Fast Multipole Algorithm For The Evaluation Of Magnetostatic Fields In 3D Ferromagnetic Samples

Ben Van de Wiele[1], **Femke Olyslager**[2], **and Luc Dupré**[1]

[1] Department of Electrical Energy, Systems and Automation,
Ghent University, Sint-Pietersnieuwstraat 41, B-9000 Ghent, Belgium
ben.vandewiele@ugent.be, luc.dupre@ugent.be

[2] Department of Information Technology,
Ghent University, Sint-Pietersnieuwstraat 41, B-9000 Ghent, Belgium
femke.olyslager@ugent.be

Abstract: The simulation of magnetization dynamics in ferromagnetic material samples is an important research area with increasing application possibilities, for instance to magnetic non destructive evaluation. In these simulations, sample dimensions are limited by the numerous evaluations of the magnetostatic field generated by the magnetic sample. This contribution shows how the fast multipole method (FMM) for static electromagnetic problems can be adopted for the evaluation of the magnetostatic field in 3D ferromagnetic samples. This FMM approach is compared to the widely used algorithm that uses Fast Fourier Transforms (FFT) for the evaluation of the magnetostatic field. It is shown that a basic FMM implementation for the evaluation of the magnetostatic field has already very good specifications: it has a much lower memory consumption, the CPU time is only a small factor higher compared with the FFT computations and it has a good precision. Furthermore, it is outlined how a more complex FMM algorithm can result in an execution time that outperforms the FFT method.

Keywords: Non destructive evaluation, Magnetostatic field and Fast multipole method.

1. Introduction

Since long research has been performed on the magnetization dynamics in ferromagnetic materials (see e.g. [1]). This ongoing research has made it possible to understand the magnetic dynamics at a microscopic time and space scale and hence to miniaturize magnetic storage entities and to develop fast writing processes. This research is based on the micromagnetic theory [2].

The large computer resources nowadays make it possible to apply the micromagnetic theory also on much larger ferromagnetic bodies. This is an attractive prospect, since the micromagnetic theory is able to simulate the influence of the materials microstructure on the macroscopic magnetic properties (e.g. the hysteresis loop). Hence, by applying the micromagnetic theory on large magnetic samples, one can investigate the relations between the distinct microstructural parameters of the material (grain size, lattice orientation, defects, etc.) and the macroscopic magnetic properties. When these relationships are found, variations in the macroscopic magnetic properties can evidence certain –possibly dangerous– variations in the microstructure of the considered work piece. Since magnetic measurements are very easy to perform, this can lead to an easy non destructive magnetic evaluation technique for ferromagnetic work pieces that e.g. experience cyclic loading.

In the micromagnetic simulations one or more evaluations of the magnetostatic field generated by the ferromagnetic sample itself have to be performed every time step [3]. Only the development of fast and memory efficient techniques for the evaluation of these magnetostatic fields can make it possible to apply the micromagnetic theory on the intended large samples.

When not saturated, the ferromagnetic samples are not homogeneously magnetized, domain structures depending on the external applied magnetic field and the microstructure are present. The magnetostatic field $\mathbf{H}_{ms}$ satisfying $\nabla \cdot \mathbf{H}_{ms} = -\nabla \cdot \mathbf{M}$ and $\nabla \times \mathbf{H}_{ms} = 0$, with $\mathbf{M}$ the local magnetization, has to be computed in every point of the sample. This can be done using

$$\mathbf{H}_{ms}(\mathbf{r}) = -\frac{1}{4\pi} \int\limits_{V'} \nabla\nabla \frac{1}{|\mathbf{r}-\mathbf{r}'|} \cdot \mathbf{M}(\mathbf{r}')\,\mathrm{d}\mathbf{r}'. \tag{1}$$

When the sample volume V is discretized in N finite difference (FD) cells, the classical evaluation of $\mathbf{H}_{ms}$ in the N FD cells due to the magnetizations in each FD cell takes $\mathcal{O}(N^2)$ computations. In this contribution an FMM algorithm is presented that reduces this computational cost to $\mathcal{O}(N \log N)$.

2. Geometry Description

The 3D sample is divided into cubical FD cells using a tree structure typical for FMM algorithms [4]. The total sample is enclosed by a cubic box. On a next level (*level one*) this box is divided into 8 smaller cubical boxes with identical volumes. This division is performed recursively until the boxes on the lowest level (level LEV) in the tree have the desired dimensions. Consequently, the number of boxes on a level L is 8^L. In this implementation of the FMM algorithm the boxes on the highest level LEV are called basis boxes. Each basis box itself is further subdivided in cubical FD cells. The number of FD cells in a basis box is 8^{lev} i.e. 2^{lev} FD cells in each dimension. Hence, the total geometry contains $8^{LEV+lev}$ FD cells.

In the FMM theory according to [4] a distinction is made between boxes that are *well separated* from each other and boxes that are *near* to each other. Boxes that are well separated from each other interact via *far field computations.* Neighboring basis boxes interact with each other via *near field computations.* Depending on the difference in computation time of the far field computations and the near field computations an optimal tree has to be constructed. Indeed, for a geometry with 8^{tot_lev} FD cells different choices of LEV and lev are possible such that $LEV + lev = tot\_lev$.

3. Far Field Computations

In the far field computations, groups of basis boxes *well separated* from the point $\mathbf{r}$ in (1) are combined: the magnetostatic field generated by the group is described by its multipole (MP) expansion. By translating these MP expansions, larger groups on lower levels are formed, all with their own MP expansion. To compute the magnetostatic field in a group, the outgoing MP expansions from well separated groups are combined: the magnetostatic field generated by well separated groups, described by their MP expansions is translated into a local expansion valid in the considered group. These local expansions are translated to local expansions valid in the basis boxes. An extended discussion on the translation of MP and local expansions can be found in [4].

The basis of the manipulations described above are the MP expansions of the homogeneously magnetized FD cells, which will be derived now. When the vectors $\mathbf{r}$ and $\mathbf{r}'$ are defined by the spherical coordinates (ρ, α, β) and (r, θ, ϕ) respectively, the kernel $1/|\mathbf{r}-\mathbf{r}'|$ can be written in an expansion of spherical harmonics, using the spherical harmonic addition theorem for Legendre polynomials [5]. Hence the magnetostatic field can be written as

$$\mathbf{H}_{ms}(\mathbf{r}) = \frac{1}{4\pi}\nabla \sum_{n=0}^{\infty}\sum_{m=-n}^{n}\int_{V'} \nabla' \rho^n Y_n^{-m}(\alpha,\beta)\cdot \mathbf{M}'\,\mathrm{d}V' \frac{Y_n^m(\theta,\phi)}{r^{n+1}}. \tag{2}$$

With $\rho < r$ and the spherical harmonics $Y_n^m(\theta,\phi)$ defined as in [5]. Equation (2) defines the MP expansion coefficients O_n^m of a FD cell

$$O_n^m = \int_V \nabla \rho^n Y_n^{-m}(\alpha,\beta)\,\mathrm{d}V\cdot \mathbf{M} = \mathbf{M}\cdot \int_{\partial V} \rho^n Y_n^{-m}(\alpha,\beta)\mathbf{u}\,\mathrm{d}S. \tag{3}$$

It is the intention to develop an algorithm that allows to compute the magnetostatic fields multiple times for different inputs $\mathbf{M}$ in a fast way. Hence, it is beneficial to perform generic computations as much as possible during the set up phase of the algorithm in order to avoid repeating the same computations. Since all FD cells have the same size, the surface integrals in (3) can be computed in the set up phase for the different edges of the FD cell. In that way the computation of the MP expansions O_n^m reduces to the linear combination $O_n^m = F_{n,x}^m M_x + F_{n,y}^m M_y + F_{n,z}^m M_z$.
The MP expansion of a basis box P_i^j is then computed by translating the MP expansions of its FD cells ($q = 1\ldots 8^{lev}$) to the center of the considered basis box with a translation operator $T_{ij,nm}$ different for each the FD cell q.

$$\begin{aligned} P_i^j &= \sum_{q=1}^{8^{lev}}\sum_n\sum_m T_{ij,nm}^q (F_{n,x}^m M_x^q + F_{n,y}^m M_y^q + F_{n,z}^m M_z^q) \\ &= \sum_{q=1}^{8^{lev}}\left(M_x^q \sum_n\sum_m T_{ij,nm}^q F_{n,x}^m + M_y^q \sum_n\sum_m T_{ij,nm}^q F_{n,y}^m + M_z^q \sum_n\sum_m T_{ij,nm}^q F_{n,z}^m\right). \end{aligned} \tag{4}$$

The quantities $\sum\sum T_{ij,nm}^q F_{n,x}^m$, $\sum\sum T_{ij,nm}^q F_{n,y}^m$ and $\sum\sum T_{ij,nm}^q F_{n,z}^m$ are computed in the set up phase of the algorithm. Hence the computation of the MP expansion of a basis box reduces to a simple linear combination of magnetization components M_x^q, M_y^q and M_z^q of the different FD cells. The number of MP expansion coefficients will be truncated to degrees $n < p = 6$.

The MP expansions of the basis boxes are now translated through the FMM tree as explained in [4]. To improve the CPU time, symmetry properties of the spherical harmonics are exploited and the translations of MP expansion to local expansions are accelerated using FFTs as outlined in [6]. Now the magnetostatic field values in every FD cell of a basis box are deduced from the local expansion belonging to the considered basis box. The field in a FD difference cell with coordinates (r,θ,ϕ) with respect to the center of the basis box with local expansion L_n^m is given by [4]

$$\mathbf{H}_{ms} = \frac{1}{4\pi}\sum_{n=0}^{p-1}\sum_{m=-n}^{n} L_n^m \nabla r^n Y_n^m(\theta,\phi). \tag{5}$$

In matrix notation this local to field operator can be written as: $H_{i,ms} = \sum_n\sum_m G_{i,n}^m L_n^m$ ($i = x,y,z$). The matrix elements $G_{i,n}^m$ are also computed during the set up phase of the algorithm. Moreover, when symmetry properties are exploited, the number of summations can be reduced:

$$H_{i,ms} = \sum_{n=0}^{p-1} \left[\mathcal{R}e(G^0_{i,n})\mathcal{R}e(L^0_n) + 2\sum_{m=1}^{n} \left\{ \mathcal{R}e(G^m_{i,n})\mathcal{R}e(L^m_n) - \mathcal{I}m(G^m_{i,n})\mathcal{I}m(L^m_n) \right\} \right]. \qquad (6)$$

4. Near Field Computations

To compute the total magnetostatic field in a FD cell, also the contributions of the FD cells in the 26 neighboring basis boxes and the FD cells in the considered basis box itself have to be added to the far field. For the near field computations, the expression (1) is written as

$$\mathbf{H}_{ms}(\mathbf{r}_i) = -\frac{1}{4\pi}\sum_{j=1}^{n}\int_{\partial V} \frac{(\mathbf{r}_i - \mathbf{r}_j + \boldsymbol{\rho})\cdot \mathbf{M}_j}{|\mathbf{r}_i - \mathbf{r}_j + \boldsymbol{\rho}|^3}\,\mathbf{u}_S\,\mathrm{d}\boldsymbol{\rho}. \qquad (7)$$

Each term in the summation is a contribution to the magnetostatic field of a FD cell situated in a neighboring basis box or in the same basis box. The integral is a surface integral over the 6 edges of the j^{th} FD cell. Since a classical computation scales $\mathcal{O}(n^2)$, the near field computations are accelerated exploiting the convolution structure $\mathbf{H}_{ms}(\mathbf{r}) = \mathbf{g}(\mathbf{r}) \star \mathbf{M}(\mathbf{r})$ of expression (7) with

$$\mathbf{g}(\mathbf{r}) = -\frac{1}{4\pi}\int_{\partial V} \frac{(\mathbf{r} + \boldsymbol{\rho})}{|\mathbf{r} + \boldsymbol{\rho}|^3}\,\mathbf{u}_S\,\mathrm{d}\boldsymbol{\rho}. \qquad (8)$$

The elements of this symmetrical Greens tensor contain all the geometrical information. The near field contributions for all the FD cells in a considered basis box are now computed at once by

1. Assembling and zero padding the magnetization data.
2. Fourier transforming the input from step 1.
3. Performing the point wise products of the Fourier transformed data from step 2 with the Fourier transformed Greens function tensor $\mathbf{g}(\mathbf{r})$.
4. Backward Fourier transforming the result from step 3.
5. Selecting the magnetostatic fields from the considered basis box from the result of step 4.

The Fourier transformed Greens tensor elements are computed in the set up phase of the algorithm.

5. Evaluation Of The Algorithm

The properties of the FMM algorithm are compared with a traditionally used FFT algorithm for samples of different sizes in a micromagnetic equilibrium state. For a geometry with $N = 8^{tot\_lev}$ FD cells different parameters LEV an lev can be combined. The optimal size of the basis boxes depends on the CPU time of the algorithm. Table 1 shows the CPU time for the evaluation of the magnetostatic fields for the different sample dimensions. Between brackets, the number of levels in the far field computations, LEV, and the number of levels in the near field computations, lev, is mentioned. The last row shows the CPU times of the FFT algorithm. For all sample dimensions the simulations with $lev = 3$ need the least execution time. Hence, the optimal size of the basis boxes is $8 \times 8 \times 8$ FD cells. When compared with the FFT algorithm, the FMM scheme is slower. However, the larger the sample, the smaller the difference becomes. The memory consumption of the FMM scheme with $lev = 3$ and of the FFT scheme is given in Table 2 for the same sizes. There is a remarkable difference in memory needs between the two algorithms. The normalized root-mean-square error for different $LEV - lev$ parameters and different sample dimensions are shown in table 3. This rms error is defined as

Table 1: Timing of the FMM algorithm for different sample dimensions (p=6) using different $(LEV - lev)$ combinations. The last row shows the run time for the FFT scheme.

$64 \times 64 \times 64$		$128 \times 128 \times 128$		$256 \times 256 \times 256$		$512 \times 512 \times 512$	
$(4-2)$	1.95 s	$(5-2)$	17.34 s	$(6-2)$	142.0 s		
$(3-3)$	1.50 s	$(4-3)$	12.42 s	$(5-3)$	95.7 s	$(6-3)$	13 min 36 s
$(2-4)$	1.86 s	$(3-4)$	14.71 s	$(4-4)$	115.3 s	$(5-4)$	15 min 23 s
FFT	0.277 s	FFT	3.13 s	FFT	30.2 s		

Table 2: Memory consumption of the FMM scheme $lev = 3$ and of the FFT scheme.

	$64 \times 64 \times 64$	$128 \times 128 \times 128$	$256 \times 256 \times 256$	$512 \times 512 \times 512$
FMM	16 Mb	64 Mb	0.48 Gb	3.78 Gb
FFT	82 Mb	654 Mb	5.23 Gb	41.9 Gb (estimation)

$$\text{error} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\frac{|\mathbf{H}_{ms,i}^{FMM} - \mathbf{H}_{ms,i}^{FFT}|^2}{|\mathbf{H}_{ms,i}^{FFT}|^2}}. \tag{9}$$

Table 3 shows that the error slightly increases when the dimensions of the sample increase. Further, the error decreases when the size of the base boxes increases. This is when lev is large. This is to be expected because for large base boxes, a relatively large number of interactions is computed using near field computations which have an accuracy corresponding with the machine precision.

Fig. 1 shows the magnetization, the amplitude of the magnetostatic field and the normalized rms error in two planes of a sample with dimensions of $128 \times 128 \times 128$ FD cells. In the rms error plots the basis boxes are visible: at the edges of the basis boxes, the largest errors occur.

6. Conclusions and prospects

In general, FFT schemes and FMM schemes scale $\mathcal{O}(N \log N)$. However, when in the FMM schemes all FD cells have identical dimensions, the scaling factor for FFT schemes is usually much smaller than for FMM schemes. However, in this FMM implementation, the difference in scaling factor is small and rapidly diminishing for growing dimensions (factor 3.16 for $256 \times 256 \times 256$ FD cells). The memory needs of the FMM scheme are by far better than for the FFT scheme: the memory needs are no limiting factor anymore. In the framework of the micromagnetic computations, an error smaller than 1 percent is sufficient. This is easily achieved in the FMM algorithm.

The CPU time of the FMM algorithm can be improved by using an adaptive mesh that allows FD cells of different sizes. This will decrease the CPU time vastly. Indeed, as already mentioned,

Table 3: Normalized rms error for different sample dimensions and $(LEV - lev)$ parameters.

$64 \times 64 \times 64$		$128 \times 128 \times 128$		$256 \times 256 \times 256$	
$(4-2)$	$2.13\,e-3$	$(5-2)$	$2.30\,e-3$	$(6-2)$	$2.44\,e-3$
$(3-3)$	$1.98\,e-3$	$(4-3)$	$2.22\,e-3$	$(5-3)$	$2.37\,e-3$
$(2-4)$	$1.54\,e-3$	$(3-4)$	$2.01\,e-3$	$(4-4)$	$2.24\,e-3$
		$(2-5)$	$1.49\,e-3$	$(3-5)$	$1.96\,e-3$

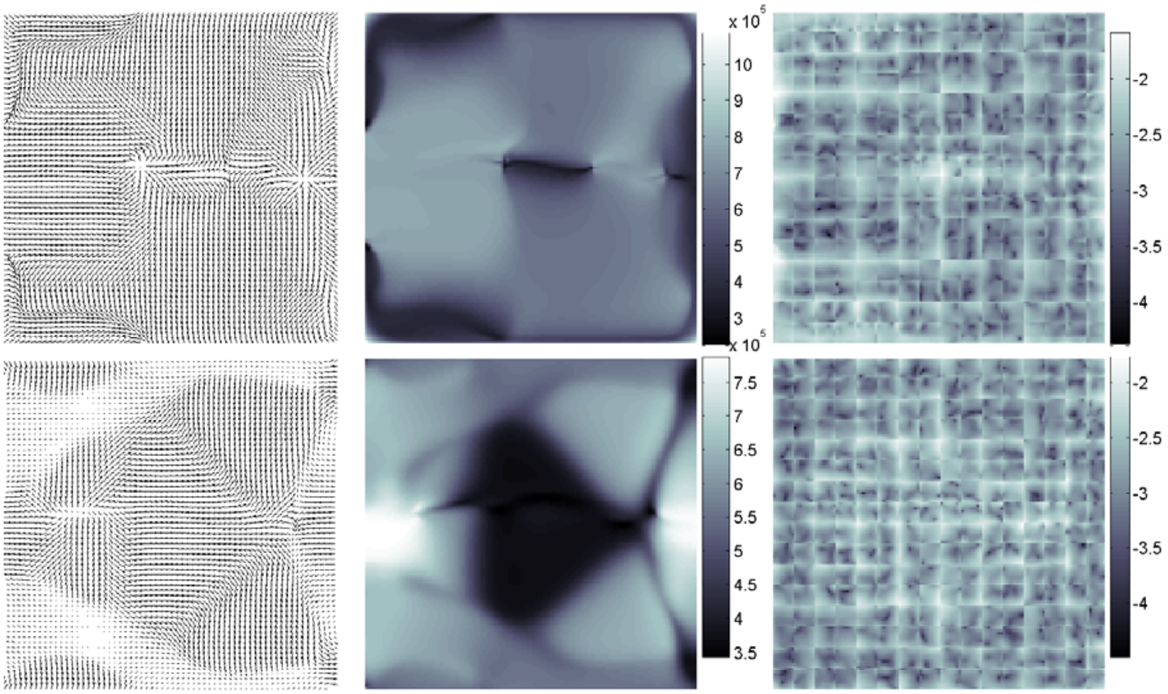

Figure 1: Magnetization (left), amplitude of the magnetostatic field (middle) and normalized rms error on a logarithmic scale (right) in planes $z = 0$ (up) and $z = 16\Delta$ (bottom) of a sample of $128 \times 128 \times 128$ FD cells.

the ferromagnetic samples consist of homogeneously magnetized domains separated by domain walls, as in the magnetization plots of Fig. 1. Hence very large FD cells can be used to discretize the magnetic domains while small FD cells can be used to discretize the domain walls. When using FMM with cells of different sizes the algorithm will outperform the FFT method in CPU time and memory requirements. Similar computation techniques as presented here can be used in this FMM scheme with adaptive mesh. However, since the domain structure varies during the dynamic magnetization processes, the adaptive mesh has to be adjusted every time step. This will increase the complexity of the algorithm.

References

[1] J. G. Zhu, “Modelling of multilayer thin film recording media,” *IEEE Trans. Magn.*, Vol. 28, No. 5, pp. 3267-3269, 1992.

[2] H. Kronmüller and M. Fänhle, *Micromagnetism and the microstructure of ferromagnetic solids*, Cambridge University Press, 1994.

[3] B. Van de Wiele, F. Olyslager, and L. Dupré, “Fast numerical 3D-scheme for the simulation of hysteresis in ferromagnetic grains,” *J. Appl. Phys.* Vol. 101, 073909, 2007.

[4] L. Greengard and V. Rokhlin, “A new version of the fast multipole method for the Laplace equation in three dimensions,” *Acta Numerica*, Vol. 6, pp. 229-269, 1997.

[5] M. Abramowitz and I. A. Stegun, *Handbook of mathematical functions*, McGraw Hill, 1960.

[6] W. D. Elliott and J. A. Board Jr., “Fast Fourier transform accelerated fast multipole algorithm,” *SIAM J. Sci. Comput.*, Vol. 17, No. 2, pp 398-415, 1996.