Review

Numerical methods for the stray-field calculation: A comparison of recently developed algorithms

Claas Abert [a,b,*], Lukas Exl [c], Gunnar Selke [b,d], André Drews [b,d], Thomas Schrefl [c]

[a] Fachbereich Mathematik, Universität Hamburg, Bundesstr. 55, 20146, Hamburg, Germany
[b] Institut für Angewandte Physik und Zentrum für Mikrostrukturforschung, Universität Hamburg, Jungiusstr. 11, 20355 Hamburg, Germany
[c] University of Applied Sciences, Department of Technology, A-3100 St.Poelten, Austria
[d] Arbeitsbereich Technische Informatiksysteme, Universität Hamburg, Vogt-Kölln-Str. 30, 22572 Hamburg, Germany

ARTICLE INFO

ABSTRACT

Different numerical approaches for the stray-field calculation in the context of micromagnetic simulations are investigated. We compare *finite difference based fast Fourier transform methods*, *tensor-grid methods* and the *finite-element method with shell transformation* in terms of computational complexity, storage requirements and accuracy tested on several benchmark problems. These methods can be subdivided into integral methods (fast Fourier transform methods, tensor-grid method) which solve the stray field directly and in differential equation methods (finite-element method) which compute the stray field as the solution of a partial differential equation. It turns out that for cuboid structures the integral methods, which work on cuboid grids (fast Fourier transform methods and tensor-grid methods), outperform the finite-element method in terms of the ratio of computational effort to accuracy. Among these three methods the tensor-grid method is the fastest for a given spatial discretization. However, the use of the tensor-grid method in the context of full micromagnetic codes is not well investigated yet. The finite-element method performs best for computations on curved structures.

1. Introduction

Micromagnetic simulations nowadays are highly important for the investigation of ferromagnetic materials which are used in storage systems and electric motors and generators. In these simulations the magnetic state of the ferromagnet is represented by a classical magnetization vector field.

The computation of the non-local magnetostatic interactions is the most time-consuming part of micromagnetic simulations. Naive implementation of the superposition-based integral operators (5) or solvers for the underlying differential equation (Poisson equation (3)) yield computational costs proportional to the square of the number of grid points, i.e. $\mathcal{O}(N^2)$. Several methods have been introduced in the literature in order to reduce these costs.

The magnetic scalar potential can be computed by solving the Poisson equation. The solution of the Poisson equation with the finite-element method (FEM) has a complexity of $\mathcal{O}(N)$ if boundary conditions are given at the boundary of the sample and a multigrid preconditioner is used [1]. However, the stray-field problem has open boundary conditions, where the potential is known at infinity. Two possible solutions for the open boundary problem are the coupling of the boundary element method (BEM) with the finite-element method [2] and the application of a shell transformation [3]. BEM gives an additional complexity of $\mathcal{O}(M^2)$ where M is the number of boundary nodes. This complexity can be reduced to $\mathcal{O}(M \log M)$ by application of the $\mathcal{H}$-matrix approximation for the dense and unstructured boundary element matrices [4–6]. The storage requirements and computational complexity of the FEM with shell transformation will be described in the forthcoming text.

Another class of methods rely on the evaluation of volume and/or surface integrals for the direct computation of the magnetostatic potential or the field, e.g. fast multipole methods [7,8], nonuniform grid methods [9] and fast Fourier transform (FFT) methods [10,11], scaling from $\mathcal{O}(N)$ to $\mathcal{O}(N \log N)$. The more recent tensor-grid method (TG), which also belongs to this class scales even better under certain assumptions.

In this paper we compare recently developed algorithms, namely the FFT-based methods for the computation of the field via the scalar potential (SP) and directly (DM) [10,12], a recently developed approach from numerical tensor-structured methods (TG) [14], and the finite-element method with shell transformation (FES), which is a FEM method that does not rely on BEM approaches and thus only introduces sparse matrices.

* Corresponding author at: Fachbereich Mathematik, Universität Hamburg, Bundesstr. 55, 20146, Hamburg, Germany.
E-mail address: cabert@physnet.uni-hamburg.de (C. Abert).

2. Stray-field problem

Consider a magnetization configuration $\boldsymbol{M}$ that is defined on a finite region $\Omega = \{\boldsymbol{r} : \boldsymbol{M}(\boldsymbol{r}) \neq 0\}$. In order to perform minimization of the full micromagnetic energy functional or solve the Landau–Lifshitz–Gilbert (LLG) equation it is necessary to compute the stray field within the finite region Ω. The stray-field energy is given by

$$e_d = -M_s \frac{1}{2} \int_\Omega \boldsymbol{M} \cdot \boldsymbol{H} \, \mathrm{d}^3 \boldsymbol{r}. \tag{1}$$

The Landau–Lifshitz–Gilbert equation reads

$$\boldsymbol{M}_t = -\frac{\gamma}{1+\alpha^2} \boldsymbol{M} \times \boldsymbol{H}_{\mathrm{eff}} + \frac{\alpha\gamma}{M_s(1+\alpha^2)} \boldsymbol{M} \times (\boldsymbol{M} \times \boldsymbol{H}_{\mathrm{eff}}), \tag{2}$$

where α is the Gilbert damping constant and $\boldsymbol{H}_{\mathrm{eff}}$ is the effective field given by the variational derivative of the energy w.r.t. the magnetization [15,16]. In both cases the stray field is only required to be known within Ω. The stray field $\boldsymbol{H}$ has a scalar potential ϕ, which is the solution of a Poisson equation [17]

$$\boldsymbol{H} = -\nabla\phi, \tag{3}$$

$$\Delta\phi = \nabla \cdot \boldsymbol{M}. \tag{4}$$

The stray field $\boldsymbol{H}$ and thus also the scalar potential ϕ are required to vanish at infinity. This boundary condition is often referred to as open boundary condition [18].

3. Methods

3.1. FFT Methods (SP and DM)

One way to reduce the computational complexity is to employ the fast Fourier transform (FFT). FFT methods solve an integral solution of the Poisson equation by applying the convolution theorem. The solution to the Poisson equation (3) is given by the integral, see [17],

$$\phi(\boldsymbol{r}) = -\frac{1}{4\pi} \int_\Omega \frac{\nabla' \cdot \boldsymbol{M}(\boldsymbol{r}')}{|\boldsymbol{r}-\boldsymbol{r}'|} \, \mathrm{d}^3 \boldsymbol{r}' + \frac{1}{4\pi} \int_{\partial\Omega} \frac{\boldsymbol{n}' \cdot \boldsymbol{M}(\boldsymbol{r}')}{|\boldsymbol{r}-\boldsymbol{r}'|} \, \mathrm{d}A', \tag{5}$$

which directly fulfills the required open boundary condition. Performing integration by parts yields

$$\phi(\boldsymbol{r}) = \frac{1}{4\pi} \int_\Omega \boldsymbol{M}(\boldsymbol{r}') \cdot \nabla' \frac{1}{|\boldsymbol{r}-\boldsymbol{r}'|} \mathrm{d}^3 \boldsymbol{r}' \tag{6}$$

$$\phi(\boldsymbol{r}) = \boldsymbol{S}(\boldsymbol{r}-\boldsymbol{r}') * \boldsymbol{M}(\boldsymbol{r}'). \tag{7}$$

By employing the convolution theorem

$$\phi = \boldsymbol{S} * \boldsymbol{M} = \mathcal{F}^{-1}(\mathcal{F}(\boldsymbol{S}) \cdot \mathcal{F}(\boldsymbol{M})), \tag{8}$$

this convolution can be discretized and solved with the fast Fourier transform. A prerequisite for this procedure is the usage of an equidistant grid, which is required for a discrete convolution. The stray field

$$\boldsymbol{H}(\boldsymbol{r}) = -\boldsymbol{\nabla}\phi(\boldsymbol{r}), \tag{9}$$

can be obtained by applying finite differences. This method is referred to as the scalar-potential method (SP) in the following. It is described in detail in [10].

It is also possible to compute the stray field $\boldsymbol{H}$ directly as a result of a matrix–vector convolution.

$$\boldsymbol{H}(\boldsymbol{r}) = -\frac{1}{4\pi} \int \left(\boldsymbol{\nabla}\boldsymbol{\nabla}' \frac{1}{|\boldsymbol{r}-\boldsymbol{r}'|} \right) \boldsymbol{M}(\boldsymbol{r}') \, \mathrm{d}^3 \boldsymbol{r}' \tag{10}$$

$$\boldsymbol{H}(\boldsymbol{r}) = \boldsymbol{N}(\boldsymbol{r}-\boldsymbol{r}') * \boldsymbol{M}(\boldsymbol{r}'). \tag{11}$$

Here $\boldsymbol{N}$ denotes the demagnetization tensor. Similar to (8) the convolution can be solved as an element-wise matrix–vector multiplication in Fourier space. This method is referred to as the demagnetization-tensor method (DM) in the following and is implemented by different finite-difference codes [19,20,12]. For the numerical experiments we use MicroMagnum [12] which implements both the SP and the DM method.

3.2. Tensor grid methods (TG)

Tensor grid methods (TG) for micromagnetic stray-field computation were recently introduced in [14,21]. They were developed for the purpose of handling so called low-rank tensor or compressed tensor magnetization, see [22] for a survey, in order to accelerate the computations and relieve storage requirements, see [14]. In the following we give a brief introduction into the ideas behind this method, also see [14] for a detailed description.

3.2.1. Analytical preparations

The computation of the stray field within the magnetic body is based on the explicit integral formula for the scalar potential (6). The main idea is the usage of a representation for the integral kernel in (6) as an integral of a *Gaussian function* by the formula

$$\frac{1}{|\boldsymbol{r}-\boldsymbol{r}'|^3} = \frac{2}{\sqrt{\pi}} \int_{\mathbb{R}} \tau^2 e^{-\tau^2 |\boldsymbol{r}-\boldsymbol{r}'|^2} \, \mathrm{d}\tau, \tag{12}$$

which leads from (6) to

$$\phi(\boldsymbol{r}) = \frac{1}{2\pi^{3/2}} \int_{\mathbb{R}} \tau^2 \int_\Omega e^{-\tau^2 |\boldsymbol{r}-\boldsymbol{r}'|^2} \boldsymbol{M}(\boldsymbol{r}') \cdot (\boldsymbol{r}-\boldsymbol{r}') \, \mathrm{d}^3 \boldsymbol{r}' \, \mathrm{d}\tau. \tag{13}$$

Eq. (13) reduces the computation to independent spatial integrals along each principal direction (the part of the Ω integral without the magnetization is now a product of independent 1D integrals). This analytical preparation directly results in a reduction of the computational effort from $\mathcal{O}(N^2)$ to $\mathcal{O}(N^{4/3})$ if discretized on a *tensor-product grid* before even using compressed/low-rank tensor formats for the discretized magnetization components. A similar method was introduced for the computation of the electrostatic scalar potential [24].

The additional τ-integration is carried out by the exponentially convergent *Sinc quadrature* [25], the spatial integrals are computed by *Gauss–Legendre quadrature*, both resulting in a numerical error of about the machine epsilon.

3.2.2. Discretization on a tensor-product grid

The magnetic body Ω is discretized on a *tensor-product grid* arising from the tensor outer product of three vectors $h_p \in \mathbb{R}^{N_p}$, $p = 1 \ldots 3$ related to the grid spacings along each axis (see Fig. 1). This results in a not necessarily uniform *Cartesian grid* but in contrast to methods like DM/SP described before, tensor-grid methods make use of the tensor-product interpretation of such grids.

The magnetization on the center points of the cells is given as a 3-tensor [22] for each component, i.e.

$$\mathcal{M}^{(p)} \in \mathbb{R}^{N_1 \times N_2 \times N_3}, \quad p = 1 \ldots 3 \tag{14}$$

where N_1, N_2, N_3 are the number of cells in the principal directions. Thus it is possible to use *low-rank* representation for the magnetization like Canonical/Parallel Factors Decomposition (CP) or *Tucker formats*, see Appendix A or [22]. We obtain the potential on the center points of the computational cells, as the discrete
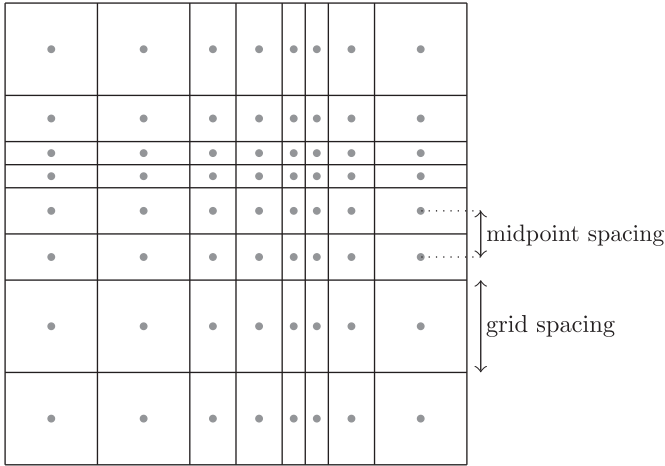
Fig. 1. Grid spacing and midpoint spacing in TG methods.

Table 1
Storage in number of floating point values w.r.t. number of computational cells/degrees of freedom N. In TG methods r denotes the tensor rank and R denotes the number of Sinc-quadrature nodes.

Method	Scratch space	Magnetization	Potential	Field
DM	$30N$	$3N$	–	$3N$
SP	$27N$	$3N$	N	$3N$
TG (dense)	$6RN^{2/3}$	$3N$	N	$3N$
TG (Tucker)	$6RN^{2/3}$	$3(r^3 + 3rN^{1/3})$	$9R(r^3 + 3rN^{1/3})$	$27R(r^3 + 3rN^{1/3})$
TG (CP)	$6RN^{2/3}$	$3(r + 3rN^{2/3})$	$9(r + 3rRN^{1/3})$	$27(r + 3rRN^{1/3})$
FES	$\approx 48N$	$3N$	N	$3N$

analogue of (13), by a so-called *block-CP tensor* [14]

$$\mathbb{R}^{N_1 \times N_2 \times N_3} \ni \Phi = \frac{1}{2\pi^{3/2}} \sum_{p=1}^{3} \sum_{l=1}^{R} \omega_l \sinh(\tau_l)^2 \, \mathcal{M}^{(p)} \times_1 \mathbf{D}_1^l \times_2 \mathbf{D}_2^l \times_3 \mathbf{D}_3^l. \quad (15)$$

Here (τ_l, ω_l) are the nodes and weights arising from the Sinc-quadrature of the integral kernel (12) with R terms, chosen as in [14], where $R = 50$ turned out to give sufficient accuracy for grids up to 8×10^6 nodes.[1] The *Gaussian matrices* $\mathbf{D}_q^l \in \mathbb{R}^{N_q \times N_q}$ come from

$$d_{i_q j_q}^l := \int_{\Omega_{j_q}} g(x_{i_q}^c, x', \tau_l) \, dx', \quad (16)$$

$$\mathbf{D}_q^l := (d_{i_q j_q}^l). \quad (17)$$

where Ω_{j_q} denotes the *j*-th interval on the (partitioned) *q*-th axis with length $(h_q)_j$ and $x_{i_q}^c$ is the midpoint of the *i*-th interval on the *q*-th axis. The Gaussian integrands are given by

$$g^{(q)}(\alpha, \alpha', \tau) := \begin{cases} \exp(-\sinh(\tau)^2(\alpha-\alpha')^2) & q \neq p, \\ (\alpha-\alpha')\exp(-\sinh(\tau)^2(\alpha-\alpha')^2) & q = p, \end{cases} \quad (18)$$

and are approximated using Gauss–Legendre quadrature, as mentioned above. The field within Ω is derived from (9) by finite-difference operators of second order.

3.2.3. Low-rank magnetization

Eq. (15) allows the treatment of specially structured magnetization tensors, like CP or Tucker tensors [22] that have a reduced number of degrees of freedom, see Table 1, and accelerates the computation up to *sub-linear* effort (below the volume size N^3), see Table 2. As a consequence TG methods using low-rank magnetization allow larger models with finer discretization density than conventional methods.

We now show by means of numerical experiments that typical single domain states [26] have highly accurate low-rank representations. Fig. 2 shows the approximation properties of a flower and a vortex state as described in Section 5 via the CP format and the Tucker format using an *alternating least squares algorithm* (ALS) [22] for the approximations. The plots 2a and b are computed independently from random initial guesses used in the ALS algorithm. We set the parameters in (30) as $a = c = 0.5$, $b = 1$ and in (31) as $r_c = 1/2$. Fig. 2(a) shows the dependence of the relative error (19) w.r.t. the

---

[1] A value of $R = 65$ was used in the experiments for much larger grids.

rank for fixed discretization density, where Fig. 2(b) indicates the dependence w.r.t. the discretization density N for fixed rank.

The relative errors are measured in the Frobenius norm, i.e.

$$relerr = \left(\sum_{p=x,y,z} \|\mathcal{M}_{dense}^{(p)} - \mathcal{M}_{low-rank}^{(p)}\|_F^2 \right)^{1/2} \Big/ \left(\sum_{p=x,y,z} \|\mathcal{M}_{dense}^{(p)}\|_F^2 \right)^{1/2}. \quad (19)$$

The Tucker format generally leads to a better approximation, where Fig. 2(b) essentially shows no loss of accuracy while increasing the mesh density.

We next perform energy minimization for the parameterized vortex to gain results on the separation rank in a more realistic situation. For this purpose we augment the micromagnetic energy with a Lagrange multiplier and penalty term for the micromagnetic constraint ($|\mathbf{m}| = 1$ a.e., where $\mathbf{m} = \mathbf{M}/M_s$) and update the multipliers with a first order rule while increasing the penalty parameter. Each subproblem is therefore unconstrained and we can address it by a limited memory BFGS quasi-Newton method, i.e.

For given multipliers-vector $\boldsymbol{\lambda} \in \mathbb{R}^N$, penalty parameter $\mu > 0$ and constraint violation $\mathbf{c}(\mathbf{m}_x, \mathbf{m}_y, \mathbf{m}_z) := \mathbf{m}_x^2 + \mathbf{m}_y^2 + \mathbf{m}_z^2 - \mathbf{1} \in \mathbb{R}^N$ the corresponding subproblem reads

$$\min_{\mathbf{m}_x, \mathbf{m}_y, \mathbf{m}_z} e_{tot}(\mathbf{m}_x, \mathbf{m}_y, \mathbf{m}_z) + \boldsymbol{\lambda}^T \mathbf{c}(\mathbf{m}_x, \mathbf{m}_y, \mathbf{m}_z) + \frac{\mu}{2} \|\mathbf{c}(\mathbf{m}_x, \mathbf{m}_y, \mathbf{m}_z)\|^2, \quad (20)$$

where $(\mathbf{m}_x^T, \mathbf{m}_y^T, \mathbf{m}_z^T)^T \in \mathbb{R}^{3N}$ denotes the discretized magnetization cosines. The multipliers are updated according to the rule

$$\boldsymbol{\lambda}_{new} = \boldsymbol{\lambda}_{old} + \mu \mathbf{c}(\mathbf{m}_x, \mathbf{m}_y, \mathbf{m}_z), \quad (21)$$

which follows by requiring the first order Karush–Kuhn–Tucker (KKT) optimality conditions. The method described above is known as *augmented Lagrangian method* or *method of multipliers* [27]. We relax the magnetization for zero external field and differing length L of the cube and two different constants for the anisotropy energy ($Q = 0.1$ (soft) and $Q = 2$ (hard) in reduced units, see [26]). The approximation rank is then calculated by ALS algorithms for a given accuracy, both for CP and for Tucker tensors. The results in Fig. 3 show again lower ranks for the Tucker case and better compression factors. Moreover it can be observed that for increasing length of the cube the ranks grow. This is reasonable particulary in view of the fact that above the single domain limit more complex structures appear. The results for soft and hard material only differ slightly.

3.3. FEM Methods (FES)

Within the finite-element framework the Poisson equation (3) is solved by the weak formulation

$$\int_{\Omega} \nabla \phi \cdot \nabla \nu \, d^3 x = \int_{\Omega} \mathbf{M} \cdot \nabla \nu \, d^3 x \quad \forall \, \nu, \phi \in V \quad (22)$$

Table 2
Computational complexity w.r.t. number of computational cells/degrees of freedom N. In TG methods r denotes the tensor product rank, where we assume the same rank for each magnetization (and mode in case of Tucker tensors) for sake of simplicity. Every column depends on its left neighbor, e.g. the calculation of the field requires the previous calculation of the potential etc.

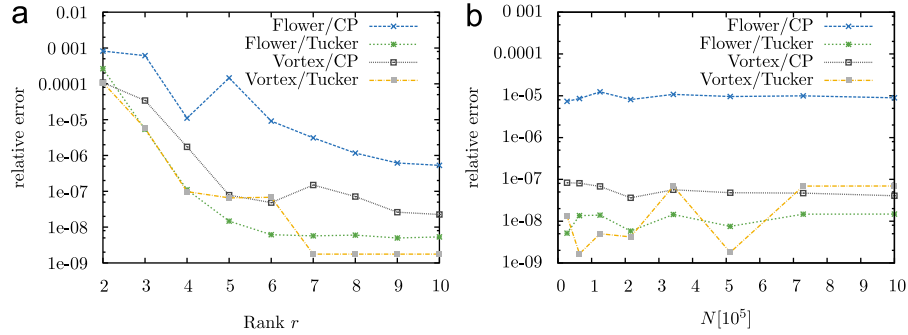| Method	Setup	Potential	Field	Energy
DM	$\mathcal{O}(N \log N)$	–	$\mathcal{O}(N \log N)$	$\mathcal{O}(N)$
SP	$\mathcal{O}(N \log N)$	$\mathcal{O}(N \log N)$	$\mathcal{O}(N)$	$\mathcal{O}(N)$
TG (dense)	$\mathcal{O}(R \sum_{j=1}^{3} N_j^2)$	$\mathcal{O}(RN \sum_{j=1}^{3} N_j)$	$\mathcal{O}(N)$	$\mathcal{O}(N)$
TG (Tucker)	$\mathcal{O}(R \sum_{j=1}^{3} N_j^2)$	$\mathcal{O}(R(\sum_{j=1}^{3} N_j^2 r + \sum_{j=1}^{3} N_j r^3))$	$\mathcal{O}(rR \sum_{j=1}^{3} N_j)$	$\mathcal{O}(r^2 R \sum_{j=1}^{3} N_j + r^4 R)$
TG (CP)	$\mathcal{O}(R \sum_{j=1}^{3} N_j^2)$	$\mathcal{O}(R \sum_{j=1}^{3} N_j^2 r)$	$\mathcal{O}(rR \sum_{j=1}^{3} N_j)$	$\mathcal{O}(r^2 R \sum_{j=1}^{3} N_j)$
FES	$\mathcal{O}(N)$	$\mathcal{O}(N \log^\alpha N), \alpha \ll 1$	$\mathcal{O}(N)$	$\mathcal{O}(N)$

Fig. 2. Low-rank approximation of flower and vortex state via Tucker and CP decomposition. (a) Relative error w.r.t. approximation rank r. $N = 1e+06$. (b) Relative error w.r.t. discretization density N. Rank $r = 5$.
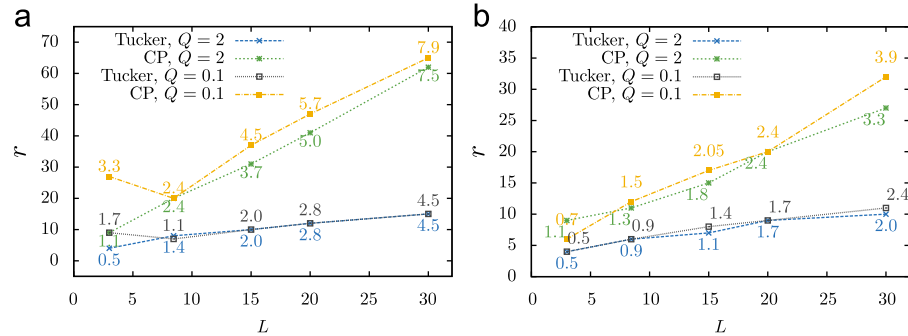

Fig. 3. Minimal tensor product and Tucker rank for the representation of a relaxed magnetic cube for a given tolerance (vortex-configuration (31) as the initial state). The anisotropy Q and the side length L of the cube are varied. The labels of the measure points give the storage in percent compared to full magnetization (100%). (a) Minimal rank for a tolerance of 1×10^{-3}, (b) minimal rank for a tolerance of 5×10^{-3}.

where Dirichlet boundary conditions are embedded in the trial function space V, i.e. the function space of the solution ϕ. In the case of the stray-field problem the boundary conditions at the sample boundary $\partial \Omega$ are unknown. They are defined as zero at infinity. The treatment of these open boundary conditions is the main difficulty for finite-element stray-field calculations.

We present the results for a transformation technique. The sample is surrounded by a finite shell which is also meshed. A bijective transformation from the finite shell to the complete exterior of the sample is applied by introducing a metric tensor to the weak formulation. The particular transformation we use is known as "parallelepipedic shell transformation" [3]. The sample is put into a cuboid volume and a shell consisting of six parallelepipeds is created (see Fig. 4a).

The transformation is chosen such that points located at the inner boundary of the shell are mapped to themselves. Points on the outer boundary of the shell are mapped to infinity. The Jacobian of the transformation is requested to be 1 on the inner boundary of the shell in order to be continuous across the sample boundary.

These conditions still leave some space for the choice of transformation. The most important aspect of this method is the distortion of the test and trial functions in the transformed area. In order to get a good result, the test and trial functions must be distorted such that they are able to model the natural decay of the magnetic potential. This obviously also depends on the choice of test and trial functions. From (6) it is seen that the potential decays with $1/r^2$. We choose our test and trial function $\phi_h, \nu_h \in V_h$ to be continuous and piecewise third-order polynomial ($\mathcal{P}_3$)

$$V_h = \{\nu_h \in H^1(\Omega) : \nu_h|_T \in \mathcal{P}_3(T) \forall T \in \mathcal{T}_h\} \tag{23}$$

where H^1 is a Sobolev space and $\mathcal{T}_h$ is a tetrahedron tesselation (see Fig. 4(b)). The transformation per shell patch is carried out in a radial sense as sketched in Fig. 5. The scalar transformation is given by

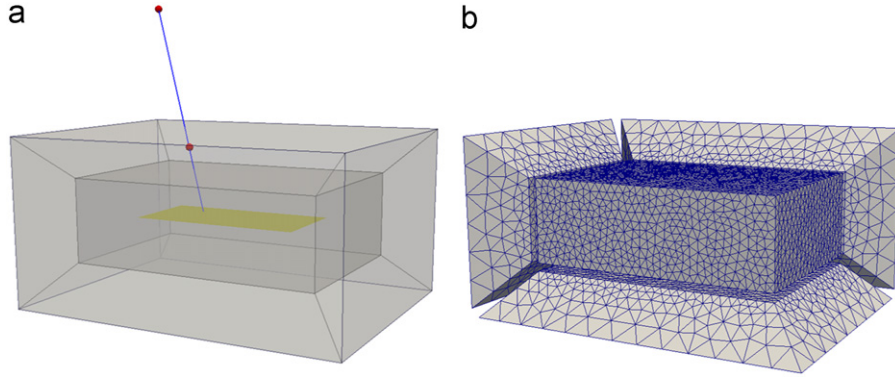$$X = R_1 \frac{R_2 - R_1}{R_2 - |x|} \tag{24}$$

Fig. 4. Parallelepipedic shell surrounding the cuboid sample. (a) The transformation is carried out along the blue line. The origin of the transformation moves along the yellow middle plane. (b) Since the area of interest is the sample, the mesh is coarsened in the shell. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this article.)
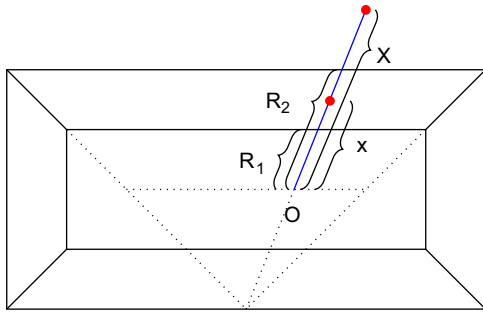

Fig. 5. Sketch of the shell transformation in two dimensions. Within each shell patch the shell points are transformed in a radial sense. In order to achieve continuity between the patches the origin O of the one-dimensional transformation has to be continuous between the patches. We choose the origin O to move on the middle plane of the sample. The third dimension is treated in the same manner.

with R_1, R_2, x, and X as shown in Fig. 5. This transforms the third order polynomial test and trial functions as

$$a + bx + cx^2 + dx^3 \rightarrow a' + b'\frac{1}{X} + c'\frac{1}{X^2} + d'\frac{1}{X^3}. \tag{25}$$

The discretized weak formulation then reads

$$\int_\Omega (\nabla \phi_h)^T \boldsymbol{g} \nabla v_h \, \mathrm{d}^3 x = \int_\Omega \boldsymbol{M} \cdot \nabla v_h \, \mathrm{d}^3 x \quad \forall v_h \in V_h, \tag{26}$$

$$\boldsymbol{g} = \begin{cases} \mathbb{1} & \text{if } \boldsymbol{x} \in \Omega_{\text{sample}}, \\ \boldsymbol{J}^{-1^T} |\boldsymbol{J}| \boldsymbol{J}^{-1} & \text{if } \boldsymbol{x} \in \Omega_{\text{shell}}, \end{cases} \tag{27}$$

where Ω_{sample} and Ω_{shell} denote the disjoint regions of the sample and the transformed shell with $\Omega_{\text{sample}} \cup \Omega_{\text{shell}} = \Omega$ and $\boldsymbol{J}$ is the Jacobian matrix of the transformation. This directly translates to a linear system of equations, where the solution vector contains the coefficients in terms of the discrete function basis. The size of this solution vector is referred to as degrees of freedom (DoF). The implementation of this method is done with FEniCS [28].

4. Storage requirements and computational complexity

The costs of the different methods are compared in terms of storage requirements and computational complexity. Tables 1 and 2 show the results. We choose N to be the number of computational cells in the case of DM, SP and TG methods. In the case of finite-element methods (FES) N refers to the number of degrees of freedom.

Besides the memory needed for the storage of the magnetization configuration and the stray field, all methods require a certain amount of extra storage for auxiliary constants. In case of the DM and SP methods this includes the convolution kernels, TG needs the one dimensional Gaussian matrices, and finite-element methods (FES) require the stiffness matrix as an auxiliary constant. These constants depend on the geometry and discretization only. This means that the computation of auxiliary constants has to be done only once for different magnetization configurations. Thus their complexity is almost irrelevant in the context of LLG computations and energy minimization. The storage requirements for these constants as well as the computational complexity of their calculation are summarized in the scratch space column and the setup column respectively.

Storage requirements for TG methods depend on the rank r used for the low-rank tensor representation of the magnetization components. Often the rank is much smaller than $N^{1/3}$, the discretization size in one spatial dimension. This makes the storage requirements for the magnetization, potential and field proportional to $rN^{1/3}$. For the setup the $(N^{1/3} \times N^{1/3})$ Gaussian matrices need to be computed and stored, thereby R in Tables 1 and 2 denotes the number of Sinc-quadrature nodes. The computational effort in TG methods also depends on the tensor format used for the representation of the magnetization, see Table 2. If the magnetization has a low-rank representation, TG methods usually reduce this complexity below the number of computational cells (*sub-linear*), making this methods the fastest available nowadays.

However, the constant in the operation count of methods using a Cartesian grid depends on the relative sizes of N_1, N_2 and N_3. In particular, for a thin film N_1 and N_2 will be closer to $N^{1/2}$ than $N^{1/3}$, e.g. in this case the costs for the dense TG method is rather $\mathcal{O}(N^{3/2})$ than $\mathcal{O}(N^{4/3})$.

The storage requirements for the other three methods are proportional to N, which is a result of the dense representation of the magnetization, see Table 1. For both the DM and the SP method an acyclic convolution needs to be computed. Thus the inputs of the Fourier transform in (8) have to be zero-padded in each spatial dimension [29]. Since the demagnetization tensor $\boldsymbol{N}$ is a symmetric 3×3 matrix in each point the storage requirement amounts to $6 \cdot 2^3 N = 48N$. However all components of the tensor field are functions with even or odd properties in all spatial dimensions. These symmetries carry over to Fourier space and lead to noncomplex frequencies, which reduces the storage requirement to $6N$. Additional storage is also required for the zero-padded magnetization. In a straightforward implementation this amounts to $3 \cdot 2^3 N = 24N$. The same value holds in Fourier space, since the real valued magnetization is complex in Fourier space, but is conjugate-symmetric with respect to one axis. By interleaving the FFT, the multiplication and the inverse FFT in (8) as described in [30], the amount of storage can be reduced, but

this technique is not considered here. The same considerations apply to the SP method, which leads to a scratch storage requirement of $30N$ for the DM method and $27N$ for the SP method.

A well-known result is the $N \log N$ complexity of the convolution in FFT methods (DM/SP), likewise this is the asymptotic operation count for those methods, Table 2. In the FES method sparse linear systems have to be solved for the computation of the scalar potential. We used a conjugate gradient solver (CG) with an algebraic multigrid preconditioner (AMG) and measured the complexity w.r.t. N (DoF) experimentally, finding a linear dependence on the system size (with a small logarithmic scaling factor).

5. Numerical experiments

5.1. Homogeneously magnetized cube

As a first benchmark we take a homogeneously magnetized unit cube and compute the magnetostatic energy for varying grid-size N, where the exact value is $e_d = 1/6\,[\mu_0 M_s^2]$. Table 3 shows for each of the described methods the relative errors in the energy w.r.t. the exact value and the relative error in the field computed by (28), as well as the angular deviation (error in the field-angle) to a reference field. This reference field is computed with the DM method with quadruple precision in every spatial dimension and then averaged on the $40 \times 40 \times 40$ Cartesian grid. For the comparison the same grid is used for the DM, SP and TG methods. The results of the IFE method are interpolated on the Cartesian grid with a simple oversampling method. We take the relative l_2- error as a measurement for the field-error, i.e.

$$relerr = \left(\frac{1}{N} \sum_{p=x,y,z} \| \boldsymbol{H}_{\mathrm{demag}}^{(p)} - \boldsymbol{H}_{\mathrm{method}}^{(p)} \|_F^2 \right)^{1/2}. \tag{28}$$

The errors in the field-angle in Table 3 mostly occur at the edges of the cube. We further mention that for full micromagnetic simulations the system evolution depends not on the field

Table 3
Errors for the homogeneously magnetized unit cube: relative error of the energy, relative error of the field/field-angle (w.r.t. $160 \times 160 \times 160$ calculations with the DM method).

Method	N	relerr e	relerr h	av. relerr h [deg.]	max. err h [deg.]
DM	$40 \times 40 \times 40$	2.9e−09	0	0	0
SP	$40 \times 40 \times 40$	1.1e−03	1.1e−03	2.3e−05	5.0e+00
TG (CP $r=1$)	$40 \times 40 \times 40$	3.8e−04	2.3e−03	6.9e−06	2.5e+00
FES	7.2e+04	8.6e−04	2.2e−03	3.2e−05	5.2e+00

directly but rather on the torque, i.e. $\boldsymbol{M} \times \boldsymbol{H}$. It is easily seen that for the normalized magnetization $\boldsymbol{m}$ there holds pointwise

$$\| \boldsymbol{m} \times \boldsymbol{H} - \boldsymbol{m} \times \boldsymbol{H}_{\mathrm{exact}} \| \leq \| \boldsymbol{H} - \boldsymbol{H}_{\mathrm{exact}} \|. \tag{29}$$

Therefore the field errors given below are upper bounds for the torque error.

Fig. 8(a) shows magnetostatic energy calculations for different spatial discretizations. The DM method is almost exact and does not depend on spatial discretization. The reason is that the discretized demagnetization tensor is computed assuming homogeneously magnetized computational cells. Also the resulting stray field is analytically averaged per cell. Since the energy calculation is bilinear in the magnetization $\boldsymbol{M}$ and the stray field $\boldsymbol{H}$, the error is a pure rounding error.

The FES method is the slowest converging method for this problem. A possible reason is the large external stray field of this setup. The numerical integration of the diverging metric tensor $\boldsymbol{g}$ leads to an underestimation of the external space and consequently to an underestimation of the magnetic potential in the sample. Thus the FES method is particularly sensitive to setups with large external stray fields.

SP and TG methods are also based on the computation of the scalar potential, whereby the field is obtained by finite differences. In [14] it is shown that the TG method essentially computes the scalar potential exactly for piecewise constant magnetization. The error in the energy in both methods (SP and TG) is mostly caused by numerical approximation of the gradient in the field computation, whereas TG shows the better approximation properties for this problem. In addition to it, TG uses an exact rank-1 representation for the uniform magnetization which makes the computation sub-linear (namely $\mathcal{O}(N^{2/3})$) with small scaling factor and allows computations for dozens of millions cells without any problems related to storage and computational cost.

5.2. Flower and vortex state in a cube

We do the same comparison as in Section 5.1 for the flower state, see (30) and Fig. 6(b), and Table 4 for the results. The main magnetization direction is taken to be along the z-axis, and the flower is obtained through an in-plane perturbation along the y-axis and an out-of-plane perturbation along the x-axis. Assuming polynomial expressions for the perturbations, as in [31], our flower is the normalized version of

$$m_x(r) = \frac{1}{a} xz,$$
$$m_y(r) = \frac{1}{c} yz + \frac{1}{b^3} y^3 z^3,$$
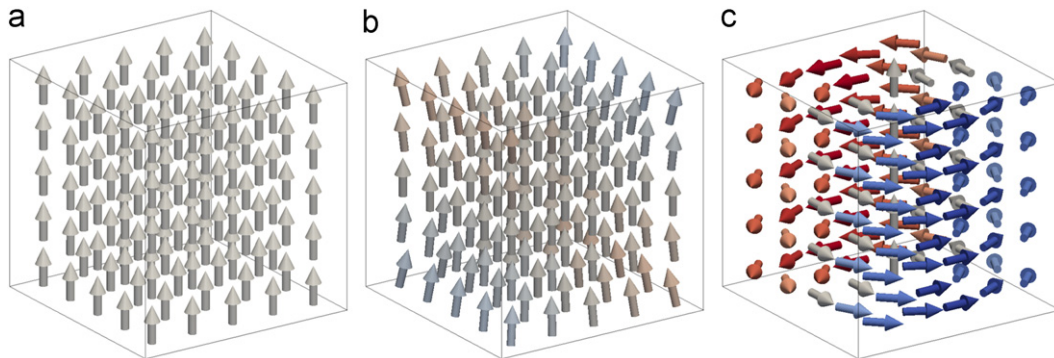$$m_z(r) = 1, \tag{30}$$

Fig. 6. Magnetization configurations in a $1 \times 1 \times 1$ cube used for numerical experiments. The magnetization is normalized, its direction is color coded. (a) Homogeneous magnetization. (b) Flower state. (c) Vortex state.

Table 4
Errors for the flower state in a unit cube: relative error of the energy, relative error of the field/field-angle (w.r.t. $160 \times 160 \times 160$ calculations with the DM method).

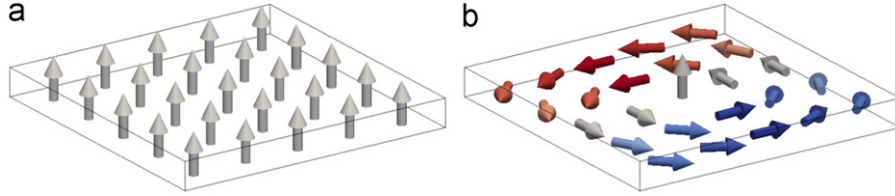| Method	N	e	relerr h	av. relerr h [deg.]	max. err h [deg.]
DM	$40 \times 40 \times 40$	$1.528e-01$	$1.6e-05$	$1.1e-09$	$4.0e-02$
SP	$40 \times 40 \times 40$	$1.526e-01$	$1.8e-03$	$5.0e-05$	$7.2e+00$
TG (CP, $r=6$)	$40 \times 40 \times 40$	$1.529e-01$	$1.8e-03$	$7.9e-06$	$2.6e+00$
FES	$7.2e+04$	$1.526e-01$	$2.5e-03$	$6.1e-05$	$6.8e+00$

Fig. 7. Magnetization configurations in a $1 \times 1 \times 0.1$ cuboid used for numerical experiments. The magnetization is normalized, its direction is color coded. (a) Homogeneous magnetization. (b) Vortex state. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this article.)
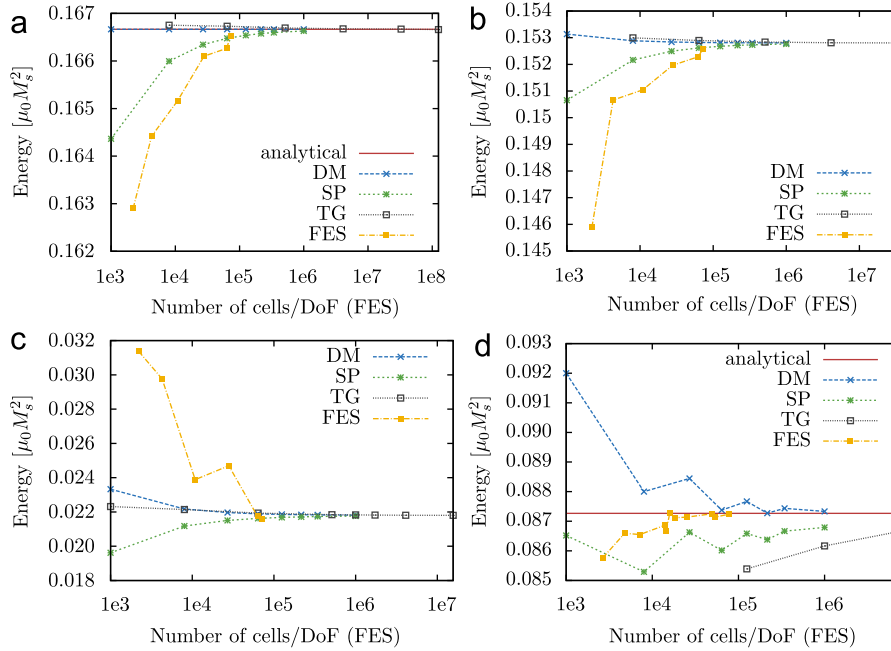

Fig. 8. Convergence of the calculated stray-field energy for different geometries and magnetization configurations. Like in Tables 1 and 2, N is the number of cells for the tensor-grid methods (DM, SP and TG) and the number of degrees of freedom in the case of finite elements (FES). (a) Homogeneously magnetized $1 \times 1 \times 1$ cube, (b) flower state in $1 \times 1 \times 1$ cube, (c) vortex state in $1 \times 1 \times 1$ cube, (d) homogeneously magnetized sphere with radius 0.5.

where the center of the cube is located at (0,0,0). We choose $a=c=1$ and $b=2$ (Fig. 7).

The results are similar to those of the homogeneously magnetized sample. In contrast the results of the DM method are not exact in this case, but Fig. 8(b) shows that the DM method converges faster than all other methods. Fig. 9 shows a log-log plot of the average cell size h against the error of the energy calculation and a table with the associated convergence rates. As reference for the error calculation the energy was computed on a $160 \times 160 \times 160$ grid with the DM method. All methods show approximately a quadratic convergence in the cell size.

The next comparison is for a vortex state in a unit cube, see Fig. 6(c), described by the model in [32], i.e.

$$m_x(r) = -\frac{y}{r}\left(1-\exp\left(-4\frac{r^2}{r_c^2}\right)\right)^{1/2},$$

$$m_y(r) = \frac{x}{r}\left(1-\exp\left(-4\frac{r^2}{r_c^2}\right)\right)^{1/2},$$

$$m_z(r) = \exp\left(-2\frac{r^2}{r_c^2}\right), \tag{31}$$

where $r = \sqrt{x^2+y^2}$, and we choose the radius of the vortex core as $r_c=0.14$. The vortex center coincides with the center of the cube, and the magnetization is assumed to be rotationally symmetric around the x/y-axis and translationally invariant along the z-axis. The results can be found in Table 5.

The most notable difference to the previous tests is the large field error in the FES method. It shows that the error occurs in the center of the vortex, where the gradient of the magnetization peaks. A possible solution for this problem would be an adaptive meshing, which is currently not implemented in our FES code.
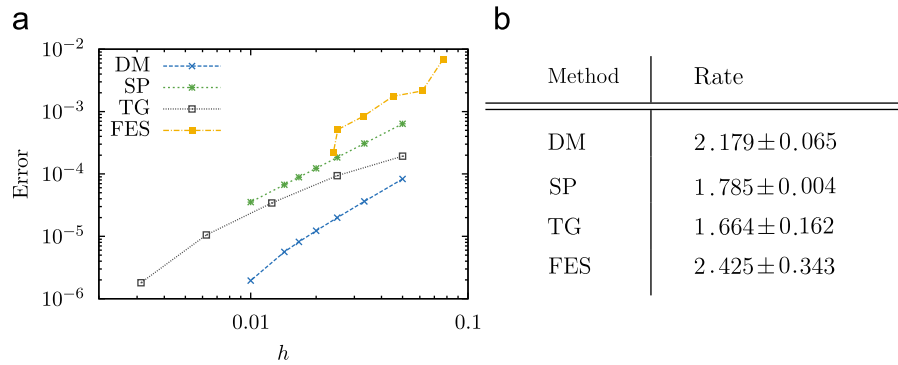
Method	Rate
DM	2.179 ± 0.065
SP	1.785 ± 0.004
TG	1.664 ± 0.162
FES	2.425 ± 0.343

Fig. 9. Convergence rates of the energy calculation of a flower state in a $1 \times 1 \times 1$ cube, where $h = 1/N^{1/3}$ and the error is computed against an asymptotic solution of the DM method.

Table 5
Errors for the vortex state in a unit cube: Relative error of the energy, relative error of the field/field-angle (w.r.t. $160 \times 160 \times 160$ calculations with the DM method).

Method	N	e	relerr h	av. relerr h [deg.]	max. err h [deg.]
DM	$40 \times 40 \times 40$	$2.189e-02$	$9.2e-04$	$2.9e-04$	$1.1e+01$
SP	$40 \times 40 \times 40$	$2.163e-02$	$2.3e-03$	$2.1e-05$	$3.4e+00$
TG (Tucker, $r=10$)	$40 \times 40 \times 40$	$2.193e-02$	$3.9e-03$	$1.5e-05$	$4.2e+00$
FES	$7.2e+04$	$2.160e-02$	$2.1e-02$	$6.1e-02$	$1.8e+02$

Table 6
Errors for the homogeneously magnetized $1 \times 1 \times 0.1$ thin film: relative error of the energy, relative error of the field/field-angle (w.r.t. $320 \times 320 \times 32$ calculations with the DM method).

Method	N	e	relerr h	av. relerr h [deg.]	max. err h [deg.]
DM	$80 \times 80 \times 8$	$4.025e-02$	0	0	0
SP	$80 \times 80 \times 8$	$4.021e-02$	$1.7e-03$	$2.6e-05$	$4.5e+00$
TG (CP, $r=1$)	$80 \times 80 \times 8$	$4.025e-02$	$3.7e-03$	$6.4e-06$	$2.1e+00$
FES	$4.9e+04$	$3.983e-02$	$5.5e-03$	$1.9e-05$	$5.0e+00$

Table 7
Errors for the vortex state in a $1 \times 1 \times 0.1$ thin film: Relative error of the energy, relative error of the field/field-angle (w.r.t. $320 \times 320 \times 32$ calculations with the DM method).

Method	N	e	relerr h	av. relerr h [deg.]	max. err h [deg.]
DM	$80 \times 80 \times 8$	$1.569e-03$	$2.3e-04$	$2.8e-06$	$6.8e-01$
SP	$80 \times 80 \times 8$	$1.555e-03$	$2.5e-03$	$4.6e-05$	$3.6e+00$
TG (Tucker $r=8$)	$80 \times 80 \times 8$	$1.569e-03$	$3.0e-03$	$1.8e-05$	$4.0e+00$
FES	$4.9e+04$	$1.496e-03$	$6.1e-03$	$6.6e-04$	$2.1e+01$

5.3. Thin film

We first take a homogeneously magnetized $1 \times 1 \times 0.1$ thin film (magnetization out of plane), see Table 6 for the results. Table 7 shows the results for the vortex state (out of plane) in the same thin film geometry.

The results for methods that do not rely on spatial discretization outside the sample perform equally well on this geometry. FES, instead, shows a deterioration of performance due to the worse ratio of shell and sample elements while leaving the number of DoF unchanged.

5.4. Sphere

As the last test a homogeneously magnetized sphere with radius $R=0.5$ is simulated. The spatial discretization in case of cuboid grids is done by setting the magnetization $\boldsymbol{M} = (0,0,M_z)$ in cells whose

center lies within the sphere. This leads to staircase artifacts as shown in Fig. 10(a). For the FES method the sphere is discretized such that the volume of the discretized sphere matches the analytical volume.

The magnetostatic energy for different spatial discretizations is displayed in Fig. 8(d). The FES method shows the fastest convergence, which is obviously a consequence of the better approximation of the curved surface, see Fig. 10(b). Also the field computation benefits from this better approximation, see Fig. 10(c) and (d).

Work was done on the treatment of curved surfaces within Cartesian grid methods [33]. Still the use of irregular grids is a more natural way of describing curved surfaces and is thus preferable.

6. Conclusion

We investigated several test magnetization configurations with different methods for the stray-field computation and
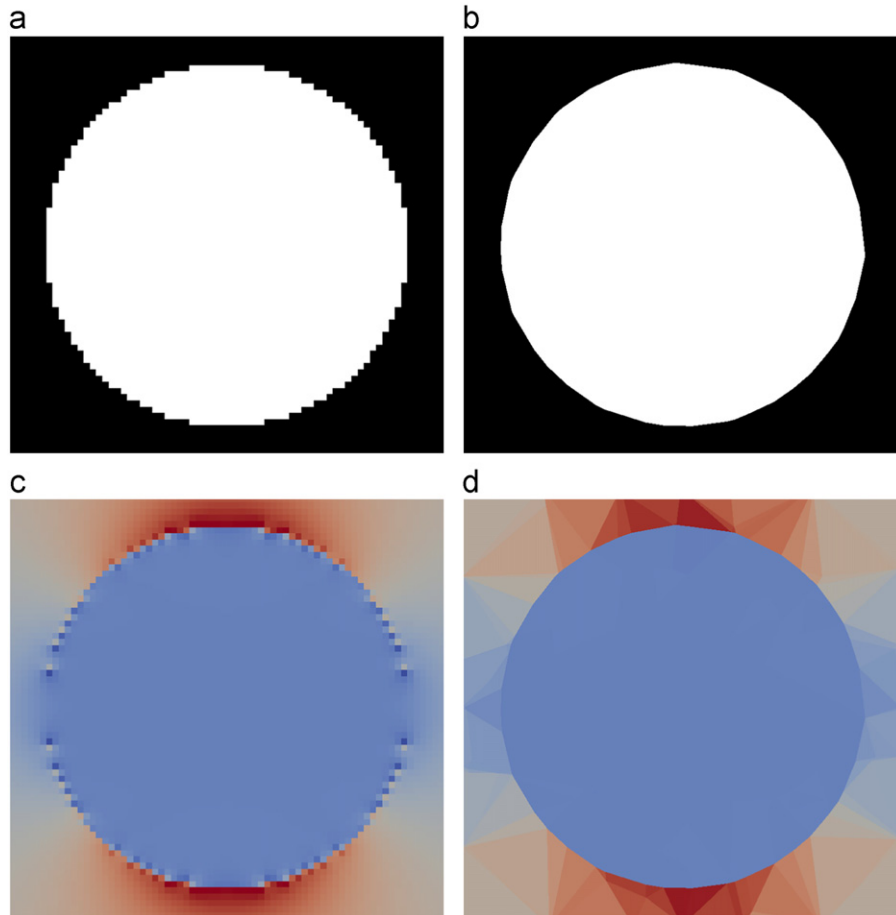
Fig. 10. Spatial discretization and stray field of a homogeneously, in z-direction magnetized sphere in the middle xz-plane. (a) Finite difference approximation of the spherical sample with $50 \times 50 \times 50$ cells. (b) Finite element approximation of the spherical sample with 9429 tetrahedra (including the shell elements). (c) z component of the stray field, calculated with the DM method. (d) z component of the stray field, calculated with the FES method.

compared the results. There is no clear winner in this comparison of numerical methods for the stray-field calculation. Computations on cuboid structures are best done with methods that compute on cuboid grids, namely the DM, SP and TG methods. The TG method is not only the fastest choice for a given spatial discretization, it is also able to handle very large grids due to low-rank tensor approximation or representation of the magnetization. However the TG method is not yet well investigated in the context of full micromagnetic simulations. In order to preserve the sublinear complexity and storage requirement features further research on the behaviour of low-rank magnetization during energy minimization or LLG integration has to be done.

The SP method is faster than the DM method by a factor of 1.5 and needs about 30% less memory. This speedup comes at the expense of accuracy. Among the Cartesian grid methods, the DM method is most accurate for a given spatial discretization since the stray field is computed directly. Both the SP and the TG method show an additional error due to the finite-difference gradient computation.

For curved structures FES is a good choice. The obvious reason for this is the use of irregular meshes, which are able to model the curvature much better than cuboid grids. The alternative FEM/BEM approach by Fredkin/Koehler [2] makes the ansatz $\phi = \phi_1 + \phi_2$ for the scalar potential. First a Neumann-problem is solved for ϕ_1 (sparse system), followed by the calculation of the boundary values for ϕ_2 due to ϕ_1 (dense matrix multiplication). Then ϕ_2 is obtained inside the volume by solving a Laplace equation with Dirichlet boundary conditions from the previous

step (sparse system). The computational complexity is therefore $\mathcal{O}(N+M^2)$, where N denotes the total number of degrees of freedom (DoF) and M the number of boundary nodes. When compression techniques like $\mathcal{H}$- or $\mathcal{H}^2$-matrices are used for the boundary element matrix, costs reduce to $\mathcal{O}(N+M \log M)$ or even $\mathcal{O}(N+M)$ respectively, but at an expense of accuracy because the boundary conditions in the last step underlie approximation errors.

FES solves one sparse preconditioned system, but needs the discretization of some outdoor space. The transformation, in general, leads to badly conditioned systems, which we addressed with algebraic multigrid preconditioners, leading to an overall complexity of $\mathcal{O}(N \log^\alpha N), \alpha \ll 1$, and N the total DoF including those of the outdoor space.

For the thin-film case, both methods scale worse, costs for FEM/BEM are dominated by the boundary element method, for FES the DoF in the outdoor space become significant.

Acknowledgments

Appendix A. Low-rank tensor formats

A tensor $\mathcal{A} \in \mathbb{R}^{N_1 \times N_2 \times N_3}$ is said to be in *canonical format (CANDECOMP/PARAFAC (CP) decomposition)* with *tensor product rank r*, if

$$\mathcal{A} = \sum_{l=1}^{r} \lambda_l \boldsymbol{u}_l^{(1)} \circ \boldsymbol{u}_l^{(2)} \circ \boldsymbol{u}_l^{(3)} \tag{A.1}$$

with $\lambda_l \in \mathbb{R}$, vectors $\boldsymbol{u}_l^{(j)} \in \mathbb{R}^{N_j}$, and $\circ$ is the vector outer product. A particular entry of a canonical tensor is given by

$$a_{ijk} = \sum_{l=1}^{r} \lambda_l (u_l^{(1)})_i (u_l^{(2)})_j (u_l^{(3)})_k. \tag{A.2}$$

Abbreviating notation as in [22], a tensor $\mathcal{A} \in \mathbb{R}^{N_1 \times N_2 \times N_3}$ in CP format can be written as

$$\mathcal{A} = [\![\, \boldsymbol{\lambda}; \boldsymbol{U}^{(1)}, \boldsymbol{U}^{(2)}, \boldsymbol{U}^{(3)} \,]\!], \tag{A.3}$$

with weight vector $\boldsymbol{\lambda} = [\lambda_1, \dots, \lambda_r] \in \mathbb{R}^r$ and factor matrices $\boldsymbol{U}^{(j)} = [\boldsymbol{u}_1^{(j)} | \dots | \boldsymbol{u}_r^{(j)}] \in \mathbb{R}^{N_j \times r}$.

From (A.3) it can be seen that the number of degrees of freedom (DoF) of a CP tensor is $r + r\sum_j N_j$ (compare with $\prod_j N_j$ for a dense tensor), also see Table 1.

A tensor $\mathcal{A} \in \mathbb{R}^{N_1 \times N_2 \times N_3}$ is said to be in *Tucker format* (Tucker tensor) if it can be represented in the form

$$\mathcal{A} = \mathcal{C} \times_1 \boldsymbol{U}_1 \times_2 \boldsymbol{U}_2 \times_3 \boldsymbol{U}_3, \tag{A.4}$$

with the so-called *core tensor* $\mathcal{C} \in \mathbb{R}^{r_1 \times r_2 \times r_3}$ and *factor matrices* $\boldsymbol{U}_j \in \mathbb{R}^{N_j \times r_j}$.

The key-operation is the n-mode (matrix) multiplication of a tensor $\mathcal{A} \in \mathbb{R}^{N_1 \times N_2 \times N_3}$ with a matrix $\boldsymbol{U} \in \mathbb{R}^{M \times N_n}$, which is the multiplication of each mode-n fiber of $\mathcal{A}$ by the matrix $\boldsymbol{U}$, i.e.

$$\mathcal{A} \times_n \boldsymbol{U} \in \mathbb{R}^{\times_{j=1}^{3} M_j}, \quad M_j = \begin{cases} N_j, & j \neq n \\ M, & j = n. \end{cases} \tag{A.5}$$

In contrast to CP tensors, the ranks in the Tucker representation can be different in each mode (dimension). In the discussions of Section 4 and the experiments in Section 5 we used the same rank r for each mode, i.e. $r \equiv r_j$.

For a tensor in Tucker format $\prod_j r_j + \sum_j r_j N_j$ entries have to be stored, which is a compression for $r_j \ll N_j$, also see Table 1. For a sum of Tucker tensors one can only store the factor matrices and core tensors of the summands, which is called *block-CP format*.

Linear algebra operations for low-rank tensors, like the inner product, tensor–matrix product etc., can be performed without forming the dense tensors [34], which makes these operations faster than their conventional counterparts.

References

[1] I. Tsukerman, A. Plaks, H.N. Bertram, Multigrid methods for computation of magnetostatic fields in magnetic recording problems, Journal of Applied Physics 83 (1998) 6344–6346.

[2] D. Fredkin, T. Koehler, Hybrid method for computing demagnetizing fields, IEEE Transactions on Magnetics 26 (1990) 415–417.

[3] X. Brunotte, G. Meunier, J. Imhoff, Finite element modeling of unbounded problems using transformations: a rigorous, powerful and easy solution, IEEE Transactions on Magnetics 28 (1992) 1663–1666.

[4] A. Buchau, W. Rucker, O. Rain, V. Rischmuller, S. Kurz, S. Rjasanow, Comparison between different approaches for fast and efficient 3-d bem computations, IEEE Transactions on Magnetics 39 (2003) 1107–1110.

[5] N. Popović, D. Praetorius, Applications of $\mathcal{H}$-matrix techniques in micromagnetics, Computing 74 (2005) 177–204, http://dx.doi.org/10.1007/s00607-004-0098-7.

[6] A. Knittel, M. Franchin, G. Bordignon, T. Fischbacher, S. Bending, H. Fangohr, Compression of boundary element matrix in micromagnetic simulations, Journal of Applied Physics 105 (2009) 07D542.

[7] R. Beatson, L. Greengard, A short course on fast multipole methods, Wavelets, Multilevel Methods and Elliptic PDEs 1 (1997) 1–37.

[8] J. Blue, M. Scheinfein, Using multipoles decreases computation time for magnetostatic self-energy, IEEE Transactions on Magnetics 27 (1991) 4778–4780.

[9] B. Livshitz, A. Boag, H.N. Bertram, V. Lomakin, Nonuniform grid algorithm for fast calculation of magnetostatic interactions in micromagnetics, Journal of Applied Physics 105 (2009) 07D541.

[10] C. Abert, G. Selke, B. Krüger, A. Drews, A fast finite-difference method for micromagnetics using the magnetic scalar potential, IEEE Transactions on Magnetics 48 (2012) 1105–1109.

[11] H. Long, E. Ong, Z. Liu, E. Li, Fast fourier transform on multipoles for rapid calculation of magnetostatic fields, IEEE Transactions on Magnetics 42 (2006) 295–300.

[12] MicroMagnum, 〈http://micromagnum-tis.informatik.uni-hamburg.de/〉, 2012.

[14] L. Exl, W. Auzinger, S. Bance, M. Gusenbauer, F. Reichel, T. Schrefl, Fast stray field computation on tensor grids, Journal of Computational Physics 231 (2012) 2840–2850.

[15] R. Courant, Calculus of Variations, New York, 1946.

[16] W.F. Brown Jr., Micromagnetics, Interscience Publisher, New York, 1963.

[17] J.D. Jackson, Classical Electrodynamics, 3rd ed., American Journal of Physics 67 (1999) 841.

[18] J. Fidler, T. Schrefl, Micromagnetic modelling - the current state of the art, Journal of Physics D: Applied Physics 33 (2000) R135.

[19] M. Donahue, D. Porter, OOMMF User's Guide, Version 1.0, Interagency Report NISTIR 6376, 1999.

[20] A. Vansteenkiste, B. Van De Wiele, MuMax: a new high-performance micromagnetic simulation tool, Journal of Magnetism and Magnetic Materials 323 (2011) 2585–2591.

[21] A. Goncharov, G. Hrkac, J. Dean, T. Schrefl, Kronecker product approximation of demagnetizing tensors for micromagnetics, Journal of Computational Physics 229 (2010) 2544–2549.

[22] T.G. Kolda, B.W. Bader, Tensor decompositions and applications, SIAM Review 51 (2009) 455–500.

[24] J. Jusélius, D. Sundholm, Parallel implementation of a direct method for calculating electrostatic potentials, The Journal of Chemical Physics 126 (2007) 094101.

[25] W. Hackbusch, B.N. Khoromskij, Low-rank kronecker-product approximation to multi-dimensional nonlocal operators. Part I. Separable approximation of multi-variate functions, Computing 76 (2006) 177–202.

[26] R. McMichael, Standard problem number 3, problem specification and reported solutions, Micromagnetic Modeling Activity Group, 1998.

[27] D.P. Bertsekas, Multiplier methods: a survey, Automatica 12 (1976) 133–145.

[28] A. Logg, K.-A. Mardal, G.N. Wells, et al., Automated Solution of Differential Equations by the Finite Element Method, Springer, 2012.

[29] W.H. Press, S.A. Teukolsky, W.T. Vetterling, B.P. Flannery, Numerical Recipes: the Art of Scientific Computing, Cambridge University Press, Cambridge, 2007.

[30] M. Donahue, Parallelizing a micromagnetic program for use on multiprocessor shared memory computers, IEEE Transactions on Magnetics 45 (2009) 3923–3925.

[31] R.P. Cowburn, M.E. Welland, Micromagnetics of the single-domain state of square ferromagnetic nanostructures, Physical Review B 58 (1998) 9217–9226.

[32] E. Feldtkeller, H. Thomas, Struktur und Energie von Blochlinien in dünnen ferromagnetischen Schichten, Zeitschrift für Physik B Condensed Matter 4 (1965) 8–14, http://dx.doi.org/10.1007/BF02423256.

[33] M.J. Donahue, R.D. McMichael, Micromagnetics on curved geometries using rectangular cells: error correction and analysis, IEEE Transactions on Magnetics 43 (2007) 2878–2880.

[34] B.W. Bader, T.G. Kolda, Efficient MATLAB computations with sparse and factored tensors, SIAM Journal on Scientific Computing 30 (2008) 205.