

The design and verification of MuMax3

Arne Vansteenkiste, Jonathan Leliaert, Mykola Dvornik, Mathias Helsen, Felipe Garcia-Sanchez, and Bartel Van Waeyenberge

Citation: [AIP Advances](#) **4**, 107133 (2014); doi: 10.1063/1.4899186

View online: <http://dx.doi.org/10.1063/1.4899186>

View Table of Contents: <http://scitation.aip.org/content/aip/journal/adva/4/10?ver=pdfcov>

Published by the [AIP Publishing](#)

Articles you may be interested in

[Width dependent transition of quantized spin-wave modes in Ni80Fe20 square nanorings](#)
J. Appl. Phys. **116**, 163912 (2014); 10.1063/1.4900616

[Magnetic vortex observation in FeCo nanowires by quantitative magnetic force microscopy](#)
Appl. Phys. Lett. **105**, 172409 (2014); 10.1063/1.4900998

[Helical domain walls in constricted cylindrical NiFe nanowires](#)
Appl. Phys. Lett. **101**, 152406 (2012); 10.1063/1.4758469

[Nanoscale magnetic structure and properties of solution-derived self-assembled La_{0.7}Sr_{0.3}MnO₃ islands](#)
J. Appl. Phys. **111**, 024307 (2012); 10.1063/1.3677985

[Study of magnetization state transition in closely spaced nanomagnet two-dimensional array for computation](#)
J. Appl. Phys. **109**, 07E513 (2011); 10.1063/1.3536795



Goodfellow
metals • ceramics • polymers
composites • compounds • glasses

Save 5% • Buy online
70,000 products • Fast shipping

The design and verification of MuMax3

Arne Vansteenkiste,^{1,a} Jonathan Leliaert,¹ Mykola Dvornik,¹
Mathias Helsen,¹ Felipe Garcia-Sanchez,^{2,3} and Bartel Van Waeyenberge¹

¹DyNaMat LAB, Department of Solid State Sciences, Ghent University, Belgium

²Institut d'Electronique Fondamentale Univ. Paris-Sud, 91405 Orsay, France

³UMR 8622, CNRS, 91405 Orsay, France

(Received 6 August 2014; accepted 10 October 2014; published online 20 October 2014)

We report on the design, verification and performance of MuMax3, an open-source GPU-accelerated micromagnetic simulation program. This software solves the time- and space dependent magnetization evolution in nano- to micro scale magnets using a finite-difference discretization. Its high performance and low memory requirements allow for large-scale simulations to be performed in limited time and on inexpensive hardware. We verified each part of the software by comparing results to analytical values where available and to micromagnetic standard problems. MuMax3 also offers specific extensions like MFM image generation, moving simulation window, edge charge removal and material grains. © 2014 Author(s). All article content, except where otherwise noted, is licensed under a Creative Commons Attribution 3.0 Unported License. [<http://dx.doi.org/10.1063/1.4899186>]

I. INTRODUCTION

MuMax3 is a GPU-accelerated micromagnetic simulation program. It calculates the space- and time-dependent magnetization dynamics in nano- to micro-sized ferromagnets using a finite-difference discretization. A similar technique is used by the open-source programs OOMMF¹ (CPU) and Micro-Magnum² (GPU), and the commercial GpMagnet³ (GPU). MuMax3 was re-designed independently of its predecessors MuMax1⁴ and MuMax2.⁵

MuMax3 is open-source software written in Go⁶ and CUDA⁷, and is freely available under the GPLv3 license on <http://mumax.github.io>. In addition to the terms of the GPL, we kindly request that any work using MuMax3 refers to the latter website and this paper. An nVIDIA GPU and a Linux, Windows or Mac platform is required to run the software. Apart from nVIDIA's GPU driver, no other dependencies are required to run MuMax3.

Finite-element micromagnetic software exists as well like, e.g., NMag,⁸ TetraMag,⁹ MagPar¹⁰ and FastMag.¹¹ They offer more geometrical flexibility than finite-difference methods, at the expense of performance.

In this paper we first describe each of MuMax3's components and asses their individual correctness and accuracy. Then we address the micromagnetic standard problems,¹² where all software components have to work correctly together to solve real-world simulations. We typically compare against OOMMF¹ which has been widely used and extensively tested for over more than a decade. Finally, we report on the performance in terms of speed and memory consumption.

The complete input files used to generate the graphs in this paper are available on <http://mumax.github.io/pub>, allowing for each of the presented results to be reproduced independently. The scripts were executed with MuMax version 3.6.

^aElectronic address: Arne.Vansteenkiste@UGent.be

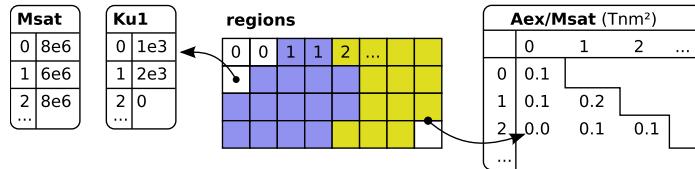


FIG. 1. Each simulation cell is attributed a region index representing the cell's material type. Material parameters like the saturation magnetization M_{sat} , anisotropy constants, etc are stored in 1D look-up tables indexed by the region index. Coupling parameters like the exchange strength are stored in a 2D lower triangular matrix indexed by a face's two neighbor region indices.

II. DESIGN

A. Material Regions

MuMAX3 employs a finite difference (FD) discretization of space using a 2D or 3D grid of orthorhombic cells. Volumetric quantities, like the magnetization and effective field, are treated at the center of each cell. On the other hand, coupling quantities like the exchange strength, are considered on the faces between the cells (Fig. 1).

In order to conserve memory, space-dependent material parameters are not explicitly stored per-cell. Instead, each cell is attributed a *region index* between 0 and 256. Different region indices represent different materials. The actual material parameters are stored in 256-element look-up tables, indexed by the cell's region index.

Coupling parameters like the exchange strength are stored in a triangular matrix, indexed by the region numbers of the two interacting cells. This allows arbitrary exchange coupling between all pairs of materials (Section III C).

Time-dependent parameters In addition to region-wise space-dependence, material parameters in each region can be time-dependent, given by one arbitrary function of time per region.

Excitations like the externally applied field or electrical current density can be set region- and time-wise in the same way as material parameters. Additionally they can have an arbitrary number of extra terms of the form $f(t) \times g(x, y, z)$, where $f(t)$ is any function of time multiplied by a continuously varying spatial profile $g(x, y, z)$. This allows to model smooth time- and space dependent excitations like, e.g., an antenna's RF field or an AC electrical current.

B. Geometry

MuMAX3 uses *Constructive Solid Geometry* to define the shape of the magnet and the material regions inside it. Any shape is represented by a function $f(x, y, z)$ that returns true when (x, y, z) lies inside the shape or false otherwise. For example, a sphere is represented by the function $x^2 + y^2 + z^2 \leq r^2$. Shapes can be rotated, translated, scaled and combined together with boolean operations like AND, OR, XOR. This allows for complex, parametrized geometries to be defined programmatically. The example in Fig. 2 shows the magnetization in the logical OR of an ellipsoid and cuboid.

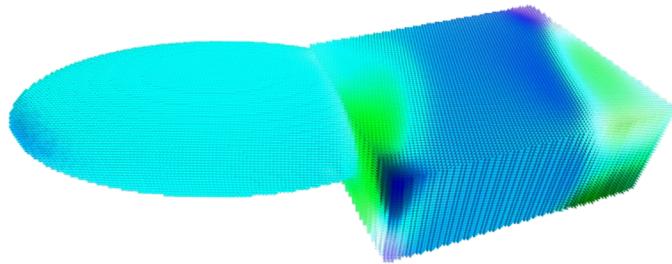


FIG. 2. Geometry obtained by logically combining an ellipsoid and rotated cuboid. The arrows depict the magnetization direction in this complex shape.

C. Interface

Input scripts MuMAX3 provides a dedicated scripting language that resembles a subset of the Go programming language. The script provides a simple means to define fairly complex simulations. This is illustrated by the code snippet below where we excite a Permalloy ellipse with a 1 GHz RF field:

```
setgridsize(128, 32, 1)
setcellsize(5e-9, 5e-9, 8e-9)
setGeom(ellipse(500e-9, 160e-9))

Msat = 860e3
Aex = 13e-12
alpha= 0.05

m=uniform(1, 0, 0)
relax()

f := 1e9 // 1GHz
A := 0.01 // 10mT
B_ext = vector(0.1, A*sin(2*pi*f*t), 0)
run(10e-9)
```

Programming The MuMAX3 libraries can also be called from native Go. In this way, the full Go language and libraries can be leveraged for more powerful input generation and output processing than the built-in scripting.

Web interface MuMAX3 provides web-based HTML 5 user interface. It allows to inspect and control simulations from within a web browser, whether they are running locally or remotely. Simulations may also be entirely constructed and run from within the web GUI. In any case an input file corresponding to the user's clicks is generated, which may later be used to repeat the simulation in an automated fashion.

Dataformat MuMAX3 uses OOMMF's "OVF" data format for input and output of all space-dependent quantities. This allows to leverage existing tools. Additionally a tool is provided to convert the output to several other data formats like paraview's VTK,¹³ gnuplot,¹⁴ comma-separated values (CSV), Python-compatible JSON, . . . , and to image formats like PNG, JPG and GIF. Finally, the output is compatible with the 3D rendering software MuVIEW, contributed by Graham Rowlands.¹⁵

III. DYNAMICAL TERMS

MuMAX3 calculates the evolution of the reduced magnetization $\vec{m}(\vec{r}, t)$, which has unit length. In what follows the dependence on time and space will not be explicitly written down. We refer to the time derivative of \vec{m} as the torque $\vec{\tau}$ (units 1/s):

$$\frac{\partial \vec{m}}{\partial t} = \vec{\tau} \quad (1)$$

$\vec{\tau}$ has three contributions:

- Landau-Lifshitz torque $\vec{\tau}_{LL}$ (Section III A)
- Zhang-Li spin-transfer torque $\vec{\tau}_{ZL}$ (Section III G)
- Slonczewski spin-transfer torque $\vec{\tau}_{SL}$ (Section III H).

A. Landau-Lifshitz torque

MuMAX3 uses the following explicit form for the Landau-Lifshitz torque:^{16,17}

$$\vec{\tau}_{LL} = \gamma_{LL} \frac{1}{1 + \alpha^2} (\vec{m} \times \vec{B}_{eff} + \alpha (\vec{m} \times (\vec{m} \times \vec{B}_{eff}))) \quad (2)$$

with γ_{LL} the gyromagnetic ratio (rad/Ts), α the dimensionless damping parameter and \vec{B}_{eff} the effective field (T). The default value for γ_{LL} can be overridden by the user. \vec{B}_{eff} has the following contributions:

- externally applied field \vec{B}_{ext}
- magnetostatic field \vec{B}_{demag} ([III B](#))
- Heisenberg exchange field \vec{B}_{exch} ([III C](#))
- Dzyaloshinskii-Moriya exchange field \vec{B}_{dm} ([III D](#))
- magneto-crystalline anisotropy field \vec{B}_{anis} ([III E](#))
- thermal field \vec{B}_{therm} ([III F](#)).

[Fig. 3](#) shows a validation of the Landau-Lifshitz torque for a single spin precessing without damping in a constant external field.

B. Magnetostatic field

Magnetostatic convolution A finite difference discretization allows the magnetostatic field to be evaluated as a (discrete) convolution of the magnetization with a demagnetizing kernel $\hat{\mathbf{K}}$:

$$\vec{B}_{demag\ i} = \hat{\mathbf{K}}_{ij} * \vec{M}_j \quad (3)$$

where $\vec{M} = M_{sat}\vec{m}$ is the unnormalized magnetization, with M_{sat} the saturation magnetization (A/m). This calculation is FFT-accelerated based on the well-known convolution theorem. The corresponding energy density is provided as:

$$\mathcal{E}_{demag} = -\frac{1}{2} \vec{M} \cdot \vec{B}_{demag} \quad (4)$$

Magnetostatic kernel We construct the demagnetizing kernel $\hat{\mathbf{K}}$ assuming constant magnetization¹⁸ in each finite difference cell and we average the resulting \vec{B}_{demag} over the cell volumes. The integration is done numerically with the number of integration points automatically chosen based on the distance between source and destination cells and their aspect ratios. The kernel is initialized on CPU in double precision, and only truncated to single before transferring to GPU.

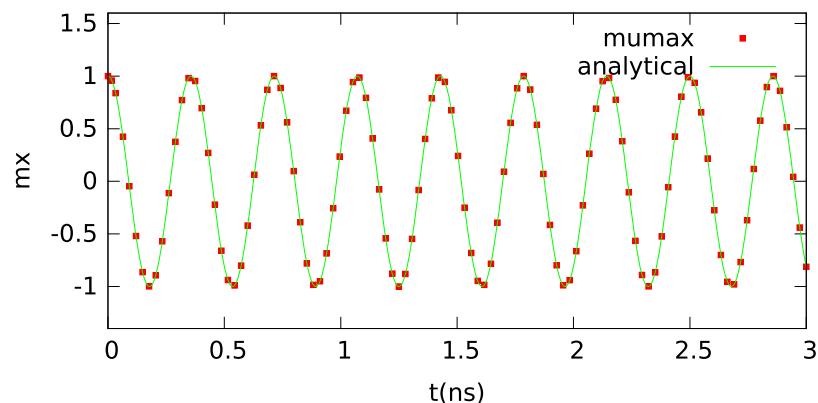


FIG. 3. Validation of Eq. (2) for a single spin precessing without damping in a 0.1 T field along z , perpendicular to \vec{m} . Analytical solution: $m_x = \cos(0.1T_{LL}t)$.

TABLE I. Demagnetizing factors ($N_{ij} = \mathbf{H}_i/M_j$) calculated for a cube discretized in the smallest possible number of tetragonal cells with given aspect ratio $\text{size}_z/\text{size}_x$ ($=\text{size}_z/\text{size}_y$). The results lie close to the analytical value of $1/3$, even for very elongated (aspect >1) or flat (aspect <1) cells. The off-diagonal elements (not shown) are consistent with zero within the single-precision limit.

aspect	N_{xx}	N_{yy}	N_{zz}
8/1	-0.333207	-0.333207	-0.333176
4/1	-0.333149	-0.333149	-0.333144
2/1	-0.333118	-0.333118	-0.333118
1/1	-0.333372	-0.333372	-0.333372
1/4	-0.333146	-0.333146	-0.333145
1/16	-0.333176	-0.333176	-0.333280
1/64	-0.333052	-0.333052	-0.333639

The kernel's mirror symmetries and zero elements are exploited to reduce storage and initialization time. This results in a $9\times$ or $12\times$ decrease in kernel memory usage for 2D and 3D simulations respectively, and is part of the reason for MuMax3's relatively low memory requirements (Section VII).

Accuracy The short-range accuracy of $\hat{\mathbf{K}}$ is tested by calculating the demagnetizing factors of a uniformly magnetized cube, analytically known to be $-1/3$ in each direction. The cube was discretized in cells with varying aspect ratios to stress the numerical integration scheme. The smallest possible number of cells was used to ensure that the short-range part of the field has an important contribution. The results presented in Table I are accurate to 3 or 4 digits. Standard Problem #2 (V B) is another test sensitive to the short-range kernel accuracy.¹⁹

The long-range accuracy of the magnetostatic convolution is assessed by comparing kernel and the field of a single magnetized cell to the corresponding point dipole. The fields presented in Fig. 4, show perfect long-range accuracy for the kernel, indicating accurate numerical integration in that range. The resulting field, obtained by convolution of a single magnetized cell ($B_{\text{sat}}=1$ T) with the kernel, is accurate down to about $0.01\ \mu\text{T}$ —the single-precision noise floor introduced by the FFT's.

Periodic boundary conditions MuMax3 provides optional periodic boundary conditions (PBCs) in each direction. PBCs imply magnetization wrap-around in the periodic directions, felt by stencil

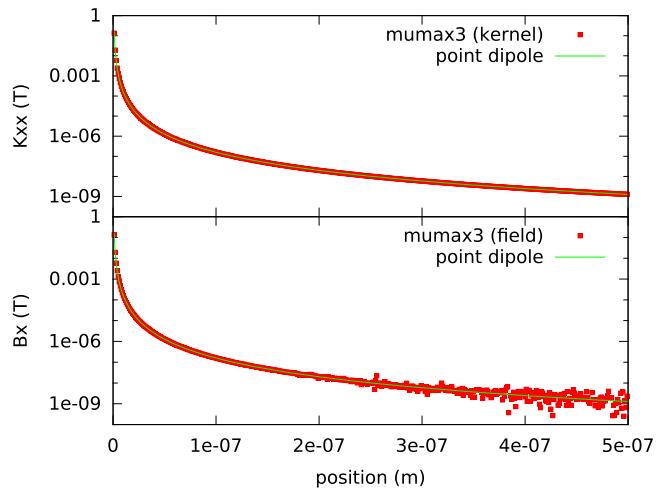


FIG. 4. Kernel element $\hat{\mathbf{K}}_{xx}$ (top) and \vec{B}_x , the field of a single magnetized cell ($1\ \text{nm}^3$, $B_{\text{sat}} = 1$ T) (bottom) along the x axis (1 nm cells), compared to the field of a corresponding dipole. The long-range field remains accurate down to the single-precision numerical limit ($\sim 10^{-7}$ T).

TABLE II. Out-of-plane demagnetizing factors for a thin film with grid size $2 \times 2 \times 1$ and 2D PBC's with $P_x (=P_y)$ copies in the x and y direction (column 1) or without PBC's but with a correspondingly enlarged grid size extended $\times 2P_x (=2P_y)$ in both x and y directions (column 2). Both give comparable results for sufficiently large P_x , P_y , verifying the PBC implementation.

$P_x (=P_y)$	N_{zz}	grid	N_{zz}
$\times 1$	-0.71257	$\times 2$	-0.76368
$\times 4$	-0.93879	$\times 8$	-0.94224
$\times 16$	-0.98438	$\times 32$	-0.98460
$\times 64$	-0.99514	$\times 128$	-0.99515
$\times 256$	-0.99713	$\times 512$	-0.99779
∞	-1	$\times \infty$	-1

TABLE III. Transverse demagnetizing factors for a long rod with grid size $2 \times 2 \times 1$ and 1D PBC's with P_z copies along z (column 1) or without PBC's but with a correspondingly enlarged grid size $\times 2P_z$ (column 2). Both give comparable results for sufficiently large P_z , verifying the PBC implementation.

P_z	N_{xx}	grid	N_{xx}
$\times 1$	-0.251960	$\times 2$	-0.3331182
$\times 4$	-0.476766	$\times 8$	-0.4809398
$\times 16$	-0.498280	$\times 32$	-0.4983577
$\times 64$	-0.499517	$\times 128$	-0.4995183
$\times 256$	-0.499590	$\times 512$	-0.4995911
∞	-0.5	$\times \infty$	-0.5

operations like the exchange interaction. A less trivial consequence is that the magnetostatic field of repeated magnetization images (copies) has to be added to \vec{B}_{demag} .

In contrast to OOMMF's PBC implementation,²⁰ MuMax3 employs a so-called macro geometry approach^{8,21} where a finite (though usually large) number of periodic images (copies) is taken into account, and that number can be freely chosen in each direction. MuMax3's `setPBC(Px, Py, Pz)` command enables P_x, P_y, P_z additional copies *on each side* of the simulation box, given that P_i is sufficiently large.

To test the magnetostatic field with PBC's, we calculate the demagnetizing tensors of a wide film and long rod in two different ways: either with a large grid without PBC's, or with a small grid but with PBC's equivalent to the larger grid. In our implementation, a gridsize (N_x, N_y, N_z) with PBC's (P_x, P_y, P_z) should approximately correspond to a gridsize ($2P_xN_x, 2P_yN_y, 2P_zN_z$) without PBC's. This is verified in tables II and III where we extend in plane for the film and along z for the rod. Additionally, for very large sizes both results converge to the well-known analytical values for infinite geometries.

C. Heisenberg exchange interaction

The effective field due to the Heisenberg exchange interaction:²²

$$\vec{B}_{\text{exch}} = 2 \frac{A_{\text{ex}}}{M_{\text{sat}}} \Delta \vec{m} \quad (5)$$

is evaluated using a 6-neighbor small-angle approximation:^{23,24}

$$\vec{B}_{\text{exch}} = 2 \frac{A_{\text{ex}}}{M_{\text{sat}}} \sum_i \frac{(\vec{m}_i - \vec{m})}{\Delta_i^2} \quad (6)$$

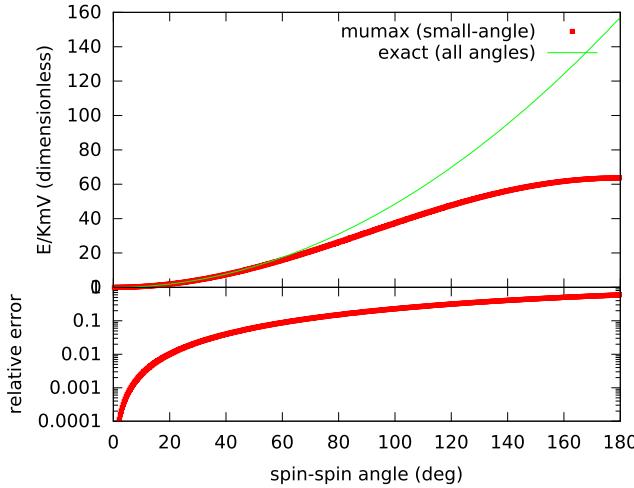


FIG. 5. Numerical (Eq. (6), (8)) and analytical (Eq. (7)) exchange energy density (in units $K_m = 1/2\mu_0 M_{\text{sat}}^2$) for spiral magnetizations as a function of the angle between neighboring spins (independent of material parameters). To ensure an accurate energy, spin-spin angles should be kept below $\sim 20\text{--}30^\circ$ by choosing a sufficiently small cell size.

where i ranges over the six nearest neighbors of the central cell with magnetization \vec{m} . Δ_i is the cell size in the direction of neighbor i .

At the boundary of the magnet some neighboring magnetizations \vec{m}_i are missing. In that case we use the cell's own value \vec{m} instead of \vec{m}_i , which is equivalent to employing Neumann boundary conditions.^{23,24}

The corresponding energy density is provided as:

$$\mathcal{E}_{\text{exch}} = A_{\text{ex}}(\nabla \vec{m})^2 \quad (7)$$

$$= -\frac{1}{2} \vec{M} \cdot \vec{B}_{\text{exch}} \quad (8)$$

MuMAX3 calculates the energy from the effective field using Eqns. (6), (8). The implementation is verified by calculating the exchange energy of a 1D magnetization spiral, for which the exact form (Eq. (7)) is easily evaluated. Fig. 5 shows that the linearized approximation is suited as long as the angle between neighboring magnetizations is not too large. This can be achieved by choosing a sufficiently small cell size compared to the exchange length.

Inter-region exchange The exchange interaction between different materials deserves special attention. A_{ex} and M_{sat} are defined in the cell volumes, while Eq. (6) requires a value of $A_{\text{ex}}/M_{\text{sat}}$ properly averaged out between the neighboring cells. For neighboring cells with different material parameters A_{ex1} , A_{ex2} and M_{sat1} , M_{sat2} MuMAX3 uses a harmonic mean:

$$\vec{B}_{\text{exch}} = 2S \frac{\frac{2}{M_{\text{sat1}}} \frac{A_{\text{ex2}}}{M_{\text{sat2}}}}{\frac{A_{\text{ex1}}}{M_{\text{sat1}}} + \frac{A_{\text{ex2}}}{M_{\text{sat2}}}} \sum_i \frac{(\vec{m}_i - \vec{m})}{\Delta_i^2} \quad (9)$$

which can easily be derived, and where we set $S = 1$ by default. S is an arbitrary scaling factor which may be used to alter the exchange coupling between regions, e.g., to lower the coupling between grains or antiferromagnetically couple two layers.

D. Dzyaloshinskii-Moriya interaction

MuMAX3 provides induced Dzyaloshinskii-Moriya interaction for thin films with out-of-plane symmetry breaking according to Ref. 25, yielding an effective field term:

$$\vec{B}_{\text{DM}} = \frac{2D}{M_{\text{sat}}} \left(\frac{\partial m_z}{\partial x}, \frac{\partial m_z}{\partial y}, -\frac{\partial m_x}{\partial x} - \frac{\partial m_y}{\partial y} \right) \quad (10)$$

where we apply boundary conditions:²⁶

$$\frac{\partial m_z}{\partial x} \Big|_{\partial V} = \frac{D}{2A} m_x \quad (11)$$

$$\frac{\partial m_z}{\partial y} \Big|_{\partial V} = \frac{D}{2A} m_y \quad (12)$$

$$\frac{\partial m_x}{\partial x} \Big|_{\partial V} = \frac{\partial m_y}{\partial y} \Big|_{\partial V} = -\frac{D}{2A} m_z \quad (13)$$

$$\frac{\partial m_x}{\partial y} \Big|_{\partial V} = \frac{\partial m_y}{\partial x} \Big|_{\partial V} = 0 \quad (14)$$

$$\frac{\partial m_x}{\partial z} \Big|_{\partial V} = \frac{\partial m_y}{\partial z} \Big|_{\partial V} = \frac{\partial m_z}{\partial z} \Big|_{\partial V} = 0 \quad (15)$$

Numerically, all derivatives are implemented as central derivatives, i.e., the difference between neighboring magnetizations over their distance in that direction: $\partial \vec{m}/\partial i = (\vec{m}_{i+1} - \vec{m}_{i-1})/(2\Delta_i)$. When a neighbor is missing at the boundary (∂V), its magnetization is replaced by $\vec{m} + \frac{\partial \vec{m}}{\partial i}|_{\partial V} \Delta_i \vec{n}$ where \vec{m} refers to the central cell, \vec{n} to the surface normal and the relevant partial derivative is selected from Eq. (11)–(15).

In case of nonzero D , these boundary conditions are simultaneously applied to the Heisenberg exchange field.

The effective field in Eq. (10) gives rises to an energy density:

$$\mathcal{E}_{\text{exch(DM)}} = m_z(\nabla \cdot \vec{m}) - (\vec{m} \cdot \nabla)m_z \quad (16)$$

$$= -\frac{1}{2}\vec{M} \cdot \vec{B}_{\text{exch(DM)}} \quad (17)$$

Similar to the anisotropic exchange case, MuMAX3 calculates the energy density from Eqns. (17), (10). Eq. (16) is the exact form, well approximated for sufficiently small cell sizes.

In Fig. 6, the DMI implementation is compared to the work of Thiaville *et al.*,²⁷ where the transformation of a Bloch wall into a Néel wall by varying D_{ex} is studied.

E. Magneto-crystalline anisotropy

Uniaxial MuMAX3 provides uniaxial magneto-crystalline anisotropy in the form of an effective field term:

$$\begin{aligned} \vec{B}_{\text{anis}} &= \frac{2K_{\text{ul}}}{B_{\text{sat}}}(\vec{u} \cdot \vec{m})\vec{u} \\ &+ \frac{4K_{\text{u2}}}{B_{\text{sat}}}(\vec{u} \cdot \vec{m})^3\vec{u} \end{aligned} \quad (18)$$

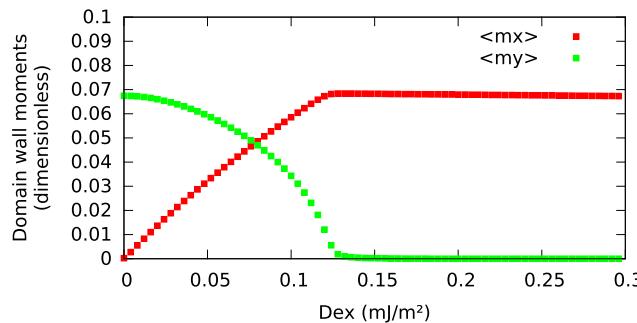


FIG. 6. Simulated domain wall magnetization in a 250 nm wide, 0.6 nm thick Co/Pt film ($M_{\text{sat}}=1100 \times 10^3$ A/m, $A_{\text{ex}}=16 \times 10^{-12}$ J/m, $K_{\text{ul}}=1.27 \times 10^6$ J/m³) as a function of the Dzyaloshinskii-Moriya strength D_{ex} . The left-hand and right-hand sides correspond to a Bloch and Néel wall, respectively. Results correspond well to Ref. 27.

where K_{u1} and K_{u2} are the first and second order uniaxial anisotropy constants and \vec{u} a unit vector indicating the anisotropy direction. This corresponds to an energy density:

$$\mathcal{E}_{\text{anis}} = -K_{u1}(\vec{u} \cdot \vec{m})^2 - K_{u2}(\vec{u} \cdot \vec{m})^4 \quad (19)$$

$$= -\frac{1}{2}\vec{\mathbf{B}}_{\text{anis}}(K_{u1}) \cdot \vec{\mathbf{M}} - \frac{1}{4}\vec{\mathbf{B}}_{\text{anis}}(K_{u2}) \cdot \vec{\mathbf{M}} \quad (20)$$

MuMAX3 calculates the energy density from the effective field using Eq. (20), where $\vec{\mathbf{B}}_{\text{anis}}(K_{ui})$ denotes the effective field term where only K_{ui} is taken into account. The resulting energy is verified in Fig. 7. Since the energy is derived directly from the effective field, this serves as a test for the field as well.

Cubic MuMAX3 provides cubic magneto-crystalline anisotropy in the form of an effective field term:

$$\begin{aligned} \vec{\mathbf{B}}_{\text{anis}} = & \\ -2K_{c1}/M_{\text{sat}}(& ((\vec{c}_2 \cdot \vec{m})^2 + (\vec{c}_3 \cdot \vec{m})^2)((\vec{c}_1 \cdot \vec{m})\vec{c}_1) + \\ & ((\vec{c}_1 \cdot \vec{m})^2 + (\vec{c}_3 \cdot \vec{m})^2)((\vec{c}_2 \cdot \vec{m})\vec{c}_2) + \\ & ((\vec{c}_1 \cdot \vec{m})^2 + (\vec{c}_2 \cdot \vec{m})^2)((\vec{c}_3 \cdot \vec{m})\vec{c}_3)) \\ -2K_{c2}/M_{\text{sat}}(& ((\vec{c}_2 \cdot \vec{m})^2(\vec{c}_3 \cdot \vec{m})^2)((\vec{c}_1 \cdot \vec{m})\vec{c}_1) + \\ & ((\vec{c}_1 \cdot \vec{m})^2(\vec{c}_3 \cdot \vec{m})^2)((\vec{c}_2 \cdot \vec{m})\vec{c}_2) + \\ & ((\vec{c}_1 \cdot \vec{m})^2(\vec{c}_2 \cdot \vec{m})^2)((\vec{c}_3 \cdot \vec{m})\vec{c}_3)) \\ -4K_{c3}/M_{\text{sat}}(& ((\vec{c}_2 \cdot \vec{m})^4 + (\vec{c}_3 \cdot \vec{m})^4)((\vec{c}_1 \cdot \vec{m})^3\vec{c}_1) + \\ & ((\vec{c}_1 \cdot \vec{m})^4 + (\vec{c}_3 \cdot \vec{m})^4)((\vec{c}_2 \cdot \vec{m})^3\vec{c}_2) + \\ & ((\vec{c}_1 \cdot \vec{m})^4 + (\vec{c}_2 \cdot \vec{m})^4)((\vec{c}_3 \cdot \vec{m})^3\vec{c}_3)) \end{aligned} \quad (21)$$

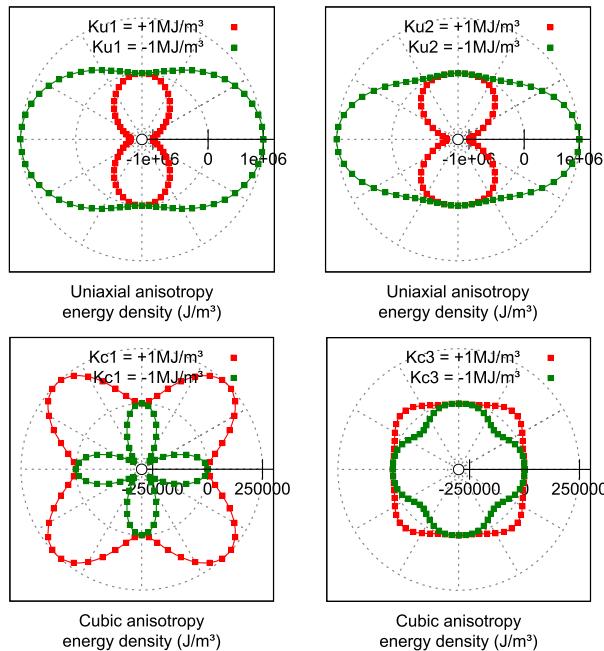


FIG. 7. Uniaxial (top) and cubic (bottom) anisotropy energy density of a single spin as a function of its orientation in the xy -plane. The uniaxial axis is along x , the cubic axes along x , y and z . The dots are computed with MuMAX3 (Eqs. (20) and (23)), lines are analytical expressions (Eq. (19), (22)). Positive and negative K values denote hard and easy anisotropy, respectively. In this case the 2nd order cubic energies are zero and not shown, but have been verified separately for angles out of the xy -plane.

where K_{cn} is the n th-order cubic anisotropy constant and $\vec{c}_1, \vec{c}_2, \vec{c}_3$ a set of mutually perpendicular unit vectors indicating the anisotropy directions. (The user only specifies \vec{c}_1 and \vec{c}_2 . We compute \vec{c}_3 automatically as $\vec{c}_1 \times \vec{c}_2$.) This corresponds to an energy density:

$$\begin{aligned} \mathcal{E}_{\text{anis}} = & \\ K_{c1} & ((\vec{c}_1 \cdot \vec{m})^2 (\vec{c}_2 \cdot \vec{m})^2 + \\ & (\vec{c}_1 \cdot \vec{m})^2 (\vec{c}_3 \cdot \vec{m})^2 + \\ & (\vec{c}_2 \cdot \vec{m})^2 (\vec{c}_3 \cdot \vec{m})^2) + \\ K_{c2} & (\vec{c}_1 \cdot \vec{m})^2 (\vec{c}_2 \cdot \vec{m})^2 (\vec{c}_3 \cdot \vec{m})^2 + \\ K_{c3} & ((\vec{c}_1 \cdot \vec{m})^4 (\vec{c}_2 \cdot \vec{m})^4 + \\ & (\vec{c}_1 \cdot \vec{m})^4 (\vec{c}_3 \cdot \vec{m})^4 + \\ & (\vec{c}_2 \cdot \vec{m})^4 (\vec{c}_3 \cdot \vec{m})^4) \end{aligned} \quad (22)$$

which, just like in the uniaxial case, MuMAX3 computes using the effective field:

$$\begin{aligned} \mathcal{E}_{\text{anis}} = & -\frac{1}{4} \vec{B}_{\text{anis}}(K_{c1}) \cdot \vec{M} - \frac{1}{6} \vec{B}_{\text{anis}}(K_{c2}) \cdot \vec{M} \\ & - \frac{1}{8} \vec{B}_{\text{anis}}(K_{c3}) \cdot \vec{M} \end{aligned} \quad (23)$$

which is verified in Fig. 7.

F. Thermal fluctuations

MuMAX3 provides finite temperature by means of a fluctuating thermal field \vec{B}_{therm} according to Brown.²⁸

$$\vec{B}_{\text{therm}} = \vec{\eta}(\text{step}) \sqrt{\frac{2\mu_0\alpha k_B T}{B_{\text{sat}}\gamma_{\text{LL}}\Delta V\Delta t}} \quad (24)$$

where α is the damping parameter, k_B the Boltzmann constant, T the temperature, B_{sat} the saturation magnetization expressed in Tesla, γ_{LL} the gyromagnetic ratio (1/Ts), ΔV the cell volume, Δt the time step and $\vec{\eta}(\text{step})$ a random vector from a standard normal distribution whose value is changed after every time step.

Solver constraints \vec{B}_{therm} randomly changes between time steps. Therefore, only MuMAX3's Euler and Heun solvers (IV) can be used as they do not require torque continuity between steps. Additionally, with thermal fluctuations enabled we enforce a fixed time step Δt . This avoids feedback issues with adaptive time step algorithms.

Verification We test our implementation by calculating the thermal switching rate of a single (macro-)spin particle with easy uniaxial anisotropy. In the limit of a high barrier compared to the thermal energy, the switching rate f is known analytically to be:²⁹

$$f = \gamma_{\text{LL}} \frac{\alpha}{1 + \alpha^2} \sqrt{\frac{8K_{\text{ul}}^3 V}{2\pi M_{\text{sat}}^2 k T}} e^{-KV/kT} \quad (25)$$

Fig. 8 shows Arrhenius plots for the temperature-dependent switching rate of a particle with volume $V=(10 \text{ nm})^3$ and $K_{\text{ul}}=1\times 10^4$ or $2\times 10^4 \text{ J/m}^3$. The MuMAX3 simulations correspond well to Eq. (25).

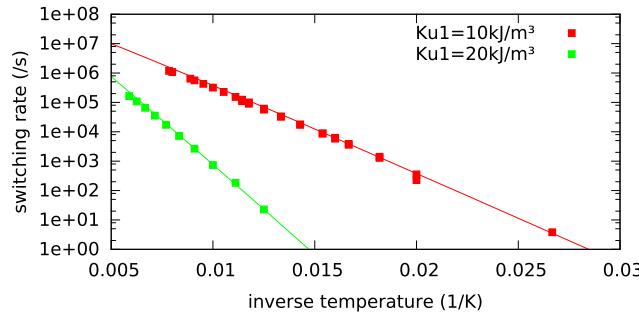


FIG. 8. Arrhenius plot of the thermal switching rate of a 10 nm large cubic particle (macrospin), with $M_{\text{sat}}=1\text{MA}/\text{m}$, $\alpha=0.1$, $\Delta t=10^{-12}\text{s}$, $K_{\text{u}1}=1\times 10^4$ or $2\times 10^4\text{ J/m}^3$. Simulations were performed on an ensemble of 512^2 uncoupled particles for $0.1\mu\text{s}$ (high temperatures) or $1\mu\text{s}$ (low temperatures). Solid lines are the analytically expected switching rates (Eq. (25)).

G. Zhang-Li Spin-transfer torque

MuMAX3 includes a spin-transfer torque term according to Zhang and Li,³⁰ applicable when electrical current flows through more than one layer of cells:

$$\vec{\tau}_{\text{ZL}} = \frac{1}{1+\alpha^2} ((1+\xi\alpha) \vec{\mathbf{m}} \times (\vec{\mathbf{m}} \times (\vec{\mathbf{u}} \cdot \nabla) \vec{\mathbf{m}}) + (\xi-\alpha) \vec{\mathbf{m}} \times (\vec{\mathbf{u}} \cdot \nabla) \vec{\mathbf{m}}) \quad (26)$$

$$\vec{\mathbf{u}} = \frac{\mu_B \mu_0}{2e\gamma_0 B_{\text{sat}}(1+\xi^2)} \vec{j} \quad (27)$$

where \vec{j} is the current density, ξ is the degree of non-adiabaticity, μ_B the Bohr magneton and B_{sat} the saturation magnetization expressed in Tesla.

The validity of our implementation is tested by Standard Problem #5 (Section V E).

H. Slonczewski Spin-transfer torque

MuMAX3 provides a spin momentum torque term according to Slonczewski,^{31,32} transformed to the Landau-Lifshitz formalism:

$$\vec{\tau}_{\text{SL}} = \beta \frac{\epsilon - \alpha \epsilon'}{1 + \alpha^2} (\vec{\mathbf{m}} \times (\vec{\mathbf{m}}_P \times \vec{\mathbf{m}})) - \beta \frac{\epsilon' - \alpha \epsilon}{1 + \alpha^2} \vec{\mathbf{m}} \times \vec{\mathbf{m}}_P \quad (28)$$

$$\beta = \frac{j_z \hbar}{M_{\text{sat}} e d} \quad (29)$$

$$\epsilon = \frac{P(\vec{\mathbf{r}}, t) \Lambda^2}{(\Lambda^2 + 1) + (\Lambda^2 - 1)(\vec{\mathbf{m}} \cdot \vec{\mathbf{m}}_P)} \quad (30)$$

where j_z is the current density along the z axis, d is the free layer thickness, $\vec{\mathbf{m}}_P$ the fixed-layer magnetization, P the spin polarization, the Slonczewski Λ parameter characterizes the spacer layer, and ϵ' is the secondary spin-torque parameter.

MuMAX3 only explicitly models the free layer magnetization. The fixed layer is handled in the same way as material parameters and is always considered to be on top of the free layer. The fixed layer's stray field is not automatically taken into account, but can be pre-computed by the user and added as a space-dependent external field term.

As a verification we consider switching an MRAM bit in $160\text{ nm} \times 80\text{ nm} \times 5\text{ nm}$ Permalloy ($M_{\text{sat}}=800\times 10^3\text{ A/m}$, $A_{\text{ex}}=13\times 10^{-12}\text{ J/m}^2$, $\alpha=0.01$, $P=0.5669$) by a total current of -6 mA along the z axis using $\Lambda=2$, $\epsilon'=1$. These parameters were chosen so that none of the terms in Eq. (28) are zero. The fixed layer is polarized at 20° from the x axis to avoid symmetry problems and the initial

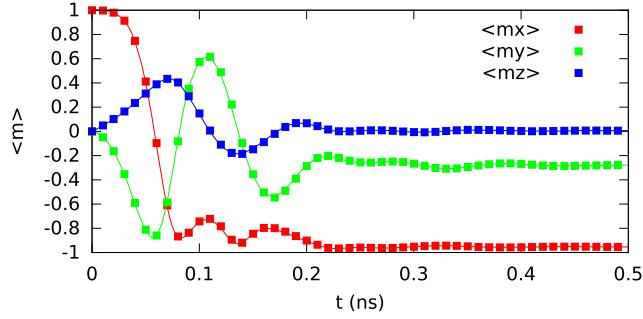


FIG. 9. Verification of the Slonczewski torque: average magnetization in a 160 nm × 80 nm × 5 nm rectangle with $M_{\text{sat}}=800 \times 10^3$ A/m, $A_{\text{ex}}=13 \times 10^{-12}$ J/m², $\alpha=0.01$, $P=0.5669$, $j_z=4.6875 \times 10^{11}$ A, $\Lambda=2$, $\epsilon'=1$, $\vec{m}_P=(\cos(20^\circ), \sin(20^\circ), 0)$, initial $m=(1,0,0)$. Solid line calculated with OOMMF, points by MuMax3 .

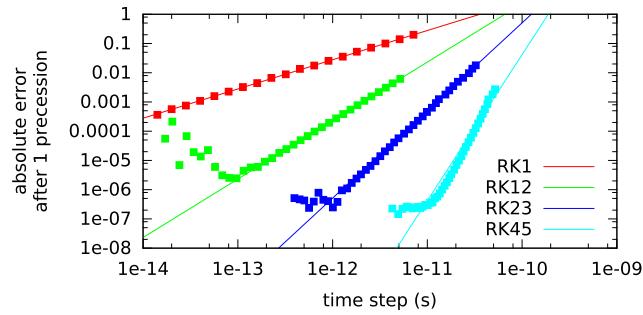


FIG. 10. Absolute error on a single spin after precessing without damping for one period in a 0.1 T field, as a function of different solver's time steps. The errors follow 1st, 2nd, 3rd or 5th order convergence (solid lines) for the respective solvers down to a limit set by the single precision arithmetic.

magnetization was chosen uniform along x . The MuMax3 and OOMMF results shown in Fig. 9 correspond well.

IV. TIME INTEGRATION

A. Dynamics

MuMax3 provides a number of explicit Runge-Kutta methods for advancing the Landau-Lifshitz equation (Eq. (2)):

- RK45, the Dormand-Prince method, offers 5-th order convergence and a 4-th order error estimate used for adaptive time step control. This is the default for dynamical simulations.
- RK32, the Bogacki-Shampine method, offers 3-th order convergence and a 2nd order error estimate used for adaptive time step control. This method is used when relaxing the magnetization to its ground state in which case it performs better than RK45.
- RK12, Heun's method, offers 2nd order convergence and a 1st order error estimate. This method is used for finite temperature simulations as it does not require torque continuity between time steps.
- RK1, Euler's method is provided for academic purposes.

These solvers' convergence rates are verified in Fig. 10, which serves as a test for their implementation and performance.

Adaptive time step RK45, RK23 and RK12 provide adaptive time step control, i.e., automatically choosing the time step to keep the error per step ϵ close to a preset value ϵ_0 . As the error per step we use $\epsilon = \max |\tau_{\text{high}} - \tau_{\text{low}}| \Delta t$, with τ_{high} and τ_{low} high-order and low-order torque estimates provided

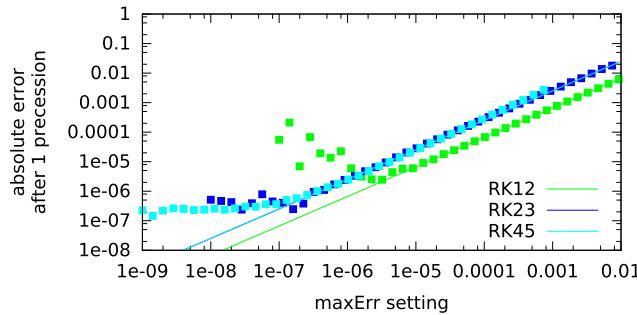


FIG. 11. Absolute error on a single spin after precessing without damping for one period in a 0.1 T field, as a function of different solver's MaxErr settings. Solid lines represent 1st order fits. The same lower bound as in Fig. 10 is visible.

by the particular Runge-Kutta method, and Δt the time step. The time step is adjusted using a default headroom of 0.8.

In MuMax3, ϵ_0 is accessible as the variable MaxErr. Its default value of 10^{-5} was adequate for the presented standard problems. The relation between ϵ_0 and the overall error at the end of the simulation is in general hard to determine. Nevertheless, we illustrate this in Fig. 11 for a single period of spin precession under the same conditions as Fig. 10. It can be seen that the absolute error per precession scales linearly with ϵ_0 , although the absolute value of the error depends on the solver type and exact simulation conditions.

B. Energy minimization

MuMax3 provides a `relax()` function that attempts to find the systems' energy minimum. This function disables the precession term Eq. (2), so that the effective field points towards decreasing energy. Relax first advances in time until the total energy cuts into the numerical noise floor. At that point the state will be close to equilibrium already. We then begin monitoring the magnitude of the torque instead of the energy, since close to equilibrium the torque will decrease monotonically and is less noisy than the energy. So we advance further until the torque cuts into the noise floor as well. Each time that happens, we decrease MaxErr and continue further until $\text{MaxErr}=10^{-9}$. At this point it does not make sense to increase the accuracy anymore (see Fig. 11) and we stop advancing.

This Relax procedure was used in the presented standard problems, where it proved adequate. Typical residual torques after Relax are of the order of $10^{-4}\text{--}10^{-7} \gamma_{\text{LL}} \text{T}$, indicating that the system is indeed very close to equilibrium. Nevertheless, as with any energy minimization technique, there is always a possibility that the system ends up in a saddle point or very flat part of the energy landscape.

Relax internally uses the RK23 solver, which we noticed performs better than RK45 in most relaxation scenarios. Near equilibrium, both solvers tend to take similarly large time steps, but RK23 needs only half as many torque evaluations per step as RK45.

V. STANDARD PROBLEMS

In this section we provide solutions to micromagnetic standard problems #1–4 provided by the μ Mag modeling group¹² and standard problem #5 proposed by Najafi *et al.*³³ Reference solutions were taken from Ref. 12 as noted, or otherwise calculated with OOMMF 1.2 alpha 5 bis.¹

A. Standard Problem #1

The first μ Mag standard problem involves the hysteresis loops of a $1\text{ }\mu\text{m} \times 2\text{ }\mu\text{m} \times 20\text{ nm}$ Permalloy rectangle ($A_{\text{ex}} = 1.3 \times 10^{-11} \text{ J/m}$, $M_{\text{sat}} = 8 \times 10^5 \text{ A/m}$, $K_{\text{u1}} = 5 \times 10^2 \text{ J/m}^3$ uniaxial, with easy axis nominally parallel to the long edges of the rectangle) for the field approximately parallel

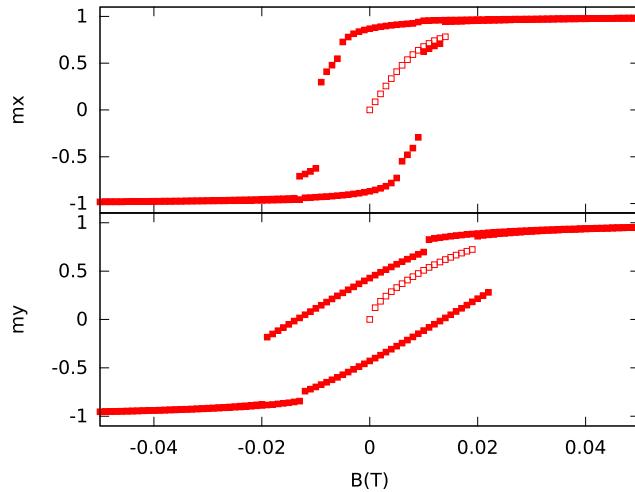


FIG. 12. MuMax3 solution for standard problem #1, using a 2D grid of 3.90625 nm wide cells. Open symbols represent the virgin curve starting from a vortex state. After each field step we applied thermal fluctuations with $\alpha = 0.05$, $T = 300K$ for 500ps to allow the magnetization to jump over small energy barriers. There are no consistent standard solutions to compare with.

to the long and short axis, respectively. Our solution is presented in Fig. 12. Unfortunately the submitted solutions¹² do not agree with each other, making it impossible to assess the correctness in a quantitative way.

B. Standard Problem #2

The second μ Mag standard problem considers a thin film of width d , length $5d$ and thickness $0.1d$, all expressed in terms of the exchange length $l_{\text{ex}} = \sqrt{2A_{\text{ex}}/\mu_0 M_{\text{sat}}^2}$. The remanence and coercitive field, expressed in units M_{sat} , are to be calculated as a function of d/l_{ex} .

The remanence is compared to OOMMF in Fig. 13. The coercivity, shown in Fig. 14, behaves interestingly in the small-particle limit where an analytical solution exists.¹⁹ In that case the magnetization is uniform and the magnetostatic field dominates the behaviour. Of the solutions submitted to the μ Mag group,^{19,34–36} the Streibl,³⁴ Donahue¹⁹ (OOMMF 1.1) and MuMax3 results best approach the small-particle limit. It was shown by Donahue *et al.*¹⁹ that proper averaging of the magnetostatic field over each cell volume is needed to accurately reproduce the analytical limit. Hence this standard problem serves as a test for the numerical integration of our demagnetizing kernel.

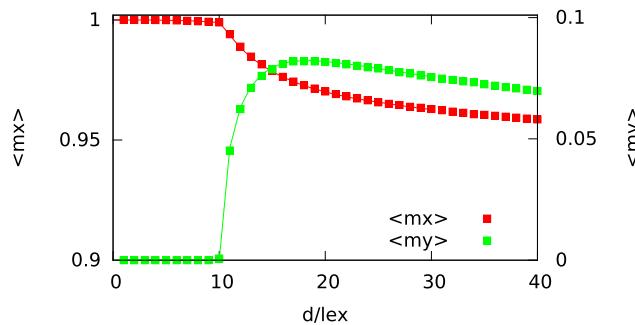


FIG. 13. Remanence for standard problem #2 as a function of the magnet size d expressed in exchange lengths l_{ex} . The MuMax3 calculations (points) use automatically chosen cell sizes between 0.25 and $0.5 l_{\text{ex}}$. OOMMF results (line) were taken from Ref. 12.

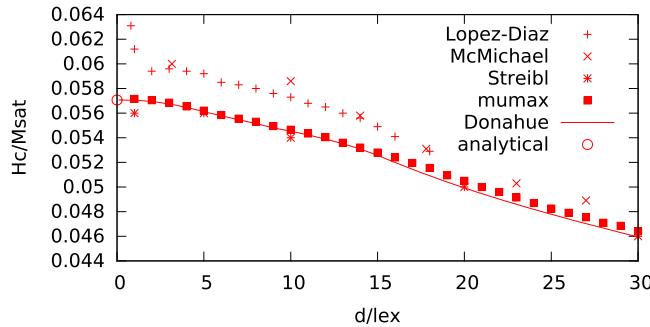


FIG. 14. Coercivity for standard problem #2 as a function of the magnet size d expressed in exchange lengths l_{ex} . MuMax3 calculations (points) use automatically chosen cell sizes between 0.25 and $0.5l_{\text{ex}}$. OOMMF results (line) taken from Ref. 12. The slight discrepancy at high d is attributed to OOMMF's solution using larger cells there. The analytical limit for very small size is by Donahue *et al.*¹⁹

C. Standard Problem #3

Standard problem #3 considers a cube with edge length L expressed in exchange lengths $l_{\text{ex}} = \sqrt{2A_{\text{ex}}/\mu_0 M_{\text{sat}}^2}$. The magnet has uniaxial anisotropy with $K_{u1} = 0.1K_m$, with $K_m = 1/2\mu_0 M_{\text{sat}}^2$, easy axis parallel to the z -axis. The critical edge size L where the ground state transitions between a quasi-uniform and vortex-like state needs to be found, it is expected around $L=8$.

This problem was solved using a $16 \times 16 \times 16$ grid. The cube was initialized with $\sim 3,000$ different random initial magnetization states for random edge lengths L between 7.5 and 9, and relaxed to equilibrium. Four stable states were found, shown in Fig. 15: a quasi-uniform flower state (a), twisted flower state (b), vortex state (c) and a canted vortex (d). Then cubes of different sizes were initialized to these states and relaxed to equilibrium. The resulting energy for each state, shown in Fig. 15, reveals the absolute ground states in the considered range: flower state for $L < 8.16$, twisted flower for $8.16 < L < 8.47$ and vortex for $L > 8.47$.

The transition at $L=8.47$ is in quantitative agreement with the solutions posted to μ Mag by Rave *et al.*³⁷ and by Martins *et al.*¹² The existence of the twisted flower state was already noted by Hertel *et al.*,¹² although without determining the flower to twisted flower transition point.

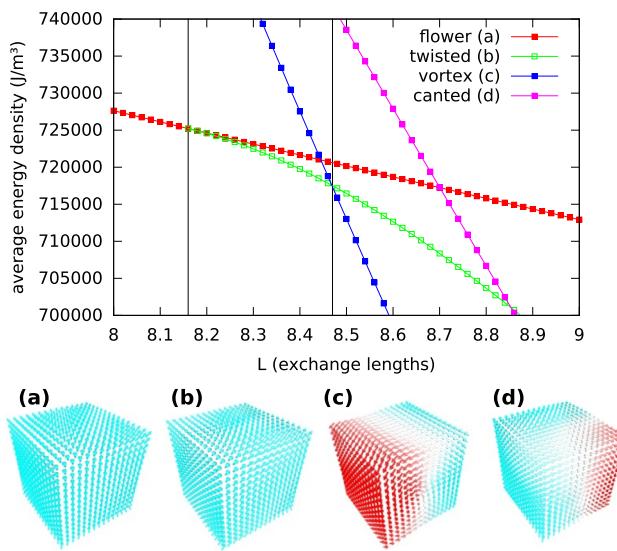


FIG. 15. Standard problem #3: energy densities of the flower (a), twisted flower (b), vortex (c) and canted vortex (d) states as a function of the cube edge length L . Transitions of the ground state are marked with vertical lines at $L = 8.16$ and $L = 8.47$.

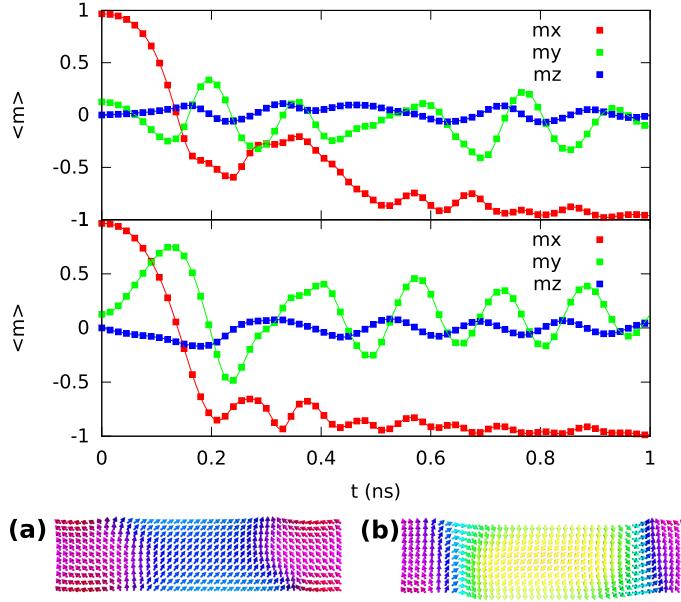


FIG. 16. MuMax3 (dots) and OOMMF (lines) solution to standard problem #4a (top graph) and #4b (bottom graph), as well as space-dependent magnetization snapshots when $\langle m_x \rangle$ crosses zero, for fields (a) and (b). All use a $200 \times 50 \times 1$ grid.

D. Standard Problem #4

Standard problem #4 considers dynamic magnetization reversal in a $500 \text{ nm} \times 125 \text{ nm} \times 3 \text{ nm}$ Permalloy magnet ($A_{\text{ex}}=1.3 \times 10^{-11} \text{ J/m}$, $M_{\text{sat}}=8 \times 10^5 \text{ A/m}$). The initial state is an S-state obtained after saturating along the (1,1,1) direction. Then the magnet is reversed by either field (a): (-24.6, 4.3, 0) mT or field (b): (-35.5, -6.3, 0) mT. Time-dependent average magnetizations should be given, as well as the space-dependent magnetization when $\langle m_x \rangle$ first crosses zero.

Our solution, shown in Fig. 16, agrees with OOMMF.

E. Standard Problem #5

Standard problem #5 proposed by Najafi *et al.*³³ considers a $100 \text{ nm} \times 100 \text{ nm} \times 10 \text{ nm}$ Permalloy square ($A = 13 \times 10^{-12} \text{ J/m}$, $M_{\text{sat}} = 8 \times 10^5 \text{ A/m}$, $\alpha = 0.1$, $\xi = 0.05$) with an initial vortex magnetization. A homogeneous current $\mathbf{j} = 10^{12} \text{ Am}^{-2}$ along x , applied at $t = 0$ drives the vortex towards a new equilibrium state. The obtained time-dependent average magnetization, shown in Fig. 17, agrees well the OOMMF solution.

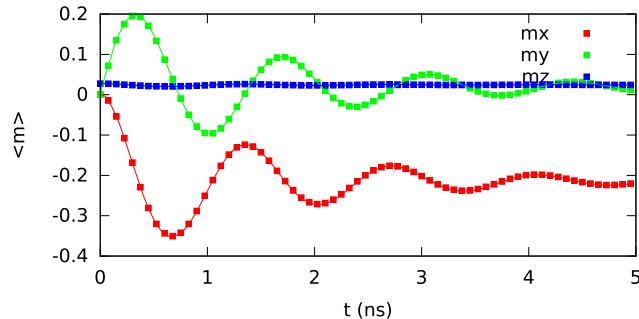


FIG. 17. MuMax3 (dots) and OOMMF (lines) solution to standard problem #5, both using a $50 \times 50 \times 5$ grid.

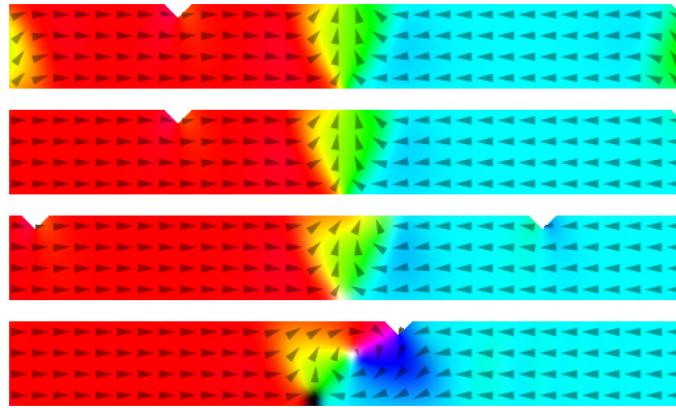


FIG. 18. Top frame: magnetization in a $1\text{ }\mu\text{m}$ wide, 20 nm thick Permalloy wire of finite length. The remaining frames apply edge charge removal to simulate an infinitely long wire. The domain wall is driven by a $3 \times 10^{12}\text{ A/m}^2$ current while being followed by the simulation window. So that it appears steady although moving at high speed (visible by the wall breakdown). While moving, new notches enter the simulation from the right.

VI. EXTENSIONS

MuMAX3 is designed to be modular and extensible. Some of our extensions, described below, have been merged into the mainline code because they may be of general interest. Nevertheless, extensions are considered specific to certain needs and are less generally usable than the aforementioned main features. For instance, MFM images and Voronoi tessellation are only implemented in 2D and only qualitatively tested.

A. Moving frame

MuMAX3 provides an extension to translate the magnetization with respect to the finite difference grid (along the x -axis), inserting new values from the side. This allows the simulation window to seemingly “follow” a region of interest like domain wall moving in a long nanowire, without having to simulate the entire wire. MuMAX3 can automatically translate the magnetization to keep an average magnetization component of choice as close to zero as possible, or the user may arbitrarily translate \vec{m} from the input script.

When following a domain wall in a long transverse magnetized wire, we also provide the possibility to remove the magnetic charges on the ends of the wire. This simulates an effectively infinitely long wire without closure domains, as illustrated in Fig. 18.

Finally, when shifting the magnetization there is an option to also shift the material regions and geometry along. The geometry and material parameters for the “new”cells that enter the simulation from the side are automatically re-calculated so that new grains and geometrical features may seamlessly enter the simulation. This can be useful for, e.g., simulating a long racetrack with notches like illustrated in Fig. 18, or a moving domain wall in a grainy material as published in Ref. 38.

B. Voronoi Tessellation

MuMAX3 provides 2D Voronoi tessellation as a way to simulate grains in thin films, similar to OOMMF.³⁹ It is possible to have MuMAX3 set-up the regions map with grain-shaped islands, randomly colored with up to 256 region numbers (Fig. 19(a)). The material parameters in each of those regions can then be varied to simulate, e.g., grains with randomly distributed anisotropy axes or even change the exchange coupling between them (Fig. 19(b)).

Our implementation is compatible with the possibility to move the simulation window. E.g., when the simulation window is following a moving domain wall, new grains will automatically

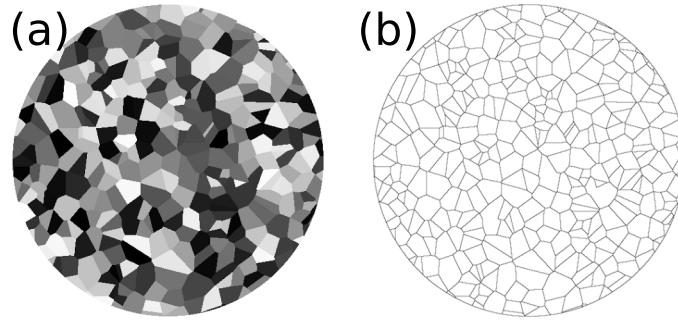


FIG. 19. Example of a Voronoi tessellation with average 100 nm grains in a 2048 μm wide disk. Left: cells colored by their region index (0–256). Right: boundaries between the grains visualized by reducing the exchange coupling between them (Eq. (9)), and outputting MuMax3’s ExchCoupling quantity, the average $M_{\text{sat}}/A_{\text{ex}}$ around each cell.

enter the simulation from the sides. The new grains are generated using hashes of the cell coordinates so that there is no need to store a (potentially very large) map of all the grains beyond the current simulation grid. More details can be found in Ref. 38.

C. Magnetic force microscopy

MuMax3 has a built-in capability to generate magnetic force microscopy (MFM) images in Dynamic (AC) mode from a 2D magnetization. We calculate the derivative of the force between tip and sample from the convolution:

$$\frac{\partial F_z}{\partial z} = \sum_{i=x,y,z} M_i(x, y) * \frac{\partial^2 B_{\text{tip},i}(x, y)}{\partial z^2} \quad (31)$$

where \vec{B}_{tip} is the tip’s stray field evaluated in the sample plane. MuMax3 provides the field of an idealized dipole or monopole tip with arbitrary elevation. No attempt is made to reproduce tip fields in absolute terms as our only goal is to produce output proportional to the actual MFM contrast, like shown in Fig. 20.

Eq. (31) is implemented using FFT-acceleration similar to the magnetostatic field, and is evaluated on the GPU. Hence MFM image generation is very fast and generally takes only a few milliseconds. This makes it possible to watch “real-time” MFM images in the web interface while the magnetization is evolving in time.

VII. PERFORMANCE

A. Simulation size

Nowadays, GPU’s offer massive computational performance of several TFlop/s per device. However, that compute power is only fully utilized in case of sufficient parallelization, i.e., for

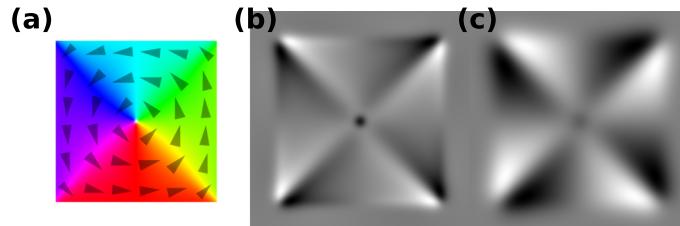


FIG. 20. (a) vortex magnetization in a 750 nm \times 750 nm \times 10 nm Permalloy square. (b), (c) are MuMax3-generated MFM images at 50 nm and 100 nm lift height respectively, both using AC mode and a monopole tip model.

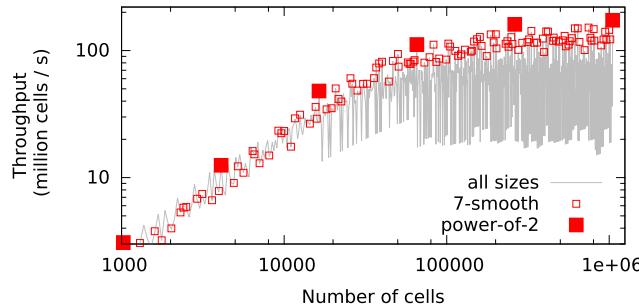


FIG. 21. MuMax3 throughput on GTX TITAN GPU, for all $N \times N$ grid sizes up to 1024×1024 . Numbers with only factors 2,3,5,7 are marked with an open box, pure powers of two (corresponding to Fig. 22) with a full box. Proper grid sizes should be chosen to ensure optimal performance.

sufficiently large simulations. This is clearly illustrated by considering how many cells can be processed per second, i.e., $N_{\text{cells}}/t_{\text{update}}$ with t_{update} the time needed to calculate the torque for N_{cells} cells. We refer to this quantity as the throughput. Given the overall complexity of $\mathcal{O}(N \log(N))$ one would expect a nearly constant throughput that slowly degrades at high N . For all presented throughputs, magnetostatic and exchange interactions were enabled and no output was saved.

The throughput presented in Fig. 21 for a square 2D simulation on a GTX TITAN GPU only exhibits the theoretical, nearly constant, behaviour starting from about 256 000 cells. Below, the GPU is not fully utilized and performance drops. Fortunately, large simulations are exactly where GPU speed-up is needed most and where performance is optimal.

MuMax3's performance is dominated by FFT calculations using the cuFFT library, which performs best for power-of-two sizes and acceptably for 7-smooth numbers (having only factors 2,3,5 and 7). Other numbers, especially primes should be avoided. This is clearly illustrated in Fig. 21 where other than the recommended sizes show a performance penalty of up to about an order of magnitude. So somewhat oversizing the grid up to a nice smooth number may be beneficial to the performance.

Note that the data in Fig. 22 is for a 2D simulation. Typically a 3D simulation with the same total number of cells costs an additional factor $\sim 1.5\times$ in compute time and memory due to additional FFTs along the z -axis.

On the other hand, simulations with periodic boundary conditions will run considerably faster than their non-periodic counterparts. This is due to the absence of zero-padding which reduces FFT sizes by 2 in each periodic direction. Memory consumption will be considerably lower as well. Only the one-time kernel initialization will take longer in this case, as a number of repeated images need to be taken into account here.

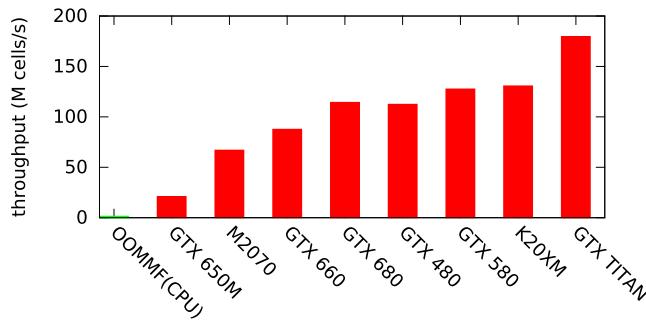


FIG. 22. MuMax3 throughput, measured in how many cells can have their torque evaluated per second (higher is better), for a 4×10^6 cell simulation (indicative for all sufficiently large simulations). For comparison, OOMMF performance on a quad-core 2.1 GHz CPU lies around 4 M cells/s.

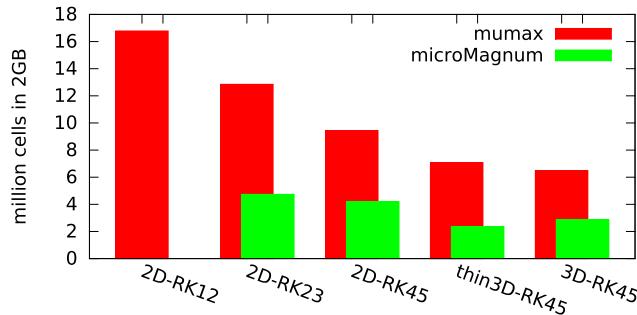


FIG. 23. Indication of the number of cells that can be addressed with 2 GB of GPU memory for simulations in 2D, 3D and thin 3D (here 3 layers) and using different solvers. RK45 is MuMAX3's default solver for dynamics, RK23 for relaxation. Only magnetostatic, exchange and Zeeman terms were enabled.

B. Hardware

Apart from the simulation size, MuMAX3's performance is strongly affected by the particular GPU hardware. We highlight the differences between several GPU models by comparing their throughput in Fig. 22. This was done for a 4 M cells simulation where all tested GPUs were fully utilized. So the numbers are indicative for all sufficiently large simulation sizes.

We also included OOMMF's throughput on a quad-core 2.1 GHz core i7 CPU to give a rough impression of the GPU speed-up. The measured OOMMF performance (not clearly distinguishable in Fig. 22) was around 4×10^6 cells/s. So with a proper GPU and sufficiently large grid sizes, a speed-up of 20–45× with respect to a quad-core can be reached or, equivalently, a 80–180× speed-up compared to a single-core CPU. This is in line with earlier MuMAX1 and MicroMagnum benchmarks.^{2,4} It must be noted however that OOMMF operates in double-precision in contrast to MuMAX3's single-precision arithmetic, and also does not suffer reduced throughput for small simulations.

Finally, MicroMagnum's throughput (not presented) was found to be nearly indistinguishable from MuMAX3. This is unsurprising since both MuMAX3's and MicroMagnum's performance are dominated by CUDA's FFT routines. In our benchmarks on a GTX650M, differences between both packages were comparable to the noise on the timings.

C. Memory use

In contrast to their massive computational power, GPUs are typically limited to rather small amounts of memory (currently 1–6 GB). Therefore, MuMAX3 was heavily optimized to use as little memory as possible. We exploited, for example, the magnetostatic kernel symmetries and zero elements and make heavy use of memory pooling and recycling.

Also, MuMAX3 employs minimal zero-padding in the common situation of 3D simulations with only a small number of layers. For up to 10 layers there is no need to use a power of two, and memory usage will be somewhat reduced as well.

In this way, MuMAX3 on a GPU with only 2 GB of memory is able to simulate about 9 million cells in 2D and 6 million in 3D, or about 2 × more than MicroMagnum v0.2² (see Fig. 23). When using a lower-order solver this number can be further increased to 12×10^6 cells with RK23 (2D) or 16×10^6 cells with RK12(2D), all in 2 GB. Cards like the GTX TITAN and K20XM, with 6 GB RAM can store proportionally more, e.g., 31 M cells for 2D with the RK45 solver.

VIII. CONCLUSION

We have presented in detail the micromagnetic model employed by MuMAX3, as well as a verification for each of its components. GPU acceleration provides a speed-up of 1–2 orders of

magnitude compared to CPU-based micromagnetic simulations. In addition, MuMAX3's low memory requirements open up the possibility of very large-scale micromagnetic simulations, a regime where the GPU's potential is fully utilized and where the speed-up is also needed most. Depending on the solver type MuMAX3 can fit 10–16 million cells in 2 GB GPU RAM—about 2× more than MuMax2 or MicroMagnum.

MuMAX3 is open-source and designed to be easily extensible, so anybody can in principle add functionality. Some extensions like a moving simulation window, edge charge removal, Voronoi tessellation and MFM images have been permanently merged into MuMAX3 and more extensions are expected in the future.

ACKNOWLEDGMENTS

This work was supported by the Flanders Research Foundation (FWO).

The authors would like to cordially thank (in alphabetical order): Ahmad Syukri bin Abdollah, Alex Mellnik, Aurelio Hierro, Ben Van de Wiele, Colin Jermain, Damien Louis, Ezio Iacocca, Gabriel Chaves, Graham Rowlands, Henning Ulrichs, Joo-Von Kim, Lasse Laurson, Raffaele Pelli-cell, Rémy Lassalle-Balier, Robert Stamps and Xuanyao (Kelvin) Fong for the fruitful discussions, contributions or feedback, as well as all others who tested early MuMAX versions.

MuMAX3 uses svgo (<http://github.com/ajstarks/svgo>), copyright Anthony Starks, and freetype-go (<http://code.google.com/p/freetype-go>), copyright Google Inc., Jeff R. Allen, Rémy Oudompheng, Roger Peppe.

- ¹ M. J. Donahue and D. G. Porter, OOMMF user's guide, version 1.0. interagency report NISTIR 6376, national institute of standards and technology, gaithersburg, MD, 1999.
- ² A. Drews, G. Selke, C. Abert, T. Gerhardt, J. M. Meyer, C. Darsow-Fromm, M. Menzel, B. Krüger, and D. P. F. Möller, *Micromagnum* (2011). <http://micromagnum-tis.informatik.uni-hamburg.de/>.
- ³ L. Lopez-Diaz, D. Aurelio, L. Torres, E. Martinez, M. A. Hernandez-Lopez, J. Gomez, O. Alejos, M. Carpentieri, G. Finocchio, and G. Consolo, "Micromagnetic simulations using graphics processing units," *Journal of Physics D: Applied Physics* **45**(32), 323001 (2012).
- ⁴ A. Vansteenkiste and B. Van de Wiele, "Mumax: A new high-performance micromagnetic simulation tool," *Journal of Magnetism and Magnetic Materials* **323**(21), 2585–2591 (2011).
- ⁵ A. Vansteenkiste, B. Van de Wiele, M. Dvornik, R. Lassalle-Balier, and G. Rowlands, mumax2, 2012. <http://code.google.com/p/mumax2>.
- ⁶ The Go Programming Language, 2009. <http://golang.org>.
- ⁷ NVIDIA CUDA C programming guide, 2014. <http://developer.nvidia.com/nvidia-gpu-computing-documentation>.
- ⁸ T. Fischbacher, M. Franchin, G. Bordignon, and H. Fangohr, "A systematic approach to multiphysics extensions of finite-element-based micromagnetic simulations: Nmag," *IEEE Transactions On Magnetics* **43**(6), 2896–2898 (2007).
- ⁹ A. Kakay, E. Westphal, and R. Hertel, "Speedup of FEM micromagnetic simulations with graphical processing units," *IEEE Transactions On Magnetics* **46**(6), 2303–2306 (2010).
- ¹⁰ W. Scholz, J. Fidler, T. Schrefl, D. Suess, R. Dittrich, H. Forster, and V. Tsiantos, "Scalable parallel micromagnetic solvers for magnetic nanostructures," *Comput. Mater. Sci.* **28**, 366–383 (2003).
- ¹¹ R. Chang, S. Li, M. V. Lubarda, B. Livshitz, and V. Lomakin, "Fastmag: Fast micromagnetic simulator for complex magnetic structures," *Journal of Applied Physics* **109**(7), 07D358 (2011).
- ¹² muMAG Micromagnetic Modeling Activity Group <http://www.ctcms.nist.gov/~rdm/mumag.org.html>.
- ¹³ Paraview. www.paraview.org.
- ¹⁴ Gnuplot. <http://www.gnuplot.info>.
- ¹⁵ Graham Rowlands. Muvview2. <http://grahamrow.github.io/Muvview2/>.
- ¹⁶ L. D. Landau and E. M. Lifshitz, "Theory of the dispersion of magnetic permeability in ferromagnetic bodies," *Phys. Z. Sowjetunion* **8**, 153–169 (1935).
- ¹⁷ T. L. Gilbert, "Lagrangian formulation of the gyromagnetic equation of the magnetization field," *Phys. Rev.* **100**, 1243–1243 (1955).
- ¹⁸ R. D. McMichael, M. J. Donahue, D. G. Porter, and Jason Eicke, "Comparison of magnetostatic field calculation methods on two-dimensional square grids as applied to a micromagnetic standard problem," *Journal of Applied Physics* **85**(8), 5816–5818 (1999).
- ¹⁹ M. J. Donahue, D. G. Porter, R. D. McMichael, and J. Eicke, "Behavior of mu mag standard problem no. 2 in the small particle limit," *Journal of Applied Physics* **87**(9), 5520–5522 (2000).
- ²⁰ K. M. Lebecki, M. J. Donahue, and M. W. Gutowski, "Periodic boundary conditions for demagnetization interactions in micromagnetic simulations," *Journal of Physics D-applied Physics* **41**(17), (2008).
- ²¹ Hans Fangohr, Giuliano Bordignon, Matteo Franchin, Andreas Knittel, Peter A. J. de Groot, and Thomas Fischbacher, "A new approach to (quasi) periodic boundary conditions in micromagnetics: The macrogeometry," *Journal of Applied Physics* **105**(7): Phys Conf Inc; IEEE, Magnet Soc, April 2009.
- ²² W. F. Brown, Jr., *Micromagnetics* (Interscience Publishers, New York, NY, 1963).

- ²³ M. J. Donahue, "A variational approach to exchange energy calculations in micromagnetics," *Journal of Applied Physics* **83**(11), 6491–6493 (1998).
- ²⁴ M. J. Donahue and D. G. Porter, "Exchange energy formulations for 3d micromagnetics," *Physica B-condensed Matter* **343**(1-4), 177–183 (2004).
- ²⁵ A. N. Bogdanov and U. K. Rößler, "Chiral symmetry breaking in magnetic thin films and multilayers," *Phys. Rev. Lett.* **87**, 037203 (2001).
- ²⁶ S. Rohart and A. Thiaville, "Skyrmion confinement in ultrathin film nanostructures in the presence of dzyaloshinskii-moriya interaction," *Phys. Rev. B* **88**, 184422 (2013).
- ²⁷ Andre Thiaville, Stanislas Rohart, Emilie Jue, Vincent Cros, and Albert Fert, "Dynamics of dzyaloshinskii domain walls in ultrathin magnetic films," *Epl* **100**(5), 57002 (2012).
- ²⁸ William Fuller Brown, "Thermal fluctuations of a single-domain particle," *Journal of Applied Physics* **34**(4), 1319–1320 (1963).
- ²⁹ L. Breth, D. Suess, C. Vogler, B. Bergmair, M. Fuger, R. Heer, and H. Brueckl, "Thermal switching field distribution of a single domain particle for field-dependent attempt frequency," *Journal of Applied Physics* **112**(2), (2012).
- ³⁰ S. Zhang and Z. Li, "Roles of nonequilibrium conduction electrons on the magnetization dynamics of ferromagnets," *Physical Review Letters* **93**(12), 127204 (2004).
- ³¹ J. C. Slonczewski, "Current-driven excitation of magnetic multilayers," *Journal of Magnetism and Magnetic Materials* **159**(1-2), L1–L7 (1996).
- ³² J. Xiao, A. Zangwill, and M. D. Stiles, "Boltzmann test of slonczewski's theory of spin-transfer torque," *Physical Review B* **70**(17), 172405 (2004).
- ³³ M. Najafi, B. Kruger, S. Bohlens, M. Franchin, H. Fangohr, A. Vanhaverbeke, R. Allenspach, M. Bolte, U. Merkt, D. Pfannkuche, D. P. F. Moller, and G. Meier, "Proposal for a standard problem for micromagnetic simulations including spin-transfer torque," *Journal of Applied Physics* **105**(11), 113914 (2009).
- ³⁴ B. Streibl, T. Schrefl, and J. Fidler, "Dynamic fe simulation of μ mag standard problem no. 2," *Journal of Applied Physics* **85**(8), 5819–5821 (1999).
- ³⁵ L. Lopez-Diaz, O. Alejos, L. Torres, and J. I. Iniguez, "Solutions to micromagnetic standard problem no. 2 using square grids," *Journal of Applied Physics* **85**(8), 5813–5815 (1999).
- ³⁶ R. D. McMichael, M. J. Donahue, D. G. Porter, and J. Eicke, "Comparison of magnetostatic field calculation methods on two-dimensional square grids as applied to a micromagnetic standard problem," *Journal of Applied Physics* **85**(8), 5816–5818 (1999).
- ³⁷ W. Rave, K. Fabian, and A. Hubert, "Magnetic states of small cubic particles with uniaxial anisotropy," *Journal of Magnetism and Magnetic Materials* **190**(3), 332–348 (1998).
- ³⁸ J. Leliaert, B. Van de Wiele, A. Vansteenkiste, L. Laurson, G. Durin, L. Dupré, and B. Van Waeyenberge, "Current-driven domain wall mobility in polycrystalline permalloy nanowires: A numerical study," *Journal of Applied Physics* **115**(23), (2014).
- ³⁹ J. W. Lau, R. D. McMichael, and M. J. Donahue, "Implementation of Two-Dimensional Polycrystalline Grains in Object Oriented Micromagnetic Framework," *Journal of Research of NIST* **114**(1), 57–67 (2009).