically large domain wall widths. For the Maxwell region such constraint does not exist, which allows to use much larger elements in this region.

In both cases FEM-BEM coupling methods are applied to handle the open-boundary problem. In addition to the fact that these methods are well suited for the solution of the individual problems they also simplify the coupling of the two methods because each method can be solved on its individual mesh without the need for a global mesh. The strayfield produced by each model which is needed to handle interactions can be calculated at any point by means of the boundary element formulas.

5.3 Interactions

The Maxwell and the LLG regions are coupled via their strayfield. The calculation of this strayfield depends on the equations used to describe the source model. The following subsections should summarize external field calculations for coil-, LLG- as well as Maxwell-models. All of these interactions have in common that they lead to dense matrices due to the long-range nature of the electromagnetic field. The various formulations used for the solution of the Maxwell problem (see Section 4.2) cause additional complications. Depending on the

used potential the magnetic field at the boundary, the magnetic field within the volume, and or the potential of the interaction field at the boundary is needed. In order to reduce the number of necessary case differentiations all interactions are split into three phases. The interaction field or potential is evaluated on the surface of the target model. Then these values are used as boundary conditions (either Neumann or Dirichlet conditions) in order to solve an internal potential equation. Finally from this potential the field as well as the potential itself can be evaluated on arbitrary internal locations.

5.3.1 Coil-Model Interaction

For the calculation of the field created by a coil the Biot-Savart law (4.24) can be used. If the current density is assumed constant within each element of the discretized coil, one can transform the volume integrals to surface integrals, by means of the Divergence theorem. In the following derivation the Einstein summation convention is used in order to simplify the notation of the vector products:

$$\begin{aligned}
 H_i &= \frac{1}{4\pi} \int_{\Omega} \epsilon_{ijk} j_j \frac{x_k - x'_k}{|\mathbf{x} - \mathbf{x}'|^3} d^3 \mathbf{x}' = \\
 &= \frac{1}{4\pi} \int_{\Omega} \epsilon_{ijk} j_j \partial'_k \frac{1}{|\mathbf{x} - \mathbf{x}'|} d^3 \mathbf{x}' = \\
 &= \frac{1}{4\pi} \int_{\partial\Omega} \epsilon_{ijk} j_j \frac{1}{|\mathbf{x} - \mathbf{x}'|} n_k d^2 \mathbf{x}' - \frac{1}{4\pi} \int_{\Omega} \epsilon_{ijk} \underbrace{\partial'_k j_j}_{=0} \frac{1}{|\mathbf{x} - \mathbf{x}'|} d^3 \mathbf{x}'
 \end{aligned} \tag{5.1}$$

In the usual vector notation the result can be summarized as

$$\mathbf{H}_{\text{coil}}(\mathbf{x}) = \frac{1}{4\pi} \int_{\partial\Omega} \frac{\mathbf{j} \times \mathbf{n}}{|\mathbf{x} - \mathbf{x}'|} d^2 \mathbf{x}' \tag{5.2}$$

where \mathbf{n} is the normal vector of the integration domain. It has to be emphasized that the integration is carried out for each volume element of the coil. Therefore surface integrals have to be calculated for all element surfaces, not only for the surfaces on the boundary of the coil.

For the calculation of the interaction potential u_{coil} the coil-field is evaluated at all surface elements of the target model. Solution of a Laplace equation with Neumann boundary conditions determined by the coil-field results in the desired interaction potential

$$u_{coil}^i \int_{\Omega} \nabla \phi^i \cdot \nabla \phi^j d\Omega - \int_{\Gamma=\partial\Omega} \underbrace{\nabla u_{coil}}_{\mathbf{H}_{coil}} \cdot \mathbf{n} \phi^j d\Gamma = 0 \quad (5.3)$$

where ϕ^i and ϕ^j are the basis functions corresponding to node i or j respectively.

5.3.2 LLG-Model Interaction

The calculation of the exterior strayfield calculated by an LLG-model is very similar to the calculation of the internal strayfield as described in Section 3.3. Application of the Fredkin-Koebler method (see Eqn. (3.30)) allows to reuse the u_1 potential from the internal strayfield calculation allows to calculate the interaction potential u_2 at arbitrary locations by means of a single matrix vector multiplication. The interaction matrix looks like the double-layer matrix from the second phase of the Fredkin-Koebler method (see Eqn. (3.30d)). Since the potential u_1 is only defined inside of the source model and it is set to 0 outside, the total interaction potential $u_{llg} = u_2$.

Since in this case the potential u_{llg} is already known at the boundary of the target model, a Laplace equation with Dirichlet boundary conditions has to be used to determine the internal potential. As before from magnetic field as well as the potential can then be evaluated at all desired locations inside of the target model.

5.3.3 Maxwell-Model Interaction

The interaction Maxwell-Model depends on the used FEM-BEM coupling methods, because the methods describe the exterior solution differently. The Johnson-Nédélec FEM-BEM coupling assumes an exterior solution of the form

$$u^- = \iint_{\Gamma} \nabla u^- G - u^- \nabla G d^2 \mathbf{x}' \quad (5.4)$$

which can be derived from Green's second identity. This formula shows that the exterior potential only depends on the potential and its normal derivative on the boundary of the source model, which is the case because the external potential fulfills a homogeneous Laplace equation. For the calculation a single-layer- as well as a double-layer-integral needs to be evaluated, which makes the computation slightly more expensive, compared with the interaction of LLG-models.

5.4 Coupling Method

In order to solve the coupled problem one needs to deal with ordinary differential equations (ODE), which arise from the spatial discretization of the LLG equations, as well as with algebraic equations arising from Maxwell's equations. Discretization of this system of Differential-Algebraic-Equations (DAE) using integration methods for ODEs can lead to numerical instabilities or to a drift error in the algebraic equations [81]. Therefore differential and algebraic equations are kept separate and a sequential method is used to combine both problems. A first implementation simply solves the Maxwell problem every time the righthand-side of the LLG equation is solved. In an abstract notation this can be written as:

$$\dot{y}_{llg} = LLG(t, y_{llg}, y_{max} = MAX(t, y_{llg})) \quad (5.5)$$

where y_{llg} and y_{max} are the unknowns of the LLG as well as of the Maxwell models. For the time discretization the backward differential formula (BDF) is applied to the ODE system and is in turn solved by means of an Inexact Newton method (the open source differential equation solver CVODE [53] have been used for this task). Since an implicit time integration scheme is used, which needs to approximately solve a system of equations within every timestep, the LLG equation (3.14) needs to be evaluated several times during each timestep. In order to calculate H_{ext} within the righthand-side of the LLG equation, the Maxwell problem (4.2) is solved under consideration of the strayfield produced

by the actual magnetization of the LLG model. After the Maxwell system is solved the back-interaction on the LLG model can be calculated and allows to finally evaluate the righthand-side of the LLG system. This procedure leads to a fully implicit scheme to solve the coupled equation.

5.4.1 Optimization

Solving the Maxwell problem within every function evaluation of the LLG time-integration can be very time-consuming for considerably large Maxwell models. Fortunately the simulation can be speeded up by using the theoretical prediction [5] that for stability of the time-integration only the exchange interaction term needs to be handled implicitly. All other terms, including the interaction with the Maxwell model, can be handled explicitly, which means that we simply use the interaction field of the last timestep to evaluate all function values needed at the next timestep. Thus we only need to solve the Maxwell problem once for every non-linear step of the time-integration. As a consequence the order to the used time-integration is reduced. For example using a BDF method of order 2 will only show a reduced order of 1, if the interactions are only updated after each non-linear step. One way to overcome this limitation is not to use a constant interaction but perform an extrapolation from previously calculated steps within the nonlinear iteration. This idea, which is also employed on multi-step methods, allows to restore the original order of 2 of the BDF(2)-method.

5.5 Test Cases

The following subsections will present some of the examples used to validate the coupling within the multiscale algorithm. In order to simplify the calculation and to narrow down the possible error sources the first test cases only consider unidirectional interactions. The test case presented only considers the interaction from a macroscopic Maxwell model onto a small micromagnetic part and the back-interaction is ignored. In practice this situation is very common since the back interaction may often be neglected because it is too weak. Coun-

terexamples are permanent magnets which can well be described by a (macroscopic) LLG model. In this case the unidirectional interaction from an LLG- to a Maxwell-model could be of interest. Finally the full bidirectional interaction will be demonstrated by means of an analytical test case as well as of an experimental measurement. Analytical results as well as measurement data could be reproduced very well by means of the multiscale algorithm presented in this work.

5.5.1 Unidirectional interaction

Problem Description

Starting from the magnetic yoke test case presented in Section 4.8.2 a small micromagnetic sphere is placed within the air-gap. The field within the air-gap is nearly homogeneously which allows to assume it as constant for an analytical estimation. Initially the LLG-sphere is homogeneously magnetized and there will not form any domains due to the small size of the particle. Therefore the external field created of the homogeneously magnetized sphere is equal to that of a single magnetic dipole. The whole problem can therefore be reduced to the interaction of a single dipole with and homogeneous external field. According to the LLG equation the solution of this problem should be a damped precessional motion, where the magnetization of the sphere is finally switched into the direction of the applied field. The precession frequency depends on the applied field strength, which could be estimated as shown in the original test case, as well as on the used damping parameter.

Analytical Calculation

The analytical calculation presented is based on Ref. [82], but with a slightly modified notation. The original paper also considers the magneto crystalline anisotropy and finally derives the achievable switching time. In this section only the external field contributes to the effective field, and the whole trajectory of the magnetic spin should be calculated. In [83] the calculation of the

z -component of the magnetization is shown, based on some symmetry considerations. It provides a simplified derivation of the switching times, but it does not derive the solution for the whole trajectory of the problem.

One starts from the LLG equation in the Gilbert form for a single spin considering only the external field term:

$$\dot{\mathbf{M}} = -\gamma \mathbf{M} \times \mathbf{H}_{\text{ext}} + \frac{\alpha}{M} \mathbf{M} \times \dot{\mathbf{M}} \quad (5.6)$$

The next step is to introduce spherical coordinates based on the vector \mathbf{M} :

$$M_x = M \sin(\theta) \cos(\phi) \quad (5.7a)$$

$$M_y = M \sin(\theta) \sin(\phi) \quad (5.7b)$$

$$M_z = M \cos(\theta) \quad (5.7c)$$

Local basis vectors can be defined by considering small variations of the new parameters r , θ and ϕ . These can be calculated by means of the chain rule for the time derivative:

$$\frac{\partial(M_x, M_y, M_z)}{\partial t} = \frac{\partial(M_x, M_y, M_z)}{\partial(M, \theta, \phi)} \frac{\partial(M, \theta, \phi)}{\partial t} \quad (5.8)$$

$$\begin{pmatrix} \dot{M}_x \\ \dot{M}_y \\ \dot{M}_z \end{pmatrix} = \begin{pmatrix} \sin(\theta) \cos(\phi) & M \cos(\theta) \cos(\phi) & -M \sin(\theta) \sin(\phi) \\ \sin(\theta) \sin(\phi) & M \cos(\theta) \sin(\phi) & M \sin(\theta) \cos(\phi) \\ \underbrace{\cos(\theta)}_{\mathbf{e}_r} & \underbrace{-M \sin(\theta)}_{M \mathbf{e}_\theta} & \underbrace{0}_{M \sin(\theta) \mathbf{e}_\phi} \end{pmatrix} \begin{pmatrix} \dot{M} \\ \dot{\theta} \\ \dot{\phi} \end{pmatrix} \quad (5.9)$$

Using this local basis vectors the original LLG can be rewritten in spherical coordinates. Since the local basis is orthonormal the cross product can be simple transformed. Therefore only the remaining vectors need to be transformed into the new basis. The external field vector is assumed to point into the $+z$ direc-

tion:

$$\mathbf{M} = M \mathbf{e}_r = \begin{pmatrix} M \\ 0 \\ 0 \end{pmatrix}_{M,\theta,\phi} \quad (5.10)$$

$$\mathbf{H}_{\text{ext}} = H \cos \theta \mathbf{e}_r + H \sin(\theta) \mathbf{e}_\theta = \begin{pmatrix} H \cos(\theta) \\ H \sin(\theta) \\ 0 \end{pmatrix}_{M,\theta,\phi} \quad (5.11)$$

Putting everything together and carry out the cross product finally yields:

$$\begin{aligned} \begin{pmatrix} \dot{M} \\ M \dot{\theta} \\ M \sin(\theta) \dot{\phi} \end{pmatrix}_{M,\theta,\phi} &= -\gamma \begin{pmatrix} M \\ 0 \\ 0 \end{pmatrix}_{M,\theta,\phi} \times \begin{pmatrix} H \cos(\theta) \\ H \sin(\theta) \\ 0 \end{pmatrix}_{M,\theta,\phi} + \\ &+ \frac{\alpha}{M} \begin{pmatrix} M \\ 0 \\ 0 \end{pmatrix}_{M,\theta,\phi} \times \begin{pmatrix} \dot{M} \\ M \dot{\theta} \\ M \sin(\theta) \dot{\phi} \end{pmatrix}_{M,\theta,\phi} = \\ &= -\gamma \begin{pmatrix} 0 \\ 0 \\ M H \sin(\theta) \end{pmatrix}_{M,\theta,\phi} + \alpha \begin{pmatrix} 0 \\ -M \sin(\theta) \dot{\phi} \\ M \dot{\theta} \end{pmatrix}_{M,\theta,\phi} \end{aligned} \quad (5.12)$$

From the first line of the equation it is evident the the absolute value of the magnetization is preserved, whereas the second and third line describe the motion into the polar and azimuthal direction. By inserting the left hand side of the third equation into the right hand side of the second one can eliminate $\dot{\phi}$ and ends up with the following differential equation:

$$\dot{\theta} = \frac{\alpha \gamma}{1 + \alpha^2} H \sin(\theta) \quad (5.13)$$

which can be solved by separation of variables:

$$\int_{\theta_0}^{\theta} \frac{d\theta}{\sin(\theta)} = \int_{t_0=0}^t \frac{\alpha \gamma}{1 + \alpha^2} H dt \quad (5.14)$$

$$\log \left(\tan \left(\frac{\theta}{2} \right) \right) - \log \left(\tan \left(\frac{\theta_0}{2} \right) \right) = \frac{\alpha \gamma}{1 + \alpha^2} H t \quad (5.15)$$

$$\theta = 2 \arctan \left(\tan \left(\frac{\theta_0}{2} \right) \exp \left(\frac{\alpha \gamma}{1 + \alpha^2} H t \right) \right) \quad (5.16)$$

The last step to derive the full form of the precessional motion is to consider the azimuthal part of the equation. The second line of the spherical LLG equation leads to a relation between $\dot{\theta}$ and $\dot{\phi}$ in form of another differential equation. This equation can again be solved by means of variable separation and fortunately the terms on the θ side are equal to those already solved for the polar part. It follows that there is a linear dependence of the azimuth angle on the time if no anisotropy is in place:

$$\int_{\phi_0}^{\phi} -\alpha d\phi = \int_{\theta_0}^{\theta} \frac{d\theta}{\sin(\theta)} = \int_{t_0=0}^t \frac{\alpha \gamma}{1 + \alpha^2} H dt \quad (5.17)$$

$$\phi - \phi_0 = -\frac{\gamma}{1 + \alpha^2} H t \quad (5.18)$$

Simulation Results

A yoke of dimension $5nm \times 5nm \times 1nm$ with an air-gap of $1nm$ is used. Applying a current of $1mA$ creates a field $H = 10^6 A/m$. The damping constant is set to $\alpha = 0.2$ and the initial angle $\theta_0 = \arctan(0.01)$. The simulated trajectories are compared with the analytical result in Fig. 5.2. Except for the small phase shift both results are in good agreement with each other.

5.5.2 Bidirectional interaction

Problem Description

This testcase utilizes the behavior of a magnetic plate with a high magnetic susceptibility. Such a plate acts like a mirror for magnetic charges (which are the components of magnetic dipoles). As before a small micromagnetic sphere is used as LLG model because it creates a simple dipole field and can thus be treated as a single spin. Fig. 5.3 shows how such a spin is mirrored on a perfect magnetic plate. In order to validate the bidirectional interaction between

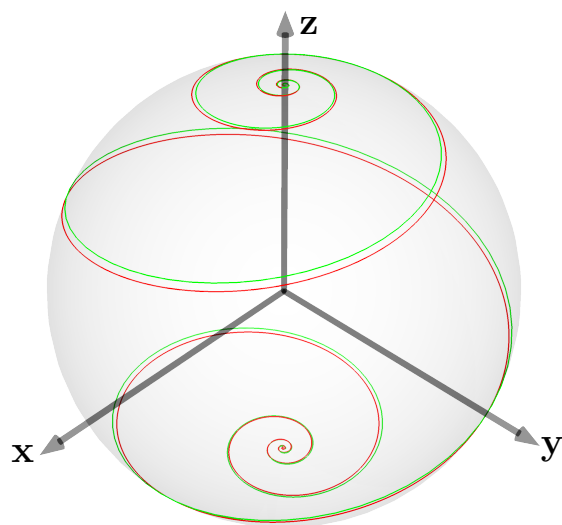


Figure 5.2: Comparison of the analytical result (red) for the trajectory of a single spin within an external field with the multiscale simulation of a small magnetic sphere within the air gap of a magnetic yoke (green). The initial magnetization was pointing into the $+z$ direction and an external field is applied in $-z$ direction. It can be seen that numerical simulation is able to reconstruct the analytical result very well.

Maxwell and LLG models, one now describes the magnetic sphere using the LLG and the (nearly) perfect magnetic plate by means of Maxwell's equations and lets the two models interact with each other. Since this model corresponds with the interaction of two individual spins there exists a lowest energy state, which is reached if both spins are in parallel alignment.

Simulation Results

The dimension of the magnetic plate is selected as $20nm \times 20nm \times 10nm$ and its susceptibility is chosen to be 10^6 . Above the center of the plate the micromagnetic sphere with radius $0.2nm$ is placed at a distance of $0.2nm$. The mesh of the plate is refined towards the location of the sphere, where for elements closest to the sphere sizes in the range of $0.1nm$ are used. The further away an element is located the larger it can be. By this way very large geometries can be discretized with a relatively small number of elements. For the plate in this test case 3244

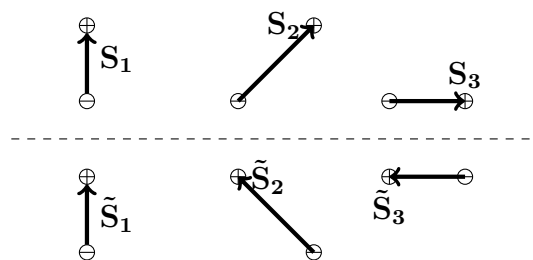


Figure 5.3: The mirroring of three individual spins on a perfectly magnetic surface is visualized. The spins are illustrated as two magnetic charges of opposite sign. The location of each charge is mirrored on the magnetic surface and its sign is reverted in order to get the mirrored charge. By this way the three initial spins S_1 , S_2 , and S_3 are transformed into the mirrored spins \tilde{S}_1 , \tilde{S}_2 , and \tilde{S}_3 .

elements were used. Different simulations have been performed with various values for the saturation polarization of the magnetic sphere as well as for the initial direction of its polarization. The different trajectories are visualized in Fig. 5.4 and it can be seen that all of them finally reach the correct stationary state, which point either in the positive or negative z direction.

5.5.3 GMR Read Head Transfer Curve

Problem Description

In this section the developed algorithm is applied to calculate the transfer curve [84] of a magnetic read head setup (see Fig. 5.5). For the transfer curve a homogeneous external field perpendicular to the medium is applied to the whole setup and the stationary output of the read head is plotted as a function of the field strength. The medium is not considered in this simulation. In practice such transfer curves are used to characterize magnetic read heads and they are experimentally measured with field sweep rates much lower than the rates which occur in the actual magnetic recording process.

The giant magnetoresistance (GMR) reader element consists of microscopic layers which can be well described by the LLG. On the other hand the setup consists of macroscopic shields which have the purpose to reduce the influence of the strayfield from neighboring bits on the output signal of the current

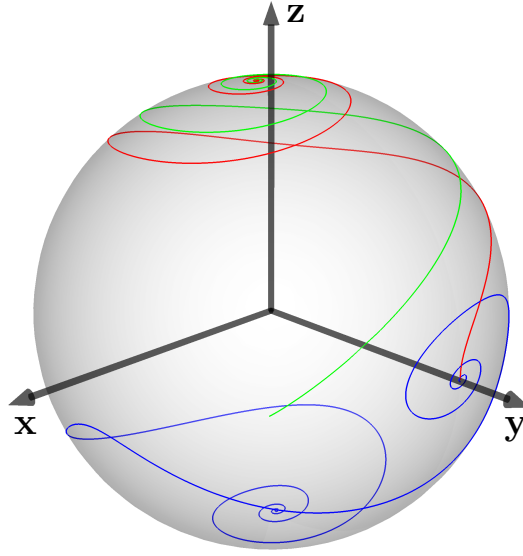


Figure 5.4: Trajectories for three different initial conditions (red, green and blue curve) of the image charge testcase. It can be seen that all three simulations end up with magnetization pointing into the positive or negative z direction. Both directions correspond with the minimum energy solution, where the spin and its mirror image need to be parallel aligned. The trajectories of the coupled system may be very complicated due to the mirror constraint. The small offset of the trajectory endpoint from the axis can be explained by the coarse mesh that was used for the discretization of the magnetic plate.

bit. Since these shields are separated from the sensor element and its domain structure is not of interest, it is possible to describe them using magnetostatic Maxwell equations. Additionally for the transfer curve stationary states for different values of the external field are needed, which means that the dynamic of the shields can be safely ignored in this case. Therefore the usage of the multiscale method allows to reduce the simulation time significantly.

The detailed structure of the giant magnetoresistance sensor is shown in Fig. 5.6. Due to the GMR effect the resistance of the freelayr changes depending on the cosine of the angle ϕ between the magnetization within free- and pinned layer. The transfer curve thus shows $\cos(\phi)$ for various external field strengths.

In order to demonstrate the strengths of the presented algorithm we calculate transfer curves for different grid sizes of the magnetic shields. The results of the multiscale algorithm (see Fig. 5.7) are compared with those of an LLG-

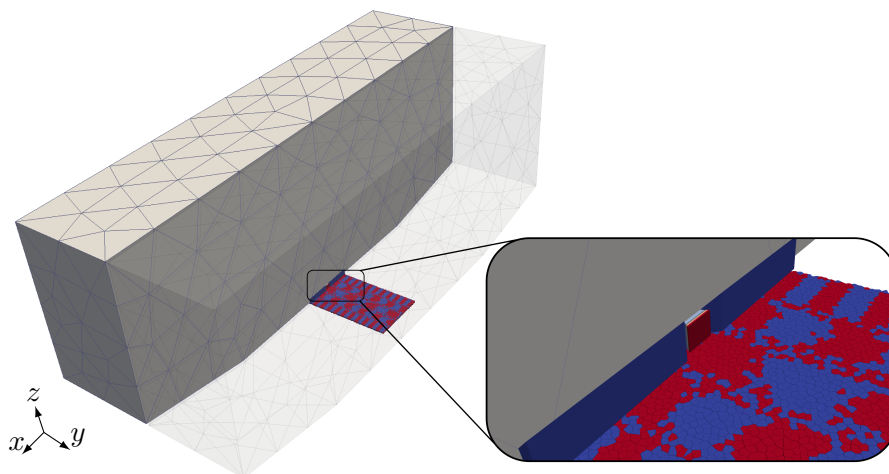


Figure 5.5: The example setup consists of a GMR sensor element in between two macroscopic shields ($5\mu m \times 2\mu m \times 2\mu m$). Beyond the GMR sensor a magnetic storage medium is indicated (it will not be considered for the calculation of the transfer curves).

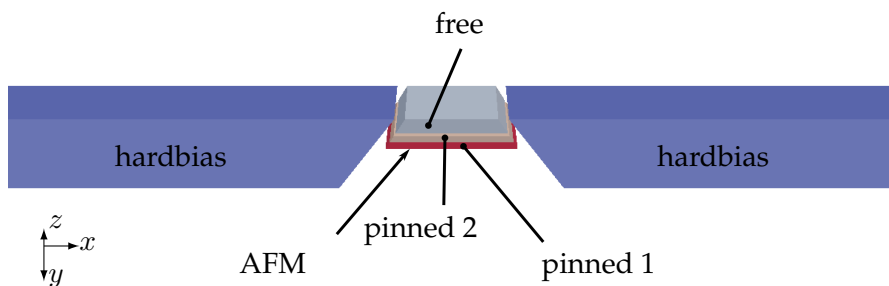


Figure 5.6: The GMR reader element for our test case consists of two pinned layers which are antiferromagnetically coupled via a thin Ruthenium layer between them. Their magnetization is pinned by a granular antiferromagnetic layer below the bottom layer. The initial magnetization of these two layers is chosen to direct in-plane into the z or $-z$ direction respectively. The freelay is located above the pinned layers and its initial magnetization is forced orthogonal to the magnetization of the pinned layer by means of two hardbias magnets whose easy axes show into the x -direction. The output signal of the sensor element is proportional to the cosine of angle ϕ between the magnetization within the pinned- and the free-layer. For sake of simplicity electrodes are not included in this model. The total size of the model is $550nm \times 60nm \times 20nm$.

only simulation. For the material of the Maxwell part we use a linear material law with permeability $\mu = 1\,000$ up to the saturation polarization J_s , which is

the same as in the LLG-only case. The LLG parameters used are summarized in Tab. 5.1. Because we are only interested in the stationary state for certain external field amplitudes, $\alpha = 1$ has been chosen, which maximizes the energy dissipation and therefore leads to the fastest transition into the stationary state. A complete hysteresis is calculated by changing the applied field amplitude from $-2T$ to $+2T$ and then back to $-2T$ again. By this way it is possible to check the reproducibility of the stationary states.

	uniaxial anisotropy constant K_1 [J/m^3]	magnetic polarization J_s [T]	exchange constant A [J/m]	Gilbert damping constant α [1]	layer thickness t [m]
shields	200.0	1.0	1.25e-11	1.0	-
hardbias	1.0e6	0.6	1.25e-11	1.0	-
free	20.0	1.21	1.25e-11	1.0	-
pinned 2	20.0	1.0	1.25e-11	1.0	-
pinned 1	20.0	1.0	1.25e-11	1.0	-
AFM	0.0	0.01	-8.0e-13	1.0	0.8e-9

Table 5.1: Summary of LLG parameters used for the read head simulation. The easy axis of all anisotropic materials is directed in x -direction.

Simulation Results

Starting with an LLG-only simulation using an external field sweep rate of $0.2T/ns$ one can clearly see that the stationary states are not reached and the curves show strong dynamic fluctuations (see Fig. 5.8). Thus another LLG-only simulation with a much slower field sweep rate was performed and it was possible to significantly reduce the dynamic artifacts in the calculated transfer curves (see Fig. 5.9). Nevertheless one also notices that both versions of the LLG-only simulation suffer from a strong dependence on the grid size used for the discretization of the magnetic shields. Due to the use of magnetostatic Maxwell equations both of these problems do not occur when using the multi-scale method (see Fig. 5.7). Finally a comparison of the different algorithms for proper grid sizes as well as field sweep rates (see Fig. 5.10) shows that the re-

sults are in good agreement with each other, provided that one uses a saturated material law within the Maxwell part.

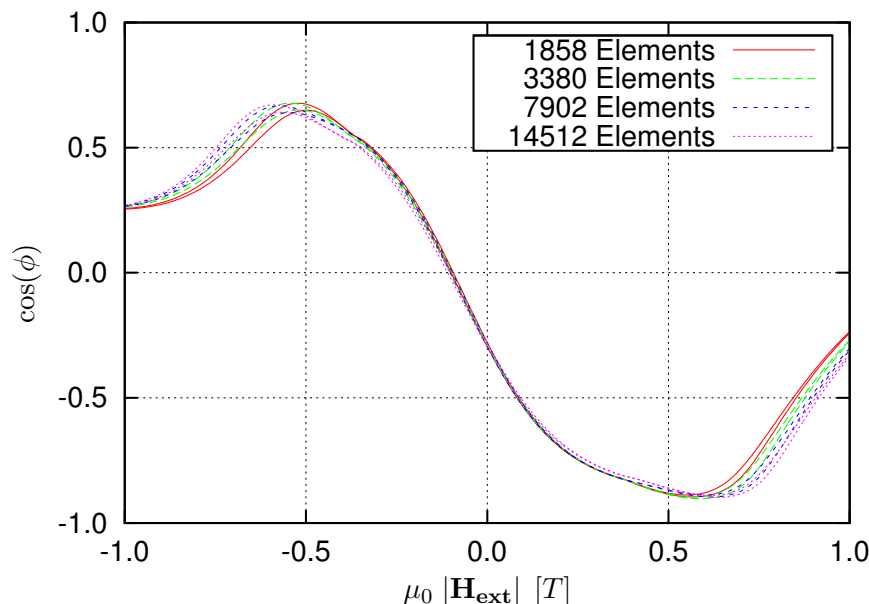


Figure 5.7: Calculated transfer curves of the presented multiscale algorithm for various shield grid sizes. The angle ϕ between the magnetizations within pinned- and free-layer is plotted as a function of the external field \mathbf{H}_{ext} . The algorithms starts to produce convergent results at around 1 000 elements and due to the use of magneto-static equations it shows no dynamic artifacts.

A comparison of the performance of the multiscale algorithm with and without optimization, as well as with LLG-only methods is presented in Tab. 5.2.

5.6 Conclusion

A multiscale algorithm was presented which combines the capabilities of LLG- as well as Maxwell- equation solvers and allows to handle much larger problem sizes. Coupling the LLG to the Maxwell part could be optimized by handling the corresponding terms explicitly within the time integration scheme. Finally the optimized algorithm was validated by means of a transfer curve simulation of a magnetic read head. The results of the multiscale algorithm match very

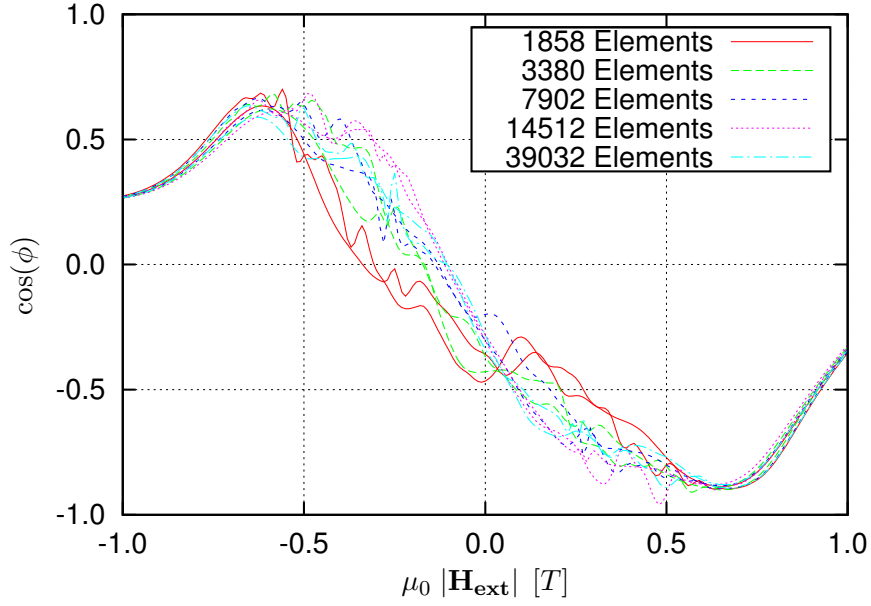


Figure 5.8: Calculated transfer curves of LIG-only simulations for various shield grid sizes. In contrast to the multiscale algorithm it shows some significant deviations up to at least 50 000 elements. Additionally there occur some artifacts due to the large timeconstant of the magnetization dynamic within the macroscopic shields. The field sweep rate is chosen too high to reach the equilibrium state and therefore leads to some fluctuations within the transfer curves.

	LLG*	LLG (conv.)	Multiscale	Multiscale (optimized)
t_{sim} [ns]	70	1 000	70	70
r_{sweep} [T/ns]	0.2	0.01	0.2	0.2
N_{shield}	3 380	39 032	3 380	3 380
N_{max}	-	-	22 274	6 038
N_{llg}	69 584	173 227	22 274	22 203
t_{run} [s]	1 084	173 184	11 857	8 107

Table 5.2: Algorithm performance for the transfer curve calculation. t_{run} is the overall runtime of the simulation. The simulation period t_{sim} as well as the shield grid size N_{shield} are chosen in a way that the algorithm produces convergent results (LLG* shows non-converged results and is listed only to give a reference time scale). r_{sweep} is the sweep rate of the applied external field. N_{max} and N_{llg} are the number of times the Maxwell- or the LLG-part is evaluated, respectively.

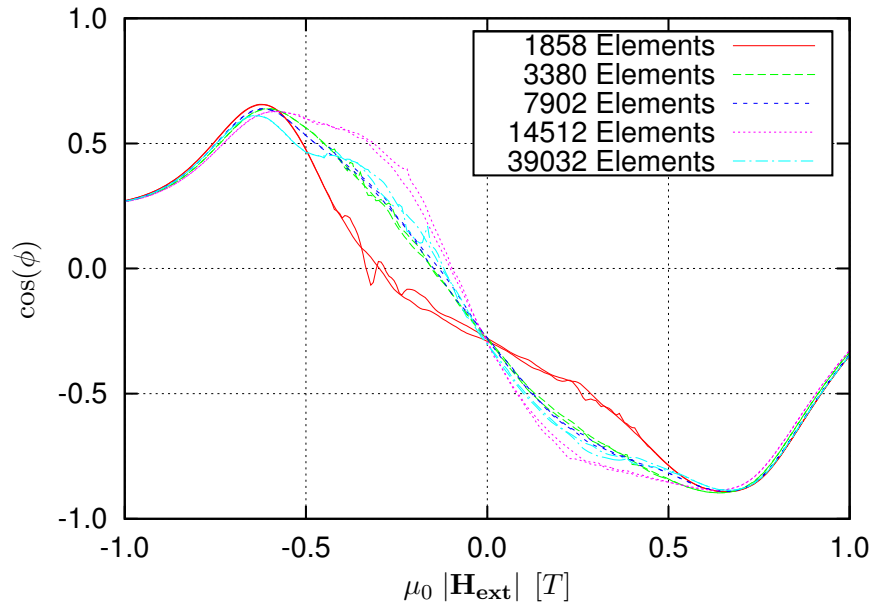


Figure 5.9: Calculated transfer curves of LLG-only simulations using a field sweep rate reduced by a factor 5. The fluctuations are reduced, but there are still significant differences due to the coarse shield discretization.

well with those of the LLG-only simulation, but it allows to significantly reduce the shield gridsize and to increase the field sweep rates.

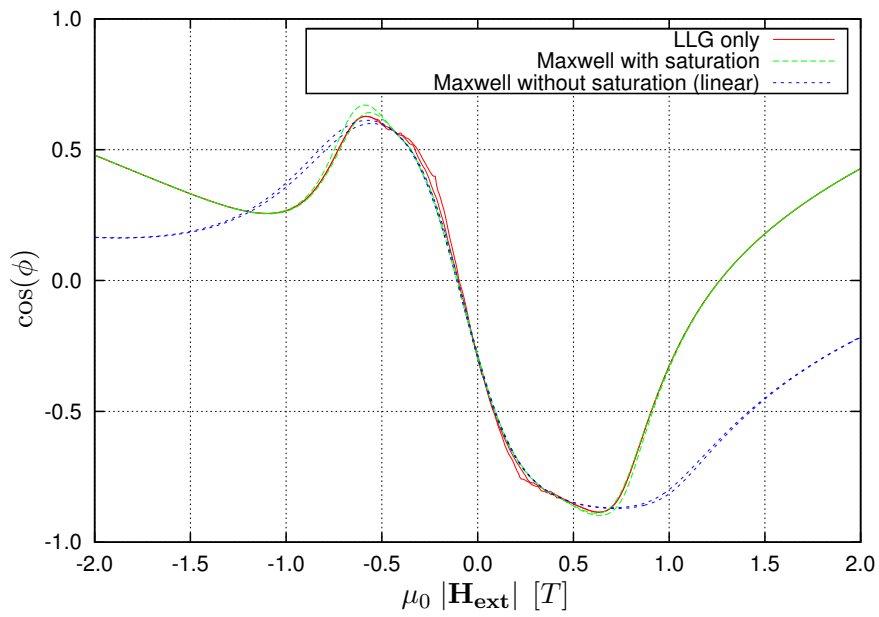


Figure 5.10: Simulation results of the multiscale solver compared with an LLG-only calculation. Both algorithms are applied to the same model with 14512 elements. A permeability of 10^3 is used for the material of the shields within the Maxwell solver. In order to get matching results at high fields one needs to use a saturated material law.

Parallelization of \mathcal{H} -matrices

6.1 Introduction

In this chapter a totally different approach of extending micromagnetic simulations is presented. Benchmarks of recent micromagnetic codes show that a large part of the total computation time is spent within the strayfield calculation. The occurring dense BEM matrices lead to relatively high costs compared with the sparse FEM matrices, even if matrix compression schemes like \mathcal{H} -matrices or fast-multipole-methods are used. One way to further optimize the computation time of algorithms is parallelization. Depending on the underlying algorithm parallelization can be a more or less sophisticated task. One advantage of BEM matrices, which are based on an integral formulation, is that their setup and matrix-vector multiplication is perfectly parallelizable. For the setup individual blocks of the corresponding \mathcal{H} -matrix can be calculated independently of each other. For the matrix-vector-product different lines within the \mathcal{H} -matrix contribute to different parts of the result vector, which also makes their evaluation independent of each other. The next section will outline the implementation of a parallel \mathcal{H} -matrix setup and matrix-vector multiplication based on an OpenMP shared memory parallelization. Similar parallelization methods can be found in [85,86].

6.2 Implementation

The parallelization presented in the following section is based on OpenMP [87, 88], which provides an easy tool for shared-memory parallelization. OpenMP uses C pragma pre-processor directives, which allows to parallelize whole section of the code with only few modifications of the original source code. Especially if there is no communication between the parallel tasks, as it is the case for that following implementations, parallelization can be archived with only a few additional lines of code.

The following list introduces a few of the most important OpenMP directives which where used to parallelize the \mathcal{H} -matrix code:

- **#pragma omp parallel**

Putting this clause in front of a code block makes it run within each of the created threads. This can be useful to setup some local data structures.

- **#pragma omp master**

Sometimes it is necessary to execute parts of the code only on a single master node. For example if the final results should be printed or written to a file, this should be done only on the master node.

- **#pragma omp parallel for**

This clause is inserted in front of a *for*-loop and it tells the compiler to split the work of the loop into several threads. The loop-counter by default becomes a local variable within each thread, whereas all other variables are shared unless they are explicitly defined as local by adding them to the *private*-list at the end of the statement. Work sharing among the different threads can be done by static or dynamic scheduling by selecting the proper scheduling algorithm at the end of the statement.

- **#pragma omp critical**

This statement allows to define critical sections for OpenMP. These provide a method to control the access to certain memory regions, devices, and so on from different threads, in order to prevent race-conditions.

6.2.1 Matrix Setup

Since \mathcal{H} -matrices are based on tree data structures recursive algorithms are well suited to handle them. Unfortunately recursive algorithms require more work to be parallelized directly, compared with procedural code, especially when using OpenMP. Therefore the first hierarchy levels of the tree structure are transformed into a list within a pre-processing step. After this is done the toplevel setup process can be replaced by a simple *for*-loop over the list items, where for each item the original recursive setup algorithm is called:

```

1 /* use OMP parallel for loop to parallelize H-matrix setup */
2 num_blocks = create_blocklist(matrix, blocklist);
3 #pragma omp parallel for schedule(dynamic)
4 for(i=0; i < num_blocks; i++) {
5     setup_block_recursively(blocklist[i]);
6 }

```

The main work of the algorithm has been shifted into the creation of the list of independent tasks. For the setup of the matrix this can be simply archived by recursively walking through the \mathcal{H} -matrix hierarchy levels until a certain number of sub-blocks is reached and then insert all of these blocks into a list, like it is visualized in Fig. 6.1.

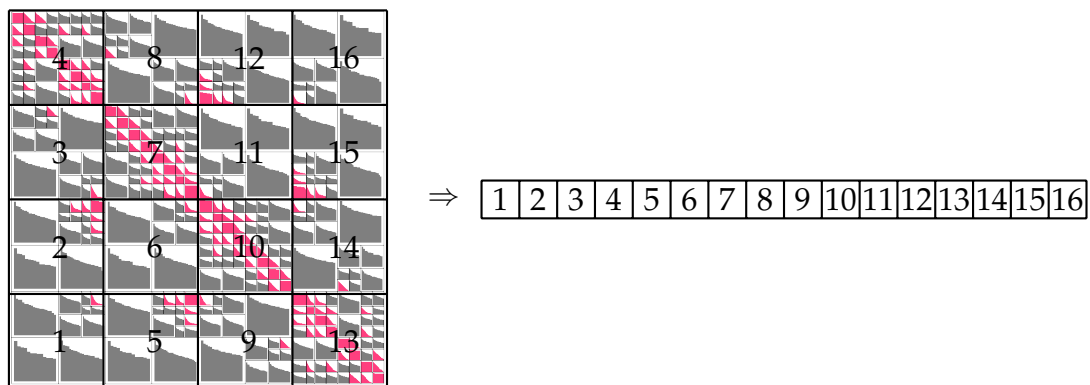


Figure 6.1: Visualization of how \mathcal{H} -matrix blocks are rearranged in an ordered list in order to allow simple parallelization using OpenMP *parallel for* constructs.

6.2.2 Matrix-Vector Multiplication

For the matrix-vector multiplication things are very similar to the setup. Instead of individual blocks of the matrix one searches for individual lines and transforms the toplevel tree structure into a list of lines. As before the new parallel algorithm which handles the line-lists looks very simple:

```

1  /* use OMP parallel for loop to parallelize H-matrix-vector
   multiplication */
2  num_lines = create_linelist(matrix, linelist);
3
4  /* loop over lines in parallel */
5  #pragma omp parallel for schedule(dynamic)
6  for(i=0; i < num_lines; i++) {
7      /* loop over blocks within each line sequentially */
8      for(j=0; j < linelist[j]->num_blocks; j++) {
9          multiply_block_recursively(linelist[i]->blocks[j]);
10     }
11 }
```

In contrast to the matrix setup the transformation of the toplevel \mathcal{H} -matrix tree-structure into lists of lines is slightly more complicated. The main problem is that not all blocks of a line have to belong to the same hierarchy level within the matrix. For example the left part of a matrix can be first horizontally split and then each part is split vertically, whereas the right part of the matrix is first split vertically and then horizontally. Therefore a tree-structure which contains the list of blocks per line is introduced as an intermediate data-structure. The toplevel matrix hierarchy as well as the corresponding data structures are presented in the Figures 6.2, 6.3, 6.4. The final transformation from this line-tree-structure of lists of blocks which belong to a certain line is straight-forward. As for the matrix setup one simply walks through the line-tree down to a hierarchy level which contains enough lines and puts the corresponding nodes into independent lists.

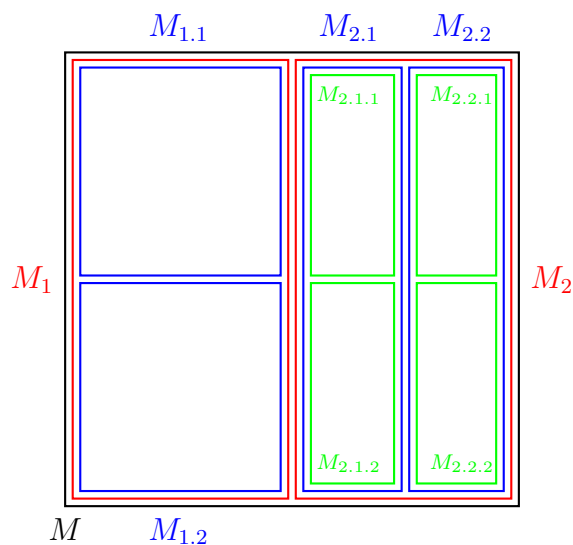


Figure 6.2: Example top-level \mathcal{H} -matrix structure used to explain the data structures introduced for the parallelization of the \mathcal{H} -matrix-vector product.

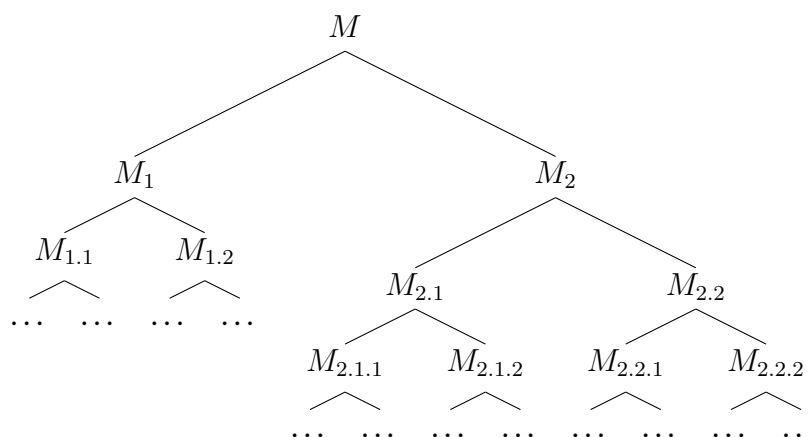


Figure 6.3: Underlying \mathcal{H} -matrix block-cluster-tree of the above example, which shows that blocks that belong to the same matrix line must not be located within the same hierarchy level of the \mathcal{H} -matrix data structure.

6.3 Benchmark

For the benchmarks a relatively large scale problem of a magnetic recording write head was simulated. The times spent within double-layer \mathcal{H} -matrix setup as well as \mathcal{H} -matrix-vector product are measured. The model consists of 408.712

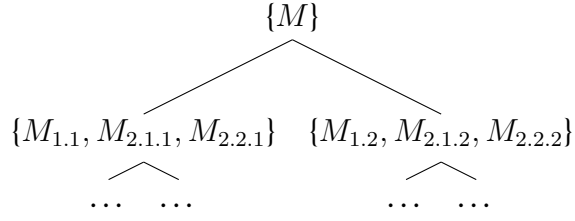


Figure 6.4: Intermediate tree data-structure, where all the blocks are added to a list corresponding to the line in which they are located. The first hierarchy level of this tree contains two lists, one for blocks in the upper of the matrix and one for the blocks in to lower part. For each further hierarchy level each list is again split into two sub-lines until enough independent lines are available.

tetrahedral elements which leads to a storage size of the measured matrix of 12.2 GB. The test machine was equipped with 2 Intel Westmere X5650 Six-Core 2.66GHz processors with Intel Hyper-Threading Technology, which makes a sum of 24 virtual cores.

The measured timings for different numbers of threads are shown in Fig. 6.5 for the matrix setup and in Fig. 6.6 for the matrix-vector multiplication. Both figures show that the parallelization of the CPU time performs very well, but unfortunately for the parallelization of the matrix-vector multiplication, the bandwidth of the main memory seems to be another bottleneck. The upper limit of 20GB/s was determined by means of `stream` [89], which is a very handy tool for measuring main memory and cache bandwidth and is available for a large range of hardware. It is interesting that on modern hardware the maximum memory bandwidth is only reachable for multi-threaded application, because simultaneous memory access is supported. Therefore one gets different results for the maximum memory bandwidth for single-threaded and multi-threaded applications.

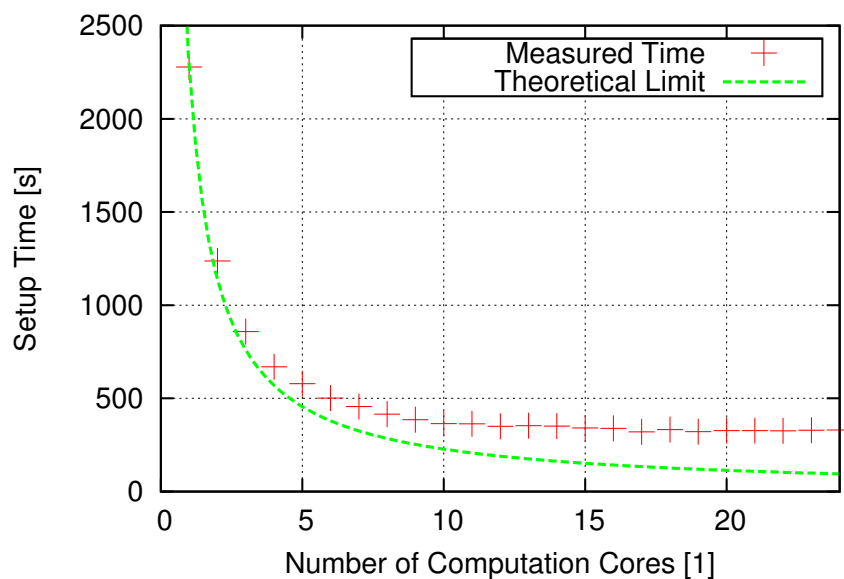


Figure 6.5: Setup time of the 12GB \mathcal{H} -matrix for a different number of computation cores. Since the setup of individual matrix blocks is independent from each other and the theoretical scaling limit of $O(1/N)$ for N computation cores is nearly reached.

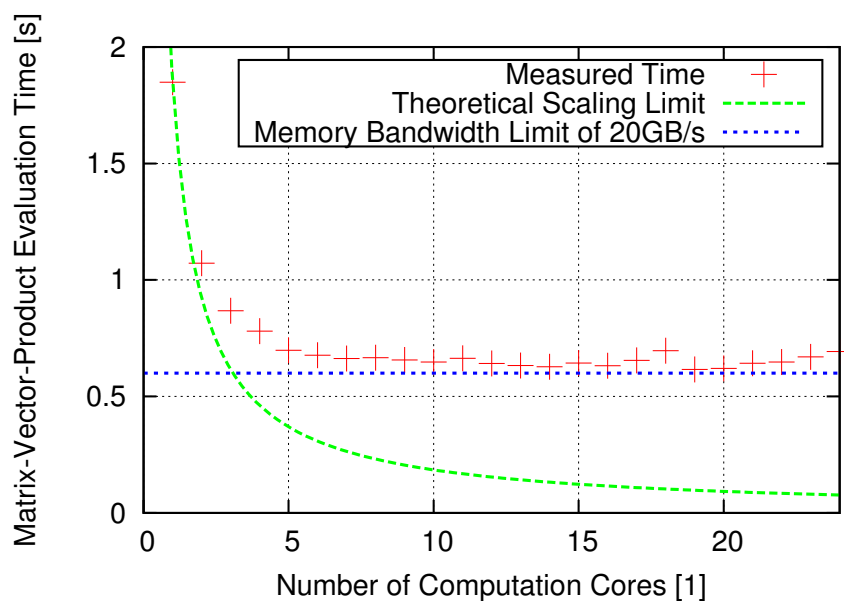


Figure 6.6: Matrix-vector multiplication time of the 12GB \mathcal{H} -matrix for a different number of computation cores. It can be seen that that the parallelization works well for a small number of computation cores, but it then stagnates due to a limited memory bandwidth.

Conclusion and Outlook

7.1 Conclusion

The multiscale algorithm developed during this thesis proved very useful in extending magnetic simulations, which consists of parts of significantly different size. Its performance could be validated by means of various test cases as well as the simulation of a magnetic read head. Testing of different coupling schemes lead to an implicit-explicit method, where the time-integration of the LLG with the highest order exchange term is handled implicitly, whereas the coupling to the Maxwell equations is handled explicitly. This scheme allows to reduce the computational costs of the multiscale method despite preserving the stability of the time-integration.

The methods implemented for the solution of the macroscopic scale Maxwell equations can also be used as a standalone solver. It is based on a FEM-BEM coupling methods, which allows to solve an open-boundary problem without having to mesh the whole outer region. Several versions of the solver using different physical descriptions (reduced-, total-, Mayergoyz-scalar potential) are selectable, which allows to choose the one which is best suited for the current problem. In order to improve the solvability of even badly conditioned systems, a proper preconditioner has been proposed and some globalization techniques

have been implemented on top of the basic solver. The use of \mathcal{H} -matrices, especially using their parallel implementation, allows to optimize the handling of the occurring BEM matrices and finally lead to a very stable, fast and easy to use method to solve nonlinear magnetostatic Maxwell equations.

Last but not least the handling of \mathcal{H} -matrices could be parallelized in a very transparent way. Since \mathcal{H} -matrices are used in very different topics this improvement impacts a wide area of applications. Considering micromagnetic simulations one of the largest contributions to the overall computation time could be eliminated.

7.2 Outlook

One possible extensions to the multiscale method presented in this work is to switch to dynamic Maxwell equations, instead of the magnetostatic ones. This would allow to include eddy current effects within the macroscopic parts. In contrast to microscopic dynamics, where the dynamic can mainly be described by the precessional motion and is only slightly modified by the influence of eddy currents, in the macroscopic scale eddy current effects are dominating. Including these into the multiscale method, would result in a reliable model for both, the microscopic and the macroscopic scale magnetization dynamic. Possible applications for such a method include the time-resolved simulation of magnetic write processes.

Another very interesting question is how models can be handled which cannot be split into separated regions. Within the connection region, where the microscopic and macroscopic parts touch each other, there needs to be a method which allows to describe the transition between the two models correctly. Such a method would make the multiscale algorithm more flexible in its use and also improves the overall performance since smaller microscopic regions may be selected.

Adding hysteresis as well as anisotropy to the macroscopic material law, would make the multiscale method interesting for the field of application of

permanent magnets. Here the multiscale method can be useful to combine the description of the micromagnetic behavior of some grains with the macroscopic behavior of the bulk magnet. Since materials used for permanent magnets need to show a large hysteresis, this would be an essential requirement for modeling of such problems.

A very nice feature would be to deriving macroscopic material parameters from micromagnetic simulations, which would assure a consistent descriptions of a material in the microscopic and macroscopic scale. By this way one can get along without the need for some experimental measurements.

From the implementational point of view it would be interesting to switch to higher order elements, which yields an improved accuracy when using the same number of unknowns. For sake of consistency one could also think about solving the LLG in its weak formulation using a FEM discretization. By this way the whole multiscale algorithm could be implemented on top of a finite element toolbox, which supports elements of arbitrary order.

Bibliography

- [1] J. Simkin and C. W. Trowbridge. On the use of the total scalar potential on the numerical solution of fields problems in electromagnetics. *Int. J. Numer. Meth. Eng.*, 14:423–440, 1979.
- [2] I. D. Mayergoyz, M. V. K. Chari, and J. D’Angelo. New scalar potential formulation for Three-Dimensional magnetostatic problems. *IEEE Trans. Magn.*, MAG-23:3889–3894, 1987.
- [3] F. Bruckner, C. Vogler, M. Feischl, D. Praetorius, B. Bergmair, T. Huber, M. Fuger, and D. Suess. 3D FEM-BEM-coupling method to solve magnetostatic maxwell equations. *J. Magn. Magn. Mater.*, 324:1862–1866, 2012.
- [4] F. Bruckner, M. Feischl, T. Führer, M. Page, D. Praetorius, and D. Suess. Combining micromagnetism and magnetostatic maxwell equations for multiscale magnetic simulations. *J. Magn. Magn. Mater.*, 2013 (in press).
- [5] F. Bruckner, M. Feischl, T. Führer, P. Goldenits, M. Page, D. Praetorius, and D. Suess. Multiscale modeling in micromagnetics: well-posedness and numerical integration. *arXiv:1209.5548*, 2012.
- [6] P. G. Ciarlet. *The Finite Element Method for Elliptic Problems*. North-Holland Publishing Company, Amsterdam, 1978.
- [7] S. Börm, L. Grasedyck, and W. Hackbusch. An introduction to hierarchical matrices. *Math. Bohem.*, 127:229–241, 2003.

- [8] S. Kurz, O. Rain, and S. Rjasanow. The adaptive cross-approximation technique for the 3D boundary-element method. *IEEE Trans. Magn.*, 38:421–424, 2002.
- [9] Steffen Börm and Lars Grasedyck. Hybrid cross approximation of integral operators. *Numer. Math.*, 101:221–249, 2005.
- [10] W. Hackbusch. *Hierarchische Matrizen: Algorithmen und Analysis*. Springer, Heidelberg, 2009.
- [11] R. S. Dembo, S. C. Eisenstat, and T. Steihaug. Inexact newton methods. *SIAM J. Numer. Anal.*, 19:400–408, 1982.
- [12] L. Fang, L. Sun, and G. He. An efficient newton-type method with fifth-order convergence for solving nonlinear equations. *Comp. Appl. Math*, 27, 2008.
- [13] A. Aharoni. *Introduction to the Theory of Ferromagnetism*. Oxford University Press, 2000.
- [14] C. Abert, L. Exl, G. Selke, A. Drews, and T. Schrefl. Numerical methods for the stray-field calculation: A comparison of recently developed algorithms. *J. Magn. Magn. Mater.*, 326:176–185, 2013.
- [15] W. F. Brown. *Micromagnetics*. Interscience Publishers, New York, 1963.
- [16] T. A. Gilbert. Armour research foundation report no. 11. *Armour Research Foundation*, Chicago, Illinois, 1955.
- [17] T. L. Gilbert. A lagrangian formulation of the gyromagnetic equation of the magnetization field. *Phys. Rev.*, 100:1243, 1955.
- [18] F. Alouges and A. Soyeur. On global weak solutions for landau-lifshitz equations: existence and nonuniqueness. *Nonlinear Anal.-Theor.*, 18:1071–1084, 1992.

- [19] F. Alouges. A new finite element scheme for landau-lifchitz equations. *Discrete Contin. Dyn. Syst. Ser. S*, 1:187–196, 2008.
- [20] P. Goldenits, D. Praetorius, and D. Suess. Convergent geometric integrator for the landau-lifshitz-gilbert equation in micromagnetics. *PAMM*, 11:775–776, 2011.
- [21] F. Alouges, E. Kritsikis, and J. C. Toussaint. A convergent finite element approximation for Landau-Lifschitz-Gilbert equation. *Physica B*, 407:1345–1349, 2012.
- [22] P. Goldenits, G. Hrkac, D. Praetorius, and D. Suess. An effective integrator for the Landau-Lifshitz-Gilbert equation. In *Proceedings of Mathmod 2012 Conference - to appear*.
- [23] G. T. Rado and J. R. Weertman. Spin-wave resonance in a ferromagnetic metal. *J. Phys. Chem. Solids*, 11:315–333, 1959.
- [24] M. Mansuripur and R. Giles. Demagnetizing field computation for dynamic simulation of the magnetization reversal process. *IEEE Trans. Magn.*, 24:2326–2328, 1988.
- [25] S.W. Yuan and H.N. Bertram. Fast adaptive algorithms for micromagnetics. *IEEE Trans. Magn.*, 28:2031–2036, 1992.
- [26] D. V. Berkov, K. Ramstöck, and A. Hubert. Solving micromagnetic problems. towards an optimal numerical method. *Phys. Status Solidi A*, 137:207–225, 1993.
- [27] H. H. Long, E. T. Ong, Z. J. Liu, and E. P. Li. Fast fourier transform on multipoles for rapid calculation of magnetostatic fields. *IEEE Trans. Magn.*, 42:295–300, 2006.
- [28] C. Abert, G. Selke, B. Kruger, and A. Drews. A fast Finite-Difference method for micromagnetics using the magnetic scalar potential. *IEEE Trans. Magn.*, 48:1105–1109, 2012.

- [29] A. V. Goncharov, G. Hrkac, J. S. Dean, and T. Schrefl. Kronecker product approximation of demagnetizing tensors for micromagnetics. *J. Comput. Phys.*, 229:2544–2549, 2010.
- [30] L. Exl, W. Auzinger, S. Bance, M. Gusenbauer, F. Reichel, and T. Schrefl. Fast stray field computation on tensor grids. *J. Comput. Phys.*, 231:2840–2850, 2012.
- [31] J. L. Blue and M. R. Scheinfein. Using multipoles decreases computation time for magnetostatic self-energy. *IEEE Trans. Magn.*, 27:4778–4780, 1991.
- [32] R. K. Beatson and L. Greengard. A short course on fast multipole methods. *Wavelets, multilevel methods and elliptic PDEs*, pages 1–37, 1997.
- [33] N. Popović and D. Praetorius. Applications of \mathcal{H} -Matrix techniques in micromagnetics. *Computing*, 74:177–204, 2005.
- [34] N. Popović and D. Praetorius. \mathcal{H} -matrix techniques for stray-field computations in computational micromagnetics. In *Large-Scale Scientific Computing*, number 3743 in Lecture Notes in Computer Science, pages 102–110. Springer Berlin Heidelberg, 2006.
- [35] A. Knittel, M. Franchin, G. Bordignon, T. Fischbacher, S. Bending, and H. Fangohr. Compression of boundary element matrix in micromagnetic simulations. *J. Appl. Phys.*, 105:07D542, 2009.
- [36] X. Brunotte, G. Meunier, and J. F. Imhoff. Finite element modeling of unbounded problems using transformations: A rigorous, powerful and easy solution. *IEEE Trans. Magn.*, 28:1663–1666, 1992.
- [37] P. Bettess. Infinite elements. *Int. J. Numer. Methods Eng.*, 11:53–64, 1977.
- [38] P. P. Silvester, D. A. Lowther, C. J. Carpenter, and E. A. Wyatt. Exterior finite elements for 2-Dimensional field problems with open boundaries. *Proc. IEE*, 124:1267–1270, 1977.

- [39] A. Khebir, A. B. Kouki, and R. Mittra. Asymptotic boundary conditions for finite element analysis of Three-Dimensional transmission line discontinuities. *IEEE Trans. Microwave Theory Tech.*, 38:1427–1432, 1990.
- [40] O. C. Zienkiewicz, D. W. Kelly, and P. Bettess. The coupling of the finite element method and boundary solution procedures. *Int. J. Numer. Meth. Eng.*, 11:355–375, 1977.
- [41] D. R. Fredkin and T. R. Koehler. Hybrid method for computing demagnetizing fields. *IEEE Trans. Magn.*, 26:415–417, 1990.
- [42] C. J. Garcia-Cervera and A. M. Roma. Adaptive mesh refinement for micromagnetics simulations. *IEEE Trans. Magn.*, 42:1648–1654, 2006.
- [43] C. Johnson and J. C. Nédélec. On the coupling of boundary integral and finite element methods. *Math. Comp.*, 35:1063–1079, 1980.
- [44] M. Costabel. *Symmetric methods for the coupling of finite elements and boundary elements*. Technische Hochschule Darmstadt, Fachbereich Mathematik, 1987.
- [45] M. Kuczmann. Potential formulations in magnetics applying the finite element method. *Lecture Notes, Laboratory of Electromagnetic Fields*, Széchenyi István University, Győr, Hungary, 2009.
- [46] D. A. Lindholm. Three-Dimensional magnetostatic fields from Point-Matched integral equations with linearly varying scalar sources. *IEEE Trans. Magn.*, MAG-20:2025–2032, 1984.
- [47] F. J. Sayas. The validity of Johnson-Nédélec’s BEM-FEM coupling on polygonal interfaces. *SIAM J. Numer. Anal.*, 47:3451–3463, 2009.
- [48] M. Aurada, M. Feischl, T. Führer, M. Karkulik, J. M. Melenk, and D. Praetorius. Classical FEM-BEM coupling methods: nonlinearities, well-posedness, and adaptivity. *Comput. Mech.*, 51:399–419, 2013.

- [49] W. Hackbusch. *Integral equations: theory and numerical treatment*. Birkhäuser, Basel, 1995.
- [50] S. A. Sauter and C. Schwab. Quadrature for hp-galerkin BEM in \mathbb{R}^3 . *Numer. Math.*, 78:211–258, 1997.
- [51] M. Rech. Effiziente Nahfeldkubatur in der galerkin-randelementmethode. *Diploma Thesis*, University of Bonn (in German language), 2002.
- [52] L. Grasedyck. Adaptive recompression of \mathcal{H} -matrices for BEM. *Computing*, 74:205–223, 2005.
- [53] A. C. Hindmarsh, P. N. Brown, K. E. Grant, S. L. Lee, R. Serban, D. E. Shumaker, and C. S. Woodward. SUNDIALS: suite of nonlinear and differential/algebraic equation solvers. *ACM Trans. Math. Softw.*, 31:363–396, 2005.
- [54] R. Barrett. *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods*. SIAM, New York, 1994.
- [55] J. E. Dennis and R. B. Schnabel. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. SIAM, New York, 1983.
- [56] A. Collier, A. Hindmarsh, R. Serban, and C. Woodward. User documentation for kinsol v2.7.0, 2011.
- [57] I. G. Graham and W. McLean. Anisotropic mesh refinement: The conditioning of galerkin boundary element matrices and simple preconditioners. *SIAM J. Numer. Anal.*, 44:1487–1513, 2006.
- [58] Team problem 13, 3-d non-linear magnetostatic model, <http://www.compumag.org/jsite/images/stories/TEAM/problem13.pdf>, 1997.
- [59] T. Nakata and Fujiwara K. Summary of results for benchmark problem 13 (3-d nonlinear magnetostatic model). *Int. J. Comput. Math. Electr. Electron. Eng.*, 11:345–369, 1992.

- [60] A. Vansteenkiste and B. Van de Wiele. MuMax: a new high-performance micromagnetic simulation tool. *J. Magn. Magn. Mater.*, 323:2585–2591, 2011.
- [61] S. Li, B. Livshitz, and V. Lomakin. Graphics processing unit accelerated $O(N)$ micromagnetic solver. *IEEE Trans. Magn.*, 46:2373–2375, 2010.
- [62] A. Prohl. *Computational Micromagnetism*. Teubner B.G. GmbH, Leipzig, 2001.
- [63] M. Kružík and A. Prohl. Recent developments in the modeling, analysis, and numerics of ferromagnetism. *SIAM Rev.*, 48:439–483, 2006.
- [64] I. Cimrák. A survey on the numerics and computations for the Landau-Lifshitz equation of micromagnetism. *Arch. Comput. Method. E.*, 15:1–37, 2007.
- [65] C. J. García-Cervera. Numerical micromagnetics: A review. *Boletín SEMA*, 2008.
- [66] X. P. Wang, C. J. García-Cervera, and E. Weinan. A Gauss-Seidel projection method for micromagnetics simulations. *J. Comput. Phys.*, 171:357–372, 2001.
- [67] D. Suess, V. Tsiantos, T. Schrefl, J. Fidler, W. Scholz, H. Forster, R. Dittrich, and J. J. Miles. Time resolved micromagnetics using a preconditioned time integration method. *J. Magn. Magn. Mater.*, 248:298–311, 2002.
- [68] M. d’Aquino, C. Serpico, and G. Miano. Geometrical integration of Landau-Lifshitz-Gilbert equation based on the mid-point rule. *J. Comput. Phys.*, 209:730–753, 2005.
- [69] S. Bartels and A. Prohl. Convergence of an implicit finite element method for the Landau-Lifshitz-Gilbert equation. *SIAM J. Numer. Anal.*, 44:1405–1419, 2006.

- [70] L. Bañas, M. Page, and D. Praetorius. A convergent linear finite element scheme for the maxwell-landau-lifshitz-gilbert equation. *arXiv:1303.4009*, 2013.
- [71] Y. Kanai, M. Saiki, K. Hirasawa, T. Tsukamomo, and K. Yoshida. Landau-Lifshitz-Gilbert micromagnetic analysis of Single-Pole-Type write head for perpendicular magnetic recording using Full-FFT program on PC cluster system. *IEEE Trans. Magn.*, 44:1602–1605, 2008.
- [72] B. Livshitz, A. Boag, N. H. Bertram, and V. Lomakin. Nonuniform grid algorithm for fast calculation of magnetostatic interactions in micromagnetics. *J. Appl. Phys.*, 105:07D541, 2009.
- [73] H. Forster, T. Schrefl, R. Dittrich, W. Scholz, and J. Fidler. Fast boundary methods for magnetostatic interactions in micromagnetics. *IEEE Trans. Magn.*, 39:2513–2515, 2003.
- [74] M. Slodička and I. Cimrák. Numerical study of nonlinear ferromagnetic materials. *Appl. Numer. Math.*, 46:95–111, 2003.
- [75] O. Vacus and N. Vukadinovic. Dynamic susceptibility computations for thin magnetic films. *J. Comput. Appl. Math.*, 176:263–281, 2005.
- [76] L. Bañas, S. Bartels, and A. Prohl. A convergent implicit finite element discretization of the Maxwell-Landau-Lifshitz-Gilbert equation. *SIAM J. Numer. Anal.*, 46:1399–1422, 2008.
- [77] M. Aziz. Sub-nanosecond electromagnetic-micromagnetic dynamic simulations using the finite-difference time-domain method. *PIER B*, 15:1–29, 2009.
- [78] K.-N. Le and T. Tran. A convergent finite element approximation for the quasi-static maxwell-landau-lifshitz-gilbert equations. *arXiv:1212.3369*, 2012.

- [79] L. Bañas and M. Slodička. Space discretization for the Landau-Lifshitz-Gilbert equation with magnetostriction. *Comput. Method. Appl. M.*, 194:467–477, 2005.
- [80] L. Bañas, M. Page, D. Praetorius, and J. Rochat. On the landau-lifshitz-gilbert equation with magnetostriction. *arXiv:1303.4060*, 2013.
- [81] L. R. Petzold. Differential/algebraic equations are not ODEs. *SIAM J. Sci. Stat. Comput.*, 3:367–384, 1982.
- [82] J. C. Mallinson. Damped gyromagnetic switching. *IEEE Trans. Magn.*, 36:1976–1981, July 2000.
- [83] G. Bertotti, I. Mayergoyz, C. Serpico, and M. Dimian. Comparison of analytical solutions of Landau-Lifshitz equation for damping and precessional switchings. *J. Appl. Phys.*, 93:6811–6813, May 2003.
- [84] S. X. She, D. Wei, Y. Zheng, B. J. Qu, T. L. Ren, X. Liu, and F. L. Wei. Micromagnetic simulation of transfer curve in giant-magnetoresistive head. *Chinese Phys. Lett.*, 26, 2009.
- [85] M. Bebendorf and R. Kriemann. Fast parallel solution of boundary integral equations and related problems. *Comput. Vis. Sci.*, 8:121–135, 2005.
- [86] R. Kriemann. Parallel \mathcal{H} -matrix arithmetics on shared memory systems. *Computing*, 74:273–297, 2005.
- [87] L. Dagum and R. Menon. OpenMP: an industry standard API for shared-memory programming. *IEEE IEEE Comput. Sci. Eng.*, 5:46–55, 1998.
- [88] R. Chandra. *Parallel Programming in Open Mp*. Morgan Kaufmann Publishers, San Francisco, 2001.
- [89] J. D. McCalpin. The stream benchmark. <http://www.streambench.org/>, 2013. [Online; accessed 25-Mar-2013].

Curriculum Vitae

Name: Florian BRUCKNER
Date of birth: 17.09.1983 in Waidhofen / Thaya, Austria
Nationality: Austria

Education

2010 – 2013	Vienna University of Technology, Austria Doktoratsstudium der technischen Wissenschaften (Doctoral Program in Engineering Sciences)
2004 – 2010	Vienna University of Technology, Austria Technische Physik, Dipl.-Ing. (Physics, M.Sc.) Thesis: <i>Stabilized source of entangled photon pairs characterized by quantum state tomography</i>
1998 – 2003	Senior High School: HTL Hollabrunn - Abteilung Steuerungs- und Regelungstechnik, Polytechnic for Control Engineering
1994 – 1998	Junior High School: Bundesrealgymnasium Waidhofen / Thaya
1990 – 1994	Elementary School: Volksschule Waidhofen / Thaya