Fast Algorithms for Micromagnetic Computations

Alex Solomonoff
Institute for Math and its Applications
University of Minnesota
Minneapolis, MN 55455

October 14, 1993

Abstract

This paper describes a new approach to computing the evolution in time of the magnetization of a ferromagnetic material. In many cases, a separation-of-time-scales argument can be used to say that the system is almost always near an equilibrium state. This allows us to use a continuation algorithm to follow the equilibrium state as it moves. However, there are still times when the system is not near equilibrium, and the underlying dynamical equations have to be integrated in those cases. These non-equilibrium computations can be very time-consuming, and we also describe an idea for doing this more efficiently. This depends on the often-valid assumption that the particles in the micromagnetic system are only weakly coupled by magnetostatic forces. This means that during the non-equilibrium times, usually only a few particles will take part in the dynamics and the others can be frozen. We present various tables and pretty pictures describing the success of these ideas.

1 Introduction

The behavior of magnetic materials such as floppy disks and magnetic tape is not well understood and is currently a subject of active research. A popular idealization of ferromagnetic materials is the Landau-Lifshitz equation/micromagnetics model, which is the subject of this paper.

A typical calculation of the simulation of a hysteresis loop. In a hysteresis loop, a sample of magnetic material is immersed in a time-varying magnetic field. The direction of the external field is fixed, but its amplitude changes from negative to positive and then back to negative. As the external field changes, the average magnetization of the sample also changes, and for a hysteretic material such as a ferromagnet, the magnetization on the return path is different than on the forward path.

The simulation of a hysteresis loop can be a very expensive calculation. There are several reasons for this. The first is that each timestep in the Landau-Lifshitz time integration requires several evaluations of the magnetostatic field, a calculation similar to the solution of Poisson's equation. The other is that the system of ordinary differential equations that has to be integrated has a couple of different kinds of stiffness. If an explicit ODE solver is used, the number of timesteps required would be very large.

In this paper, we describe some methods of making this less computationally intensive. We do this by exploiting two properties of the micromagnetic system. The first is that it is often the case that the only action of the Landau-Lifshitz dynamics is to force the system to follow an equilibrium state as the equilibrium moves around in the state space. In that case, the details of the Landau-Lifshitz dynamics are unimportant, and any process that followed the equilibrium would give the same results. We exploit this fact by employing a continuation algorithm to follow the equilibrium as it moves around under the influence of a changing applied field, and assuming that the state is at the equilibrium point. However, occasionally, the equilibrium will stop attracting the system. When this happens, the system wanders about the state space, looking for an attractive equilibrium. This process is governed by the Landau-Lifshitz equations, and a numerical algorithm has to detect this transition and switch back to the Landau-Lifshitz dynamics until the system reaches a new stable equilibrium.

A property that strongly effects that nature of the micromagnetic system is the fact that the interaction between magnetic particles is somewhat weak. The magnetic media on tapes and floppy disks is a collection of small particles imbedded in a plastic film. Their weak interations has several consequences. One is that it makes the Landau-Lifshitz equations stiff, even during the brief non-equilibrium

times. Another is that the evolution of the system at the non-equilibrium times tends to involve only a few particles, with all the others nearly fixed. And since each nonequilibrium event involves only a few particles, the number of these events is large. Therefore, the calculation of the nonequilibrium events is very expensive.

We can use the fact that nonequilibrium events usually involve only a few particles to speed up this calculation, by figuring out somehow which particles take part in each event and calculating only their evolution, while freezing all the others.

In section 2, we briefly review micromagnetics. In section 3, we derive equations for the evolution of the equilibrium state. In section 4 we discuss numerical algorithms for computing the evolution of the equilibrium. In section 5 we show some numerical results and compare the performance of our quasi-equilibrium or quasi-steady algorithm with an established algorithm. In section 6 we derive and analyze algorithms for the partly-frozen treatment of the nonequilibrium events.

2 Preliminaries of Micromagnetism

The material of this section appears in many other places, for example [1] and [2], but we will review it here to make the paper more self-contained.

2.1 Ferromagnetism

Each atom in a magnetic material can be thought of as being (or containing) a tiny magnet. The strength of this magnet is fixed, but it can point in any direction. In the absence of any organizing influence, they would all point in random directions and the average magnetization of the material would be zero. In a ferromagnetic material, there is a phenomenon, called exchange energy, which is such an organizing force.

Let s_i be the magnetic moment of the $i-$th atom, so $\|s_i\| = \|s_j\|\ \forall i,j$. Then the exchange energy is usually approximated by the expression

$$E_{exch} = -\alpha \sum_{i,j} s_i \cdot s_j \tag{1}$$

where the sum is over all pairs of neighboring atoms in the lattice. (The material is assumed to be crystalline, with its atoms arranged on the points of some lattice.) This energy is minimized when all the atomic magnets point in the same direction. It allows a ferromagnetic material to have a nonzero average magnetization.

The exchange energy forces orientation of the atomic magnets to vary only over distances much larger than the interatomic distance. This makes it possible to think of the magnetization as a function of a continuous variable (i.e. space), $M(x)$. Like the atomic magnets, $|M(x)|$ is constant. This allows us to express quantities that are sums over all the atoms in the material by integrals over space.

2.2 Other Energies

There are other components of the energy of the system. If the material is immersed in an externally applied magnetic field $H_{app}(x)$, there is energy associated with that:

$$E_{app} = -\beta \int M(x) \cdot H_{app}(x) dx, \tag{2}$$

with β some positive constant.

In a crystalline material the axes of the crystal lattice form a coordinate system, and the atomic magnets prefer some directions of orientation over others with respect to the lattice axes. This is called crystalline anisotropy, and there is anisotropy energy associated with this. In general, there is some function f such that

$$E_{ani} = \int f(M(x)) dx \tag{3}$$

The function f will have the same symmetries that the crystal lattice has. In the case that the lattice has only one distinguished direction (usually called the "easy axis"), a common idealization of the anisotropy energy is

$$E_{ani} = \int |M(x) \cdot K(x)|^2 dx \tag{4}$$

where $K(x)$ is the distinguished direction. It is constant away from grain boundaries, impurity atoms, crystal defects, etc.

Each atomic magnet generates its own magnetic field, which affects every other atomic magnet. This interaction results in the magnetostatic energy. Let $B(s, x-y)$ be the magnetic field at the point y generated by by a magnetic moment s at the point x. It is a linear operator in s. Then there is a magnetic field generated by all of the atomic magnets which is

$$H_{ms}(y) \quad = \quad \int B(M(x)), x-y) dx \tag{5}$$

and the magnetostatic energy is then:

$$E_{ms} \quad = \quad -\frac{1}{2} \int M(x) \cdot H_{ms}(x) dx \tag{6}$$

The $\frac{1}{2}$ comes from the fact that it is a quadratic form in $M(x)$.

The total energy is the sum of all the different energies:

$$E = E_{exch} + E_{ani} + E_{ms} + E_{app}. \tag{7}$$

2.3 Dynamics

The condition for the system to be in equilibrium is that the energy to be a minimum, subject to the constraint that the $|M(x)|$ be fixed. If we define the *effective field* to be

$$H_{eff}(x) = -\frac{\partial E}{\partial M(x)} \tag{8}$$

then the condition of equilibrium is

$$M(x) \times H_{eff}(x) = 0. \tag{9}$$

What if we are not at equilibrium? How does the system behave? A first guess at a dynamic equation would be

$$\frac{d}{dt} M(x) = \gamma M(x) \times H_{eff}(x) \tag{10}$$

This has the problem that it contains no dissipation. If the system does not start in equilibrium, it will not approach it, which actual magnetic materials are known to do. The real mechanisms of dissipation are quite complicated (see [3]) and are often replaced in theoretical and numerical studies by the *Landau-Lifshitz* equation:

$$\frac{d}{dt} M(x) = \gamma M(x) \times H_{eff}(x) - \alpha M(x) \times \big(M(x) \times H_{eff}(x)\big). \tag{11}$$

This is the simplest dynamic equation that is not obviously wrong, and it actually predicts many phenomena fairly accurately.

2.4 Coherent Rotation

If there were only the anisotropy energy, the magnetization would consist of regions where M is nearly pointing in one of the "easy directions", (local minima of the anisotropy energy), separated by abrupt transitions. The exchange energy forbids abrupt transitions. In the presence of both anisotropy and exchange energy, the magnetization consists of regions of nearly constant magnetization (called domains), separated by transition regions (domain walls).

The characteristic thickness of domain walls is determined by the relative strength of the exchange and anisotropy energies. Large anisotropy energy favors thin walls, large exchange energy favors thicker walls. Call the characteristic distance d_w.

If the size of a piece of magnetic material is much smaller than d_w, then the magnetization tends to be nearly uniform. If the magnetic system consists of a group of tiny, magnetic particles, then a common assumption is the

Coherent Rotation Assumption:

At any time, in a collection of small ferromagnetic particles, $M(x)$ is a constant function inside each particle.

This is a large simplification of the system. Instead of M depending on space, it depends on which particle. It goes from being a function to a vector. Also the exchange energy plays no role in the dynamics except to force $|M(x)|$ to be fixed.

3 Separation of Time Scales

There is a characteristic time associated with the Landau-Lifshitz equation. If H_0 is a characteristic magnetic field strength, for example, the coercivity, then

$$t_{inner} = \frac{1}{\gamma H_0} \tag{12}$$

In magnetic recording (MR) applications, t_{inner} is typically about 10^{-10} seconds. Actually there is a second characteristic time associated with dissipation and the constant α, but if $\alpha|M|/\gamma = O(1)$, then the two characteristic times are the same. This is actually the case in MR, with this ratio being typically 0.3.

There is another characteristic time of interest. This is the characteristic time for changes in the external applied field, which will be called t_{outer}. In MR, this is approximately the time required to read or write a bit on a magnetic disk, which is typically 10^{-6} seconds. So

$$\frac{t_{outer}}{t_{inner}} = O(10^4). \tag{13}$$

3.1 Evolution by Stepped Applied Fields

If an explicit ODE solver were used, it would have to use timesteps of order t_{inner}, and the computation would be very expensive. An idea which is sometimes used to get around this is to "skip" most of the computation, since things are happening very slowly anyway. This is evolution by stepped applied fields,

and it works like this: The external magnetic field is a function of time which we will write as $h(t)$. The system is initialized in some random state, and then the Landau-Lifshitz equations are integrated forward in (inner) time with the external field fixed at $h(0)$ until the system reaches equilibrium. Then the external magnetic field is changed to $h(\Delta t)$ and the system is brought to equilibrium again using the Landau-Lifshitz dynamics. This is repeated until the external magnetic field has passed through all of the values that make up the hysteresis loop.

There is no real guarantee that this produces the same results as continuously changing the external field, and later we will describe a situation where the discrete jumps change the dynamics of the system.

3.2 Quasi-Steady-State Approximation

Suppose that the magnetic system requires $O(t_{inner})$ time to reach equilibrium from a nonequilibrium state. Then the applied field will have hardly changed in this time. Suppose that the equilibrium state is a smooth function of the external applied field. Then the equilibrium state will have hardly moved also in the time required for the system to reach equilibrium. Then hopefully a valid assumption is the

Quasi-Steady-State Assumption:

If the inner time is much smaller than the outer time, and the system is dissipative, then the state of the system is always near equilibrium.

In this case, a valid replacement to the original dynamics of the system is the

Quasi-Steady-State Approximation:

Replace the actual state by the equilibrium state, and replace the actual time evolution equation, by the time evolution for the equilibrium state.

3.3 Quasi-Steady Evolution Equation

The material in this section is very linear-algebraic in nature, and I will be changing the notation slightly to agree more with linear algebra conventions. Let $\{a_i\}_{i=1}^n$ denote the vector obtained by concatenating the vectors a_i. Let $\mathrm{diag}(D_i)$ denote a block diagonal matrix having D_i as its diagonal blocks. Let m_i be a 3-vector containing the magnetization of the i-th particle, and the magnetization of the whole system is $m = \{m_i\}_{i=1}^n$, assuming that there are n particles in the collection. Let $h_{app} = \{h_{app\,i}\}_{i=1}^n$ be the vector containing the applied field at the location of each of the particles. Recall that two components of the effective field, the anisotropy and the magnetostatic field, are linear operators in the magnetization.

$$H_{ani} = \mathrm{diag}(c_i c_i^t) \tag{14}$$

is the linear operator giving the anisotropy field when applied to the magnetization vector, where c_i is a 3-vector whose direction is the easy axis of the i-th particle and whose length is the strength of the anisotropy of the i-th particle. Define the matrix H_{ms} to be such that $H_{ms}m$ is the magnetostatic field. Its elements are

$$(H_{ms})_{ij} = B(|m_i|, x_i - x_j), \tag{15}$$

where x_i is the location of the i-th particle. Then the linear part of the effective field operator is

$$H = H_{ms} + H_{ani} \tag{16}$$

and the effective field is given by

$$h_{eff} = Hm + h_{app}. \tag{17}$$

What is the state space? It is the subset of $(\mathbb{R}^3)^n$ consistent with the constraint that $|m_i|$ be fixed. Let us assume, without loss of too much generality, that $|m_i| = 1$, $\forall i$. Then $m_i \in S_2$, where S_2 is the surface of the unit sphere in $\mathbb{R}^3$. Consider the 2-dimensional subspace of vectors perpendicular to m_i. Call it $\mathcal{T}_i$. This is the tangent subspace of S_2 at m_i. Small changes in m_i must lie in $\mathcal{T}_i$ for $|m_i|$ to be fixed. A matrix that will be important is the orthogonal projector onto $\mathcal{T}_i$. This is

$$P_i = I_3 - m_i m_i^t. \tag{18}$$

If $x \in \mathcal{T}_i$, i.e. $x \perp m_i$, then

$$P_i x = x - m_i m_i^t x = x. \tag{19}$$

If $x = \alpha m_i$, i.e. $x \perp \mathcal{T}_i$, then

$$P_i x = \alpha m_i - \alpha m_i m_i^t m_i = 0. \tag{20}$$

The state space for the whole system is $(S_2)^n$, the tangent space at m is $\mathcal{T} = \prod_i \mathcal{T}_i$, and the projector onto $\mathcal{T}$ is

$$P(m) = \mathrm{diag}(P_i). \tag{21}$$

We will often write $P(m)$ just as P when the identity of m is clear from context.

The equation of equilibrium can be expressed using $P(m)$. Recall the equilibrium equation is

$$m_i \times h_{eff\,i} = 0, \quad \forall i \tag{22}$$

Since the left-hand side is perpendicular to m_i and $h_{eff\,i}$, we can vector-multiply by it m_i without

changing its meaning. Applying the vector triple product identity gives us

$$\begin{aligned} m_i \times (m_i \times h_{effi}) &= (m_i \cdot h_{effi})m_i - (m_i \cdot m_i)h_{effi} \\ &= -m_i^t m_i (I_3 - \frac{m_i m_i^t}{m_i^t m_i}) h_{eff\,i} \\ &= -\|m_i\|^2 P_i h_{eff\,i} \end{aligned} \quad (23)$$

and expressing all n of the eqations (22) at once is

$$P(m)h_{eff} = P(m)(Hm + h_{app}) = 0. \quad (24)$$

To find the equation for the evolution of the equilibrium in time, we need to take the time derivative of (24):

$$0 = \dot{P}(m)h_{eff} + P(m)(H\dot{m} + \dot{h}_{app}) \quad (25)$$

What is $\dot{P}(m)$?

$$\begin{aligned} \dot{P}(m) &= \frac{d}{dt}\mathrm{diag}(I_3 - m_i m_i^t) & (26) \\ &= -\mathrm{diag}(\dot{m}_i m_i^t + m_i \dot{m}_i^t). & (27) \end{aligned}$$

From the equilibrium equation we can say

$$0 = P(m)h_{eff} = \{h_{eff\,i} - m_i m_i^t h_{eff\,i}\} \quad (28)$$

and so

$$h_{eff\,i} = m_i(m_i^t h_{eff\,i}). \quad (29)$$

This allows us to write

$$\begin{aligned} \dot{P}(m)h_{eff} &= -\{\dot{m}_i m_i^t h_{effi} + m_i \dot{m}_i^t h_{eff\,i}\} \\ &= -\{\dot{m}_i m_i^t h_{effi} + m_i \dot{m}_i^t m_i m_i^t h_{effi}\} \end{aligned} \quad (30)$$

But since $m_i^t \dot{m}_i = \frac{1}{2}\frac{d}{dt}|m_i|^2 = 0$, the second term of (30) is zero, giving us

$$\begin{aligned} \dot{P}(m)h_{eff} &= -\{\dot{m}_i m_i^t h_{eff\,i}\}_{i=1}^n \\ &= -\mathrm{diag}(m_i^t h_{eff\,i} I_3)\dot{m}. & (31) \end{aligned}$$

This allows us to write the quasi-steady evolution equation as

$$-\left(P(m)H - R\right)\dot{m} = P(m)\dot{h}_{app} \quad (32)$$

with

$$R = \mathrm{diag}(m_i^t h_{eff\,i} I_3). \quad (33)$$

Note that

$$P(m)\,\dot{m} = \dot{m} \quad (34)$$

and

$$P(m)R = RP(m) \quad (35)$$

so (32) is equivalant to

$$-P(m)\big(H - R\big)P(m)\dot{m} = P(m)\dot{h}_{app} \quad (36)$$

This version has the property that the matrix on the left side is symmetric and is an operator from the tangent space of the state manifold to itself. This means that the singular system of equations that must be solved to obtain $\dot{m}$ is consistent.

3.4 Stable Equilibrium and Switching

In (36), $\dot{m}$ and $\dot{h}_{app}$ have a linear relationship. This means that if $\dot{h}_{app}$ is multiplied by -1, then $\dot{m}$ is also multiplied by -1, which means that the quasi-steady dynamics are reversible. But real micromagnetic systems exhibit hysteresis, in fact hysteresis is the main reason they are of interest. What is wrong?

A micromagnetic system will follow an equilibrium point as long as it continues to exist. But it can happen that as h_{app} changes, the equilibrium point vanishes. A schematic of this is described in figure 1. When this happens, the system stops following the now vanished equilibrium point, and begins to wander about the state manifold in search of a new equilibrium point. When it finally wanders into the basin of attraction of an equilibrium point, it makes its way to the equilibrium and begins to follow it. This process of leaving the old equilibrium point and traveling to the new one is called a switching event, and it is irreversible. It is the only part of the dynamics of the system that is so.

The quasi-steady evolution equation is only valid when the system is following an equilibrium point. When a switching event begins, it stops being valid and must be replaced by the Landau-Lifshitz equations. When the system has reached the new equilibrium point, then the quasi-steady equation becomes valid again.

When is an equilibrium point stable? It is stable when the energy is at a minimum, considered as a function restricted to the state manifold. We will investigate this by mapping the state manifold of the system onto a Euclidean space, and looking at the system in the Euclidean space. So let D be a subset of the state manifold which is a neighborhood of an equilibrium point m^*. Let

$$m(s) : \mathbb{R}^{2n} \to D \quad (37)$$

be a smooth one-to-one mapping from a $2n$-dimensional Euclidean space to the state manifold. So now s is a new, alternative representation of the state of the micromagnetic system. Then an equilibrium point is stable if the matrix whose entries are

$$E_{ss} = \frac{\partial^2 E(m(s))}{\partial s_i \partial s_j} \quad (38)$$

is positive definite.

Definitions:

Suppose that a $k \times n$ matrix B is an orthonormal basis for some subspace of $\mathbb{R}^n$, and so $Q = B^t B$ is the projector onto that subspace. Then a matrix A is called Q-positive definite if the $k \times k$ matrix BAB^t

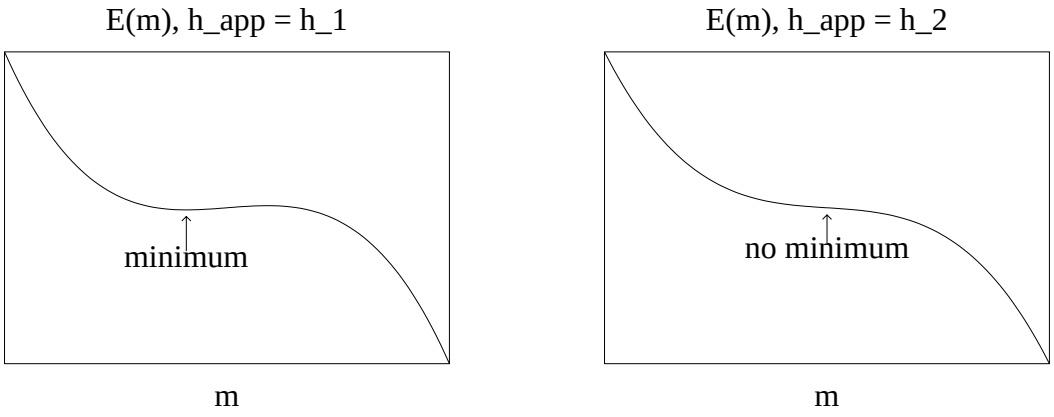

Figure 1: Onset of a switching event

is positive definite. So I-positive definiteness corresponds to the usual notion of positive definiteness.

I claim that $E_{ss} > 0$ if and only if the matrix

$$P(R - H)P \tag{39}$$

is P-positive definite.

To show this, first expand out the matrix E_{ss},

$$\frac{\partial^2 E}{\partial s \partial s} = \frac{\partial m}{\partial s}\frac{\partial^2 E}{\partial m \partial m}\frac{\partial m}{\partial s} + \frac{\partial E}{\partial m}\frac{\partial^2 m}{\partial s \partial s} \tag{40}$$

First, let us look at the second term. To understand this, let us look at the second derivative matrix for a single particle. So $m = m(s)$ is a one-to-one map from $\mathbb{R}^2$ to S_2, the unit sphere. The derivative matrix $\partial m/\partial s$ is a 2×3 matrix whose columns span the tangent subspace of S_2. We can arrange for $\partial m/\partial s$ to be orthogonal, at least at a single point. So

$$\Big(\frac{\partial m}{\partial s}\Big)\Big(\frac{\partial m}{\partial s}\Big)^t = I_2 \tag{41}$$

and

$$\Big(\frac{\partial m}{\partial s}\Big)^t\Big(\frac{\partial m}{\partial s}\Big) = I_3 - mm^t. \tag{42}$$

I claim that

$$\frac{\partial^2 m}{\partial s \partial s} = -mI_{2\times 2} \tag{43}$$

where each entry of this matrix is taken to be a 3-vector. We prove this by playing various games with derivatives of equation (41) and the expression $m^t m = 1$. So we have

$$(\frac{\partial m}{\partial s})^t m = 0 \tag{44}$$

and

$$\Big(\frac{\partial^2 m}{\partial s \partial s}\Big)^t m + \Big(\frac{\partial m}{\partial s}\Big)^t\Big(\frac{\partial m}{\partial s}\Big) = 0 \tag{45}$$

which gives

$$\Big(\frac{\partial^2 m}{\partial s \partial s}\Big)^t m = -I_{2\times 2} \tag{46}$$

when combined with equation (41). All of the multiplications here are to be interpreted as inner products in $\mathbb{R}^3$. Now we combine derivatives of (46), showing all the subscripts explicitly,

$$\frac{\partial}{\partial s_k}\Big((\frac{\partial^2 m}{\partial s_i \partial s_j})^t m\Big) - \frac{\partial}{\partial s_j}\Big((\frac{\partial^2 m}{\partial s_k \partial s_i})^t m\Big) = 0 \tag{47}$$

which gives us

$$\Big(\frac{\partial^2 m}{\partial s_i \partial s_j}\Big)^t\Big(\frac{\partial m}{\partial s_k}\Big) - \Big(\frac{\partial^2 m}{\partial s_k \partial s_i}\Big)^t\Big(\frac{\partial m}{\partial s_j}\Big) = 0. \tag{48}$$

But if we differentiate (41) we get

$$\frac{\partial}{\partial s_i}\Big((\frac{\partial m}{\partial s_j})^t(\frac{\partial m}{\partial s_k})\Big) = 0 \tag{49}$$

or

$$\Big(\frac{\partial^2 m}{\partial s_i \partial s_j}\Big)^t\Big(\frac{\partial m}{\partial s_k}\Big) + \Big(\frac{\partial m}{\partial s_j}\Big)^t\Big(\frac{\partial^2 m}{\partial s_k \partial s_i}\Big) = 0 \tag{50}$$

which we can combine with equation (48) to give

$$\Big(\frac{\partial^2 m}{\partial s \partial s}\Big)^t\Big(\frac{\partial m}{\partial s}\Big) = 0_{2\times 2\times 2} \tag{51}$$

So the entries of $\partial^2 m/\partial^2 s$ are orthogonal to the tangent subspace of S_2, and therefore must be parallel

to m. We combine this with (46) to give equation (43).

So now that we understand the single-particle case, we want to glue together n copies to make a mapping from $\mathbb{R}^{2n}$ to $(S_2)^n$. Let $m_i(s_i)$ be a mapping for the i-th particle having the properties derived above. Then let the mapping for the whole system be

$$m(s) = m(\{s_i\}_{i=1}^n) = \{m_i(s_i)\}_{i=1}^n. \tag{52}$$

Since the single-particle mappings are completely decoupled, the jacobian matrix for the whole mapping is a $2n \times 3n$ block diagonal matrix with 2×3 diagonal blocks. Since the diagonal blocks are orthogonal, the whole Jacobian matrix is orthogonal. Similarly, the second derivatives are a $2n \times 2n \times 3n$ third-rank diagonal tensor, with $2 \times 2 \times 3$ blocks on the diagonal. So we can also see that the second term of (40) is

$$\Big(\frac{\partial E}{\partial m}\Big)^t \Big(\frac{\partial^2 m}{\partial s \partial s}\Big) = \mathrm{diag}(h_{eff\,i}^t m_i I_{2\times 2}), \tag{53}$$

since

$$\frac{\partial E}{\partial m} = -h_{eff}. \tag{54}$$

We know that

$$\frac{\partial^2 E}{\partial m \partial m} = -H \tag{55}$$

and so we now have

$$\frac{\partial^2 E}{\partial s \partial s} = -\frac{\partial m}{\partial s} H \frac{\partial m}{\partial s} + \mathrm{diag}(h_{eff\,i}^t m_i I_{2\times 2}). \tag{56}$$

We can project E_{ss} onto the tangent subspace by multiplying before and after by $\partial m/\partial s$, which is an orthogonal basis for the subspace. The matrix will gain n zero eigenvalues, which correspond to the orthogonal complement of the tangent subspace, but the other eigenvalues will be unchanged. Since $\partial m/\partial s$ is an orthogonal basis for the tangent subspace, we have

$$\Big(\frac{\partial m}{\partial s}\Big)\Big(\frac{\partial m}{\partial s}\Big)^t = P(m) \tag{57}$$

and so

$$\begin{aligned}
&\Big(\frac{\partial m}{\partial s}\Big)\frac{\partial^2 E}{\partial s \partial s}\Big(\frac{\partial m}{\partial s}\Big)^t \\
&\quad = -PHP + \Big(\frac{\partial m}{\partial s}\Big)\mathrm{diag}(h_{eff\,i}^t m_i I_{2\times 2})\Big(\frac{\partial m}{\partial s}\Big)^t
\end{aligned}$$

But since $\partial m/\partial s$ is block diagonal, we have

$$\begin{aligned}
&\Big(\frac{\partial m}{\partial s}\Big)\frac{\partial^2 E}{\partial s \partial s}\Big(\frac{\partial m}{\partial s}\Big)^t \\
&= -PHP + \mathrm{diag}\Big(\frac{\partial m_i}{\partial s_i}(h_{eff\,i}^t m_i) I_2 \frac{\partial m_i}{\partial s_i}\Big) \\
&= -PHP + \mathrm{diag}\big((h_{eff\,i}^t m_i)(I_3 - m_i m_i^t)\big) \\
&= -PHP + \mathrm{diag}(h_{eff\,i}^t m_i I_3)\mathrm{diag}(I_3 - m_i m_i^t) \\
&= -PHP + RP \\
&= -P(H-R)P
\end{aligned} \tag{58}$$

which is the result we wanted.

3.5 Errors Due to Non-Equilibrium

A question that needs to be answered is whether the quasi-steady equation is stable with respect to not being exactly at equilibrium. So, suppose

$$e = P(Hm + h_{app}). \tag{59}$$

where e is some small deviation from equilibrium. Notice that e must lie in the tangent subspace of the solution manifold. Take the time derivative of this,

$$\begin{aligned}
\dot{e} &= \dot{P}h_{eff} + P(H\dot{m} + \dot{h}_{app}) \\
&= -R\dot{m} - \{m_i \dot{m}_i^t h_{eff\,i}\} + P(H\dot{m} + \dot{h}_{app})
\end{aligned} \tag{60}$$

This is similar to the derivation of (36) except that here the second term is not zero. Substituting (36) in for $\dot{m}$ and simplifying makes all but one term disappear,

$$\dot{e}_i = -m_i \dot{m}_i^t h_{eff\,i}. \tag{61}$$

If we take the dot product of this with e_i we see that

$$\frac{1}{2}\frac{d}{dt}|e_i| = e_i^t \dot{e}_i = -e_i^t m_i \dot{m}_i^t e_i \tag{62}$$

$$= 0 \tag{63}$$

since e_i and m_i must be orthogonal. So we see that the deviation from equilibrium does not grow in time.

A related question is whether $m_{equilibrium} - m$ can be large when e is small. If $-P(H-R)P$ has no small eigenvalues, the answer is no. But if $-P(H-R)P$ is nearly singular, as it is near the onset of a switching event, then perhaps it can happen.

4 Computational Considerations

So how do we compute the evolution of the magnetic system? First we need to be able to solve (36) for $\dot{m}$. Then we use $\dot{m}$ in some ODE solver such as RK4 to integrate m in time. At each timestep we determine whether the matrix in (39) is positive definite. If it is, we integrate forward another timestep. If not, then we switch to the Landau-Lifshitz equation.

We integrate that forward in time until equilibrium is reached, then switch back to the quasi-steady equation.

At each timestep we have to do two things: solve several linear systems of equations of the form (36), and determine if the matrix on the left side of (36) is positive definite.

A natural choice for these tasks is the conjugate gradient algorithm (CG). This algorithm is described in detail in many sources, including [5],chap. 10. This has 3 convenient features:
1) it is able to solve consistent singular systems of equations with no extra difficulty and no modifications to the algorithm. CG is an iterative algorithm, and each iteration involves inner products, linear combinations of vectors, and a matrix-vector multiplication. All of these operations can occur on a linear subspace with nothing bad happening. However, because of roundoff error, it is necessary to project the vectors onto the subspace every several iterations.
2) Almost all of the calculations done in the CG algorithm are the same computations required in the Lanczos algorithm for estimating the extremal eigenvalues of a matrix. If k iterations of CG are performed, then the extra work involved in estimating eigenvalues is $O(k)$.
3) The only thing that CG/Lanczos does with the matrix is to multiply it by vectors, allowing it to take advantage of any fast algorithm for computing the matrix-vector product. The computationally expensive part of the matrix multiplication in our case is the computation of the magnetostatic field, and CG can take advantage of fft-based or multipole algorithms for this.

4.1 Sturm Sequencing For Positive Definiteness Testing

Suppose the $n \times n$ linear symmetric positive definite system

$$Ax = b \tag{64}$$

is to be solved using the conjugate gradient algorithm (CG). CG works by projecting the matrix onto a Krylov subspace

$$S_k = \text{span}(b, Ab, \ldots, A^{k-1}b) \tag{65}$$

and solving the projected system of equations. At each iteration, the dimension of S_k is increased by one. When the subspace gets big enough, the solution of the projected system is a good approximation to the exact solution. After k iterations, there is a symmetric tridiagonal $k \times k$ matrix T_k (whose elements do not depend on k) which is the projection of A onto the S_k.

The most negative eigenvalue of T_k is greater than the most negative eigenvalue of A. Therefore if T_k is not positive definite, then A is not positive definite. We can use the Sturm sequencing property to easily test T_k and therefore A for positive definiteness. Of course, T_k being indefinite is a sufficient but not necessary condition for A to be indefinite. However, in practice it seems reliable.

Let $\{T_k\}$ be the $k \times k$ principle submatrices of a symmetric tridiagonal matrix. Consider the sequence

$$d_k(\mu) = \det(T_k - \mu I). \qquad k = 0, \ldots \tag{66}$$

They satisfy the recurrence formula

$$\begin{aligned} d_0(\mu) &= 1, \\ d_1(\mu) &= t_{11} - \mu, \\ d_k(\mu) &= (t_{kk} - \mu)d_{k-1}(\mu) - t_{k,k-1}^2 d_{k-2}(\mu). \end{aligned} \tag{67}$$

This sequence has the following interesting property:
Sturm Sequencing Property

The number of sign changes in this sequence is equal to the number of eigenvalues of eigenvalues of T_k which are less than μ. This is proved in many sources, for example [6], p.300.

By setting $\mu = 0$, we see that if the $d_k(0)$ are not all positive, then T_k is not positive definite and therefore A is not positive definite. By using the recurrence relation, they can be computed with only $O(1)$ operations per CG iteration. This is much easier than the more obvious technique of computing the eigenvalues of T_k at each iteration.

4.2 Preconditioning

Preconditioning is an important modification to the CG algorithm that can make it much more efficient in some cases. If the matrix is ill-conditioned, then many iterations of CG will be required to solve the system of equations. Suppose we are solving the system $Ax = b$, and suppose A' somewhat resembles A, except that it is easily inverted. Such a matrix is called a preconditioner for A. Preconditioned conjugate gradient consists of solving the system

$$A'^{-\frac{1}{2}} A A'^{-\frac{1}{2}} y = A'^{-\frac{1}{2}} b \tag{68}$$

with $x = A'^{-\frac{1}{2}} y$. At each iteration, this requires computing Au and $A'^{-1}v$ for some vectors u and v. Assuming that A' resembles A, $A'^{-\frac{1}{2}} A A'^{-\frac{1}{2}}$ will be a well-conditioned matrix and the CG process will converge quickly.

In this case, however, the Lanczos part of the algorithm would not be computing the eigenvalues

of A. Instead it would compute the eigenvalues of $A'^{-\frac{1}{2}}AA'^{-\frac{1}{2}}$. But this is good enough.

Lemma:

Suppose a matrix X is symmetric and invertible. Then a matrix A is positive definite iff XAX is positive definite.

Proof:

Suppose that XAX is indefinite. This means that there are vectors x and y such that

$$\begin{aligned} & x^t XAXx > 0 \\ \text{and} \quad & y^t XAXy < 0. \end{aligned} \tag{69}$$

Then there are vectors x' and y', namely $x' = Xx$ and $y' = Xy$ such that

$$\begin{aligned} & x'^t Ax' > 0 \\ \text{and} \quad & y'^t Ay' < 0 \end{aligned} \tag{70}$$

and so A is also indefinite. The reverse direction proof is similar, but uses X^{-1} rather than X.

So the eigenvalues computed by PCG/Lanczos are not the eigenvalues of $-P(m)\big(H-R\big)P(m)$, but they still reveal whether it is positive definite.

4.3 A Simple Preconditioner

The magnetostatic force is the only complicated part of the matrix-vector multiplication. All of the other parts are block diagonal with 3×3 blocks. So an obvious first try for a preconditioner is to throw away the magnetostatic field matrix and leave everything else. Recalling that $H = H_{ms} + H_{ani}$, this would give

$$\begin{aligned} A' &= -P(m)\big(H_{ani} - R\big)P(m) \\ &= -\mathrm{diag}\big(P_i(k_i k_i^t - z_i I_3)P_i\big) \\ &= \mathrm{diag}(A_i') \end{aligned} \tag{71}$$

with $P_i = I_3 - m_i m_i^t$ and $z_i = m_i^t h_{eff\,i}$. This is easily inverted,

$$A'^{-1} = \mathrm{diag}(A_i'^{-1}), \tag{72}$$

So we need only to invert all of the A_i's. Recall that all of these computations are taking place in a 2-dimensional subspace of $\mathbb{R}^3$, so what we really want is the generalized inverse. A little algebra gives us

$$A_i' = zP_i - (P_i k_i)(P_i k_i)^t = z_i P_i - x_i x_i^t \tag{73}$$

with $x_i = P_i k_i$. Applying the Sherman-Morrison formula to this and noting that $P_i x_i = x_i$ we have

$$A_i'^{-1} = \frac{1}{z_i}\big(P_i + \frac{1}{z_i - x_i^t x_i} x_i x_i^t\big) \tag{74}$$

Now the preconditioner has to be definite, and our preconditioner is not guaranteed to be. What are its eigenvalues? We note that m_i and x_i are orthogonal, and that they are eigenvectors. We can see that the eigenvalues of A_i' are

$$\sigma(A_i') = \{z_i, z_i - x_i^t x_i, 0\} \tag{75}$$

and the eigenvalues of $A_i'^{-1}$ are

$$\sigma(A_i'^{-1}) = \{\frac{1}{z_i}, \frac{1}{z_i - x_i^t x_i}, 0\} \tag{76}$$

If these are not both positive, then the matrix will have to be modified. One possible strategy is to refuse to precondition any component having a non-positive eigenvalue. This means, that if an eigenvalue is negative, modify the matrix by setting the eigenvalue to 1 without changing the associated eigenvector. So

$$\lambda_1 = \begin{cases} \frac{1}{z_i}, & \frac{1}{z_i} > 0 \\ 1, & \frac{1}{z_i} < 0 \end{cases} \tag{77}$$

$$\lambda_2 = \begin{cases} \frac{1}{z_i - x_i^t x_i}, & \frac{1}{z_i - x_i^t x_i} > 0 \\ 1, & \frac{1}{z_i - x_i^t x_i} < 0 \end{cases} \tag{78}$$

and then

$$A_i''^{-1} = \lambda_1 P_i + \frac{\lambda_2 - \lambda_1}{x_i^t x_i} x_i x_i^t \tag{79}$$

is the inverse of the new preconditioning matrix.

4.4 The Jacobi Matrix for the Quasi-Steady Equation

We are interested in the matrix

$$J = \frac{\partial \dot{m}}{\partial m}. \tag{80}$$

If (36) is integrated in time using an explicit numerical scheme, the eigenvalues of J must lie in the stability region of the scheme for it to be (linearly) stable. The size of the stability region is inversely proportional to the size of time steps for the scheme. To be able to take large time steps, either the scheme needs to have a large stable region, or J has to have small eigenvalues. It also helps if the shape of the stable region is a good fit to the placement of eigenvalues in the complex plane.

Define $A(m) = -P(m)(H - R(m,t))P(m)$ and $b(m,t) = P(m)\dot{h}_{app}$, so that

$$\dot{m} = A^{-1}(m)b(m) \tag{81}$$

and

$$J = \frac{\partial}{\partial m}\left(A^{-1}(m)b(m) \right). \tag{82}$$

Noting that

$$\frac{\partial}{\partial m}A^{-1} = -A^{-1}\frac{\partial A}{\partial m}A^{-1}, \tag{83}$$

we have

$$\begin{aligned} J &= -A^{-1}\frac{\partial A}{\partial m}A^{-1}b + A^{-1}\frac{\partial b}{\partial m} && (84)\\ &= -A^{-1}\left(\frac{\partial A}{\partial m}\dot{m} - \frac{\partial b}{\partial m}\right) && (85) \end{aligned}$$

It can be easily shown that $\partial b/\partial m$ is symmetric:

$$\frac{\partial b}{\partial m} = \frac{\partial}{\partial m}P\dot{h}_{app} = -\mathrm{diag}(\dot{h}_{app\,i}m_i^t + m_i\dot{h}_{app\,i}^t) \tag{86}$$

which is symmetric. We can also see that $(\partial A/\partial m)\dot{m}$ is symmetric. The derivative of a symmetric matrix is symmetric and so $(\partial A/\partial m)\dot{m}$ is a linear combination of symmetric matrices, which is symmetric. So we see that J is the product of two symmetric matrices, and therfore J will have real eigenvalues. It will not be symmetric, though unless the two matrices commute.

The meaning of this is that a numerical ODE solving scheme having a low, wide stability region (such as a backward differention scheme) will allow larger timesteps than a scheme having a tall, narrow stability region. Actually, Runge-Kutta schemes (which are used in the numerical experiments in this paper) are a somewhat poor choice because of this.

4.5 Continuation Algorithms

It turns out that the numerical algorithms described in sections 3 and 4 are a simple example of a class of algorithms called *Continuation algorithms* and that more efficient and sophisticated algorithms exist than the ones used here. Some references are [9] and [10]. I used the algorithms described mainly because I didn't find out about continuation algorithms until after doing most of this work.

Two difficulties with the approach I have used here occur near switching events. The first is that $\|\dot{m}\|$ gets large and so small time steps are necessary in the quasi-steady integration. The second is that the linear system gets very ill-conditioned and so many iterations of the conjugate-gradient algorithm are required. An idea of Keller at least partly eliminates these problems is the arc-length and pseudo-arc-length parameterizations.

Another drawback to the present approach is that it treats the quasi-steady evolution strictly as the solution of the differential equation (36). and ignores the fact that at each time, the state also satisfies the algebraic equation (24). An way of exploiting this fact is to use a predictor-corrector algorithm, where the predictor is some explicit ODE solver such as an Adams method, and the corrector is one or two iterations of Newton's method.

Several public-domain continuation method software packages exist. Among them are Pitcon and Hompack, both available from Netlib. Unfortunately, the micromagnetics problem has the special feature that the quasi-steady differential equation (36) is a consistent singular system. It is not clear how to express our problem in the format that a general-purpose software package would require.

Continuation algorithms have been used in solid mechanics, see [4]. The quasi-steady state idea appears chemical reaction kinetics. An approach somewhat resembling the the one developed here for micromagnetics has been suggested in [7] and [8]. Actually, evolution by stepped applied fields can be thought of as a very crude type of continuation algorithm.

5 Numerical Results

We calculated the forward half of a hysteresis loop, switching between equations (36) and (11) as required. A periodic cubic lattice of particles was simulated. The anisotropy field of each particle was taken to be $c_i c_i^t m_i$, with c_i a Gaussian random 3-vector with mean 0 and covariance matrix $\alpha^2 I_3$, and $\alpha = 2.0$.

The magnetostatic field was taken to be

$$H_{ms}m = -\mathcal{F}^{-1}\mathrm{diag}(\hat{h}_k)\mathcal{F}m \tag{87}$$

with

$$\hat{h}_k = \begin{cases} \dfrac{kk^t}{k^t k}, & k \neq 0 \\ \dfrac{1}{3}I_3, & k = 0 \end{cases} \tag{88}$$

and $\mathcal{F}$ is the 3-dimensional discrete Fourier transform. This approximation of H_{ms} is correct for a smooth distribution of magnetization, rather than discrete particles. It was chosen mainly for convenience.

The timestepping algorithm used for the quasi-steady evolution was a low-storage fourth order Gill-Runge-Kutta scheme, with step size control using $\max(\|m_i\| - m_s)$ as an error estimate. The timestepping algorithm used for the Landau-Lifshitz time integration was RKF45, obtained from Netlib.

The linear systems of equations of the quasi-steady evolution were solved using conjugate gradient, both

with and without the preconditioning described in section 4.

Figure 3 shows the z-component of the average magnetization for the forward half of the hysteresis loop, computed using stepped applied fields. the Figure 4 shows the same hysteresis loop, calculated in two different ways. First, using the quasi-steady process described in this paper, and second with evolution by stepped applied fields, with applied field increments of 0.03. There were 512 particles in the collection. This calculation was also done using 64 particles. The results of the two calculations are similar enough that they would have been indistinguishable if they were plotted on the same scale as figure 3. Therefore figure 4 shows only a small portion of the hysteresis loop, expanded so that the differences are visible. Figure 5 compares the hysteresis loops calculated using Landau-Lifshitz integration and applied field steps of 0.03 and 0.015. The curves corresponding to stepped applied field calculations have a stepped pattern. The upper corners of the steps are the final equilibrium states of Landau-Lifshitz integrations. They are the only parts of the curve which are really parts of the hysteresis loop. It can be seen in 4 that the quasi-steady curve passes exactly throught each corner of the stepped applied field calculation. The two stepped applied field curves in figure 5 are also effectively identical.

Table 1 shows some information about these computations. The results for $n = 64$ were computed on a Sparcstation 10, and the $n = 512$ computations were done on an Iris Indigo.

I found that the Landau-Lifshitz equations seem to be somewhat stiff, and when integrating them, RKF45 seemed to take about the same size timestep regardless of the situation. This means that the CPU time consumed in the Landau-Lifshitz integration is proportional to the total number of nondimensional timeunits computed in bringing the system to equilibria. If we assume that the Landau-Lifshitz integration consumes most of the CPU time, then the number of Landau-Lifshitz timeunits computed is a reasonable estimate of the computational cost of the calculations.

We see that for $n = 64$ the quasi-steady algorithm is more efficient than the Landau-Lifshitz . However, with $n = 512$ the quasi-steady algorithm is more efficient only if Δh_{app} is sufficiently small.

In table 2 the effect of the preconditioner is shown. In the case of weak magnetostatic forces, it speeds up the quasi-steady calculations considerably. With $n = 64$ it is 3 times faster, and with $n = 512$, 4.4 times faster. In the case of stronger magnetostatic forces, the improvement is less. When $\alpha = .666$ the speedup is less than a factor of 2. This is reasonable considering that the preconditioner is based on the idea of ignoring magnetostatic forces entirely.

5.1 Switch Delays

If the particles in the collection do not interact, then choice of Δh_{app} cannot effect the dynamics of the system. However, if the particles *do* interact, then there is an event which I call a *switch delay*, and it can be disrupted if Δh_{app} is too large. Consider the picture of figure 2: The anisotropy axes of both particles are vertical. Their directions of magnetization are also both vertical. They are close enough that they have significant magnetostatic interation. The soft particle is almost ready to switch. If the soft particle were frozen and the applied field slowly increased, the hard particle would switch a little later, at field strength which we will denote h_2.

Suppose the applied field were increased slowly and continuously, or in small enough jumps. Then first the soft particle would switch, then its magnetostatic interaction with the hard particle would prevent it from switching until the applied field reached a much higher strength.

But, if the applied field were instead suddenly increased past h_2, the hard particle would switch before the soft particle had a chance to flip over and protect it. This is an error. It can be prevented by incrementing the applied field in small enough jumps, but this makes the computation more expensive.

It is not clear how common switch delays are. However, a few trends are clear. If the interparticle interaction is strengthened, then each particle has more neighbors that it significantly interacts with. This will increase the frequency of switch delays. If the applied field increments are increased in a stepped applied field calculation, then the number of switching events that an increment covers will increase, and this will increase the likelihood of a switch delay disruption.

In figure 5 shows a small part of a half hysteresis loop, computed using stepped applied fields with two different applied field increments, $\Delta h_{app} = 0.015$ and 0.03. The value of α was 2.0. There is effectively no difference between these two curves, so we can conclude that neither of these computations had any switch delay disruptions.

Figure 6 shows the same situation with $\alpha = 0.666$. This situation is much more vulnerable to switch delay disruptions, partly because the increased particle interation and partly because the coercivity is much smaller, making the applied field increments effectively bigger. This computation was done

Table 1: Number of inner (i.e. Landau-Lifshitz) timeunits required, $\alpha = 2.0$.

# of particles	algorithm	# of inner timeunits	# CPU seconds Landau-Lifshitz	# of CPU sec. Quasi-Steady	comments
64	Stepped applied fields	5109	505		$\Delta h_{app} = 0.03$
64	Quasi-steady	2172	181	158	52 switch events
512	Stepped applied fields	17736	19300		$\Delta h_{app} = 0.015$
512	Stepped applied fields	10173	11200		$\Delta h_{app} = 0.03$
512	Quasi-steady	13503	13400	7032	344 switch events

Table 2: Effect of the Preconditioner on CPU Time (measured in seconds)

CG algorithm	n	α	Quasi-Steady CPU Time
Preconditioned	64	2.0	158
No Precond.	64	2.0	53
Preconditioned	512	2.0	1593
No Precond.	512	2.0	7032
Preconditioned	512	0.666	2240
No Precond.	512	0.666	3942

three times, once with the quasi-steady algorithm, and twice with stepped applied fields. The quasi-steady curve and the stepped applied field curve with $\Delta h_{app} = 0.005$ are identical, and therefore presumably correct. The curve for stepped applied fields with $\Delta h_{app} = 0.03$ is noticeably different from the other two, suggesting that it has been corrupted by switch delay disruptions.

6 Partly Frozen Evolution

We see from the experiments that even when the hysteresis computations use the quasi-steady algorithm, the bulk of the computation is in the switching events. Part of the reason for this is the large number of them. A possible explanation for the large number of switching events is that the magnetostatic field may be weak compared to the anisotropy field.

Consider a micromagnetic system with anisotropy but no magnetostatic field. Then the particles are uncoupled. As the external magnetic field changes, individual particles switch one at a time. Since they do not interact, the switching of one particle does not effect any other particle.

Now consider a system with anisotropy and a weak magnetostatic interation. When one particle switches, it will effect a few nearby particles, and a few more distant particles that were almost ready to switch themeselves. All other particles will be almost fixed during the switching event.

Therefore, a possible strategy for reducing the computational work in a switching event is to somehow identify a set containing all of the active particles in a switching event and as few inactive particles as possible. Then the magnetization of all other particles (the "frozen particles") will be kept fixed during the switch. Calculation of the evolution of the active particles will be much less costly than that of the whole system. The magnetostatic field only needs to be evaluated at the active particles, and the contribution of the frozen particles to the magnetostatic field only needs to be calculated once at the beginning of the switching event since their magnetization is constant.

If the frozen particles are brought to equilibrium by Landau-Lifshitz evolution, their final equilibrium point will not be exactly the same as the equilibrium of the unfrozen system, since the dynamics are not quite the same. So the unfrozen system has to be brought to an unfrozen equilibrium by some means. Hopefully, the frozen equilibrium point is close enough to the true equilibrium that any iterative scheme for energy minimization will find the unfrozen equilibrium using the frozen equilibrium point as a starting point.

Figure 7 shows the L_2-norm of $\|\dot{m}\|$ against time during several switching events. The top curve is $\|\dot{m}\|$ summed over all 512 particles in the system. Also, at each timestep, particles were sorted according to $\|\dot{m}_i\|$ and $\|\dot{m}\|$ summed over the $512 - 20$ slowest-moving particles. This quantity is the bottom curve. Finally, at the beginning of each switching event, a set of 20 particles was selected according to a process that will be described in the next section. The middle

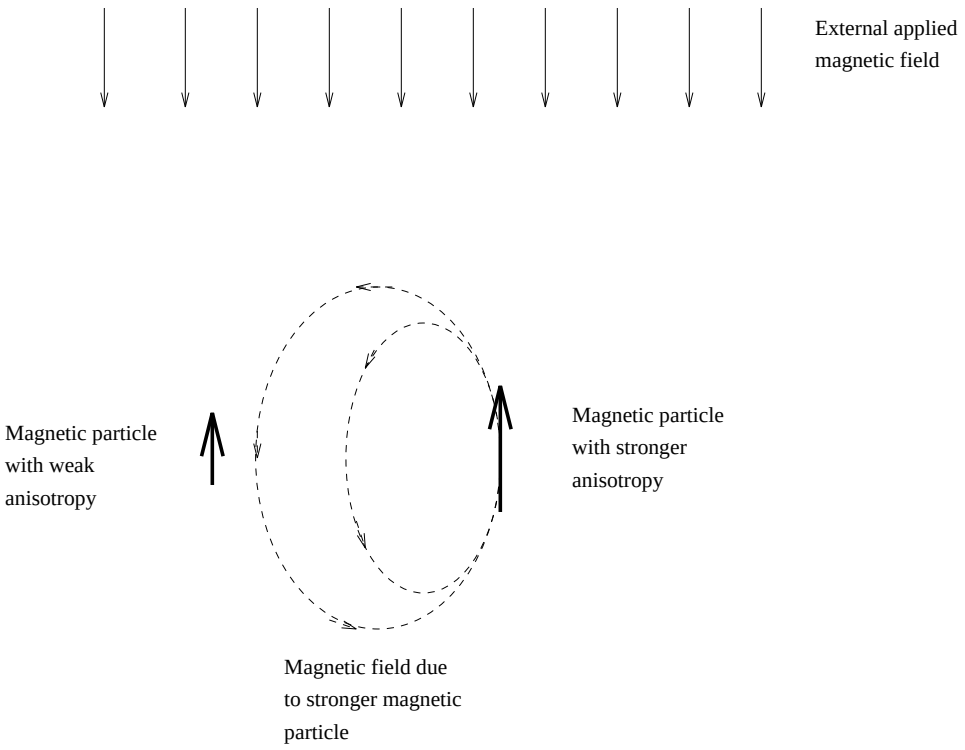

Figure 2: A switch delay

curve in the figure is $\|\dot{m}\|$ summed over all of the particles that were *not* in that set.

We can see that except for a short time at the beginning of each switching event, almost all of the activity in the system is in a few particles. In addition, the set selection strategy shown here does nearly as well as it is possible to do.

6.1 Active Set Selection Strategy

Just before the onset of a switching event, the matrix $P(H-R)P$ becomes singular. The eigenvector x associated with the zero eigenvalue is the direction in the state space that the system can move in without any increase of energy. Therefore, at least at the beginning of the switching event, this is the direction that the system will move in. We can partition x into sections corresponding to different particles, $x = \{x_i\}_{i=1}^n$, $x_i \in \mathbb{R}^3$. Then a reasonable choice of particles in the active set is to pick those particles i such that $\|x_i\|$ is largest.

An approximation of this strategy can be implemented with very little work. If a matrix A is almost singular, then for any vector b, $A^{-1}b$ is an approximation to the eigenvector corresponding to the smallest eigenvalue. Now at each quasi-steady timestep we have to compute the vector $\dot{m} = \big(P(H-R)P\big)^{-1}P\dot{h}_{app}$ several times. So just before the beginning of the switching event, $\dot{m}$ is an approximation of the eigenvector x.

On the last linear system solved before the switching event, the conjugate gradient algorithm iterates for a while, discovers that the matrix is indefinite, and stops. The partly computed $\dot{m}$ obtained from this process is used as an approximation to the eigenvector x in the active set selection strategy.

This strategy only looks at the system near the state that it is in at the beginning of the switching event. Therefore we would expect it to be less successful if the system wanders far from that point during the switch than if it stays nearby. In figure 8 some switching events from later in the computation are shown. In the last, very complicated switch shown in this figure, we see that the picked set contains all of the active particles until partway through the switching event. After this point other particles are active. The multiple peaks in $\dot{m}$ presumably mean that several particles were active in this event. So we see that if the switching event is not too complicated, this strategy works, but it can fail if the dynamics become too complex.

6.2 Fixed Point/Partly Frozen Dynamics

In this section we describe process that seems to do a fairly good job of approximating the dynamics of the micromagnetic system during switching events.

The following simple algorithm has been used by some researchers as a mechanism for driving the micromagnetic system to equilibrium. It has been used in place of the Landau-Lifshitz evolution in the evolution by stepped external fields. However, since it resembles the true dynamics only slightly, there is no guarantee that it reaches the bottom of the same basin of attraction that the Landau-Lifshitz equation does.

Fixed point iteration approach to equilibrium

$$\begin{aligned} m^0 &= \text{ initial guess} \\ \text{While } \|m^{i+1} - m^i\| &< \epsilon, \\ h_{eff} &= Hm^i + h_{app} \\ m_j^{i+1} &= \|h_{eff\,j}\|^{-1} h_{eff\,j} \end{aligned} \quad (89)$$

I now propose the following algorithm for computing the final equilibrium point of a switching event.

Fixed Point/Partly Frozen Dynamics

1) Do fixed point iteration. Continue as long as $\|m^{i+1} - m^i\|$ is monotone decreasing. Stop when it begins to increase.

2) If $\|m^{i+1} - m^i\| < \epsilon$, then stop and let m^{i+1} be the final equilibrium. Otherwise pick an active particle set according to the large entries of the vector $m^{i+1} - m^i$ and integrate the partly frozen micromagnetic system to equilibrium, using m^{i+1} as the initial condition.

3) go to 1

The idea is that as long as $\|m^{i+1} - m^i\|$ is decreasing, it is probably converging to a nearby equilibrium point. But if $\|m^{i+1} - m^i\|$ decreases and then increases, probably the system is moving toward a distant equilibrium point. If so, the fixed point iteration dynamics should be replaced by the more correct partly-frozen Landau-Lifshitz dynamics.

6.3 Convergence Analysis of Fixed Point Iteration

In this section we investigate the relationship between convergence or divergence of the fixed point iteration and stability of the equilibrium state.

We have a map which we iterate,

$$m^{j+1} = f(m^j), \quad (90)$$

with

$$f(m) = \text{norm}\big(Hm + h_{app}\big), \quad (91)$$

where

$$\text{norm}(x) = \{\|x_i\|^{-1} x_i\}_{i=1}^n. \quad (92)$$

We want to show that if the iterates of this map have a direction that they diverge in, that this direction corresponds to the particles which would be active in a switching event.

We will start by assuming that there is an equilibrium point m^* of the map, and all iterates of the map are near the equilibrium point. We expand the map in a Taylor series about m^*;

$$f(m) \approx f(m^*) + J(m - m^*) \quad (93)$$

where J is the Jacobian of $f(m)$ at m^*. Since $f(m^*) = m^*$, we have

$$m^{j+1} - m^* \approx J(m^j - m^*) \quad (94)$$

or

$$m^j - m^* \approx J^j(m^0 - m^*). \quad (95)$$

So if the iterates converge monotonically to m^*, then all of the eigenvalues of J satisfy $|\lambda_i| < 1$. If the iterates diverge, then there some eigenvalues with $|\lambda_i| > 1$. We would like to show that J has some eigenvalues outside $[-1, 1]$ iff the hessian matrix for the energy $Q = -P(H - R)P$ is not positive definite. We also want to show that the eigenvector of J associated with the divergence of $f(m)$ is at least similar to the eigenvector of Q associated with the negative eigenvalue.

First, what is J? We note that the Jacobian of norm(x) is $R'^{-1}P'$, where

$$R'(x) = \text{diag}(I_3 \|x_i\|) \quad (96)$$

and

$$P'(x) = \text{diag}\big(I_3 - \frac{x_i x_i^t}{x_i^t x_i}\big) \quad (97)$$

so the Jacobian of $f(m)$ is

$$J = R'^{-1}(h_{eff})P'(h_{eff})HP'(m). \quad (98)$$

We have some observations.

2) When $R'(h_{eff})$ and $P'(h_{eff})$ are evaluated at equilibrium points, $R'(h_{eff}) = R$ and $P'(h_{eff}) = P$.

1) R'^{-1} and $P'HP'$ are both real symmetric matrices, so J has real eigenvalues.

So actually

$$J = R^{-1}PHP. \quad (99)$$

We need to understand the relationship between J and Q. First, let us look at the possibility that J

might have an eigenvalue > 1, so $P - J$ would have a negative eigenvalue.

$$\begin{aligned} P - J &= P - R^{-1}PHP && (100)\\ &= -R^{-1}P(H - R)P && (101)\\ &= R^{-1}Q. && (102) \end{aligned}$$

This has the same eigenvalues, and similar eigenvectors as

$$R^{-\frac{1}{2}}(P - J)R^{\frac{1}{2}} = R^{-\frac{1}{2}}QR^{-\frac{1}{2}}. \quad (103)$$

Now we invoke the Sylvester inertia theorem, which says that (103) is has a negative eigenvalue iff Q has a negative eigenvalue.

6.3.1 A Failure Mode for Fixed Point Iteration

Now we must consider the possibility that J has an eigenvalue less than -1. This possibility is unrelated to the definiteness of Q. This is the same as asking whether $P + J$ is positive definite.

$$\begin{aligned} P + J &= P + R^{-1}PHP && (104)\\ &= R^{-1}P\big(H + R\big)P && (105) \end{aligned}$$

and this is positive definite iff $Q' = P\big(H + R\big)P$ is positive definite. We would like to show that Q' is pd because this means that the iterative map diverges only if the equilibrium point m^* is saddle point of the energy.

Unfortunately, this is not true in general. It is possible for Q to be positive definite while Q' is indefinite. We will now construct a situation where that is true.

First pick a state m^* according to a criterion that will be described later. Now let the applied field be

$$h_{app} = -Hm^* \quad (106)$$

and so the effective field

$$h_{eff} = Hm^* + h_{app} = 0. \quad (107)$$

This satisfies the equilibrium equation $P(m^*)h_{eff} = 0$ and so m^* is a (not necessarily stable) equilibrium state.

Notice that since $P(m^*)m^* = 0$, we can modify the applied field by

$$h_{app} \leftarrow h_{app} + \alpha m^*,\ \alpha \in \mathbb{R} \quad (108)$$

and m^* is still an equilibrium state. This would make

$$R = \mathrm{diag}\big(I_3\|h_{eff\,i}\|\big) = \mathrm{diag}\big(I_3\|\alpha m_i\|\big) = \alpha I. \quad (109)$$

We can use this to make m^* a stable equilibrium. Suppose PHP is indefinite and let its eigenvalues be

$$-h_0 = \lambda_1 < \ldots < \lambda_{2n} = h_1. \quad (110)$$

Then let $\alpha = h_1 + \epsilon$, where ϵ is some very small positive number. Then the eigenvalues of $Q = P(\alpha I - H)P$ are

$$\epsilon = \alpha - h_1 < \ldots < \alpha + h_0 \quad (111)$$

and the eigenvalues of $Q' = P(\alpha I + H)P$ are

$$\alpha - h_0 < \ldots < \alpha + h_1, \quad (112)$$

and so if $h_0 > h_1$ then Q is p.d. while Q' is not.

We can use our choice of m^* to arrange this. Recall that in the case of our test problem,

$$H = -\mathcal{F}^{-1}\mathrm{diag}(\|k\|^{-2}kk^t)\mathcal{F} + \mathrm{diag}(c_ic_i^t) \quad (113)$$

where $\mathcal{F}$ is the 3-d Fourier transform matrix, k is the 3-vector identifying each Fourier mode, and c_i is the anisotropy vector of the i-th particle. The first term is the magnetostatic field, and is negative definite. Moreover, even with arbitrary locations and shapes of particles, the magnetostatic part of H will be negative definite. The second term is the anisotropy, and it is positive definite. Now suppose we make the equilibrium magnetization of each particle point in the direction of its easy axis, i.e.,

$$m^* = \mathrm{norm}\big(\{c_i\}_{i=1}^n\big). \quad (114)$$

Then

$$P\,\mathrm{diag}\big(c_ic_i^t\big)P = 0 \quad (115)$$

and the biggest positive eigenvalue of PHP is zero, fulfilling the eigenvalue condition.

We can draw some rough conclusions from this (very formal) argument. For almost all states of the system, the anisotropy would be only partly masked by the projector P. If the positive eigenvalues of H are large compared to its negative eigenvalues, then it is more difficult for this eigenvalue condition to be met. Since the positive eigenvalues come from anisotropy forces, and the negative ones from the magnetostatic field, we would expect that if the anisotropy forces are strong compared to the magnetostatic ones, this situation would happen only rarely.

6.3.2 Conclusions

One final task is to show that components of the dominant eigenvector of J that have large entries are the same components that have large entries in the negative eigenvector of Q. If $R = qI$, then the eigenvectors of J are the eigenvectors of Q. If not, then we

would expect that the simple structure of R would let us prove something.

What have we shown? If fixed point iteration converges, it converges to a stable equilibrium point of the micromagnetic system. If it diverges, then it may be because the nearby equilibrium point is unstable. If so, the direction of divergence probably bears some resemblence to the unstable eigenvector of Q. Or it may diverge because of the failure mode described in the last section. In this case, the direction of divergence probably has no connection with the stability of the micromagnetic system. In the following section, we used fixed point iteration and just hoped for the best.

6.4 Numerical Results Using Partly Frozen Evolution

In this section, half hysteresis loops were calculated using the quasi-steady evolution equation. Switching events were calculated in different ways. In each calculation, the switch was calculated twice, once using the Landau-Lifshitz evolution, and once using either the fixed point/partly frozen evolution algorithm described in section 6.2, or fixed point iteration alone. If the final equilibrium state of the Landau-Lifshitz evolution was the same as the final equilibrium state of the second algorithm, then the switch was called a sucess. If they were different, then it was called a failure. The result of the Landau-Lifshitz calculation was used to continue the quasi-steady evolution.

These calculations were repeated for three different strengths of the particle anisotropy. The anisotropy vectors c_i were Gaussian random 3-vectors with mean zero and covariance matrix $\alpha^2 I_3$, and α took the values 2.0, 1.0, and 0.666. The strength of the magnetostatic interaction was kept fixed. The partly-frozen evolution used 20 particles. There were 512 particles in the whole system.

Table 3 lists the results of these computations. Large values of α correspond to weak magnetostatic interactions between particles, and small α means strong interaction. We see that with small α/strong interaction there are fewer switching events, suggesting that particles switch in larger groups.

When $\alpha = 2.0$, both the fixed point iteration and the fixed point/partly frozen algorithm find the correct final equilibrium almost all the time. For smaller α, the performance of both algorithms degrades. In all cases, the fixed point/partly frozen algorithm works better than fixed point iteration alone. When $\alpha = .666$, many of the switch failures for fixed point iteration alone were due to the fixed point iteration refusing to converge.

The next question to ask is, what do the hystersis loops look like? Figure 9 shows the case where $\alpha = 2.0$. The switching events have been calculated three different ways: Landau-Lifshitz evolution, fixed point iteration alone, and the fixed point/partly frozen algorithm. The three curves are almost exactly alike. Figure 10 shows the $\alpha = 0.666$ case. The stronger particle interactions make this a more difficult test. The curve for fixed point iteration alone was not shown because the fixed point iteration refused to converge for many switching events. The two remaining curves are not exactly alike, there are small but visible differences. In both figures, 20 particles are used in the partly-frozen switching.

One final question is how much cpu time do these different algorithms use? This is addressed in table 4. All computations were done on an Iris Indigo. Two things are clear. The first is that the fixed point iteration and partly frozen stages are very quick, compared to the other parts of the computations. The second is that the quasi-steady evolution is surprisingly time-consuming, requiring more than half of the time required by the Landau-Lifshitz evolution.

7 Conclusions

We have found that the quasi-steady/fixed point/partly frozen algorithm that we developed in this paper is considerably more efficient than the Landau-Lifshitz /stepped applied fields algorithm, at least in the case of strong anisotropy. With $\alpha = 2.0$ and $\Delta h_{app} = 0.03$, the stepped applied field algorithm used 11,200 CPU seconds to calculate the half hysteresis loop. In comparison, the preconditioned quasi-steady evolution required 1600 CPU seconds, and the fixed point/partly frozen treatment of the switching events required 850 CPU seconds, for a total of 2450 CPU seconds. This is more than 4.5 times faster, and there are no visible differences between the results computed by the two algorithms.

When anisotropy is weaker, our approach is less successful. The preconditioner used in the quasi-steady evolution is less effective, and the Landau-Lifshitz equations are less stiff, making the whole partly-frozen idea less advantagous.

The traditional approach of evolution by stepped applied fields has the problem of possibly causing switch delay disruptions. The only way of being sure that a calculation doesn't have any disruptions is to repeat it with two different values of Δh_{app} and making sure that they give the same results. The quasi-steady evolution approach does not have this problem, which is a significant advantage.

Table 3: Number of Switch Failures

switch algorithm	α	# of switch failures	total # of switches
fixed-point/partly frozen	2.0	6	344
fixed-point	2.0	11	344
fixed point/partly frozen	1.0	10	207
fixed point	1.0	20	207
fixed-point/partly-frozen	.666	11	158
fixed-point	.666	91	158

Table 4: CPU time for different parts of the computations (seconds)

switching event algorithm used	α	Quasi-Steady	Landau-Lifshitz	Fixed-Point	Partly-Frozen	Total
Landau-Lifshitz	2.0	7000	13,300			20,300
Fixed-point/Partly-Frozen	2.0	7400		600	255	8,250
Fixed-point only	2.0	7250		1100		8,350
Landau-Lifshitz	0.666	3003	4620			7,623
Fixed-point/Partly-Frozen	0.666	3200		660	410	4,270

The difficulty of approximating switching events with a partly-frozen strategy depends very much on the strength of the magnetostatic interaction. In the case of weak interaction, almost any process that brings the micromagnetic system to equilibrium will give the same results as the Landau-Lifshitz evolution. With stronger interactions, much more care is required. Fixed point iteration often did not work at all, refusing to converge. The fixed point/partly frozen algorithm gave results that were quite close to the Landau-Lifshitz, and at much lower cost.

The efficiency of the continuation algorithm described in sections 3 and 4 of this paper depends dramatically upon whether a preconditioner is used in the conjugate gradient calculations. Without it, even computations that used the fixed point/partly frozen algorithm to compute the switching events would have only a modest speedup over calculations that used stepped applied fields and Landau-Lifshitz evolution. But the preconditioner makes the whole calculation several times faster. Also, if the most modern ideas in continuation algorithms were built into the quasi-steady part of the code, the resulting calculations should require even less CPU time.

Acknowledgments

I would like to acknowledge Tom Hoffend, Martin Vos, Roger Anderson, and Rob Brott, for many helpful discussions.

References

[1] William F. Brown, Jr., *Micromagnetics*, Interscience Publishers, 1963.

[2] Tom R. Hoffend, Jr., *Foundations of the Landau-Lifshitz Model and Related Numerical Algorithms for Study of Magnetization Reversal in Particulate Recording Media*, IMA Preprint Series, to appear

[3] Marshall Sparks, *Ferromagnetic Relaxation Theory*, McGraw-Hill, 1964.

[4] E.I. Grigolyuk, V.I. Shalashilin, *Problems of Nonlinear Deformation*, Kluwer Academic Publishers, 1991.

[5] Gene H. Golub, Charles F. Van Loan, *Matrix Computations, 2nd Edition*, Johns Hopkins University Press, 1989.

[6] J.H. Wilkinson, *The Algebraic Eigenvalue Problem*, Clarenden Press, 1965.

[7] Gordon F. Hughes, *Magnetization Reversal in Cobalt-Phosphorus Films*, J. Appl. Phys., V.54, no. 9, p.5306-5313 (1983)

[8] Marc A. Pinto, *Morphology of Micromagnetics*, Phys. Rev. B, V. 38, no. 10, p.6824-6831 (1988).

[9] E.L. Allgower, K. George, *Numerical Continuation Methods*, Springer-Verlag, 1990.

[10] H. B. Keller, *Numerical Methods in Bifurcation Problems*, Springer-Verlag, 1987.

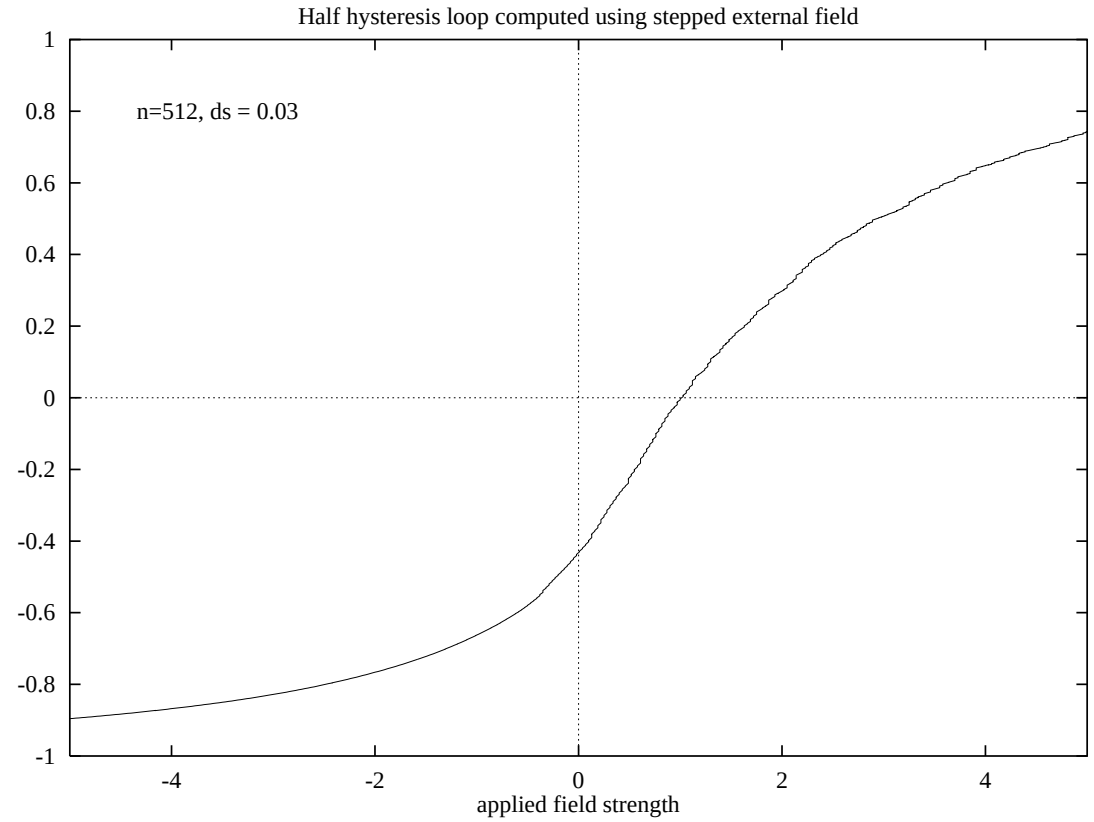

Figure 3: The forward half of a hysteresis loop.

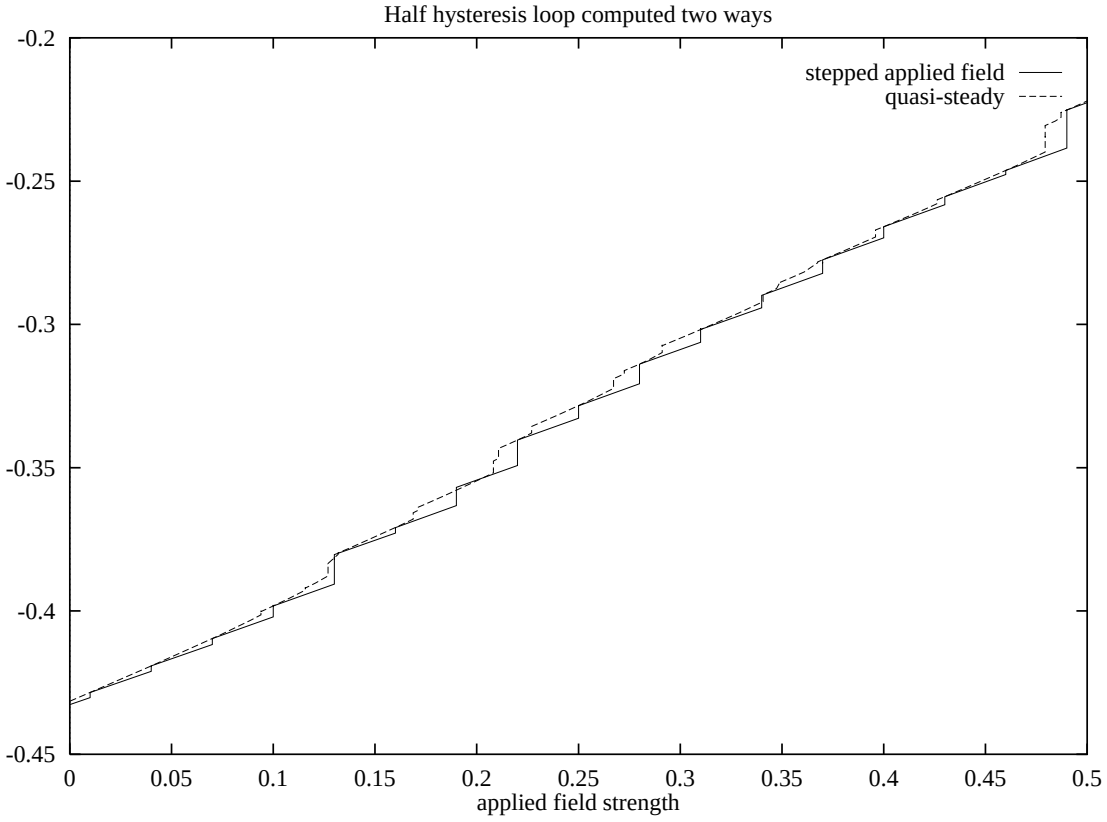

Figure 4: Comparison of Quasi-steady and stepped applied field computations

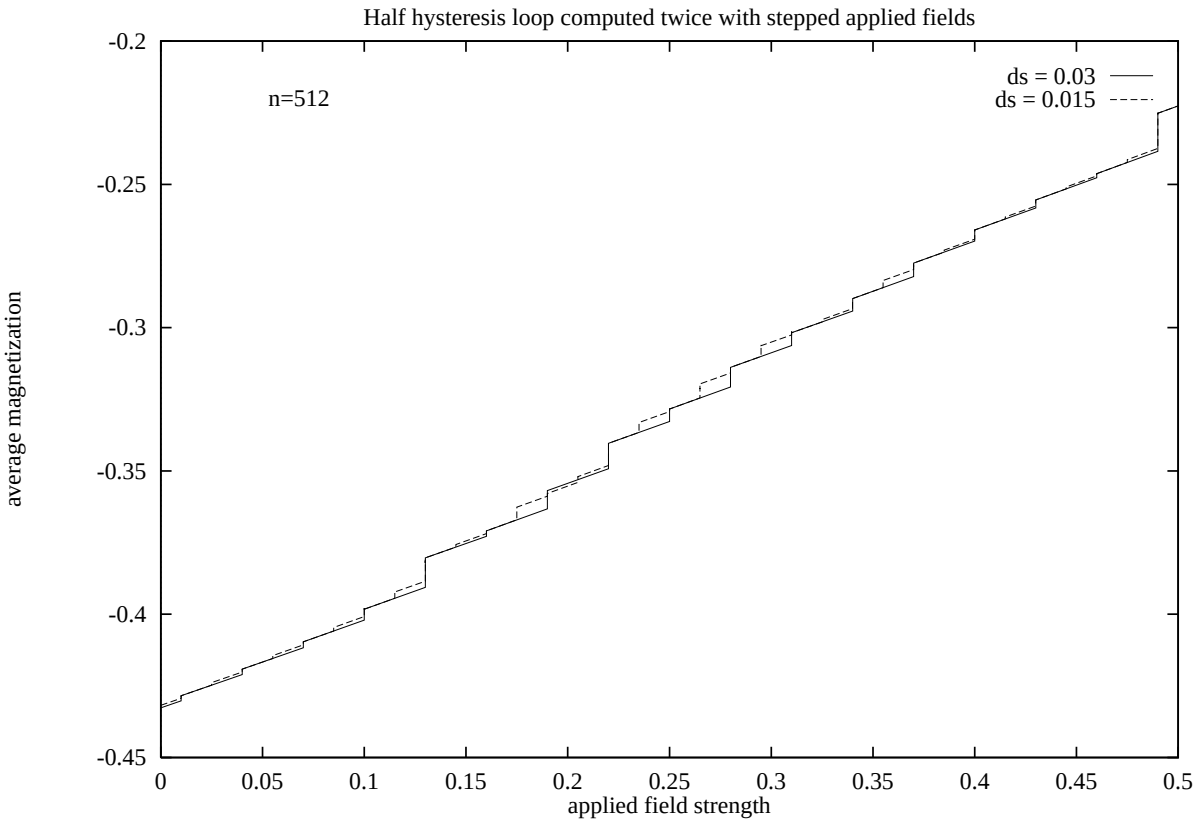

Figure 5: Comparison of stepped applied field computations with two different stepsizes, $\alpha = 2.0$

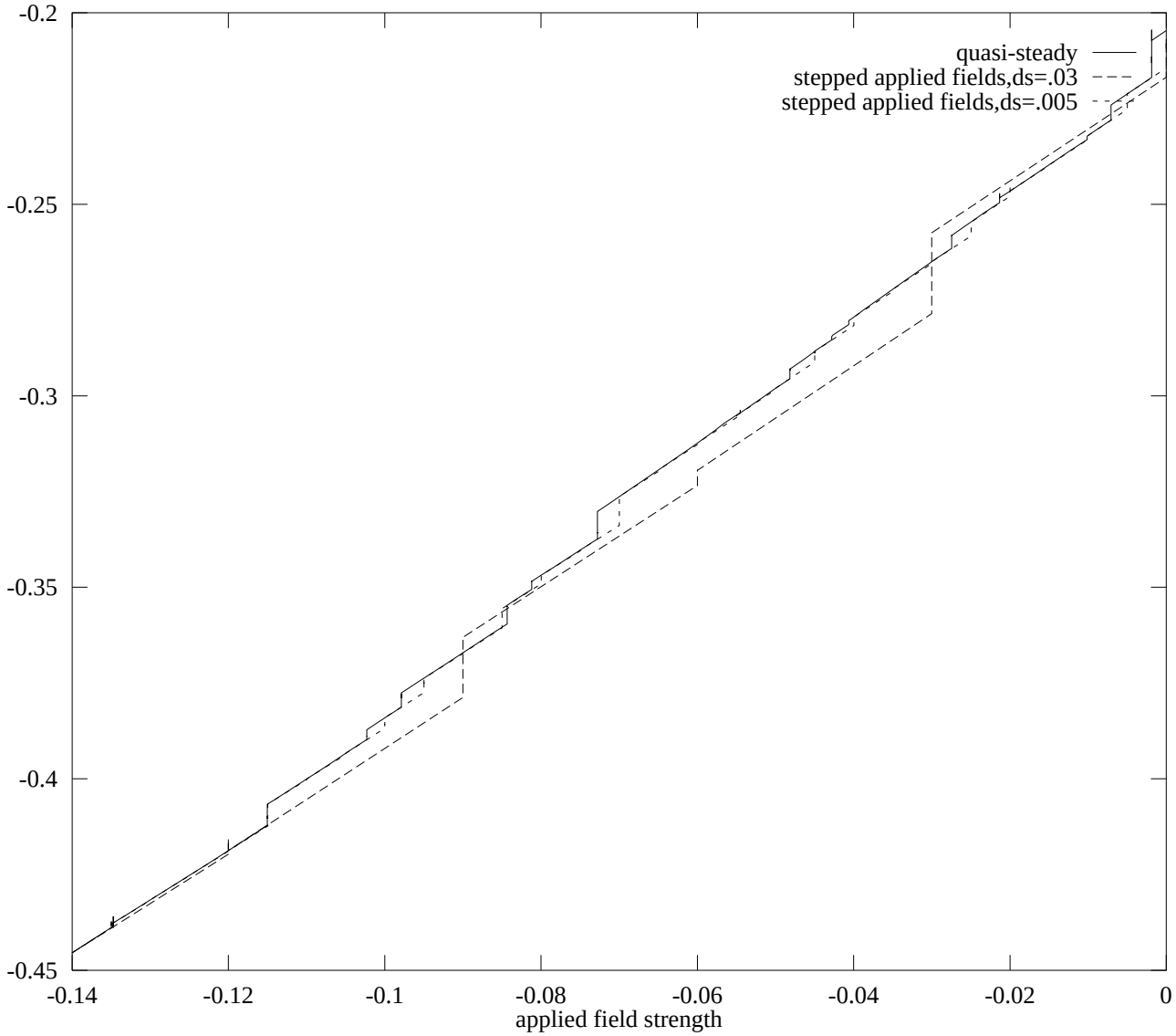

Figure 6: comparison of quasi-steady calculation and two stepped applied field computations when $\alpha = 0.666$

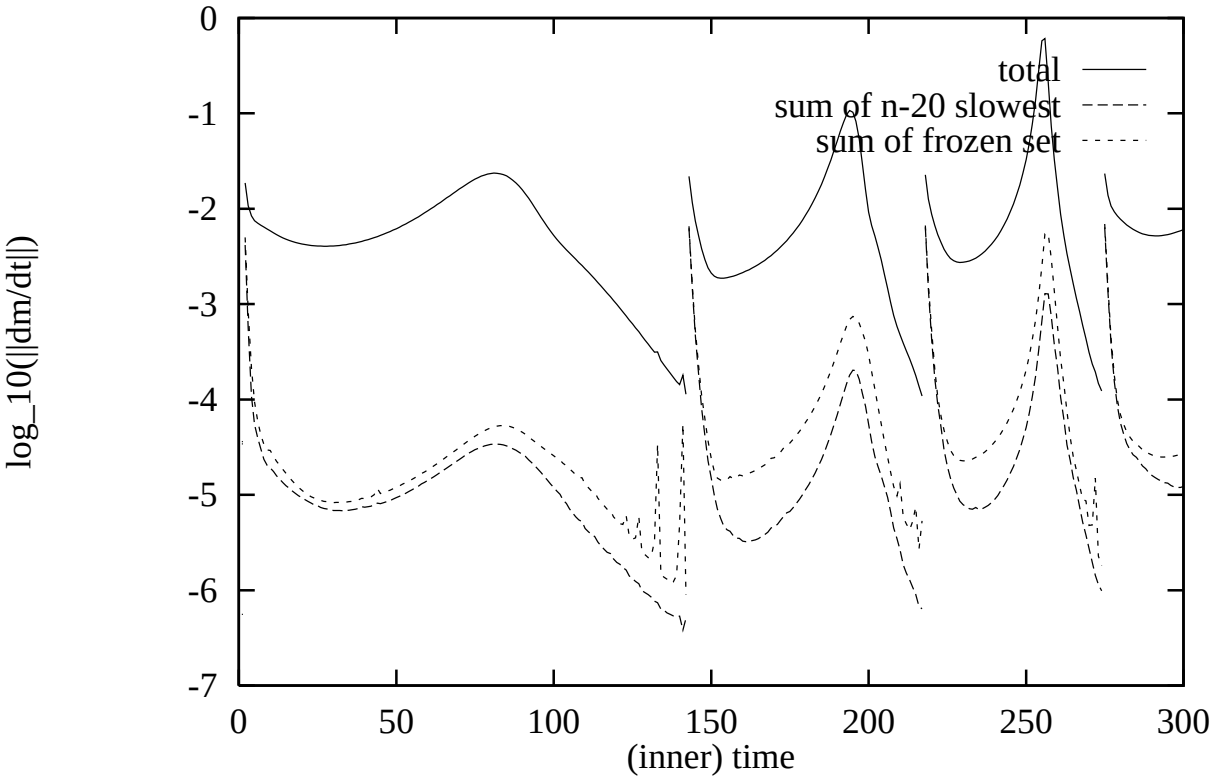

Figure 7: The norm of $\dot{m}$ summed over different sets of particles, for the first few switching events.

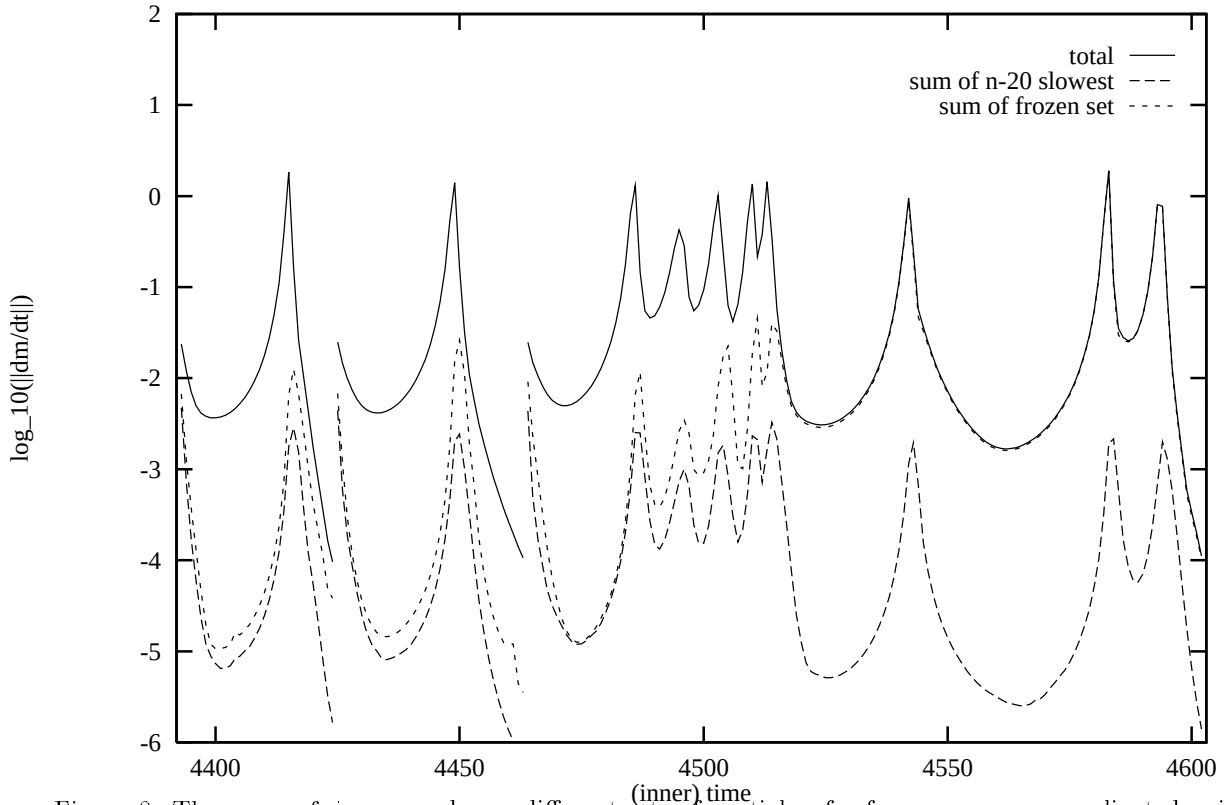

Figure 8: The norm of $\dot{m}$ summed over different sets of particles, for for some more complicated switching events.

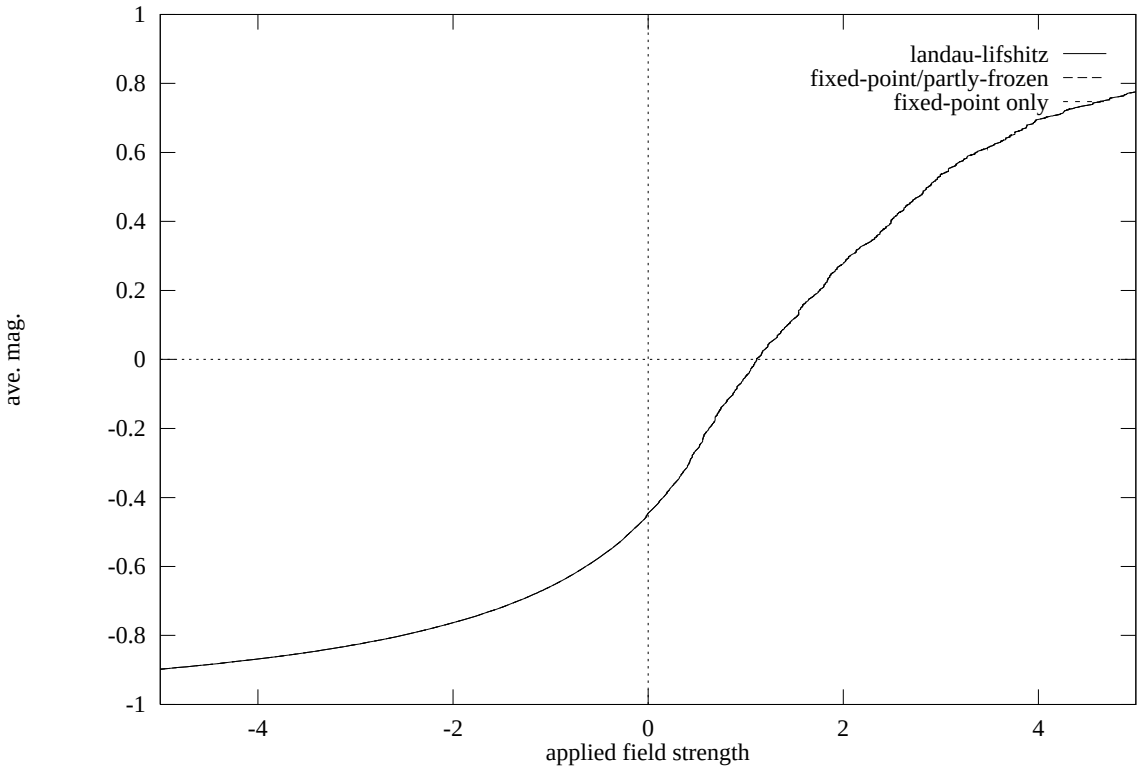

Figure 9: Half hysteresis loops computed using quasi-steady evolution and three different treatments of the switching events; Landau-Lifshitz evolution, the fixed-point/partly-frozen algorithm, and fixed point iteration alone. The value of α is 2.

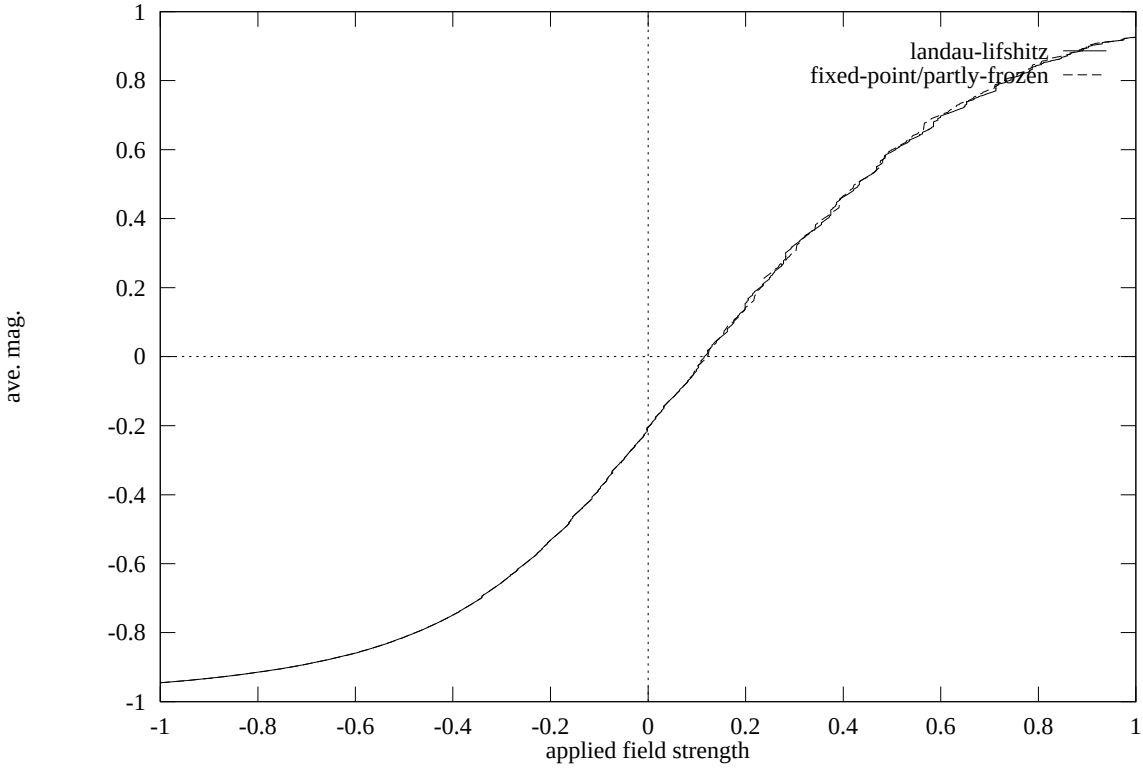

Figure 10: Half hysteresis loops computed using quasi-steady evolution and two different treatments of the switching events; Landau-Lifshitz evolution and the fixed-point/partly-frozen algorithm, the $\alpha = 0.666$ case.