



ISTITUTO NAZIONALE DI RICERCA METROLOGICA Repository Istituzionale

Adaptive geometric integration applied to a 3D micromagnetic solver

This is the author's accepted version of the contribution published as:

Original

Adaptive geometric integration applied to a 3D micromagnetic solver / Ferrero, Riccardo; Manzin, Alessandra. - In: JOURNAL OF MAGNETISM AND MAGNETIC MATERIALS. - ISSN 0304-8853. - 518:(2021), p. 167409. [10.1016/j.jmmm.2020.167409]

Availability:

This version is available at: 11696/64516 since: 2021-01-13T17:44:05Z

Publisher:

Elsevier

Published

DOI:10.1016/j.jmmm.2020.167409

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright

(Article begins on next page)

Adaptive geometric integration applied to a 3D micromagnetic solver

Riccardo Ferrero^a and Alessandra Manzin^a

^a*Istituto Nazionale di Ricerca Metrologica (INRIM), Strada delle Cacce 91, 10135 Torino, Italy*

Corresponding author:

Riccardo Ferrero

Istituto Nazionale di Ricerca Metrologica

Strada delle Cacce 91

10135 Torino, Italy

tel. +39-011-3919825

e-mail: r.ferrero@inrim.it

Abstract

This paper presents a GPU-parallelized 3D micromagnetic code for the efficient calculation of the magnetization dynamics, equilibrium configuration and static hysteresis loops of magnetic nanostructures, by solving the Landau-Lifshitz-Gilbert (LLG) equation. The time-integration of the LLG equation is carried out by using a technique based on the Cayley transform, which allows us to fulfil the constraint on the magnetization amplitude. The computational domain is reconstructed with a structured hexahedral mesh. The spatial-integration of the magnetostatic field is performed via a Fast Fourier Transform (FFT) algorithm, and the exchange field is computed with a 26-node-based finite difference technique. A careful validation of the developed solver was carried out, also by comparison to OOMMF and MuMax3. Then, we analysed the computational efficiency of the geometrical time-integrator and of its time-adaptive variant, investigating the role of the numerical damping introduced by the Cayley transform-based time-discretization.

Keywords: Micromagnetics; Landau-Lifshitz-Gilbert equation; GPU computing; Numerical modelling.

1. Introduction

The numerical integration of the Landau-Lifshitz-Gilbert (LLG) equation can be computationally very demanding, due to the need of simulating phenomena at the exchange length scale (5-10 nm) as well as long-range interactions, like the magnetostatic field. Additionally, particular care has to be devoted to the choice of the time-integration scheme, which should guarantee the preservation of the magnetization amplitude and the Lyapunov structure of the LLG equation, with sufficiently large time-steps [1]. This task becomes very critical in the computation of static hysteresis loops, since at each field update the magnetization time evolution has to be calculated up to the reaching of equilibrium.

In general, both explicit and implicit methods do not guarantee that the peculiar properties of the LLG equation are preserved. A plethora of solutions and approaches have been proposed during the years. Explicit time-integration schemes impose strong limitations on the time-steps due to their limited stability region. To overcome this limitation, semi-analytical [2, 3] or high-order Runge-Kutta [4, 5] methods can be used. Implicit methods show good stability performance, however, they do not generally guarantee the preservation of the magnetization amplitude nor the energy conservation properties of the LLG equation. They also tend to be computationally heavy, requiring to solve large coupled systems of non-linear equations, therefore it is preferred to use semi-implicit methods [6–11] or fixed-point algorithms [12].

Advantages can be found in geometrical integrators, which are a family of solvers able to preserve the LLG equation properties and, before being applied in micromagnetics, were used in many research fields in computational mechanics, where non-convex constraints appear [13–16]. In micromagnetics, geometrical integrators were developed using semi-implicit algorithms and quasi-Newton method [1]. However, these approaches require the implementation of iterative techniques; therefore, they need that, at each iteration, the effective field and thus the magnetostatic field are recalculated, resulting in very time-consuming operations.

To face the above problems, we performed the time-integration of the LLG equation by means of an explicit method based on the Cayley transform [17–20], which enables us to guarantee the preservation of the constraint on the magnetization independently of integration scheme order and time-step size. First, we incorporated a first-order (Euler) and a second-order (Heun) time-integration scheme. To further improve computational performance, an adaptive time-integration scheme based on the Euler-Heun method was implemented, applying a technique developed for the embedded Runge-Kutta methods for the evaluation of the local truncation error [21–23]. The proposed time-integration scheme was incorporated within a 3D micromagnetic solver, which implements a Fast Fourier Transform (FFT)-based approach for the magnetostatic field evaluation [24–27], and exploits GPU-parallelization to accelerate both the spatial- and time-integration of the LLG equation [28–31]. The solver was already successfully applied to investigate the hysteresis properties of 3D magnetic nanostructures for possible hyperthermia applications [32].

First, we performed a careful validation of the solver focusing on Standard Problem #4, proposed by the *Micromagnetic Modeling Activity Group* (μ MAG) at the US *National Institute of Standards and Technology* (NIST) [33]. Then, we tested the code in the calculation of the static hysteresis loop of a 3D sample, by comparison to MuMax3 [34]. Finally, we investigated the computational efficiency of the Cayley transform-based time-integration scheme, showing the advantages and potential drawbacks of the artificial damping introduced numerically as well as of the time-adaptive integration.

2. Numerical method

The time evolution of the magnetization vector $\mathbf{M}(t, x, y, z)$ in a tridimensional magnetic sample Ω is described following the LLG equation:

$$\frac{\partial \mathbf{M}}{\partial t} = \frac{\gamma}{(1 + \alpha^2)} \left[\mathbf{H}_{\text{eff}} + \frac{\alpha}{M_s} (\mathbf{M} \times \mathbf{H}_{\text{eff}}) \right] \times \mathbf{M}, \quad (1)$$

where $\gamma = 2.21 \times 10^5 \text{ mA}^{-1}\text{s}^{-1}$ is the absolute value of the gyromagnetic ratio, α is the damping constant, M_S is the saturation magnetization and \mathbf{H}_{eff} is the effective field, which is the sum of four contributions: the externally applied field \mathbf{H}_a ; the exchange field \mathbf{H}_{ex} ; the magnetostatic (or demagnetizing) field \mathbf{H}_{ms} ; the anisotropy field \mathbf{H}_{an} .

The magnetic domain Ω is discretized using hexahedral finite difference cells, in which vectors \mathbf{M} and \mathbf{H}_{eff} are assumed to be uniform. In the following sections we describe the time- and spatial-integration procedures.

2.1 Time-integration

In the generic n -th finite difference cell, equation (1) can be written in the compact form:

$$\dot{\mathbf{M}}^n(t) = \mathbf{A}(\mathbf{M}^n) \times \mathbf{M}^n, \quad (2)$$

where $\mathbf{A}(\mathbf{M})$ is the generator. Given the properties of the external product, \mathbf{A} can be generalized by adding a term parallel to \mathbf{M} , resulting in the vector quantity $\boldsymbol{\omega} = \mathbf{A}(\mathbf{M}) + \sigma(\mathbf{M})\mathbf{M}$, where σ is an arbitrary scalar function, here generally assumed to be constant in space and time. Then, we numerically solve the following equation:

$$\dot{\mathbf{M}}^n(t) = \boldsymbol{\omega}(\mathbf{M}^n) \times \mathbf{M}^n. \quad (3)$$

Since magnetization dynamics evolves on the surface of a sphere, there exists a family of curves

$$Q(t): \Omega \rightarrow SO(3) / \mathbf{M}^n(t) = Q(t)\mathbf{M}_0^n, \quad (4)$$

with \mathbf{M}_0^n being the magnetization vector at the time instant $t = 0$. $Q(t)$ belongs to the special orthogonal Lie-group $SO(3)$. The Lie-Algebra $\mathfrak{so}(3)$ associated to the group consists of the 3×3 skew real matrices that describe infinitesimal rotations. Thanks to the isomorphism between the Lie-algebra induced by the external product in \mathbb{R}^3 and the Lie-Algebra $\mathfrak{so}(3)$, the following equations hold true:

$$\begin{aligned} \dot{\mathbf{M}}^n(t) &= \dot{Q}(t) \mathbf{M}_0^n, \\ \dot{Q}(t) &= \text{skew}[\boldsymbol{\omega}(Q(t) \mathbf{M}_0^n)]Q(t), \end{aligned} \quad (5)$$

$$\text{where } \text{skew}[\boldsymbol{\omega}] = \begin{pmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{pmatrix}.$$

In this framework, it is possible to derive time-integration schemes of any order using the Cayley transform:

$$\text{cay}(\mathbf{v}) = (\mathbf{I} + \text{skew}[\mathbf{v}/2])(\mathbf{I} - \text{skew}[\mathbf{v}/2]), \quad (6)$$

which is a second-order approximation of the algorithmic exponential for the rotation group [16, 18, 35]. In this way, magnetization amplitude is intrinsically preserved.

In our code, we implemented three time-integration schemes; for the first two ones the time-step Δt is fixed. In the first-order scheme (CAY1), based on the Euler explicit algorithm, the magnetization at the $(i+1)$ -th instant is computed as

$$\mathbf{M}_{i+1}^n = \text{cay}(\Delta t \boldsymbol{\omega}_i^n) \mathbf{M}_i^n. \quad (7)$$

In the second-order scheme (CAY2), based on the Heun explicit algorithm, the magnetization at the $(i+1)$ -th instant is evaluated as

$$\mathbf{M}_{i+1}^n = \text{cay}(\Delta t \tilde{\boldsymbol{\omega}}^n) \mathbf{M}_i^n, \quad (8)$$

where $\tilde{\boldsymbol{\omega}}^n = [\boldsymbol{\omega}_*^n(\mathbf{M}_*^n) + \boldsymbol{\omega}_i^n(\mathbf{M}_i^n)]/2$ and $\mathbf{M}_*^n = \text{cay}(\Delta t \boldsymbol{\omega}_i^n) \mathbf{M}_i^n$.

Scheme CAY2 results equivalent to CAY1 with the addition of a step.

The third scheme (CAY12) is a time-adaptive embedded second-order algorithm, which is based on CAY2 and uses the first-order solution \mathbf{M}_* provided by the first step to estimate the local truncation error and thus chose an appropriate time-step for the evaluation of \mathbf{M}_{i+1} . Since CAY12 uses a second-order formula we expect that the truncation error at the $(i+1)$ -th instant ε_{i+1} scales with the square of the chosen time-step $\Delta t_{i+1} = t_{i+1} - t_i$. In general, one can predict the time-step Δt_* needed to obtain a desired accuracy ε_* from the following equation:

$$\Delta t_* = \Delta t_{i+1} \sqrt{\left| \frac{\varepsilon_*}{\varepsilon_{i+1}} \right|}. \quad (9)$$

If the truncation error is smaller than the desired accuracy, Δt_{i+1} can be used as the time-step to calculate the solution at t_{i+1} , otherwise the solution is discarded and the $(i+1)$ -th instant is recalculated using Δt_* [36].

The definition of the truncation error is a critical task and may depend on the peculiarity of the problem; here, we chose the following relationship:

$$\varepsilon_{i+1} = \max_n |\tau_{\text{CAY1}}^n - \tau_{\text{CAY2}}^n| \Delta t_{i+1}, \quad (10)$$

where τ_{CAY1}^n and τ_{CAY2}^n are the torques $\mathbf{M}_*^n \times \mathbf{H}_{\text{eff}_*}^n$ and $\mathbf{M}_{i+1}^n \times \mathbf{H}_{\text{eff}_{i+1}}^n$, respectively, obtained from the two steps of the Heun-based scheme, which provides, with the first step, the first-order approximation (equivalent to CAY1) and, with the second step, the second-order approximation (equivalent to CAY2). With this definition the computational cost of the time-adaptive scheme is minimized. By using a precautionary approach the time-step was adjusted introducing a multiplicative factor b generally equal to 0.8 when computing the solution at the new time-step, i.e. $\Delta t_* = b \Delta t_{i+1} \sqrt{\left| \frac{\varepsilon_*}{\varepsilon_{i+1}} \right|}$. Moreover, it is possible to specify an upper and a lower limit to the time-step to ensure a better control of the magnetization time evolution.

2.2 Spatial-integration

Particular care has to be devoted to the spatial-integration of the effective field and of its terms. The computation of the magnetostatic field is one of the most time consuming tasks in micromagnetic solvers because of its integral expression:

$$\mathbf{H}_{\text{ms}}(\mathbf{r}) = -\frac{1}{4\pi} \nabla \int_{\Omega} \nabla' \left(\frac{1}{|\mathbf{r}-\mathbf{r}'|} \right) \cdot \mathbf{M}(\mathbf{r}') dV'. \quad (11)$$

The finite difference spatial discretization here implemented and the assumption of uniform magnetization in each cell allow us to use the discrete convolution for the evaluation of \mathbf{H}_{ms} . This type of spatial discretization, which requires structured meshes, may present some limitations in the exact reconstruction of the sample, but the increase in computational efficiency achievable with the discrete convolution approach is considerable [4, 24-27, 37]. In particular, the implementation of the Fast Fourier Transform (FFT) algorithm [38,39] reduces the computational complexity from $\mathcal{O}(N^2)$ to $\mathcal{O}(N \log N)$, where N is the number of finite difference cells.

In the n -th cell \mathbf{H}_{ms} is expressed as:

$$\mathbf{H}_{\text{ms}}(\mathbf{r}_n) = \sum_{n \neq m} \mathbf{G}(\mathbf{r}_n - \mathbf{r}_m) \cdot \mathbf{M}(\mathbf{r}_m) \quad (12)$$

$$\mathbf{G}(\mathbf{r}) = \begin{pmatrix} g_{xx}(\mathbf{r}) & g_{xy}(\mathbf{r}) & g_{xz}(\mathbf{r}) \\ g_{xy}(\mathbf{r}) & g_{yy}(\mathbf{r}) & g_{yz}(\mathbf{r}) \\ g_{xz}(\mathbf{r}) & g_{yz}(\mathbf{r}) & g_{zz}(\mathbf{r}) \end{pmatrix},$$

where $\mathbf{G}(\mathbf{r})$ is the demagnetizing tensor. The spatial-domain convolution can be changed into a scalar product in the frequency domain using the Fourier transform. It results that:

$$\begin{cases} \widehat{H}_{ms,x}^n = \widehat{g}_{xx}^n \widehat{M}_x^n + \widehat{g}_{xy}^n \widehat{M}_y^n + \widehat{g}_{xz}^n \widehat{M}_z^n \\ \widehat{H}_{ms,y}^n = \widehat{g}_{xy}^n \widehat{M}_x^n + \widehat{g}_{yy}^n \widehat{M}_y^n + \widehat{g}_{yz}^n \widehat{M}_z^n, \\ \widehat{H}_{ms,z}^n = \widehat{g}_{xz}^n \widehat{M}_x^n + \widehat{g}_{yz}^n \widehat{M}_y^n + \widehat{g}_{zz}^n \widehat{M}_z^n \end{cases} \quad (13)$$

where $\widehat{\mathbf{M}}(\mathbf{r}) = FFT[\mathbf{M}(\mathbf{r})]$ and $\widehat{\mathbf{G}}(\mathbf{r}) = FFT[\mathbf{G}(\mathbf{r})]$ are the Fourier transforms of the magnetization vector and of the demagnetizing tensor, respectively. Now, applying the inverse FFT to the three components of $\widehat{\mathbf{H}}_{ms}$ it is possible to obtain the magnetostatic field in the spatial-domain.

To compute the demagnetizing tensor, we implemented two methods. The first one uses the analytical formulation proposed by Nakatani et al. [40], the second one solves the integrals in (12) numerically, employing the Green's surface integral formula with 36×36 quadrature nodes. Since this tensor depends only on the geometrical properties of the magnetic sample, it is calculated, FFT-transformed and stored, once at the beginning of the simulation, and then reused at each time instant.

The exchange field $\mathbf{H}_{ex} = \frac{2k_{ex}}{\mu_0 M_S^2} \Delta \mathbf{M}$, with exchange constant k_{ex} , was calculated using a standard finite difference scheme, which can be easily implemented for structured meshes. For the Laplacian operator we used both a 6-neighbor formula, accurate up to the second-order, and a 26-neighbour formula, accurate up to the fourth-order [41, 42], i.e.

$$\mathbf{H}_{ex}(\mathbf{r}_n) = \frac{2k_{ex}}{\mu_0 M_S^2} \sum_{q=1}^Q \beta_q \frac{(\mathbf{M}_q - \mathbf{M}_n)}{a_q(x_q - x_n)^2 + b_q(y_q - y_n)^2 + c_q(z_q - z_n)^2}, \quad (14)$$

where β_q, a_q, b_q and c_q are coefficients depending on the q -th neighbor ($Q = 6$ or 26). On the boundaries, where some of the neighbors are not present, the Neumann boundary condition $\partial \mathbf{M}_n / \partial \mathbf{n} = 0$ (with \mathbf{n} indicating the unit normal vector) was imposed by introducing fictitious points outside the domain, where \mathbf{M}_q is replaced with \mathbf{M}_n [42, 43].

The anisotropy field is evaluated in each cell as

$$\mathbf{H}_{an}(\mathbf{r}_n) = -\frac{1}{\mu_0} \frac{\partial \phi_{an}(\mathbf{M}_n)}{\partial \mathbf{M}_n}, \quad (15)$$

where ϕ_{an} is the magnetocrystalline anisotropy energy density function. In the case of uniaxial anisotropy with easy axis having unit vector \mathbf{u} , ϕ_{an} is expressed as

$$\phi_{an}(\mathbf{M}_n) = \frac{k_{an}}{M_S^2} \left[1 - (\mathbf{M}_n \cdot \mathbf{u})^2 \right], \quad (16)$$

where k_{an} is the anisotropy constant.

3. Validation of the numerical code

In this Section we provide details about the procedure we followed to validate the developed micromagnetic solver. To investigate the code performance in the evaluation of magnetization dynamics, we initially focused on Standard Problem #4, proposed by the *Micromagnetic Modeling Activity Group* (μ MAG) at NIST [33]. Then, we tested the code in the calculation of the static hysteresis loop of a 3D sample, comparing the results obtained for a 100 nm diameter permalloy sphere with the solution calculated with MuMax3 [34].

3.1 μ MAG Standard Problem #4

We referred to Standard Problem #4 [33] to test the ability of the code to reproduce the magnetization reversal process consequent to the application of a constant uniform field, almost antiparallel to the initial direction of the magnetization. For this test, we considered a thin film of permalloy with the following specifications: sample size ($500 \times 125 \times 3 \text{ nm}^3$); material properties ($M_S = 800 \text{ kA/m}$, $k_{\text{ex}} = 13 \text{ pJ/m}$, $k_{\text{an}} = 0 \text{ J/m}^3$, $\alpha = 0.02$).

Starting from an initial s-state at zero field, we monitored the magnetization evolution towards equilibrium after the application in the thin film plane (xy) of two different external fields:

- **Field 1:** $\mu_0 H_x = -24.6 \text{ mT}$, $\mu_0 H_y = 4.3 \text{ mT}$, which is a field approximately equal to 25 mT , directed 170° counterclockwise from the positive x -axis;
- **Field 2:** $\mu_0 H_x = -35.5 \text{ mT}$, $\mu_0 H_y = -6.3 \text{ mT}$, which is a field approximately equal to 36 mT , directed 190° counterclockwise from the positive x -axis.

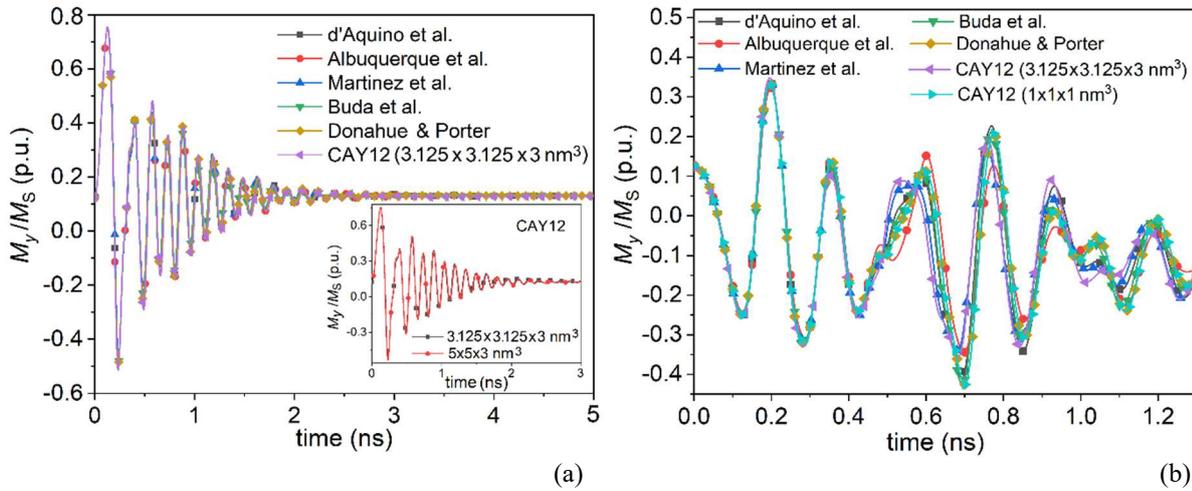


Figure 1. Comparison of the solutions calculated with our code (time-integration scheme CAY12 and accuracy requirement $\epsilon_* = 1 \times 10^{-5}$) and the solutions available on [33] for μ MAG Standard Problem #4. Time evolution of the normalized spatially averaged y -component of the magnetization for (a) Field 1 and (b) Field 2. Regarding our code, for Field 1 the solution reported in the main graph in Fig. 1(a) was computed with a mesh size equal to $3.125 \times 3.125 \times 3 \text{ nm}^3$; in the inset this is compared to a solution obtained with a coarser mesh ($5 \times 5 \times 3 \text{ nm}^3$). For Field 2 we also included the solution obtained with a much finer cubic mesh (size of 1 nm) for a direct comparison with the result provided by Donahue and Porter.

The time evolution of the normalized spatially averaged y -component of the magnetization is reported in Fig. 1(a), for the case of Field 1, and in Fig. 1(b), for the case of Field 2. Regarding time-integration, for both cases we used the time-adaptive integration scheme CAY12 by setting the accuracy requirement ϵ_* at 1×10^{-5} and parameter b at 0.8 . Regarding spatial integration, we tested different mesh sizes, also considering the role of discretization along the film thickness. Figure 1 compares our solutions with some of the results available on the μ MAG site [33], which were obtained with mesh sizes in the order of 3 nm , except for the solution from Donahue and Porter, computed with a much finer cubic mesh, with size of 1 nm .

For the case of Field 1, we found a very good agreement among all the solutions, as illustrated by the main graph in Fig. 1(a), where the curve calculated with our code corresponds to a mesh size equal to $3.125 \times 3.125 \times 3 \text{ nm}^3$ (no discretization along the z -axis was introduced). To test the convergence versus cell size, we previously performed the simulation also with a coarser mesh ($5 \times 5 \times 3 \text{ nm}^3$), which led to a magnetization time evolution practically coincident with the finer mesh one, as shown by the inset in Fig. 1(a).

For Field 2, the magnetization curves calculated with the different codes are very similar up to $\sim 0.4 \text{ ns}$ from the beginning of the transient, then appreciable differences appear, as illustrated by Fig. 1(b). For this specific case, which displays a strong sensitivity to spatial discretization, the mesh size of $3.125 \times 3.125 \times 3 \text{ nm}^3$ is not

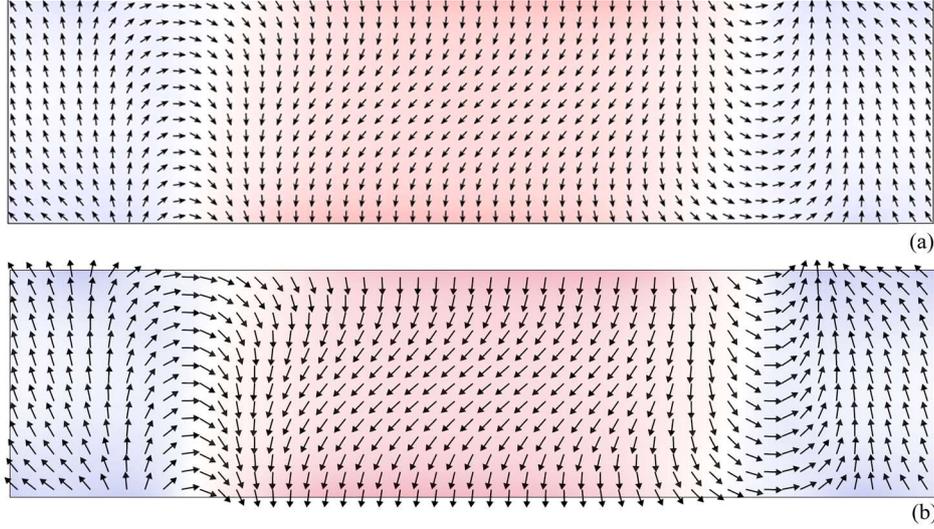


Figure 2. Comparison between the magnetization reversal processes under Field 2, calculated with (a) OOMMF and (b) our code (time-integration scheme CAY12), setting mesh size at 1 nm. The magnetization configuration was taken when the spatially averaged x -component of the magnetization first crosses zero. The arrows represent the magnetization direction in the xy -plane. The background color indicates the z -component of the magnetization, with blue directed away from the viewer and red directed towards the viewer.

sufficient to guarantee the solution convergence and discretization along the z -axis is required. As a proof, we also report in Fig. 1(b) the result obtained with our code with a 1 nm sized cubic mesh; this is in perfect agreement, for the whole duration of the simulation, with the one obtained by Donahue and Porter with the same mesh [33].

As an example of spatial reconstruction of the solution, Fig. 2 shows the distribution of the magnetization vector, obtained when M_x crosses zero for the first time, under the action of Field 2. The map calculated with our code (time-integration scheme CAY12), reported in Fig. 2(b), is in good agreement with the one computed with OOMMF [44], shown in Fig. 2(a), as well as with the other results submitted on [33]. The two maps were calculated by setting mesh size at 1 nm.

3.2 Validation of hysteresis loop computation

To test the accuracy in static hysteresis loop computation, we compared the results of our code to the ones obtained with MuMax3 [34]. The loops were calculated by applying the external field in discrete steps and thus letting the magnetization evolve until equilibrium is reached. For each field interval, at each time instant, we locally monitor the amplitude of the torque exerted by the effective field on the magnetization. We assume that the equilibrium state is reached, when the maximum normalized value of the torque is lower than a fixed threshold τ_* (typically in the order of 10^{-7}), i.e.

$$\max_n |\mathbf{M}^n \times \mathbf{H}_{\text{eff}}^n| / M_S^2 \leq \tau_*, \quad n = 1, \dots, N \quad (17)$$

where N is the total number of elementary cells in the spatial discretization.

Figure 3 reports the static hysteresis loops computed for a permalloy sphere of diameter equal to 100 nm. The following simulation parameters were considered: mesh size ($5 \times 5 \times 5 \text{ nm}^3$); material properties ($M_S = 860 \text{ kA/m}$, $k_{\text{ex}} = 13 \text{ pJ/m}$, $k_{\text{an}} = 0 \text{ J/m}^3$, $\alpha = 0.1$). The external field, applied along x -axis, was varied between -300 kA/m and 300 kA/m in steps of 1 kA/m . Our solution was obtained with the time-adaptive integration scheme CAY12, fixing the accuracy requirement ε_* to 1×10^{-6} , parameter σ to 9γ and the torque threshold τ_* for the equilibrium state evaluation to 1×10^{-7} . With MuMax3, the equilibrium condition was derived by minimizing the total free energy via the conjugate gradient method [34]. As demonstrated by Fig. 3, the two solvers are in

good agreement both qualitatively, calculating similarly shaped hysteresis loops, and quantitatively, predicting approximately the same irreversible jump fields.

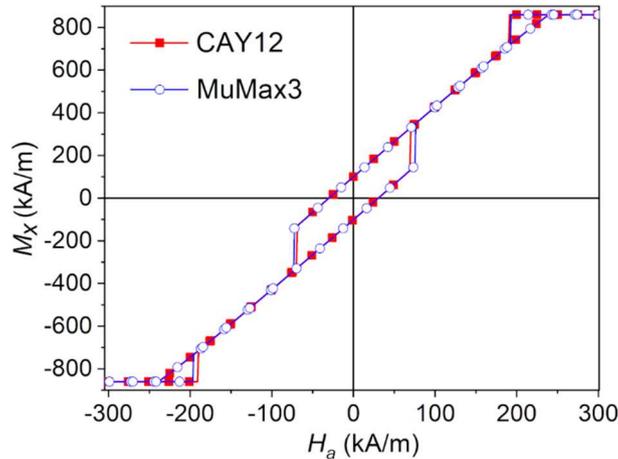


Figure 3. Comparison of the static hysteresis loops of a 100 nm diameter permalloy sphere, calculated with CAY12 (red line with square marks) and MuMax3 (blue line with circle marks).

4 Analysis of time-integration scheme performance

4.1 Comparison of first-order, second-order and adaptive time-integration schemes

Here we compare the computational performance of the three time-integration schemes CAY1, CAY2 and CAY12, described in Sub-section 2.1. For CAY12 the simulations were performed by varying the accuracy requirement ϵ_* and fixing parameters b and σ to 0.8 and 0, respectively. To test numerical stability, we consider μ MAG Standard Problem #4 [33] and, in particular, the case with Field 1, solved with mesh size equal to $3.125 \times 3.125 \times 3 \text{ nm}^3$ (for the parameter specifications refer to Sub-section 3.1).

In Fig. 4(a) we can see that the maximum time-step Δt able to guarantee the convergence and stability of the solution using CAY2 is more than one order of magnitude larger than the one required for CAY1 (0.25 ps against 0.0125 ps). Moreover, it is evident that for certain time windows CAY12 is able to select a time-step higher than the maximum Δt that preserves stability for the second-order scheme (CAY2). The lowering of the accuracy requirement (i.e. the increase in ϵ_*) has a very weak effect on Δt in the last part of the simulation, where the precessional motion reduces, while it leads to a rise in Δt at the beginning of the transient, allowing the algorithm to be more computationally efficient. For all the considered values of ϵ_* , a sudden decrease in Δt occurs in correspondence of the largest oscillation of the magnetization.

Figure 4(b) reports the time evolution of the exchange energy density, as an indicator of loss of stability [11, 45]. For the present problem the exchange energy is expected to decrease after the switching of the magnetization and stabilize around a very low value. The time-adaptive algorithm CAY12 is able to correctly choose the time-step and, even for very low accuracy, it does not lead to any non-physical increase in the exchange energy. On the contrary, with CAY1 the energy suddenly rises around 1 ns when using a time-step of 0.025 ps, leading to completely altered results; in order to guarantee stability a very small time-step is required, i.e. 0.0125 ps. CAY2 is stable when $\Delta t = 0.25$ ps, but becomes clearly unstable when Δt is set at 0.3 ps, as shown again by the abrupt increase in the exchange energy at about 1 ns. The time-step of 0.275 ps, around which CAY12 stabilizes after the initial part of the transient [Fig. 4(a)], is inadequate to guarantee the stability of CAY2, as demonstrated by the rise in the exchange energy at the end of the simulation, with the consequent impossibility of reaching the equilibrium condition [Fig. 4(b)]. This wrong behavior cannot be immediately detected in the time evolution of the magnetization, which at 5 ns has values very close to the correct ones, but can be noticed by looking at the spatial distribution of the exchange field, which shows a non-physical checkerboard pattern (Fig. 5).

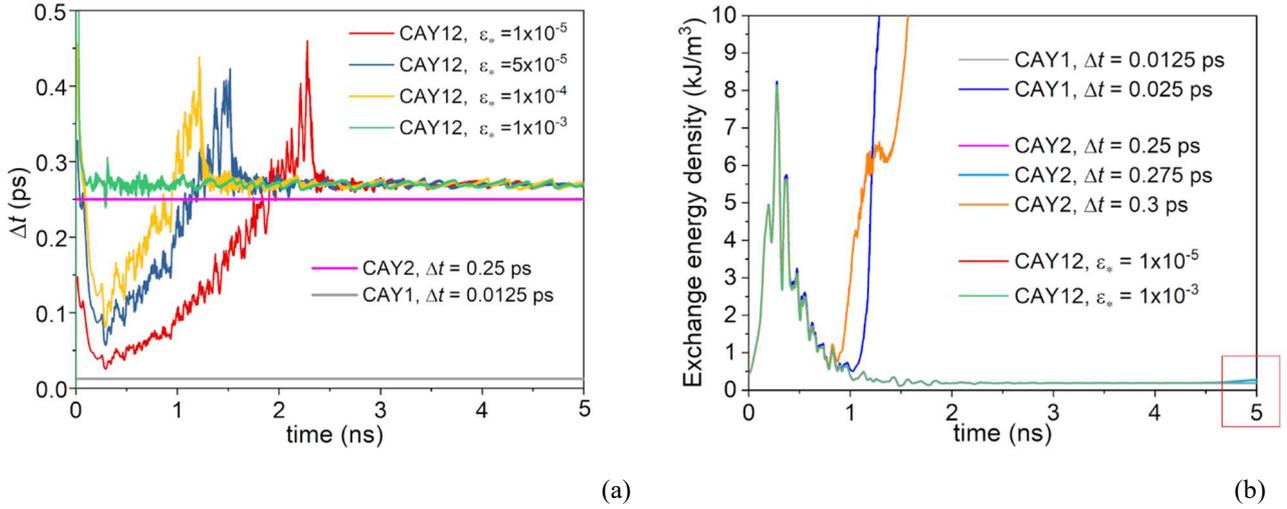


Figure 4. (a) Evolution, during the simulation of μ MAG Standard Problem #4 (Field 1), of the time-step used by the time-adaptive integration scheme CAY12 for different levels of the required accuracy ϵ_* , compared to the maximum time-step achievable with CAY1 and CAY2. (b) Time evolution of the exchange energy density as an indicator of the stability of the adopted time-integration scheme. The red line square at the end of the simulation marks the increase in the energy due to the instability arisen when using CAY2 and $\Delta t = 0.275$ ps.

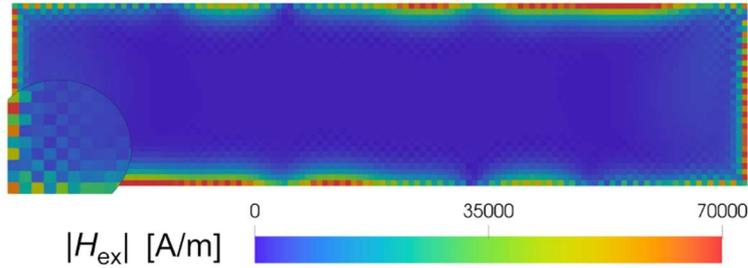


Figure 5. Spatial distribution of the amplitude of the exchange field calculated at 3 ns with CAY2, setting the time-step at 0.275 ps (μ MAG Standard Problem #4, case Field 1). The inadequate choice of the time-step leads to numerical instability in the form of a non-physical checkerboard pattern, magnified in the bottom left corner.

The first-order scheme CAY1 shows its strong limits, requiring 400000 time-steps for a 5 ns long simulation, while the second-order scheme CAY2 needs 20000 time steps to conclude the simulation with the best possible choice of Δt . The time-adaptive scheme CAY12 performs well in comparison to CAY2, requiring ~ 18500 time-steps for $\epsilon_* = 1 \times 10^{-3}$ and ~ 20000 for $\epsilon_* = 1 \times 10^{-4}$. The increase in accuracy requirement leads to a moderate raise in the number of time-steps, i.e. for $\epsilon_* = 5 \times 10^{-5}$ roughly 20% of additional time-steps are needed with respect to CAY2, however for $\epsilon_* = 1 \times 10^{-5}$ this percentage increases up to 75%. Even if for this specific case we cannot see a significant improvement in performance when comparing the time-adaptive second-order scheme with the Heun method (CAY2), it is important to point out that the time-adaptive algorithm enables us to avoid a prior trial-and-error process to find the optimal Δt , guaranteeing at the same time a good level of accuracy. Without this search for the maximum Δt able to preserve solution stability, recommended when employing CAY2, one would not obtain a priori the same computational efficiency of its time-adaptive counterpart. Another drawback is the insidious risk of using an inadequate time-step and thus performing a simulation that can lose stability, preventing the reaching of equilibrium state, as depicted in Fig. 4(b) for CAY2 and $\Delta t = 0.275$ ps.

4.2 Comparison with classical Runge-Kutta time-integration schemes

In this Sub-section we compare the geometrical integrator within our micromagnetic code to the standard explicit time-adaptive Runge-Kutta (RK) methods implemented in MuMax3 [34]. In particular, we analyze

the evolution of the time-steps selected during the simulation and how they differ when algorithms of different order are used, even if the same level of accuracy is required. The numerical test was performed on a permalloy square sample with size of $50 \times 50 \times 20 \text{ nm}^3$, using a mesh of 5 nm along all the directions and considering the same material properties of μMAG Standard Problem #4. The sample, which is saturated along the positive x -axis, is exposed to a uniform constant field applied in the opposite direction, with an amplitude of 50 kA/m. The accuracy requirement ε_* was fixed to 1×10^{-5} for both codes and we set parameter σ at zero for CAY12.

As illustrated by Fig. 6, which reports the time evolution of the maximum time-step able to guarantee solution convergence and stability, CAY12 can reach better performance than the classic RK12 method. In particular, for a precautionary factor $b = 0.8$, CAY12 selects Δt two times larger than RK12 for similar accuracy requirement, with a resulting computational efficiency between RK12 and RK23 time-adaptive schemes.

It is important to state that different values of factor b were tested to ensure a correct choice of the time-step, without excessively underestimating it. For this test case, we observed that, without the precautionary term ($b = 1$), the number of time-steps required to perform the simulation (taking into account the discarded time-steps that do not meet the desired accuracy) is 20% higher than when $b = 0.8$. Moreover, we found that, when $b = 0.8$, no re-evaluation is necessary. The lowering of b below 0.8 has no advantage, slowing down significantly the calculations, as also shown by Fig. 6 for $b = 0.5$. When $b = 0.9$ only a very small number of time-steps has to be discarded, providing better computational performance than $b = 0.8$, nevertheless it is advisable to opt for the safer value (0.8) that compromises computational performance in a limited way.

Higher-order algorithms can have better convergence and stability properties at the expense of a larger number of intermediate steps, and the consequent increase in the overall computational burden. At each time-step, the number of operations required by CAY12 is greater than the number of operations needed by RK12 due to the Cayley transformation application, but the complexity of the solver part devoted to the only magnetization update scales as $\mathcal{O}(N)$ for both algorithms. However, for each time-step, multiple calculations of the effective field have to be performed, depending on the order of the time-integration scheme and on the accuracy requirement. If an FFT discrete convolution method is adopted for the calculation of the magnetostatic field, which is the most computationally expensive term, the complexity of the effective field evaluation is $\mathcal{O}(N \log N)$. This operation has a strong impact on the total computational cost of the simulation, thus in terms of overall computational efficiency it could not be convenient to opt for higher-order time-integration schemes, like RK45, even if it enables us to use larger time-steps. As an example, at the end of the transient RK45 allows us to use time-steps at least twice larger than CAY12, but at each time-step the effective field has to be calculated five times, resulting in a greater computational cost.

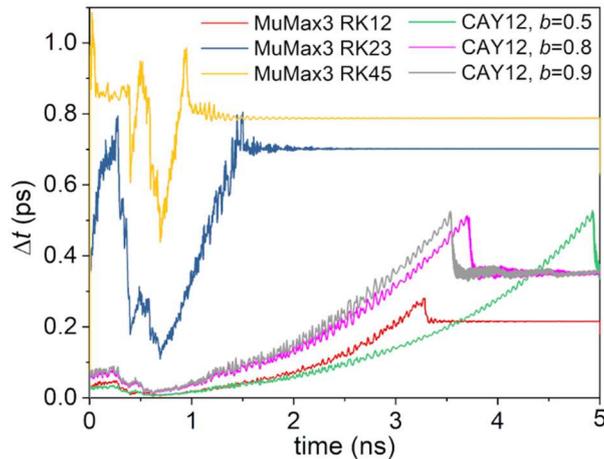


Figure 6. Comparison between time-adaptive integration scheme CAY12 for different values of precautionary factor b and diverse order time-adaptive RK schemes implemented in MuMax3. The graph shows the evolution of the time-step used by the different algorithms for a fixed accuracy ε_* of 1×10^{-5} , when calculating magnetization switching processes in a permalloy sample with size of $50 \times 50 \times 20 \text{ nm}^3$.

4.3 Influence of damping effects on equilibrium calculation

In many applications one cannot be interested in the system dynamics, but in the final equilibrium configuration, for example when calculating the static hysteresis loop of a sample. To speed-up the computation of equilibrium states, one of the most commonly used strategies is to artificially increase the damping parameter α . This can lead to an improvement in computational efficiency due to both greater stability and possibility of accelerating the reaching of the transient end.

One of the features of our micromagnetic code is the possibility of solving a generalized form of the LLG equation, which contains a term parallel to the magnetization vector, $\sigma\mathbf{M}$, as described in Sub-section 2.1. From the mathematical and physical points of view, the contribution of this term to the LLG equation is zero, but when applying the time integration, it becomes responsible for the generation of artificial viscous effects, whose magnitude is proportional to σ and Δt [46]. The term $\sigma\mathbf{M}$ provides a second mean that can be exploited to enhance damping phenomena, leading to a reduction in the transient duration and to a possible acceleration of the convergence to the equilibrium configuration.

In this Section we show the effects of the increase in α and of the additive term introduced in the generator of the LLG equation, the potential drawbacks of the inclusion of artificial damping and how a proper tuning of parameters α and σ can improve the convergence to equilibrium. The tests were performed on a 100 nm diameter permalloy sphere, with $M_S = 860$ kA/m, $k_{ex} = 13$ pJ/m, $k_{an} = 0$ J/m³ and variable α , discretized with a mesh having size of $5 \times 5 \times 5$ nm³. In the considered problem, the magnetization was relaxed from a quasi-saturated state along x -axis to remanence, in the absence of external field. The different analyzed cases were compared focusing on the time evolution of the x -component of the magnetization. Moreover, we discuss the modification of the time-adaptive integration scheme CAY12, required to exploit the benefits of artificial damping phenomena in reaching the equilibrium states.

In Fig. 7(a) we show the role of α setting σ at zero and the threshold τ_* for the convergence to equilibrium at 1×10^{-8} ; the time-integration scheme CAY2 was used. We can see that the increase in α from the material physical value of 0.02 up to 0.1 can reduce considerably the transient duration and speed-up the convergence to equilibrium, due to the reduction in precessional phenomena. Moreover, as discussed in [46], it leads to an increase in the time-step guaranteeing stability. However, over a certain value, the further increase in α causes the simulation to slow down, as demonstrated by the variation in the time evolution observed when $\alpha \geq 0.2$. In particular, the enhancement of damping effects causes the system to remain frozen in an intermediate configuration for a larger time interval, thus decelerating the reaching of the real equilibrium. As an additional drawback, a tighter criterion has to be applied to verify the convergence to equilibrium (i.e. by significantly reducing τ_*), otherwise this condition could be satisfied when the magnetization is in its intermediate configuration, where it remains blocked for a long time. As an example, an erroneous evaluation of the equilibrium state is made when $\alpha = 1$, since for this case the threshold value of 1×10^{-8} is not sufficient to guarantee the correct convergence, stopping the simulation at about 1.3 ns. Thus, the increase in α can be a valid strategy for improving computational efficiency only for limited ranges of variation, otherwise it could be detrimental [46].

In Fig. 7(b) we analyze the artificial damping effects due to term $\sigma\mathbf{M}$, by varying parameter σ and contextually the threshold τ_* for the convergence to equilibrium; the time-integration scheme CAY2 was used and α was fixed to 0.1. The role of $\sigma\mathbf{M}$ becomes noticeable only when σ has the same order of magnitude of γ and Δt is in the order of 1 ps, producing an enhancement of damping phenomena [46]. In this case, there is also an increase in the maximum time-step that guarantees stability, varying from 0.1 ps, when $\sigma = 0$, to 1 ps, when $\sigma = 4.5\gamma$. The total number of time-steps necessary to reach the equilibrium condition for $\sigma = 0$, $\Delta t = 0.1$ ps and $\tau_* = 1 \times 10^{-7}$ is $\sim 3 \times 10^6$, and reduces to $\sim 1 \times 10^5$ when using $\sigma = 9\gamma$, $\Delta t = 1$ ps and a lower torque threshold, i.e. $\tau_* = 1 \times 10^{-8}$, required to calculate the correct equilibrium state. For the latter, it is important to notice that equilibrium is reached 30 times faster. However, a further increase in σ does not provide any advantages, since the system tends to remain much longer in the local minima of the free energy, needing a tighter requirement on the torque threshold to guarantee the reaching of the correct equilibrium configuration. As an example,

when $\sigma = 22.5\gamma$ and $\tau_* = 1 \times 10^{-9}$, the simulation stops before. If $\tau_* = 1 \times 10^{-10}$, the correct equilibrium is reached, but the number of time-steps doubles with respect to the case with $\sigma = 9\gamma$.

In the following, we investigate the role of term $\sigma \mathbf{M}$, employing time-integration scheme CAY12 with $\epsilon_* = 1 \times 10^{-5}$ and $b = 0.8$. As shown in Fig. 7(c), the selection of a time-constant value for parameter σ does not lead to an improvement in computational efficiency. The reason is that the artificial damping effects introduced by the Cayley transform-based time-integration are proportional to both σ and Δt . During the initial magnetization precessional motion, the time-adaptive algorithm tends to reduce the time-step to accurately reproduce the oscillations, thus greatly diminishing the numerical damping introduced by term $\sigma \mathbf{M}$. By comparing Fig. 7(b) and Fig. 7(c), it is clear that for the same values of σ the use of CAY12 limits the damping effects, with a consequent more oscillatory behavior. To obtain appreciable damping of the initial transient, much larger values of σ are required, e.g. $\sigma = 45\gamma$, but the drawback is that when the time-step increases, after the initial magnetization oscillation, the high value of σ causes the system to remain frozen in the local energy minimum for a very long time.

To exploit the numerical advantages of CAY12 and term $\sigma \mathbf{M}$, we implemented a modified time-adaptive integration scheme, where σ is a function of time, inversely proportional to the time-variable Δt . In this case, instead of setting σ at a specific value, one can keep fixed the product $\psi = \sigma(t) \times \Delta t$ for the entire duration of the simulation. For the results shown in Fig. 7(d) the value of 1×10^{-6} m/A assigned to the product ψ was chosen starting from the knowledge that when using CAY2 the stability is preserved if $\sigma = 4.5\gamma$ and $\Delta t = 1$ ps. We can see that, with such approach, the solution strongly improves with respect to the one calculated with the non-modified time-adaptive scheme. Moreover, it behaves comparably to the corresponding case where both σ and Δt have fixed values (CAY2), preserving the advantages of the time-adaptive algorithm as well as the quicker convergence to the equilibrium obtainable with term $\sigma \mathbf{M}$. Thanks to this implementation numerical damping effects are enhanced and become noticeable also when small time-steps are used, due to the increase in σ . On the contrary, they reduce when very large time-steps are selected, thus avoiding the detrimental blocking of the magnetization. Another advantage is the possibility of increasing the torque threshold τ_* for the equilibrium condition. As a consequence of all these aspects, in a time-window of 6 ns, the number of time-steps reduces to one sixth in comparison to the use of CAY2.

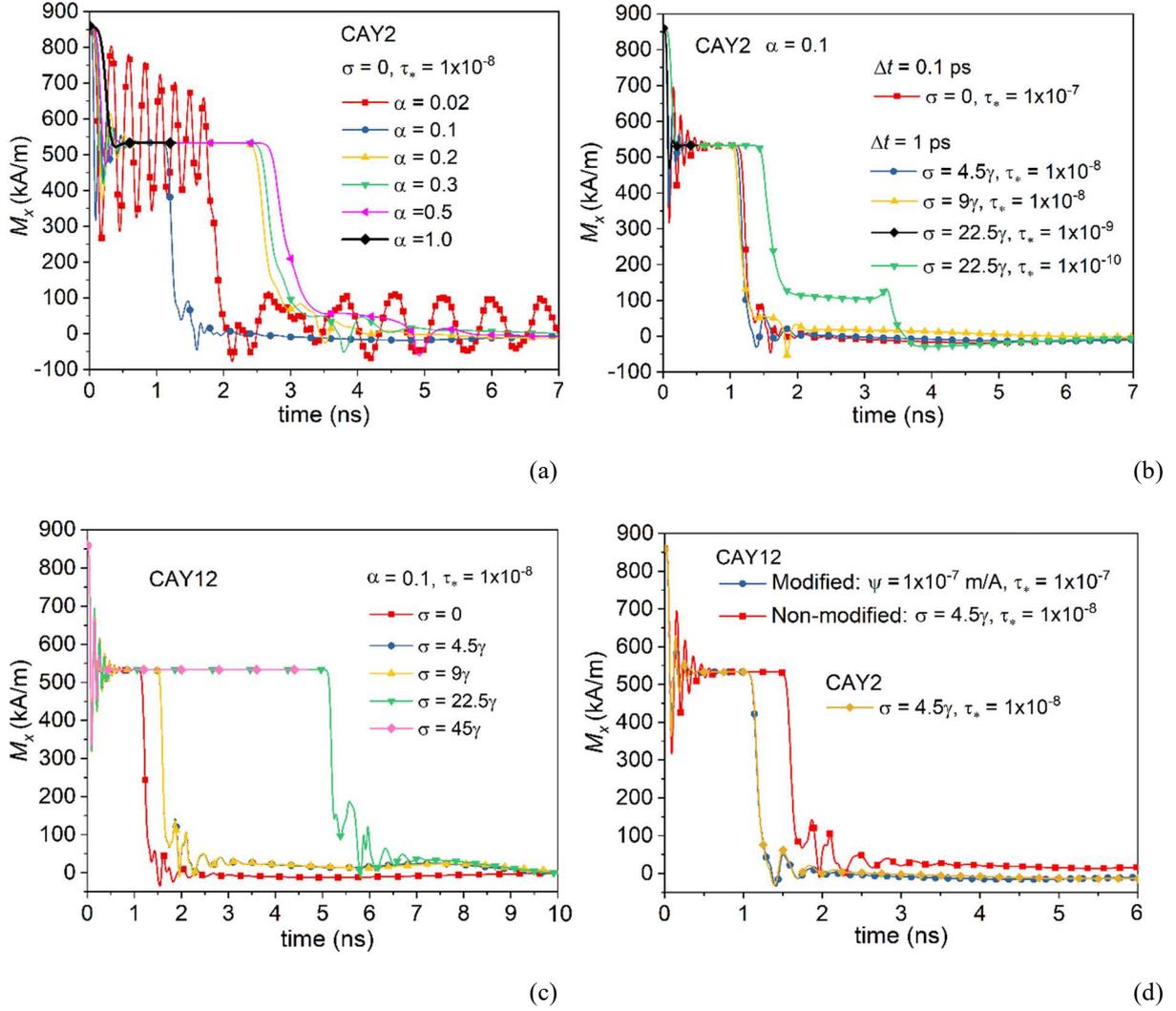


Figure 7. Analysis of the role of intrinsic and artificial damping phenomena, for the case of magnetization relaxation in a 100 nm diameter permalloy sphere. The graphs report the time evolution of the spatially averaged magnetization component along x -axis (initial quasi-saturation direction). (a) Role of damping coefficient α , when using CAY2 and fixing σ to zero. (b) Role of parameter σ , when using CAY2 and fixing α to 0.1. (c) Role of σ , when using CAY12 and fixing α to 0.1 (d) Role of time-variable σ and Δt when using modified CAY12 (the product of σ and Δt , ψ , is kept constant during the time evolution). The simulation is compared to the corresponding ones, obtained with CAY2 and non-modified CAY12.

5. Conclusions

We developed a GPU-parallelized 3D micromagnetic solver able to efficiently simulate magnetization dynamics in 3D magnetic samples discretized with a structured mesh. Thanks to the implementation of time-integration schemes based on the Cayley transform, several advantages can be obtained, such as the preservation of the magnetization amplitude and the possibility of increasing the maximum time-step that guarantees stability. Moreover, the time-adaptive version is able to select time-steps that are two times larger than the ones used by explicit algorithms, like classic RK schemes, of the same order.

The use of an explicit method enables us to avoid iteration processes, keeping the number of evaluations of the effective field to a minimum. In addition, the good stability guaranteed by the geometric integrator permits the use of a second-order algorithm, further reducing the number of effective field calculations. Moreover, thanks to the treatment of a generalized version of the LLG equation, which contains a term parallel to the magnetization, it is possible to speed-up the computation of equilibrium states and of static hysteresis loops, by simply exploiting the artificial damping phenomena arising with the numerical integration. This procedure

enables us to accelerate the convergence to equilibrium as well as to increase the size of the time-step guaranteeing stability, thus leading to a further improvement of computational efficiency.

Despite the possible introduction of approximations in the reconstruction of complex shaped-samples, the use of a structured hexahedral mesh permits the implementation of computationally efficient algorithms for the accurate evaluation of the exchange field (26-neighbor finite difference method) and of the magnetostatic field (GPU-accelerated FFT-based algorithm). Further development of the code will regard the inclusion of the thermal field in the expression of the effective field, and of specific interface models for the exchange field computation in the presence of different materials.

Acknowledgments

The work here presented was developed in the framework of the 18HLT06 RaCHy Project that received funding from the EMPIR Program, co-financed by the Participating States and from the European Union's Horizon 2020 Research and Innovation Program.

Declaration of Competing Interest

There are no conflicts of interest.

References

- [1] M. d'Aquino, C. Serpico, G. Miano, Geometrical integration of Landau-Lifshitz-Gilbert equation based on the mid-point rule, *J. Comput. Phys.* 209 (2005), pp. 730–753. doi:10.1016/j.jcp.2005.04.001.
- [2] B. Van de Wiele, F. Olyslager, L. Dupré, Fast Semianalytical Time Integration Schemes for the Landau–Lifshitz Equation, *IEEE Trans. Magn.* 43 (2007), pp. 2917–2919. doi:10.1109/TMAG.2007.892534.
- [3] D. G. Porter, M. J. Donahue, Precession axis modification to a semianalytical Landau–Lifshitz solution technique, *J. Appl. Phys.* 103 (2008), 07D920. doi:10.1063/1.2838461.
- [4] A. Vansteenkiste, B. Van De Wiele, MUMAX: A new high-performance micromagnetic simulation tool, *J. Magn. Magn. Mater.* 323 (2011), pp. 2585–2591. doi:10.1016/j.jmmm.2011.05.037.
- [5] A. Romeo, G. Finocchio, M. Carpentieri, L. Torres, G. Consolo, B. Azzerboni, A numerical solution of the magnetization reversal modeling in a permalloy thin film using fifth order Runge-Kutta method with adaptive step size control, *Phys. B Condens. Matter.* 403 (2008), pp. 464–468. doi:10.1016/j.physb.2007.08.076.
- [6] X. Wang, C. J. García-Cervera, W. E, A Gauss–Seidel Projection Method for Micromagnetics Simulations, *J. Comput. Phys.* 171 (2001), pp. 357–372. doi:10.1006/jcph.2001.6793.
- [7] C. Serpico, I. D. Mayergoyz, G. Bertotti, Numerical technique for integration of the Landau–Lifshitz equation 6991 (2001), pp. 129–132. doi:10.1063/1.1358818.
- [8] L. Exl, N. J. Mauser, T. Schrefl, D. Suess, The extrapolated explicit midpoint scheme for variable order and step size controlled integration of the Landau–Lifshitz–Gilbert equation, *J. Comput. Phys.* 346 (2017), pp. 14–24. doi:10.1016/j.jcp.2017.06.005.
- [9] D. Suess, V. Tsiantos, T. Schrefl, J. Fidler, W. Scholz, H. Forster, R. Dittrich, J. J. Miles, Time resolved micromagnetics using a preconditioned time integration method, *J. Magn. Magn. Mater.* 248 (2002), pp. 298–311. doi:10.1016/S0304-8853(02)00341-4.
- [10] R. Chang, S. Li, M. V. Lubarda, B. Livshitz, V. Lomakin, FastMag: Fast micromagnetic simulator for complex magnetic structures, *J. Appl. Phys.* 109 (2011), 07D358. doi:10.1063/1.3563081.
- [11] O. Bottauscio, M. Chiampi, A. Manzin, A Finite Element Procedure for Dynamic Micromagnetic Computations, *IEEE Trans. Magn.* 44(11) (2008), pp. 3149–3152. doi:10.1109/TMAG.2008.2001666.
- [12] S. Bartels, Constraint Preserving, Inexact Solution of Implicit Discretizations of Landau–Lifshitz–Gilbert Equations and Consequences for Convergence, *PAMM Proc. Appl. Math. Mech.* 6 (2006), pp. 19–22. doi:10.1002/pamm.20061000.
- [13] P. E. Crouch, R. Grossman, Numerical integration of ordinary differential equations on manifolds, *J. Nonlinear Sci.* 3 (1993), pp. 1–33. doi:10.1007/BF02429858.

- [14] D. Lewis, J. C. Simo, Conserving algorithms for the dynamics of Hamiltonian systems on Lie groups, *J. Nonlinear Sci.* 4 (1994), pp. 253–299. doi:10.1007/BF02430634.
- [15] H. Munthe-Kaas, Runge-Kutta methods on Lie groups, *BIT Numer. Math.* 38 (1998), pp. 92–111. doi:10.1007/BF02510919.
- [16] E. Celledoni, H. Marthinsen, B. Owren, An introduction to Lie group integrators – basics, new developments and applications, *J. Comput. Phys.* 257 (2014), pp. 1040–1061. doi:10.1016/j.jcp.2012.12.031.
- [17] P. S. Krishnaprasad, X. Tan, Cayley transforms in micromagnetics, *Physica B: Condensed Matter* 306 (2001), pp. 195–199. doi:10.1016/S0921-4526(01)01003-1.
- [18] D. Lewis, N. Nigam, Geometric integration on spheres and some interesting applications, *J. Comput. Appl. Math.* 151 (2003), pp. 141–170. doi:10.1016/S0377-0427(02)00743-4.
- [19] O. Bottauscio, A. Manzini, Efficiency of the Geometric Integration of Landau–Lifshitz–Gilbert Equation Based on Cayley Transform, *IEEE Trans. Magn.* 47 (2011), pp. 1154–1157. doi:10.1109/TMAG.2010.2095831.
- [20] A. Manzini, R. Ferrero, A 2.5D micromagnetic solver for randomly distributed magnetic thin objects, *J. Magn. Magn. Mater.* 492 (2019), 165649. doi:10.1016/j.jmmm.2019.165649.
- [21] A. Vansteenkiste, J. Leliaert, M. Dvornik, M. Helsen, F. Garcia-Sanchez, B. Van Waeyenberge, The design and verification of MuMax3, *AIP Adv.* 4 (2014), 107133. doi:10.1063/1.4899186.
- [22] J. R. Cash, A. H. Karp, A Variable Order Runge-Kutta Method for Initial Value Problems with Rapidly Varying Right-Hand Sides, *ACM Trans. Math. Softw.* 16 (1990), pp. 201–222. doi:10.1145/79505.79507.
- [23] J. R. Dormand, P. J. Prince, A reconsideration of some embedded Runge-Kutta formulae, *J. Comput. Appl. Math.* 15 (1986), pp. 203–211. doi:10.1016/0377-0427(86)90027-0.
- [24] R. D. Ryne, On FFT-based convolutions and correlations, with application to solving Poisson’s equation in an open rectangular pipe, (2011). <http://arxiv.org/abs/1111.4971>.
- [25] L. Exl, T. Schrefl, Non-uniform FFT for the finite element computation of the micromagnetic scalar potential, *J. Comput. Phys.* 270 (2014), pp. 490–505. doi:10.1016/j.jcp.2014.04.013.
- [26] B. Van de Wiele, F. Olyslager, L. Dupré, D. De Zutter, On the accuracy of FFT based magnetostatic field evaluation schemes in micromagnetic hysteresis modeling, *J. Magn. Magn. Mater.* 322 (2010), pp. 469–476. doi:10.1016/j.jmmm.2009.09.077.
- [27] C. J. García-Cervera, Z. Gimbutas, E. Weinan, Accurate numerical methods for micromagnetics simulations with general geometries, *J. Comput. Phys.* 184 (2003), pp. 37–52. doi:10.1016/S0021-9991(02)00014-1.
- [28] A. Kakay, E. Westphal, R. Hertel, Speedup of FEM Micromagnetic Simulations With Graphical Processing Units, *IEEE Trans. Magn.* 46 (2010), pp. 2303–2306. doi:10.1109/TMAG.2010.2048016.
- [29] R. Chang, S. Li, M. V. Lubarda, B. Livshitz, V. Lomakin, FastMag: Fast micromagnetic simulator for complex magnetic structures, *Journal of Applied Physics* 109, (2011) 07D358. doi.org/10.1063/1.3563081.
- [30] L. Lopez-Diaz, D. Aurelio, L. Torres, E. Martinez, M. A. Hernandez-Lopez, J. Gomez, O. Alejos, M. Carpentieri, G. Finocchio, G. Consolo, Micromagnetic simulations using Graphics Processing Units, *Journal of Physics D: Applied Physics* 45 (2012), 323001. doi.org/10.1088/0022-3727/45/32/323001.
- [31] O. Bottauscio, A. Manzini, Parallelized micromagnetic solver for the efficient simulation of large patterned magnetic nanostructures, *J. Appl. Phys.* 115 (2014), 17D122. doi:10.1063/1.4862379.
- [32] R. Ferrero, A. Manzini, G. Barrera, F. Celegato, M. Coisson, P. Tiberto, Influence of shape, size and magnetostatic interactions on the hyperthermia properties of permalloy nanostructures, *Sci. Rep.* 9 (2019), 6591. doi:10.1038/s41598-019-43197-4.
- [33] NIST CTCMS, μ MAG Micromagnetic Modeling Activity Group. <https://www.ctcms.nist.gov/~rdm/mumag.org.html>.
- [34] J. Leliaert, M. Dvornik, J. Mulders, J. De Clercq, M. V. Milošević, B. Van Waeyenberge, Fast micromagnetic simulations on GPU - Recent advances made with mumax3, *J. Phys. D.: Appl. Phys.* 51 (2018), 123002. doi:10.1088/1361-6463/aaab1c.
- [35] E. Celledoni, A. Iserles, Approximating the Exponential from a Lie Algebra to a Lie Group, *Math. Comp.* 69

(2000), pp.1457–1480. doi:10.1090/S0025-5718-00-01223-0.

- [36] W. Press, S. Teukolsky, W. Vetterling, B. Flannery, E. Ziegel, B. Flannery, S. Teukolsky, W. Vetterling, *Numerical Recipes: The Art of Scientific Computing*, Cambridge University Press, 1987. doi:10.2307/1269484.
- [37] R. Zhu, Grace: A cross-platform micromagnetic simulator on graphics processing units, *SoftwareX* 3–4 (2015), pp. 27–31. doi:10.1016/j.softx.2015.11.001.
- [38] B. J. W. Cooley, J. W. Tukey, An Algorithm for the Machine Calculation of Complex Fourier Series 19 (1965), pp. 297–301. doi:10.2307/2003354.
- [39] Nvidia, CUFFT library. <https://developer.nvidia.com/%0Acufft>.
- [40] Y. Nakatani, Y. Uesaka, N. Hayashi, Direct solution of the Landau-Lifshitz-Gilbert equation for micromagnetics, *Jpn. J. Appl. Phys.* 28 (1989), pp. 2485–2507. doi:10.1143/JJAP.28.2485.
- [41] R. C. O. Reilly, J. M. Beck, A Family of Large-Stencil Discrete Laplacian Approximations in Three Dimensions, *International Journal for Numerical Methods in Engineering* (2006), pp. 1–16.
- [42] M. J. Donahue, D. G. Porter, Exchange energy formulations for 3D micromagnetics, *Physica B: Condensed Matter* 343 (2004), pp. 177–183. doi:10.1016/j.physb.2003.08.090.
- [43] J. E. Miltat, M. J. Donahue, Numerical Micromagnetics: Finite Difference Methods, *Handb. Magn. Adv. Magn. Mater.* (2007) 1–23. doi:10.1002/9780470022184.hmm202.
- [44] OOMMF User's Guide, Version 1.0, M. J. Donahue and D. G. Porter, Interagency Report NISTIR 6376, National Institute of Standards and Technology, Gaithersburg, MD (Sept 1999). <http://math.nist.gov/oommf>.
- [45] B. Van de Wiele, A. Manzin, L. Dupré, F. Olyslager, O. Bottauscio, M. Chiampi, Comparison of Finite-Difference and Finite-Element Schemes for Magnetization Processes in 3-D Particles, *IEEE Transactions on Magnetics* 45 (2009), pp. 1614-1617. doi:10.1109/TMAG.2009.2012756.
- [46] A. Manzin, O. Bottauscio, Connections between numerical behavior and physical parameters in the micromagnetic computation of static hysteresis loops, *J. Appl. Phys.* 108 (2010), 093917. doi:10.1063/1.3503873.