TECHNICAL RESEARCH REPORT

Fast Evaluation of Demagnetizing Field in Three Dimensional Micromagnetics using Multipole Approximation

by X. Tan, J.S. Baras, P.S. Krishnaprasad

CDCSS T.R. 2000-6
(ISR T.R. 2000-19)

CENTER FOR DYNAMICS
AND CONTROL OF
SMART STRUCTURES

The Center for Dynamics and Control of Smart Structures (CDCSS) is a joint Harvard University, Boston University, University of Maryland center, supported by the Army Research Office under the ODDR&E MURI97 Program Grant No. DAAG55-97-1-0114 (through Harvard University). This document is a technical report in the CDCSS series originating at the University of Maryland.

Web site http://www.isr.umd.edu/CDCSS/cdcss.html

Fast evaluation of demagnetizing field in three dimensional micromagnetics using multipole approximation*

Xiaobo Tan, John S. Baras, and P. S. Krishnaprasad

Institute for Systems Research and
Department of Electrical and Computer Engineering
University of Maryland, College Park, MD 20742 USA

ABSTRACT

Computational micromagnetics in three dimensions is of increasing interest with the development of magnetostrictive sensors and actuators. In solving the Landau-Lifshitz-Gilbert (LLG) equation, the governing equation of magnetic dynamics for ferromagnetic materials, we need to evaluate the effective field. The effective field consists of several terms, among which the demagnetizing field is of long-range nature. Evaluating the demagnetizing field directly requires work of $O(N^2)$ for a grid of N cells and thus it is the bottleneck in computational micromagnetics. A fast hierarchical algorithm using multipole approximation is developed to evaluate the demagnetizing field. We first construct a mesh hierarchy and divide the grid into boxes of different levels. The lowest level box is the whole grid while the highest level boxes are just cells. The approximate field contribution from the cells contained in a box is characterized by the box attributes, which are obtained via multipole approximation. The algorithm computes field contributions from remote cells using attributes of appropriate boxes containing those cells, and it computes contributions from adjacent cells directly. Numerical results have shown that the algorithm requires work of $O(NlogN)$ and at the same time it achieves high accuracy. It makes micromagnetic simulation in three dimensions feasible.

Keywords: Micromagnetics, demagnetizing field, multipole approximation

1. INTRODUCTION

Computational micromagnetics in three dimensions is of increasing interest with the development of magnetostrictive sensors and actuators. The Landau-Lifshitz-Gilbert (LLG) equation is the governing equation of magnetic dynamics for ferromagnetic materials.[1] By integrating the LLG equation, we can solve for the evolution of magnetization as well as the steady state magnetization profile. This helps us understand the underlying physical principles, characterize material properties, and design and control magnetostrictive transducers.

The effective magnetic field $\mathbf{H_{eff}}$ needs to be evaluated in solving the LLG equation. $\mathbf{H_{eff}}$ is the sum of several terms: the externally applied field $\mathbf{H_{ext}}$, the anisotropy field $\mathbf{H_{anis}}$ due to the crystalline anisotropy, the exchange field $\mathbf{H_{exch}}$ due to the quantum-mechanical exchange effect between nearest neighbours, and the demagnetizing field $\mathbf{H_{demag}}$ produced by the whole magnetization distribution. The last term is non-local, because the decay of the demagnetizing field with distance is so slow that all interactions must be accounted for. Therefore evaluation of $\mathbf{H_{demag}}$ is the most time-consuming part and thus the bottleneck in computational micromagnetics. For a discretization grid of N cells, an amount of work of $O(N^2)$ is required to evaluate all pairwise interactions. In three dimensional micromagnetics, to be of physical interest, the simulation is usually involved in thousands of cells. As a result, the computation would be prohibitive if we calculate the demagnetizing field directly.

Many techniques have been proposed to speed up the evaluation of $\mathbf{H_{demag}}$. The simplest method is truncation of the interaction range.[2] Although it reduces the computation time to $O(N)$, the loss of accuracy is significant. The Fast Fourier Transform (FFT) technique is used widely and it requires work of $O(NlogN)$.[3–5] Multipole approximation is another useful method to accelerate the computation of $\mathbf{H_{demag}}$.[6,4] The general strategy of the multipole algorithm is clustering cells at different spatial scales and using multipole expansions to evaluate the

* This research was supported by the Army Research Office under the ODDR&E MURI97 Program Grant No. DAAG55-97-1-0114 to the Center for Dynamics and Control of Smart Structures (through Harvard University).

Further author information: (Send correspondence to X.T.)
E-mail: X.T.: xbtan@isr.umd.edu J.S.B.: baras@isr.umd.edu P.S.K.: krishna@isr.umd.edu

interactions between clusters that are sufficiently far away from each other. The interactions between nearby cells are calculated directly. Greengard adopted a two-pass (upward and downward) multipole algorithm in electrostatic calculations and obtained $O(N)$ complexity.[7] Following this approach, a fast algorithm fully implementing the multipole and local expansions of the field integral was shown to yield $O(N)$ computation time in 2D micromagnetics.[4] But this technique can't be generalized to 3D directly because the closed forms of multipole and local expansions in 3D do not exist. Blue and Scheinfein used only a single upward pass in multipole expansion for 2D micromagnetics, which yielded a computation time of $O(NlogN)$.[6] Direct generalization of this algorithm to 3D case is not trivial either.

Inspired by the work of Blue and Scheinfein,[6] this paper is aimed at proposing a simple but efficient hierarchical algorithm for evaluation of $\mathbf{H_{demag}}$ in 3D. We first construct a mesh hierarchy and divide the grid into boxes of different levels. The approximate field contribution from the cells contained in a box is characterized by the box attributes, which are obtained via multipole approximation. The algorithm computes field contributions from remote cells using attributes of appropriate boxes containing those cells, and it computes contributions from adjacent cells directly. The paper is organized as follows: Section 2 describes the fast algorithm in detail, Section 3 presents the numerical results and Section 4 provides the conclusions.

2. FAST HIERARCHICAL ALGORITHM USING MULTIPOLE APPROXIMATION

Assuming that a ferromagnetic body with magnetization $\mathbf{M}$ occupies a region V, the demagnetizing field at $\mathbf{r}$, $\mathbf{H_{demag}}(\mathbf{r})$, can be written down directly from magnetostatics[8]:

$$\mathbf{H_{demag}}(\mathbf{r}) = -\int_V \frac{(\mathbf{r}-\mathbf{r}')\nabla\cdot\mathbf{M}(\mathbf{r}')}{\parallel \mathbf{r}-\mathbf{r}' \parallel^3} d\tau' + \oint_{\partial V} \frac{(\mathbf{r}-\mathbf{r}')\mathbf{M}(\mathbf{r}')\cdot\mathbf{n}(\mathbf{r}')}{\parallel \mathbf{r}-\mathbf{r}' \parallel^3} ds', \tag{1}$$

where $d\tau', ds'$ are the volume element and the area element, respectively, the integrals are taken over the space variable $\mathbf{r}'$, "$\nabla\cdot$" is the divergence operator with respect to $\mathbf{r}'$, $\parallel \cdot \parallel$ is the Euclidean norm of a 3D vector, ∂V is the boundary of V, and $\mathbf{n}(\mathbf{r}')$ is the unit outward normal at $\mathbf{r}'$. It is clear from (1) that $\mathbf{H_{demag}}$ depends on the whole distribution of $\mathbf{M}$.

In this paper, we consider a ferromagnetic body with rectangular geometry. We discretize the body into cells of size d_0. The numbers of cells along the x, y, z axes are denoted by N_x, N_y and N_z, so the total number of cells $N = N_xN_yN_z$. $\mathbf{M}$ is assumed to be constant within each cell (including at the boundary).

In this section, we will first look at the demagnetizing field $\mathbf{H_d}$ produced by a cell with magnetization $\mathbf{M}$. The analytic expression for $\mathbf{H_d}$ turns out to be very complicated. We then show that $\mathbf{H_d}$ can be approximated by the field produced by a magnetic dipole with moment $\mathbf{M}d_0^3$ located at the center of the cell. Although the dipole formula looks much simpler, direct pairwise evaluation is still very time-consuming when N is large. A hierarchical algorithm incorporating multipole approximation is then used to accelerate the evaluation of $\mathbf{H_{demag}}$.

2.1. Demagnetizing Field Contribution from a Cell

We want to evaluate the demagnetizing field $\mathbf{H_d}$ at the center of cell (i, j, k) produced by cell (i', j', k') with magnetization $\mathbf{M}$. Denote the region that cell (i', j', k') occupies as Ω. Since $\mathbf{M}$ is constant within Ω, we have $\nabla\cdot\mathbf{M} = 0$ and only the second term in (1) survives. Let H_{dx}, H_{dy}, H_{dz} be the x, y, z components of $\mathbf{H_d}$, and let $n_x = i - i', n_y = j - j', n_z = k - k'$. It follows that

$$H_{dx} = \oint_{\partial\Omega} \frac{(x-x')\mathbf{M}\cdot\mathbf{n}}{[\sqrt{(x-x')^2+(y-y')^2+(z-z')^2}]^3} dx'dy'dz', \tag{2}$$

where $(x \quad y \quad z)^T = d_0(n_x \quad n_y \quad n_z)^T$, and x', y', $z' \in [-\frac{d_0}{2}, \frac{d_0}{2}]$. Carrying out the integral, we will get the analytic expression for H_{dx},

$$H_{dx} = M_x(\sum_{i=1}^{4}\arctan l_{xi} - \sum_{i=5}^{8}\arctan l_{xi}) + M_y \ln\frac{\prod_{i=1}^{4} l_{yi}}{\prod_{i=5}^{8} l_{yi}} + M_z \ln\frac{\prod_{i=1}^{4} l_{zi}}{\prod_{i=5}^{8} l_{zi}}, \tag{3}$$

where M_x, M_y, M_z are the x, y, z components of $\mathbf{M}$, and $l_{xi}, l_{yi}, l_{zi}, i = 1, \cdots 8$ are functions of n_x, n_y, n_z, as defined in Appendix A. Similar expressions can be obtained for H_{dy} and H_{dz}. Note that $\mathbf{H_d}$ does not depend on the cell size d_0; instead, it depends only on $\mathbf{M}$ and $(n_x \quad n_y \quad n_z)^T$.

2.2. Approximation to $\mathbf{H_d}$ by the Dipole Formula

Expanding

$$\frac{x-x'}{[\sqrt{(x-x')^2+(y-y')^2+(z-z')^2}]^3} \tag{4}$$

in (2), we will get

$$\frac{x}{\| \mathbf{r} \|^{\mathbf{3}}}+\frac{3x^2-\| \mathbf{r} \|^{\mathbf{2}}}{\| \mathbf{r} \|^{\mathbf{5}}}x'+\frac{3xy}{\| \mathbf{r} \|^{\mathbf{5}}}y'+\frac{3xz}{\| \mathbf{r} \|^{\mathbf{5}}}z'+O(\frac{x'^2+y'^2+z'^2}{\| \mathbf{r} \|^{\mathbf{4}}}), \tag{5}$$

where $\mathbf{r}=(x \quad y \quad z)^T$. Plugging the first order approximation in (5) rather than (4) into (2) and carrying out the integral leads to the approximation to H_{dx}, denoted as $\tilde{H}_{dx}$:

$$\tilde{H}_{dx}=\frac{[3(\mathbf{M}\cdot\mathbf{r})x-M_x \| \mathbf{r} \|^{\mathbf{2}}]d_0^3}{\| \mathbf{r} \|^5}. \tag{6}$$

Similarly, we can get $\tilde{H}_{dy}$ and $\tilde{H}_{dz}$. It's easy to see that, letting $\tilde{\mathbf{H}}_{\mathbf{d}}=(\tilde{H}_{dx} \quad \tilde{H}_{dy} \quad \tilde{H}_{dz})^T$ and $\hat{\mathbf{r}}=\mathbf{r}/\| \mathbf{r} \|$,

$$\tilde{\mathbf{H}}_{\mathbf{d}}=\frac{[3(\mathbf{M}\cdot\hat{\mathbf{r}})\hat{\mathbf{r}}-\mathbf{M}]d_0^3}{\| \mathbf{r} \|^{\mathbf{3}}}, \tag{7}$$

which is the familiar *dipole formula.*

Defining the error

$$e=\frac{\| \tilde{\mathbf{H}}_{\mathbf{d}}-\mathbf{H_d} \|}{\| \mathbf{H_d} \|}, \tag{8}$$

we expect e to be of order $O(1/(n_x^2+n_y^2+n_z^2))$ from (2), (5) and x', y', $z' \in [-\frac{d_0}{2}, \frac{d_0}{2}]$.

The dipole formula (7) is much simpler than (3), which we shall refer as the *integral formula.* But we can't be too optimistic here. When N goes large, the computation time of direct pairwise evaluation increases with $O(N^2)$. In other words, halving the cell size will multiply the computation time by 64. Even with the dipole formula, the computation may become prohibitive well before the grid is fine enough. Thus we resort to the following hierarchical algorithm.

2.3. A Hierarchical Evaluation Algorithm

As pointed out earlier, the cell size d_0 does not appear in the analytic expression of $\mathbf{H_d}$, therefore from now on we will assume unit cell size. We start with a 3D grid of N cells and embed the grid into a box of size 2^m, with the smallest possible m to enclose the grid. Mesh level 0 corresponds to the entire computational box, while mesh level $L+1$ is obtained from level L by subdividing each box into eight subboxes. Continue this process until there is at most one cell in each box. A tree structure is imposed on this mesh hierarchy, so that if B is a box at level L, the eight boxes at level $L+1$ obtained by subdividing B are considered its children.

As we will show in the next subsection, the approximate demagnetizing field contribution from cells contained in each box can be characterized with two attributes of that box. Assume that we have got these attributes for all boxes and pick a threshold value β. The following steps can be taken to evaluate the field at the center of each cell in the grid:

- Step 1. Put all boxes of level 1 on a stack (since the box of level 0 will contain the current cell);

- Step 2. If the stack is empty, go to Step 6; otherwise

- Step 3. Take the last box on the stack. Denote the distance between the center of the box and the field point (i.e., the center of the current cell) as d, and the size of the box as a. Let $\rho=\frac{\sqrt{2}}{2}\frac{a}{d}$. If $\rho<\beta$, which implies that the box is far enough away, compute the contributions from all cells contained in that box using only the attributes of the box and remove the box from the stack; otherwise

- Step 4. If the box contains only one cell, compute the contribution of the cell directly using the integral formula and remove the box from the stack; otherwise
- Step 5. Remove the box from the stack, put all its children boxes on the stack, and go to Step 2;
- Step 6. If the field evaluation at centers of all cells is completed, go to Step 7; otherwise go to Step 1 and start field evaluation for the next cell.
- Step 7. End

Figure 1 illustrates the idea for a 2D 8 by 8 grid, but the same idea applies in a 3D grid. To evaluate the field at the center of cell A, we need only calculate the contributions from the solid line boxes. Since in a N-cell grid, the field evaluation at one point involves $O(logN)$ boxes, the work to evaluate the field at centers of all N cells is $O(NlogN)$.

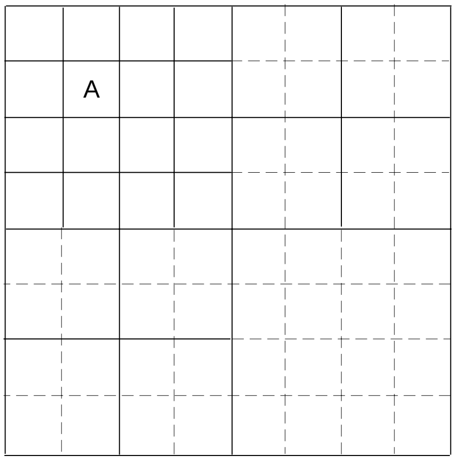

Figure 1. An 8×8 grid of cells divided into boxes suitable for evaluation of the demagnetizing field at the center of cell A.

2.4. Multipole Approximation to the Dipole Formula

We now derive the attributes for each box. Assume that a box is centered at the origin and that a cell with magnetization $\mathbf{M}$ is contained in the box. Let the center of the cell be $\mathbf{r}_0$. From (7), the demagnetizing field at $\mathbf{r}$, produced by the cell, is approximately

$$\tilde{\mathbf{H}}_{\mathbf{d}}(\mathbf{r}-\mathbf{r}_0) = \frac{3[\mathbf{M}\cdot(\mathbf{r}-\mathbf{r}_0)](\mathbf{r}-\mathbf{r}_0) - \parallel \mathbf{r}-\mathbf{r}_0 \parallel^2 \mathbf{M}}{\parallel \mathbf{r}-\mathbf{r}_0 \parallel^5}. \tag{9}$$

Taking Taylor's series expansion to the first order, we have

$$\begin{aligned} \tilde{\mathbf{H}}_{\mathbf{d}}(\mathbf{r}-\mathbf{r}_0) &\approx \tilde{\mathbf{H}}_{\mathbf{d}}(\mathbf{r}) + \frac{\mathbf{d}\tilde{\mathbf{H}}_{\mathbf{d}}(\mathbf{r})}{\mathbf{dr}}(-\mathbf{r}_0) && (10) \\ &= \frac{3(\mathbf{M}\cdot\hat{\mathbf{r}})\hat{\mathbf{r}} - \mathbf{M}}{\parallel \mathbf{r} \parallel^3} - \frac{3[\hat{\mathbf{r}}\mathbf{M}^{\mathbf{T}} + \mathbf{M}\hat{\mathbf{r}}^{\mathbf{T}} + (\mathbf{M}\cdot\hat{\mathbf{r}})(\mathbf{I} - 5\hat{\mathbf{r}}\hat{\mathbf{r}}^T)]\mathbf{r}_0}{\parallel \mathbf{r} \parallel^4}, && (11) \end{aligned}$$

where $\mathbf{I}$ is the identity matrix. If we define the approximation error in (10) in a similar way as in (8), it is expected to be $O(\parallel \mathbf{r}_0 \parallel^2 / \parallel \mathbf{r} \parallel^2)$. Therefore when the field point is far enough away from the center of a box (comparing with the size of the box), the total approximation error incurred in (5) and (10) will be small enough. The first term of (11) is linear in $\mathbf{M}$, and the second term is linear in elements of the matrix $\mathbf{M}\mathbf{r}_0^T$. Having noticed that $\mathbf{M}$ and $\mathbf{r}_0$ are all the useful information associated with the cell, we are now ready for deriving the attributes for each box.

Suppose a box b has k cells, each with center position $\mathbf{r}_0(s)$ and magnetization $\mathbf{M}(s)$, $s = 1, \cdots, k$. There are two attributes associated with box b: $\mathbf{A}_1(b) = \sum_{s=1}^{k} \mathbf{M}(s)$ and $\mathbf{A}_2(b) = \sum_{s=1}^{k} \mathbf{M}(s)\mathbf{r}_0^T(s)$.

Let $D_{ij}, (i, j = 1, 2, 3)$ be the i-th row j-th column element of $d\tilde{\mathbf{H}}_{\mathbf{d}}(\mathbf{r})/d\mathbf{r}$, and write D_{ij} as $\theta_{ij}^T\mathbf{M}$. Note that θ_{ij} is a function of only $\mathbf{r}$. Let $\mathbf{\Theta}_i = (\theta_{i1}, \theta_{i2}, \theta_{i3})$, $i = 1, 2, 3$, and let

$$\mathbf{h}(b) = [trace(\mathbf{\Theta}_1^T \mathbf{A}_2(b)), trace(\mathbf{\Theta}_2^T \mathbf{A}_2(b)), trace(\mathbf{\Theta}_3^T \mathbf{A}_2(b))]^T. \tag{12}$$

Then it's easy to verify that the demagnetizing field at $\mathbf{r}$ produced by all cells inside box b is approximately

$$\sum_{s=1}^{k} \tilde{\mathbf{H}}_{\mathbf{d}}(\mathbf{r} - \mathbf{r}_0(s)) \approx \frac{3[\mathbf{A}_1(b) \cdot \hat{\mathbf{r}}]\hat{\mathbf{r}} - \mathbf{A}_1(b)}{\| \mathbf{r} \|^3} - \mathbf{h}(b). \tag{13}$$

The work to compute the attributes for all boxes is $O(NlogN)$ since each cell is involved in $O(logN)$ boxes. After we obtain the attributes, the field evaluation at centers of N cells is of $O(NlogN)$ as mentioned in Subsection 2.3. Thus the total work of the algorithm is still $O(NlogN)$.

3. NUMERICAL RESULTS

Three methods of evaluating the demagnetizing field are compared: direct pairwise evaluation using the integral formula (Integral Algorithm, abbreviated as IA), direct pairwise evaluation using the dipole formula (Dipole Algorithm, abbreviated as DA), and the fast hierarchical algorithm (abbreviated as FA). We take the result of IA as the true value, and calculate the root mean square (RMS) errors for DA and FA. Let $\mathbf{H}_{\mathbf{IA}}^{i,j,k}, \mathbf{H}_{\mathbf{DA}}^{i,j,k}, \mathbf{H}_{\mathbf{FA}}^{i,j,k}$ be the demagnetizing field at the center of cell (i, j, k) calculated using IA, DA, FA, respectively. Then the RMS errors for DA and FA are defined as

$$e_{RMS}^{DA} = \sqrt{\frac{\sum_{i,j,k} \| \mathbf{H}_{\mathbf{DA}}^{i,j,k} - \mathbf{H}_{\mathbf{IA}}^{i,j,k} \|^{\mathbf{2}}}{\sum_{i,j,k} \| \mathbf{H}_{\mathbf{IA}}^{i,j,k} \|^{\mathbf{2}}}},$$

$$e_{RMS}^{FA} = \sqrt{\frac{\sum_{i,j,k} \| \mathbf{H}_{\mathbf{FA}}^{i,j,k} - \mathbf{H}_{\mathbf{IA}}^{i,j,k} \|^{\mathbf{2}}}{\sum_{i,j,k} \| \mathbf{H}_{\mathbf{IA}}^{i,j,k} \|^{\mathbf{2}}}}.$$

Table 1 compares IA, DA and FA for an 8 by 8 by 16 grid. All the numerical experiments reported in this paper were done on Ultra 10 workstations of Sun Microsystems.

Table 1. Results of IA, DA and FA for an 8 by 8 by 16 grid. β is the threshold value in FA.

Algorithm	Time (sec.)	RMS error
IA	89	0
DA	6	1.5×10^{-1}
FA, $\beta = 0.2$	17	5.9×10^{-4}
FA, $\beta = 0.4$	5	1.7×10^{-2}
FA, $\beta = 0.6$	2	6.6×10^{-2}
FA, $\beta = 0.7$	1	1.4×10^{-1}

From Table 1, we can control the accuracy of FA by varying the threshold β. The algorithm can be as accurate as desired. FA outperforms DA in both computing time and RMS error within a wide range of β.

Table 2 compares IA and FA for grids of different sizes.

From Table 2, it is clear that the computation time of IA increases with $O(N^2)$, while that of FA increases with $O(N \log N)$. When N goes bigger, we'll get more out of the fast algorithm. Note that the error in Table 2 is acceptable in consideration of the error incurred by the discretization.

Table 2. Results of DA and FA for different grid sizes. $\beta = 0.4$.

Grid size	Time (IA) (sec.)	Time (FA) (sec.)	RMS error of FA
6 by 6 by 12	15	2	1.3×10^{-2}
8 by 8 by 16	89	5	1.5×10^{-2}
12 by 12 by 24	928	28	1.7×10^{-2}
16 by 16 by 32	5,612	80	1.7×10^{-2}

The fast algorithm has been used in a 3D micromagnetics and magnetostriction computation program.[9] Figure 2 shows the $H - M$ hysteresis curve of a 3D $2 \times 2 \times 5$ grid computed using the program, where H is the external field and M is the bulk magnetization of the ferromagnetic body.

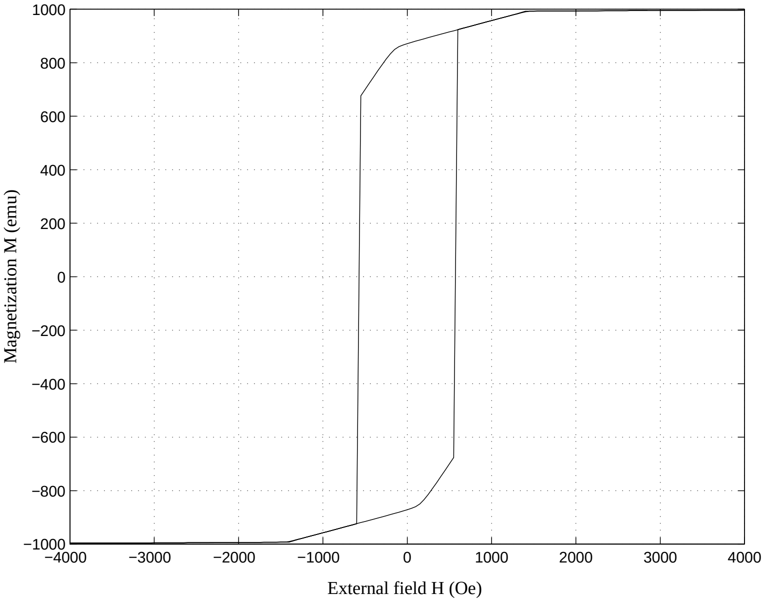

Figure 2. Hysteresis curve of a $2 \times 2 \times 5$ grid computed using the 3D micromagnetic program.

4. CONCLUDING REMARKS

A fast algorithm using multipole approximation is presented. It's easy to implement and yields computation time of $O(NlogN)$. By choosing the appropriate threshold value, we can make the algorithm as accurate as desired while maintaining acceptable computing speed. It makes micromagnetic computation in three dimensions feasible.

APPENDIX A. EXPRESSIONS FOR $l_{xi}, l_{yi}, l_{zi}, i = 1 \cdots 8$, in (3)

$$l_{x1} = \frac{(n_y + \frac{1}{2})(n_z - \frac{1}{2})}{(n_x + \frac{1}{2})\sqrt{(n_x + \frac{1}{2})^2 + (n_y + \frac{1}{2})^2 + (n_z - \frac{1}{2})^2}}, \quad l_{x2} = \frac{(n_y - \frac{1}{2})(n_z + \frac{1}{2})}{(n_x + \frac{1}{2})\sqrt{(n_x + \frac{1}{2})^2 + (n_y - \frac{1}{2})^2 + (n_z + \frac{1}{2})^2}},$$

$$l_{x3} = \frac{(n_y + \frac{1}{2})(n_z + \frac{1}{2})}{(n_x - \frac{1}{2})\sqrt{(n_x - \frac{1}{2})^2 + (n_y + \frac{1}{2})^2 + (n_z + \frac{1}{2})^2}}, \quad l_{x4} = \frac{(n_y - \frac{1}{2})(n_z - \frac{1}{2})}{(n_x - \frac{1}{2})\sqrt{(n_x - \frac{1}{2})^2 + (n_y - \frac{1}{2})^2 + (n_z - \frac{1}{2})^2}},$$

$$l_{x5} = \frac{(n_y + \frac{1}{2})(n_z + \frac{1}{2})}{(n_x + \frac{1}{2})\sqrt{(n_x + \frac{1}{2})^2 + (n_y + \frac{1}{2})^2 + (n_z + \frac{1}{2})^2}}, \quad l_{x6} = \frac{(n_y - \frac{1}{2})(n_z - \frac{1}{2})}{(n_x + \frac{1}{2})\sqrt{(n_x + \frac{1}{2})^2 + (n_y - \frac{1}{2})^2 + (n_z - \frac{1}{2})^2}},$$

$$
\begin{aligned}
l_{x7} &= \frac{(n_y+\tfrac{1}{2})(n_z-\tfrac{1}{2})}{(n_x-\tfrac{1}{2})\sqrt{(n_x-\tfrac{1}{2})^2+(n_y+\tfrac{1}{2})^2+(n_z-\tfrac{1}{2})^2}}, &\quad l_{x8} &= \frac{(n_y-\tfrac{1}{2})(n_z+\tfrac{1}{2})}{(n_x-\tfrac{1}{2})\sqrt{(n_x-\tfrac{1}{2})^2+(n_y-\tfrac{1}{2})^2+(n_z+\tfrac{1}{2})^2}},\\
l_{y1} &= n_z+\tfrac{1}{2}+\sqrt{(n_x+\tfrac{1}{2})^2+(n_y+\tfrac{1}{2})^2+(n_z+\tfrac{1}{2})^2}, &\quad l_{y2} &= n_z+\tfrac{1}{2}+\sqrt{(n_x-\tfrac{1}{2})^2+(n_y-\tfrac{1}{2})^2+(n_z+\tfrac{1}{2})^2},\\
l_{y3} &= n_z-\tfrac{1}{2}+\sqrt{(n_x-\tfrac{1}{2})^2+(n_y+\tfrac{1}{2})^2+(n_z-\tfrac{1}{2})^2}, &\quad l_{y4} &= n_z-\tfrac{1}{2}+\sqrt{(n_x+\tfrac{1}{2})^2+(n_y-\tfrac{1}{2})^2+(n_z-\tfrac{1}{2})^2},\\
l_{y5} &= n_z+\tfrac{1}{2}+\sqrt{(n_x-\tfrac{1}{2})^2+(n_y+\tfrac{1}{2})^2+(n_z+\tfrac{1}{2})^2}, &\quad l_{y6} &= n_z+\tfrac{1}{2}+\sqrt{(n_x+\tfrac{1}{2})^2+(n_y-\tfrac{1}{2})^2+(n_z+\tfrac{1}{2})^2},\\
l_{y7} &= n_z-\tfrac{1}{2}+\sqrt{(n_x+\tfrac{1}{2})^2+(n_y+\tfrac{1}{2})^2+(n_z-\tfrac{1}{2})^2}, &\quad l_{y8} &= n_z-\tfrac{1}{2}+\sqrt{(n_x-\tfrac{1}{2})^2+(n_y-\tfrac{1}{2})^2+(n_z-\tfrac{1}{2})^2},\\
l_{z1} &= n_y+\tfrac{1}{2}+\sqrt{(n_x+\tfrac{1}{2})^2+(n_y+\tfrac{1}{2})^2+(n_z+\tfrac{1}{2})^2}, &\quad l_{z2} &= n_y+\tfrac{1}{2}+\sqrt{(n_x-\tfrac{1}{2})^2+(n_y+\tfrac{1}{2})^2+(n_z-\tfrac{1}{2})^2},\\
l_{z3} &= n_y-\tfrac{1}{2}+\sqrt{(n_x+\tfrac{1}{2})^2+(n_y-\tfrac{1}{2})^2+(n_z-\tfrac{1}{2})^2}, &\quad l_{z4} &= n_y-\tfrac{1}{2}+\sqrt{(n_x-\tfrac{1}{2})^2-(n_y-\tfrac{1}{2})^2+(n_z-\tfrac{1}{2})^2},\\
l_{z5} &= n_y+\tfrac{1}{2}+\sqrt{(n_x-\tfrac{1}{2})^2+(n_y+\tfrac{1}{2})^2+(n_z+\tfrac{1}{2})^2}, &\quad l_{z6} &= n_y+\tfrac{1}{2}+\sqrt{(n_x+\tfrac{1}{2})^2+(n_y+\tfrac{1}{2})^2+(n_z-\tfrac{1}{2})^2},\\
l_{z7} &= n_y-\tfrac{1}{2}+\sqrt{(n_x+\tfrac{1}{2})^2+(n_y-\tfrac{1}{2})^2+(n_z+\tfrac{1}{2})^2}, &\quad l_{z8} &= n_y-\tfrac{1}{2}+\sqrt{(n_x-\tfrac{1}{2})^2+(n_y-\tfrac{1}{2})^2+(n_z-\tfrac{1}{2})^2}.
\end{aligned}
$$

REFERENCES

1. W. F. Brown, Jr., *Micromagnetics*, John Wiley & Sons, New York, 1963.
2. J. Zhu and H. N. Bertram, “Micromagnetic studies of thin metallic films,” *J. Appl. Phys.* **63**(8), pp. 1788–1791, 1987.
3. D. V. Berkov, K.Ramstöck, and A. Hubert, “Solving micromagnetic problems: towards an optimal numerical method,” *Phys. Stat. Sol. (a)* **137**(1), pp. 207–225, 1993.
4. S. W. Yuan and H. N. Bertram, “Fast adaptive algorithms for micromagnetics,” *IEEE Trans. Magn.* **28**(5), pp. 2031–2035, 1992.
5. M. Mansuripur and R. Giles, “Demagnetizing field computation for dynamic simulation of the magnetization reversal process,” *IEEE Trans. Magn.* **24**(6), pp. 2326–2328, 1988.
6. J. L. Blue and M. R. Scheinfein, “Using multipoles decreases computation time for magnetostatic self-energy,” *IEEE Trans. Magn.* **27**(6), pp. 4778–4780, 1991.
7. L. Greengard, *The rapid evaluation of potential fields in particle systems*, the MIT press, Cambridge, Massachusetts, 1988.
8. W. F. Brown, Jr., *Magnetoelastic interactions*, Springer-Verlag, Berlin, New York, 1966.
9. X. Tan, J. S. Baras, and P. S. Krishnaprasad, “Computational micromagnetics for magnetostrictive actuators,” *Proc. SPIE* **3984**, 2000.