Simple iterative calculation of micromagnetic kernels

D. M. Apalkov and P. B. Visscher
Department of Physics and Astronomy and MINT Center
University of Alabama, Tuscaloosa AL 35487-0324
(Dated: September 7, 2002)

To calculate the magnetostatic field in a micromagnetic calculation, it is customary to use a stored "kernel", which is the average field in a cubic cell due to a uniform magnetization in some other cell. This can be calculated from a dipole approximation if the cells are far apart, but is obtained from a table of stored kernels for a finite set of small cell-separation vectors. The kernel is usually calculated numerically or analytically, either of which is laborious and error-prone. We have found an iterative method that requires almost no coding beyond what is already present in a micromagnetic code, and converges much more rapidly than finite-element integration.

PACS numbers: 75.40.Mg

I. INTRODUCTION

In micromagnetic simulation of magnetic materials [1], the most time-consuming step is the computation of the magnetostatic field at each computational cell. If the field at each cell due to the dipole moment of every source cell is calculated separately, the computer time required is proportional to N^2, where N is the number of cells (an "N^2 algorithm"). Various methods exist for reducing this time, including fast Fourier transform ($N \ln N$, but requires a regular grid of cells) and fast multipole (N, but with a large pre-factor). When two cells are far apart, it is adequate to approximate the source as a point dipole, and evaluate the field at the center of the field cell. However, for nearby cells this is not adequate and one often assumes a uniform magnetization over the source cell and averages its field over the field cell. This average (for unit magnetization) is called a "micromagnetic kernel" and can be formally written as a double integral over the two cells, as in Eq. 3 below.

This integral is not extremely difficult to do numerically, and can even be done analytically[2],[3],[4], but doing it numerically requires one to spend a lot of time learning numerical integration techniques (or at least learning the peculiarities of a numerical integration package) and doing it analytically requires a large amount of algebra and formula-checking (and there appear to be some incorrect formulas in the literature). Either approach introduces a substantial amount of complication and opportunity for error into a micromagnetic calculation.

We have developed a very simple recursive procedure for calculating the kernel that does not require any mathematical coding beyond what is already present (and presumably debugged) in a micromagnetic code. The basic idea is that if we have an estimate (which can be the simple point-dipole approximation) of the kernel for cells of size a, our existing code can calculate the kernels for cell size $2a$, and these are more accurate than the estimate. Repetition of this process converges to the exact kernels.

II. RECURSIVE CALCULATION OF KERNEL

We consider a system of rectangular cells of dimensions $a_x \times a_y \times a_z$. The i-component ($i = x,\ y,$ or z) of the magnetic field at a field point $\mathbf{r}_f$ is given (in SI units) by

$$H_i(\mathbf{r}_f) = \sum_j \int d^3\mathbf{r}_s \frac{\partial}{\partial r_{fi}} \frac{\partial}{\partial r_{sj}} \frac{M_j(\mathbf{r}_s)}{4\pi\,|\mathbf{r}_s - \mathbf{r}_f|} \qquad (1)$$

where the integral is over the source cell. One of the partial derivatives turns the potential $\frac{1}{4\pi|\mathbf{r}_s-\mathbf{r}_f|}$ of a monopole into that of a point dipole of unit strength in direction j. We multiply this by the actual magnitude $M_j d^3\mathbf{r}_s$ of the dipole, and the other partial derivative turns the potential into a field. Averaging this field over the field cell gives

$$H_i^{\rm av}(\mathbf{r}_f) = \frac{1}{a_x a_y a_z} \int d^3\mathbf{r}_f H_i(\mathbf{r}_f) = \sum_j K_{ij} M_j \qquad (2)$$

where we have factored out the constant magnetization and defined the kernel

$$K_{ij} = \frac{1}{a_x a_y a_z} \int d^3\mathbf{r}_f \int d^3\mathbf{r}_s \frac{\partial}{\partial r_{fi}} \frac{\partial}{\partial r_{sj}} \frac{1}{4\pi\,|\mathbf{r}_s - \mathbf{r}_f|} \qquad (3)$$

Here i and j are Cartesian indices and K_{ij} is a 3x3 matrix. If the source cell is centered at $\mathbf{r}_s^0$ and the field cell at $\mathbf{r}_f^0$, the kernel is a function of the center-to-center displacement $\mathbf{r}_f^0 - \mathbf{r}_s^0 = (n_x a_x, n_y a_y, n_z a_z)$ where the n_i's are integers. Note that the kernel is dimensionless – it depends on the aspect ratio but not on the absolute size of the cells. We can deduce a constraint on the trace of the kernel: when we sum the diagonal matrix elements, the derivative in Eq. 3 becomes a Laplacian, which gives a Dirac delta function, which integrates to -1 for self-kernels ($n = 0$) and to 0 for non-self-kernels. This is a useful consistency check on the results. If the cells have cubic symmetry, the self-kernel is diagonal with equal diagonal elements, which must then equal -1/3. This well-known value of the demagnetization tensor for a sphere is valid for any object with cubic (not just spherical) symmetry.

In general, a micromagnetic code stores the kernel $K_{ij}(\mathbf{n})$ for inter-cell separations less than some value R, and uses an analytic form for separations larger than R. The simplest analytic form is just the point-dipole approximation, approximating the integrals in Eq. 3 by the value when the source and field points are both at the respective cell centers. However, accurate results can be obtained using smaller R by using a higher order multipole approximation to the field of a uniformly magnetized cube; we use a fast multipole (FMM) code[5] that uses such multipoles, so the error vanishes as a high power of $1/R$. [The FMM also involves lumping cells together at larger distances for higher efficiency, but this aspect is not important here.] We give results here for both the "dipole" and "multipole" treatment of non-stored kernels (those with separations $> R$). To make R dimensionless, it is convenient to scale the cells so the volume $a_x a_y a_z = 1$; in the cubic case, this means the cell separation is just the magnitude $|\mathbf{n}|$.

The main point of this paper is that we never need to explicitly evaluate the integral in Eq. 3. Assuming we already have a micromagnetic code (using some estimated kernels), we can calculate the fields for any magnetization distribution. In particular, if we assume a unit magnetization along the x axis ($M_j = \delta_{xj}$) in a source cell and calculate H_i^{av}, we can then calculate three components of the kernel matrix:

$$K_{ix} = H_i^{\text{av}} \tag{4}$$

Repeating this with unit magnetizations along y and z gives the other components of the kernel. If we did this with cells of the same dimensions $a_x \times a_y \times a_z$ for which we previously defined the kernel, we would trivially get the original kernel back. The idea of the iterative method is that we compute the kernels for large cells (dimensions $2a_x \times 2a_y \times 2a_z$), and that these are more accurate than the input kernels on the smaller scale.

The geometry of the recursive method for calculating the kernel is indicated in Fig. 1. A uniform magnetization is assumed in the left-hand (source) cell, and the micromagnetic code calculates (using the assumed kernel) the average field in each of the 8 subcells of the field cell. These fields are averaged to give the large-cell average used in Eq. 4 to obtain the new large-cell kernel. Since the correct kernel is actually scale-independent, this can be used in the next iteration as the small-cell kernel.

Formally, we can write Eq. 3 for the large cells, and re-express it in terms of small-cell kernels to obtain an explicit equation for the new kernel in terms of the old one:

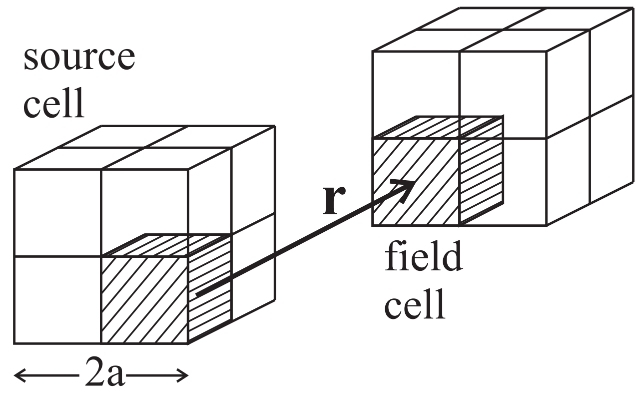

FIG. 1: Cells used in the iterative calculation of the micromagnetic kernel. One small-scale source-cell-field-cell pair is shown, out of the 64 pairs that are involved in computing the large-cell average field that determines the kernel.

$$K_{ij}^{\text{new}} = \frac{1}{8a_x a_y a_z} \sum_{\substack{8\text{ small}\\ \text{field cells}}} \sum_{\substack{8\text{ small}\\ \text{source cells}}} \int d^3\mathbf{r}_f \int d^3\mathbf{r}_s \frac{\partial}{\partial r_{fi}} \frac{\partial}{\partial r_{sj}} \frac{1}{4\pi\,|\mathbf{r}_s - \mathbf{r}_f|} = \frac{1}{8} \sum_{\substack{8\text{ small}\\ \text{field cells}}} \sum_{\substack{8\text{ small}\\ \text{source cells}}} K_{ij}^{\text{old}} \tag{5}$$

where we have not written the arguments of K_{ij}^{old}, but these are the 64 separations between pairs of small cells in Fig. 1.

Eq. 5 is the iteration equation which underlies our method. However, we emphasize that we have not coded this equation explicitly – it is merely implicit in the iteration procedure (described below). Explicitly, we use only the trivial Eq. 4. We can get some insight into possible instabilities from Eq. 5, however. In particular, the self-kernel $K_{ij}^{\text{old}}(n = 0)$ appears 8 times on the right hand side, so its overall coefficient is 1.0. Thus after convergence, the rest of the terms on the right hand side must add to exactly zero. If we replace the correct self-kernel with an incorrect one, it will never return to the correct value. The iteration procedure can thus **not** give us information about the self-kernel. Fortunately, in the usual case of cubic cells, this is not a problem because we deduced (above) the exact self-kernel. For non-cubic cells with rational aspect ratios, we can construct the cell from smaller cubic cells and use a formula similar to Eq. 5 to compute the self-kernel from cubic-cell kernels.

It is easy to see that the kernel $K(100)$ appears four times on the right hand side of Eq. 5, for an overall gain factor of 1/2, the largest such factor. This suggests that the non-self-kernels will converge; this is also supported by the results in Table I.

To understand why this scheme converges, it helps to think of the result after one iteration as a crude 8-cell

TABLE I: Convergence of the kernel calculation, for the slowest-converging case ($\mathbf{n} = (1,0,0)$). Starting point (iteration 0) is the dipole-dipole approximation. Left columns used a simple dipole code for separations $> R = 3.75$, right columns used 6th order multipoles.

iter-	order 2		order 6	
ation	K_{xx}	K_{yy}	K_{xx}	K_{yy}
0	0.159155	-0.0795775	0.159155	-0.0795775
1	0.147695	-0.0738474	0.147695	-0.0738474
2	0.141656	-0.0708033	0.141453	-0.0707267
3	0.138403	-0.0691659	0.138258	-0.0691288
...	...	...	...	...
11	0.134699	-0.0673416	0.135030	-0.0675151
12	0.134692	-0.0673378	0.135024	-0.0675119
13	0.134688	-0.0673359	0.135021	-0.0675103
14	0.134686	-0.0673350	0.135019	-0.0675095
15	0.134685	-0.0673345	0.135018	-0.0675091

Riemann approximation (i.e, discrete sum) to the integral. Then after two iterations the result is equivalent to a sum over 64 cells, and after N iterations there are effectively 2^N cells in each direction. Generally Riemann approximations converge as a power of the cell size; empirically the error decreases by a factor of two in each iteration, for at least 15 iterations with single precision arithmetic.

III. RESULTS

In principle, we should be able to use any starting approximation to the stored kernel (the part with separation $< R$) to begin the iteration. In fact, of course, the result will converge sooner if we have a good starting approximation. Table I shows the convergence using the point-dipole approximation as a starting point, for the slowest-converging (smallest n) kernel with $\mathbf{n} = (100)$. Note that the starting value differs from the correct kernel by only 0.03 in the worst case [$K_{xx}(100)$], a relative error of about 25%; at other separations the starting errors are smaller.

Table I gives the results of using two different codes for calculating the field in Eq. 4, the "dipole" and "multipole" (order 6) methods discussed above, both with cutoff radius $R = 3.75$. [Note that "dipole" here refers to the treatment of distant cells ($> R$); in both cases we use the point-dipole result for near cells as our starting kernel.] The error for multipole order p should be proportional to $R^{-(p+2)}$, or $3.75^{-4} = 0.005$ for the dipole data in Table I; we can estimate the proportionality coefficient by noting that the actual error (3×10^{-4} is more than ten times smaller. Using this estimate for the error in the order-6 multipole indicates that it should be exact to six figures.

Table II shows the converged kernel tensors for several inter-cell separations.

TABLE II: Selected values of the micromagnetic kernel tensor for cubic cells, after 15 iterations of the multipole code (complete table available upon request).

n_x	n_y	n_z	$\begin{pmatrix} K_{xx} & K_{xy} & K_{xz} \\ K_{yx} & K_{yy} & K_{yz} \\ K_{zx} & K_{zy} & K_{zz} \end{pmatrix}$
0	0	0	$\begin{pmatrix} -0.333333 & 0 & 0 \\ 0 & -0.333333 & 0 \\ 0 & 0 & -0.333333 \end{pmatrix}$
1	0	0	$\begin{pmatrix} 0.135018 & 0 & 0 \\ 0 & -0.067509 & 0 \\ 0 & 0 & -0.067509 \end{pmatrix}$
1	1	0	$\begin{pmatrix} 0.013786 & 0.045565 & 0 \\ 0.045565 & 0.013786 & 0 \\ 0 & 0 & -0.027572 \end{pmatrix}$
3	1	0	$\begin{pmatrix} 0.004273 & 0.002252 & 0 \\ 0.002252 & -0.001765 & 0 \\ 0 & 0 & -0.002508 \end{pmatrix}$

IV. CONCLUSIONS

We have described an efficient iterative method for the calculation of micromagnetic kernels which minimizes coding complexity and therefore the opportunity for errors, by using the existing capability of a micromagnetic code to calculate magnetostatic fields.

V. ACKNOWLEDGEMENTS

This work was partially supported by DOE grant No. DE-FG02-98ER45714 and NSF grant DMR-9809423, and by the Computational Materials Sciences Network sponsored by the Materials Sciences Division of the DOE Office of Science. The authors are indebted to Michael Donahue for leading them to some of the literature on kernels.

[1] A. Aharoni, *Introduction to the theory of ferromagnetism*, Oxford U. P., 1996.

[2] A. J. Newell, W. Williams, and J. J Dunlop, "A Generalization of the Demagnetizing Tensor for Nonuniform Magnetization", J. Geophys. Res.**98**, 9551-9555 (1993).

[3] M. Maicas, E. Lopez, M. C. Sanchez, C. Aroca, and P. Sanchez "Magnetostatic Energy Calculations in Two- and Three-Dimensional Arrays of Ferromagnetic Prisms",

IEEE Trans. Mag. **34** 601-607 (1998).

[4] M. E. Schabes and A. Aharoni, "Magnetostatic interaction fields for a 3-D array of ferromagnetic cubes", IEEE Trans. Magn. **23**, 3882-3888 (1987).

[5] P. B. Visscher and D. M. Apalkov, preprint at http://bama.ua.edu/~visscher/mumag/cart.tex.