Unterschrift Betreuer

TECHNISCHE UNIVERSITÄT WIEN

Diplomarbeit

Atomistic Spin Dynamics

Ausgeführt am Institut für Festkörperphysik
der Technischen Universität Wien

unter der Anleitung von

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

und

Dr. Claas Willem Abert

durch

Paul Thomas Heistracher
Schönbrunnerstraße 77/8, 1050 Wien

Wien, 12. September 2017

Unterschrift Student

Abstract

Within the scope of this thesis a GPU-accelerated software for the simulation of spin dynamics is presented. The program features a solver for the Landau–Lifshitz–Gilbert equation for an atomistic spin model considering dipole-dipole, exchange, anisotropy and Dzyaloshinskii-Moriya interactions thereby allowing for the investigation of magnetic skyrmions. By applying a continuum assumption, a micromagnetic description is derived and also implemented in the software. The discrete numerical methods needed for the solution of these models are discussed with a focus on finite difference methods, fast convolutions and adaptive time integration techniques. Furthermore, an implementation of the string method, a numerical method for the calculation of minimum energy paths in barrier-crossing events, is presented and used to investigate the annihilation energy of magnetic skyrmions.

Kurzfassung

Im Rahmen dieser Arbeit wird eine GPU-beschleunigte Software zur Simulation von Spindynamik vorgestellt. Das Programm enthält einen Solver für die Landau-Lifshitz-Gilbert-Gleichung für ein atomistisches Spinmodell, das Dipol-Dipol-, Austausch-, Anisotropie- und Dzyaloshinskii-Moriya-Wechselwirkungen berücksichtigt und damit die Untersuchung magnetischer Skyrmionen ermöglicht. Weiters wird mittels einer Kontinuumsannahme des Magnetfeldes das mikromagnetische Modell eingeführt und in der Software implementiert. Die numerischen Methoden, die zur Lösung dieser Modelle benötigt werden, werden mit einem Schwerpunkt auf Finite-Differenzen-Methoden, Faltungen mittels Fourier-Transformationen und adaptive Zeitintegrationstechniken präsentiert. Darüber hinaus wird eine Implementierung der Stringmethode vorgestellt und zur Untersuchung der Annihilationsenergie magnetischer Skyrmionen verwendet.

Table of Contents

1 Introduction

Magnetic skyrmions have recently attracted a lot of attention in the scientific community as a potential candidate for high density storage and logic devices [1–4]. The term skyrmion is named after Tony Skyrme, who introduced a mathematical description for topological stable field configurations which exhibit particle-like solutions [5]. This concept applies to various mathematical objects in different fields of research ranging from elementary particles to liquid crystals, Bose–Einstein condensates and quantum Hall magnets [1]. Magnetic skyrmions are nano-scale chiral spin structures which cannot be continuously deformed to other magnetic configurations such as spin spirals or ferromagnetic states [1]. This property is referred to as topological protection and gives rise to the comparably high stability of these particle-like spin configurations. In the current notion, the characteristic structure originates from chiral interactions, also known as Dzyaloshinskii–Moriya (DM) interactions, which arise from spin-orbit scattering of electrons in an inversion-asymmetric crystal field [6]. This effect was initially observed in bulk non-centrosymmetric crystals such as MnSi, (FeCo)Si and FeGe where extended lattices of magnetic skyrmions were found [7, 8]. These lattices can be moved by small electric currents in the order of $1 \times 10^2\,\mathrm{Acm}^{-2}$ [9] and corresponding simulations indicate that the flexibility of skyrmion lattices allows them to overcome defects more easily compared to the current-induced motion of domain walls [9]. Extremely small sized skyrmion lattices, with periodicity in the nano-meter range, have been observed using spin-polarized scanning tunnelling microscopy in a monolayer of Fe on Ir(1,1,1) in the absence of external magnetic fields [10]. The high skyrmion density is explained by very strong DM interactions at the Fe/Ir interface.

More recently, increasing attention has also been devoted to isolated skyrmions [1–3]. These are single skyrmions or chains of skyrmions which can be nucleated as a metastable state in thin films. The controlled manipulation of these isolated skyrmions is opening a path to new concepts of spintronic devices such as information storage or logic devices. Numerical micromagnetic investigations predicted that an isolated skyrmion can be a stable configuration in a nanostructure, that it can be locally nucleated by injection of a spin-polarised current and that it can be displaced by current-induced spin torques, even in the presence of large defects [2]. Accordingly, subsequent experiments show the feasibility to write and erase such spin textures in a controlled fashion using local spin-polarized currents from a scanning tunnel microscope [3]. The controlled creation and annihilation of isolated magnetic skyrmions demonstrates the applicability of topological charge as a carrier of information and its potential for future storage-concepts. In the experimental set up, an external magnetic field is used in order to tune the energy landscape and the temperature is controlled to prevent thermally-activated switching between topologically distinct states. Under this conditions, the switching rate and

direction can then be controlled using the injected current [3]. This suggests isolated magnetic skyrmions as promising candidates for future storage devices due to their size of only several nanometres and the low current densities needed for their displacement.

Even though micromagnetic simulations are capable of describing skyrmion configurations, they cannot account for skyrmion creation or annihilation processes mediated by Bloch points. As mentioned above, a skyrmionic structure cannot be continuously deformed to a ferromagnetic or other non-skyrmionic magnetic state, giving rise to its topological protection. This is in contradiction with the main assumption of micromagnetics, which assumes that the magnetic field can be approximated as a continuous field. Therefore, we consider an atomistic spin model to account for such skyrmion creation and annihilation processes.

Within this thesis, we present the simulation software *pth-mag* which allows for the description of magnetic skyrmions in the framework of an atomistic spin model. This model assumes discrete atomic spins and can thereby account for skyrmion annihilation processes. The structure of the thesis is as follows:

In chapter 2 we discuss the physical models in order to describe the dynamics of such magnetic skyrmions. At first, we introduce an atomistic spin model in section 2.1. The interactions considered in this model are dipole-dipole, exchange, anisotropy and antisymmetric exchange or Dzyaloshinskii-Moriya interactions. In section 2.2 we introduce the Landau–Lifshitz– Gilbert equation which describes the time-dependent precessional motion of the magnetisation. In the following, we apply a continuum assumption onto the magnetic field and derive the micromagnetic model with demagnetisation, exchange, anisotropy and antisymmetric exchange interactions in section 2.3. In Chapter 3 we discuss discrete mathematical concepts that are applied in the simulation in order to solve for the atomistic and micromagnetic models. We present finite-difference methods for the numerical solution of differential equations and highlight the applicability of convolutions in this framework, pointing out the efficiency of Fourier-space methods. Moreover, we discuss explicit time integrators starting with the straightforward Euler method. We generalize this concept to explicit Runge-Kutta methods and present a variety of coefficient tables in the following. By the usage of a embedded Runge-Kutta method, which features two different orders of the function approximation, an adaptive step-size control algorithm is introduced. In section 3.4 we present the *string method*, a numerical method to identify minimum energy paths in barrier-crossing events. Several aspects of the implementation of the simulation software are discussed in chapter 4. We highlight the applicability of parallelisation onto this mathematical framework and discuss the advantages of the used GPU-accelerated high-performance mathematical library, emphasizing the hardware neutrality of the code. This allows users, depending on their available hardware, to execute the program either on GPUs, OpenCL devices or CPUs with the former yielding the highest performance. We perform simulations of the μMAG standard problem #4 with and compare the results to measurement data of the well established code magnum.fd and find very good agreement. In chapter 5 we apply the string method to calculate the annihilation energy of magnetic skyrmions in an atomistic model, show the inapplicability of the micromagnetic model in this scope and present various simulation results which are in accordance with reference calculations.

2 Magnetism

The phenomenon of magnetism is physically considered as a purely quantum mechanical effect [11]. As stated by the *Bohr–van Leeuwen theorem*, a classical theory is not capable of describing diamagnetism, paramagnetism or ferromagentism. The theorem consistently applies classical mechanics with statistical mechanics and shows that in this model the thermal average of the magnetisation is always zero. As a consequence, magnetism can only be approached quantum mechanically.

One of the first atomistic models of magnetic materials was introduced by Ising in 1925 [12]. In this model the atoms possess local magnetic moments along a fixed quantization axis which are in two states only, either spin-up or spin-down. This approach allows for an analytical description in one and two dimensions. While the Ising model is capable of describing phase transitions, it is limited in applicability to magnetic materials and cannot describe dynamical processes.

A prominent extension of the Ising model – where all spins are orientated along a specified axis – is the classical Heisenberg model [13]. It allows the atomic spins to be orientated along an arbitrary axis in 3D but still neglects quantum mechanical effects on the atomic spins. An approach by Monte Carlo methods allows the description of phase transitions, surface and finite size effects in magnetic systems. The use of Monte Carlo methods intrinsically limited the study of dynamic phenomena until the development of dynamic [14, 15] and stochastic [16] Landau–Lifshitz–Gilbert atomistic spin models. A major achievement of the atomistic spin model is its capability to close the gap between *ab initio* electronic structure calculations and micromagnetics by using a multi-scale model, e.g. as discussed in [17]. Another strength of the atomistic model is its ability to directly interface with the micromagnetic model allowing to simulate large systems by treating surface effects atomistically while calculating the bulk with a coarse micromagnetic discretization [18].

2.1 Atomistic Spin Model

In this section we focus on the phenomenon of ferromagnetism and introduce an atomistic spin model based on [19, 20]. The basic assumption of the model is the localization of unpaired electrons to atomic sites which is expressed as an effective local atomistic magnetic moment. The degree of localization of electrons in 3d metals has historically been debated as magnetic effects arise from electrons which are "loosely bound" to the atoms. However, *ab initio* calculations of the electron density show that even for ferromagnets with itinerant magnetic moments – that are magnets modelled with nearly free electrons which are shared amongst the entire solid in an electron gas – the spin polar-

ization is well-localized to the atomic sites. This indicates that the bonding electrons are unpolarized and when we account for the bonding charge, the remaining d-electrons form a well-defined localized spin moment on the atomic site [19].

As mentioned above, magnetic systems are described fundamentally quantum- mechanically The electron energy levels are quantized and the electrons are subject to the quantum mechanical effect of exchange interaction. This effect arises due to the quantum mechanical behaviour of spins and the implications of the Pauli exclusion principle [11, ch. 37]. The exchange interaction is responsible for ferromagnetic alignment and favours neighbouring spins to align in parallel. We expand the model by considering further interactions such as the dipole-dipole interaction of the spins. This interaction has the tendency to reduce the total magnetic moment and is thus referred to as *demagnetisation* [21]. The geometry of the interacting dipoles affects the resulting field which then features a preferred direction of alignment. This gives rise to the effect of *shape anisotropy* which causes, for example, spacial anisotropy in ferromagnets with a single magnetic domain and is responsible for the formation of magnetic domains in larger ferromagnets. Another effect considered is magnetic anisotropy which is caused by spin-orbit coupling. We discuss the uniaxial anisotropy where magnetically anisotropic materials exhibit an energetically favourable axis of spontaneous magnetisation and the magnetic moment tends to align with this axis. There exist other types of anisotropy such as cubic anisotropy, for example. We further include antisymmetric exchange in the model. This interaction prefers spin canting (i.e. the spins are tilted by a small angle) in contrast to the usual parallel alignment and also arises from spin-orbit coupling. Furthermore, we allow the system to be affected by an applied field by introducing the Zeeman interaction. The field contributions arising from these interactions allow us to accurately describe the dynamics of micron-scale ferromagnetic systems by introducing the Landau–Lifshitz–Gilbert equation in section 2.2. At the end of this chapter, we derive the well established micromagnetic model in section 2.3 by applying a continuum assumption onto the atomistic model.

2.1.1 Interaction Hamiltonians

All these interactions can be treated by the extended Heisenberg spin model considering a classical spin Hamiltonian which takes the form

$$\mathcal{H} = \mathcal{H}_{\text{exc}} + \mathcal{H}_{\text{dip}} + \mathcal{H}_{\text{ani}} + \mathcal{H}_{\text{DM}} + \mathcal{H}_{\text{zee}}, \tag{2.1}$$

denoting terms for the exchange interaction, dipole-dipole interaction, magneto-crystalline anisotropy, asymmetric exchange or Dzyaloshinskii-Moriya (DM) interaction and Zeeman interaction.

Exchange Interaction

The Heisenberg exchange energy is the dominant term in the spin Hamiltonian of equation (2.1) and is a consequence of the symmetry of the electron wave-function and the

Pauli exclusion principle, which determines the orientation of electronic spins in overlapping electron orbitals. The exchange interaction has an electrostatic origin leading to energies in the range of 1–2 eV. This energy range is typically around 1000 times larger than the next largest contribution to the Hamiltonian and causes magnetic ordering temperatures between 300–1300 K.

In the following we discuss the exchange energy for a system of interacting atomic moments. We start by considering two non-interacting electrons. The Hamiltonian of the system is given by

$$\hat{\mathcal{H}} = -J\hat{\boldsymbol{S}}_1 \cdot \hat{\boldsymbol{S}}_2 \tag{2.2}$$

where J denotes the exchange integral, $\hat{\boldsymbol{S}}_1$ and $\hat{\boldsymbol{S}}_2$ denote the spin operators of the first and second electron, respectively.

This result can now be generalized to the quantum mechanical Heisenberg model

$$\hat{\mathcal{H}}_{\text{exc}} = -\frac{1}{2}\sum_{i\neq j} J_{ij}\hat{\boldsymbol{S}}_i \cdot \hat{\boldsymbol{S}}_j \tag{2.3}$$

where J_{ij} are exchange integrals between atoms at lattice sites i and j, $\hat{\boldsymbol{S}}_i$ is the spin operator of the atom at site i and the factor $\frac{1}{2}$ accounts for double counting. The sign of the exchange integral J_{ij} determines whether the preferred spin alignment is parallel ($J_{ij} > 0$) or antiparallel ($J_{ij} < 0$).

This quantum mechanical expression can be approximated by a classical model when applying the *atomic sphere approximation* (ASA). The atomic spin operator's expectation values are decoupled by assuming

$$\langle\hat{\mathcal{H}}_{\text{exc}}\rangle = -\frac{1}{2}\sum_{i\neq j} J_{ij}\langle\hat{\boldsymbol{S}}_i\hat{\boldsymbol{S}}_j\rangle \approx -\frac{1}{2}\sum_{i\neq j} J_{ij}\langle\hat{\boldsymbol{S}}_i\rangle \cdot \langle\hat{\boldsymbol{S}}_j\rangle \tag{2.4}$$

.

We further can consider the adiabatic approximation in which we assume that the electrons, whose dynamics take place in the femtosecond regime, almost instantaneously adapt to the dynamics of the spins which take place in the nanosecond regime. From this it follows that the values of the exchange integrals stay constant in time and we thus obtain the classical Heisenberg exchange model

$$\mathcal{H}_{\text{exc}} = -\frac{1}{2}\sum_{i\neq j} J_{ij}\boldsymbol{s}_i \cdot \boldsymbol{s}_j \tag{2.5}$$

where $\boldsymbol{s}_i$ denotes the classical spin direction on site i with $\|\boldsymbol{s}_i\| = 1$ (i.e. the unit vector pointing in the direction of the total angular momentum).

The exchange interaction is highly dependent on the distance and in most cases the sum in equation (2.5) can be modified to include nearest neighbours only. We further introduce the multi-indices $\boldsymbol{i}$ and $\boldsymbol{j}$ with $\boldsymbol{i} = (x_i, y_i, z_i)$ and $\boldsymbol{j} = (x_j, y_j, z_j)$ for a more compact notation. So we finally obtain

$$\mathcal{H}_{\text{exc}} = -\frac{J}{2}\sum_{<\boldsymbol{ij}>} \boldsymbol{s_i} \cdot \boldsymbol{s_j} \tag{2.6}$$

where $< \boldsymbol{ij} >$ restricts the sum to nearest neighbours.

While the exchange interaction is responsible for magnetic ordering at the atomic level, the thermal stability of a magnet is mainly caused by magnetic anisotropy.

Anisotropy

The magnetic anisotropy describes the tendency of the spin moments to align along particular spacial directions. Several physical effects give rise to magnetic anisotropy with the most important being magneto-crystalline anisotropy. This effect describes the preference of the atomic moments to align with a particular crystallographic axis and arises from the interaction of atomic electron orbitals with the local crystal structure. Opposed to exchange interaction, the effect of anisotropy is of relativistic origin and significantly smaller.

A common form of anisotropy is single-ion uniaxial anisotropy where the spins tend to align along a single axis denoted as $\boldsymbol{e}$ and often referred to as the easy axis of the system. Uniaxial anisotropy is the simplest form of anisotropy and is commonly found in materials where the crystal lattice is distorted along a single axis such as hexagonal Cobalt or $L1_0$ ordered FePt [19].

The classical Hamiltonian for uniaxial anisotropy reads

$$\mathcal{H}_{\text{ani}} = -k \sum_{\boldsymbol{i}} (\boldsymbol{s_i} \cdot \boldsymbol{e}_{\text{ani}})^2, \tag{2.7}$$

where k is the anisotropy energy per atom in units of Joule.

Dipole-Dipole Interaction

The dipole-dipole interaction, or demagnetisation, results from the magneto-static interaction of the atomic spins. It has the tendency to reduce the total magnetic moment of the system and gives rise to shape anisotropy in ferromagnets with a single magnetic domain or the formation of magnetic domains in lager ferromagnets.

The dipole-dipole interaction yields a Hamiltonian of the form

$$\mathcal{H}_{\text{dip}} = -\frac{\mu_0 \mu_s}{2} \sum_{\boldsymbol{i}} \boldsymbol{H}_{\text{dip}}(\boldsymbol{s_i}) \cdot \boldsymbol{s_i}, \tag{2.8}$$

where μ_0 is the vacuum permeability, μ_s the magnitude of the atomistic magnetic moment and $\boldsymbol{H}_{\text{dip}}$ is the dipole field of equation (2.18) which is discussed in detail in the following sections.

Antisymmetric Exchange

The antisymmetric exchange is an interaction which favours spin canting rather than (anti)parallel spin alignment. Spin canting means that the spins tend to tile by a small angle about their axis opposed to being aligned perfectly (anti)parallel. This interaction was first postulated by Igor Dzyaloshinskii by phenomenological considerations of the

Landau theory [22]. Toru Moriya then discovered the spin-orbit coupling to be the microscopic mechanism of the antisymmetric exchange [23] and thus the interaction is often referred to as Dzyaloshinskii-Moriya interaction (DMI).

Experimentally, large values of antisymmetric exchange are encountered at interfaces between a magnetic film and a heavy metal with high spin-orbit coupling [24, 25].

In an atomic description the DMI is written as

$$\mathcal{H}_{DM} = \sum_{<\boldsymbol{ij}>} \boldsymbol{d_{ij}} \cdot (\boldsymbol{s_i} \times \boldsymbol{s_j}), \tag{2.9}$$

where $\boldsymbol{d}_{ij}$ is the DM interaction vector between atomic sites i and j in Joule, $\boldsymbol{u_i}$ is the atomic moment unit vector at site i (which yields $\boldsymbol{u_i} = \frac{\boldsymbol{s_i}}{\mu_0 \mu_s}$) and the summation is performed over nearest neighbours $< \boldsymbol{ij} >$. The direction of $\boldsymbol{d}_{ij}$ is dependent on the system under consideration. In the case of magnetic ultra-thin films, DMI is caused by the interaction with the high spin-orbit coupled heavy metal of the adjacent layer and the DM interaction vector $\boldsymbol{d_{ij}}$ can be written as $d\, \boldsymbol{u_{ij}} \times \boldsymbol{e}_{\mathrm{DM}}$, where $\boldsymbol{u_{ij}}$ is the unit vector from site $\boldsymbol{i}$ to site $\boldsymbol{j}$ and $\boldsymbol{e}_{\mathrm{DM}}$ is the unit vector normal to the film and d is the atomistic antisymmetric exchange constant in Joule.

In this case the Hamiltonian takes the form of

$$\mathcal{H}_{DM} = d \sum_{<\boldsymbol{ij}>} (\boldsymbol{u_{ij}} \times \boldsymbol{e}_{DM}) \cdot (\boldsymbol{s_i} \times \boldsymbol{s_j}). \tag{2.10}$$

Zeeman Interaction

The influence of an external applied field $\boldsymbol{H}_{\mathrm{zee}}$, which may arise due to an electric field from an electric current or nearby magnetic materials, is considered in the classical Hamiltonian by

$$\mathcal{H}_{\mathrm{zee}} = -\mu_0 \mu_s \sum_{\boldsymbol{i}} \boldsymbol{s_i} \cdot \boldsymbol{H^i}_{\mathrm{zee}}, \tag{2.11}$$

where $\boldsymbol{H^i}_{\mathrm{zee}}$ denotes the external field at site i.

2.1.2 Effective Field Contributions

In the previous section we discussed the contributions to the classical Heisenberg Hamiltonian for the atomistic spin model. However, when it comes to the description of dynamics of the system, we are interested in effective fields generated by these interactions. These field contributions sum up to a net effective field $\boldsymbol{H}_{\mathrm{eff}}$ which is then plugged into the Landau-Lifshitz-Gilbert equation (2.27) to describe the time evolution of a spin system.

The effective field at the atomic site $\boldsymbol{i}$ can be calculated using the corresponding Hamiltonian by

$$\boldsymbol{H^i}_{\mathrm{eff}} = -\frac{1}{\mu_0 \mu_s} \frac{\partial \mathcal{H}}{\partial \boldsymbol{s_i}}, \tag{2.12}$$

where $\mathcal{H}$ is the classical Heisenberg Hamiltonian of equation (2.1) which is considering all interactions. The effective field can also be split up into contributions arising from each type of interaction and thus we can write

$$\boldsymbol{H}_{\text{eff}} = \boldsymbol{H}_{\text{exc}} + \boldsymbol{H}_{\text{ani}} + \boldsymbol{H}_{\text{dip}} + \boldsymbol{H}_{\text{ant}} + \boldsymbol{H}_{\text{zee}}. \tag{2.13}$$

In the following, we use equation (2.12) to derive the field terms of each interaction introduced above.

Exchange Field

As of the exchange interaction, the calculation of the exchange field yields

$$\boldsymbol{H^i}_{\text{exc}} = -\frac{1}{\mu_0\mu_s}\frac{\partial\mathcal{H}_{\text{exc}}}{\partial\boldsymbol{s_i}} = -\frac{1}{\mu_0\mu_s}\frac{\partial}{\partial\boldsymbol{s_i}}\Big[-\frac{J}{2}\sum_{<\boldsymbol{jk}>}\boldsymbol{s_j}\cdot\boldsymbol{s}_k\Big] = \frac{J}{\mu_0\mu_s}\sum_{<\boldsymbol{j}>_i}\boldsymbol{s_j}, \tag{2.14}$$

where $<\boldsymbol{j}>_{\boldsymbol{i}}$ denotes the sum over nearest neighbours at site $\boldsymbol{i}$. In the last step a factor of 2 arises due to double counting.

In many cases we have already calculated the field and want to determine its corresponding energy. In the case of the exchange interaction, we can calculate the exchange energy of a given exchange field by

$$\mathcal{H}_{\text{exc}} = -\frac{\mu_0\mu_s}{2}\sum_{\boldsymbol{i}}\boldsymbol{H^i}_{\text{exc}}\boldsymbol{s_i}. \tag{2.15}$$

Anisotropy Field

Considering the field contribution of the anisotropy we obtain

$$\boldsymbol{H^i}_{\text{ani}} = -\frac{1}{\mu_0\mu_s}\frac{\partial\mathcal{H}_{\text{ani}}}{\partial\boldsymbol{s_i}} = -\frac{1}{\mu_0\mu_s}\frac{\partial}{\partial\boldsymbol{s_i}}\Big[-k\sum_{\boldsymbol{i}}(\boldsymbol{s_i}\cdot\boldsymbol{e}_{\text{ani}})^2\Big] = \frac{2k}{\mu_0\mu_s}(\boldsymbol{s_i}\cdot\boldsymbol{e}_{\text{ani}})\boldsymbol{e}_{\text{ani}}. \tag{2.16}$$

The corresponding energy is given by

$$\mathcal{H}_{\text{ani}} = -\frac{\mu_0\mu_s}{2}\sum_{\boldsymbol{i}}\boldsymbol{H}^i_{\text{ani}}\boldsymbol{s_i}. \tag{2.17}$$

Dipole-Dipole Field

In the case of the dipole-dipole interaction, we first define the dipole-dipole field which then yields the Hamiltonian. The dipole-dipole field for a certain atomic site i describes the interaction of this spin with each other spin in the entire system an thus is a so-called global interaction. The field is given by

$$\boldsymbol{H^i}_{\text{dip}} = \mu_s\sum_{\boldsymbol{j}}\tilde{\boldsymbol{D}}_{\boldsymbol{i-j}}\boldsymbol{s_j}, \tag{2.18}$$

where $\tilde{\boldsymbol{D}}_{\boldsymbol{i}-\boldsymbol{j}}$ is the dipole-dipole tensor for the atomic sites $\boldsymbol{i}$ and $\boldsymbol{j}$.

In a classical description of the magnetic pole limit, we can define a scalar potential of the magnetic dipole by

$$\psi(\boldsymbol{d}, \boldsymbol{r}) = \frac{\boldsymbol{d} \cdot \boldsymbol{r}}{4\pi r^3}, \tag{2.19}$$

where $\boldsymbol{d}$ is the magnetic moment of the dipole and $\boldsymbol{r}$ is denoting the spacial vector with components r_x, r_y, r_z and magnitude r.

From this, we obtain the dipole field by taking the negative gradient and which reads [26]

$$\boldsymbol{H}_{\text{dip}}^{\text{classic}}(\boldsymbol{d}, \boldsymbol{r}) = -\nabla\psi = \frac{1}{4\pi}\left(\frac{3\boldsymbol{r}(\boldsymbol{d}\cdot\boldsymbol{r})}{r^5} - \frac{\boldsymbol{d}}{r^3}\right) = \frac{1}{4\pi}\begin{pmatrix} \frac{3r_x^2}{r^5} - \frac{1}{r^3} & \frac{3r_x r_y}{r^5} & \frac{3r_x r_z}{r^5} \\ \frac{3r_x r_y}{r^5} & \frac{3r_y^2}{r^5} - \frac{1}{r^3} & \frac{3r_x r_z}{r^5} \\ \frac{3r_x r_z}{r^5} & \frac{3r_y r_z}{r^5} & \frac{3r_z^2}{r^5} - \frac{1}{r^3} \end{pmatrix} \cdot \begin{pmatrix} d_x \\ d_y \\ d_z \end{pmatrix}. \tag{2.20}$$

We identify the dipole-dipole tensor with the matrix of equation (2.20) assuming the vector $\boldsymbol{r} = \boldsymbol{i} - \boldsymbol{j}$ points from atomic site i to j and find:

$$\tilde{\boldsymbol{D}}_{\boldsymbol{i}-\boldsymbol{j}} = \frac{1}{4\pi}\begin{pmatrix} \frac{3r_x^2}{r^5} - \frac{1}{r^3} & \frac{3r_x r_y}{r^5} & \frac{3r_x r_z}{r^5} \\ \frac{3r_x r_y}{r^5} & \frac{3r_y^2}{r^5} - \frac{1}{r^3} & \frac{3r_x r_z}{r^5} \\ \frac{3r_x r_z}{r^5} & \frac{3r_y r_z}{r^5} & \frac{3r_z^2}{r^5} - \frac{1}{r^3} \end{pmatrix} \tag{2.21}$$

Due to the long-range character of this interaction each spin interacts with every other spin in the entire system leading to a high computational complexity. In chapter 3 we discuss an efficient way to compute the dipole-dipole field.

The corresponding energy can be calculated by

$$\mathcal{H}_{\text{dip}} = -\frac{\mu_0\mu_s}{2}\sum_{\boldsymbol{i}} \boldsymbol{H}_{\text{dip}}^{i}\boldsymbol{s_i}. \tag{2.22}$$

Antisymmetric Field

We obtain the field contribution of the antisymmetric exchange in the same way by considering

$$\boldsymbol{H^i}_{\text{DM}} = -\frac{1}{\mu_0\mu_s}\frac{\partial\mathcal{H}_{\text{DM}}}{\partial\boldsymbol{s_i}} = -\frac{1}{\mu_0\mu_s}\frac{\partial}{\partial\boldsymbol{s_i}}\Big[d\sum_{<ij>}(\boldsymbol{u}_{ij}\times\boldsymbol{e}_{DM})\cdot(\boldsymbol{s_i}\times\boldsymbol{s}_j)\Big]. \tag{2.23}$$

Again, the corresponding energy is calculated with

$$\mathcal{H}_{\text{DM}} = -\frac{\mu_0\mu_s}{2}\sum_{\boldsymbol{i}} \boldsymbol{H^i}_{\text{DM}}\boldsymbol{s_i}. \tag{2.24}$$

Zeeman Field

Given an external field $\boldsymbol{H}_{\text{zee}}$, the corresponding energy is given by

$$\mathcal{H}_{\text{zee}} = -\mu_0 \mu_s \sum_{\boldsymbol{i}} \boldsymbol{H^i}_{\text{zee}} \boldsymbol{s_i}. \tag{2.25}$$

The Hamiltonians introduced allow for a description of the energetics of magnetic systems. However, they do not provide information concerning their time evolution, thermal fluctuation or ground state. In the following section we will discuss the Landau-Lifshitz-Gilbert equation which is capable of describing the dynamics of the atomistic magnetisation.

2.2 Spin Dynamics

The first description of spin dynamics originated from ferromagnetic resonance experiments, where the dynamics of a magnetic material is described by an equation introduced by Landau and Lifshitz [27]. In modern form the Landau Lifshitz (LL) equation reads

$$\frac{\partial \boldsymbol{m}}{\partial t} = -\gamma \boldsymbol{m} \times \boldsymbol{H}_{\text{eff}} - \alpha\gamma \boldsymbol{m} \times (\boldsymbol{m} \times \boldsymbol{H}_{\text{eff}}). \tag{2.26}$$

where $\boldsymbol{m}$ denotes the direction of the sample magnetisation, γ the gyromagnetic ratio, α a phenomenological damping constant dependent on the material and $\boldsymbol{H}_{\text{eff}}$ the effective field which acts on the sample. Note that the LL equation was initially introduced for the dynamics of macroscopic systems and thus we introduced the variable $\boldsymbol{m}$ which refers to a sample magnetisation and is principally not related to the atomistic magnetisation direction $\boldsymbol{s}$.

Two physical effects give rise to this equation. The first term in the LL equation 2.26 describes the precession of the magnetisation and originates from the quantum mechanical interaction of an atomic spin with the applied field. The second term in LL equation 2.26 is introduced due to the relaxation of the magnetisation and accounts for energy dissipation [19]. It tends to align the magnetisation along the field direction with a characteristic coupling strength given by α. However, this relaxation rate of the magnetisation to the field direction is a linear function in the damping rate α and Gilbert has shown that this leads to an incorrect description of the dynamics for systems with high damping.

As a consequence, Gilbert adapted the LL equation by introducing critical damping, leading to the Landau-Lifshitz-Gilbert (LLG) equation given by

$$\frac{\partial \boldsymbol{m}}{\partial t} = -\frac{\gamma}{1+\alpha^2} \boldsymbol{m} \times \boldsymbol{H}_{\text{eff}} - \frac{\alpha\gamma}{1+\alpha^2} \boldsymbol{m} \times (\boldsymbol{m} \times \boldsymbol{H}_{\text{eff}}). \tag{2.27}$$

Even though this equation was initially derived to describe the dynamics of macroscopic systems, the LLG is the standard equation of motion in the field of numerical micromagnetism, capable of describing small magnetic elements.

Furthermore, the LLG equation can also be applied on an atomistic level. The precession term still has the same origin as it naturally arises from the quantum mechanical interaction of the spins with the effective field. The relaxation term now considers direct angular momentum transfer between the spins and the heat bath including contributions from the lattice and the electrons.

For the atomistic model the LLG equation reads

$$\frac{\partial \boldsymbol{s}}{\partial t} = -\frac{\gamma}{1+\alpha^2}\boldsymbol{s} \times \boldsymbol{H}_{\text{eff}} - \frac{\alpha\gamma}{1+\alpha^2}\boldsymbol{s} \times (\boldsymbol{s} \times \boldsymbol{H}_{\text{eff}}), \tag{2.28}$$

where we only interchanged the direction of the sample magnetisation $\boldsymbol{m}$ with the atomistic magnetisation direction $\boldsymbol{s}$.

A Side Note on Langevin Dynamics

It is important to mention that the LLG equation (2.27) is strictly only capable of describing systems at zero temperature. Thermal effects lead to thermodynamic fluctuation of the magnetic spin moments. For sufficient high temperatures these effects exceed the exchange interaction and oppress magnetic ordering, leading to the ferromagnetic-paramagnetic transition. A common method considering thermal effects is the Langevin Dynamics developed by Brown [28]. The main assumption is that thermal fluctuation on each atomic site can be represented by a Gaussian white noise term. The width of the Gaussian distribution increases with increasing temperature thus modelling stronger thermal fluctuations. In the framework of spin dynamics these thermal effects are considered by an additional thermal field term $\frac{1}{\mu_0}\boldsymbol{H}^{\boldsymbol{i}}_{\text{th}}$ at each time step which takes the form

$$\boldsymbol{H}^{\boldsymbol{i}}_{\text{th}} = \boldsymbol{\Gamma}(t)\sqrt{\frac{2\lambda k_B T}{\gamma \mu_s \Delta t}}, \tag{2.29}$$

where $\boldsymbol{\Gamma}(t)$ is the three dimensional Gaussian distribution with a mean of zero which is non-correlated in time, λ the Gilbert damping term, k_B the Boltzmann constant, T the Temperature, γ the absolute value of the gyromagnetic ratio, μ_s the absolute value of the magnetic spin moment and Δt the integration time step.

The resulting effective field then reads

$$\boldsymbol{H}^{\boldsymbol{i}}_{\text{eff, LD}} = -\frac{1}{\mu_0 \mu_s}\frac{\partial \mathcal{H}}{\partial \boldsymbol{s_i}} + \frac{1}{\mu_0}\boldsymbol{H}^{\boldsymbol{i}}_{\text{th}}. \tag{2.30}$$

However, within this thesis we consider systems at low temperatures where the thermal fluctuation becomes negligible and therefore do not include the additional field term in the current implementation.

When it comes to the description of larger magnetic systems the atomic model with its consideration of every single spin in the magnet is neither feasible nor – at least in most cases – necessary. In this case the more reasonable approach to describe the magnetics of micron-scale systems is the well established model of micromagnetics which we will discuss in the following.

2.3 Micromagnetic Model

The micromagnetic model can be derived from the atomistic model by introducing a continuum assumption which is justifiable for magnetic materials at length-scales between 1 nm and 1μm . Derivations are based on [19–21, 29]. In the atomistic model the exchange interaction by far is the dominating contribution to the Hamiltonian. Thus, the micromagnetic description of ferromagnets assumes only a slight tile between neighbouring atomic spins and a smooth change of the magnetisation direction within the sample. Therefore the discrete lattice of atomic spins can be replaced by a continuous varying magnetisation density function $\boldsymbol{M}(\boldsymbol{r})$ with constant modulus $|\boldsymbol{M}| = M_s$ and the spin Hamiltonian becomes an energy functional. The assumption that the elementary spins are aligned almost parallel holds true within a characteristic length scale λ, also referred to as the exchange length, which depends on the strength of the exchange interaction. Correspondingly, we state

$$\boldsymbol{M_i} \approx \boldsymbol{M_j} \quad \text{for} \quad |\boldsymbol{r_i} - \boldsymbol{r}_j| \ll \lambda \tag{2.31}$$

with $\boldsymbol{M_i}$ and $\boldsymbol{M_j}$ being the magnetic moments at sites $\boldsymbol{i}$ and $\boldsymbol{j}$, respectively. Moreover, a homogeneous density of elementary magnets is assumed and therefore we can express the vectorial magnetisation function $\boldsymbol{M}(\boldsymbol{r})$ in terms of a magnetisation direction vector $\boldsymbol{m}(\boldsymbol{r})$ by writing

$$\boldsymbol{m}(\boldsymbol{r}) = \frac{\boldsymbol{M}(\boldsymbol{r})}{M_s} \quad \text{with} \quad |\boldsymbol{m}(\boldsymbol{r})| = 1, \tag{2.32}$$

where M_s is the saturation magnetisation.

Within the framework of micromagnetics, the same interactions as introduced by the atomistic model are considered. In the following, we take into account the implications of the continuum approximation onto these interaction terms and introduce the corresponding micromagnetic interactions.

Exchange Interaction

Starting from the atomistic exchange Hamiltonian in equation (2.6), we derive the energy functional for the micromagnetic exchange interaction based on [20]. We consider the atomic spins to be placed on a three-dimensional square lattice with Cartesian coordinates and a lattice constant a. From this it follows that each spin has six nearest neighbours. We associate each atomic site $\boldsymbol{i}$ with a vector (x_i, y_i, z_i) and write

$$
\begin{aligned}
\mathcal{H}_{\text{exc}} &= -\frac{J}{2}\sum_{<\boldsymbol{ij}>}\boldsymbol{s_i}\cdot\boldsymbol{s_j} \\
&= -\frac{J}{2}\sum_{<ij>}\boldsymbol{s}(x_i,y_i,z_i)\cdot\boldsymbol{s}(x_j,y_j,z_j) \\
&= -\frac{J}{2}\sum_{i}\boldsymbol{s}(x_i,y_i,z_i)\cdot\Big[\boldsymbol{s}(x_i+a,y_i,z_i)+\boldsymbol{s}(x_i-a,y_i,z_i) \\
&\quad +\boldsymbol{s}(x_i,y_i+a,z_i)+\boldsymbol{s}(x_i,y_i-a,z_i)+\boldsymbol{s}(x_i,y_i,z_i+a)+\boldsymbol{s}(x_i,y_i,z_i-a)\Big]
\end{aligned}
\tag{2.33}
$$

Assuming the lattice constant a is much smaller than the system size we perform Taylor-expansion by writing

$$\boldsymbol{s}(x_i\pm a,y_i,z_i)\approx\boldsymbol{s}(x_i,y_i,z_i)\pm a\partial_x\boldsymbol{s}(x_i,y_i,z_i)+\frac{a^2}{2}\partial_x^2\boldsymbol{s}(x_i,y_i,z_i) \tag{2.34a}$$

$$\boldsymbol{s}(x_i,y_i\pm a,z_i)\approx\boldsymbol{s}(x_i,y_i,z_i)\pm a\partial_y\boldsymbol{s}(x_i,y_i,z_i)+\frac{a^2}{2}\partial_y^2\boldsymbol{s}(x_i,y_i,z_i) \tag{2.34b}$$

$$\boldsymbol{s}(x_i,y_i,z_i\pm a)\approx\boldsymbol{s}(x_i,y_i,z_i)\pm a\partial_z\boldsymbol{s}(x_i,y_i,z_i)+\frac{a^2}{2}\partial_z^2\boldsymbol{s}(x_i,y_i,z_i) \tag{2.34c}$$

The exchange Hamiltonian can then be written as

$$
\begin{aligned}
\mathcal{H}_{\text{exc}}\approx -\frac{J}{2}\sum_i\boldsymbol{s}(x_i,y_i,z_i)\cdot\Big[&6\boldsymbol{s}(x_i,y_i,z_i)+a^2\partial_x^2\boldsymbol{s}(x_i,y_i,z_i) \\
&+a^2\partial_y^2\boldsymbol{s}(x_i,y_i,z_i)+a^2\partial_z^2\boldsymbol{s}(x_i,y_i,z_i)\Big].
\end{aligned}
\tag{2.35}
$$

As $\boldsymbol{s}$ are spin moment directions with magnitude of unity we can write

$$
\begin{aligned}
&\boldsymbol{s}_i\cdot\boldsymbol{s}_i=1 \\
&\rightarrow(\partial_{x,y,z}\boldsymbol{s}_i)\cdot\boldsymbol{s}_i+\boldsymbol{s}_i\cdot(\partial_{x,y,z}\boldsymbol{s}_i)=0 \\
&\rightarrow\boldsymbol{s}_i\cdot(\partial_{x,y,z}\boldsymbol{s}_i)=0 \\
&\rightarrow(\partial_{x,y,z}\boldsymbol{s}_i)^2=-\boldsymbol{s}_i\cdot(\partial_{x,y,z}^2\boldsymbol{s}_i)
\end{aligned}
\tag{2.36}
$$

Inserting into equation (2.35) yields

$$
\begin{aligned}
\mathcal{H}_{\text{exc}} &\approx -\frac{J}{2}\sum_i 6-a^2\Big[(\partial_x\boldsymbol{s}_i)^2+(\partial_y\boldsymbol{s}_i)^2+(\partial_z\boldsymbol{s}_i)^2\Big]. \\
&\approx -3NJ+\frac{Ja^2}{2}\sum_i\Big[(\nabla s_{ix})^2+(\nabla s_{iy})^2+(\nabla s_{iz})^2\Big]. \\
&\approx -3NJ+\frac{Ja^2}{2}\sum_{\boldsymbol{i}}(\nabla\boldsymbol{s_i})^2.
\end{aligned}
\tag{2.37}
$$

Where N is the number of atoms. The sum approaches an integral in the limit and $\boldsymbol{s}$ is described by the continuous magnetisation direction $\boldsymbol{m}$ thus yielding

$$\mathcal{H}_{\text{exc}} \approx -3NJ + \frac{J}{2a}\int_{\Omega}(\nabla \boldsymbol{m})^2 \mathrm{d}\boldsymbol{r}. \tag{2.38}$$

where Ω is the integration volume. The constant first term in equation (2.38) yields the minimum exchange energy the system can reach in a configuration where all spins are aligned in parallel. The second term describes the exchange energy caused by the tilt of the atomic spins from the minimum-energy configuration.

In micromagnetics, the first term is usually neglected as it only adds a constant term to the energy and does not affect the physics of the system. Moreover, the most commonly used notation for the exchange energy reads

$$\mathcal{H}_{\text{ex}}^{\text{micro}} = A\int_{\Omega}(\nabla \boldsymbol{m})^2 \mathrm{d}\boldsymbol{r}, \tag{2.39}$$

where we introduced the micromagnetic exchange constant A with dimensions [J/m] which fulfils the relation $A = \frac{J}{2a}$. In analogy to the atomistic model, the field contribution of the micromagnetic exchange interaction is given by the negative variational derivative of the energy which reads

$$\boldsymbol{H}_{\text{exc}} = -\frac{1}{\mu_0 M_s}\frac{\delta \mathcal{H}_{\text{exc}}^{\text{micro}}}{\delta \boldsymbol{m}} = \frac{2A}{\mu_0 M_s}\Delta \boldsymbol{m}. \tag{2.40}$$

The exchange energy can be expressed in terms of $\boldsymbol{H}_{\text{ex}}$ with

$$\mathcal{H}_{\text{exc}}^{\text{micro}} = -\frac{1}{2}\mu_0 M_s \int_{\Omega}\boldsymbol{H}_{\text{exc}}(\boldsymbol{m})\cdot \boldsymbol{m}\ \mathrm{d}\boldsymbol{r}. \tag{2.41}$$

Uniaxial Anisotropy

The derivation of the micromagnetic uniaxial anisotropy is more straightforward. Again starting with the atomistic Hamiltonian we then account for the same assumptions made as in the previous section and find

$$\mathcal{H}_{\text{ani}} = -k\sum_{\boldsymbol{i}}(\boldsymbol{s_i}\cdot \boldsymbol{e}_{\text{ani}})^2 \approx -\frac{k}{a^3}\int_{\Omega}(\boldsymbol{m}\cdot \boldsymbol{e}_z)^2 \mathrm{d}\boldsymbol{r} \tag{2.42}$$

We will use the common micromagnetic notation by writing

$$\mathcal{H}_{\text{ani}}^{\text{micro}} = -K\int_{\Omega}(\boldsymbol{m}\cdot \boldsymbol{e}_z)^2 \mathrm{d}\boldsymbol{r}, \tag{2.43}$$

where by also introducing a micromagnetic uniaxial anisotropy constant K with dimensions [J/m^3] which fulfils the relation $K = \frac{k}{a^3}$.

[29] The micromagnetic anisotropy field is given by

$$\boldsymbol{H}_{\text{ani}} = -\frac{1}{\mu_0 M_s}\frac{\delta \mathcal{H}_{\text{ani}}^{\text{micro}}}{\delta \boldsymbol{m}} = \frac{2K}{\mu_0 M s}\boldsymbol{e}_u(\boldsymbol{e}_u\cdot \boldsymbol{m}) \tag{2.44}$$

and the energy functional can by calculated by evaluating

$$\mathcal{H}_{\text{ani}}^{\text{micro}} = -\frac{1}{2}\mu_0 M_s \int_{\Omega}\boldsymbol{H}_{\text{ani}}(\boldsymbol{m})\cdot \boldsymbol{m}\ \mathrm{d}\boldsymbol{r}. \tag{2.45}$$

Demagnetisation Field

The demagnetisation field accounts for the dipole-dipole interaction of the atomistic elementary magnets. In a continuous magnetisation configuration the demagnetisation field reads [29]

$$\boldsymbol{H}_{demag}(\boldsymbol{r}) = M_s \int_{\Omega} \tilde{\boldsymbol{N}}(\boldsymbol{r}-\boldsymbol{r}')\boldsymbol{m}(\boldsymbol{r}')\mathrm{d}\boldsymbol{r}', \tag{2.46}$$

where $\tilde{\boldsymbol{N}}$ is the demagnetisation tensor given by

$$\tilde{\boldsymbol{N}}(\boldsymbol{r}-\boldsymbol{r}') = -\frac{1}{4\pi}\nabla\nabla'\frac{1}{|\boldsymbol{r}-\boldsymbol{r}'|}. \tag{2.47}$$

When the continuous demagnetisation field is considered on a regular grid, an analytical expression for the demagnetisation tensor can be derived as was first shown by Newell et al [30]. Details of the numerical computation of the demagnetisation tensor are discussed in section 3.1.1.

The corresponding energy functional for a given demagnetisation field can be calculated by

$$\mathcal{H}_{\mathrm{dip}}^{\mathrm{micro}} = -\frac{1}{2}\mu_0 M_s \int_{\Omega} \boldsymbol{m}(\boldsymbol{r})\cdot\boldsymbol{H}_{\mathrm{demag}}(\boldsymbol{m})\, d\boldsymbol{r} = -\frac{1}{2}\mu_0 M_s^2 \iint_{\Omega} \boldsymbol{m}(\boldsymbol{r})\tilde{\boldsymbol{N}}(\boldsymbol{r}-\boldsymbol{r}')\boldsymbol{m}(\boldsymbol{r}')\mathrm{d}\boldsymbol{r}'\mathrm{d}\boldsymbol{r}. \tag{2.48}$$

Micromagnetic Antisymmetric Exchange Interaction

The derivation of the micromagnetic DMI is performed in analogy to the derivation of the micromagnetic exchange following [20]. For the micromagnetic DMI derivation we consider the atomistic DMI assuming a thin film in z direction and then generalize the result to arbitrary DMI axes. Again, we use Cartesian coordinates, replace spins of neighbouring sites with corresponding Taylor-expansions and then take the limit where the sum approaches the integral. We start with the atomistic Hamiltonian in equation (2.10) and find

$$
\begin{aligned}
\mathcal{H}_{DM} &= d \sum_{<\boldsymbol{ij}>} (\boldsymbol{u_{ij}} \times \boldsymbol{e}_{DM}) \cdot (\boldsymbol{s_i} \times \boldsymbol{s_j}) \\
&= d \sum_{<\boldsymbol{ij}>} (\boldsymbol{u_{ij}} \cdot \boldsymbol{s}_i)(\boldsymbol{e}_{DM} \cdot \boldsymbol{s}_j) - (\boldsymbol{u_{ij}} \cdot \boldsymbol{s}_j)(\boldsymbol{e}_{DM} \cdot \boldsymbol{s}_i) \\
&= d \sum_i s_x(x_i, y_i, z_i) \cdot \Big[s_z(x_i + a, y_i, z_i) - s_z(x_i - a, y_i, z_i) \Big] \\
&\qquad - s_z(x_i, y_i, z_i) \cdot \Big[s_x(x_i + a, y_i, z_i) - s_x(x_i - a, y_i, z_i) \Big] \\
&\qquad + s_y(x_i, y_i, z_i) \cdot \Big[s_z(x_i, y_i + a, z_i) - s_z(x_i, y_i - a, z_i) \Big] \\
&\qquad - s_z(x_i, y_i, z_i) \cdot \Big[s_y(x_i, y_i + a, z_i) - s_y(x_i, y_i - a, z_i) \Big] \\
&= \frac{2d}{a^2} \sum_i a^3 \Big(s_{ix} \partial_x s_{iz} - s_{iz} \partial_x s_{ix} + s_{iy} \partial_y s_{iz} - s_{iz} \partial_y s_{iy} \Big)
\end{aligned}
\tag{2.49}
$$

where we used the property of the cross product $(\boldsymbol{a} \times \boldsymbol{b}) \cdot (\boldsymbol{c} \times \boldsymbol{d}) = (\boldsymbol{a} \cdot \boldsymbol{c})(\boldsymbol{b} \cdot \boldsymbol{d}) - (\boldsymbol{a} \cdot \boldsymbol{d})(\boldsymbol{b} \cdot \boldsymbol{c})$ in the first step and multiplied by $\frac{a^2}{a^2}$ in the last step.

In the continuum limit, the sum approaches the integral and we find in agreement with [24]

$$
\mathcal{H}_{DM} \approx \frac{2d}{a^2} \int_\Omega \Big(m_x \partial_x m_z - m_z \partial_x m_x + m_y \partial_y m_z - m_z \partial_y m_y \Big) \mathrm{d}\boldsymbol{r}. \tag{2.50}
$$

We introduce the micromagnetic DMI constant D in units of $[\mathrm{J/m^2}]$ to find

$$
\mathcal{H}_{DM}^{\text{micro}} = D \int_\Omega \Big(m_x \partial_x m_z - m_z \partial_x m_x + m_y \partial_y m_z - m_z \partial_y m_y \Big) \mathrm{d}\boldsymbol{r} \tag{2.51}
$$

using the relation $D = \frac{2d}{a^2}$.

For this derivation we assumed the normal vector of the interface $\boldsymbol{n}$ pointing in z-direction. We can generalize equation (2.51) for arbitrary directions of $\boldsymbol{n}$ with

$$
\mathcal{H}_{DM}^{\text{micro}} = D \int_\Omega \Big((\boldsymbol{m} \cdot \boldsymbol{n})(\nabla \cdot \boldsymbol{m}) - \boldsymbol{m} \cdot (\nabla (\boldsymbol{m} \cdot \boldsymbol{n})) \Big) \mathrm{d}\boldsymbol{r} \tag{2.52}
$$

The effective micromagnetic DMI field is the negative functional derivative of the energy with respect to the magnetisation direction

$$
\boldsymbol{H}_{DM}^{\text{micro}} = \frac{1}{\mu_0 M_s} \frac{\delta \mathcal{H}_{\text{DM}}^{\text{micro}}}{\delta \boldsymbol{m}} = -\frac{2D}{\mu_0 M_s} [\boldsymbol{n} (\nabla \cdot \boldsymbol{m}) - \nabla (\boldsymbol{n} \cdot \boldsymbol{m})]. \tag{2.53}
$$

For a given DMI field the energy functional can be directly calculated evaluating

$$
\mathcal{H}_{\text{DM}}^{\text{micro}} = -\frac{1}{2} \mu_0 M_s \int_\Omega \boldsymbol{H}_{\text{DM}}(\boldsymbol{m}) \cdot \boldsymbol{m} \; d\boldsymbol{r}. \tag{2.54}
$$

Micromagnetic Zeeman Energy

In the case of an external applied field $\boldsymbol{H}_{\text{zee}}$ we approximate the discrete atomistic expression with an integral

$$\mathcal{H}_{\text{zee}} = -\mu_0 \mu_s \sum_{\boldsymbol{i}} \boldsymbol{s_i} \cdot \boldsymbol{H}^{\boldsymbol{i}}{}_{\text{zee}} \approx -\mu_0 \mu_s \int_\Omega \boldsymbol{m} \cdot \boldsymbol{H}_{\text{zee}} d\boldsymbol{r} \tag{2.55}$$

and thus we define the micromagnetic Zeeman energy by

$$\mathcal{H}_{\text{zee}}^{micro} = -\mu_0 \mu_s \int_\Omega \boldsymbol{m} \cdot \boldsymbol{H}_{\text{zee}} d\boldsymbol{r}. \tag{2.56}$$

Overview of the Atomistic and Micromagnetic Model

Within this chapter, we discussed the main assumptions of the atomistic spin model and described the interactions on a theoretical basis. We introduced a formalism to describe the dynamics of such a spin system and then derived the micromagnetic model by means of a continuum approach. Table 2.1 gives an overview of the interaction variables introduced within the atomistic and micromagnetic models and points out their relation. The compact formulation of those models in some way hides the complexity which arises in terms of the numerical computation. In the following chapter we will discuss the mathematical methods necessary to efficiently perform magnetic simulations within this framework.

Variable	Atomistic		Micromagnetic		Relation
	Symbol	Unit	Symbol	Unit	
Magnetic moment	$\boldsymbol{\mu_s}$	[J T^{-1}]	$\boldsymbol{M}$	[J $\text{T}^{-1}\text{m}^{-3}$]	μ_s=M_s a^3/n
Exchange energy	J	[J]	A	[J m^{-1}]	$J = 2Aa$
Anisotropy energy	k	[J]	K	[J m^{-3}]	$k = Ka^3$
DMI energy	d	[J]	D	[J m^{-2}]	$d = Da^2/2$

Table 2.1: Overview of the interaction variables introduced by the atomistic and micromagnetic models.

3 Discrete Mathematical Concepts

In the previous chapter we introduced the physical models which allow for the description of the dynamics of magnetic materials at micron scales. In the following we discuss the discrete numerical methods applied in order to solve these models. We especially consider the calculation of the demagnetisation field which is of high computational complexity as a result of its long-range character. We discuss the finite difference approach and investigate several discrete time integration methods in terms of precision and performance. Further considerations concerning the actual implementation of the simulation program are discussed in chapter 4. Moreover, we discuss the *string method*, a numerical method for the calculation of minimum-energy paths between arbitrary magnetic configurations [31]. This technique is applied in the calculation of magnetic skyrmion annihilation energies as described in chapter 5.

3.1 Fast Convolution Algorithm

When we consider the calculation of the atomistic dipole-dipole field or the discrete micromagnetic demagnetisation field in equations (2.18) and (3.14) , respectively, we face the computational expensive problem of solving a discrete convolution over the entire magnetisation. In a straight-forward approach, the calculation of these convolutions has a computational complexity of $\mathcal{O}(N^2)$ where N is the total number of simulation cells. However, in cases where we can convert the discrete convolution to a circular convolution, fast transform methods with convolution properties such as the Fast Fourier Transform (FFT) can be used to implement the computation which leads to a computational complexity of $\mathcal{O}(N\log N)$. In the following we introduce the convolution theorem on a continuous level and derive the corresponding discrete formulation following [32].

We start by considering a integrable function $f : \mathbb{R} \to \mathbb{C}$ and define the Fourier transform $\hat{f}(\omega)$ of $f(t)$ with $\omega, t \in \mathbb{R}$ as

$$\hat{f}(\omega) \equiv \int_{-\infty}^{\infty} f(t) e^{2\pi i t \omega} \mathrm{d}t = \mathcal{F}(f(t))(\omega), \tag{3.1}$$

and the inverse Fourier transform as

$$f(t) \equiv \int_{-\infty}^{\infty} \hat{f}(\omega) e^{-2\pi i \omega t} \mathrm{d}\omega = \mathcal{F}^{-1}(\hat{f}(\omega))(t), \tag{3.2}$$

where we denoted the Fourier transformation itself by $\mathcal{F}$ and the inverse transformation by $\mathcal{F}^{-1}$. Now considering two functions $g(t)$ and $h(t)$ with their corresponding Fourier

transforms $\mathcal{F}(g(t))$ and $\mathcal{F}(h(t))$, we define the convolution of the two functions, denoted as $g * h$ as [32, eq. 12.0.9]

$$(g * h)(t) \equiv \int_{-\infty}^{\infty} g(\tau)h(t-\tau)\mathrm{d}\tau \tag{3.3}$$

where $g*h = h*g$. The convolution theorem states that the function $g*h$ is one member of a simple transform pair

$$g * h \Leftrightarrow \mathcal{F}(g)\mathcal{F}(h) \tag{3.4}$$

and thus the Fourier transform of the convolution can be written as the product of the individual Fourier transforms

$$\mathcal{F}(g * h) = \mathcal{F}(g)\mathcal{F}(h). \tag{3.5}$$

On a discrete level we consider a even sequence of N complex numbers f_k with $k = 0, 1, \cdots, N-1$ with sampling interval Δ. The discrete variables t_k and ω_n are given by $t_k = k\Delta$ and $\omega_n = \frac{n}{N\Delta}$ with $n = -\frac{N}{2}, \cdots, \frac{N}{2}$.

The continuous Fourier transform can now be approximated by

$$\hat{f}(\omega_n) = \int_{-\infty}^{\infty} f(t)e^{2\pi i t\omega_n}\mathrm{d}t \approx \sum_{k=0}^{N-1} f_k e^{2\pi i t_k \omega_n}\Delta = \Delta\sum_{k=0}^{N-1} f_k e^{2\pi i k n/N} = \Delta\hat{f}_n \tag{3.6}$$

where the summation is referred to as the *discrete Fourier transform* $\mathcal{D}(f_k)$ of the N points f_k:

$$\mathcal{D}(f_k) = \sum_{k=0}^{N-1} f_k e^{2\pi i k n/N} = \hat{f}_n. \tag{3.7}$$

The continuous Fourier transform evaluated on the sequence f_k is then written as $\mathcal{F}(\omega_n) \approx \Delta\mathcal{D}_n$ and the discrete Fourier transform itself does not depend on any dimensional parameter such as the scale Δ. The inverse discrete Fourier transform $\mathcal{D}^{-1}$ recovers exactly the initial sequence f_k and reads

$$\mathcal{D}^{-1}(\hat{f}_n) = \frac{1}{N}\sum_{k=0}^{N-1} \hat{f}_n e^{-2\pi i k n/N} = f_k. \tag{3.8}$$

We define the discrete convolution of the complex functions g and h defined on the set $\mathbb{Z}$ of integers as

$$(g * h)_j \equiv \sum_{k=-\infty}^{\infty} g_{j-k}\ h_k. \tag{3.9}$$

Typically, one of these functions represents a data stream and the other a response function which is also referred to as convolution *kernel*. In most cases the kernel is of finite size N and the sum of the convolution can be truncated and is written as

$$(g * h)_j = \sum_{k=-N/2+1}^{N/2} g_{j-k}h_k. \tag{3.10}$$

The *discrete convolution theorem* states that for a periodic signal g_j with period M, such that it is completely define by its values $g_0, \cdots, g_{N-1}$, its discrete convolution with a response function of equal size M is a member of the discrete Fourier pair

$$\sum_{k=-N/2+1}^{N/2} g_{j-k}\ h_k \Leftrightarrow \hat{g}_n\ \hat{h}_n \tag{3.11}$$

where $\hat{g}_n$ is the discrete Fourier transform of the values g_j, $j = 0, \cdots, N-1$ and $\hat{h}_n$ is the discrete Fourier transform of the values h_k where $k = 0, \cdots, N-1$. These values h_k are the same as for the previous range $k = -N/2+1, \cdots, N/2$ but in 'wrap around' order, where the first half of the values corresponds to the positive coordinates and the second half to the negative coordinates.

As a result, we can write the discrete Fourier transform of the convolution as the point-wise product of the transformed functions:

$$\mathcal{D}(g * h) = \mathcal{D}(g)\mathcal{D}(h). \tag{3.12}$$

Finally, we can express the discrete convolution by the inverse discrete Fourier transform of the point-wise product of the transformed functions:

$$(g * h) = \mathcal{D}^{-1}\Big(\mathcal{D}(g)\mathcal{D}(h)\Big). \tag{3.13}$$

This numerical method is referred to as *fast convolution* and finds application in many fields such as signal and image processing. Moreover, we can use this method to efficiently calculate the micromagnetic demagnetisation field and the atomistic dipole-dipole field as shown in the following.

3.1.1 Discrete Micromagnetic Demagnetisation Field

The calculation of both the demagnetisation tensor and demagnetisation field is a computational expensive task. In this section we discuss the mathematical methods applied to allow for an efficient numerical calculation. Derivations follow [29, 33].

We consider the micromagnetic demagnetisation field of equation (2.46) and evaluate the continuous field on a regular cuboid gird where we denote each simulation cell of size $\Delta r_1 \times \Delta r_2 \times \Delta r_3$ by a multi-index $\boldsymbol{i} = (i_1, i_2, i_3)$. All continuous quantities are discretised correspondingly to match the spacial discretization and we write $\boldsymbol{m}(\boldsymbol{i}) \equiv \boldsymbol{m}_{\boldsymbol{i}}$. In this case the convolution integral becomes a discrete convolution which reads

$$\boldsymbol{H}^{\boldsymbol{i}}_{\text{demag}} = M_s \sum_{\boldsymbol{j}} \tilde{\boldsymbol{N}}_{\boldsymbol{i}-\boldsymbol{j}} \boldsymbol{m}_{\boldsymbol{j}}, \tag{3.14}$$

where the discrete demagnetisation tensor is given by

$$\tilde{\boldsymbol{N}}_{\boldsymbol{i}-\boldsymbol{j}} = \frac{1}{\Delta r_1 \Delta r_2 \Delta r_3} \iint_{\Omega_{\text{cell}}} \tilde{\boldsymbol{N}}\Big(\sum_k (i_k - j_k)\Delta r_k \boldsymbol{e}_k + \boldsymbol{r} - \boldsymbol{r}'\Big) \mathrm{d}\boldsymbol{r}\mathrm{d}\boldsymbol{r}'. \tag{3.15}$$

In the equation above, Ω_{cell} denotes a cuboid reference cell and $\boldsymbol{e}_k$ a unit vector in direction of the kth coordinate axis. The magnetisation is assumed constant within each simulation cell and the field generated by each source cell is averaged over each target cell. As a result, we encounter a sixfold integral when calculating the discrete demagnetisation tensor $\tilde{\boldsymbol{N}}_{\boldsymbol{i-j}}$ of equation (3.15). Newell et al. [30] derived an analytical solution of equation (3.15).

The diagonal element $N^{1,1}$ is computed by

$$\begin{aligned} N^{1,1}_{\boldsymbol{i-j}} = -\frac{1}{4\pi\Delta r_1\Delta r_2\Delta r_3}\sum_{\boldsymbol{k},\boldsymbol{l}\in[0,1]} (-1)^{\sum_x k_x+l_x} f\big[(i_1-j_1+k_1-l_1)\Delta r_1, \\ (i_2-j_2+k_2-l_2)\Delta r_2, (i_3-j_3+k_3-l_3)\Delta r_3\big], \end{aligned} \tag{3.16}$$

where f is an auxiliary function defined as

$$\begin{aligned} f(r_1,r_2,r_3) =& \frac{|r_2|}{2}(r_3^2-r_1^2)\sinh^{-1}\left(\frac{|r_2|}{\sqrt{r_1^2+r_3^2}}\right) \\ &+ \frac{|r_3|}{2}(r_2^2-r_1^2)\sinh^{-1}\left(\frac{|r_3|}{\sqrt{r_1^2+r_2^2}}\right) \\ &- |r_1r_2r_3|\tanh^{-1}\left(\frac{|r_2r_3|}{r_1\sqrt{r_1^2+r_2^2+r_3^2}}\right) \\ &+ \frac{1}{6}\Big(2r_1^2-r_2^2-r_3^2\Big)\sqrt{r_1^2+r_2^2+r_3^2} \end{aligned} \tag{3.17}$$

The elements $N^{2,2}$ and $N^{3,3}$ can be obtained by cyclic permutation:

$$N^{2,2}_{\boldsymbol{i-j}} = N^{1,1}_{(i_2,i_3,i_1)-(j_2,j_3,j_1)} \tag{3.18}$$

$$N^{3,3}_{\boldsymbol{i-j}} = N^{1,1}_{(i_3,i_1,i_2)-(j_3,j_1,j_2)} \tag{3.19}$$

The off-diagonal element $N^{1,2}$ calculates as

$$\begin{aligned} N^{1,2}_{\boldsymbol{i-j}} = -\frac{1}{4\pi\Delta r_1\Delta r_2\Delta r_3}\sum_{\boldsymbol{k},\boldsymbol{l}\in[0,1]} (-1)^{\sum_x k_x+l_x} g\big[i_1-j_2+k_1-l_1)\Delta r_1, \\ (i_2-j_2+k_2-l_2)\Delta r_2, (i_3-j_3+k_3-l_3)\Delta r_3\big], \end{aligned} \tag{3.20}$$

where we introduced another auxiliary function g which is given by

$$\begin{aligned} g(r_1,r_2,r_3) =& \big(r_1r_2r_3\big)\sinh^{-1}\left(\frac{r_3}{\sqrt{r_1^2+r_2^2}}\right) \\ &+ \frac{r_2}{6}\big(3r_3^2-r_2^2\big)\sinh^{-1}\left(\frac{r_1}{\sqrt{r_2^2+r_3^2}}\right) \\ &+ \frac{r_1}{6}\big(3r_3^2-r_1^2\big)\sinh^{-1}\left(\frac{r_2}{\sqrt{r_1^2+r_3^2}}\right) \\ &- \frac{r_3^3}{6}\tanh^{-1}\left(\frac{r_1r_2}{r_3\sqrt{r_1^2+r_2^2+r_3^2}}\right) \\ &- \frac{r_3r_2^2}{2}\tanh^{-1}\left(\frac{r_1r_3}{r_2\sqrt{r_1^2+r_2^2+r_3^2}}\right) \\ &- \frac{r_3r_1^2}{2}\tanh^{-1}\left(\frac{r_2r_3}{r_1\sqrt{r_1^2+r_2^2+r_3^2}}\right) \\ &- \frac{r_1r_2\sqrt{r_1^2+r_2^2+r_3^2}}{3}. \end{aligned} \tag{3.21}$$

The other off-diagonal elements are obtained by permutation once again:

$$N^{1,3}_{\boldsymbol{i-j}} = N^{1,2}_{(i_1,i_3,i_2)-(j_1,j_3,j_2)} \tag{3.22}$$

$$N^{2,3}_{\boldsymbol{i-j}} = N^{1,2}_{(i_2,i_3,i_1)-(j_2,j_3,j_1)} \tag{3.23}$$

The demagnetisation tensor is symmetric and thus the remaining components can be obtained by using the relation $N^{i,j} = N^{j,i}$, explicitly given as $N^{2,1} = N^{1,2}$, $N^{3,1} = N^{1,3}$ and $N^{3,2} = N^{2,3}$.

Fractions in the auxiliary functions f and g might feature zero denominators. However, limit considerations yield that all fractions tend to zero in that case. As of the implementation, these limit considerations are applied by setting the fraction to zero in case of zero denominators in order to avoid division-by-zero errors.

Considering a regular grid of size $n_1 \times n_2 \times n_3$, possible index distances are $-n_x \leq i_x - j_x \leq n_x$ and likewise for y and z. This leads to $(2n_1-1)\times(2n_2-1)\times(2n_3-1)\times 6$ entries of the demagnetisation tensor. The factor 6 arises as we can consider the symmetry of the tensor.

In the implementation of the assembly of the demagnetisation tensor we take into account that the actual calculation of the demagnetisation field is performed in Fourier space. This affects the spacial ordering we use during the tensor assembly. Instead of ordering the tensor array by increasing the distance starting with the largest negative distance, numbering starts at zero distance and then cycles periodically as shown in Figure 3.1. By this choice the numbering is adapted in a way that the discrete convolution

can be written as a cyclic discrete convolution. This gives rise to the usage of the fast convolution algorithm as is shown in the following.

By choosing a regular grid the convolutional structure of the demagnetisation field is maintained which gives rise to an evaluation using the fast convolution algorithm. The discrete convolution in equation (3.14) decomposes to a point-wise multiplication in Fourier space:

$$\mathcal{D}(\tilde{\boldsymbol{N}} * \boldsymbol{m}) = \mathcal{D}(\tilde{\boldsymbol{N}})\mathcal{D}(\boldsymbol{m}). \tag{3.24}$$

The actual result is then obtained by performing an inverse fast Fourier transform:

$$\boldsymbol{H}_{\text{demag}} = \mathcal{D}^{-1}\left(\mathcal{D}\big(\tilde{\boldsymbol{N}}\big)\mathcal{D}\big(M_s\boldsymbol{m}\big)\right). \tag{3.25}$$

This approach allows for a significantly reduced computational complexity of $\mathcal{O}(N \log N)$ when using the fast Fourier transform algorithm. In the following, we refer to this method as *fast convolution*.

In order to perform a point-wise multiplication in Fourier space, both the demagnetisation tensor and the discrete magnetisation are expected to have the same size. However, the demagnetisation tensor is of size $(2n_1 - 1) \times (2n_2 - 1) \times (2n_3 - 1)$ and the discrete magnetisation of $n_1 \times n_2 \times n_3$. In fast convolution method, the convolution kernel (in our case this is the demagnetisation tensor) is applied in a cyclic manner [32]:

$$(f * g)_i = \sum_{j=0}^{n-1} f_{(i-j+n)\%n} \cdot g_j \tag{3.26}$$

We expand the discrete magnetisation by entries of zero to match the size of the demagnetisation tensor, a method often referred to as zero-padding. By applying this method, only contributions of the magnetisation at physically possible cell distances are taken into account. After performing the inverse Fourier transform, the previously expanded values of the discrete magnetisation are neglected and the zero-padding does not affect the physical result.

At this point we want to emphasize the similar structure of equation (3.14) with its atomistic pedant given in equation (2.18). As a consequence, the Fourier based evaluation method for the demagnetisation field described above is also applicable to the calculation of the atomistic dipole-dipole field when considering equation (2.18) with atomistic dipole-dipole tensor $\tilde{\boldsymbol{D}}_{i-j}$ as given in equation (2.21).

3.2 Finite-Difference Methods

Finite-Difference methods (FD) are discrete mathematical methods for finding an approximate solution to differential equations. The basic idea of these methods is the approximation of differential equations by difference equations where finite-difference quotients are used to approximate derivatives [32].

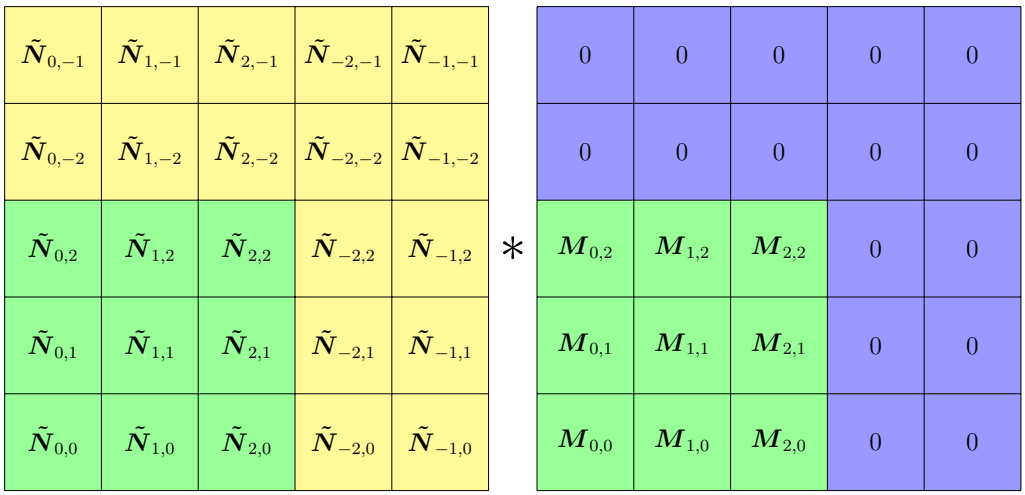


Figure 3.1: Schematic structure of the discrete convolution of the demagnetisation kernel $\tilde{\boldsymbol{N}}$ with the zero-padded discrete magnetisation $\boldsymbol{m}$. The tensor entries start with zero distance and proceed in a cyclic fashion where negative distances are wrapped around.

Other methods for numerically solving partial differential equations are the Finite Element Methods (FEM) or Monte Carlo, spectral and variational methods. However, FD methods are a common choice when it comes to the numerical solution of partial differential equations due to straightforward implementation and performance.

In general, we can write a finite-difference as $f(x+b)-f(x+a)$. Depending on where we evaluate the function we may categorize finite-differences in three different forms:

The forward-difference

$$\Delta_f f(x) = f(x+h) - f(x), \tag{3.27}$$

the backward-difference

$$\Delta_b f(x) = f(x) - f(x-h), \tag{3.28}$$

and the central-difference

$$\Delta_c f(x) = f(x+h/2) - f(x-h/2). \tag{3.29}$$

By adding an appropriate denominator, finite-difference quotients are used to approximate derivatives in numerical methods. As an example, we can define the derivative of a function f at point x by the limit

$$f'(x) = \lim_{h\to 0} \frac{f(x+h)-f(x)}{h}. \tag{3.30}$$

3.2.1 Finite-Differences as Discrete Convolutions

When we consider the calculation of a finite-difference derivative of a regularly discretised field we find that the result can be written as a convolution of the field with a small-sized

kernel. In general, a discrete convolution is written as [32]

$$(f * g)_n = \sum_{m=-\infty}^{\infty} f_m g_{n-m} = \sum_{m=-\infty}^{\infty} f_{n-m} g_m, \tag{3.31}$$

where f is the discrete field and g denotes the convolution kernel. The coefficients of the kernel depend on the type of derivative and the dimensionality of the field. Note that this method only is applicable if the considered field is discretised on a regular grid where the cell sized are constant for each respective dimension.

As an example let us assume we want to calculate the discrete Laplace operator in central-difference of an three-dimensional discrete field. In this case the convolution kernel is of size $3 \times 3 \times 3$ and can be written down as three 3×3 matrices

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & \frac{1}{\Delta r_3} & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & \frac{1}{\Delta r_2} & 0 \\ \frac{1}{\Delta r_1} & \frac{6}{\Delta r_1 \Delta r_2 \Delta r_3} & \frac{1}{\Delta r_1} \\ 0 & \frac{1}{\Delta r_2} & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 \\ 0 & \frac{1}{\Delta r_3} & 0 \\ 0 & 0 & 0 \end{bmatrix}, \tag{3.32}$$

where $\frac{1}{\Delta r_i}$ are the cell sizes in the respective spacial dimension.

The calculation of finite-differences with convolutions is well suited for GPU processing as these operations can easily be parallelized. However, when calculating a finite-difference in terms of a convolution we have to especially consider the correct calculation for the boundary elements and in many cases additional terms have to be added on the boundaries. Considering a GPU implementation, this may result in a high number of index operations leading to a performance loss when calculated on a GPU.

3.2.2 Calculation of the Micromagnetic Exchange Field

The exchange field in the micromagnetic model in equation(2.6) is calculated in finite-differences where the second order derivative is approximated in lowest order centred finite-difference which reads

$$f''(x) \approx \frac{f(x + \Delta x) - 2f(x) + f(x - \Delta x)}{\Delta x^2}. \tag{3.33}$$

We use this approximation in order to calculate the micromagnetic exchange field in equation (2.40). Accordingly, the three dimensional Laplacian is approximated by

$$\Delta \boldsymbol{m} \approx \sum_i \frac{\boldsymbol{m}(\boldsymbol{r} + \Delta r_i \boldsymbol{e}_i) - 2\boldsymbol{m}(\boldsymbol{r}) + \boldsymbol{m}(\boldsymbol{r} - \Delta r_i \boldsymbol{e}_i)}{\Delta r_i^2}. \tag{3.34}$$

where Δr_i is chosen to naturally match the cell size.

On the boundary, where neighbouring cells are missing, the magnetisation of the missing cell is assumed to be of the same value as the cell in the centre. This choice implicitly accounts for the Neumann boundary condition [34]. Generally, the Neumann boundary condition specifies the values in which the derivative of a solution is applied within the boundary of the domain $\partial_{\boldsymbol{n}} \boldsymbol{m}$, where $\boldsymbol{n}$ is the normal vector of the boundary.

In our case, the derivative is implicitly set to zero by assuming the missing neighbours with the same value as the cell itself. This yields a boundary condition given by

$$\partial_{\boldsymbol{n}} \boldsymbol{m} = 0. \tag{3.35}$$

As a side note, in the special case of the exchange field this choice of boundary condition has no effect on the time-derivative of the magnetisation, however. Adding a neighbour cell with the same value as the boundary cell only adds a term in the exchange field which is proportional to the magnetisation itself. On the right-hand side of the LLG, the effective field only appears in cross products with the magnetisation and thus this contribution is erased as parallel contributions drop out in cross-products.

Note that in the actual GPU implementation, index operations are high-priced and should be avoided whenever possible. Considering the missing neighbours would require $\sim 2(n_1 n_2 + n_2 n_3 + n_3 n_1)$ index operations. Therefore, neglecting the missing neighbours not only does not affect the physics but also leads to a notable performance gain in the GPU implementation.

Up to this point we introduced several numerical methods which are applied in order to calculate the effective field contributions of the different interactions. These contributions then sum up to the effective field in the LLG equation. In order to describe the dynamics of a magnetic system, we start with a given magnetisation, calculate the effective field and then integrate the LLG with respect to time. Mathematically, this corresponds to the problem of solving an ordinary differential equation with a given initial value problem which can be addressed by explicit time integration methods as discussed in the following.

3.3 Discrete Time Integration

In this section we present numerical methods for the approximate solution of ordinary differential equations (ODEs) using temporal discretization. Derivations follow [32]. Other methods are e.g. implicit methods or multi-step methods. We start with the Euler method, introduce the well known Runge-Kutta method of 4th order and then generalize the approach to the family of explicit Runge-Kutta (RK) methods. Later, we discuss adaptive step-size RK methods.

We consider a first-order differential equation as an initial value problem of the form

$$\dot{y} = f(t, y(t)),\ y(t_0) = y_0, \tag{3.36}$$

where y is a function of time t with value y_0 at $t_0 = 0$. The time derivative of y is given by the function $f(t, y(t))$. Without loss of generality, we only consider first-order differential equation as higher-order ODEs can be converted into systems of first-order equations. This allows any integration scheme for first-order systems to be applied for the approximate solutions of higher-order ODEs.

We now perform a Taylor expansion of the function y around t_0 with a small time step h and find

$$y(t_0 + h) = y(t_0) + hy'(t_0) + \frac{1}{2}y''(t_0) + \mathcal{O}(h^3). \tag{3.37}$$

By neglecting the quadratic and higher-order terms and substituting $y' = f(t, y)$ we obtain the *explicit Euler method*:

$$y(t_0 + h) = y(t_0) + hf(t_0, y(t_0)). \tag{3.38}$$

Starting from the initial value $y(t_0) = y_0$, the Euler method is a scheme which allows for the approximation of the function y at the subsequent time $t_0 + h = t_1$ and in general we write

$$y_{n+1} = y_n + h_n f(t_n, y_n), \tag{3.39}$$

where $y_n = y(t_n)$ and h_n is the – not necessarily constant – time step from t_n to t_{n+1}. The Euler method is the most basic explicit method for the numerical integration of ODEs. This method is of first order and therefore the local truncation error – which is the error per step – is proportional to the square of the step size, and the global truncation error – which is the accumulated error up to a given point in time – is proportional to the step size. When we consider the integration of the LLG equation using explicit integration schemes, we perform a re-normalisation of the magnetisation field after every integration step. The magnetisation direction $\boldsymbol{m_i}$ is a vector field with $|\boldsymbol{m_i}| = 1 \; \forall \; \boldsymbol{i}$ and we attempt to reduce the error per integration step by performing this re-normalisation. Due to its simplicity, the Euler method often serves as a basis for integration methods but is not recommended for practical use as the method involves a high local and global error and low stability. In the following, methods with higher accuracy and better overall performance are introduced.

A common extension of the Euler method is the *midpoint* method. In this method we take a first step as in equation (3.39) which serves as a trial step to the midpoint of the interval, denoted as k_1. The actual step is then evaluated using y and t at that midpoint:

$$\begin{aligned} k_1 &= hf(t_n, y_n) \\ k_2 &= hf(t_n + \frac{1}{2}h, y_n + \frac{1}{2}k_1) \\ y_{n+1} &= y_n + k_2 + \mathcal{O}(h^3) \end{aligned} \tag{3.40}$$

The symmetrization cancels out the first-order error term yielding a method of second order. By convention, a method of order p with step size h exhibits a local truncation error in the order of $\mathcal{O}(h^{p+1})$ and a global truncation error in the order of $\mathcal{O}(h^p)$. This approach of inserting trial steps in the interval can be advanced, resulting in a further decrease of the error. On the one hand, inserting further trial steps leads to a higher computational cost. On the other hand, higher-order methods allow for the choice of

higher step-sizes and, in most cases, this results in a considerably improved overall performance [32].

We can find many expressions to evaluate the function $f(t, y)$ that all agree to first order but feature different coefficient of higher-order. By adding up proper combinations of these expressions, the local truncation error can be eliminated order by order. This is the approach of general Runge-Kutta methods. In the following we introduce the standard Runge-Kutta method, a well known 4th-order method, and we extend the formulation to the family of Runge-Kutta methods.

3.3.1 Runge-Kutta methods

The widely used Runge-Kutta method of order 4 (RK4) uses 4 left-hand-side evaluations per time step [32]. It is calculated by

$$\begin{aligned} k_1 &= hf(t_n, y_n), \\ k_2 &= hf(t_n + \frac{h}{2}, y_n + \frac{1}{2}k_1), \\ k_3 &= hf(t_n + \frac{h}{2}, y_n + \frac{1}{2}k_2), \\ k_4 &= hf(t_n + h, y_n + k_3), \\ y_{n+1} &= y_n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4). \end{aligned} \tag{3.41}$$

This 4th order method exhibits a local truncation error in the order of $\mathcal{O}(h^5)$ and a total accumulated error in the order of $\mathcal{O}(h^4)$. Due to its straightforward implementation, robustness and performance the RK4 is a common choice for calculating approximate solutions of ODEs [32]. When we compare equations (3.40) and (3.41) we notice the similar structure of the methods. Consequently, we introduce the notation of general Runge-Kutta methods.

Generalization of Runge-Kutta methods

We denote the proximate step in a generalized Runge-Kutta method with

$$y_{n+1} = y_n + \sum_{i=1}^{s} b_i k_i. \tag{3.42}$$

At this point we distinguish between explicit and implicit Runge-Kutta methods. Explicit methods calculate the the proximate step y_{n+1} by considering the current state only, whereas implicit methods solve an equation which involves the current step and and the proximate step itself. In the following, we will only discuss explicit methods.

The intermediate steps of such an explicit method are calculated by

$$\begin{aligned} k_1 &= hf(t_n, y_n), \\ k_2 &= hf(t_n + c_2 h, y_n + a_{21} k_1), \\ k_3 &= hf(t_n + c_3 h, y_n + a_{31} k_1 + a_{32} k_2), \\ &\dots \\ k_s &= hf(t_n + c_n h, y_n + a_{s1} k_1 + a_{s2} k_2 + \dots + a_{s,s-1} k_{s-1}), \end{aligned} \tag{3.43}$$

where s is the number of stages or function evaluations, the coefficients a_{ij} with $1 \leq j < i \leq s$ are referred to as the Runge-Kutta matrix, the coefficients b_i with $i = 1, 2, ..., s$ as the weights and the coefficients c_i with $i = 2, 3, ..., s$ as the nodes. For visualization these coefficients are usually given in the form of a *Butcher tableau* as shown in Table 3.1.

i	c_i	$a_{i,j}$				b_i
1						b_1
2	c_2	$a_{2,1}$				b_2
3	c_3	$a_{3,1}$	$a_{3,2}$			b_3
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$		$\vdots$
s	c_s	$a_{s,1}$	$a_{s,2}$	$\dots$	$a_{s,s-1}$	b_s
	j=	1	2	$\dots$	$s-1$	

Table 3.1: Coefficients of an explicit Runge-Kutta method arranged in a Butcher tableau.

When we express the Runge-Kutta 4th order method within this formulation, we obtain the tableau given in Table 3.2.

i	c_i	$a_{i,j}$			b_i
1					1/6
2	1/2	1/2			1/3
3	1/2	0	1/2		1/3
4	1	0	0	1	1/6
	j=	1	2	3	

Table 3.2: Butcher tableau of the standard Runge-Kutta 4th order method.

We emphasize that the number of stages s does not necessarily correspond to the order p of the method. The minimum number of stages s required to yield a Runge-Kutta method of order p is an open problem. Several known values are listed in Table 3.3 an it can be shown that for $M > 8$ at least $M + 3$ stages and thus function evaluations are required [32, 35].

We further point out that the number of function evaluations, also referred to as right-hand-side (RHS) evaluations, per time step is an important indicator of the efficiency of the algorithm when considering the corresponding order of the method. Depending on

p	1	2	3	4	5	6	7	8
min s	1	2	3	4	6	7	9	11

Table 3.3: Minimal number of stages s to yield a Runge-Kutta method of order p [35].

the problem under investigation, one function evaluation may be computationally more expensive than the final addition of the intermediate values k. In the case of the LLG for example, three cross products of the vector-based fields have to be performed for each RHS evaluation. This leads to a trade off between achieving a high-order approximation and restricting the number of RHS evaluations to a minimum which is highly dependent on the respective problem.

Up to this point, the integration methods discussed take a particular value for the size of each consecutive step as an input parameter. A convenient choice of the step size is key to an efficient and accurate time integration algorithm. Most systems described by ODEs exhibit high variations in the dynamics of the characteristic quantities. Periods with high fluctuations require very small time steps whereas periods with small changes can be handled with larger time steps while still remaining sufficiently accurate. Therefore, defining a constant step size for all time steps is an inappropriate choice in most problems. Either the step size would be too small in certain periods leading to an inefficient calculation with high computation times or the time step would be too large resulting in insufficient accuracy.

These considerations suggest that the integration algorithm should be capable of determining an adequate step size which is adapted to the respective problem. This leads us to introducing time integrators with *adaptive step-size control*.

3.3.2 Adaptive Step-size Control

The aim of adaptive step-size control is to choose the highest step size possible in order to retain a desired accuracy. As mentioned above, this is intended to increase the integration efficiency as the algorithm chooses small step sizes for periods with high fluctuations and large step sizes in periods with less changes. For the implementation of an adaptive step-size integration method, the stepping control algorithm has to supervise the integration by some measure. As we are most concerned about achieving a desired accuracy, an estimate of the local truncation error is a good measure for choosing an appropriate step size. Deviations follow [32].

Considering the RK4 method, the most straightforward approach to obtain an error estimate is *set doubling*. In this approach we take each step twice, once in a full step $2h$ and once in two half-steps of h. The overhead in terms of right-hand-side evaluations created by this technique is $\frac{11}{8} = 1.375$ when considering the achieved accuracy. The step-doubling technique requires 11 evaluations as one point is shared whereas two simple steps would require 8 evaluations.

We denote the exact solution of these two steps by

$$\begin{aligned} y(t+2h) &= y_1 + (2h)^5\phi + \mathcal{O}(h^6) + \cdots \\ y(t+2h) &= y_1 + 2(h^5)\phi + \mathcal{O}(h^6) + \cdots , \end{aligned} \tag{3.44}$$

where, to the order of h^5, ϕ is a constant with order of magnitude of $\frac{y^{(5)}}{5!}$, y_1 denotes the approximate solution of the full step and y_2 the approximate solution of the half step. The difference between the two approximations is an adequate estimation of the local truncation error which we denote by Δ:

$$\Delta \equiv y_1 - y_2. \tag{3.45}$$

This estimation of the local truncation error is the quantity that the step-size controller intends to keep at the desired degree of accuracy by choosing an adequate step size.

Performing a certain Runge-Kutta scheme once at full and once at half step size is descriptive and straightforward, however, this approach requires a high amount of function evaluations for the achieved accuracy.

A more convenient way is the development of a *embedded Runge-Kutta* tableau which is designed especially for the application of adaptive step-size algorithms. The objective is to find a set of Runge-Kutta coefficients $a_{ij}, b_i, \hat{b}_i, c_i$ which yields two approximate solutions (one of order p using b_i and the other of order $p-1$ using $\hat{b}_i$) with as few right-hand-side evaluations as possible. This is achieved by finding a set of coefficients where the lower order approximation uses the same intermediate steps k as the high order approximation, but with other coefficients $\hat{b}$. In this case, no further function evaluations have to be performed and the lower-order approximation is obtained by simply adding the intermediate steps k with the second set of coefficients $\hat{b}$. This approach was originally proposed by Merson and popularized in a method of Fehlberg [32]. Later on, Fehlberg developed a fifth-order method with six function evaluations where another combination of the same functions yields a fourth-order method [36]. In general, such an embedded Runge-Kutta scheme is written as

$$y_{n+1} = y_n + \sum_{i=1}^{s} b_i k_i, \quad \hat{y}_{n+1} = y_n + \sum_{i=1}^{s} \hat{b}_i k_i, \tag{3.46}$$

where the function approximation y_{n+1} is of order p and the approximation $\hat{y}_{n+1}$ is of order $p-1$. The main performance gain compared to the RK4 double-step method is due to the shared intermediate steps k_i. The corresponding Butcher tableau for such an integration scheme is extended by one row of $\hat{b}$ coefficients, as shown in Table 3.4.

The truncation error Δ is again approximated by the difference of the two results:

$$\Delta = y_{n+1} - \hat{y}_{n+1} = \sum_{i=1}^{s} (b_i - \hat{b}_i) k_i. \tag{3.47}$$

This approximation of the truncation error applies to the lower-order value $\hat{y}_{n+1}$. However, we will use the higher order approximation y_{n+1} to continue the integration, a method called *local extrapolation* [32].

i	c_i	$a_{i,j}$				b_i	$\hat{b}_i$
1						b_1	$\hat{b}_1$
2	c_2	$a_{2,1}$				b_2	$\hat{b}_2$
3	c_3	$a_{3,1}$	$a_{3,2}$			b_3	$\hat{b}_3$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$		$\vdots$	$\vdots$
s	c_s	$a_{s,1}$	$a_{s,2}$	$\dots$	$a_{s,s-1}$	b_s	$\hat{b}_s$
	j=	1	2	$\dots$	$s-1$		

Table 3.4: Extended Butcher tableau including the coefficients $\hat{b}$ of the $p-1$ order method.

In the following we consider how to keep the approximated error within desired bounds. At first we define a scale and require the error to be less or equal than that value

$$|\Delta| = |y_{n+1} - \hat{y}_{n-1}| \leq scale. \tag{3.48}$$

The scale is defined by

$$scale \equiv atol + |y|\ rtol, \tag{3.49}$$

where *atol* is the absolute error tolerance and *rtol* is the relative error tolerance. In practice, $|y|$ is replaced by $\max(|y_n|, |y_{n+1}|)$ in the above equation to account for the case where one of them is close to zero. However, when solving the LLG, the magnetisation direction of each cell has the magnitude of unity and considering $|y|$ only is sufficient.

Out of equation 3.48 we define the scaled error quantity *err* as the quotient of Δ and the defined *scale* subject to a specified norm

$$err \equiv \left\| \frac{\Delta_i}{scale_i} \right\| \tag{3.50}$$

Two common choices for the norm are the euclidean norm or the maximum norm (i.e. worst offender) where the latter is chosen in the code.

In the step-size control algorithm, we accept the step if $err \leq 1$, otherwise reject it. We now search for the relation between the scaled error *err* and the chosen step size h. The error estimate Δ scales as h^p where p is the order of the higher-order method, and so does *err*. Assume we take a step h_1 with the scaled error err_1. The step h_0, which would have given the scaled error err_0, can be estimated as

$$h_0 = h_1 \left| \frac{err_0}{err_1} \right|^{\frac{1}{p}}. \tag{3.51}$$

When err_0 denotes the desired error, which is 1 in an efficient integration, the above equation tells us how to adapt the step size h for the next integration step. If err_1 is larger than 1 and the step is rejected, the equation tells us how much to decrease the step size for the next try of the present failed step. In the other case, when err_1 is

smaller than 1, it tells us how much we can safely increase the step size for the next step.

As the error estimates are not exact, but only accurate in leading order in h, a common procedure is to include a safety factor s which is typically a little smaller than unity. The subsequent step size is then given by

$$h_{n+1} = s\ h_n \left(\frac{1}{err}\right)^{\frac{1}{p}}. \tag{3.52}$$

Another commonly used procedure in adaptive step-size control is a limitation of the factor with which the step size increases or decreases. This measurement intends to avoid extreme changes in the step size which could lead to instability. A common choice is to restrict the next step size such that it increases by a factor not higher that 10 and decreases by a factor not less than 0.2.

In cases where the step size is limited by the stability properties of the integration method rather than the accuracy of the individual steps, the above presented step-size controller may not perform properly. The step size increases slowly as subsequent steps are accepted up to a point where the method becomes unstable. The sudden increase in the error estimate results in a drastic reduction of the following step and the procedure repeats itself.

An effective way to overcome this problem is to use ideas of *control theory* [32]. The classical controller of equation (3.52) is known as an integration controller with $\log h$ as the discrete control variable for which the control variable is obtained by integrating the control error signal. The approach from control theory is to add a term proportional to the control error. This type of controller is referred to as PI controller where P stands for proportional feedback and I for integral feedback. The resulting expression for the next step size takes the form

$$h_{n+1} = s\ h_n\ err_n^{-\alpha}\ err_{n-1}^{\beta}. \tag{3.53}$$

The exponents α and β should be chosen to scale at $1/p$. Setting $\alpha = 1/k$ and $\beta = 0$ recovers the classical controller on equation (3.52). Non-zero values for β improve the stability but reduces the efficiency for integration periods with less changes. As a compromise, [32] recommends $\beta \approx 0.4/k$ and $\alpha \approx 0.7/k = 1/k - 0.75\beta$.

In the following, we introduce several embedded Runge-Kutta schemes which allow for the application of the adaptive step-size algorithm. The first number used within the notation of these methods refers to the order used to advance the integration whereas the second number (in brackets) denotes the order used to calculate the error estimate.

Runge-Kutta-Fehlberg 5(4) method

The wide spread Runge-Kutta-Fehlberg 5(4) method yields one fifth-order and one fourth-order approximation by performing six function evaluations [36]. In the literature, this method is commonly abbreviated with RKF45. Comparing with Table 3.3,

we notice that this is the minimum amount possible in order to obtain a method accurate in fifth-order. This method is designed such that the fourth-order solution yields a lower error than the fifth-order solution which would not suggest the usage of local extrapolation (using the higher-order expression to advance the integration). Despite the original intention to use the fourth-order solution for the integration, Shampine and others suggest the usage of local extrapolation [37]. Table 3.5 shows the coefficients $a_{ij}, b_i, \hat{b}_i, c_i$ of the method arranged in a Butcher tableau.

i	c_i	$a_{i,j}$					b_i	$\hat{b}_i$
1							$\frac{16}{135}$	$\frac{25}{126}$
2	$\frac{1}{4}$	$\frac{1}{4}$					0	0
3	$\frac{3}{8}$	$\frac{3}{32}$	$\frac{9}{32}$				$\frac{6656}{12825}$	$\frac{1408}{2565}$
4	$\frac{12}{13}$	$\frac{1932}{2197}$	$-\frac{7200}{2197}$	$\frac{7296}{2197}$			$\frac{28561}{56430}$	$\frac{2197}{4104}$
5	1	$\frac{439}{216}$	-8	$\frac{3680}{513}$	$-\frac{854}{4104}$		$-\frac{9}{50}$	$-\frac{1}{5}$
6	$\frac{1}{2}$	$-\frac{8}{27}$	2	$-\frac{3544}{2565}$	$\frac{1859}{4104}$	$-\frac{11}{40}$	$\frac{2}{55}$	0
	j=	1	2	3	4	5		

Table 3.5: Runge-Kutta-Fehlberg 5(4) coefficients arranged in a Butcher tableau.

Dormand-Prince 5(4) method

The following fifth- and fourth-order method was developed by Dormand and Prince [38] and will be denoted as DP45. The coefficients of this method are chosen to minimize the error of the fifth-order solution which suggests the usage of local extrapolation. Even though this method uses seven stages, only six function evaluations per step are necessary in most cases. This is achieved by designing the method such that the last stage – k_7 in this case – is evaluated at the same point as the first stage k_1 of the next step, a method referred to as first-same-as-last (FSAL). This property is directly observable in the Butcher tableau as in this case, the $a_{i,j}$ coefficients of the last stage equal the respective b_i coefficients such that $a_{s,1} = b_1, a_{s,2} = b_2, \cdots , , a_{s,s-1} = b_{s-1}$. As a result, the number of function evaluations decreases by one evaluation per step (except for the first step). Another specific restriction on the usage of the FSAL property occurs by the integration of the LLG equation: As the magnetisation direction is normalized after every time step, the first stage of the subsequent step does no longer match the seventh stage of the previous step and thus the full seven stages have to be calculated in every step.

Bogacki-Shampine 3(2) method

The Bogacki-Shampine 3(2) method (BS23) is an embedded Runge-Kutta method of third- and second-order [39] based on a second-order method proposed by Ralston [40].

i	c_i	$a_{i,j}$						b_i	$\hat{b}_i$
1								$\frac{35}{384}$	$\frac{5179}{57600}$
2	$\frac{1}{5}$	$\frac{1}{5}$						0	0
3	$\frac{3}{10}$	$\frac{3}{40}$	$\frac{9}{40}$					$\frac{500}{1113}$	$\frac{7571}{16695}$
4	$\frac{4}{5}$	$\frac{44}{45}$	$-\frac{56}{15}$	$\frac{32}{9}$				$\frac{125}{192}$	$\frac{393}{640}$
5	$\frac{8}{9}$	$\frac{19372}{6561}$	$-\frac{25360}{2187}$	$\frac{64448}{6561}$	$-\frac{212}{729}$			$-\frac{2187}{6784}$	$-\frac{92097}{339200}$
6	1	$\frac{9017}{3168}$	$-\frac{355}{33}$	$\frac{46732}{5247}$	$\frac{49}{176}$	$-\frac{5103}{18656}$		$\frac{11}{84}$	$\frac{187}{2100}$
7	1	$\frac{35}{384}$	0	$\frac{500}{113}$	$\frac{125}{192}$	$-\frac{2187}{6784}$	$\frac{11}{84}$	0	$\frac{1}{40}$
	j=	1	2	3	4	5	6		

Table 3.6: Dormand-Prince 5(4) coefficients arranged in a Butcher tableau.

The authors recommend using local extrapolation due to accuracy and stability considerations. Therefore, the third-order approximation is used for the advancement of the steps. The pair is designed to feature the FSAL property resulting in only three net function evaluations per step. Even though this method is of comparatively low order, the authors state that such a low-order method is more efficient at crude accuracies [39], but only when considering linear differential equations [41]. The stability properties of higher-order methods worsen considerably at such accuracies and lower-order pairs perform better within this scope. Compared to the Dormand-Prince 3(2) and Fehlberg 2(3) methods, this method proves to perform better in terms of efficiency, reliability and stability [39]. The coefficients of the pair are given in Table 3.7.

i	c_i	$a_{i,j}$			b_i	$\hat{b}_i$
1					2/9	7/24
2	1/2	1/2			1/3	1/4
3	3/4	0	3/4		4/9	1/3
4	1	2/9	1/3	4/9	0	1/8
	j=	1	2	3		

Table 3.7: Bogacki-Shampine 3(2) coefficients arranged in a Butcher tableau.

Cash-Karp 5(4) method

The following method proposed by Cash and Karp is a variable-order Runge-Kutta method designed for initial value problems with rapidly varying right-hand sides [42]. The introduced fifth-order formula contains embedded formulas of all orders 1 through 4 and thus would be denoted as Cash-Karp 5(4,3,2,1) in our notation. This approach allows for computing solutions at several different orders and thus detecting discontinuities

before all function evaluations have been performed, either accepting the lower-order result or aborting the step.

However, within the scope of this thesis we use the firth-order to advance the integration and consider the fourth-order estimate only to calculate the estimated error, thus denoting the method as Cash-Karp 5(4) or CK45. The coefficients of this pair are given in Table 3.8.

i	c_i	$a_{i,j}$					b_i	$\hat{b}_i$
1							$\frac{37}{378}$	$\frac{2825}{27648}$
2	$\frac{1}{5}$	$\frac{1}{5}$					0	0
3	$\frac{3}{10}$	$\frac{3}{40}$	$\frac{9}{40}$				$\frac{250}{621}$	$\frac{18575}{48384}$
4	$\frac{3}{5}$	$\frac{3}{10}$	$-\frac{9}{10}$	$\frac{6}{5}$			$\frac{125}{594}$	$\frac{13525}{55296}$
5	1	$-\frac{11}{54}$	$\frac{5}{2}$	$-\frac{70}{27}$	$\frac{35}{27}$		0	$\frac{277}{14336}$
6	$\frac{7}{8}$	$\frac{1631}{55296}$	$\frac{175}{512}$	$\frac{575}{13824}$	$\frac{44275}{110592}$	$\frac{253}{4096}$	$\frac{512}{1771}$	$\frac{1}{4}$
	j=	1	2	3	4	5		

Table 3.8: Cash-Karp 5(4) coefficients arranged in a Butcher tableau.

Bogacki Shampine 5(4) method

This fifth- and fourth-order method proposed by Bogacki and Shampine (BS45) uses eight stages with a net value of seven function evaluations per step due to exploiting the idea of FSAL [43,44]. The unusual aspect of this pair is that it provides two fourth-order formulas ($\hat{b}_i$ with $i = 1, 2, \cdots, 7$ and $\hat{b}_i^*$ with $i = 1, 2, \cdots, 8$) in order to obtain two independent estimates of the local truncation error, thus enhancing the robustness of the error control. The error estimate obtained by the $\hat{b}$ coefficients is used after the seventh stage whether to accept the step or not. If proceeded, the second error estimate calculated with $\hat{b}^*$ is used as a second input for the step-size controller. If the step is accepted for both error estimations, the next step size is calculated using the error estimation obtained with the $\hat{b}^*$ coefficients, as these yield a more precise error estimate.

The main advantages of this pair are significantly low truncation error coefficients and a fifth-order solution with low error coefficients. This method is more efficient compared to the DP45 and RKF45 in terms of achieving a very high accuracy for cheap function calls [43]. The coefficients of this method are shown in Table 3.9.

Dormand Prince 8(7) method

The Dormand Prince 8(7) method uses 13 stages in order to obtain an 8th order approximation used for proceeding the integration in local extrapolation [41]. The 7th order solution is used to obtain the error estimation for the step-size algorithm. The high number of 13 function evaluations per step is compensated by the step-size algorithm

i	c_i	$a_{i,j}$							b_i	$\hat{b}_i$	$\hat{b}_i^*$
1									$\frac{587}{8064}$	$\frac{6059}{80640}$	$\frac{2479}{34992}$
2	$\frac{1}{6}$	$\frac{1}{6}$							0	0	0
3	$\frac{2}{9}$	$\frac{2}{27}$	$\frac{4}{27}$						$\frac{4440339}{15491840}$	$\frac{8559189}{30983680}$	$\frac{123}{416}$
4	$\frac{3}{7}$	$\frac{183}{1372}$	$-\frac{162}{343}$	$\frac{1053}{1372}$					$\frac{24353}{124800}$	$\frac{26411}{124800}$	$\frac{612941}{3411720}$
5	$\frac{2}{3}$	$\frac{68}{297}$	$-\frac{4}{11}$	$\frac{42}{143}$	$\frac{1960}{3861}$				$\frac{387}{44800}$	$-\frac{927}{89600}$	$\frac{43}{1440}$
6	$\frac{3}{4}$	$\frac{597}{22528}$	$\frac{81}{352}$	$\frac{63099}{585728}$	$\frac{58653}{366080}$	$\frac{4617}{20480}$			$\frac{2152}{5985}$	$\frac{443}{1197}$	$\frac{2272}{6561}$
7	1	$\frac{174197}{959244}$	$-\frac{30943}{79937}$	$\frac{8152137}{19744439}$	$\frac{666106}{1039181}$	$-\frac{29421}{29068}$	$\frac{482048}{414219}$		$\frac{7267}{94080}$	$\frac{7267}{94080}$	$\frac{79937}{1113912}$
8	1	$\frac{587}{8064}$	0	$\frac{4440339}{15491840}$	$\frac{24353}{124800}$	$\frac{387}{44800}$	$\frac{2152}{5985}$	$\frac{7267}{94080}$			$\frac{3293}{556956}$
	j=	1	2	3	4	5	6	7			

Table 3.9: Bogacki-Shampine 5(4) coefficients.

allowing for very large time steps due to the high order of the method. A preference for low order methods in the case of low accuracy calculations is only justified when treating linear differential equations. However, if a high order method is efficient or not highly depends on the application. In cases where the step size is restricted externally due to output or stability considerations, lower order methods perform better. The coefficients of this pair are given in [44].

3.4 String Method

In this section we present the *string method* proposed by Weinan E et al. [31, 45], a technique for computing transition pathways, free energy barriers and transition rates in complex systems with relatively smooth energy landscapes. The method advances by evolving strings using ODE solvers to solve initial value problems. The Runge-Kutta methods described above are a convenient choice for this application. A string describes, in this context, a smooth curve with an intrinsic parametrisation spawned between two metastable regions in configuration space. The string satisfies a differential equation which is constructed such that the dynamics of the string lead it to the most probable transition path between those two points.

In the following, we denote the two metastable points by A and B. Our object of interest is the most probable transition path between those points. Such most probable transition paths correspond to the minimum energy paths (MEPs) which are paths in configuration space along which the potential force is parallel to the path at every point. The MEPs allow for the identification of the relevant saddle points which represent the bottlenecks for a particular barrier-crossing event.

The basic idea of the string method is spanning a string between the points A and B and then evolving this string under a potential force field in order to find the MEP. This

field originates from the potential energy $V(x)$ and the two points A and B are assumed to be local minima of $V(x)$.

By definition, a MEP is a curve γ connecting A and B with satisfies

$$(\nabla V)^{\perp}(\gamma) = 0, \tag{3.54}$$

where $(\nabla V)^{\perp}$ is the component of ∇V normal to γ, thus

$$(\nabla V)^{\perp}(\gamma) = (\nabla V)^{\perp} - (\nabla V(\gamma), \hat{\tau})\hat{\tau}. \tag{3.55}$$

Here, $\hat{\tau}$ is defined as the unit tangent of the curve γ and $(\cdot, \cdot)$ denotes the Euclidean inner product. It can be proven that, in an appropriate mathematical setting, the MEP is the most probable path that the system will take under the over-damped dynamics to move between A and B while crossing the barriers in-between [46].

In an abstract notation, the dynamics for the evolution of such a curve is given by

$$v_n = -(\nabla V)^{\perp}, \tag{3.56}$$

where v_n denotes the normal velocity of the curve. We emphasize that for the evolution of the curve, only the normal component of the velocity affects the result. The tangential velocity moves points along the curve, changing its parametrisation only. As we are free to chose any particular parametrisation, this does not affect the curve itself.

We choose a particular parametrisation of the curve $\gamma : \gamma = \{\varphi(\alpha) : \alpha \in [0, 1]\}$ and in this case obtain $\hat{\tau}(\alpha) = \varphi_\alpha / \|\varphi_\alpha\|$, where φ_α denotes the derivative of φ with respect to α. A simple parametrisation is the equal arc-length parametrisation in which α is a constant multiple of the arc-length from point A to $\varphi(\alpha)$. With this choice, we have $\|\varphi_\alpha\| = l_\gamma$ =const where l_γ is the length of the curve γ.

In the original proposal of the string method [45], equation (3.56) is represented in the following model:

$$\dot{\varphi} = -\nabla V(\varphi)^{\perp} + \lambda(\alpha, t)\hat{\tau}(\alpha, t), \tag{3.57}$$

where $\dot{\varphi}$ denotes the time derivative of φ and the term $\lambda(\alpha, t)\hat{\tau}(\alpha, t)$ is a Lagrange multiplier added in order to enforce the particular parametrisation. In analogy to above, the Lagrange multiplier term does not affect the evolution of the string itself, but only its parametrisation as it does not contribute to the normal velocity of the curve. The stationary states of the dynamics of equation (3.57) satisfy equation (3.55) (as the time derivative vanishes). The action of the Lagrange multiplier term is, in the actual implementation, implicitly accounted for by an interpolation step.

The main issue of this method is in the computation of the projected force. In many cases, the method to calculate the tangent vector has to be changed before and after the saddle points are crossed due to numerical stability requirements [45]. This procedure lowers the accuracy of the entire method. However, the unfavourable projection step can be eliminated by using another model as proposed in the *simplified and improved string method* [31]. This method is not only simpler but also more stable and more accurate than the original method. Instead of the approach in equation (3.57), the new method uses

$$\dot{\varphi} = -\nabla V(\varphi) + \bar{\lambda}(\alpha, t)\hat{\tau}(\alpha, t), \tag{3.58}$$

where $\bar{\lambda}(\alpha,t)\hat{\tau}(\alpha,t)$ again is a Lagrange multiplier term used to enforce a particular parametrisation. This equation is equivalent to equation (3.57) by identifying $\bar{\lambda}(\alpha,t) = \lambda(\alpha,t) + (\nabla V, \hat{\tau})$, but better suited for the numerical implementation.

The string is discretised by a number of images $\{\varphi_i(t), i = 0, 1, \cdots, N\}$ where each image is evolved by iteration over the following two-step procedure:

1. The discrete images on the string are evolved over some time interval Δt according to the full potential force:
$$\dot{\varphi}_i = -\nabla V(\varphi_i). \tag{3.59}$$
2. The images are redistributed along the string using interpolation/re-parametrisation.

In the first step, the images are integrated in time according to equation (3.59) using an ODE solver. For this application, the adaptive Runge-Kutta methods discussed above are very efficient. In the second step, the string is re-parametrised by redistributing the images along the string according to a given parametrisation method.

The standard choice is enforcing parametrisation by equal arc length. In this case, given the images $\{\varphi_i^*\}$ on a non-uniform mesh $\{\alpha_i^*\}$, we interpolate these values onto a uniform mesh with the same number of images. This re-parametrisation is performed by:

(a) Calculation of the arc length of the current images:
$$s_0 = 0,\ s_i = s_{i-1} + \|\varphi_i^* - \varphi_{i-1}^*\|,\ i = 1, 2, \cdots, N, \tag{3.60}$$
where the mesh $\{\alpha_i^*\}$ is obtained by normalization of $\{s_i\}$:
$$\alpha_i^* = \frac{s_i}{s_N}. \tag{3.61}$$

(b) The new images $\{\varphi_i\}$ are obtained by interpolation of the old images $\{\varphi_i^*\}$ onto the uniform grid points $\alpha_i = i/N$.

After obtaining these re-parametrised images, we restart with step one and iterate until convergence.

Another choice of the parametrisation is using energy-weighted arc length. This approach gives finer resolution around saddle points and thus a better estimate of the energy barrier. In the re-parametrisation procedure, we calculate the energy-weighted arc length corresponding to the current images with
$$s_0^w = 0,\ s_i^w = s_{i-1}^w + W_{i-(1/2)}\|\varphi_i^* - \varphi_{i-1}^*\|,\ i = 1, 2, \cdots, N, \tag{3.62}$$
with $W_{i-(1/2)} = W(V_{i+1/2})$, where $W(z)$ is some positive, increasing function of $z \in \mathbb{Z}$ and $V_{i+1/2}$ being the average of the potential energy at φ_{i-1}^* and φ_i^*. The non-uniform mesh $\{\alpha_i^*\}$ is again obtained by normalizing $\alpha_i^* = \frac{s_i^w}{s_N^w}$ and the new images φ_i on $\alpha_i = i/N$ are obtained by interpolation.

3.4.1 String Method applied to Micromagnetics

We consider the micromagnetic models introduced in chapter 2 and want to apply the string method within this framework. The objective is to find an expression for the potential force of equations (3.57) or (3.58). In general, the potential force is a gradient field of the potential energy and the negative gradient points in the direction of the steepest decent on the energy landscape. Applied onto micromagnetics, the energy potential is given by the free energy $\mathcal{H}$.

Its potential force $\boldsymbol{F}$ is given by the negative variational derivative with respect to the magnetisation direction $\boldsymbol{m}$. Accounting for the conservation of the magnetisation, only the component perpendicular to the magnetisation is considered in $\boldsymbol{F}$ and we write

$$\boldsymbol{F}(\boldsymbol{m}) = -\Big(\frac{\delta \mathcal{H}}{\delta \boldsymbol{m}}\Big)_{\perp}. \tag{3.63}$$

This expression can be rewritten as

$$\begin{aligned}
\boldsymbol{F}(\boldsymbol{m}) &= -\Big(\frac{\delta \mathcal{H}}{\delta \boldsymbol{m}}\Big)_{\perp} \\
&= -\left[\frac{\delta \mathcal{H}}{\delta \boldsymbol{m}} - \Big(\frac{\delta \mathcal{H}}{\delta \boldsymbol{m}}\Big)_{\parallel}\right] \\
&= -\Big(\boldsymbol{m}\cdot\boldsymbol{m}\Big)\frac{\delta \mathcal{H}}{\delta \boldsymbol{m}} + \Big(\boldsymbol{m}\cdot\frac{\delta \mathcal{H}}{\delta \boldsymbol{m}}\Big)\cdot\boldsymbol{m} \\
&= \boldsymbol{m}\times\Big(\boldsymbol{m}\times\frac{\delta \mathcal{H}}{\delta \boldsymbol{m}}\Big) \\
&= -\boldsymbol{m}\times\Big(\boldsymbol{m}\times\mu_0\mu_s\boldsymbol{H}_{\text{eff}}\Big),
\end{aligned} \tag{3.64}$$

where we used the identity $A\times(B\times C) = (A\cdot C)\cdot B - (A\cdot B)\cdot C$ between the third and the fourth step and the identification $\boldsymbol{H}^i_{\text{eff}} = -\frac{1}{\mu_0\mu_s}\frac{\partial \mathcal{H}}{\partial \boldsymbol{m}_i}$ with $|\boldsymbol{m}| = 1$ as in equation (2.12). This expression is now used for the simplified string method and, according to equation (3.59), the dynamics of the images are given by

$$\dot{\boldsymbol{m}} = \boldsymbol{F}(\boldsymbol{m}). \tag{3.65}$$

As a remark, an additional pre-factor for $\boldsymbol{F}$ in equation (3.65) does only change the arbitrary time scale. In the string method this timescale has no physical meaning and thus we chose to adapt the pre-factor to equal the second them of the LLG in order to get a better 'feeling' for the time parametrisation.

4 Implementation

Up to this point, we have introduced both the atomistic spin model and the micromagnetic model and discussed the respective discrete numerical methods necessary for their computation. In this chapter we discuss several aspects of the actual implementation of the simulation software, which we will refer to as *pth-mag*. In the first section, we discuss several decisions regarding the design of the software. Following, we perform simulations for the well-known μMAG standard problem # 4. This problem considers dynamic aspects of micromagnetism and is highly sensitive to the interplay of the demagnetisation field and the exchange field. The presented results obtained with pth-mag show high agreement with reference simulations performed with magnum.fd. In the third section, we discuss several aspects of the embedded Runge-Kutta methods.

4.1 Software Design

Considering the equations describing the atomistic model and the micromagnetic model on a discrete gird, most of the calculation steps can be performed in parallel. When we consider the calculation of the uniaxial-anisotropy field in equation (2.16), for example, the numerical calculation for each simulation cell is straightforward. For an increasing number of cells, the operations performed on each cells has the same structure. In the field of parallel computing, this is referred to as singe instruction, multiple data (SIMD). For a large number of cells or nodes, this results in large blocks of data to be processed at once, which gives rise to the usage of graphical processing units (GPUs).

A common application programming interface (API) for parallel computing is the platform CUDA® (Compute Unified Device Architecture) created by Nvidia® [47]. It is a platform for general-purpose computing on GPUs (GPGPU) and allows for high performance computing on CUDA-enabled GPUs giving direct access to the GPU's virtual instruction set, parallel computation elements and compute kernels. CUDA supports the programming framework OpenCL and the interface of the platform is designed to work with the programming languages C,C++ and Fortran.

For the implementation of the simulation software, we decided to use the open-source high-level general-purpose software library ArrayFire [48] in version 3.5 at the time of writing. ArrayFire is a C/C++ library targeting parallel and massively-parallel architectures including central processing units (CPUs) and GPUs. Main benefits of this library are:

(a) Cross platform compatibility: ArrayFire provides a high-level of abstraction allowing the same code to be executed with CUDA, OpenCL and native CPU.

(b) Multilingual support: Although written in C/C++, the library provides wrappers for the programming languages Python, Rust, .NET, Java, R, Go, Node.js, Javascript, Lua, and Fortran.

(c) Open Source and distributed and maintained on GitHub: By going open-source the library receives contributions by the community accelerating the development and maintenance.

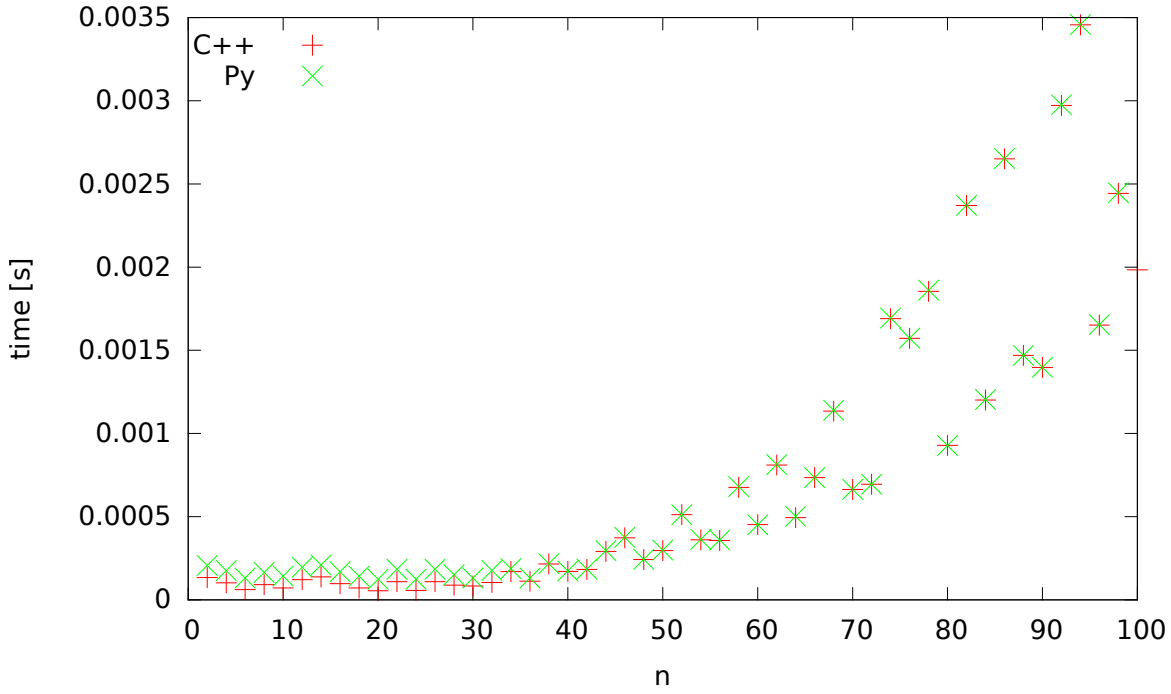
Although optimized for GPU computing, ArrayFire and – thus pth-mag – is compatible with CUDA, OpenCL and CPU which allows for hardware neutrality. Therefore, pth-mag is executable on CUDA-capable NVIDIA GPUs, OpenCL devices such as AMD GPUs/APUs (i.e. Accelerated Processing Units) and Intel Xenon Phi co-processors.

In order to decide for the actual programming language used in the implementation, two main aspects are considered. First of all, high performance is a key requirement needed for the ability to simulate large systems and minimize computation time. This would suggest a fully pre-compiled code like C++. However, higher flexibility for the adaption of simulation parameters and configurations as well as and extensions of the functionality would advocate a high-level programming language like Python.

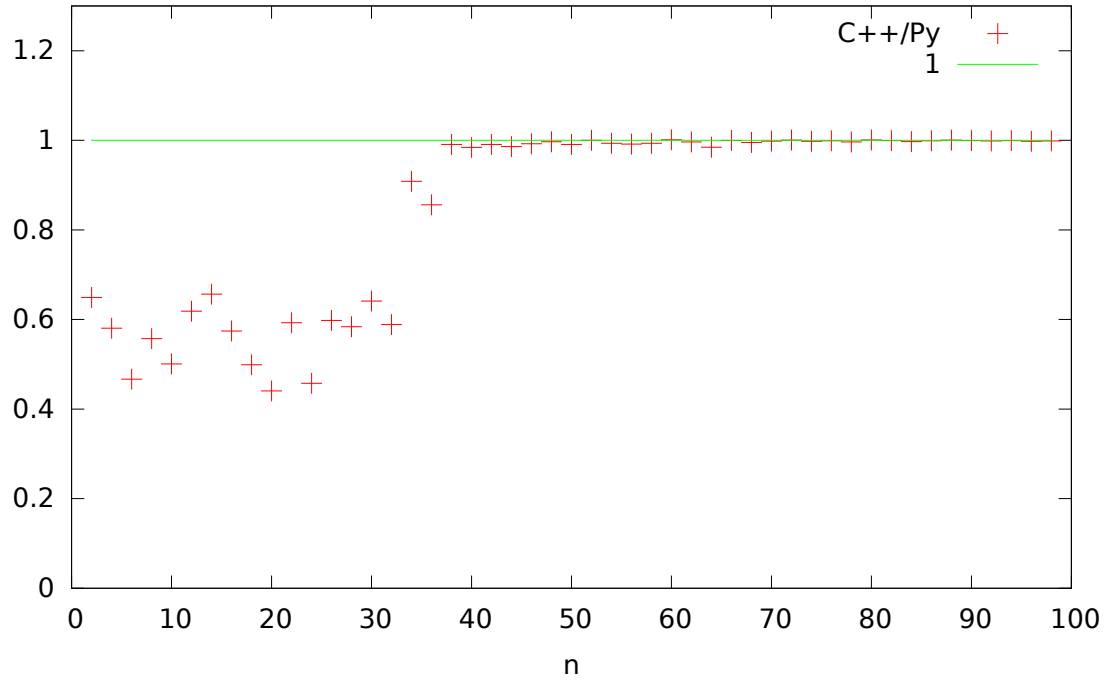
For a benchmark, we timed two computational expensive operations needed in the micromagnetic simulation: 3D-FFTs and 3D convolutions with small kernel sizes. The 3D-FFT is used twice in the fast convolution algorithm and represents an important operation for the calculation of the demagnetisation field. The ArrayFire implementation of the FFT uses a Cooley-Tukey scheme in a divide-and-conquer approach. The convolutions with kernel sizes of $(3 \times 3 \times 3)$ elements allow for an efficient calculation of the field contributions of the exchange and DM interactions in both the atomistic and the micromagnetic code and therefore represent another important operation in the implementation.

We performed both these operations on cubic data arrays once using the C++ ArrayFire version and once the ArrayFire-Python wrapping and measured the computation time. The used hardware is a Nvidia® Tesla® K20m GPU and an Intel® Xeon® E5-2630 0 CPU. Starting with two elements per dimension, the number of elements in each dimension (denoted by n) was increased up to 100 elements. For each n, one hundred operations are timed and the average time is plotted over n in Figures 4.1a and 4.2a. For a better comparison between the C++ and the Python version, the quotient of those timings is given in Figures 4.1b and 4.2b. The timings show that for smaller systems, the C++ version performs considerably better whereas for larger systems both versions achieve the same calculation time and the overhead of the Python wrapping is negligible. However, in many simulations scenarios, magnetic films with few simulation elements in one direction are investigated. In these applications, the C++ version is assumed to perform significantly better.

Following these considerations, we decided to implement the time-critical components of the code in C++ and provide an additional Python wrapper to allow for more flexibility and a user-friendly interface. Furthermore, with this self-made wrapping, new functionalities can be added easily to the existing code with full access to the functions provided by the ArrayFire-Python wrapping.

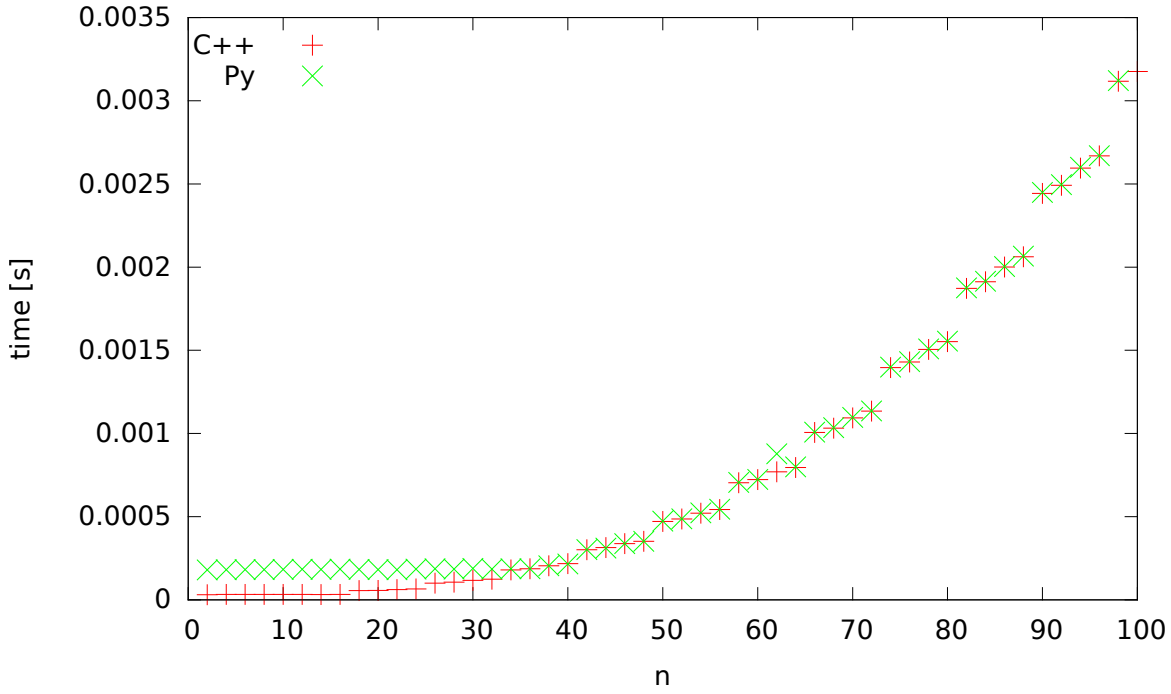


(a) Average time per 3D-FFT plotted over the number of elements per direction n.

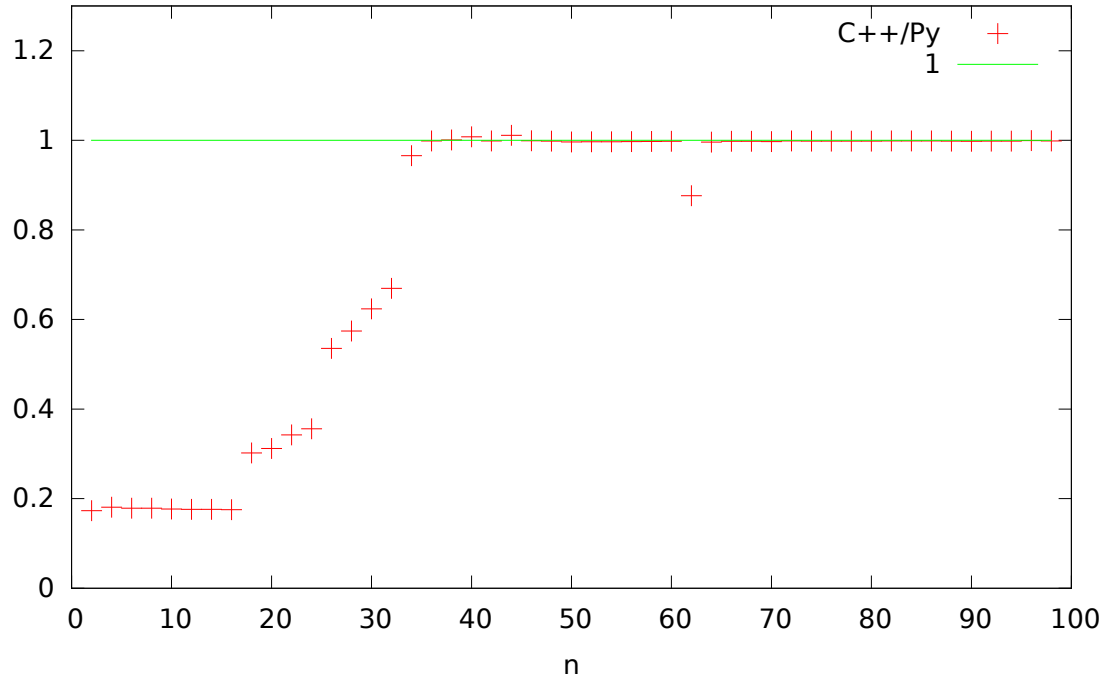


(b) Ratio of the C++ and Python timings plotted over the number of elements per direction for a better comparison.

Figure 4.1: Timings of 3D-FFT operations of the C++ and Python versions.



(a) Average time per 3D-convolution plotted over the number of elements per direction n.



(b) Ratio of the C++ and Python timings plotted over the number of elements per direction for a better comparison.

Figure 4.2: Timings of 3D-convolution operations of the C++ and Python versions.

4.2 The μMAG Standard Problem 4

In order to validate the software, we simulate the Standard Problem number 4 (SP4) proposed by the Micromagnetic Modelling Activity Group (μMAG) [49]. This problem focuses on dynamic aspects of micromagnetic computations. The specifications of the problem statement are as follows:

We consider a thin magnetic film in an initial 'S-state' in equilibrium. An S-state is a magnetisation configuration such as obtained by applying a saturating field along the $(1, 1, 1)$ direction and then reducing the external field to zero. In this equilibrium state, the magnetisation direction is S-shaped, giving this configuration its name. In the following, external fields of a given direction and magnitude are applied and the time evolution of the system is investigated as it moves towards equilibrium in the new circumstances.

The magnetic probe is of dimensions $l_x = 500$ nm, $l_y = 125$ nm and $l_z = 3$ nm and has material parameters similar to permalloy:

$$\begin{aligned} A &= 1.3 \times 10^{-11}\ \mathrm{J/m} \\ M_s &= 8.0 \times 10^{5}\ \mathrm{A/m} \\ K &= 0\ \mathrm{J/m^3} \end{aligned} \tag{4.1}$$

Further parameters are

$$\begin{aligned} \gamma &= 2.211 \times 10^{5}\ \frac{\mathrm{m}}{\mathrm{As}} \\ \alpha &= 0.02 \end{aligned} \tag{4.2}$$

The applied field is of strength $\mu_0 H_{\mathrm{app}} = 25$ mT and is directed at 170°counter-clockwise from the positive x-axis. At time $t = 0$ the field is applied instantaneously on the initial S-state. A meaningful description of the resulting dynamics is obtained by plotting the components of the spatially averaged magnetisation of the sample over the time.

Figure 4.3 shows the initial state of the simulation. The arrows indicate the direction of the magnetisation and the colour scale gives the magnitude of the y-component of $\boldsymbol{m}$. In the implementation, this state is obtained by relaxing a homogeneous magnetisation in (1,1,1)-direction until equilibrium.

In Figure 4.4, the respective spatially averaged magnetisation directions are plotted as a function of time. The results of pth-mag are compared to a simulation performed with the well established program magnum.fd [50] and show very good agreement. Figure 4.5 compares the $\langle m_y \rangle$ component of both simulations for a better visualization.

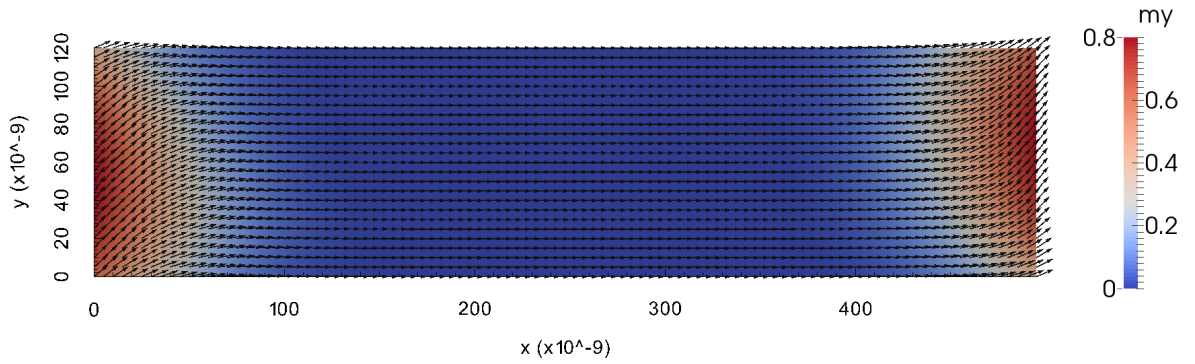


Figure 4.3: Initial magnetisation of the SP4 in equilibrium forming an S-state. The arrows indicate the magnetisation direction, the colour the magnitude of m_y.

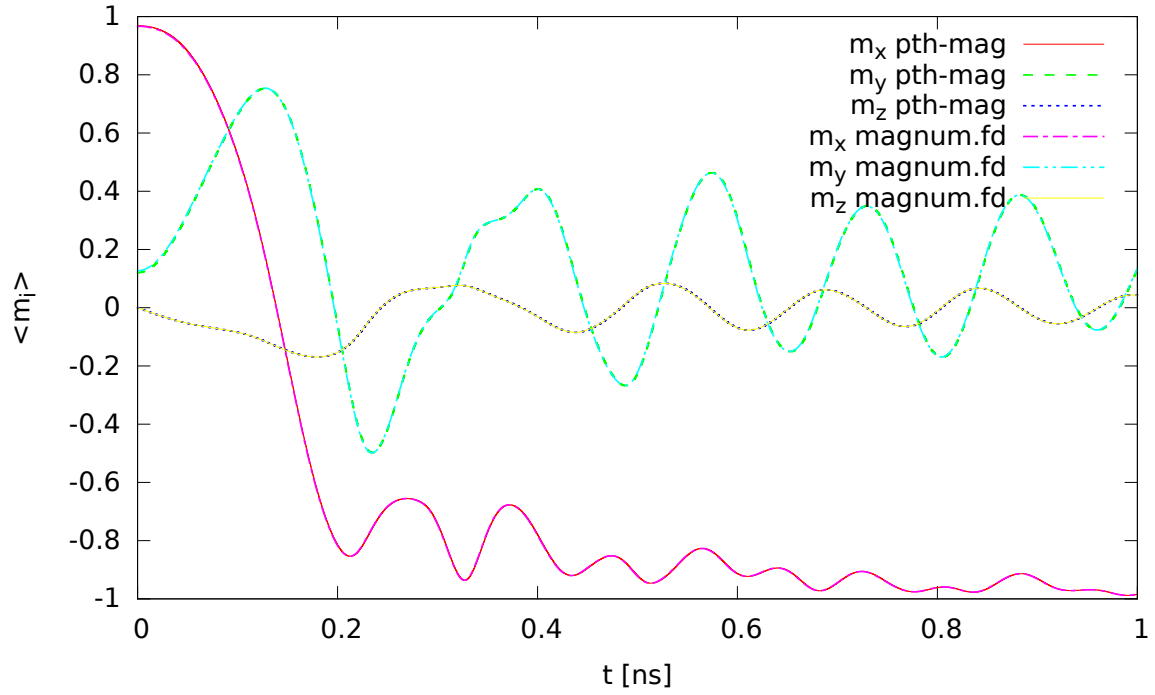


Figure 4.4: Spatially averaged magnetisations of the SP4 over time calculated by pth-mag and compared to magnum.fd.

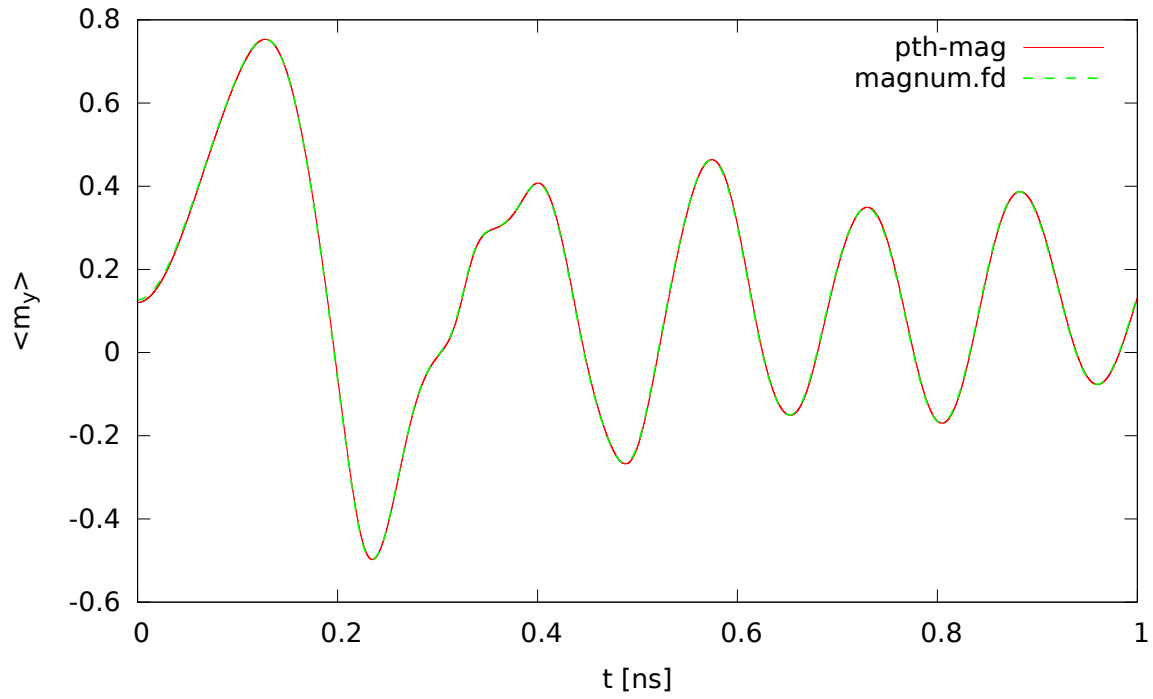


Figure 4.5: The $\langle m_y \rangle$ component of the spatially averaged magnetisation direction of the SP4 as a function of time calculated by pth-mag and compared to magnum.fd.

4.3 Comparison of Adaptive Runge-Kutta Methods

All Runge-Kutta methods discussed in chapter 3 are implemented in the code. We apply the different methods in the calculation of the SP4 and compare them in terms of accuracy and computational cost. The methods with fixed step size are not considered in the following as they are clearly outperformed by the embedded Runge-Kutta methods using adaptive step-size control.

We investigate the number of right-hand-side evaluations performed to simulate one nano second by each method for a given value of *atol*. Figure 4.6 shows the number of RHS evaluations as a function of *atol* for each method. As we already mentioned in chapter 3, the FSAL properties of the methods BS23, DP45, BS45 and BS45de cannot be exploited in this application as the magnetisation is re-normalized after every time step in order to enforce the norm preservation of the LLG, leading to one more function evaluation per time step. For lower accuracy requirements, the lower-order method (BS23) uses comparably few evaluations, as might be expected. For all methods, going from high *atol* values to lower ones, the number of RHS evaluations remains rather constant at the beginning up to some individual point, from where on the number of function evaluation increases exponentially (i.e. linearly for a logarithmic scale as in Figure 4.6). The slope of the graphs depends on the order of the method and is of the same magnitude for all fifth-order methods, highest for the third order method BS23, lowest for the eight order DP78 . The values of *atol* where the exponential increase starts are approximately $atol \approx 10^{-4}$ for BS23, $atol \approx 10^{-5}$ for RK45, $atol \approx 10^{-7}$ for the remaining order 5 methods and $atol \approx 10^{-8}$ for DP78. Remarkable is the low number of RHS-evaluations of the RKF45 method up to $atol \approx 10^{-6}$. The evaluations of the DP78 method even decreases between $atol = 10^{-5}$ and $atol \approx 10^{-8}$ and for values of $atol \approx 10^{-9}$ and below, this method needs the least function evaluations.

In order to get an estimate for the accuracy obtained depending on the chosen tolerance *atol* of equation (3.49), we compared the value of the averaged magnetisation in x-direction at 1 ns of simulation time between the several methods. The value of *rtol* is set to *atol*, as suggested in [32]. As a reference v_{ref}, we choose the value of $\langle m_x(t = 1\text{ns})\rangle$ obtained by the DP78 method with *atol* set to 10^{-11} and express the deviation v_{dev} of a value v in terms of the absolute relative error

$$v_{\text{dev}} = \left| \frac{v - v_{\text{ref}}}{v_{\text{ref}}} \right| \tag{4.3}$$

Figure 4.7 shows v_{dev} of $\langle m_x(t = 1\text{ns})\rangle$ for each method as a function of *atol*. The different Runge-Kutta methods are abbreviated as in chapter 3. Additionally, we consider the Bogacki-Shampine 5(4) method once with single error estimation (BS45) and once with double error estimation (BS45de). For the high value of $atol = 0.1$, most of the methods do not converge to the expected result as seen in the high discrepancy of the values. For $atol = 0.001$, all methods give already results with a relative error smaller than $1 * 10^{-5}$ in magnitude. From $atol = 10^{-5}$ up to $atol = 10^{-11}$, the different methods are very close together. When we consider the DP78 results, decreasing the error tolerance *atol* yields values slightly closer to the reference value. For the other

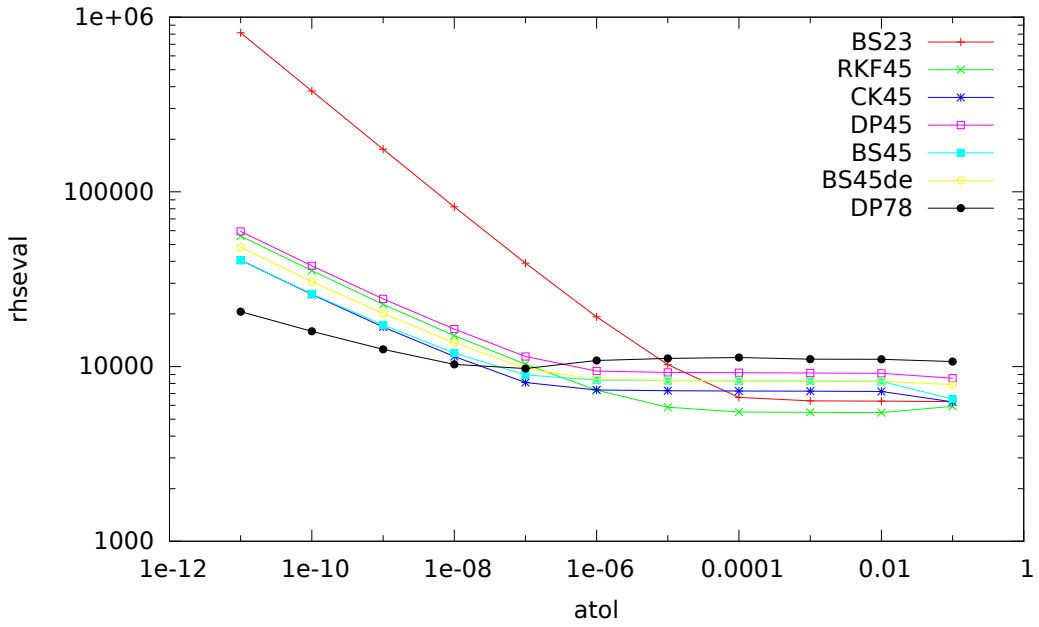


Figure 4.6: Number of right-hand-side evaluations of each Runge-Kutta method for simulating the SP4 for 1ns as a function of *atol* plotted on a logarithmic scale.

methods, no clear tendencies are observed. These results suggest that, at least at some point, the various methods all perform similarly for a tolerance up to 10^{-5}. However, we emphasize that these results are only meant to indicate the behaviour of the respective RK methods and are in no way a rigorous accuracy estimation.

Considering the low number of function evaluations we decide to set the RKF45 method with $atol = 10^{-6}$ as a default for further calculations due to its performance while supposedly maintaining reasonable accuracy.

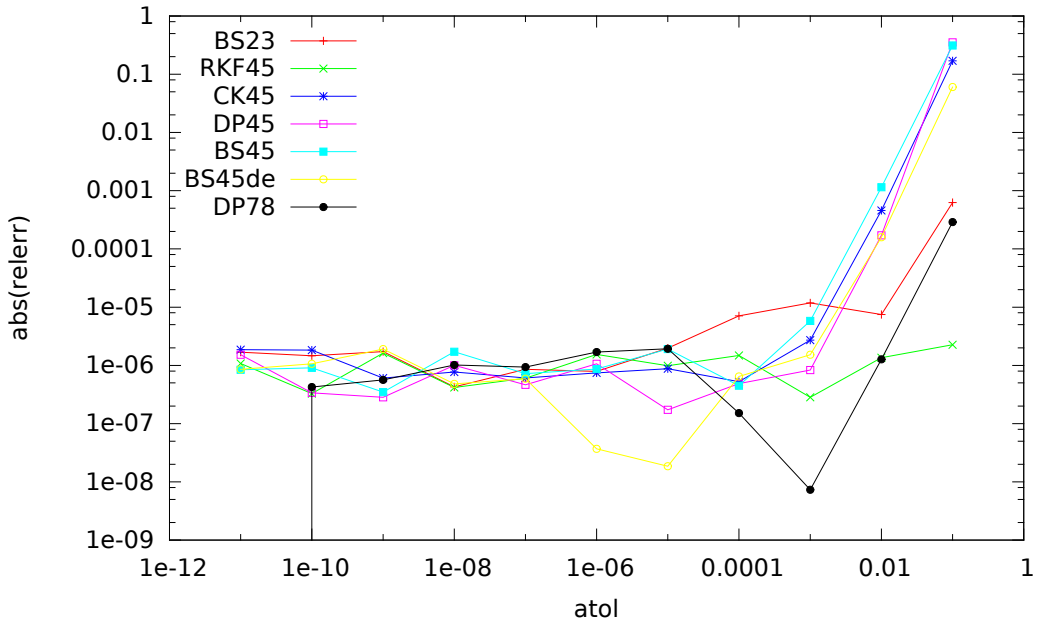


Figure 4.7: Absolute value of the relative error v_{dev} comparing $\langle m_x(t = 1\text{ns})\rangle$ of each Runge-Kutta method to the reference value v_{ref} as a function of *atol* on a logarithmic scale.

5 Skyrmion Annihilation Energy Barriers

In this chapter we investigate the formation and annihilation of magnetic skyrmions using pth-mag. We use both the atomistic and the micromagnetic model with all four interaction types as introduced in chapter 2. Namely, these are exchange, dipole-dipole/demagnetisation, uniaxial-anisotropy and antisymmetric exchange(DMI). Even though the two models are similar in many aspects, we point out the limitations of the micromagnetic model to describe atomic-scale processes.

By implementing the string method (described in section 3.4) and combining it with the models and numerical techniques described in chapters 2 and 3, we provide a tool to compute the minimum-energy-path (MEP) between two arbitrary magnetic configurations. In the following, we apply this method for the calculation of skyrmion annihilation energy barriers. Such a barrier determines the energy needed in order to erase a skyrmion and is correlated to the stability of this magnetic configuration. Systems featuring a high magnitude of the energy barrier are more stable compared to configurations with a lower barrier. When discussed in the scope of future storage devices, this barrier is the energy which thermal fluctuations or other excitations need to provide in order switch one equilibrium state into another (i.e. 'flipping a bit'). Therefore, this energy barrier can be used to estimate the average time the system remains in a specific state which provides the time scale over which it is possible to store information without corruption [4].

5.1 Energy Barrier Calculation

In the following, we demonstrate the application of the string method onto the atomistic model and how we obtain the energy barrier within this calculation. As a starting point for applying the string method, we define two magnetic configurations A and B which are in equilibrium and represent local minima on the energy landscape. These points are initial images A and B of the string method (c.f. section 3.4). In this context, an image describes a certain magnetisation configuration. The approach of the string method now is to define a curve in configuration space connecting these two images. In the numerical computation, the curve is split into a discrete amount of configurations which are located with a specific distance along the curve. In the implementation, we define the number of images and calculate the initial curve by interpolation of the two input images. Optionally, further images can be used as additional configurations located at the curve between A and B.

The dynamics of the string are such that each image heads in the direction of the

steepest descent on the energy landscape. In the case of the atomistic and the micromagnetic model, these dynamics are given by equation (3.65). The dynamical process is divided into discrete time steps and after each time step, the string is re-parametrised in order to sustain the desired spacings between the images. Speaking figuratively, one could imagine marbles placed in a line across a hill. At the beginning, the marbles are equally distanced (note that the distancing is not necessarily equal, however) . When released, the marbles on the left side roll off to the left and the marbles on the right side head to the right. The re-parametrisation could be thought of drawing a line between the marbles, picking them up and placing them on the drawn line according to specified distances. In the algorithm, this re-parametrisation is performed by an interpolation routine. The string method now repeats the two steps of (1.) evolving the string for a certain time and (2.) and re-parametrising the string to match some given parametrisation, until the curve approaches the MEP and reaches convergence.

For the calculation of the energy barrier we identify the skyrmion with image A and specify the second configuration to be a homogeneous magnetisation pointing in positive z-direction, identified with image B. The initial string is then obtained by linear interpolation between these two configurations. For the interpolation, we choose parametrisation according to equal arc-length (c.f. equation (3.60)).

We obtain the initial skyrmion configuration by time-evolving a circular magnetic configuration until it reaches equilibrium. Depending on the interactions we enable in the system and on the choice of the respective interaction strengths, such a circular magnetic configuration can relax into a skyrmionic state.

In the following, we discuss a exemplary atomistic simulation in order to present the evaluation procedure to determine an energy barrier. The parameters used in this simulation are given in Table 5.1. We note that the cell size of 1nm does not represent a physical lattice size but is used for an internal comparison with other calculations. Simulations with a set of parameters featuring atomistic distances are presented in section 5.2.

We consider a circular magnetisation in an atomistic description with parameters as given in Table 5.1 subject to exchange, uniaxial-anisotropy and DM interactions. The magnetic moments within the circle of diameter $l_x/2$ point in negative z-direction whereas the remaining moments point in positive z-direction. This configuration is shown in Figure 5.1a. By time-evolving this configuration until it reaches equilibrium, we obtain the magnetic skyrmion shown in Figure 5.1b. The magnetisation direction vectors for this type of skyrmions form a hedgehog-like structure and thus these field configurations are denoted as *hedgehog skyrmions*. There exist other skyrmion shapes such as *bubble skyrmions* or *chiral skyrmions* [51]. However, the hedgehog type is favoured by interfacial DMI and we only observe this skyrmion type in the following simulations. In Figure 5.2 we compare the geometry of this hedgehog skyrmion with calculations of Desplat et al. and a micromagnetic simulation performed with magnum.fe and notice good agreement with those results. This skyrmionic configuration is used as the initial state A for the string method. As mentioned above, state B is chosen to be a homogeneous magnetisation pointing in positive z-direction and the string is obtained by interpolation using equal arc-length parametrisation.

Quantity	Variable	Value	Unit
x-length	l_x	29	nm
y-length	l_y	29	nm
z-length	l_z	1	nm
Cell size	a	1	nm
Atomic magnetic moment	μ_s	1.1×10^{-21}	J/T
Exchange energy	J	3.2×10^{-20}	J
Anisotropy energy	k	6.4×10^{-21}	J
DMI energy	d	1.152×10^{-20}	J
Anisotropy direction	$\boldsymbol{e}_{\text{ani}}$	(0,0,1)	
DMI direction	$\boldsymbol{e}_{\text{DM}}$	(0,0,-1)	
Number of images	Image ID	60	
Time step	$\mathrm{d}t$	10^{-13}	s
Termination limit	l_{term}	10^{-8}	

Table 5.1: Exemplary set of simulation parameters used to present the evaluation procedure of the energy barrier calculation and for internal comparison.

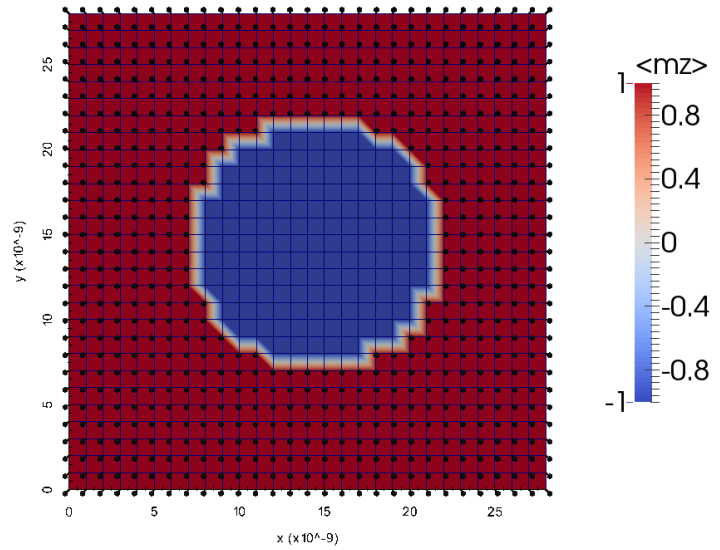
For each string we associate an energy curve which determines the energy barrier we have to overcome in order to change the magnetisation configuration according to this string. As we can choose the energy scale arbitrarily, we set the energy of the first image (Image ID=0) to zero. As a result, the maximum of the energy curve represents the energy barrier for the annihilation of the skyrmion.

In the discrete case, we have a fixed number of images representing the string. Each image has a corresponding energy as given by the Hamiltonian (c.f. equation (2.1)) of the system. For every step of the string method, we evolve each image for a certain time step $\mathrm{d}t$, re-parametrise the string and calculate the according energy curve until we reach convergence. As a measure of convergence m_{conv}, we consider the absolute value of the relative difference of the energy barrier between two consecutive steps. Denoting the energy barriers of the current step i and the previous step $i-1$ with E_i and E_{i-1}, we write the convergence measure as

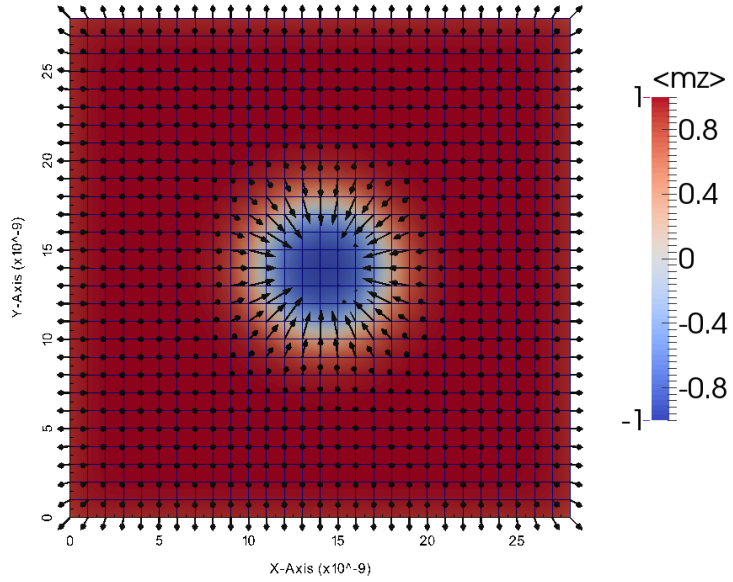
$$m_{\text{conv}} = \left| \frac{E_i - E_{i-1}}{\frac{1}{2}(E_i + E_{i-1})} \right|. \tag{5.1}$$

If this value drops below a defined limit l_{term}, we terminate the calculation.

For a time step of 10^{-13} s per iteration and a limit of $l_{\text{term}} = 10^{-8}$, the termination criterion was reached after 1462 iterations. The obtained energy curve of the string is shown in Figure 5.3. By setting the energy of the first image to zero, the value of the maximum determines the magnitude of the energy barrier. A selection of four images shows the respective magnetisation configurations at certain energies and indicates the annihilation of the skyrmion. The third configuration corresponds to the highest energy and depicts a point singularity. Such a singularity is also referred to as a Bloch point [21]. This magnetic field configuration is of special interest in the calculation of the energy



(a) Initial magnetic configuration. The magnetic moments inside the circle point in negative z-direction whereas the outer moments point in positive z-direction.



(b) Resulting skyrmionic configuration obtained by evolving (a) until equilibrium.

Figure 5.1: Initial magnetic configuration (a) in order to obtain the magnetic skyrmion in (b). The set of parameters defining the interaction strengths is given in Table 5.1. The initial configuration evolved in time until equilibrium results in the skyrmionic configuration.

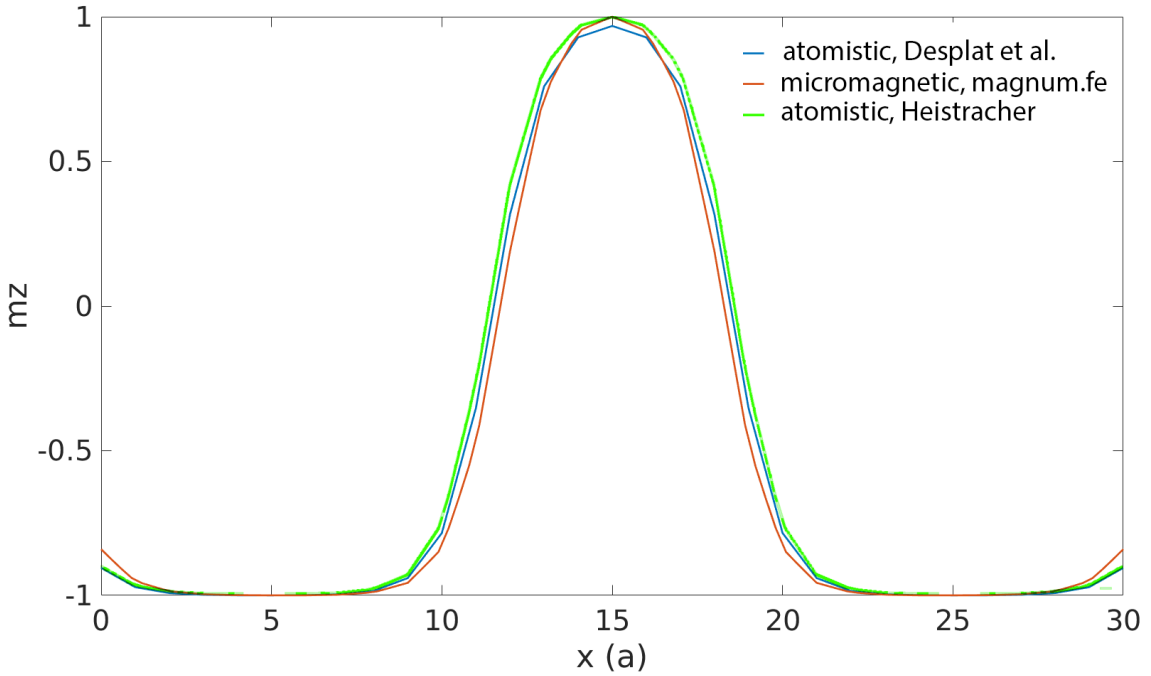


Figure 5.2: m_z component of the relaxed initial skyrmion (c.f. Figure 5.1b) plotted over the x-axis compared to an atomistic calculation of Desplat et al. and a micromagnetic calculation performed with magnum.fe.

barrier as it features the highest energy and therefore determines the magnitude of the energy barrier. For this set of parameters the resulting energy barrier has a magnitude of

$$4.421 \times 10^{-20}\text{J}.$$

This value is in good agreement with atomistic calculations performed by Desplat et al. [1] which yield a value of 4.416×10^{-20}J. Figure 5.4 shows the energy curves for each iteration step and displays the convergence of the individual curves against the result. The initial barrier (black line in Figure 5.4 corresponding to iteration 0) is around $\sim 2.7 \times 10^{-19}$J and decreases rapidly in the further iterations. The corresponding energy barrier for each respective curve is plotted in Figure 5.3 and shows at first a rapid decrease followed by minor changes until convergence. After 30 iterations, for example, the value of the energy barrier differs from the final result by $\sim 2.2\%$ and after 100 iterations $\sim 0.6\%$ are obtained.

A convenient choice for the time step dt is crucial for an efficient calculation. Too small values result in unnecessary computational effort due to the interpolation steps whereas too large time steps interfere with the convergence of the method. Considering the energy barrier per iteration (c.f. Figure 5.3) we observe kinks of the graph when the time step is chosen too large. Observing this graph for each simulation, we can adapt the time step accordingly.

[1] Internal communication with Louise Desplat

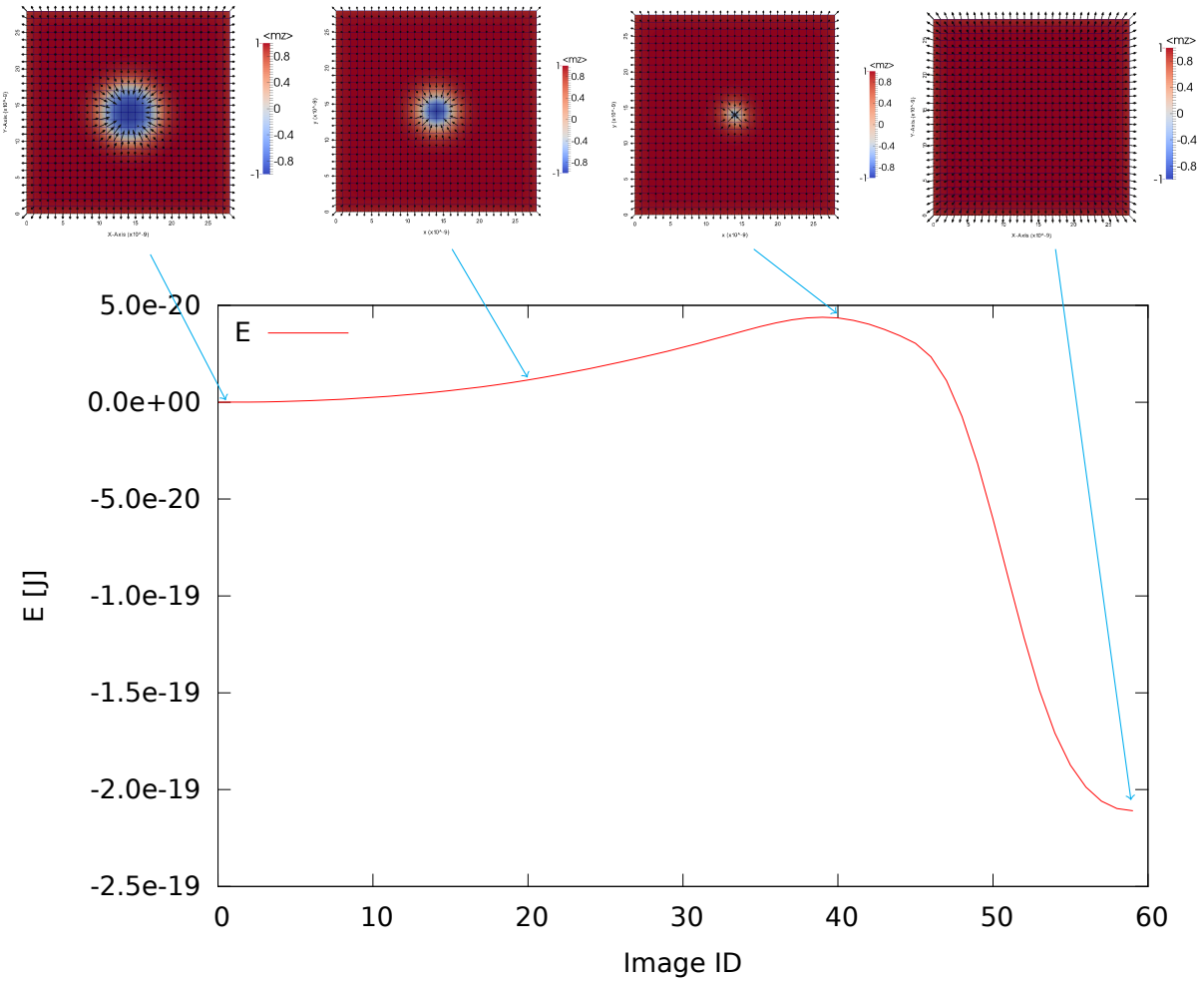


Figure 5.3: Converged energy curve of the string plotted over the image positions. Each image represents a magnetisation configuration along the string with an associated energy. Four selected configurations are depicted above and indicate their respective position on the curve by arrows. The energy barrier for the skyrmion annihilation is given by the difference of the highest energy value and the initial image energy (which is, by convention, set to zero).

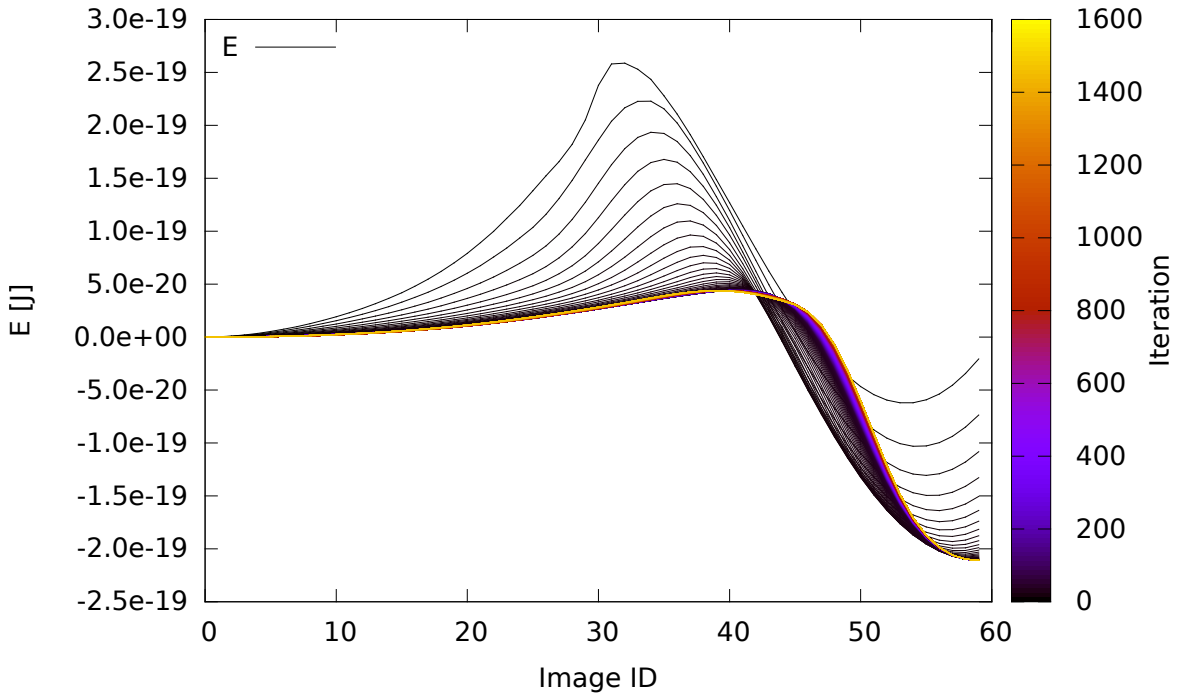


Figure 5.4: Energy curve of the string given for each iteration.

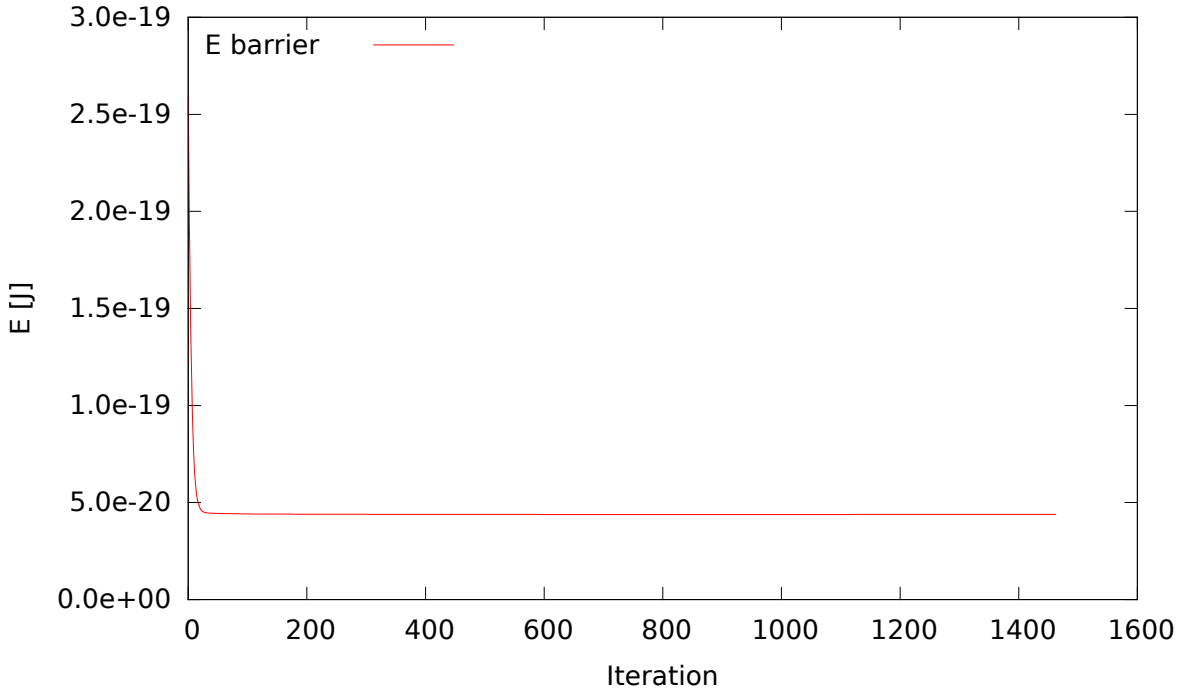


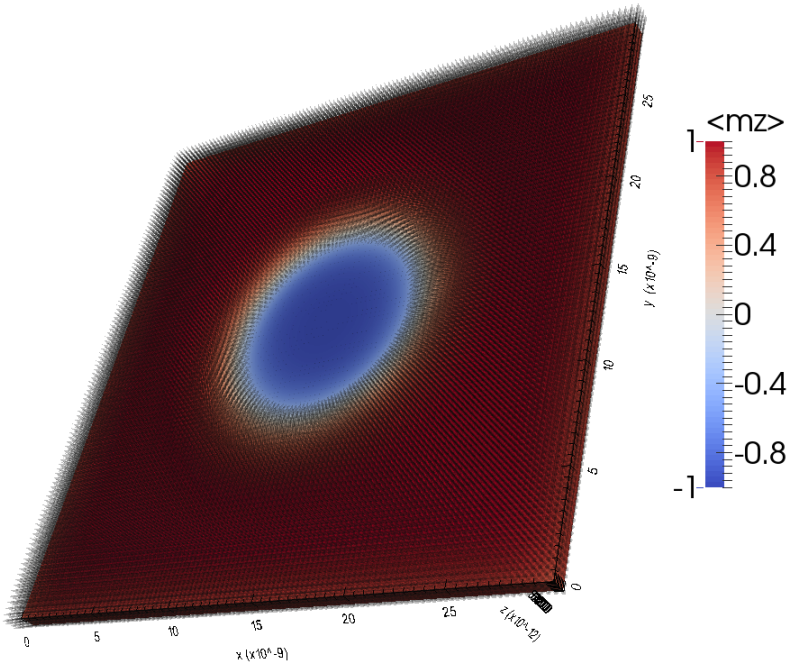
Figure 5.5: Magnitude of the energy barrier as a function of the iteration.

5.2 Atomistic Simulations

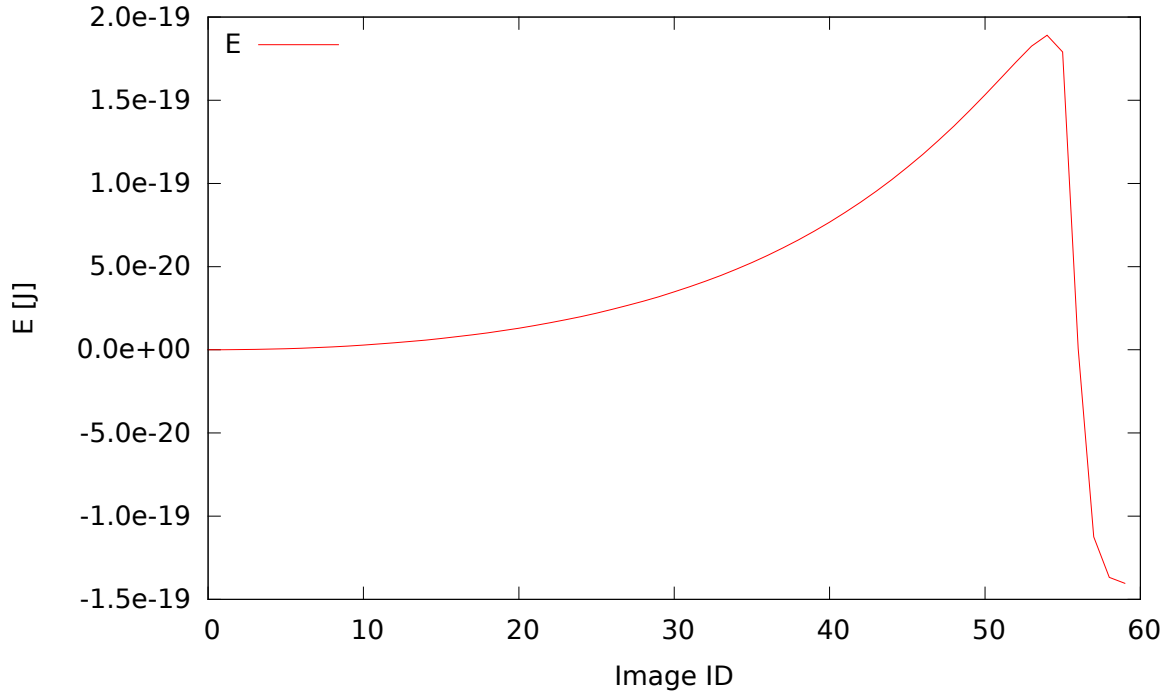
In the previous section we presented the application of the string method onto the magnetic models and discussed the calculation of the energy barrier presenting results obtained with a set of exemplary model parameters. In the following, we consider an atomic spin system modelled with the parameters given in Table 5.2. We assume a cell size of $a = 0.2715\,\mathrm{nm}$, which is the elementary magnet cell size for a monolayer of Fe on Ir(1,1,1) [10] and consider all four interactions which are exchange, dipole-dipole, uniaxial-anisotropy and antisymmetric exchange. This system represents a magnetic thin-layer with dimensions $30.1365\,\mathrm{nm} \times 30.1365\,\mathrm{nm} \times 1.086\,\mathrm{nm}$ and we consider $112 \times 112 \times 4$ simulation cells. Figure 5.6a shows the initial skyrmionic configuration representing the first image of the string method. The resulting energy curve is shown in Figure 5.6b and yields an energy barrier of 1.891×10^{-19} J. The energy curves per each iteration are given in Figure 5.7a and show the convergence of the curves towards the final energy curve. The according energy barrier for each step are given in Figure 5.7b and are used to monitor the choice of the time steps of the string method.

Quantity	Variable	Value	Unit
x-length	l_x	30.408	nm
y-length	l_y	30.408	nm
z-length	l_z	1.086	nm
Cell size	a	0.2715	nm
Atomic magnetic moment	μ_s	2.2014×10^{-23}	J/T
Exchange energy	J	3.2×10^{-20}	J
Anisotropy energy	k	6.4×10^{-21}	J
DMI energy	d	1.152×10^{-20}	J
Anisotropy direction	$\boldsymbol{e}_{\mathrm{ani}}$	(0,0,1)	
DMI direction	$\boldsymbol{e}_{\mathrm{DM}}$	(0,0,-1)	
Number of images	Image ID	60	
Time step	$\mathrm{d}t$	5×10^{-13}	s
Termination limit	l_{term}	1×10^{-6}	

Table 5.2: Model parameters for an atomistic simulation.

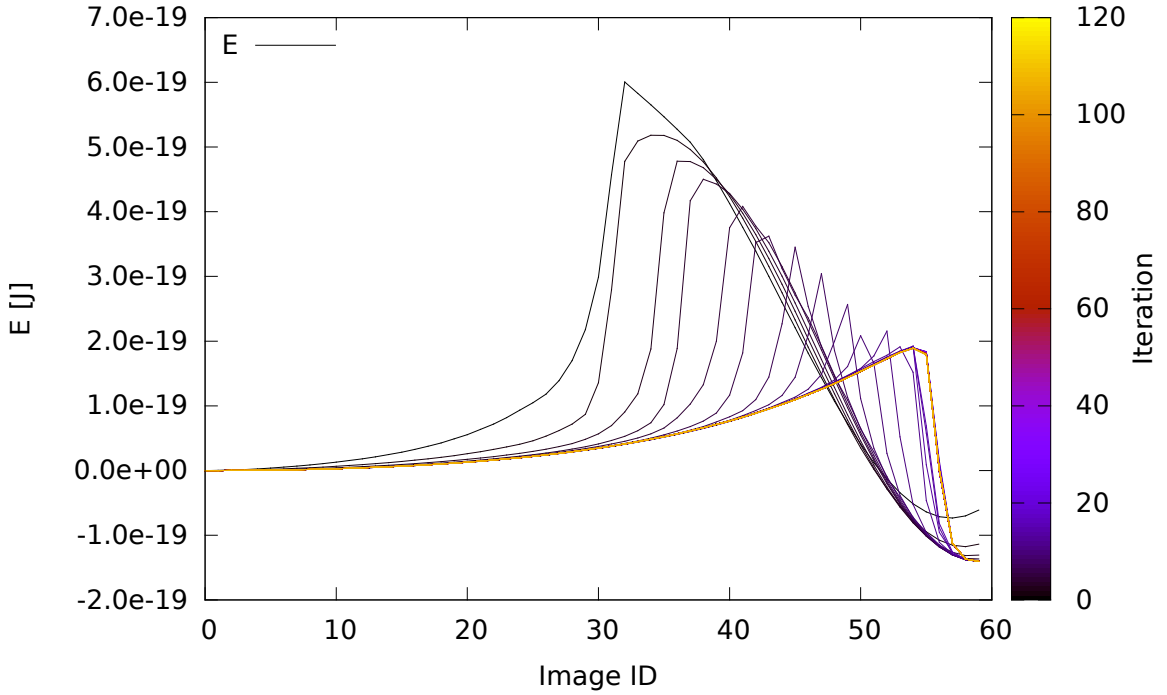


(a) Initial skyrmionic configuration obtained with the parameters given in Table 5.2. The transparent arrows indicate the magnetisation direction, the colour the magnitude of m_z.

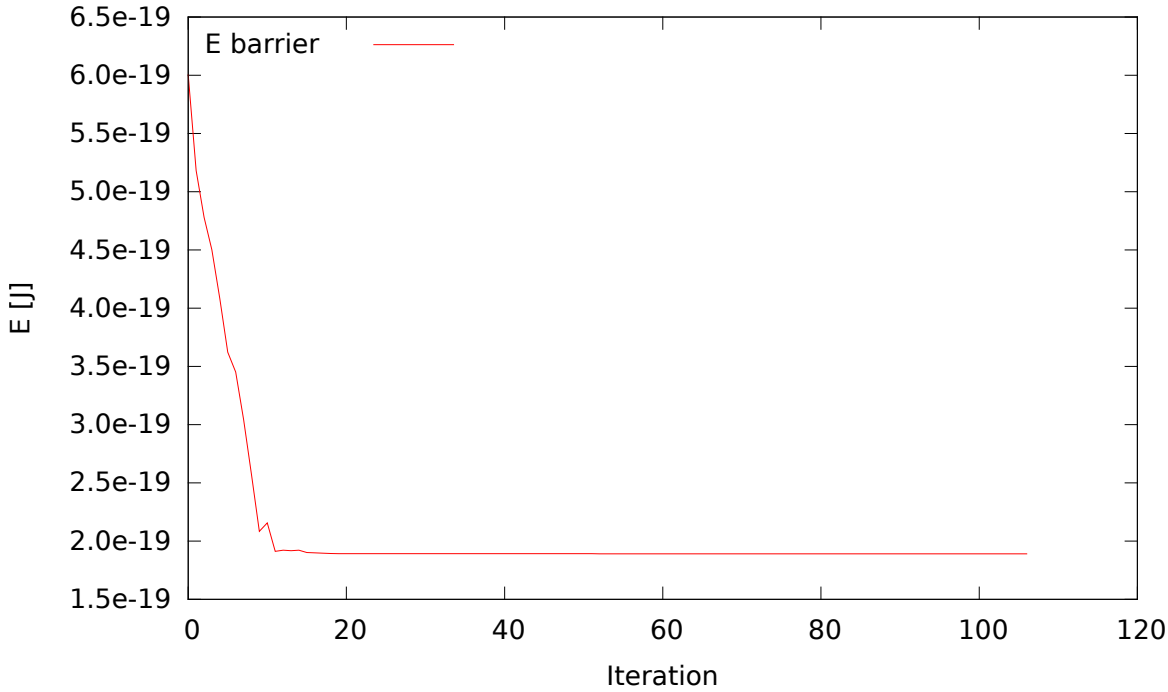


(b) Energy curve of the converged string yielding a barrier of 1.891×10^{-19} J.

Figure 5.6: Initial skyrmion image (a) and converged energy curve (b). The corresponding energy of the magnetic configuration (a) is set to zero.



(a) Energy curve of the string, plotted for each iteration.



(b) Energy barrier as a function of the iteration.

Figure 5.7: Energy curves (a) and energy barrier (b) of the string for each iteration.

5.3 Limitations of the Micromagnetic Model

In the previous section, we presented the results of the energy barrier calculation for a magnetic thin-film with parameters according to Table 5.2 yielding a skyrmion annihilation energy of 1.891×10^{-19} J. In the following, we compare this result to micromagnetic simulations and calculate the according micromagnetic interaction energies by the relations given in Table 2.1. A requirement for the validity of the continuum assumption is that the results converge against a solution when the discretizaiton is decreased and, in the limit, the approximation should converge against the true solution.

In the following simulation, however, we show that the micromagnetic model is not capable of describing skyrmion destruction mechanisms (if they are not mediated by a boundary), even though the similar formulation of micromagnetics and the atomistic model would suggest otherwise. Under a continuum description of the magnetisation field, as is assumed in micromagnetism, skyrmion destruction mechanism not mediated by a boundary are forbidden due to the topological protection of the skyrmion and estimates of the energy barriers will depend on the numerical discretization [4].

Figure 5.8 shows the energy barriers calculated with the atomistic code (green, fixed cell size) and the micromagnetic code (red) as a function of the cell size. When we compare the atomistic simulation to the micromagnetic simulation with the corresponding cell size (i.e. $a = 0.2715$nm), we encounter a significant difference in the magnitude of the energy barrier and the values differ by a factor of ~ 2.2. This discrepancy is notable considering the closely related models. Furthermore, the energy barrier calculated by the micromagnetic code diverges for smaller cell sizes. This indicates that the model's continuum assumption is no longer valid and that the model is not capable of describing such skyrmion annihilation processes.

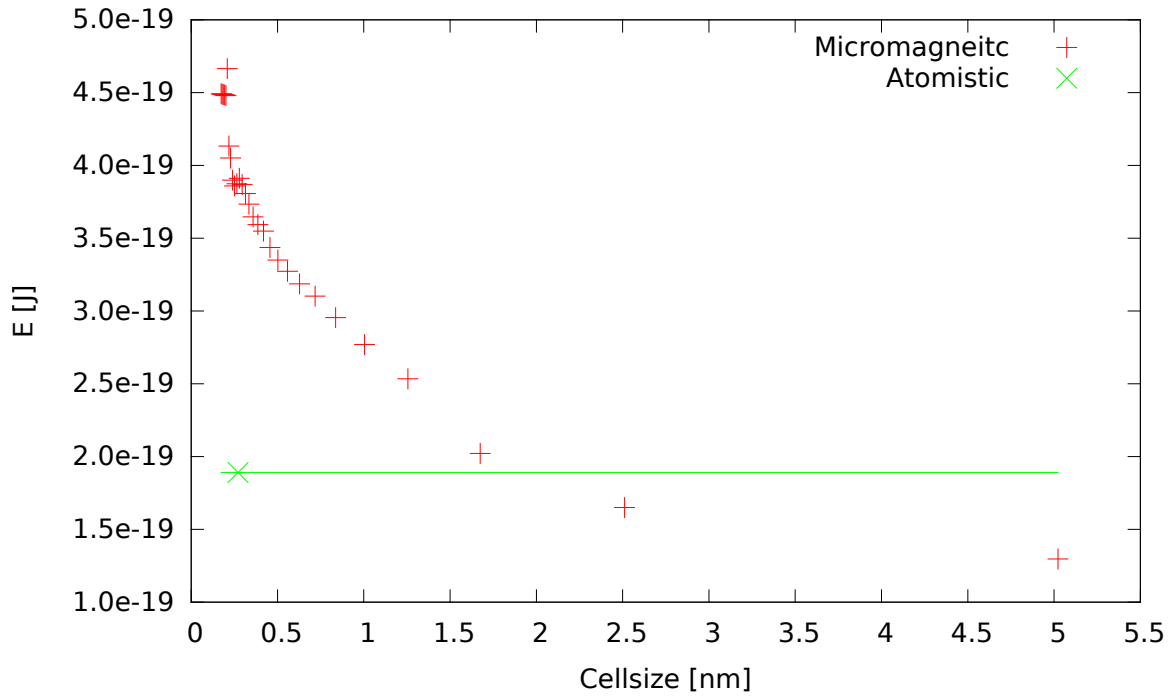


Figure 5.8: Atomistic and micromagnetic energy barriers for parameters given in Table 5.2. The interaction energies and the saturation magnetisation for the micromagnetic simulations are obtained using the relations of Table 2.1.

5.4 Phase Diagram

Based on the atomistic simulation with the parameters given in Table 5.2, we perform a variation of the interaction energies for both the DM interaction and the uniaxial-anisotropy. The resulting energy barriers for these simulations are shown in the phase diagram of Figure 5.9. A selection of five initial skyrmionic configurations is displayed and indicates their corresponding position on the phase diagram by arrows. Figure 5.10 shows three initial configurations for various values of d and the corresponding energy curves per iteration. The magnetisation direction is indicated by transparent arrows.

The colour coding in Figure 5.9 represents the energy barrier found for the respective set of parameters. When we perform a variation of the parameters k and d, we encounter two opposing trends: The higher the uniaxial-anisotropy constant k, the more energetically favourable are spins directed in parallel or antiparallel to the uniaxial vector, which is – in our case – out-of-plane. On the other hand, the DM interaction prefers spins to tilt with respect to each other. Therefore higher values of d favour the creation of domain-walls and thus skyrmions.

The black area in Figure 5.9 corresponds to vanishing energy barriers which indicate that no stable skyrmionic configuration is obtained for this choice of parameters. In this case, the high anisotropy prevents the formation of a skyrmion and the initial image is a homogeneous configuration. This results in an energy barrier of zero.

For increasing values of the DM interaction constant d, we observe the formation of skyrmions in the initial image and encounter an annihilation via a Bloch point. When going from small values of d to higher values, these skyrmion formations start earlier for smaller values of k due to the two opposing effects discussed above. We identify straight lines of equal energies in the phase diagram indicating a linear relation.

Simulations performed with very high values of d and comparable low values of k – which lie within the dashed rectangle in the phase diagram – do not describe skyrmion annihilations. For these parameters, the initial homogeneous configuration used for point B in the string method is no longer a local energetic minimum and evolves to a skyrmionic configuration itself. From this it follows that the energy barrier for these simulations does no longer refer to the annihilation of a skyrmion via a Bloch point but yields the barrier between two skyrmionic field configurations. Therefore, these energy-barrier values cannot be directly compared to the other values of the phase diagram which do represent skyrmion annihilations.

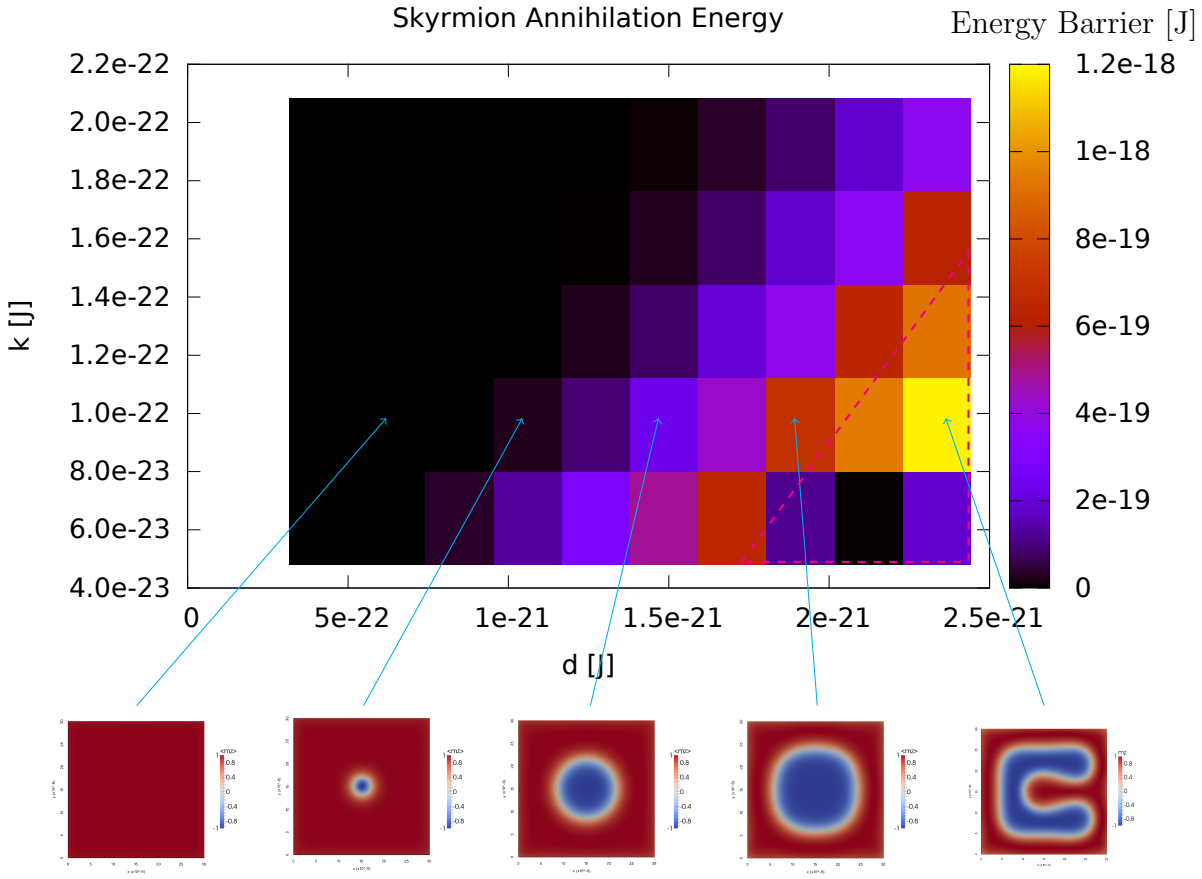
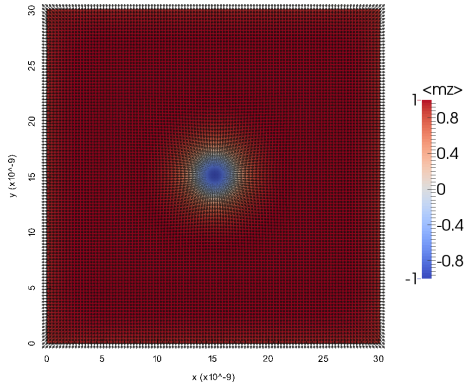
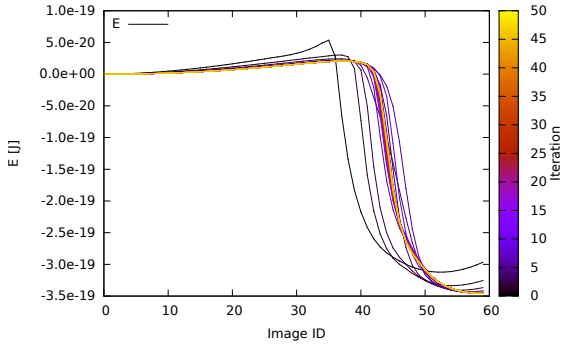


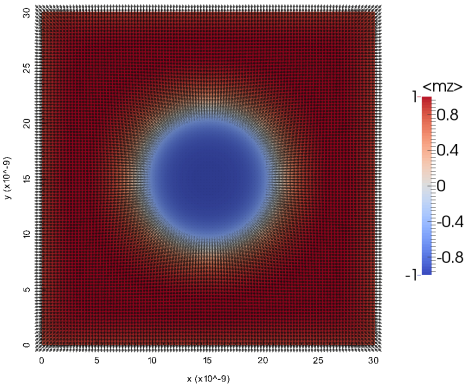
Figure 5.9: Phase diagram showing the magnitude of the energy barrier as a function of the atomistic parameters k and d. The simulation parameters are given in Table 5.2. Five selected magnetisation configurations corresponding to the first string image are displayed and indicate their position in the phase diagram. Values within in dashed rectangle do not refer to skyrmion annihilations.



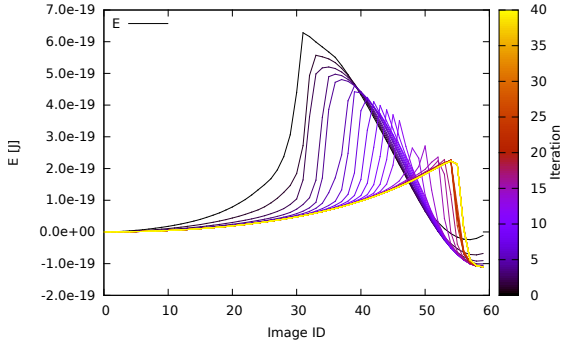
(a) Initial image for $d = 1.0615e \times 10^{21}$ and $k = 9.6062 \times 10^{23}$.



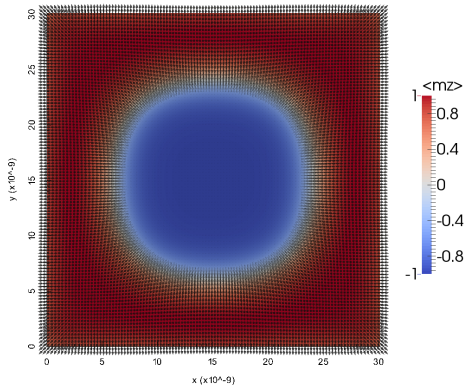
(b) Energy curves corresponding to image (a) for each iteration yielding a barrier of 2.0633×10^{-20} J.



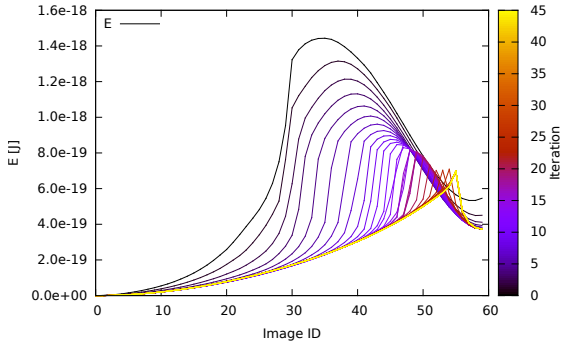
(c) Initial image for $d = 1.4860 \times 10^{21}$ and $k = 9.6062 \times 10^{23}$.



(d) Energy curves corresponding to image (c) for each iteration yielding a barrier of 2.2297×10^{-19} J.



(e) Initial image for $d = 1.9106 \times 10^{21}$ and $k = 9.6062 \times 10^{23}$.



(f) Energy curves corresponding to image (e) for each iteration yielding a barrier of 6.9805×10^{-19} J.

Figure 5.10: Several initial skyrmion configurations and the corresponding energy curves as function of the image ID for each iteration.

6 Conclusion

In this thesis we present the GPU-accelerated simulation software *pth-mag* which is developed for the simulation of spin dynamics. The current modules include an atomistic solver and a micromagnetic solver for the Landau–Lifshitz–Gilbert equation considering dipole-dipole/demagnetisation, exchange, anisotropy and Dzyaloshinskii-Moriya interactions. The mathematical models are introduced in chapter 2 and the discrete numerical methods in order to solve for these models are presented in chapter 3. We discuss finite difference methods for the numerical solution of differential equations, discrete convolutions for the calculation of the local interaction terms, fast convolutions using Fourier-methods for an efficient calculation of global interactions and embedded Runge-Kutta methods with adaptive step-size control for explicit time integration. We point out several software design considerations in chapter 4 and present simulations of the well-known μMAG standard problem #4 which are in perfect agreement with simulations obtained by the software magnum.fd. Moreover, an implementation of the string method is provided in the software package to allow for the calculation of minimum-energy-paths between two arbitrary magnetic configurations (c.f. section 3.4). In chapter 5 we use this routine in the calculation of energy barriers of magnetic skyrmion annihilations mediated by a Bloch point and find good agreement with reference calculations. We show the inapplicability of the micromagnetic model for these annihilation processes and present several calculations of energy barriers for various magnetic parameters.

Acknowledgement

At this point I would especially like to thank

PD. Dr. Dieter Süss for supervising this thesis, mastering all administrative issues, providing guidance and spreading a lot of motivation.

Dr. Claas Willem Abert for answering all of my sheer endless questions and being a wonderful mentor.

Dr. Florian Bruckner and Dr. Christoph Vogler for all on- and off-topic discussions in the office.

The entire team for providing a very pleasant working atmosphere.

The Christian Doppler Laboratory for their financial support.

My parents and my brothers for their sincere support and for being *home* for me.

My friends for their delightful company.

Julia for being the most wonderful person.

Bibliography

[1] João Sampaio Albert Fert, Vincent Cros. Skyrmions on the track. *Nature Nanotechnology*, 8:152–156, 2013.

[2] J. Sampaio, V. Cros, S. Rohart, A. Thiaville, and A. Fert. Nucleation, stability and current-induced motion of isolated magnetic skyrmions in nanostructures. *Nature Nanotechnology*, 8:839–844, 2013.

[3] Niklas Romming, Christian Hanneken, Matthias Menzel, Jessica E. Bickel, Boris Wolter, Kirsten von Bergmann, André Kubetzka, and Roland Wiesendanger. Writing and deleting single magnetic skyrmions. *Science*, 341(6146):636–639, 2013.

[4] David Cortés-Ortuño, Weiwei Wang, Marijan Beg, Ryan A. Pepper, Marc-Antonio Bisotti, Rebecca Carey, Mark Vousden, Thomas Kluyver, Ondrej Hovorka, and Hans Fangohr. Thermal stability and topological protection of skyrmions in nanotracks. *Scientific Reports*, 7, 2017.

[5] T.H.R. Skyrme. A unified field theory of mesons and baryons. *Nuclear Physics*, 31:556 – 569, 1962.

[6] M. Bode1, M. Heide, K. von Bergmann, P. Ferriani, S. Heinze1, G. Bihlmayer, A. Kubetzka1, O. Pietzsch, S. Blügel, and R. Wiesendanger. Chiral magnetic order at surfaces driven by inversion asymmetry. *Nature*, 447:190–193, 2007.

[7] S. Mühlbauer, B. Binz, F. Jonietz, C. Pfleiderer, A. Rosch, A. Neubauer, R. Georgii, and P. Böni. Skyrmion lattice in a chiral magnet. *Science*, 323(5916):915–919, 2009.

[8] X. Z. Yu, Y. Onose, N. Kanazawa, J. H. Park, J. H. Han, Y. Matsui, N. Nagaosa, and Y. Tokura. Real-space observation of a two-dimensional skyrmion crystal. *Nature*, 465:901–904, 2010.

[9] Junichi Iwasaki, Masahito Mochizuki, and Naoto Nagaosa. Universal current-velocity relation of skyrmion motion in chiral magnets. *Nature Communications*, 4, 2013.

[10] Stefan Heinze, Kirsten von Bergmann, Matthias Menzel, Jens Brede, André Kubetzka, Roland Wiesendanger, Gustav Bihlmayer, and Stefan Blügel. Spontaneous atomic-scale magnetic skyrmion lattice in two dimensions. *Nature Physics*, 7:713–718, 2011.

[11] R.P. Feynman, R.B. Leighton, and M. Sands. *The Feynman Lectures on Physics, Vol. II: The New Millennium Edition: Mainly Electromagnetism and Matter*. Feynman Lectures on Physics. Basic Books, 2011.

[12] Ernst Ising. Beitrag zur theorie des ferromagnetismus. *Zeitschrift für Physik*, 31(1):253–258, Feb 1925.

[13] R. E. Watson, M. Blume, and G. H. Vineyard. Spin motions in a classical ferromagnet. *Phys. Rev.*, 181:811–823, May 1969.

[14] R. H. Kodama and A. E. Berkowitz. Atomic-scale magnetic modeling of oxide nanoparticles. *Phys. Rev. B*, 59:6321–6336, Mar 1999.

[15] C. Mitsumata, A. Sakuma, and K. Fukamichi. Mechanism of the exchange-bias field in ferromagnetic and antiferromagnetic bilayers. *Phys. Rev. B*, 68:014437, Jul 2003.

[16] B Skubic, J Hellsvik, L Nordström, and O Eriksson. A method for atomistic spin dynamics simulations: implementation and examples. *Journal of Physics: Condensed Matter*, 20(31):315203, 2008.

[17] L. Szunyogh, B. Lazarovits, L. Udvardi, J. Jackson, and U. Nowak. Giant magnetic anisotropy of the bulk antiferromagnets irmn and $irmn_3$ from first principles. *Phys. Rev. B*, 79:020403, Jan 2009.

[18] Thomas Jourdan, Alain Marty, and Frédéric Lançon. Multiscale method for heisenberg spin simulations. *Phys. Rev. B*, 77:224428, Jun 2008.

[19] R F L Evans, W J Fan, P Chureemart, T A Ostler, M O A Ellis, and R W Chantrell. Atomistic spin model simulations of magnetic nanomaterials. *Journal of Physics: Condensed Matter*, 26(10):103202, 2014.

[20] Matteo Stifano. Skyrmion manipulation in ultrathin films by modification of exchange interactions. Master's thesis, Radbound Universiteit, 2017.

[21] A. Hubert and R. Schaefer. *Magnetic Domains: The Analysis of Magnetic Microstructures*. Springer, 1998.

[22] I. Dzyaloshinsky. A thermodynamic theory of "weak" ferromagnetism of antiferromagnetics. *Journal of Physics and Chemistry of Solids*, 4(4):241 – 255, 1958.

[23] Tôru Moriya. Anisotropic superexchange interaction and weak ferromagnetism. *Phys. Rev.*, 120:91–98, Oct 1960.

[24] S. Rohart and A. Thiaville. Skyrmion confinement in ultrathin film nanostructures in the presence of dzyaloshinskii-moriya interaction. *Phys. Rev. B*, 88:184422, Nov 2013.

[25] A. Fert and Peter M. Levy. Role of anisotropic exchange interactions in determining the properties of spin-glasses. *Phys. Rev. Lett.*, 44:1538–1541, Jun 1980.

[26] Ahmed M. Mkadmh, Alan Hinchliffe, and Fakhr M. Abu-Awwad. Atom additive model based on dipole field tensor to compute static average molecular dipole polarizabilities. *Journal of Molecular Structure: THEOCHEM*, 901(1):9 – 17, 2009.

[27] LALE Landau and Evgeny Lifshitz. On the theory of the dispersion of magnetic permeability in ferromagnetic bodies. *Phys. Z. Sowjetunion*, 8(153):101–114, 1935.

[28] W. Brown. Thermal fluctuation of fine ferromagnetic particles. *IEEE Transactions on Magnetics*, 15(5):1196–1208, Sep 1979.

[29] Claas Abert, Florian Bruckner, Christoph Vogler, Roman Windl, Raphael Thanhoffer, and Dieter Suess. A full-fledged micromagnetic code in fewer than 70 lines of numpy. *Journal of Magnetism and Magnetic Materials*, 387:13 – 18, 2015.

[30] Andrew J. Newell, Wyn Williams, and David J. Dunlop. A generalization of the demagnetizing tensor for nonuniform magnetization. *Journal of Geophysical Research: Solid Earth*, 98(B6):9551–9555, 1993.

[31] Weinan E, Weiqing Ren, and Eric Vanden-Eijnden. Simplified and improved string method for computing the minimum energy paths in barrier-crossing events. *The Journal of Chemical Physics*, 126(16):164103, 2007.

[32] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes 3rd Edition: The Art of Scientific Computing*. Cambridge University Press, New York, NY, USA, 3 edition, 2007.

[33] Claas Willem Abert. *Discrete Mathematical Concepts in Micromagnetic Computations*. PhD thesis, Universität Hamburg, 2013.

[34] G.T. Rado and J.R. Weertman. Spin-wave resonance in a ferromagnetic metal. *Journal of Physics and Chemistry of Solids*, 11(3):315 – 333, 1959.

[35] J. C. Butcher. *Numerical Methods for Ordinary Differential Equations*. John Wiley & Sons, Ltd, Chichester, UK, 2008.

[36] E Fehlberg. Low-order classical runge-kutta formulas with stepsize control and their application to some heat transfer problems. Nasa technical report 315, National Aeronautics and Space Administration, 1969.

[37] L.F. Shampine. *Numerical Solution of Ordinary Differential Equations*. Number v. 4 in Chapman & Hall mathematics. Taylor & Francis, 1994.

[38] J.R. Dormand and P.J. Prince. A family of embedded runge-kutta formulae. *Journal of Computational and Applied Mathematics*, 6(1):19 – 26, 1980.

[39] P. Bogacki and L.F. Shampine. A 3(2) pair of runge - kutta formulas. *Applied Mathematics Letters*, 2(4):321 – 325, 1989.

[40] A. Ralston and P. Rabinowitz. *A First Course in Numerical Analysis*. International series in pure and applied mathematics. McGraw-Hill, 1978.

[41] P.J. Prince and J.R. Dormand. High order embedded runge-kutta formulae. *Journal of Computational and Applied Mathematics*, 7(1):67 – 75, 1981.

[42] J. R. Cash and Alan H. Karp. A variable order runge-kutta method for initial value problems with rapidly varying right-hand sides. *ACM Trans. Math. Softw.*, 16(3):201–222, September 1990.

[43] P. Bogacki and L.F. Shampine. An efficient runge-kutta (4,5) pair. *Computers and Mathematics with Applications*, 32(6):15 – 28, 1996.

[44] L.F. Shampine R.W. Brankin, I. Gladwell. Rksuite release 1.0. `http://www.netlib.org/ode/rksuite/`, November 1991. Accessed on: 2017-05-08.

[45] Weinan E, Weiqing Ren, and Eric Vanden-Eijnden. String method for the study of rare events. *Phys. Rev. B*, 66:052301, Aug 2002.

[46] Alexander D. Wentzell Mark I. Freidlin. *Random Perturbations of Dynamical Systems, 2nd edition*. Springer, New York, 1998.

[47] Nvidia homepage. `http://www.nvidia.com/object/cuda_home_new.html`. Accessed on: 2017-09-04.

[48] Pavan Yalamanchili, Umar Arshad, Zakiuddin Mohammed, Pradeep Garigipati, Peter Entschev, Brian Kloppenborg, James Malcolm, and John Melonakos. ArrayFire - A high performance software library for parallel computing with an easy-to-use API. `https://github.com/arrayfire/arrayfire`, 2015.

[49] Micromagnetic modeling activity group. `http://www.ctcms.nist.gov/~rdm/org.html`. Accessed on: 2017-09-05.

[50] magnum.fd homepage. `http://micromagnetics.org/magnum.fd/`. Accessed on: 2017-09-06.

[51] Y. Zhou, E. Iacocca, A. A. Awad, R. K. Dumas, F. C. Zhang, and H. B. Braun and J. Åkerman. Dynamically stabilized magnetic skyrmions. *Nature Communications*, 6(8193), 2015.