Time resolved micromagnetics using a preconditioned time integration method

D. Suess, V. Tsiantos, T. Schrefl, J. Fidler, W. Scholz, H. Forster, R. Dittrich

Institute of Applied and Technical Physics, Vienna University of Technology,

Wiedner Hauptstr. 8-10, A-1040 Vienna, Austria

and

J. J. Miles

Department of Computer Science, University of Manchester,

Oxford Road, Manchester M13 9PL, United Kingdom

Keywords: finite elements, preconditioning, micromagnetics, magnetic elements, time integration

Abstract

A detailed description for the solution of the Landau-Lifshitz-Gilbert equation with the finite element method is given. The use of implicit time integration schemes with proper preconditioning is reported. Simulations of a single phase magnetic nanoelement without surface roughness and a magnetic nanoelement with a granular structure are performed to investigate the influence of the microstructure on the numerical behavior. Nanoelements with a granular structure cause an inhomogeneous computational grid. In granular systems preconditioning for time integration speeds up the simulations by three orders of magnitude as compared to conventional time integration schemes like the Adams method.

1 Introduction

Numerical micromagnetics is an essential tool to optimize magnets in magnetic storage and sensors. The application of these devices requires a profound knowledge of the reversal mechanism. Using the Landau-Lifshitz-Gilbert (LLG) equation the time evolution of the magnetization can be calculated. The treatment of systems with realistic size leads to a system

of ordinary differential equations with up to one million unknowns. State of the art time integration schemes provide an efficient numerical solution of the equations.

Traditionally explicit time integration methods are used in numerical micromagnetics. These methods are easy to apply since only the right hand side of the Landau-Lifshitz-Gilbert equation has to be evaluated. Victora applied a Runge-Kutta [1] method to solve the Landau-Lifshitz-Gilbert in an array of columnar CoNi particles. Different types of Adams formulas are commonly used in micromagnetics. Mansuripur applied an Adams method [2] to calculate the time evolution of a thin film recording media. The Adams method that Zhu and Bertram used [3] is a varying order and varying step-size method and is well described by Gear [4]. Jones and Miles [5] used the LLG equation to simulate the magnetic behavior of a metal evaporated tape. They performed the integration by using a variable-order variable-step Adams method. More specifically, they used the NAG D02CHF routine. Tako et al [6] used the Adams 4th-order predictor-corrector method with error control and step size adjustment described in [7]. McMichael and Donahue use a second order predictor-corrector technique of the Adams type to calculate the dynamic response of magnetic nanoelements [8].

However, for highly exchange coupled systems or complex microstructures the Adams methods and explicit time integration schemes require an intolerable small time step to maintain numerical stability. These problems are supposed to be stiff and can be more suitably solved with backward differentiation forumla methods. Originally, Hayashi and Nakatani [9] applied the backward Euler method which is a backward differentiation method (BDF) of order 1, to solve the Landau-Lifshitz equation for magnetic bubble domain wall motion. Albuquerque and Miltat [11] treat the exchange term implicitly applying the Crank Nicolson method. This scheme applies a feedback mechanism from monitoring the damping coefficient, maximum torque and total energy to obtain high accuracy. E and Wang [10] developed a projection method. The key point of this method is that they relax the condition that the magnitude of the magnetization vector remains constant. After each time step the magnetization vector is projected back to the unit sphere. The proposed method is unconditionally stable.

The above methods are used together with a finite difference method for space discretization. The finite element method allows irregular grids suitable for complex microstructures and adaptive refinement. For explicit methods the maximum suitable time step is proportional to h_g^2, where h_g is the mesh size. Thus a fine mesh which, is required to resolve magnetization inhomogeneities near edges or grain boundaries, limits the time step of explicit

methods. BDF methods are more suitable. However, the linear system which has to be solved at every time step as part of the Newton method is ill conditioned. Proper preconditioning decreases the number of iterations when the system is solved iteratively and thus speeds up the computation considerably.

Yang and Fredkin [12] developed a numerical procedure to study the dynamic behavior in micromagnetic systems using the finite element method. They solved the damped Gilbert equation for a continuous magnetic medium, including all the interactions in standard micromagnetic theory in 3D regions of arbitrary geometry and physical properties. The magnetization is linearly interpolated in each tetrahedral element in a finite element mesh from its value on the nodes, and they use the Galerkin method to discretize the dynamic equation. The demagnetizing field is computed by solving Poisson's equation and they treat the external region by means of an asymptotic boundary condition. They apply the CVODE code [14-15] to solve the stiff system of ordinary differential equations. CVODE provides the option to solve the equations either with the Adams or a backward differentiation formula (BDF) method. In order to apply a preconditioner for the linear system, CVODE requires an approximate Jacobian, which is the first derivative of the right hand side of the Landau-Lifshitz-Gilbert equation. Yang and co-workes cannot supply the Jacobian explicitly because they use the Galerkin method for space discretization. However, they are able to give a suitable approximation. A more detailed description of the numerical methods for ODEs in micromagnetics is given in [13].

In this paper we use the finite element method to compute the dynamic response of thin film elements of different shape and structure. Starting from the finite element discretization of the total energy, the effective field can be evaluated using the box method. Each node has its magnetic moment and its effective field. Thus the right hand side of the LLG equation and the Jacobian can be calculated explicitly. For the preconditioning only an approximation of the Jacobian is needed. Thus it is possible to keep the approximated Jacobian sparse, omitting the stray field part. In section 2 we introduce the finite element techniques used for space discretization, the calculation of the effective fields, and the approximate Jacobian. The demagnetizing field is calculated using a hybrid finite element boundary method as discussed in section 3. Section 4 treats the time integration of the discretized LLG equation. Section 5 gives numerical examples. The Adams method, the BDF method and the preconditioned BDF method are compared for μMAG Problem # 4 and a thin film element with granular

microstructure. The results show a considerable decrease of the CPU time for the preconditioned BDF method for all investigated samples.

2 Method

Using micromagnetics the theoretical treatment of magnetization dynamics at zero temperature starts from the Gilbert equation [17],

$$\frac{\partial \boldsymbol{J}}{\partial t} = -|\gamma| \boldsymbol{J} \times \boldsymbol{H}_{\boldsymbol{eff}} + \frac{\alpha}{J_\mathrm{s}} \boldsymbol{J} \times \frac{\partial \boldsymbol{J}}{\partial t}, \tag{1}$$

where $\boldsymbol{J}$ is the magnetic polarization vector, $\boldsymbol{H}_\mathrm{eff}$ is the effective field, and J_s is the spontaneous polarization. $\boldsymbol{J}$ is assumed to be a continuous function of space. To obtain the general form $\frac{d\boldsymbol{y}}{dt} = f(t, \boldsymbol{y})$ for an ordinary-differential equation (ODE), we transform equation (1) into the mathematically equivalent Landau-Lifshitz-Gilbert (LLG) equation

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

with the gyromagnetic ratio

$$|\gamma| = 2,210175 \times 10^5 \frac{\mathrm{m}}{\mathrm{As}}, \tag{3}$$

and the Gilbert damping constant α.

The finite element method is used to discretize equation (2). The magnetic polarization $\boldsymbol{J}(\boldsymbol{x})$ is expanded with linear basis functions $\varphi_i(\boldsymbol{x})$. For one component of $\boldsymbol{J}$ we can write

$$J^k(\boldsymbol{x}) \approx J^k_{app}(\boldsymbol{x}) = J_s(\boldsymbol{x}) \sum_{i=1}^{n} u_i^k \varphi_i(\boldsymbol{x}). \tag{4}$$

$J^k_{app}(\boldsymbol{x})$ is the finite element approximation of the k - component of the magnetic polarization. The coefficient u_i^k denotes the k-th component of the normalized spontaneous polarization ($\left|u_i^k\right| \le 1$) on the node point i. The number of unknowns (u_i^k) is three times the number of node points of the finite element mesh. The basis functions obey

$$\varphi_i(\boldsymbol{x}) \varphi_j(\boldsymbol{x}) = \delta_{ij}. \tag{5}$$

For practical reasons, the expansion of the unknown function is usually done on an element by element fashion in finite element packages. For the magnetic polarization follows

$$J_{app}^{k}(\boldsymbol{x}) = \sum_{e=1}^{N_e} J_{s,e} \sum_{i=1}^{N_b} u_{e,i}^{k} \varphi_{e,i}(\boldsymbol{x}) . \tag{6}$$

N_e and N_b denotes the number of finite elements and number of nodal points, respectively. $J_{s,e}$ is the spontaneous polarization of element e. If we use linear polynomials for the shape functions $\varphi_{e,i}(\boldsymbol{x})$ the vertices of the finite elements are equal to the nodal points. At a nodal point i the values of the reduced magnetic polarization $u^{k}(\boldsymbol{x})$ are given by the coefficients $u_{e,i}^{k}$. If tetrahedrons form the finite element mesh and linear shape functions are used $N_b = 4$. For quadratic shape functions $N_b = 10$. The shape functions obey

$$\sum_{i=1}^{N_b} \varphi_{e,i}(\boldsymbol{x}) = 1 , \tag{7}$$

$$\varphi_{e,i}(\boldsymbol{x}_j) = \delta(\boldsymbol{x}_j - \boldsymbol{x}_i) , \tag{8}$$

where $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ denote the positions of the local node points i and j of the element e, respectively. To perform the time integration, we have to calculate the effective field on every node point of the finite element mesh. However we cannot directly use the analytic formula for the effective field, which follows form the negative functional derivative of the total Gibbs energy (equation 10) as,

$$\boldsymbol{H}_{eff} = -\frac{\delta E_t}{\delta \boldsymbol{J}} = -\frac{2A}{J_s^2}\Delta \boldsymbol{J} + \frac{2}{J_s^2} K_1 (\boldsymbol{J}\boldsymbol{a})\boldsymbol{a} + \boldsymbol{H}_S + \boldsymbol{H}_{ext} . \tag{9}$$

Here A is the exchange constant, K_1 is the magnetocrystalline anisotropy constant, $\boldsymbol{a}$ is the unit vector parallel to the anisotropy axis, $\boldsymbol{H}_{\mathrm{s}}$ is the magnetostatic stray field, and $\boldsymbol{H}_{\mathrm{ext}}$ is the external applied field. The first term on the right side of equation (9) is the exchange field. Its calculation needs the second derivative of the magnetic polarization. Numerically the second derivatives can not be calculated directly using linear basis functions. In addition, the calculation of the stray field which follows from the gradient of a scalar potential is crucial. With linear basis functions the gradient of the potential, which is proportional to the stray field, is only defined within an element but not on the node points. To overcome this problem we start from the total Gibbs energy for a ferromagnetic particle [18]

$$E_t = \int_{\Omega} e_t(\boldsymbol{J})dV = \int_{\Omega}\left(A\left[\sum_{k}^{x,y,z}(\nabla u_k)^2\right] + K_1[1-(\boldsymbol{ua})^2] - \frac{1}{2}J_s\boldsymbol{uH}_s - J_s\boldsymbol{uH}_{ext}\right)dV. \quad (10)$$

The exchange energy, the anisotropy energy, the demagnetization energy and the Zeeman energy contribute to the total energy. The second term is simple uniaxial anisotropy energy. It would be no problem to replace it with any other form of anisotropy energy. No surface anisotropy is assumed. We neglect the contributions to the total energy which arise from the conversion of the true microscopic exchange and dipole interactions to the continuum form as well as intrinsic surface anisotropy [18].

The total energy is an integral over the particle volume Ω. In the following we consider a discretization of the problem domain Ω into N_e finite elements Ω_e.

$$\Omega = \bigcup_{e=1}^{N_e} \Omega_e. \quad (11)$$

Thus the integral in equation (10) decomposes into a sum of integrals over finite elements

$$E_t = \int_{\Omega} e_t(\boldsymbol{J})dV \cong \sum_{e=1}^{N_e}\int_{\Omega_e} e_t\left(J_{s,e}\sum_{i=1}^{N_b} u_{e,i}^{k}\varphi_{e,i}(\boldsymbol{x})\right)dV\ . \quad (12)$$

For the magnetic polarization $\boldsymbol{J}$ the expansion according to equation (6) is used. The k-th component of the effective field on node i is approximated using the box scheme [19],

$$H_{i\,eff}^{k} \approx -\frac{1}{m_i}\frac{\partial E_t}{\partial u_i^k}, \quad (13)$$

where m_i denotes the magnetic moment on the node point i. It follows from the integral

$$m_i = \int_{i} J_s(\boldsymbol{x})dV, \quad (14)$$

where V_i , as shown in figure 1, is the surrounding the volume of the node i, such that

$$\sum_{j} V_j = V \text{ and } V_i \cap V_j = 0 \text{ for } i \neq j. \quad (15)$$

Usually, in a single phase magnetic material the spontaneous polarization, J_s, is a constant in space. However the spontaneous polarization is a function of space and is discontinuous at grain boundaries if a magnet with different magnetic phases is modeled. In our model the nodes of the finite elements are located at grain boundaries. Regions with different values of the spontaneous polarization surround these points. Thus we assume an average magnetic moment for these nodes as given by equation (14).

The derivative of the total energy is calculated in an element-by-element fashion. Firsr we have to calculate $\partial E_t / \partial u^k_{e,j}$ for all local coefficients $u^k_{e,j}$. Then the element level derivatives are assembled to obtain $\partial E_t / \partial u^k_i$, the derivative of the total energy with respect to the global expansion coefficients. The assembling process can be formally written using the connectivity matrix C^e_{ij} which is defined as,

$$C^e_{ij} = \begin{cases} 1 & \text{if global node } i \text{ corresponds to the local node } j \text{ in element } e. \\ 0 & \text{else} \end{cases} . \tag{16}$$

Using C^e_{ij} the derivative of the total energy with respect to the global coefficients u^k_i follows from

$$\frac{\partial E_t}{\partial u^k_i} = \sum_{e=1}^{N_e} \sum_{j=1}^{N_b} C^e_{ij} \frac{\partial E_t}{\partial u^k_{e,j}} . \tag{17}$$

Let us first consider only the contribution of the exchange energy to the total energy, in order to give an example for the calculations of the derivative of the total energy with respect to the local coefficients $u^k_{e,j}$, . We start from the discrete form of the total energy (12). The derivative follows as

$$\frac{\partial E_{ex}}{\partial u^k_{e,j}} = \sum_{\tilde{e}=1}^{N_e} \int_{\Omega_e} A \sum_{l}^{x,y,z} \frac{\partial}{\partial u^k_{e,j}} \left(\nabla \sum_{i=1}^{N_b} u^l_{\tilde{e},i} \varphi_{\tilde{e},i}(\boldsymbol{x}) \right)^2 dV. \tag{18}$$

Applying the chain rule gives

$$\frac{\partial E_{ex}}{\partial u^k_{e,j}} = \sum_{\tilde{e}=1}^{N_e} \int_{\Omega_e} \left\{ A \sum_{l}^{x,y,z} 2 \left(\nabla \sum_{i=1}^{N_b} u^l_{\tilde{e},i} \varphi_{\tilde{e},i}(\boldsymbol{x}) \right) \cdot \frac{\partial}{\partial u^k_{e,j}} \left(\nabla \sum_{i=1}^{N_b} u^l_{\tilde{e},i} \varphi_{\tilde{e},i}(\boldsymbol{x}) \right) \right\} dV. \tag{19}$$

The derivative with respect to the local coefficient $u^k_{e,j}$ in the second factor gives

$$\frac{\partial E_{ex}}{\partial u^k_{e,j}} = \sum_{\tilde{e}=1}^{N_e} \int_{\Omega_e} \left\{ A \sum_{l}^{x,y,z} 2 \left(\nabla \sum_{i=1}^{N_b} u^l_{\tilde{e},i} \varphi_{\tilde{e},i}(\boldsymbol{x}) \right) \cdot \nabla \varphi_{e,j}(\boldsymbol{x}) \delta_{kl} \delta_{\tilde{e}e} \right\} dV \tag{20}$$

which finally leads to

$$\frac{\partial E_{ex}}{\partial u^k_{e,j}} = \int_{\Omega_e} A \left[2 \left(\sum_{i=1}^{4} u^k{}_{e,i} \nabla \varphi_{e,i}(\boldsymbol{x}) \right) \nabla \varphi_{e,j}(\boldsymbol{x}) \right] dV. \tag{21}$$

Introducing the element matrix M^e_{lj},

$$M^e_{lj} = \int_{\Omega_e} \nabla \varphi_{e,l}(\boldsymbol{x}) \nabla \varphi_{e,j}(\boldsymbol{x}) dV, \tag{22}$$

the derivative of the exchange energy can be written as,

$$\frac{\partial E_{ex}}{\partial u^k_{e,j}} = 2A \left[\sum_{l=1}^{N_b} u^k{}_{e,l} M^e_{lj} \right]. \tag{23}$$

In a similar way one can calculate the Zeeman energy E_{H}, stray field energy E_{s} and anisotropy energy E_{K}. The derivatives are

$$\frac{\partial E_H}{\partial u^k_{e,j}} = -J_s \int_{\Omega_e} H^k_{ext}(\boldsymbol{x}) \varphi_{e,j}(\boldsymbol{x}) dV, \tag{24}$$

$$\frac{\partial E_s}{\partial u^k_{e,j}} = -J_s \int_{\Omega_e} H^k_d(\boldsymbol{x}) \varphi_{e,j}(\boldsymbol{x}) dV, \text{ and} \tag{25}$$

$$\frac{\partial E_K}{\partial u^k_{e,j}} = -2K_1 a^k \int_{\Omega_e} \sum_{c=1}^{3} J^c(\boldsymbol{x}) a^c \varphi_{e,j}(\boldsymbol{x}) dV. \tag{26}$$

The space discretization of equation (2) leads to a system of ordinary differential equations. Its numerical solution can be very efficient if one can provide the Jacobian matrix of the system. The calculation of the Jacobian needs the second derivative of the total energy. Again

the calculation is done element by element. Using equation (23)-(26) we build the second derivative of the total energy with respect to the local coefficients $u^k_{e,j}$. For the exchange energy and anisotropy energy follows

$$\frac{\partial^2 E_{ex}}{\partial u^k_{e1,m}\partial u^l_{e2,n}} = 2AM_{mn}\delta_{kl}\delta_{e1,e2}. \tag{27}$$

$$\frac{\partial^2 E_K}{\partial u^k_{e1,m}\partial u^l_{e2,n}} = -2K_1\delta_{e1,e2}\int_{\Omega_e} a^k a^l \varphi_{e,m}(\boldsymbol{x})\varphi_{e,n}(\boldsymbol{x})dV. \tag{28}$$

Here a^k are the cartesian components of the unit vector parallel to the anisotropy axis. The uniform external field does not contribute to the second derivative. The stray field is omitted, in order to keep the Jacobian sparse. The assembling process from the local second derivatives to the global derivatives of the total energy can be formally written as

$$\frac{\partial^2 E_t}{\partial u^k_m \partial u^l_n} = \sum_{e=1}^{N_e}\sum_{j,i=1}^{N_b} C^e_{mi}\,\frac{\partial^2 E_t}{\partial u^k_{e,i}\partial u^l_{e,j}} C^e_{nj} \quad . \tag{29}$$

2.1 Calculation of the stray field

The stray field $\boldsymbol{H}_S$ is obtained from a boundary value problem,

$$\Delta u = \frac{\nabla \boldsymbol{J}_s}{\mu_0} \text{ and } H_S = -\nabla u \tag{30}$$

To apply the boundary condition $u = 0$ at infinity, a hybrid finite element boundary element method [20] is used. No finite elements are needed outside the magnetic particle to solve the boundary value problem (30). This is the advantage of the hybrid FE/BE method. For the solution of (30) with the hybrid FE/BE method one Poisson equation with Neumann boundary conditions and one Laplace equation with Dirichlet boundary conditions have to be solved. To obtain the boundary conditions a matrix vector product has to be performed. We split the total magnetic potential u into two parts, $u = u_1 + u_2$. The potential u_1 solves the Poisson equation (30) inside the magnetic particles with Neumann boundary conditions at the surface of the magnets and it is zero outside the magnets. The potential u_2 solves the

Laplace equation everywhere in space and shows a jump at the surfaces of the magnets. Thus u_2 is the potential from a dipole sheet at the surfaces of the magnet. After discretization the integral operator may be expressed as a matrix vector product

$$\boldsymbol{u}_2 = B\boldsymbol{u}_1 \tag{31}$$

The storage requirement for the matrix B is the bottleneck of the method since B is a fully populated $N_S \times N_S$ matrix. N_S is the number of boundary nodes. Especially for thin films the method loses efficiency since most of the nodes are located at the boundary.

3 Time integration

We use the CVODE code [14-15] for solving the LLG equation. The relative performance of the Adams method, the BDF method and the BDF method with preconditioning changes depending on the microstructure, material parameters, and the finite element mesh. The Adams method seems to be attractive because of the low cost per time step. In contrast to the one-step methods (e.g. Runge-Kutta) multistep methods make use of the past values of the solution. At each time step a non linear system of equations has to be solved. The Adams method solves the nonlinear system with functional iteration and thus requires only the evaluation of the right hand side of equation (2). However, if the problem is stiff the convergence of the functional iterations is slow. For a stiff problem it is advisable to use an implicit method such as BDF. The nonlinear system is solved using a Newton method. Normally only a few Newton steps are required. Within CVODE the linear system for each Newton-step is solved either with a direct solver or with a Krylov subspace method. Krylov subspace methods have been explored in micromagnetics by Tsiantos et al. [13,16]. The solution is approximated iteratively by a linear combination of the basis vectors of the Krylov subspace. At each iteration step one orthonormal basis vector is added which increases the subspace dimension by one. If the Krylov subspace dimension is equal to the number of unknowns the exact solution is found. For practical applications a very good approximation is obtained for a Krylov subspace dimension much smaller than the number of unknowns. The default value for the maximum Krylov subspace dimension in CVODE is 5. As discussed in the next section the computation time drastically decreases when this value is increased to about 300. An additional parameter in the CVODE package is the maximum order of the time integration

method. Especially for small damping constants, we found that a maximum order, $qu_{\max} = 2$, improves the stability of the solution.

The BDF method leads to the nonlinear system [21]

$$\boldsymbol{F}(\boldsymbol{y}_n) = \boldsymbol{y}_n - \boldsymbol{a}_n - h\beta_0 \boldsymbol{f}(t_n, \boldsymbol{y}_n) = 0 \tag{32}$$

which is solved by Newton iteration. $\boldsymbol{a}_n$ and β_0 are constants which depend on the order of the time integration method and on the previous time step size. Equation (32) is successively solved by adding corrections $\Delta\boldsymbol{y}$ to the solution vector. $\Delta\boldsymbol{y}$ is the solution of the linear system,

$$\frac{\partial \boldsymbol{F}(\boldsymbol{y}_{n-1})}{\partial \boldsymbol{y}} \Delta\boldsymbol{y} = A\Delta\boldsymbol{y} = -\boldsymbol{F}(\boldsymbol{y}_{n-1}) \quad , \tag{33}$$

with

$$A = 1 - h\beta_0 J \text{ and } J = \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{y}}. \tag{34}$$

Equation (33) is solved with a Krylov method. It starts with a guess $\boldsymbol{x}_0$ for $\Delta\boldsymbol{y}$, and corrects it successively to get iterates $\boldsymbol{x}_1$, $\boldsymbol{x}_2$ ... . After a few iterations a good aproximation for $\Delta\boldsymbol{y}$ is found. In every iteration only the product $A\boldsymbol{x}_i$ has to be calculated. So the large matrix A has to be neither explicitly constructed nor stored. In micromagentics A is fully populated owing to the long range magnetostatic interaction. $A\Delta\boldsymbol{y}$ is approximated using finite differences,

$$A(\boldsymbol{y}_{n-1})\boldsymbol{v} \approx \frac{\boldsymbol{F}(\boldsymbol{y}_{n-1} + \varepsilon\boldsymbol{v}) - \boldsymbol{F}(\boldsymbol{y}_{n-1})}{\varepsilon}. \tag{35}$$

However the number of Krylov iterations strongly depends on the matrix A. For some matrices A the Krylov methods converge slowly. This problem can be overcome by preconditioning. Instead of the system

$$A\Delta\boldsymbol{y} = \boldsymbol{b} \tag{36}$$

the equivalent system

$$(AP^{-1})(P\Delta\boldsymbol{y}) = A'\boldsymbol{x}' = \boldsymbol{b} \tag{37}$$

is solved. The matrix P should be an approximation to A. Then AP^{-1} is close to the identity matrix and the system $A'\boldsymbol{x}' = \boldsymbol{b}$ can be solved very efficiently with only a few Krylov iterations. The the matrix AP^{-1} cannot be calculated directly since generally the matrix A is not explicitly constructed. To calculate the matrix vector product, $\boldsymbol{s}:=AP^{-1}\boldsymbol{x}$, which is needed in every Krylov iteration, the following procedure is applied. Instead of the calculation of $\boldsymbol{s}$ from

$$\boldsymbol{s} = AP^{-1}\boldsymbol{x}, \tag{38}$$

we multiply equation 38 by A^{-1} and calculate $\boldsymbol{w}$ from

$$A^{-1}\boldsymbol{s} = P^{-1}\boldsymbol{x} =: \boldsymbol{w} \tag{39}$$

$\boldsymbol{w}$ is calculated by solving the linear system

$$P\boldsymbol{w} = \boldsymbol{x} \tag{40}$$

with an efficient iterative solver, since the inversion of a matrix is very time consuming. Once $\boldsymbol{w}$ has been calculated, $\boldsymbol{s}$ follows from equation (39) simply be the matrix vector product

$$\boldsymbol{s} = A\boldsymbol{w} \tag{41}$$

$$A = 1 - h\beta_0 J \tag{42}$$

We found that the solution of equation (40) is the most time consuming part in our micromagnetic simulation. We tested different methods to solve equation (40) (see section 5). The linear system of equations (40) is sparse. As the number of unkowns is large, the use of an iterative method is more appropriate than solving (40) with a direct method. Among different generalized minimum residual methods (GMRES), the Bi-Conjugate Gradient method was found to be the most efficient one. For some problems it is important to apply a preconditioner to the linear system (equation 40) to achieve good convergence. A good choice is the incomplete factorization technique RILU [22].

3.1 Error control

In the CVODE code the local truncation error e_i is estimated. The tolerance parameters *reltol* and the vector *abstol*, which can be defined for every solution component separately,

can be used to control the error. The time step of the integration is determined so that the error satisfies the inequality

$$\sqrt{\frac{\sum_{i=1}^{N}\left(e_i \frac{1}{reltol|u_i| + abstol_i}\right)^2}{N}} \le 1 . \tag{43}$$

In micromagnetics a useful error indicator follows from the structure of the LLG equation. The LLG equation maintains the magnitudes of the magnetic polarization. However the discretization of the LLG equation using the BDF formula violates this relation. For example for the implicit Euler method with time step h (BDF method or Adams method of order one) follows,

$$\frac{\boldsymbol{J}_{i+1} - \boldsymbol{J}_i}{h} = -\frac{|\gamma|}{1+\alpha^2}\boldsymbol{J}_{i+1} \times \boldsymbol{H}_{\boldsymbol{eff}}(\boldsymbol{J}_{i+1}) - \frac{\alpha}{1+\alpha^2}\frac{|\gamma|}{J_s}\boldsymbol{J}_{i+1} \times (\boldsymbol{J}_{i+1} \times \boldsymbol{H}_{\boldsymbol{eff}}(\boldsymbol{J}_{i+1})) . \tag{44}$$

If equation (44) is multiplied by $\boldsymbol{J}_{i+1}$ the right hand side vanishes.

$$\left(\frac{\boldsymbol{J}_{i+1} - \boldsymbol{J}_i}{h}\right)\boldsymbol{J}_{i+1} = 0 . \tag{45}$$

Substituting $\boldsymbol{J}_i\boldsymbol{J}_{i+1}$ by $\frac{1}{2}(\boldsymbol{J}_i^2 + \boldsymbol{J}_{i+1}^2 - (\boldsymbol{J}_i - \boldsymbol{J}_{i+1})^2)$ in equation (45) it follows,

$$(\boldsymbol{J}_{i+1})^2 = (\boldsymbol{J}_i)^2 + (\boldsymbol{J}_{i+1} - \boldsymbol{J}_i)^2 . \tag{46}$$

As a consequence $(\boldsymbol{J}_{i+1})^2 \neq \boldsymbol{J}_0^2$ for $i > 0$. To guarantee that the norm of the magnetic polarization does not drift away, we normalize the magnetic polarization on every node if on at least one node the deviation becomes larger than the specified tolerance *rentol*. To determine the accuracy of an integration method we use the number of renormalization steps during the simulation as well as the deviation norm of the spontaneous polarization,

$$DN = \frac{1}{J_s} max(|\boldsymbol{J}|_i - J_s) \quad , \tag{47}$$

where $|\boldsymbol{J}|_i$ is the magnitude of the polarization vector on node *i*.

3.2 Storage requirements

The size of the workspace required for the scaled preconditioned GMRES iterative solver for the linear system is

$$n(l_{max}+5)+l_{max}(l_{max}+4)+1\,, \tag{48}$$

where $n = 3N$ is the number of unknowns, $l_{\max}$ the maximum dimension of the Krylov subspace and N the number of nodes of the finite element mesh. For our micromagnetic simulations the storage requirement of the GMRES solver does not dominate the total memory. For example the simulation for a granular element (Section 5) with 3512 nodes and 2466 boundary nodes requires 48Mb for the storage of the $N_S \times N_S$ boundary matrix B (equation 31), where N_S is the number of nodal points on the surface. An increase of $l_{\max}$ from 5 to 300 increases the storage requirement for the GMRES solver from 0.2 Mb to 9.2 MB. However the CPU time decreases by one order of magnitude when $l_{\max}$ is changed from 5 to 300. For thin samples where all nodes are located on the surface $N_S = N$ the storage requirements of the B-matrix scales with N^2. However, the Krylov subspace only scales with N. The situation slightly changes if the magnetic sample has almost the same lateral dimension in all three directions in space. For a sphere, which is discretized with tetrahedrons inside and triangles on the surface, the number of node points inside and the number of points on the surface is $N_V \sim l^3(16\pi/\sqrt{2})$ and $N_S \sim l^2(8\pi/\sqrt{3})$, respectively. l is the ratio between the radius of the sphere and the edge length of one tetrahedron. The storage requirement for the B matrix is $(N_S)^2$, so it scales with l^4. However, the workspace needed for the Krylov method only scales with l^3. So even for the case where the boundary matrix is relatively small it requires much more storage then the GMRES iterative solver.

4 µMAG Problem # 4

We used the µMAG standard problem number 4 to compare the efficiency of different time integration schemes. In the µMAG problem the reversal process of a permalloy film with the dimensions x = 500 nm, y = 125 nm and z = 3 nm has to be calculated. We used A = 1.2 x 10^{-11} J/m, J_s = 1 T and zero magnetocrystalline anisotropy. We focused our investigation on the reversal process when the field $\mu_0 H$=-35.5 mT, $\mu_0 H_y$=-6.3 mT, $\mu_0 H_z$= 0.0 is applied instantaneously to the initial S-state. The S-state is obtained after applying a saturating field along the [1,1,1] direction which is slowly reduced to zero. The mesh size of the finite

element mesh is 5 nm, in the following calculations. The reversal process obtained from a simulation with a mesh size of 3 nm is very similar. So we conclude that in our model a mesh size of 5nm is sufficient to resolve the micromagnetic details. Figure 2 shows the time evolution of the y - component of the magnetic polarization. The time evolution significantly depends on the initial state. A small difference in the initial state leads to a different time evolution. Thus the initial equilibrium state has to be calculated with a very strict stopping criterion. For the solid line in Figure 2 the magnetization configuration of the S-state was calculated using a stopping criterion

$$r_e = \frac{1}{J_s}\left|\frac{\Delta \boldsymbol{J}}{\Delta t}\right| < 10^{-6}. \tag{49}$$

on every node. This quantity is proportional to the torque. For the dashed line in Figure 2 $r_e < 10^{-3}$. Starting from the accurate initial state the reversal process is calculated with different options and methods of the CVODE package. In the following simulation the accuracy of the time integration was defined with the ***abstol*** parameter of CVODE. We set ***abstol*** = 10^{-3} and ***reltol*** = 0. To be able to compare the accuracy of the different methods we use the number of renormalization steps which were required to keep $DN <$ ***rentol***. This number of renormalization steps, ***nrn***, serves as an error criterion. The smaller ***nrn*** the more accurate is the time integration method. Figure 3 shows the CPU time as a function of the simulation time. The calculations are performed on a Digital EV6 workstation (523 MHz). The CPU

Table 1: Summary of statistical data for different time integration schemes after 3 ns of simulated time. **(Adams)** Adams method $qu_{\max} = 15$. **(BDF 1)** BDF-method with maximum integration order $qu_{\max} = 3$, Krylov Subspace dimension $l_{\max} = 400$. **(BDF 2)** BDF-method with $qu_{\max} = 2$ and $l_{\max} = 400$. **(BDF 3)** BDF-method with preconditioning, $qu_{\max} = 2$ and $l_{\max} = 400$. **(BDF 4)** BDF with $qu_{\max} = 2$ and $l_{\max} = 15$.

	Adams	**BDF 1**	**BDF 2**	**BDF 3**	**BDF 4**
Total CPU time (s)	3.43×10^4	9.9×10^4	2.33×10^4	9.7×10^3	2.37×10^4
average CPU per time step (s)	2.37	4.58	19.1	7.87	11.1
average CPU for equation (40) per time step (s)	-	-	-	3.1	-
average number of iteration to solve (eq. 40)	-	-	-	7.1	-

Table 1: Summary of statistical data for different time integration schemes after 3 ns of simulated time. **(Adams)** Adams method $qu_{max} = 15$. **(BDF 1)** BDF-method with maximum integration order $qu_{max} = 3$, Krylov Subspace dimension $l_{max} = 400$. **(BDF 2)** BDF-method with $qu_{max} = 2$ and $l_{max} = 400$. **(BDF 3)** BDF-method with preconditioning, $qu_{max} = 2$ and $l_{max} = 400$. **(BDF 4)** BDF with $qu_{max} = 2$ and $l_{max} = 15$.

	Adams	**BDF 1**	**BDF 2**	**BDF 3**	**BDF 4**
average number of Krylov subspace iterations per Newton step	-	1.3	11.26	1.47	9.88
average time step (ps)	0.21	0.14	2.47	2.48	2.03
number of function evaluations	30 774	88 963	19 243	3 134	19 798
number of renormalization steps *nrn* (error indicator)	1	115	9	10	7

time of the Adams method as a function of the simulation time is linear because the time step is constant (0.21 ps) during the whole simulation. The small time step provides a high accuracy. The number of renormalization steps is $nrn = 1$. Besides the Adams method the performance of the BDF method with different options is investigated. For the BDF method the default value for the maximum integration order, $qu_{max} = 5$, leads to a very bad performance as shown in figure 3 (BDF: $qu_{max} = 5$). A restriction of the maximum integration order to 2 drastically improves the performance (Figure 3 BDF 2). Although in other micromagnetic simulations the maximum dimension of the Krylov subspace dimension l_{max} was found to be a very crucial parameter the default value, $l_{max} = 5$, leads to the same CPU consumption as $l_{max} = 400$. For $l_{max} = 15$ the average time step is smaller than for $l_{max} = 400$. Figure 4 shows a significant difference in the time step for the simulation with $l_{max} = 15$ and $l_{max} = 400$ at the end of the simulation. If the maximum number of Krylov iterations is restricted to $l_{max} = 15$ the linear system equation (33) cannot be solved with the desired accuracy. As a consequence the time step is reduced to obey equation (43). However the overall CPU time consumption for $l_{max} = 15$ is appoximately the same as for $l_{max} = 400$. In this example a large time steps with a large cost per time step ($l_{max} = 400$) gives the same amount of work as small time steps with a low cost per time step ($l_{max} = 5$). The solid line in figure 3 shows that the BDF method with proper preconditioning yields the best performance during the whole simulation. Preconditioning speeds up the simulations by factor 2.5 compared to BDF without preconditiong with the same parameters. A very time consuming part for preconditioning is the solution of the linear system (equation 40). It requires almost one half of the total CPU

time. Figure 5 shows that preconditioning drastically decreases the number of Krylov iterations which a required to solve the linear system (equation 36). Without preconditioning the number of linear iterations increases with time because the time step becomes larger when the system gets closer to equilibrium (figure 5). For a large time step the resulting linear system requires more linear iteration for its solution. For $t > 3$ ns the number of linear iterations decreases from more than 100 without preconditioning to 2 with a preconditioner. Fewer linear iteration requires fewer number of function evaluations. Thus preconditioning reduces the number of function evaluations, in this example by a factor of 6.5. The BDF method with $l_{max} = 15$, $l_{max} = 400$ and with preconditioning has comparable accuracy. The number of renormalization steps, which serves as an error indicator, are 9, 7 and 10 for $l_{max} = 15$, $l_{max} = 400$ and $l_{max} = 400$ with preconditioning, respectively. Table 1 summarizes the key performance parameters of the different integration methods.

5 Granular media

As a second test case we have calculated the hysteresis loop for a Co element with a rough surface. The surface can be seen in Figure 6. To create a rough surface the element was built up of 8 nm wide columnar grains. The basal planes of the grains are irregular, constructed from voronoi cells. These grains lead to edge irregularities of the same size. Each grain consists in average of 26 finite elements. The grain structure leads to an inhomogeneous finite element grid. Since the time step for explicit time integration scheme has to be proportional to h^2, where h is the size of the spatial grid, a inhomogeneous grid causes very small time steps. The investigated element is 400 nm long, 100 nm wide and 25 nm thick. No uniaxial anisotropy is assumed. The sponateneous polarization $J_s = 1.76$ T and the exchange constant $A = 1.3 \times 10^{-11}$ J/m. The damping constant α was set to 0.1. Figure 6 shows magnetization states during the demagnetizing process. The angle between the external field the long axis of the Co element is 1°. The letters in figure 6 identify states marked on the hysteresis loop in figure 7. The nucleation of reversed domains starts at edge irregularities.

For the comparison of differnt time integration schems we simulated the switching process under the action of an external field. The external field 1° off the easy axis and a strength of 100 kA/m is applied instantaneously to the remanent state. After 0.76 ns the normalized magnetization parallel to the external field direction becomes smaller than -0.99. Then we regard the particle to be switched.

Figure 8 shows the efficiency of different time integration schemes. The CPU time is plotted as a function of the simulated time. It clearly shows that the Adams method (A) is not suitable to solve the LLG equation for samples with granular structure. In all simulations with the BDF method the maximum integration order, $qu_{max} = 2$, was used. The BDF method shows a very bad performance if the dimension of the Krylov subspace is restricted ($l_{max} = 5$ or $l_{max} = 15$). The maximum dimension of the Krylov subspace restricts the number of iterations to solve the linear system (equation 36), which has to be solved in every Newton step. The linear iteration stops if the desired accuracy or l_{max} is reached. If l_{max} is large enough, so that the number of iterations does not reach l_{max}, the ratio nli/nni (nli = number of linear iterations, nni = number of nonlinear iterations) gives the average number of linear iterations per Newton step. From Table 2 it follows that the average number of linear iterations for the BDF method without preconditioning is 64. Figure 8 shows that the efficiency of the time integration scheme increases with increasing l_{max}. With higher dimension of the Krylov subspace, the linear system can be solved more accurately which in turn enables a larger time step. The maximum number of Krylov iterations is 162. So a further increase of $l_{max} > 162$ has no influence on the simulation. It is interesting to note that the CPU time decreases when the absolute tolerance of the time integration scheme *abstol* is enhanced from 10^{-4} (curve D in fig. 8) to 10^{-5} (E). A smaller tolerance keeps the norm of the magnetization vector constant for a longer time period. Thus less renormalization steps which also require to restart the time integration algorithm are required. Restarting the BDF method requires some precomputation steps like factorizing the matrices for the linear systems. In addition the initial time step is small.

The curve marked with (F) in figure 8 shows that the CPU time is drastically decreased if preconditioning for the time integration is used. For preconditioning the solution of equation (40) is the most time consuming part. We have tested different solvers for the solution of equation (40). The Methods of *Orthomin*, *GCR* and *MinRes* which belong to the family of non-symmetric Krylov solvers were found to be less efficient than the Bi-Conjugate Gradient method. We emphasize that also the Bi-Conjugate Gradient method has to be combined with a preconditioner in order to speed up the convergence rate. We apply the incomplete factorization technique *RILU*. The fill-in entries during the incomplete factorization process are multiplied by a factor ω before adding them to the main diagonal. The reduction of the parameter ω from the all-round choice of $\omega = 0.95$ to $\omega = 0$ is sometimes necessary to avoid convergence problems. For the investigated sample RILU preconditioner was absolutely nec-

essary to achieve convergence. Although we have tried different options of the linear solver it is still the most time consuming part in the simulations. The solution of the linear system needs almost 50% of the total CPU time. Figure 9 shows the deviation norm *DN* (equation 47) as a function of the simulated time for a calculation with and without preconditioning. In both calculations l_{max} is 400. A large deviation norm indicates a large error. Although for both simulation the same values for the error tolerances, ***abstol*** = 10^{-4} and ***reltol*** = 0, were used, the accuracy is much better if preconditioning is performed. Table 2 summarizes the statistical data after 0.76 ns of simulated time.

6 Conclusions

Both, for the soft magnetic thin film with perfect microstructure and a granular media with surface roughness the BDF method is faster than the Adams method. In both cases preconditioning speeds up the computational time. The solution of the linear system which has to be solved for preconditioning is very time consuming. Approximately 50% of the total CPU time is spent to solve the linear system. Nevertheless the overall speed up as compared to the Adams method is three orders of magnitude. The accuracy is higher with preconditioning.

Table 2: Compares statistical data for the BDF method with and without preconditioning after 0.76 ns of simulated time. The maximum integration order qu_{max} is 2. The Krylov subspace dimension l_{max} = 400 in both calculations.

	BDF 400	**BDF Precond**
Total CPU time (s)	27.8 x 10^4	1.48 x 10^4
average CPU per timestep (s)	69.4	9.69
average CPU for equation (40) per timestep (s)	-	4.75
average number of iteration within to solve (40)	-	8.4
average number of Krylov subspace iterations per Newton step	64.3	1.18
average timestep (ps)	0.18	0.50

Table 2: Compares statistical data for the BDF method with and without preconditioning after 0.76 ns of simulated time. The maximum integration order $qu_{\max}$ is 2. The Krylov subspace dimension $l_{\max}$ = 400 in both calculations.

	BDF 400	**BDF Precond**
number of function evaluations	451 937	3 475
number of renormalization steps ***nrn*** (error indicator)	216	6

References

[1] R. A. Victora, Phys. Rev. Lett. **58** (1987) 1788.

[2]M. Mansuripur, J. Appl. Phys. **63** (1988) 5809.

[3] J. G. Zhu and H. N. Bertram, J. Appl. Phys. **66** (1989) 1291.

[4] C. W. Gear, *Numerical Initial Value Problems in Ordinary Differential Equations*, Prentice-Hall, Englewood Cliffs, (1971).

[5] M Jones and J. J. Miles, JMMM **171** (1997) 190.

[6] K. M. Tako, K.M., M. A. Wongsam. and R. W. Chantrell, JMMM, **155** (1996) 40.

[7] R. L. Burden J. D. Faires, *Numerical Analysis*, sixth edition, Brooks/Cole Publ. Co., Pasific Grove, (1997).

[8] R. D. McMichael M. J. Donahue, IEEE Trans. Magn. **33** (1997) 4167.

[9] N. Hayashi, Y. Nakatani, Jpn. J. Appl. Phys. **25** (1986) 406.

[10] W. E and X.-P. Wang, SIAM Journal on Numerical Analysis **38** (2000) 1647.

[11] G. Albuquerque, J. Miltat and A Thiaville, J. Appl. Phys. **89** (2001) 6719.

[12] B. Yang and D. R. Fredkin, IEEE Trans. Magn. **34** (1998) 3842.

[13] V. Tsiantos, Ph.D. Thesis, University of Manchester, 2000.

[14] S. D. Cohen and A. C. Hindmarsh, CVODE User Guide, Lawrence Livermore National Laboratory report UCRL-MA-118618, Sept. 1994.

[15] S. D. Cohen and A. C. Hindmarsh, Computers in Physics **10** (1996) 138.

[16] V. D. Tsiantos, T. Schrefl, J. Fidler, and A. Bratsos, "Cost-Effective Way toSpeed-up Micromagnetic Simulations in Granular media," Applied Numerical Mathematics, (2001) in press.

[17] W. F. Brown, Phys. Rev. **58** (1940) 736.

[18] W. F. Brown Jr., *Micromagnetism*, New York, Interscience, 1963.

[19] C. W. Gardiner, *Handbook of Stochastic Methods*, Berlin, Springer, 1985.

[20] D.R. Fredkin and T. R. Koehler, IEEE Trans. Magn. **24** (1988) 2362.

[21]A. C. Hindmarsh and L. R. Petzold, Computers in Physics **9**, (1995)148.

[22]A. M. Bruaset, Krylov subspace iterations for sparse linear systems in Numerical Meth and Software Tools in Industrial Mathematics, Eds. Morten Daehlen and Aslak Tveito, Birkhauser, Boston, (1997) 21.

[23] D. Weller and A. Moser, IEEE Trans. Magn. **35** (1999) 4423.

Figures

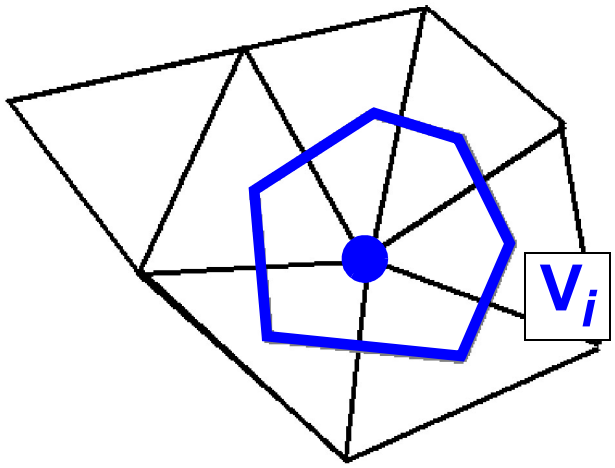

Figure 1: The volume V_i surrounding the node i shown in a 2 dimensional example.

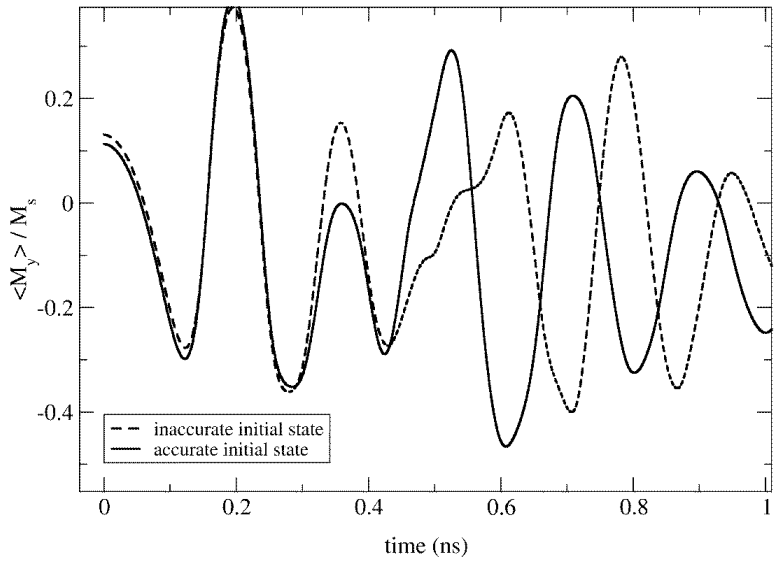

Figure 2: Time evolution of the *y*-component of the magnetic polarization of the μMAG Problem # 4. The initial state for the simulation was calculated with a stopping criterion $r_e=10^{-6}$ (solid line) and $r_e=10^{-3}$ (dashed line)

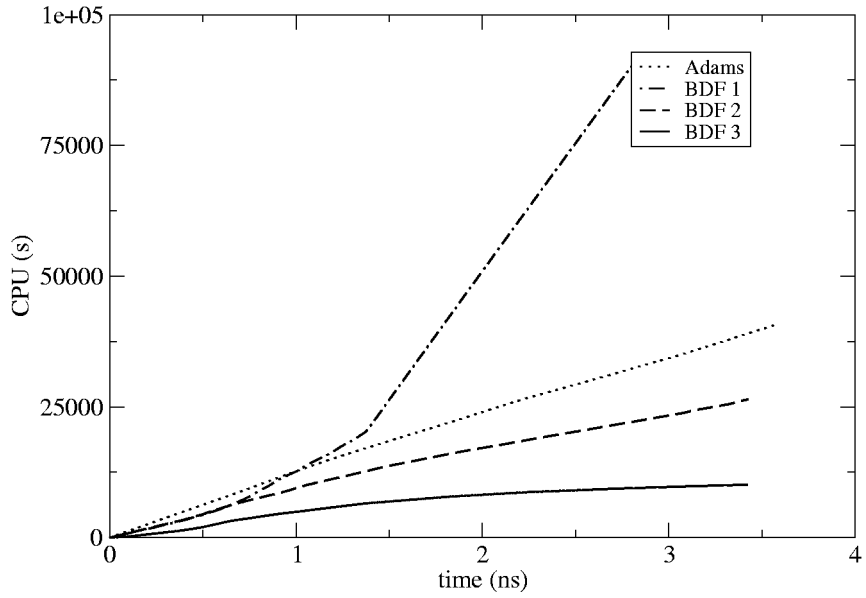

Figure 3: CPU time (Digital EV6 - 523 MHz) versus the simulation time for different time integration schemes for the μMAG Problem # 4. (Adams) Adams method; (BDF1) BDF-method with maximum integration order $qu_{\max} = 3$, Krylov Subspace dimension $l_{\max}$=400; (BDF2) BDF with $qu_{\max} = 2$ and $l_{\max} = 400$; (BDF3) BDF-method with preconditioning, $l_{\max} = 400$, $qu_{\max} = 2$.

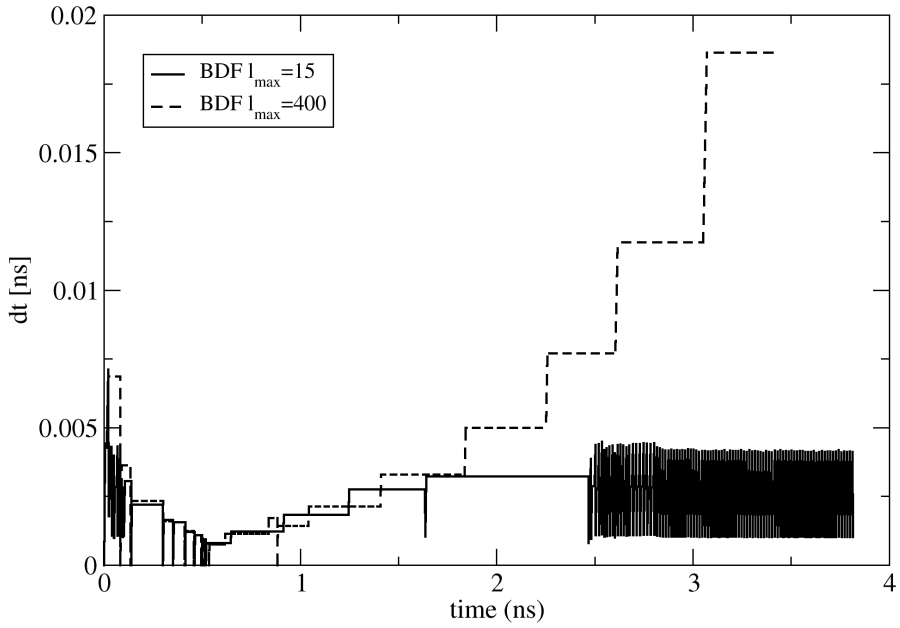

Figure 4: Comparison of the time step as a function of the simulated time of two simulations with a maximum Krylov subspace dimension $l_{\max} = 15$ and $l_{\max} = 400$. The maximum order of the time integration scheme was $qu_{\max} = 2$ in both simulations.

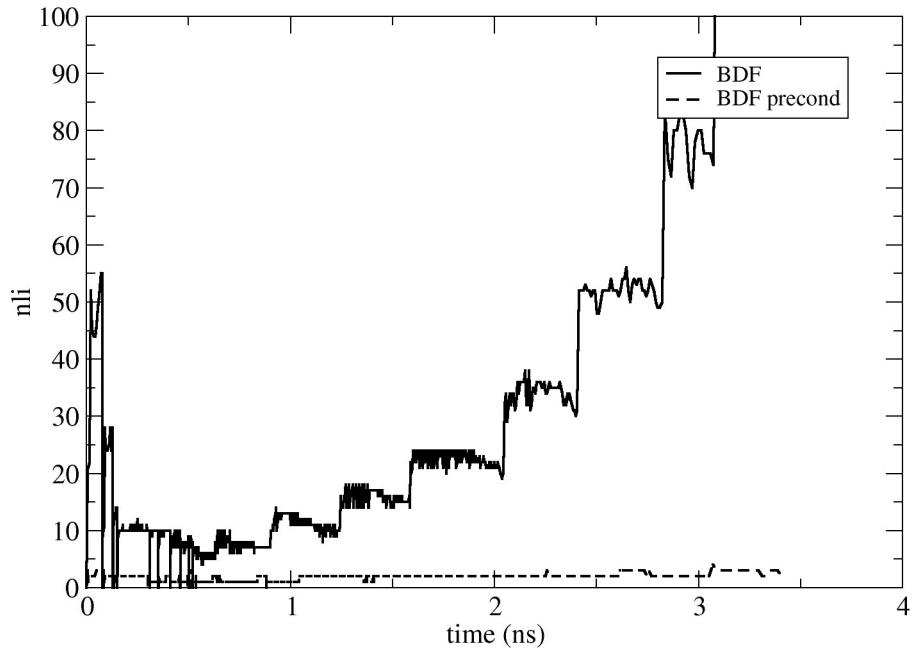

Figure 5: The number of Krylov iterations per time step for the BDF method with and without preconditioning. The dimension of the Krylov subspace $l_{\max} = 400$. The maximum integration order $qu_{\max} = 2$.

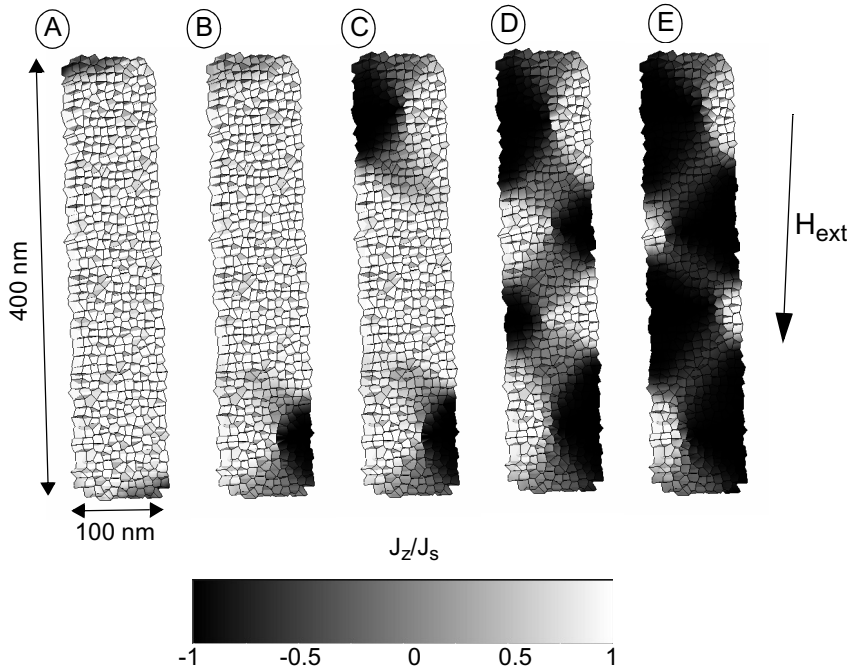

Figure 6: Equilibrium states during the demagnetization process for a Co-grain with granular structure. The film thickness is 25 nm. The letters identify the states marked on the demagnetization curve in figure 7.

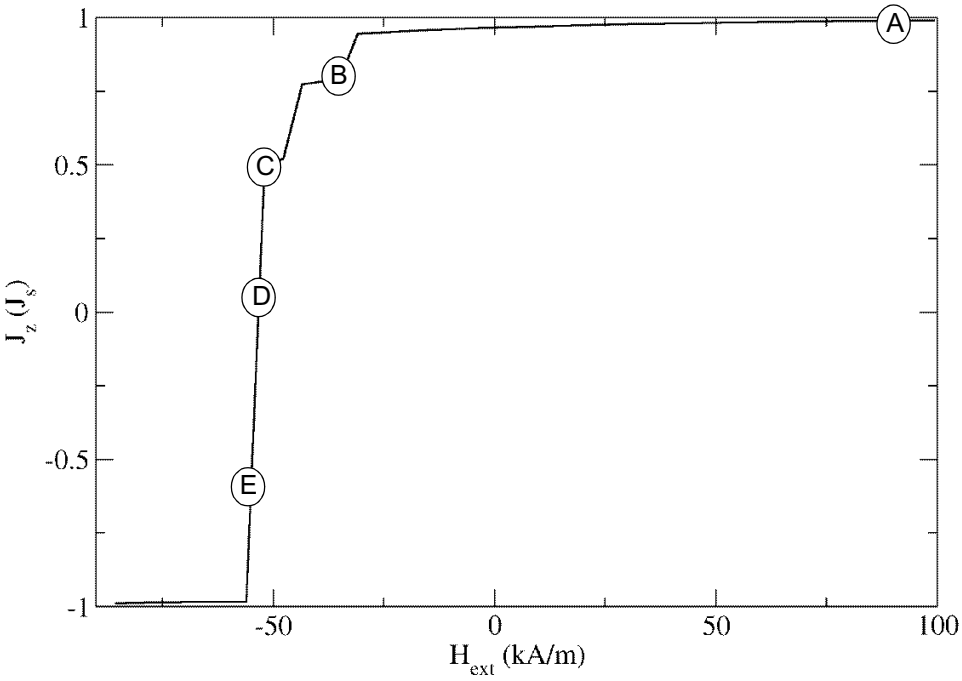

Figure 7: Hysteresis loop for the granular Co-particle. The field step is 4 kA/m.

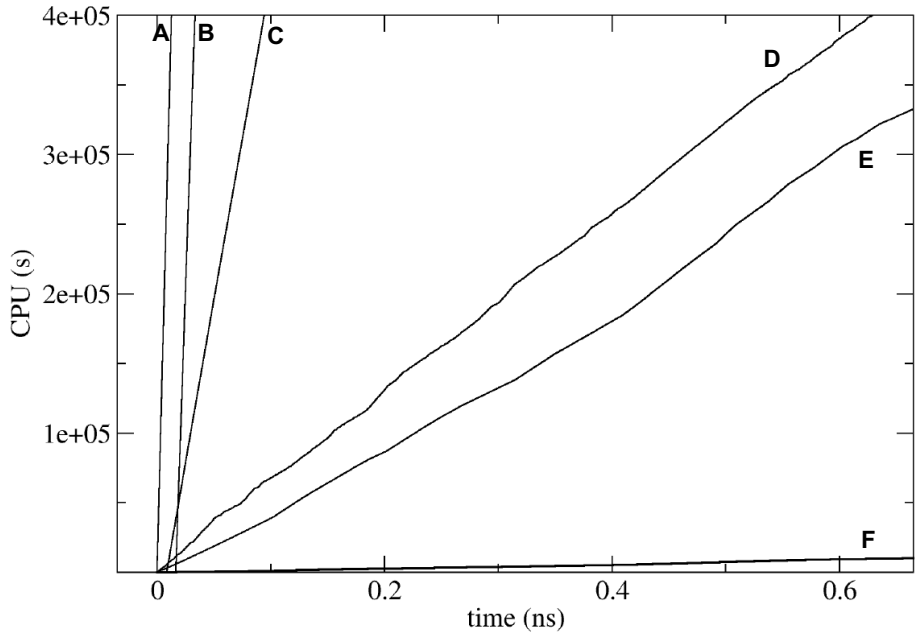

Figure 8: CPU time (Digital EV6 - 523 MHz) versus the simulated time for different time integration schemes for a Co element with a grain structure. (A) Adams method (B-D) BDF method without preconditioning for different dimension of the Krylov subspace. (B) Krylov subspace dimension $l_{max} = 5$, (C) $l_{max} = 15$, (D) $l_{max} = 400$, (E) $l_{max} = 400$ and high accuracy (*abstol* = 10^{-5}). (F) BDF-method with preconditioning, $l_{max} = 15$.

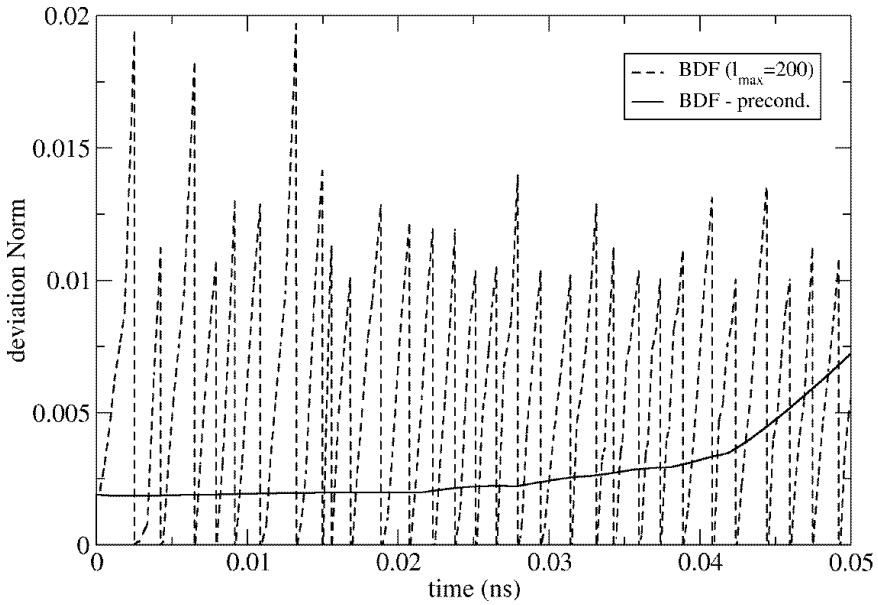

Figure 9: The deviation of the magnitude of the magnetic polarization vector from J_s indicates the quality of the time integration method. Comparison of the deviation norm *DN* for the BDF method with and without preconditioning. The maximum order of the time integration was qu_{max}=2.