RC 22414 (W0204-123) April 26, 2002
Physics

IBM Research Report

Performance of a Novel Algorithm for Perpendicular Magnetic Recording Simulation

D.M. Newns, W.E. Donath, G.J. Martyna
IBM Research Division
Thomas J. Watson Research Center
P.O. Box 218
Yorktown Heights, NY 10598

M.E. Schabes, B. E. Lengsfield,T. Schrefl
IBM Almaden Research Center
650 Harry Rd., San Jose, CA 95120

Research Division
Almaden - Austin - Beijing - Delhi - Haifa - India - T. J. Watson - Tokyo - Zurich

Performance of a Novel Algorithm for Perpendicular Magnetic Recording Simulation

D.M. Newns, W.E. Donath, G.J. Martyna
IBM T.J. Watson Research Center, P.O.B. 218, Yorktown Heights, NY
M.E. Schabes, B. E. Lengsfield, and T. Schrefl
IBM Almaden Research Center, 650 Harry Rd., San Jose, CA 95120

Abstract

The Grand Challenge problem presented by simulation of perpendicular magnetic recording requires an algorithm scaling close to linearly with the number of elements in the simulation cell. We derive an appropriate algorithm based on 2D FFT's of field and magnetization in the disk plane, combined with a novel treatment of field propagation in the z-direction. The near-linear scaling of the algorithm with number of elements is demonstrated by computer experiments. Although the computation contains a strong data-parallel element, in that an ensemble of 20 members is required in order to achieve statistics, some parallelization is desirable. Adequate parallelization is achieved on an SP machine. On an Intel cluster, speed-up saturation is observed, and is understood in terms of bandwidth limitations.

1 Introduction

Bit densities on hard discs used for magnetic storage are at present increasing at a rate of 100 percent per year. It is probable that, in the drive to increase bit density, current storage media employing magnetic bits aligned parallel to the disk surface (longitudinal recording) will be replaced by a new design concept with bits aligned perpendicular to the disk surface (perpendicular recording). The proposed technology shift [1, 2, 7] is motivated by the need to maintain resistance to thermal demagnetization in bits of successively smaller physical dimension. The required thermal demagnetization resistance can be achieved by use of harder magnetic materials in the storage medium. However, harder magnetic materials require larger write fields, which above a certain level may only be achievable in the perpendicular configuration. The source of the available enhanced write field lies in the use of a highly permeable Soft magnetic Under Layer (SUL) beneath the magnetic storage medium at the disk surface. The magnetic field leaves the write pole essentially pointing normal to the disk surface (say 'DOWN'), crosses through the narrow air gap and medium, which becomes magnetically polarized in the desired write direction, bends over horizontally in the SUL, re-enters the medium again oriented normal to the disk surface (say 'UP') and crosses the air gap to enter the return pole (see Fig. 1). The return pole has a larger area than the write pole, thus reducing the flux per unit area and hence minimizing its write effect, and the return pole is also located ahead of the write pole, both design features maximizing control of the written bit by the write pole. In contrast to longitudinal recording, where there is no SUL and the write field is not strongly affected by the medium, perpendicular recording presents a more complex engineering situation in which the write process critically depends on the combined SUL and write head designs.

The complexities intrinsic to perpendicular recording technology give rise to the necessity for numerical simulations to be closely coupled to the process of disk and head design[9, 5, 3, 8]. It is necessary to simulate the head plus disk (disk including medium and SUL) together, and hence to implement in software the process of writing several bits. In order to adequately model this aspect of perpendicular recording, a typical simulation box (Fig. 1) will be around $2560 \times 1280 \times 620$ nm^3 at a resolution of 5 nm or less. In the present approach, the elements are orthorhombic in shape (Fig. 1). The resolution requirement is maximum in the recording medium layer. The head can be modelled at lower resolution in the regions well above the disk plane. The SUL can also be modelled at reduced resolution. Thus while the medium will be modelled at typically 512×256 resolution, and in elements 5 nm. in depth, in regions above and below the medium the element size can be several times 5 nm.. Because of the massive nature of the required simulation, it is sometimes referred to as a computational *Grand Challenge* problem.

The main features[6] of the simulation approach are illustrated in Fig. 2. Associated with every element **i** there is

a magnetization vector $\mathbf{m_i}$, assumed to be constant within the element, and a magnetic field vector $\mathbf{h_i}$. Given $\mathbf{h_i}(t_n)$ and $\mathbf{m_i}(t_n)$ at time t_n, the Landau-Lifschitz-Gilbert (LLG) equation

$$\frac{d\mathbf{m_i}}{dt} = -|\gamma|\mathbf{m_i} \times \mathbf{h_i} + \frac{\alpha}{m_s}\mathbf{m_i} \times \frac{d\mathbf{m_i}}{dt}$$

is solved using a numerical integrator (Runge-Kutta) in order to update the magnetization $\mathbf{m_i}(t_n)$ to $\mathbf{m_i}(t_{n+1})$. Here γ is the gyromagnetic ratio, m_s is the saturation magnetization of the material, and α is the damping constant.

The updated magnetization leads to an updated field $\mathbf{h_i}(t_{n+1})$, by taking into account the interactions illustrated in Fig. 2. The head field changes at each time step, both due to movement of the head and to changes in its magnetization originating at its driving coil (the driving coil is external to, and assumed to be independent of, the simulation, being treated as an 'external variable'). These changes need to be put in place before the update of the field is started. The components involved in the field update are briefly summarized as follows

- For finite temperature simulations a Langevin-like noise term is included in the simulation by adding it into the field; this term is local in nature. It is necessary to repeat the calculation for a thermal ensemble of magnetic configurations. This requirement needs to be considered in the overall assessment of simulation cost.

- The anisotropy term is uniaxial and drives the magnetization to lie close to an *easy axis*. In perpendicular recording the easy axis is close to the normal to the plane of the disk. The energy term associated with the anisotropy is local and in its simplest form is

$$E_u = K_u \sin^2 \theta_\mathbf{i},$$

where $\theta_\mathbf{i}$ is the angle between the magnetization $\mathbf{m_i}$ and the local easy axis, and K_u is the (positive) anisotropy parameter for the material. For the medium, the local easy axis in a given element $\mathbf{i}$ lies within a narrow cone determined physically by crystallite orientation, and is set randomly, within an experimentally determined standard deviation, in a pre-computation step (inter-grain correlations thus being neglected).

- The energy associated with the exchange term involves an interaction between neighboring elements $\mathbf{i}$ and $\mathbf{j}$

$$E_{ex} = -2J_{ij}\mathbf{m_i} \cdot \mathbf{m_j},$$

where J_{ij} is the material-dependent exchange integral. J_{ij} is short range, and is included only for nearest-neighbor interactions. J_{ij} is positive for the ferromagnetic case.

- The demag term is the long range interaction between magnetic elements. For two elements $\mathbf{i}$ and $\mathbf{j}$ the corresponding energy is given by

$$E_{dm} = \mathbf{m_i} \int_{Cell\ \mathbf{i}} d^3r \int_{Cell\ \mathbf{j}} d^3r' \, \Phi\left(\mathbf{r} - \mathbf{r}'\right)\ \mathbf{m_j},$$

where in the demagnetization calculation the pairwise interaction Φ is the dipole-dipole interaction tensor with components α and β

$$\Phi_{\alpha\beta}\left(\mathbf{r}\right) = \frac{3x_\alpha x_\beta}{r^5} - \frac{\delta_{\alpha\beta}}{r^3}.$$

The magnetic field is retrieved from the nonlocal energy expressions by differentiation with respect to magnetization

$$h_{\mathbf{i},\alpha} = -\frac{\partial E}{\partial m_{\mathbf{i}\alpha}},$$

where $E = E_{ex}$ or E_{dm}.

The simulation needs to be run for at least 10^5 time steps in order to simulate a meaningful write process.

2 Scalable Algorithm

Almost all the calculations involved in the update are local or nearest-neighbor in nature, and thus scale as O(N), where $N = N_x N_y N_z$ is the number of elements in the simulation cell, N_α being the number of elements along the α^{th} direction. The exception is the demag component, which naively scales as N^2, a scaling which would be prohibitively costly in this Grand Challenge application. In order to implement the simulation on a system of size which is of practical interest, an algorithm for demag scaling close to O(N) is **indispensible**. The scalable approach for the demag calculation which we have developed is customized for the present problem. It is FFT-based in the two dimensions lying in the disk plane, the x and y directions, but uses a novel approach to treat the z direction. The treatment in the z-direction is adapted to a relatively small number of irregularly spaced z-layers (on the order of $40 - 60$), and is cheaper than a full 3-dimensional FFT, which must maintain the highest resolution along the z-direction and whose dimension must be extended beyond the z dimension of the simulation cell to allow zero padding.

The 2D FFT approach is ideally suited to treat periodic boundary conditions, which limit edge effects. We define images or replicas of the simulation cell located at $\mathbf{A_N} = (a_x n_x, a_y n_y)$, where $a_x,\ a_y$ are the simulation cell dimensions and the components of $\mathbf{N} = (n_x,\ n_y)$ are integers. Here a notation is introduced where bold capitals define 2D vectors, and scalars are used for z-components. Thus element location $\mathbf{i} = (\mathbf{I}, i)$, where $\mathbf{I} = (i_x, i_y)$ defines element location in the xy-plane, and i defines the element

position on the z-axis. Similarly $\mathbf{r} = (\mathbf{X},z)$ etc. We introduce a 2D Discrete Fourier Transform (DFT) as

$$\begin{aligned}\bar{\mathbf{m}}_{\mathbf{K},i} &= \sum_{\mathbf{I}=\mathbf{0}}^{\mathbf{N}_{\max}-\mathbf{1}} e^{i\mathbf{K}.\mathbf{X}_\mathbf{I}}\mathbf{m}_{\mathbf{I},i}\ ;\ \mathbf{K} = 2\pi(\frac{l_x}{a_x},\frac{l_y}{a_y});\\ & l_\alpha = 0,1,2,..N_\alpha - 1.\end{aligned}$$

Here $\mathbf{N}_{\max} = (N_x, N_y)$ and $\mathbf{X_I} = (i_x\alpha, i_y\beta)$, where α and β are the x and y dimensions of the element (in this paper we shall for notational simplicity assume that α and β, and also the z- dimension of the element η, are independent of z-layer i). Define element volume $v = \alpha\beta\eta$.

Next, we use the Poisson Summation Formula to transform the total demag energy

$$\begin{aligned}V_{dm}^{tot} &= \frac{1}{2}\sum_{\mathbf{I},\mathbf{J},i,j}\mathbf{m}_{\mathbf{I},i}^T\int_{el.\ \mathbf{I},i}d^2Xdz\int_{el.\ \mathbf{J},j}d^2X'dz'\\ &\times\sum_{\mathbf{N}}\Phi\left(\mathbf{X}-\mathbf{X}'+\mathbf{A}_\mathbf{N},z-z'\right)\ \mathbf{m}_{\mathbf{J},j},\end{aligned}$$

to Fourier space. We then obtain in mixed Fourier-z space the magnetic field as

$$\bar{\mathbf{h}}_{\mathbf{K},i} = \bar{\mathbf{h}}^{>}_{\mathbf{K},i} + \bar{\mathbf{h}}^{<}_{\mathbf{K},i} + \bar{\mathbf{h}}^{=}_{\mathbf{K},i},$$

where, introducing the row vector $\mathbf{V}(\mathbf{K}) = (iK_x, iK_y, K)$

$$\begin{aligned}\bar{\mathbf{h}}^{>}_{\mathbf{K},i} &= \frac{1}{\alpha\beta v}\sum_{\mathbf{G}}\mathbf{V}^T(\mathbf{K}+\mathbf{G})\left|f_{\mathbf{K}+\mathbf{G}}\right|^2 h^2_{|\mathbf{K}+\mathbf{G}|}\\ &\times\frac{2\pi}{|\mathbf{K}+\mathbf{G}|}\chi_i^+\left(\mathbf{K}+\mathbf{G}\right)\end{aligned}$$

$$\begin{aligned}\bar{\mathbf{h}}^{<}_{\mathbf{K},i} &= \frac{-1}{\alpha\beta v}\sum_{\mathbf{G}}\mathbf{V}^T(\mathbf{K}+\mathbf{G})^*\left|f_{\mathbf{K}+\mathbf{G}}\right|^2 h^2_{|\mathbf{K}+\mathbf{G}|}\\ &\times\frac{2\pi}{|\mathbf{K}+\mathbf{G}|}\chi_i^-\left(\mathbf{K}+\mathbf{G}\right)\end{aligned}$$

and

$$\begin{aligned}\bar{\mathbf{h}}^{=}_{\mathbf{K},i} &= \frac{1}{\alpha\beta v}\sum_{\mathbf{G}}\ \left|f_{\mathbf{K}+\mathbf{G}}\right|^2\frac{2\pi}{|\mathbf{K}+\mathbf{G}|}g_{|\mathbf{K}+\mathbf{G}|}\\ &\times\sum_i\left[\mathrm{Re}\{\mathbf{V}^T(\mathbf{K}+\mathbf{G})\mathbf{V}(\mathbf{K}+\mathbf{G})\}\right]\mathbf{m}_{\mathbf{K},i}.\end{aligned}$$

Here the reciprocal lattice vector $\mathbf{G} = \mathbf{2}\pi\left(\frac{m_x}{\alpha},\frac{m_y}{\beta}\right)$, m_x and m_y being integers. The form factors $f_\mathbf{K}$, h_K and g_K are defined by

$$f_\mathbf{K} = \frac{4\sin\left(\frac{K_x\alpha}{2}\right)\sin\left(\frac{K_y\beta}{2}\right)}{K_xK_y},$$

$$h_K = \frac{2\sinh\left(\frac{K\eta}{2}\right)}{K},$$

and

$$g_K = \frac{2}{K^2}\left(K\eta + e^{-\eta K} - 1\right).$$

The 2D χ matrices are defined by

$$\chi_i^+\left(\mathbf{K}+\mathbf{G}\right) = e^{-|\mathbf{K}+\mathbf{G}|z_i}\sum_{j=0}^{i-1}e^{|\mathbf{K}+\mathbf{G}|z_j}\rho^+_{\mathbf{K}+\mathbf{G},j},$$

and

$$\chi_i^-\left(\mathbf{K}+\mathbf{G}\right) = e^{|\mathbf{K}+\mathbf{G}|z_i}\sum_{j=i+1}^{N_z-1}e^{-|\mathbf{K}+\mathbf{G}|z_j}\rho^-_{\mathbf{K}+\mathbf{G},j},$$

where the $\rho^\pm_{\mathbf{K}+\mathbf{G},j}$, related to the magnetostatic charge density, are

$$\rho^\pm_{\mathbf{K}+\mathbf{G},j} = (iK_x + iG_x, iK_y + iG_y, \pm|\mathbf{K}+\mathbf{G}|)\ .\ \bar{\mathbf{m}}_{\mathbf{K},j}.$$

Note that the expansion in reciprocal space vectors $\mathbf{G}$ is required in order to correctly reproduce the short-range properties of the magnetization and field at and below the scale of a single element.

From the transform $\mathbf{h}_{\mathbf{K},i}$ the field in a given real space element may be obtained by inverting the DFT (FFT techniques are used to get the DFT and its inverse).

So far, because there is a j-sum embedded in each χ_i, the algorithm retains the appearance of being O(N_z^2). But we notice that the j-sums in the expressions for χ_i^+ and χ_i^- do not depend on the variable i. This feature enables the calculation of the χ's to be restructured as a pair of recursion relations. We get the upward recursion relation for χ^+

$$\begin{aligned}\chi^+_{i+1}\left(\mathbf{K}+\mathbf{G}\right) &= e^{-\eta|\mathbf{K}+\mathbf{G}|}(\rho^+_{\mathbf{K}+\mathbf{G},i} + \chi_i^+\left(\mathbf{K}+\mathbf{G}\right));\\ &\chi_0^+\left(\mathbf{K}+\mathbf{G}\right) = 0\end{aligned}$$

and its downward fellow for χ^-

$$\begin{aligned}\chi^-_{i-1}\left(\mathbf{K}+\mathbf{G}\right) &= e^{-\eta|\mathbf{K}+\mathbf{G}|}(\rho^-_{\mathbf{K}+\mathbf{G},i} + \chi_i^-\left(\mathbf{K}+\mathbf{G}\right));\\ &\chi^-_{N_z-1}\left(\mathbf{K}+\mathbf{G}\right) = 0.\end{aligned}$$

In its recursive form the algorithm is manifestly of order N_z, so that the leading computational complexity for the whole calculation, formally dominated by this loop and the FFT's, is of order $N\log_2(N_xN_y)$.

Although we have demonstrated it for the simplest case of uniformly spaced layers, the recursion relation still works if the spacings z_i of the layers is nonuniform, as desired in

this application. And, it is not even necessary for the different layers to have the same element size in the x- and y-directions, provided the spacings are multiples of a minimum spacing, since the $\mathbf{K}$-spaces of different layers, extended by the infinite $\mathbf{G}$-vector periodic array, map into each other.

In the present form accuracy is controlled by the number of $\mathbf{G}$-vectors n_G (including zero) included in the sum, which can be costly if it is too large. The convergence of the $\mathbf{G}$-sum is poorest for the case $\mathbf{h}_{\overline{\overline{\mathbf{K}}},i}$, and it is useful to enhance accuracy by precomputing these self-layer terms in real space. Their evaluation then leads to no loss of accuracy and is only O(N). An additional trick is based on the short-range nature of the remaining error, and one may determine the best near-neighbor correction in order to minimise this error with a given number of $\mathbf{G}$-vectors, and absorb the correction into the exchange part of the calculation. It is then found that with $n_G = 9$ $\mathbf{G}$-vectors the results are accurate to about $1 - 2\%$ max. error. At the same time, the compute time for the recursion relations is under half the total for a time step. Hence the approach described above is viable.

A more general and complete formulation is described in a forthcoming paper by the present authors[4]. In the more complete approach, the demag computation, here implemented solely in $\mathbf{K}$-space, is split between real space and $\mathbf{K}$-space in a systematic manner. Both parts of the calculation, including that of $\mathbf{h}_{\overline{\overline{\mathbf{K}}},i}$, are rapidly convergent. In the general algorithm, any desired accuracy may be achieved.

3 Implementation and Parallelization

3.1 Implementation

We first want to supply some implementation details:

- In the computation of the field due to the head, the piece originating in the external driver coil is represented as a magnetostatic charge sheet on the upper face of the simulation box, and consists of a charge sheet covering the write pole plus a second sheet of opposite polarity covering the return pole. The integrated charge in the two sheets is zero. In terms of the 2D Fourier representation, the displacement of the head is implemented by multiplication by a phase factor. Since the head field also changes in time, in response to the current in an external driver coil, it is also multiplied by a corresponding time-varying scalar factor proportional to the instantaneous driver coil current.
- The thermal fluctuations component is added as a random number with Gaussian distribution into the field (Fig. 2).
- The demag field
  - Computation of the self-field $\bar{\mathbf{h}}_{\overline{\overline{\mathbf{K}}},i}$ of the layers; the 'response function' tensor of the self-field, a difference kernel which linearly relates the field to the magnetization in real space, is pre-computed and then Fourier transformed. At each time step it is only necessary to multiply the transformed magnetization by the response function, and add to the transformed component.
  - The inter-layer field. First the three components of the magnetization are FFT'd and converted into the scalar ρ matrices. The recursion relations for each component of the 2D $\chi_i^{\pm}\left(\left(K_x + G_x\right), \left(K_y + G_y\right)\right)$ matrices are then implemented. Now, the χ matrices are the transforms of real matrices, so that they have the property that
$$\chi_i^{\pm}\left(\left(K_x + G_x\right), \left(K_y + G_y\right)\right)^* = \chi_i^{\pm}\left(-\left(K_x + G_x\right), -\left(K_y + G_y\right)\right)$$
so that the size of the matrix is approximately the same as that of the real matrix (as a matter of fact, most FFT implementations for real matrices generate a complex matrix of $2(N_x/2+1)N_y$ elements, which for even N_x is just slightly larger than N_xN_y). The symmetry relation is used to reduce storage and computation, and to reduce the data transmission needed in parallelization. After converting the χ's to magnetic field, we inverse FFT each component. Hence there are 6 FFT's per field evaluation.
  - The use of a finite number n_G of $\mathbf{G}$-vectors entails some loss of accuracy, which is however short range in real space. Hence this error can be compensated by including a real space correction. In the present implementation, we allow only for corrections between adjacent layers of cells in the same column. More accuracy may require the transmission of more the real space of more layers; we denote by $|L|$ the number of layers for which real magnetization will have to be transmitted - where for the present implementation $|L|$ is 1.

A last item we wish to mention that, in order to solve the LLG equation by Runge-Kutta, two force field computations are done per time step: both use essentially the same resources.

3.2 Parallelization Approach

The approach taken is to map a layer, or a contiguous group of layers, onto a processor in a parallel-architecture

machine. The data transmission needed between the processors is then primarily that of the magnetization matrix and the χ matrices from the outermost layers of the layer group in each processor. This approach means that all FFT's are calculated intra-processor, which obviates a large amount of interprocessor data transmission from this source.

The volume of data to be transferred from one layer group to the next is now derived. The real space magnetization is $2 \times 3|L|N_xN_yW$ bits, where W is the number of bits in a word, 2 refers to the 2 force computations per time step, and 3 is the number of components of the magnetization. The χ matrix is represented by $2n_GN_xN_yW$ bits of information (this formula is somewhat approximate; as discussed earlier, as a matter of practice the χ matrix is slightly larger since it is a complex matrix representing a transformation of a real matrix). The total volume of data (which we denote by V_{LG}) can be given by

$$V_{LG} \quad = \quad 2(3|L| + n_G)N_xN_yW.$$

If we partition the system amongst 2 processors, the total volume of data to be transmitted and received between the 2 processors is given by $2V_{LG}$. Should we partition the system amongst more than 2 processors, the processors corresponding to the outer layers require still $2V_{LG}$ data bits, but the processors corresponding to the interior layers all require $4V_{LG}$ data bits. For $N_x = N_y = 64, n_G = 9, |L| = 1, W = 64$ we find $V_{LG} = 6.3\,10^6$.

4 Experimental Results

We first consider the scaling with number of elements. This is displayed in Fig. 3. It is seen that although there is a very tiny amount of supralinearity, the scaling is essentially as N.

The efficiency of parallelization is seen in Fig. 4 and Fig. 5 and in Tables I and II below.

The experimental results on parallelization indicate a much better scaling for the IBM SP machine (Fig. 4 and Table I): the bandwidth on that machine is more than on order of magnitude better than that of the Intel cluster (Fig. 5 and Table II). The residual slowing with increasing numbers of layers on the SP machine is probably due to the computational overhead that the communication imposes, which is a constant independent of the number of layers.

Table I: Results for 375 MHz Power3-II system

CPUs	$64 \times 64 \times 16$		$128 \times 128 \times 32$	
	sec/cycle	speedup	sec/cycle	speedup
1	1.120	1.00	10.10	1.00
2	0.580	1.93	5.80	1.74
4	0.305	3.67	3.00	3.27
8	0.154	7.27	1.54	6.56
16	0.088	12.7	0.853	11.8
32	—	—	0.58	17.4

However, the limit of scalability one observes on the Intel processors is certainly due to the bandwidth load that the communication imposes. The cluster we used has a nominal communication bandwidth of 100 Mb/sec.. In the $64 \times 64 \times 16$ experiment the communication time for $4V_{LG}$ is approximately 0.3 sec.. The experimental results for the $64 \times 64 \times 16$ experiment indeed show a breakdown of scaling at around $4 - 8$ processors, where the communication-free computation time would be $0.5 - 0.25$ secs., whence the breakdown at this point is to be anticipated. Now consider the situation for the $128 \times 128 \times 32$ experiment. The communication time scales as the number of elements per layer, and is thus $4\times$ that of the $64 \times 64 \times 16$ experiment. However, in a cluster of a given total number of processors, the number of layers per processor is doubled. Hence the computation time is $8\times$ that in the $64 \times 64 \times 16$ experiment. Thus the computation/communication ratio is actually $2\times$ more favorable in the larger system, and the breakdown in scaling occurs at a larger number of processors (Fig. 5, Table II). We should also expect better performance using a Gigabit/sec ethernet.

Table II: Results for Intel Pentium III 600 MHz with 100 MBits/sec Ethernet

CPUs	$64 \times 64 \times 16$		$128 \times 128 \times 32$	
	sec/cycle	speedup	sec/cycle	speedup
1	2.10	1.00	14.60	1.00
2	1.03	2.04	8.78	1.67
4	0.61	3.44	4.80	3.04
8	0.41	5.12	2.80	5.21
16	0.32	6.56	1.70	8.58
32	—	—	1.20	12.2

5 Conclusion

The Grand Challenge problem presented by simulation of perpendicular magnetic recording requires an algorithm scaling close to linearly with the number of elements in the simulation cell. We have derived an appropriate algorithm based on 2D FFT's of field and magnetization in the disk plane, combined with a novel treatment of field propagation

in the z-direction. Computational experiments on an early version of the algorithm, which is reasonably adequate, are given in the present paper, to be followed in a future paper by a full discussion of a more complete algorithm, enabling the demag field to be computed with arbitrary accuracy, and incorporating speed-ups from multigridding. The scaling of the algorithm linearly with number of elements is demonstrated by computer experiments.

The computation contains a strong data-parallel element, in that an ensemble of 20 members is required in order to achieve statistics. Nevertheless a moderate degree of parallelization is desirable in order to speed up turn-round time. The parallelization is investigated by experiments on both an IBM SP machine and on an Intel cluster. Adequate parallelization is achieved on the SP machine. On the Intel cluster, speed-up saturation is observed, and is understood in terms of bandwidth limitations whose scaling is investigated and enables understanding of the experimental results. If the number of layers is realistically large, and with enhancement of communication bandwidth, the Intel cluster should also form a suitable hardware platform for this type of simulation.

Taking into account anticipated moderate speed enhancements, the prospects for simulating realistic engineering situations with our current approach on accessible hardware seem excellent.

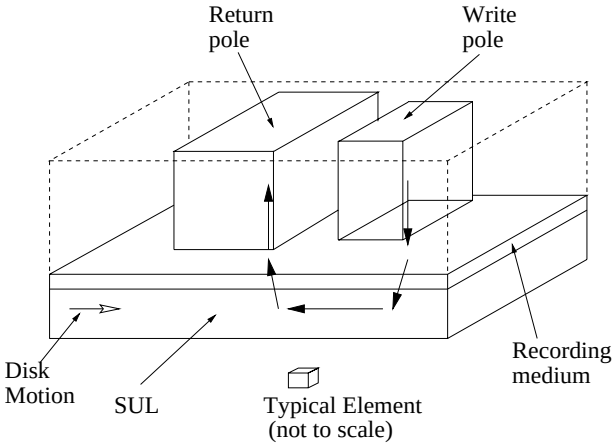

Figure 1. Sketch of simulation cell for perpendicular recording, showing write head with write and return poles, recording medium, and Soft UnderLayer. Bold arrows indicate direction of magnetic field, assuming downward direction at write pole.

References

[1] H. Bertram. *Theory of Magnetic recording*. Cambridge University Press, Cambridge, UK, 1994.

[2] H. Bertram and M. Williams. *IEEE Trans. Magnetics*, 36:4, (2000).

[3] K. Gau and H. Bertram. *Proceedings of MMM, Seattle, Washington, paper GB08*, (2001).

[4] D. Newns, W. Donath, G. Martyna, M. Schabes, B. Lengsfield, and T. Schrefl. *in preparation*, (2002).

[5] R. Victora, W. Peng, J. Xue, and J. Judy. *J.M.M.M*, 235:305, (2001).

[6] S. Wang and A. Taratorin. *Magnetic Information Storage Technology*. Academic Press, NY, 1999.

[7] R. Wood. *IEEE Trans. Magnetics*, 36:36, (2000).

[8] R. Wood, Y. Sonobe, Z. Jin, and B. Wilson. *J.M.M.M*, 235:1, (2001).

[9] J. Xue and R. Victora. *J. Appl. Physics*, 87:6361, (2000).

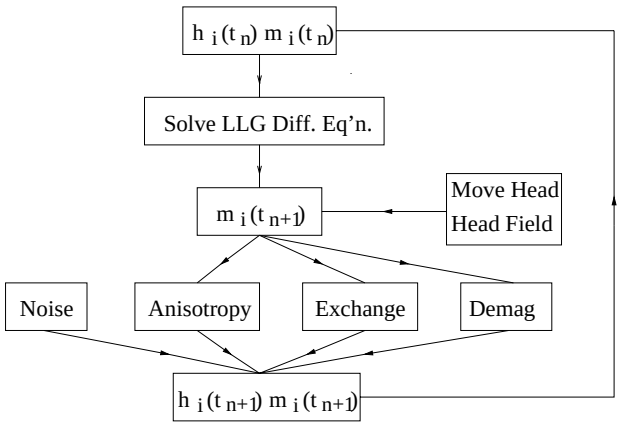

Figure 2. Flow diagram for simulation code (see text).

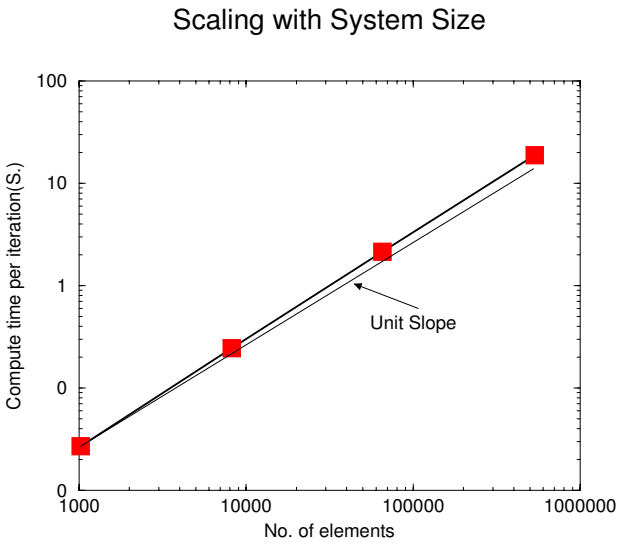

Figure 3. Scaling of compute time with system size expressed as No. of elements.

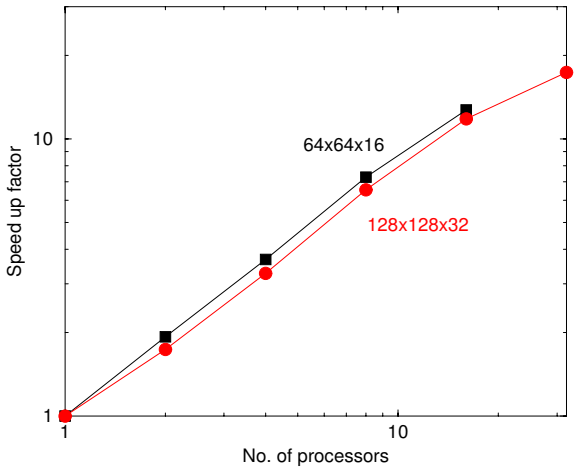

Figure 4. Speed-up factor, expressed as (time on single node/time on n nodes), plotted vs. n for SP system of Table I.

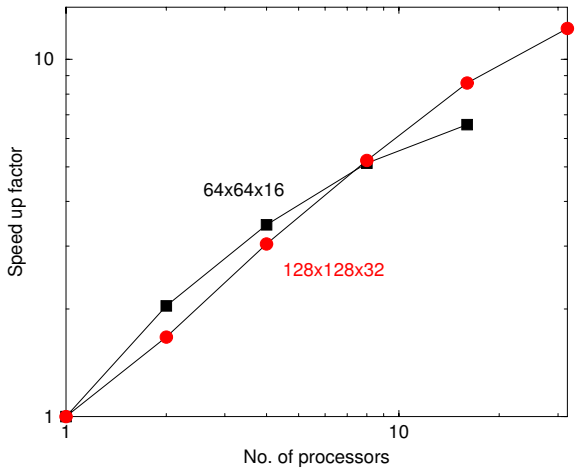

Figure 5. Speed-up factor, expressed as (time on single node/time on n nodes), plotted vs. n for Intel cluster of Table II.