Efficient adaptive pseudo-symplectic numerical integration techniques for Landau-Lifshitz dynamics

M. d'Aquino, F. Capuano, G. Coppola, C. Serpico, and I. D. Mayergoyz

Articles you may be interested in

The design and verification of MuMax3
AIP Advances **4**, 107133 (2014); 10.1063/1.4899186

A two dimensional analytical model for the study of ferromagnetic resonance responses of single and multilayer films
Journal of Applied Physics **121**, 123906 (2017); 10.1063/1.4978517

Eigenmodes of Néel skyrmions in ultrathin magnetic films
AIP Advances **7**, 055212 (2017); 10.1063/1.4983806

Making the Dzyaloshinskii-Moriya interaction visible
Applied Physics Letters **110**, 242402 (2017); 10.1063/1.4985649

A scanning tunneling microscopy based potentiometry technique and its application to the local sensing of the spin Hall effect
AIP Advances **7**, 125205 (2017); 10.1063/1.4991916

Efficient adaptive pseudo-symplectic numerical integration techniques for Landau-Lifshitz dynamics

M. d'Aquino,[1,a] F. Capuano,[2] G. Coppola,[2] C. Serpico,[3] and I. D. Mayergoyz[4]

[1]*Engineering Department, University of Naples "Parthenope", 80143 Naples, Italy*
[2]*Department of Industrial Engineering, University of Naples Federico II, 80125 Naples, Italy*
[3]*DIETI, University of Naples Federico II, 80125 Naples, Italy*
[4]*ECE Department, University of Maryland, College Park, MD 20742, USA*

Numerical time integration schemes for Landau-Lifshitz magnetization dynamics are considered. Such dynamics preserves the magnetization amplitude and, in the absence of dissipation, also implies the conservation of the free energy. This property is generally lost when time discretization is performed for the numerical solution. In this work, explicit numerical schemes based on Runge-Kutta methods are introduced. The schemes are termed *pseudo-symplectic* in that they are accurate to order p, but preserve magnetization amplitude and free energy to order $q > p$. An effective strategy for adaptive time-stepping control is discussed for schemes of this class. Numerical tests against analytical solutions for the simulation of fast precessional dynamics are performed in order to point out the effectiveness of the proposed methods. © *2017 Author(s). All article content, except where otherwise noted, is licensed under a Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).* https://doi.org/10.1063/1.5007340

Numerical integration of the Landau-Lifshitz (LL) equation is extensively used in micromagnetic simulations. A special and well known feature of magnetization dynamics of Landau-Lifshitz-type is that the amplitude of magnetization is preserved in time.[1–4] In addition, in a broad range of situations connected to technological applications, non-conservative effects in magnetization dynamics (e.g. damping, spin-torque, ac excitations, etc.) can be treated as perturbations of the precessional dynamics occurring at constant energy.[5] Hence, when conservative LL dynamics is considered, two quantities are preserved in time: (i) magnetization amplitude; (ii) free energy. Existing off-the-shelf numerical integration schemes such as, for instance, standard Runge-Kutta (RK) methods, do corrupt, in general, these intrinsic properties of conservative magnetization dynamics. As a consequence, besides usual considerations on truncation errors, a challenging accuracy test for numerical schemes for the integration of LL dynamics is given by checking the preservation properties (i)-(ii) in the discrete dynamics with respect to the time step of the chosen integrator. This is especially crucial when the long-term dynamical behavior of the system has to be simulated, and this is the circumstance where the global error is expected to increase dramatically as time advances. Moreover, in order to exploit the massive scalable parallelism of many-integrated-cores (MIC) architectures, such as graphical processing units (GPU), explicit schemes are expected to perform better since they do not require the solution of a large number of nonlinear coupled equations at each time step. In the past, several explicit schemes which fulfill the property (i) have been developed.[3,4,6–9] On the other hand, as it is well-known in the area of Hamiltonian dynamics,[1,2] the property (ii) can be exactly fulfilled only by using implicit methods, such as the midpoint rule.[7] Nevertheless, the so-called pseudo-symplectic RK schemes[10,11] are able to preserve quadratic invariants approximately up to a given order of accuracy. In this work, a class of pseudo-symplectic schemes for Landau-Lifshitz magnetization dynamics is presented, based on multi-stage RK methods, and its properties are tested on a benchmark case

---

[a]Electronic mail: daquino@uniparthenope.it

of fast precessional magnetization switching.[12] Schemes belonging to the aforementioned class are able to guarantee the preservation of both properties (i)-(ii) with order q greater than the classical order p of the method. In addition, for these methods, simple and effective adaptive time step control can be implemented in order to keep the error within prescribed bounds as time advances, which is also discussed in the sequel. Numerical schemes of this class lead to spatially uncoupled difference equations suitable for massively parallel computations typical of GPUs. Finally, it is worth stressing that, when also non-conservative effects are included (damping, spin-torque, etc.), pseudo-symplectic schemes reproduce the energy balance properties of the LL dynamics with far greater accuracy than usual time integration methods.

In order to have the main focus to time integration issues, we address the case of single-domain particle dynamics, since usual spatial discretization techniques based on finite-difference or finite-element methods do not affect the generality of the presented approach, which can be applied at no additional cost to spatially-inhomogeneous micromagnetic simulations. Conservative magnetization dynamics is governed by the Landau-Lifshitz equation, which in normalized form reads:[5]

$$\frac{d\mathbf{m}}{dt} = -\mathbf{m} \times \mathbf{h}_{\text{eff}} = \mathbf{f}(\mathbf{m}), \tag{1}$$

where $\mathbf{m}$ is the magnetization unit vector, $\mathbf{h}_{\text{eff}}$ is the effective field normalized by the saturation magnetization M_s, and time is measured in units of $(\gamma M_s)^{-1}$ (γ is the absolute value of the gyromagnetic ratio).

In the case of single-domain particles, the effective field can be written as

$$\mathbf{h}_{\text{eff}} = -\frac{\partial g}{\partial \mathbf{m}} = -D \cdot \mathbf{m} + \mathbf{h}_a, \tag{2}$$

where D is a symmetric matrix (usually $D = \text{diag}\{D_x, D_y, D_z\}$, with D_x, D_y, D_z being effective demagnetizing factors referred to the principal axes of the particle and taking into account shape and uniaxial magneto-crystalline anisotropy), $\mathbf{h}_a$ is the external applied magnetic field and g is the free energy (in units of $\mu_0 M_s^2 V$, where V is the volume of the particle), which is

$$g(\mathbf{m}, \mathbf{h}_a) = \frac{1}{2}\mathbf{m}^T \cdot D \cdot \mathbf{m} - \mathbf{h}_a^T \cdot \mathbf{m}. \tag{3}$$

It is well known that, under constant applied field $\mathbf{h}_a$, Eq. (1) admits two integrals of motion, i.e. magnetization amplitude and free energy, which can be seen by scalar multiplying both sides of (1) by $\mathbf{m}$ and $\mathbf{h}_{\text{eff}}$, respectively:

$$\mathbf{m} \cdot \frac{d\mathbf{m}}{dt} = \frac{d}{dt}|\mathbf{m}|^2 = 0 \Leftrightarrow \quad |\mathbf{m}|^2 = 1, \tag{4}$$

$$\frac{d\mathbf{m}}{dt} \cdot \mathbf{h}_{\text{eff}} = -\frac{dg}{dt} = 0 \Leftrightarrow \quad g(\mathbf{m}, \mathbf{h}_a) = g_0, \tag{5}$$

where $g_0 = g(\mathbf{m}(t = 0), \mathbf{h}_a)$. The former equation implies that magnetization dynamics must occur on the unit-sphere, whereas the second equation forces magnetization to evolve along the contour lines (at level g_0) of the free energy g on the unit-sphere.

When time integration of Eq. (1) is performed according to usual numerical techniques (Euler, Runge-Kutta, etc.), the conservation properties (4)–(5) are not preserved in a discrete sense exactly, but rather with a global truncation error of order $\mathcal{O}(\Delta t^q)$, where Δt is the time step of the method and q is an integer number which generally coincides with the classical order of the numerical method denoted with p.

In the following, we introduce a class of numerical methods which present $q > p$, namely they are able to enforce the preservation of properties (4)–(5) to a higher order of accuracy with respect to the classical order p.

A set of techniques characterized by this property is based on a special class of explicit Runge-Kutta (RK) schemes referred to as pseudo-symplectic RK methods.[10] Basically, they are s-stage explicit Runge-Kutta methods where the coefficients a_{ij} and b_i in the Butcher's tableau are designed in order to yield $q > p$. This approach has been recently developed in the area of computational fluid

dynamics for incompressible Navier-Stokes equations which can be expressed, in the inviscid limit, in the following semi-discretized form:[11]

$$\frac{d\mathbf{u}}{dt} = F(\mathbf{u}) \cdot \mathbf{u} = -P \cdot C(\mathbf{u}) \cdot \mathbf{u}, \tag{6}$$

where $\mathbf{u}$ is the discrete velocity vector, $C(\mathbf{u})$ is a linear skew-symmetric operator arising from appropriate spatial semi-discretization of the convective term, and P is a symmetric projection operator accounting for the pressure gradient. By using this approach, pseudo-symplectic s-stage RK methods capable of preserving quadratic invariants of the dynamical system (6) up to order $q = 7$ have been developed.[11]

Now we observe that conservative LL dynamics can be easily put in a form similar to Eq. (6) when $\mathbf{h}_a = 0$ ($\mathbf{h}_a \neq 0$ does not affect the results of our analysis). In fact, by introducing the skew-symmetric operator $\Lambda(\mathbf{v})$ such that $\Lambda(\mathbf{v}) \cdot \mathbf{w} = \mathbf{v} \times \mathbf{w}$, Eq. (1) can be rewritten as:

$$\frac{d\mathbf{m}}{dt} = F(\mathbf{m}) \cdot \mathbf{m} = \Lambda(D \cdot \mathbf{m}) \cdot \mathbf{m}. \tag{7}$$

In the following, we consider a generic s-stage Runge-Kutta method for Eq. (7), which reads:

$$\mathbf{m}^{n+1} = \mathbf{m}^n + \Delta t \sum_{i=1}^{s} b_i F(\mathbf{m}_i) \cdot \mathbf{m}_i, \tag{8}$$

$$\mathbf{m}_i = \mathbf{m}^n + \Delta t \sum_{j=1}^{s} a_{ij} F(\mathbf{m}_j) \cdot \mathbf{m}_j, \tag{9}$$

where a_{ij} and b_i are the scheme coefficients, and determine the order of accuracy and the properties of each particular method. Note that for explicit schemes $a_{ij} = 0$ for $j \geq i$. If one computes the variation of $|\mathbf{m}|^2 = \mathbf{m}^T \cdot \mathbf{m}$ in one time step using Eqs. (8)–(9), it can be seen that:[11]

$$\frac{\Delta |\mathbf{m}|^2}{\Delta t} = -\Delta t \sum_{i,j=1}^{s} \left(b_i a_{ij} + b_j a_{ji} - b_i b_j \right) \mathbf{m}_i^T \cdot F^T(\mathbf{m}_i) \cdot F(\mathbf{m}_j) \cdot \mathbf{m}_j. \tag{10}$$

A similar equation can be written for the energy rate of change related to Eq. (5). In order to set $\Delta |\mathbf{m}|^2$ to zero, one should impose the condition $\left(b_i a_{ij} + b_j a_{ji} - b_i b_j \right) = 0$, which would require using costly implicit schemes.[13]

However, by following the approach described in Ref. 11, appropriate RK coefficients which guarantee order $q > p$ can be derived. A list of pseudo-symplectic RK (PSRK) methods relevant to magnetization dynamics is reported in Table I. In order to test these PSRK schemes, Eq. (1) has been integrated with random initial conditions and with D being a random symmetric matrix. The time step was selected to be equally log-spaced across the interval $\Delta t \in [0.01, 0.1]$. The numerically determined orders p, q are also reported in table I. Remarkably enough, we notice that, despite the theoretically expected $q = 4$ for the 2p4q3s scheme, we obtain order 5 in both amplitude and energy conservation with 3 stages, which means only 3 evaluations of the effective field. Further tests on these schemes are reported in the sequel. The pseudo-symplectic schemes described before are able to considerably reduce the error on the conservation of quadratic invariants, on equal time step, with

TABLE I. List of pseudo-symplectic explicit (PSRK) methods for LL dynamics. The shorthand notation 2p3q2s(3) means $p = 2$, $q = 3$, $s = 2$ and 3 evaluations per time step of the right-hand side of Eq. (7). The order of accuracy found by numerical tests is also reported. Nonzero coefficients of the Butcher's tableau are listed in the last column.

| RK scheme | order of accuracy | | | RK coefficients |
	$\|\Delta m\|$	Δg	$1 - |m|^2$	
2p3q2s(2)	2	3	3	$a_{21} = 1,\ b_1 = 1/2$
2p4q3s(3)	2	5	5	$a_{21} = 1/2,\ a_{32} = 1,\ b_1 = 1/4,\ b_2 = 1/2,\ b_3 = 1/4,\ c_2 = 1/2,\ c_3 = 1$
3p6q5s(5)	4	7	7	Ref. 10, p. 453
4p7q6s(6)	4	7	7	Ref. 11, p. 90

respect to standard Runge-Kutta methods. Here, we propose to further improve the efficiency of the algorithm by employing an adaptive time step size selection, which is a widely used strategy in numerical methods aimed to obtain the same accuracy with fewer steps, or better accuracy with the same number of steps.[14] Such techniques are based on computing a proper (error) controller, and to adjust the step size dynamically to ensure that the error is kept within a user-prescribed tolerance. In most situations, the controller is the local truncation error T^{n+1}, and the updating formula for the time step size to keep it within the desired tolerance δ reads $\Delta t^{n+1} = \Delta t^n \left|\frac{\delta}{T^{n+1}}\right|^{1/(p+1)}$. In general, the local truncation error is computed by comparing two numerical solutions, with one being more accurate than the other. This usually leads to non-negligible increase in computational cost, as for the case of the Taylor-series method or the Richardson extrapolation.[14] Of particular interest are methods with built-in error estimates, such as the so-called *embedded* Runge-Kutta schemes, which are able to provide two numerical solutions of different accuracy with little additional computational cost.[15] Embedded methods will be addressed in more detail in future work. The adaptive time-stepping strategy proposed in this paper is inspired by a method recently developed by Capuano *et al.* in the context of the fluid flow equations, with the aim of preserving kinetic energy efficiently.[16] The main idea is to employ the error on the conservation of quadratic invariants as the error controller in the computation of Δt^{n+1}, and to adjust the time step accordingly. The rationale behind the method is that for conservative or slightly dissipative systems, the most important character of the temporal error to be controlled is the one contributing spuriously to the balance of quadratic invariants. Indeed, these are intrinsic properties of the continuous system and, as such, should be preserved also on a discrete level. Supposedly, the accuracy of the solution itself is in turn indirectly controlled, and the costly computation of two solutions is inherently avoided. Again, we remark the similarity between the Landau-Lifshitz and the Navier-Stokes equations, both possessing two quadratic invariants. While the analogue of the free energy g is the global kinetic energy of the flow, a Casimir inviscid invariant of the Navier-Stokes equations is the so-called helicity, which is the mean scalar product of the velocity with its curl.[17]

In the case of conservative LL dynamics, the discrete quadratic invariants evolution is governed by Eq. (10), together with its counterpart for free energy. We thus define the two error controllers

$$\chi_{\mathbf{m}} \equiv \Delta|\mathbf{m}|^2 \sim \mathcal{O}(\Delta t^{q+1}), \;\; \chi_g \equiv \frac{\Delta g}{g_0} \sim \mathcal{O}(\Delta t^{q+1}) \tag{11}$$

and select the time step as follows

$$\Delta t^{n+1} = \Delta t^n \min\left\{\left|\frac{\delta^{\mathbf{m}}}{\chi_{\mathbf{m}}^{n+1}}\right|^{\frac{1}{q+1}}, \left|\frac{\delta^g}{\chi_g^{n+1}}\right|^{\frac{1}{q+1}}\right\}, \tag{12}$$

where $\delta^{\mathbf{m}}$ and δ^g are the chosen tolerances for $\chi_{\mathbf{m}}$ and χ_g respectively. After preliminary tests we have chosen $\delta^{\mathbf{m}} = 10^{-11}$ and $\delta^g = 10^{-8}$, but of course different values can be selected depending on the required accuracy of the problem under study. We also note that the practical implementation of Eq. (12) requires the explicit computation of $|\mathbf{m}|^2$ and g, with little additional computational cost. Nevertheless, such quantities are often computed as part of the output of numerical micromagnetic codes and, therefore, do not produce as a matter of fact any practical increase of computational cost.

In order to compare the proposed methods, we apply them to simulate conservative magnetization dynamics associated with fast precessional switching, which occurs by applying a field transverse to the initial magnetization and is much faster than conventional switching.[18,19] Successful switching is obtained by switching the field off at the right moment[20] and the equilibrium magnetization is achieved after relaxation from a high to low energy state.[21] However, the ballistic dynamics is so fast that the dissipation can be neglected and the essence of the process can be inferred from analyzing conservative dynamics. We consider here precessional switching driven by hard-axis field pulses, which is very fast owing to the strong out-of-plane demagnetizing effects.[12] In this situation, despite its strongly nonlinear nature, the LL Eq. (1) can be solved in exact form.[22] Thus, we consider the demagnetizing coefficients $D_x = 0.0411$, $D_y = 0.1062$, $D_z = 0.8527$ and the driving field pulse along the hard-axis z with amplitude $h_{az} = 0.3$ greater than the critical field $\sqrt{(D_y - D_x)(D_z - D_y)} = 0.22$.
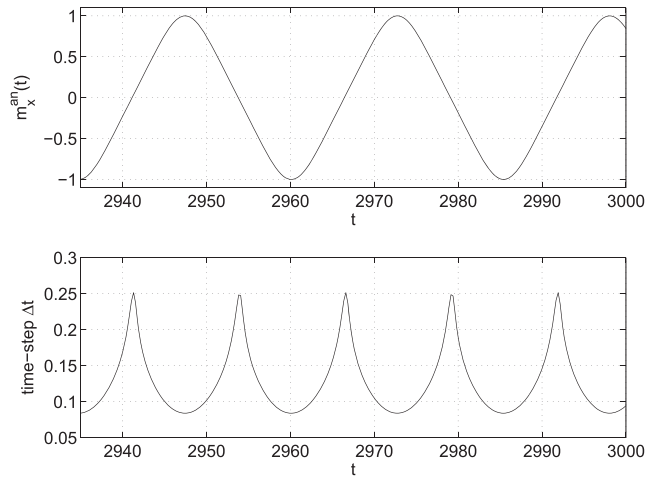
FIG. 1. (top) analytical solution $m_x^{an}(t)$ of Eq. (1) vs. time; (bottom) time step selected by adaptive controller vs. time. Results obtained using the scheme PSRK 2p4q3s.

The initial magnetization is $m_x = -1$ and the initial time step is $\Delta t = 0.1$ (corresponding to 0.57 ps if $\mu_0 M_s = 1$T).

We present results for two implementations of the second-order ($p = 2$) scheme PSRK 2p4q3s (see Table I): the former with fixed time step $\Delta t = 0.1$ and the latter with the aforementioned adaptive time step control. The tolerances for the adaptive controller have been chosen $\delta^{\mathbf{m}} = 10^{-11}$ and $\delta^g = 10^{-8}$, respectively. In order to monitor the behavior of global truncation errors on solution, amplitude and energy over time, we simulate a moderately long time interval $t \in [0, 6000]$ (corresponding to about 34.2 ns in physical units). Some periods of the analytical solution $m_x^{an}(t)$ are reported in the top panel of Fig. 1. Then, in the lower panel of Fig. 1, the time step selected by the adaptive controller, ranging from 0.081 to 0.251, is plotted as function of time. Remarkably, the adaptive controller yields a number of 51938 steps for the above simulation, which is 13.4% less than 60000 associated with the fixed time step $\Delta t = 0.1$. As far as accuracy is concerned, the deviation of the numerical
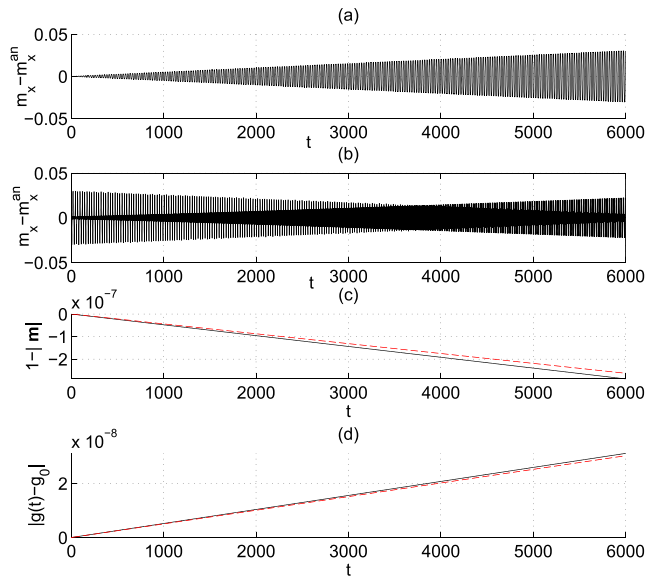

FIG. 2. Deviation of numerical solution $m_x(t)$ from the analytical solution $m_x^{an}(t)$ of Eq. (1) vs. time for (a) fixed, (b) adaptive PSRK method, respectively; (c) magnetization amplitude preservation test, $1 - |\mathbf{m}|$ vs. time; (d) free energy preservation test, $|g(t) - g_0|$ vs. time, $g_0 = \frac{D_x}{2} = 0.0205$. Solid (dashed) lines (c)-(d) refer to fixed (adaptive) method.

solution with respect to the analytical one as function of time is reported in Fig. 2(a)–(b) for the fixed and adaptive time step implementations, respectively. Although for the sake of clarity we have reported only the error for $m_x(t)$, we remark that the error norm $|\mathbf{m} - \mathbf{m}^{an}|$ has a similar behavior. By comparing Fig. 2(a)–(b), one can see that the maximum global error at the end of the simulation is 0.03 for the fixed and 0.02 for the adaptive scheme. Thus, one can conclude that the adaptive time step method is able to reduce the growth rate of the global error on the solution $m_x(t)$ with respect to the fixed time step method with a lower computational cost. This is a desirable feature when long-term magnetization dynamics has to be simulated. Furthermore, as far as the amplitude and energy conservation properties are concerned, we have reported time evolutions of the quantities $1 - |\mathbf{m}|$ and $|g(t) - g_0|$ in Fig. 2(c)–(d). On one hand, it can be observed that PSRK with effective order $q = 5$ exhibits a remarkable performance both on magnetization amplitude and energy conservation, keeping truncation errors respectively around 10^{-7} and 10^{-8} with only three effective field evaluations. On the other hand, PSRK methods can be easily integrated in micromagnetic codes by using standard numerical integration libraries. As a final remark, we again point out that, when non-conservative terms (damping, spin-torque, etc.) are considered in the LL equation, the conservation properties of PSRK methods reflect in the enforcement of the correct physical energy balance of the dynamics. Moreover, when spatially-inhomogeneous magnetization dynamics is considered, the above results still hold, provided that the spatial discretization preserves the structural properties of the effective field operator (symmetry of matrix D), as in usual finite-difference/finite-element methods.

[1] P. J. Channell *et al.*, Nonlinearity **3**, 231 (1990).
[2] M. A. Austin *et al.*, J. Comput. Phys. **107**, 105 (1993).
[3] P. S. Krishnaprasad *et al.*, Physica B **306**, 195 (2001).
[4] C. Serpico *et al.*, J. Appl. Phys. **89**, 6991 (2001).
[5] G. Bertotti, I. D. Mayergoyz, and C. Serpico, *Nonlinear Magnetization Dynamics in Nanosystems* (Elsevier, 2009).
[6] P. B. Visscher *et al.*, Phys. Rev. B **65**, 104412 (2002).
[7] M. d'Aquino *et al.*, J. Comput. Phys. **209**, 730–753 (2005).
[8] B. Van de Wiele *et al.*, IEEE Trans. Magn. **43**, 2917 (2007).
[9] D. Porter *et al.*, Journal of Applied Physics **103**, 07D920 (2008).
[10] A. Aubry *et al.*, BIT Numer. Math. **38**, 439–461 (1998).
[11] F. Capuano *et al.*, J. Comput. Phys. **328**, 86–94 (2017).
[12] M. d'Aquino *et al.*, IEEE Trans. Magn. **50**, 7100504 (2014).
[13] J. M. Sanz-Serna, BIT Numer. Math. **28**, 877–883 (1988).
[14] D. Griffiths and D. J. Higham, *Numerical Methods for Ordinary Differential Equations* (Springer, 2010).
[15] J. C. Butcher, *Numerical Methods for Ordinary Differential Equations* (Wiley, 2016).
[16] F. Capuano *et al.*, Proceedings of the XXIII AIMETA Conference, vol. 5, 2257–2269 (2017).
[17] P. J. Olver, J. Math. Anal. Appl. **89**, 233–250 (1982).
[18] G. Bertotti *et al.*, IEEE Trans. Magn. **39**, 2501 (2003).
[19] M. Bauer *et al.*, Phys. Rev. B **61**, 3410 (2000).
[20] M. d'Aquino *et al.*, J. Appl. Phys. **95**, 7055 (2004).
[21] C. Serpico *et al.*, IEEE Trans. Magn. **45**, 5224 (2009).
[22] M. d'Aquino *et al.*, Physica B **486**, 126–129 (2016).