Summer 2021

Two Aspects of Magnetic Nanoparticle Self-Assembly on Thin-Film Multilayers: Custom Media Properties and Accurate Determination of Nanoparticle Anisotropy Constant

Sara L. FitzGerald

Two Aspects of Magnetic Nanoparticle Self-Assembly on Thin-Film Multilayers: Custom Media Properties and Accurate Determination of Nanoparticle Anisotropy Constant

by

Sara L. FitzGerald

Bachelor of Science
Tulane University 2013

Submitted in Partial Fulfillment of the Requirements

for the Degree of Doctor of Philosophy in

Physics

College of Arts and Sciences

University of South Carolina

2021

Accepted by:

Thomas Crawford, Major Professor

Richard Creswick, Committee Member

Yanwen Wu, Committee Member

Thompson Mefford, External Committee Member

Tracey L. Weldon, Interim Vice Provost and Dean of the Graduate School

Dedication

To all the strong women in my life who have helped me get to this point and to everyone who fostered my love of learning along the way: family, teachers, and friends. Special thanks to my mom, Gwen, for three decades of love and support. I love you, Mom.

Acknowledgments

I would like to thank for their support and assistance in this research: Dr. Karen Livesey of University of Newcastle, Australia; Casey Chalifour of the University of Colorado, Colorado Springs; Dr. Thompson Mefford and Zichun (Tony) Yan of Clemson University; Dr. Thomas Crawford and Rahman Mohtasebzadeh of the University of South Carolina; and Drs. Bryan Chávez and Cory Dolbashian.

I would also like to acknowledge the traditional stewards of the land where the research was performed: the people of the Eastern Band of Cherokee and Congaree.

This material is based upon work supported by the National Science Foundation under Grant No. NSFIIP1556031 and through MADE in SC. MADE in SC is supported by the National Science Foundation Award No. OIA-1655740. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

Abstract

ZFC/FC moment versus temperature measurements are a common technique to determine magnetic properties of nanoparticles. In this work, I varied both applied field strength and nanoparticle concentration to study resulting changes in blocking temperature, T_B, and anisotropy constant, K. T_B and K values were obtained using both existing and new analytic methods. Accurate determination of these parameters helps researchers optimize the use of magnetic nanoparticles for a variety of applications, including magnetic heating, drug delivery, and magnetic field-directed self-assembly. For magnetic self-assembly particularly, not only nanoparticle properties, but also the magnetic properties of the substrate alter the pattern produced. In addition to examining the effects of field and concentration on ZFC/FC data, this work begins the journey toward custom magnetic recording media, optimized for magnetic self-assembly of nanoparticles.

Table of Contents

List of Tables

List of Figures

Introduction

Generally, self-assembly creates an environment in which particles naturally aggregate to form repeatable, precise shapes. Self-assembly of magnetic nanoparticles can be used to fabricate optical, medical, and electronic devices. Nanoparticles can be used as biosensors [50], to target and kill cancer cells through heating [6], and as compact electrical components [11]. In magnetically driven self-assembly, the shapes are dictated by the pattern drawn on a magnetic substrate [15, 46, 47].

One method to self-assemble nanoparticles utilizes colloidally stable superparamagnetic particles in solution. Perpendicular recording media (PRM) from commercial hard drive disks, which is recorded with a magnetic pattern, is placed into the ferrofluid, and the nanoparticles collect on the surface due to the large magnetic field gradients between differently magnetized regions. Commercial media can be written with arbitrary magnetic patterns causing the assembled nanoparticles to form controllable, complex designs [15, 46, 47]. The properties of the nanoparticles, the fluid they are immersed in, and the substrate all contribute to the final arrangement [46]. Magnetic media has been used for data storage for decades and is highly engineered for this purpose, but has only been leveraged for magnetic self-assembly since the early 2010s [15].

Commerical media has stringent requirements for data storage which are unneccessary for self-assembly. Commercial hard drives must hold data for ten years, they must be writable at sub-nanosecond timescales, and they must produce a field large enough to be read, despite the ever-decreasing bit size desired for higher recording density [41, 34]. These requirements can all be loosened for custom media used in

self-assembly, since our write process uses much slower head speeds; our media need only be stable for weeks or months; and our field strength is dictated by its effect on nanoparticles, rather than its interactions with the commercial read head.

In this magnetic field-directed self-assembly the particles used and their ionic environment can be modified to alter the complex patterns produced, but the media field properties are not tunable when using commerical media. The media properties change the external field felt by the nanoparticle and will alter the dynamics of magnetic self-assembly. In chapters 2 and 3, we discuss progress made to fabricate and test prototypes of custom magnetic media for use during self-assembly, and in 6.2, I will suggest extensions of this work for future research.

In addition to the properties of the substrate, the magnetic properties of the particles themselves can affect the assembly process. There are many types of measurements which probe a system's magnetic response, but one of particular interest for nanoparticles is the Zero-Field Cooled/Field Cooled (ZFC/FC) magnetization versus temperature measurement. One can extract important information about size distribution and anisotropy from ZFC/FC data, both of which are critical for applications of NPs even beyond magnetic field-driven self-assembly. ZFC/FC involves first cooling the sample down with no applied field and then warming in a small applied field (Zero-Field Cooled), then repeating the same cooling and warming cycle, but with field applied throughout (Field Cooled). For a superparamagnetic nanoparticle sample, during the initial cooling, the randomly oriented magnetizations of the nanoparticles are "frozen in." The temperature at which a nanoparticle moment becomes "unfrozen" and aligns with the external applied field is called its blocking temperature, T_B. The blocking temperature can be used to find the anisotropy constant of a nanoparticle (NP) sample. This constant, K, is particularly relevant for nanoparticle performance in magnetic heating, which has been proposed as a method of targeted treatment for malignant tumors [32]. In addition to the practical applica-

tion of nanoparticles, the accurate calculation of K from experimental data relates to the deeper understanding of nanoparticle magnetism. The large discrepancy between K of a superparamagnetic sample, as calculated from an incorrect blocking temperature, and that of bulk materials has led to theories of large surface effects in these nanoscale magnets, which may in fact have less bearing on K if it is calculated with the correct blocking temperature [24].

Historically the blocking temperature of a collection of particles in a measured sample was assumed to be the peak of the ZFC magnetization, but, as will be discussed in section 5.1, this is highly inaccurate [3]. Several more accurate methods for determining the blocking temperature of a collection of particles will be discussed in the remainder of chapter 5, and we will discuss the results of those methods for measurements performed on samples of varying NP concentration and with various magnetic field strengths applied during the measurement. These parameters in particular seem to be selected arbitrarily by researchers, even in very recent work, and the impacts of concentration and applied field strength on the blocking temperature and anisotropy constant results are discussed in chapters 5 and 6.

A nanoparticle's magnetic response to an external applied field, as in magnetic field-directed self-assembly, is affected by the anisotropy constant of the nanoparticle. This suggests that by choosing nanoparticles based on their effective anisotropy, as can be determined from ZFC/FC measurements, one could further control the dynamics of magnetic self assembly. This is beyond the scope of this work, but a rudimentary calculation of this relationship is given in section 2.2, and suggestions for future research are discussed in section 6.2.

Chapter 1

Nanoparticle Magnetism

1.1 Single-Domain Nanoparticles

A typical bar magnet is made up of multiple domains, all with individual magnetic moments, which can align in an external field either by rotation or by the motion of domain walls. Once all the domains in the magnet are aligned with the external field it is said to be saturated [8].

In the case of most nanoparticles and the recording layer of granular perpendicular recording media (PRM) (discussed in chapter 3), the magnets are small enough that they contain only a single domain. In this case, domain wall motion can be neglected, and the magnetic moments align with an external field through rotation, either of the particle itself or of the moment. We will restrict our discussion to rotation of the moment only and neglect the physical rotation of the particles. The measurements presented here are only of immobilized, single-domain magnets, since in the case of magnetic media, the grains are static in the thin film media, and in our measurements on superparamagnetic nanoparticles, they have been encased in a polymer to prevent physical rotation.

The "easy axis" of a material is the direction in which the magnetization (moment per volume or per mass) will preferentially lie in the absence of an applied field (Fig. 1.2). This easy axis results from properties of the material, making the energy anisotropic [8]. There are several forms of anisotropy, but the relevant types for our systems are shape anisotropy and magnetocrystalline anisotropy. The magnetocrys-

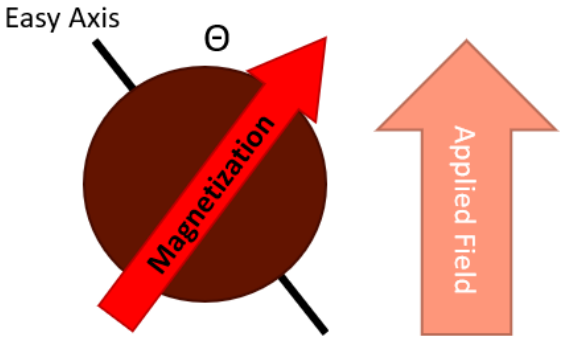

Figure 1.1 Depiction of a single-domain nanoparticle, with easy axis, magnetization vector, and applied field indicated.

talline easy axis is determined by the arrangement of atoms in a material, while the shape anisotropy is determined by the boundary of the material [8].

For a single domain magnet, the anisotropy energy can be described by:

$$E_A = K \sin^2 \theta \tag{1.1}$$

[8] where K is the anisotropy constant and θ is the angle between the easy axis and the magnetization direction (Fig. 1.2). For a single-domain magnet, the magnetization vector is of length M_s and will lie preferentially along the direction where E_A is at a minimum, which is the easy axis. The "hard axis" is typically orthogonal to the easy axis, and is an unstable equilibrium point in the anisotropy energy, where E_A is a maximum.

In the presence of an external field, H, at an angle α with the easy axis, the potential energy is given by:

$$E_p = -HM_s \cos(\alpha - \theta) \tag{1.2}$$

[8]. The total energy is then the sum of the anisotropy energy and the potential energy. Depending on the relative values of K and M_sH, one or the other energy term will dominate. The potential energy will have minima or "wells" in which the system will prefer to lie (Fig. 1.2).

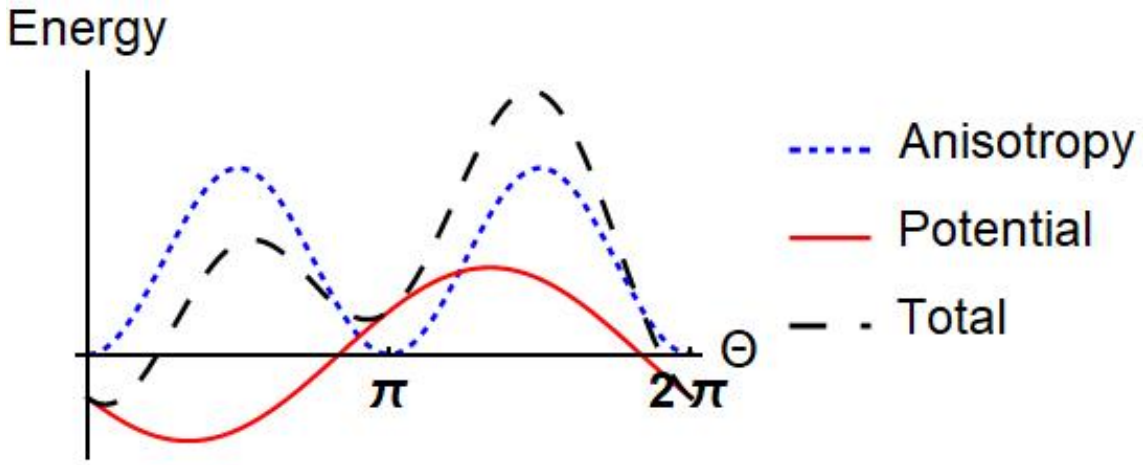

Figure 1.2 Energy of a single-domain particle versus the angle between the magnetization and the easy axis. E_A is the blue line with small dashes, E_p is the solid red line, and E_{total} is the dashed black line.

If the magnetization is measured with varying field applied along the hard axis ($\alpha = \pi$), it will be an s-shaped curve with no hysteresis. If measured along the easy axis ($\alpha = 0, 2\pi$), it will be a square loop. The intersection of this hysteresis loop with the field axis is the coercivity of the magnet, defined as the field required to return the magnetization to zero after saturation with a positive field.

1.2 Néel-Arrhenius Law

Now we can examine the effect of thermal energy on the system, given a potential described in section 1.1. In order for the single-domain nanoparticles to "flip" their magnetization from one potential energy well to another, there must be sufficient thermal energy to surmount the barrier between the wells or the shape of

the potential wells must change (such as with a stronger applied field that alters the potential).

The thermal energy of a single-domain magnet is k_BT, where k_B is Boltzmann's constant and T is the temperature. If the thermal energy is sufficient to overcome the energy barrier, then the nanoparticle magnetization can flip to lie along the direction of a neighboring energy well (Fig. 1.3) [8, 24].

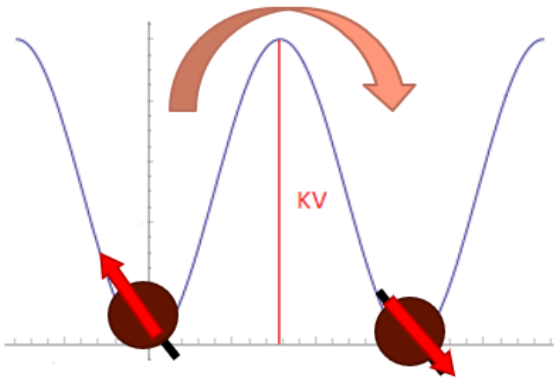

Figure 1.3 Graphic of a nanoparticle "flip."

A simple model for the mean time between flips for a collection of single-domain particles is given by the Néel time. The average Néel time, τ, is given by the Néel-Arrhenius Law:

$$\tau = \tau_0 exp\left(\frac{Kv}{k_BT}\right) \tag{1.3}$$

[2] where K is the anisotropy constant and v is the particle volume. Typically the attempt time, τ_0, is taken as 10^{-9} s [3]. However, there is some evidence that τ_0 can depend on measurement parameters and material properties. According to recent efforts by our colleagues Dr. Livesey and Casey Chalifour at the University of Newcastle, Australia, and the University of Colorado, Colorado Springs, respectively, the value of τ_0 can be shown to depend on the applied field, the anisotropy constant, and

other material properties. In Eq. 1.3 the exponent is the ratio between the thermal energy and the barrier height, Kv, which assumes that $\mu_0 M_s H \ll K$. An extension of the Néel-Arrhenius expression in an attempt to capture the effects of stronger fields is given in section 5.5. For the present, we will discuss the simpler, field-independent version provided above.

In the case of a ZFC/FC measurement, when τ is equal to the measurement time, τ_m, we can solve Eq. 1.3 for T to find the temperature required to flip the magnetization, or the blocking temperature, T_B:

$$T_B = \frac{Kv}{k_B \ln\left(\tau_m/\tau_0\right)} \tag{1.4}$$

For our purposes, the value of τ_m depends on the rate of temperature change during the measurement, as well as the blocking temperature, however, since it is inside the natural logarithm, we can estimate it, and it will not change T_B significantly. Now that we have discussed the basic concepts of single-domain magnets, we will move to discussing field-directed self-assembly of these single-domain magnetic nanoparticles in the next chapter.

Chapter 2

Self Assembly of Magnetic Nanoparticles

2.1 Magnetic Field-Directed Self-Assembly

In this and the following chapter, we will look at the process of magnetic field-directed self-assembly and the magnetic media used as a template. We will also discuss our attempts to create a custom version of this media. We begin in section 2.2 by discussing the effects of the external field on the nanoparticles during assembly, then in section 2.3 we will briefly demonstrate the potential effect of nanoparticle anisotropy on the process, and finally in section 2.4 we will discuss magnetic media, its desired properties, and those properties' qualitative effects on the external field felt by particles during assembly.

Magnetic field-directed self-assembly has been used to achieve 100 nm feature sizes with self-assembled Fe_3O_4 nanoparticles [15, 47]. In our current assembly process, the magnetic nanoparticles begin in solution. The particles are colloidally stable, so they are less prone to aggregate before reaching the surface of the media. The ferrofluid is placed in contact with a hard disk drive that has been recorded with a magnetic pattern which is designed to attract the nanoparticles into predetermined shapes. The nanoparticles collect at the interface between oppositely magnetized regions in the media (Fig. 2.1). In perpendicular recording media (PRM), these magnetizations are perpendicular to the plane of the disk, either "up" or "down." The interface between oppositely magnetized bits is called a "transition," and its properties are largely determined by the media properties [44].

2.2 Physics of Field-Directed Self-Assembly

During deposition, magnetic nanoparticles in the solution come close to the surface of the magnetic recording media, which has been prepared in advance with the desired pattern. There are two main methods of exposing the ferrofluid to the media surface. The simplest method is to place droplets of the ferrofluid on the surface of the media and let the particles settle due to gravity and the attractive forces of the media. The more complicated method is to use a specialized apparatus to hold the media vertically and stir it in the ferrofluid with the surface of the media perpendicular to the direction of travel. As the particles in the fluid settle on the surface or as they are forced close enough by stirring, they will be most attracted to preferred regions of high field gradient, dictated by the stray field of the recorded bits in the media.

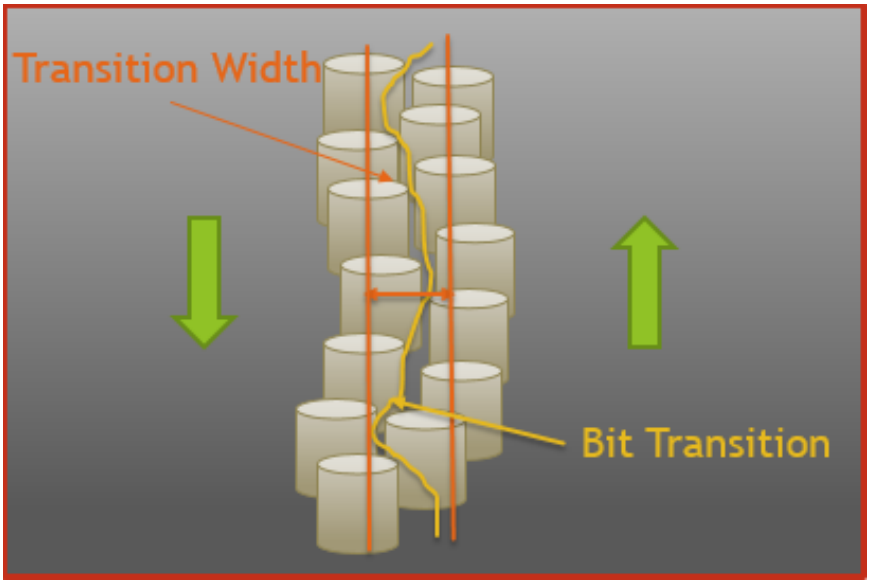

Figure 2.1 Cylindrical grains in the recording layer at the transition between differently magnetized regions. The grains are magnetized "up" and "down", and the bit transition (yellow line) follows the grain boundaries (width indicated in orange).

The magnetic force on a particle, due to the stray field of the media, depends on the gradient of the magnetic field at the center of the particle, $\nabla\mathbf{H}$, which is largest at a transition, as well as the magnetization, $\mathbf{M_p}$, and volume, V_p, of the nanoparticle [13]:

$$\mathbf{F_m} = \mu_0 V_p \left(\mathbf{M_p} \cdot \nabla\right) \mathbf{H} \tag{2.1}$$

The magnetic force on the nanoparticles must also overcome the repulsive forces between particles. Their colloidal stability leads to a spring-like opposing force between particles, as they are brought closer together by the magnetic field gradients. These forces compete, and the particles move to minimize their energy during the self-assembly process.

With a generous recording density of 1 Tbit/in^2 and a grain radius of 3.5 nm, each bit contains approximately 13 grains [41]. However, realistic packing of the grains is not precise in granular media, and bit transitions meander along the grain edges (Fig. 2.1). Larger variation in grain size or large separation between grains lead to decreased sharpness in the transition between up and down magnetizations. As the transition between bits becomes less sharp, the field gradient is weaker. Suggestions for future researchers to harness the transition properties and create custom media are given in section 6.2, but for now we will focus on more general properties of the media.

2.3 Role of Effective Anisotropy in Self-Assembly

Since the particle dynamics are related to both the media stray field and the nanoparticles' magnetic properties, as seen in the dependence of the magnetic force on $\mathbf{M_s}$ and $\mathbf{H}$ in Eq. 2.1, we now move to the effect of nanoparticle properties on $\mathbf{F_m}$.

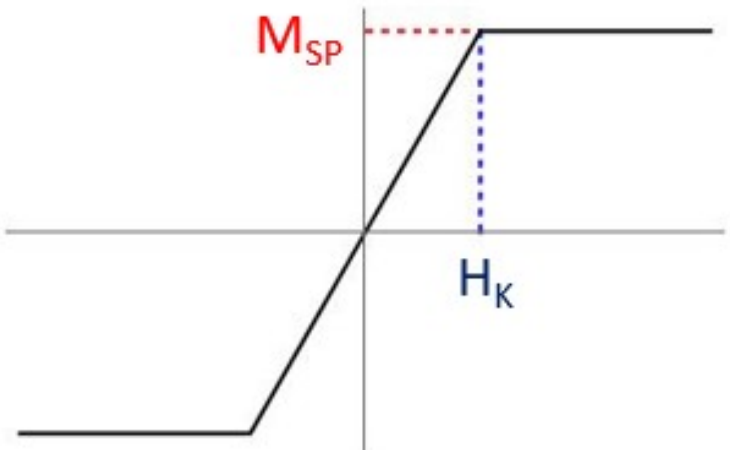

Figure 2.2 Plot of NP magnetization versus field, based on the piecewise definition.

From [13], we can model a nanoparticle's magnetization versus applied field as follows,

$$M_p = \begin{cases} M_s, & |H| > H_k \\ \chi_p H, & |H| < H_k, \end{cases} \tag{2.2}$$

where H_k is the anisotropy field, or the field required to saturate the nanoparticle's magnetization, M_s is the particle saturation magnetization, and χ_p is the linear susceptibility of the nanoparticles when the field is less than H_k (Fig. 2.2). We can represent the anisotropy field as $H_k = 2K/M_s$ [8], where K is the anisotropy constant. Thus, the unsaturated moment can be written as:

$$M_p = \begin{cases} M_s, & |H| > \frac{2K}{M_s} \\ \frac{M_s}{H_k} H = \frac{M_s^2}{2K} H, & |H| < \frac{2K}{M_s}. \end{cases} \tag{2.3}$$

Plugging these equations into Eq. 2.1, we find that for unsaturated nanoparticles the force becomes:

$$\mathbf{F_m} = \mu_0 V_p \frac{M_s^2}{2K} (\mathbf{H} \cdot \nabla) \mathbf{H} \tag{2.4}$$

If the anisotropy constant of the nanoparticles increases, the magnetic force is lessened, since the external field must be greater to saturate their magnetizations and their response to the external field is weaker. Thus, knowing the anisotropy constant of nanoparticles used in assembly can help tune the force, and potentially alter the patterns produced. As we will see chapter 5, accurately determining the anisotropy constant is not necessarily straightforward. We will stop short of testing the relationship in equation 2.4, but the suggestion that deliberately changing K can in turn change $\mathbf{F_m}$ provides new possibilities for tuning self-assembled patterns.

2.4 Magnetic Media's Role in Self-Assembly

Now that we've discussed the physics of magnetic field-driven self-assembly and the dependence of magnetic force on both media and particle properties, let's give more specific information about magnetic media and how we might go about changing its stray field by changing material properties and media structure.

Perpendicular Recording Media (PRM) entered commercial production in the late 2000s, and has enabled data storage on on the order of 1 Tbit/in^2 [41]. For use with modern recording heads, magnetic media must consist of a soft magnetic underlayer (SUL) beneath a hard magnetic recording layer (Fig. 2.3). The SUL should have in-plane anisotropy to guide the field of the write or read head, while the recording layer must have uniaxial anisotropy perpendicular to the media surface to store the recorded bits of data.

2.4.1 Recording Process and Write Head

In order to use magnetic media for data storage or as a template for self assembly, we must be able to create a magnetic patern on the surface of the media. To do this, we use a (perpendicular) write head, which "writes" by changing the

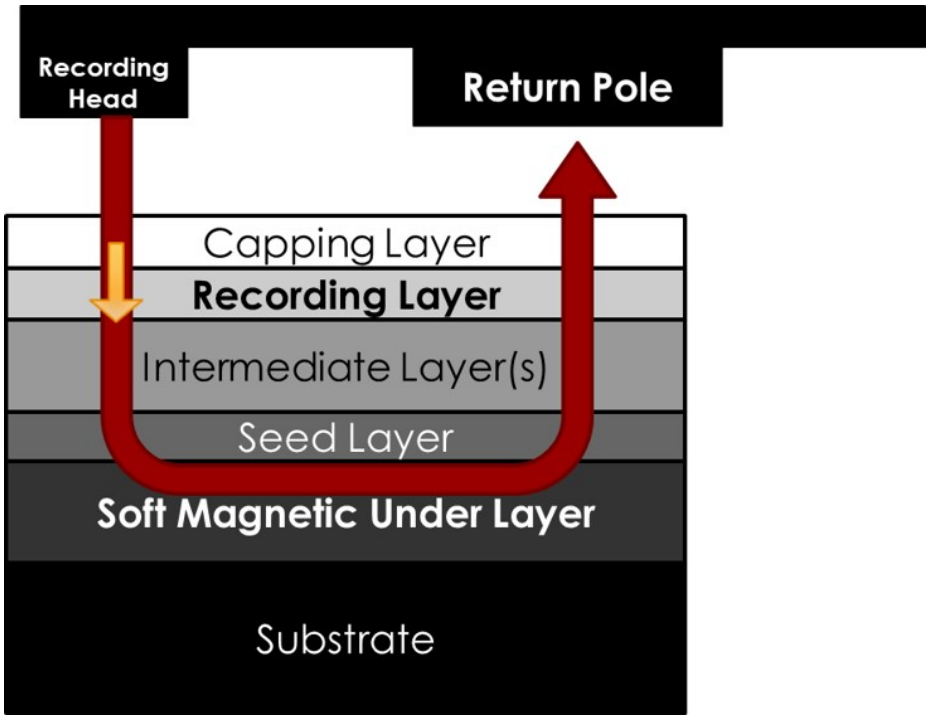

Figure 2.3 A cross-section of the media stack below the write head, where the write field passes from the write head through the recording layer perpendicularly, then parallel through the soft magnetic underlayer, and finally is collected by the return pole.

magnetization of the recording layer in sections to either "up" or "down," relative to the media surface.

The write head, in its most basic form, is an electromagnet with one end being the write pole and the other the return pole, both oriented toward the media surface (Fig. 2.3) [45]. Most commerial recording heads utilize thin film magnets as the poles. The write pole provides a large field to align the moment of the recording layer in sections or "bits." The field then forms a closed loop through the SUL and back to the return pole. The readback of written bits is measured with a nearby read head, which detects the transition between differently magnetized regions using a tunnel junction (TMR) or a giant magnetorististive (GMR) sensor [45]. A transition from

"up" to "down" or vice versa will register as a change in the signal, which is recorded as a 1, but no change in moment direction in a given region will be recorded as a 0.

The magnetic media and the recording head must have complementary properties in order to work effectively. The head must provide a field larger than the coercive field of the recording layer to overcome the energy barrier and flip its magnetization. In addition, the SUL's anisotropy must lie sufficiently in-plane, so that the field forms a closed loop and no stray field demagnetizes nearby bits. Pinning sites in the SUL can also contribute to noise during readback of bits [34, 41].

2.4.2 Recording Layer

As mentioned previously, the recording layer holds the data in PRM. This makes it crucial to the functioning of the drive, and it is what creates the template for magnetic nanoparticles in magnetic-field driven self-assembly. The read head measures changes in the stray field produced by magnetized regions of the recording layer, and magnetic nanoparticles are pulled to the regions of rapid change in stray field from "up" to "down" orientation. Thus, the recording layer must have good out-of-plane anisotropy to sustain this perpendicular magnetization. Good materials are those with a strong magnetocrystalline easy axis that can be oriented out-of-plane. The crystalline properties of this layer have a large effect on its recording performance. In the granular media discussed in this work, the recording layer is composed of grains that can have shape anisotropy in addition to magnetocrystalline anisotropy, since they are approximately cylindrical, with the long axis pointing perpendicular to the surface of the media.

Popular materials for the recording layer include cobalt and cobalt alloys, since these materials can be grown to have hexagonal close-packed (hcp) crystal structure, where the magnetocrystalline easy axis is along the c-axis of the crystal (Fig. 2.4). In

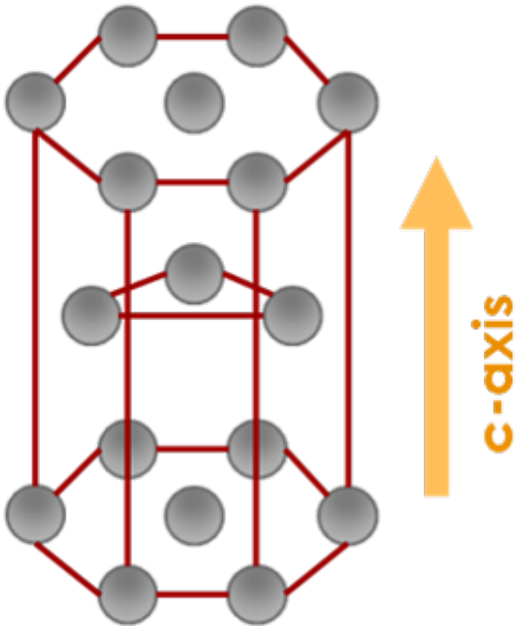

Figure 2.4 Atoms (gray) arranged in a hexagonal close packed structure, with c-axis indicated by the orange arrow.

PRM, the Co or Co-alloy grains should have 0001 crystal structure, with the c-axis oriented perpendicular to the film plane [34, 41].

As noted, the shape anisotropy also dictates the recording layer properties, and therefore, grain size has a large effect on the signal-to-noise ration (SNR) of the media, as well as the media's long-term performance. As discussed in Section 1.2 single-domain magnets can flip their magnetization direction, given enough time or thermal energy to overcome the barrier height. We can examine Eq. 1.3 to see how thermally stable the media grains will be, given their size and storage conditions, assuming that there are no interactions between neighboring grains.

Because the mean switching time varies rapidly with respect to particle volume, a small reduction in the grain size drastically decreases the stability of the media. Industry standards for media stability are on the order of 10 years [34]. Using

equation 1.3 a grain diameter of 10 nm, a temperature of 300 K and an anisotropy constant of 1.45 $\times$ 10^6 erg/cc [41], this gives a stability of nearly 2,000 years. The exponential dependence of switching time on particle volume imposes strict lower bounds on grain size for data storage, to ensure a commercial drive does not spontaneously erase over the lifetime of a computer. However, since we are not constrained to maximize recording density, we have the freedom to change the grain size, as long as the media remains stable over the timescale of the self assembly, which need only be days or weeks.

We employ CoCrPt as our recording layer material for its accessibility and high c-axis alignment [41]. In CoCrPt media, the Pt in the recording layer serves to increase the coercivity and the nucleation field of the recording layer [20]. These attributes of recording media have a large effect on the writability and thermal stability of bits. The chromium in the recording layer tends to migrate to the grain boundaries and segregate the Co grains, which can reduce their exchange interactions and lead to more stable magnetization. Annealing aids the Cr in migrating out from the center of the grains. This additional grain segregation also influences the stray field of the media and the field gradients. Grains which are too closely packed will tend to interact, and may demagnetize neighboring grains in the same bit. Those interactions may hinder sharp transitions between "up" and "down" if the more stable configuration is a canted moment. Co-sputtering the CoCrPt with SiO_2, B_2O_3, or other oxide, or sputtering the layer in an oxygenated environment can further isolate the grains [17, 19, 33, 5, 43, 21]. This is discussed more in Appendix A.

2.4.3 Soft Magnetic Underlayer

The desired traits of the soft magnetic underlayer (SUL) are dependent on the head used. This layer is designed to facilitate recording by directing the write field in a closed loop back to the return pole (Fig. 2.3). Imperfections and pinning in the

SUL can decrease the SNR of the media, and impede the write process [34]. Common materials for the SUL in modern media are CoFe-alloys [41].

In this work we will focus mostly on samples with a crystalline, NiFe (Permalloy) SUL, and the challenges in measuring the magnetic response of those samples. However, samples were also made with an amorphous CoFeB SUL, and results of tests on those samples are mentioned briefly in Appendix A.

The Permalloy SUL must have specific structural properties (good crystallinity in the 111 orientation), to promote the in-plane anisotropy necessary for writing the custom media [14]. The layers deposited on top of the Permalloy SUL must set the template for the recording layer. For information about the other layers in the custom media stacks, see Appendix A.

Since at this point the reader should have a general background in these magnetic systems, we will transition now to our specific samples and results. In the following chapter we will discuss inherent challenges associated with characterizing these systems when both an SUL and recording layer are present. We will also present a technique to bypass this difficulty in samples with a crystalline SUL.

Chapter 3

Custom Magnetic Media

3.1 Thin Film Sample Preparation

As discussed in the previous chapter, thin film magnetic media is comprised of stacks of thin layers of material (both magnetic and nonmagnetic), which are traditionally optimized to facilitate writing and reading of data. The magnetic layers are particularly important, and the nonmagnetic layers are chosen to enhance the magnetic properties of neighboring materials. The custom media described in this work contains two magnetic layers: the SUL and the hard magnetic recording layer. The SUL has in-plane or parallel anisotropy to guide the recording head field, and the hard recording layer has out-of-plane or perpendicular anisotropy.

The layers of our custom media are deposited by magnetron sputtering with a sysem from AJA International, which employs an anode and cathode (a sputtering gun) to create a plasma of argon atoms at the surface of the material to be deposited, called the sputtering target material (Fig. 3.1). The plasma bombards the target material and removes atoms, which are dispersed around the sputtering chamber at a particular sub-atmospheric pressure set by vacuum pumps and a controllable influx of Ar gas. Some of those target material atoms reach the chosen substrate, in this case SiO_2. The target material atoms will coat the substrate surface at a rate determined by material properties and sputtering conditions, such as the power supplied to the sputter guns and the Ar pressure in the sample chamber. For additional samples and results, see Appendix A.

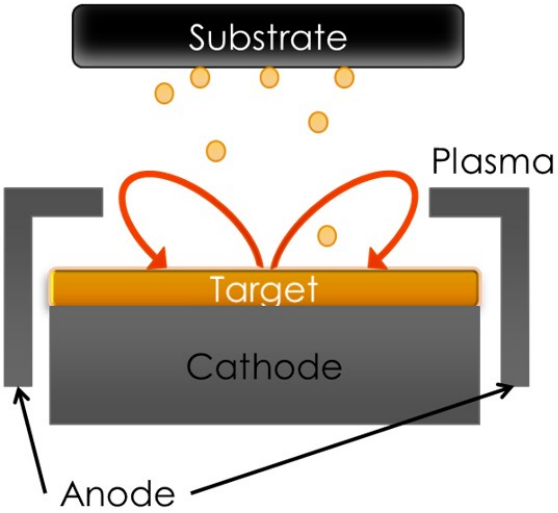

Figure 3.1 Cross-section of magnetron sputtering gun.

As shown in chapter 2, the magnetic properties of the media are important for performance, both in reading and writing data, and also as a template for self assembly. We have discussed the way material properties can change the magnetic properties, and in the following sections, we will discuss the magnetic measurements performed to characterize the samples. Since these structures are relatively complex, there are challenges with separating the contributions from different layers during certain measurements. We have found one way in particular to overcome this issue, which will be presented in the last section of this chapter and will close out our discussion of magnetic media.

3.2 Measurements with a VSM

Magnetic measurements of both the thin film multilayer media (and the NP samples described in later chapters) were performed with a Vibrating Sample Magnetometer (VSM) produced by Quantum Design as an option added to their Physi-

cal Property Measurement System (PPMS). The PPMS includes a superconducting niobium-titanium alloy magnet housed in a dewar of liquid helium. The sample chamber is a tube running vertically down through the helium dewar, separated from the dewar by a region of vacuum. The liquid He keeps the Nb alloy in the superconducting phase, and an external power supply charges the magnet to the desired field.

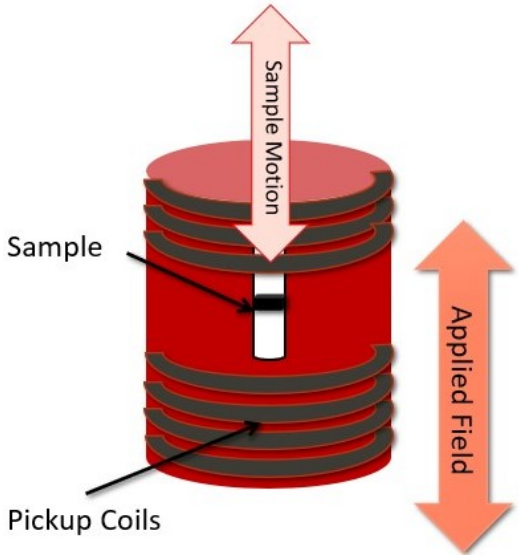

Figure 3.2 Graphic of the sample in VSM coilset.

The VSM consists of a motor, a magnet, and wound coils of wire (pickup coils) surrounding the sample (Fig. 3.2). It operates on the principle of Faraday's law that a changing magnetic field through a loop of wire produces a current in the wire. The Nb alloy solenoid creates an external field, which induces a magnetic moment in the sample, while the motor vibrates it within the pickup coils. The changing magnetic field produced by the sample's motion induces an EMF in the wire, as follows [30]:

$$\varepsilon_{emf} = -\frac{d\phi_B}{dt} = -A\frac{dB}{dt}. \tag{3.1}$$

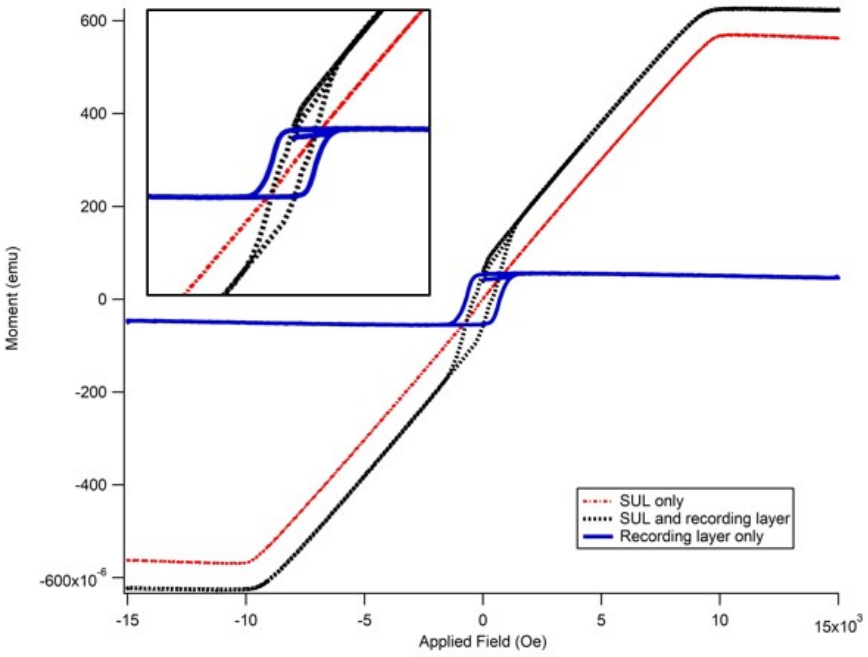

Figure 3.3 VSM data taken from a set of custom samples. The SUL (red, dashed) and recording layer (blue solid) signals combine to form a pot-bellied curve (black, dashed).

In the Quantum Design PPMS, the motor is attached at the top of the sample chamber, and the sample is threaded down to the bottom of the sample chamber on a thin rod. The pickup coils at the bottom of the sample chamber have a small opening, just wide enough to allow vertical sample vibration by the motor.

One drawback of the VSM is that it measures net magnetic moment, so for the composite samples discussed in this work, we must rely on overall measurements of the system and cannot use this technique to measure the magnetic response of isolated parts of the system. For example, we cannot separate the contribution of a single nanoparticle from a sample with many particles with the VSM, and we cannot easily separate the signal of only one layer of the custom PRM with both SUL and recording layer, especially since the magnetic moment of the SUL is much larger than that of the recording layer (Fig. 3.3).

3.3 Measurements with MOKE

Since it is difficult, and sometimes impossible, to separate the contributions of different parts of a magnetic system from the VSM signal, we have also used an alternate method, the Magneto-Optical Kerr Effect (MOKE) to examine the magnetic response of the recording layer in our custom PRM. This is one method to work around the complex magnetic structure of magnetic thin film media, and another will be described in the last section of this chapter.

MOKE is an optical technique used to measure the change in magnetization at the surface of a sample, which has been used to measure many types of nanoscale magnetic materials [1, 10]. In a MOKE setup, there is a beam of light incident on the sample surface, and with changes in the applied field, the polarization of the reflected light changes. The incident beam is linearly polarized, which is a superposition of right- and left-hand circularly polarized light. The different circularly polarized pieces interact with the medium in such a way that, in the reflected beam, one piece becomes elliptically polarized, and the resulting superposition is canted slightly [12]. By using differential detection with two photodiode detectors and a beam splitter, we can measure only the Kerr rotation of the reflected beam, which is proportional to the magnetization [1].

There are several types of MOKE measurements, all described by the relative orientation of the applied field to the plane of the incident light. We will discuss the results of Polar MOKE (or PMOKE) measurements on our thin film media samples.In PMOKE, we apply a magnetic field perpendicular to the sample surface and direct a laser normally incident on the film. The magnetization produced in the sample will cause a slight polarization change in the reflected light. Our setup allows us to observe the behavior of the media in fields over 5 kOe. Because the Kerr rotation is a small signal, we improve the SNR by taking many runs and averaging them.

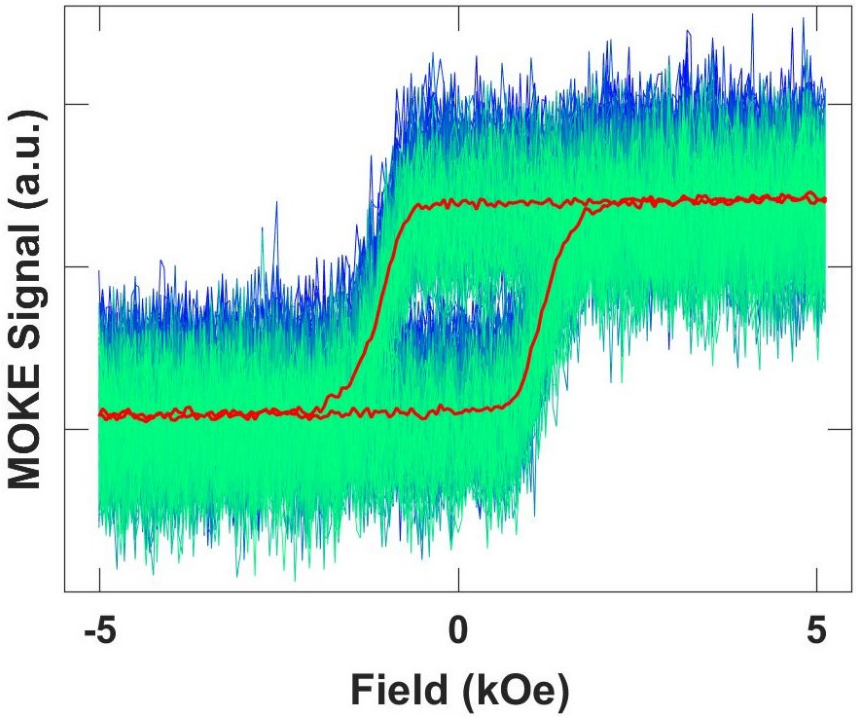

Figure 3.4 Polar MOKE signal, showing 200 low SNR runs (blue to green gradient) which have been averaged to improve the SNR (red).

One of the advantages of using MOKE to measure magnetic response is that the measured region of the sample is small and controllable. Unlike VSM, the signal produced in the MOKE measurement is only that of the illuminated region, and can therefore be used more easily on nanoscale magnets [1]. In the case of PRM, MOKE allows us to measure the magnetic response without detecting signal from the SUL, which is typically much larger than that of the recording layer due to its larger thickness. Polar MOKE signal from samples with an NiFe SUL and CoCrPt recording layer shows coercivities on the order of 1 kOe. This indicates that the Co grains are oriented with c-axis mostly perpendicular to the plane of the sample.

The Polar MOKE signal also suggests that the laser is penetrating only the first few layers of media, since there is no large soft magnetic signal contribution from

the SUL that obscures the magnetic behavior of the recording layer. By probing only the top few layers of the sample with a laser, the signal obtained is primarily from the recording layer, meaning that we avoid the drawbacks of the VSM measurement. In addition, the Polar MOKE signal saturates at a much lower field than NiFe does, further indicating that the signal is only from the CoCrPt layer.

However, it is much more difficult to measure the SUL *in situ*, since it is buried within the media stack and not illuminated with the laser. In order to test the SUL, it would require expensive and time consuming alterations to the samples, such as carefully removing the substrate material from the back of the sample to access the SUL from behind. In the next section we will explore an alternative method to measure the SUL and recording layer separately and compare this to measurements of the complete media stack.

3.4 Non-Magnetic Layer Substitution

As mentioned in the previous sections, characterizing these multilayer stacks can be difficult with the VSM, since it measures the total moment of the sample, from both the SUL and the recording layer. In addition to MOKE, there is another method to separate these contributions for media with a crystalline SUL (like NiFe in our case) that relies on replacing the SUL with a material that is similar in structure, but nonmagnetic. By nonmagnetic, we refer to materials with either a para- or diamagnetic response that has a much smaller saturation value than the recording layer. The contribution to the VSM signal will then be negligible during our measurement.

Replacing the SUL, which we call "non-magnetic layer substitution," is only necessary for samples with a crystalline SUL, since the structure of the layers on top of a crystalline SUL typically depend on the structure of the SUL as a template for their crystallinity. Thus we must chose a nonmagnetic material that has the same

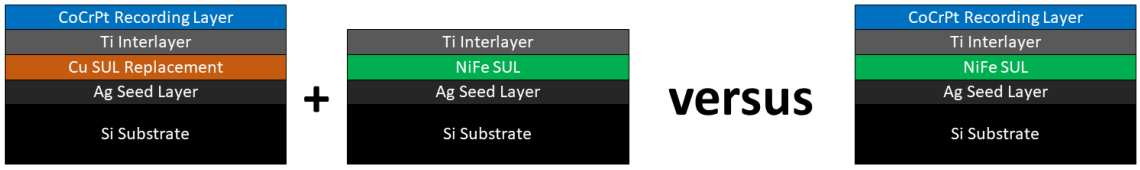

Figure 3.5 Graphic of nonmagnetic layer substitution. From left to right: the recording layer-only sample with Cu in place of NiFe, the SUL-only sample with NiFe and no recording layer, and the full sample with both magnetic layers.

unit cell structure and similar lattice parameters to be certain that the recording layer grows with the necessary out-of-plane anisotropy.

We have used this method to see if the recording layer and SUL may be interacting during the measurement. A simplified graphic of the layer substitution comparison is shown in Fig. 3.5. The sample with both SUL and recording layers is our "full" sample, the sample omitting the recording layer is referred to as "SUL-only," and the sample whose SUL is replaced with a nonmagnetic material is called "recording layer-only," since its VSM signal will be only that of the recording layer.

In our case the replacement material is Cu, which is non-magnetic but has similar crystal structure (111) and lattice parameters to NiFe [14]. Since the crystalline properties of the substitute material are similar to that of the SUL, we can be reasonably sure that the recording layers of both samples are identical. This is confirmed by MOKE measurements on the full sample and the recording layer-only samples, which are indeed very similar (Fig. 3.6), and as discussed in the previous section, our PMOKE setup measures only the recording layer. In addition, VSM data shows good out-of-plane properties for the recording layer-only samples (Fig. 3.7). Since the Cu replacement does not noticeably alter orientation of the CoCrPt easy axis, it is a suitable substitute for NiFe to overcome VSM data acquisition challenges.

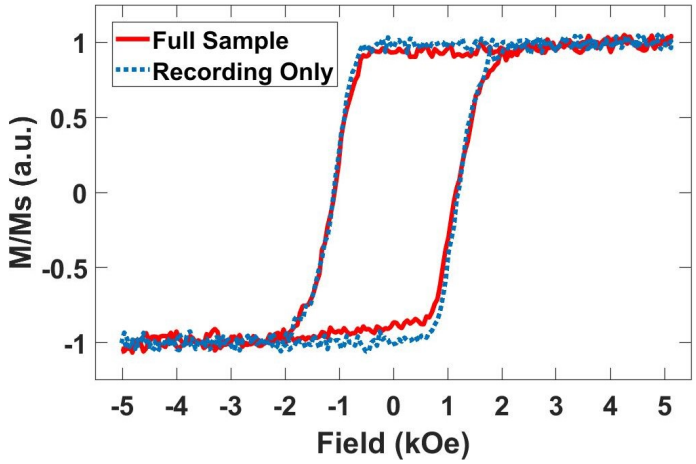

Figure 3.6 Comparison of Polar MOKE signal from a sample with both SUL and recording layer (red solid curve) to signal from a recording layer-only sample (blue dashed curve).

By replacing the SUL with Cu, we can create a composite response by adding the data from a VSM measurement of a recording layer-only sample (leftmost in Fig. 3.5) to that of a SUL-only sample (middle stack in Fig. 3.5). We will then compare the added signals to that of a "full" sample, which has both layers (rightmost in Fig. 3.5). The full sample is composed of Ag/NiFe/Ti/CoCrPt on an Si substrate. In the recording layer-only sample, Cu replaces the NiFe (Si/Ag/Cu/Ti/CoCrPt) to allow a VSM measurement of the recording layer properties while omitting the SUL magnetic contribution. Finally, the SUL-only sample is Si/Ag/NiFe/Ti. This process allows us to compare the hysteresis loop of the two layers behaving independently (SUL-only plus recording layer-only) and the layers potentially interacting during the VSM measurement (full samples).

The added signals result in the dashed curve in Fig. 3.8, which has been normalized using the saturation moment of the full sample. The full sample (which has both an SUL and a recording layer) measurement shows a clear pinching (red

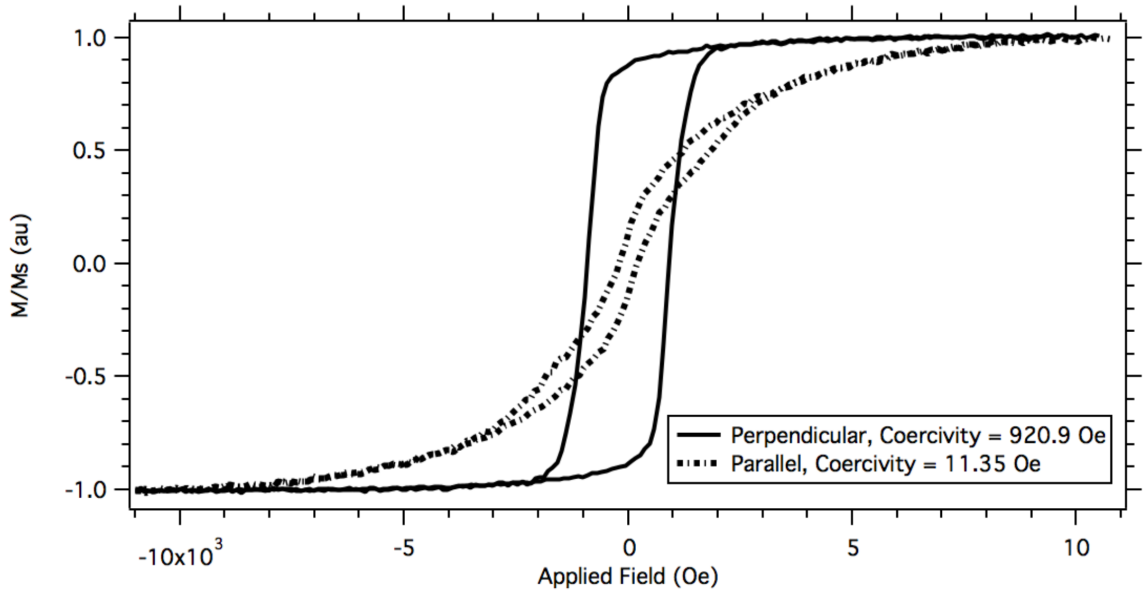

Figure 3.7 Normalized hysteresis loops for custom recording layer sample (Ag 50 nm/Cu 200 nm/Ti 5 nm/CoCrPt 15 nm). The perpendicular signal (solid line) shows a coercivity of $\sim$ 1 kOe, while the parallel signal (dotted line) shows a coercivity of $\sim$ 10 Oe.

curve in Fig. 3.8 inset), which likely indicates interaction between the layers, and is clearly absent in the summed, independent layer data (dashed).

Since there is very little change in the Polar MOKE data, which show similar signals for both full and recording layer-only samples, the interaction does not seem to affect the recording layer's magnetic response. It is possible that since the SUL material is much softer, it is being affected by the stray field of the recording layer. Further research could shed light into this discrepancy, and suggestions for next steps are provided in section 6.2.

In summation, we have used non-magnetic layer-substitution to compare the magnetic response of a sample including both a hard recording layer and an SUL *in situ* to that of the layers measured independently, the results of which show evidence of interaction. This result and technique can inform future choices of materials for

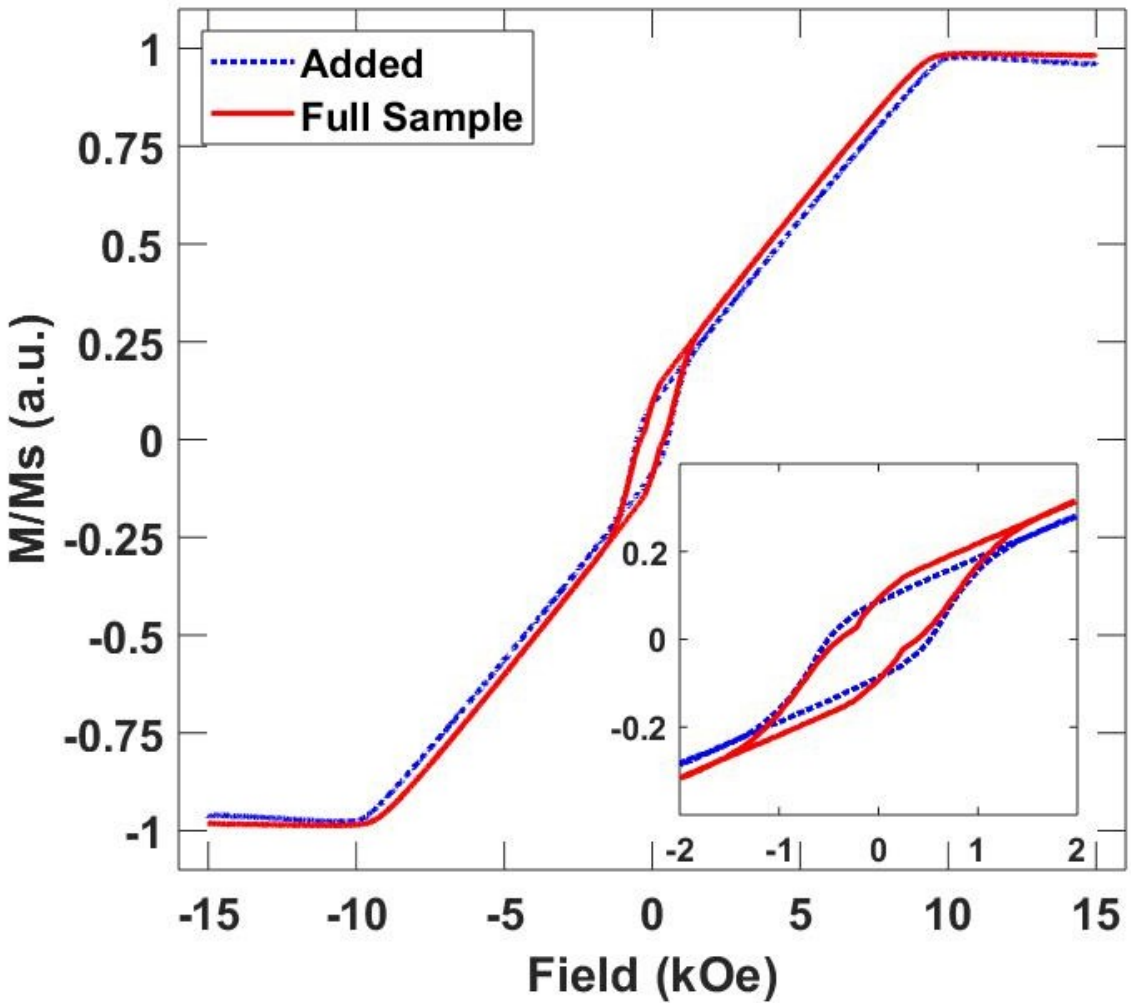

Figure 3.8 Normalized VSM signal from a full sample (solid red), compared to the summed signal of SUL-only and recording layer-only samples (blue dashed line). The inset shows pinching of the full sample data.

custom magnetic media designed to tune the forces during magnetic field-directed self-assembly of nanoparticles.

Chapter 4

ZFC/FC Data Collection

Now we will transition to discussing ZFC/FC measurements on nanoparticle samples. As mentioned in the introduction, ZFC/FC magnetization versus temperature measurements characterize the nanoparticles' magnetic response to an applied field. The results are useful in determining NP properties for use in magnetic heating [6], as biosensors [50], and field-directed self-assembly [15, 46, 47], among others. We will begin by discussing the samples themselves in section 4.1, then the nanoparticle size and separation details in section 4.2, the NP interactions in 4.3.2, and finally the measurement details in section 4.4.

4.1 Nanoparticle Sample Preparation

Initially, nanoparticle samples were prepared for measurement by soaking them from solution into a cotton swab, which was then cut and ahered to the sample holder with double-sided tape. This method resulted in low SNR and samples sometimes falling off during measurement, since the rapid vibration and large temperature variations caused the tape to lose adhesion. In addition, the cotton swab does not hold the particles rigidly, so there is vibration within the sample and possibly even particle rotation during measurement, all of which results in less accurate determination of blocking temperature. As mentioned in section 1.2, it is important that only the rotation of nanoparticle moment is measured, since our theoretical description does not include NP rotation. Currently, the samples are immobilized in a polymer of polystyrene-divinyl benzene and cured to a cylindrical shape.

This immobilization serves multiple purposes, the first of which is to remove the contribution of particle motion from the measurement. This ensures that the only signal produced during a ZFC/FC measurement is from the alignment of the individual nanoparticle moments with the external field and not from particles themselves rotating. The second purpose is to allow for easier mounting of the sample onto the Quantum Design brass sample holders provided with the VSM option (Fig. 4.1), since it reduces the previously mentioned pitfalls of the cotton swab.

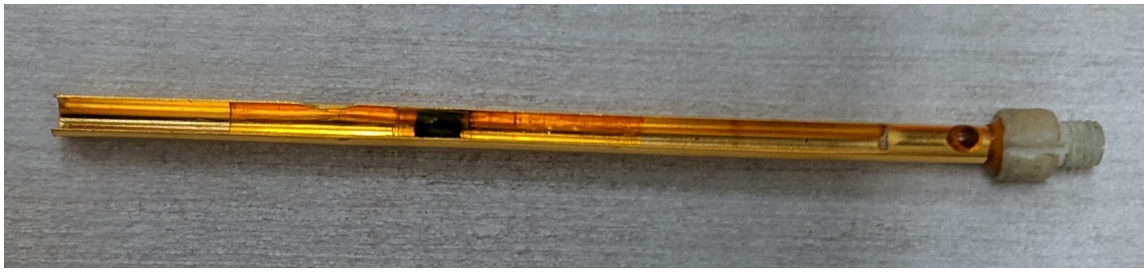

Figure 4.1 Brass sample holder from Quantum Design. The sample shown is a small cylinder of polymerized nanoparticles.

4.1.1 Materials

For the remainder of this work, we will be discussing the results of measurements on superparamagnetic iron oxide nanoparticles, marketed as having the properties given in Table 4.1, bought from Cytodiagnostics, Inc.

Table 4.1 Advertised properties of commercial iron oxide nanoparticles from Cytodiagnostics, Inc.

Property	**Units**	**Value**
Composition	N/A	Fe_3O_4
Mean Diameter	nm	10 ± 1.5
Saturation Magnetization	emu/g	> 45
Concentration	mg/mL	5

Nanoparticles are capped with a molecule during synthesis to stop their growth, to prevent aggregation when the particles are in solution, and possibly to functionalize them for their eventual use in biomedical, optical, electromagnetic, or other applications [23, 16]. These capping agents are called "ligands" and they bond to the nanoparticle surface atoms. In the case of the iron oxide nanoparticles studied here, they are capped with oleic acid as the surface ligand. The oleic acid has structural properties that bond strongly with the iron oxide NP surface atoms, so the particles remain well-dispersed in solution for longer periods [16]. The particles are provided in solution with toluene, since the oleic acid ligands allow the particles to be easily dispersed in a non-polar solvent like hexane or toluene.

The polymer chosen to immobilize the particles is polystyrene divinyl benzene, which is compatible with the organic, hydrophobic ligands on the nanoparticle surface and has high thermal resistance up to 400 K [25]. Though any compatible solid matrix can be used, the thermal resistance of polystyrene divinyl benzene is especially important for the measurements in section 4.3.2.

The polymer is produced by mixing a monomer/initiator solution of divinyl benzene (DVB), 2,2'-Azobis(2-methylpropionitrile) (AIBN), and styrene and then heating based on the guide given in the Materials section of [25]. All the chemicals used for the polymerization were bought from VWR.

4.1.2 Process

By using commercial particles, we have reasonable assurance that the particles are of consistent quality. Our first step in comparing the differences between data analysis techniques is to rule out issues arising from the production of the nanoparticles themselves, and an established NP supplier helps provide the assurance that the particles are of consistent quality.

To begin the polymerization of the NP samples, the particles are first dried from solution, then added to the monomer/initiator solution. The mixture is then cured at 70°C for at least 4 hours until the polymer solidifies [25].

The immobilized nanoparticle samples were prepared in five concentrations with various calculated weight percents relative to the total sample weight (see Table 4.2). The concentration was found from an ICP-OES measurement, performed at Clemson University, of the mass of iron in the sample, from which the mass of Fe_3O_4 NPs can be estimated. However, most likely due to evaporation losses during curing, the actual concentrations were found to be larger, sometimes more than twice that of the calculated values. We will examine how the concetration of nanoparticles changes their response during the ZFC/FC measurement and examine the relationship between concentration and mode blocking temperature in chapter 5.

Table 4.2 Calculated and measured concentrations of polymerized commerical iron oxide nanoparticle samples for ZFC/FC measurement.

Sample Name	**Calculated wt%** ($\mathbf{g}_{Fe_3O_4}/\mathbf{g}_{total}$)	**Measured wt%** ($\mathbf{g}_{Fe_3O_4}/\mathbf{g}_{total}$)
2x	0.62	1.69
1x	0.37	0.81
0.5x	0.18	0.32
0.25x	0.08	0.19
0.125x	0.05	0.1

Before moving into a discussion of other properties of these NP samples, it will be useful to note the naming convention used in this work. Because the sample with calculated wt% 0.37 $g_{Fe_3O_4}/g_{total}$ (the second highest concentration) was produced with the standard mass of 10mg of NPs to 1.5 mL of monomer/initiator solution, we will refer to this concentration as "1x." Moving forward, all other samples are referred to by its initial mass of NPs in the mixture, relative to the 1x sample. These initial masses decrease by a factor of 2 each time. Thus, the highest concentration sample

is 2x, then 1x, then 0.5x, then 0.25x, and finally 0.125x. This naming convention is given in the first column in Table 4.2.

4.2 Size and Separation Distance in Nanoparticle Samples

The sizes of the commercial nanoparticles used for this study was determined from TEM images using analysis by ImageJ. The resulting diameter data was then fit with two distributions for comparison: lognormal and Weibull. These two distributions are common for fitting NP size data and are compared to find the superior fit. All TEM imaging and sample preparation for imaging were performed at Clemson University by Zichun (Tony) Yan (see Figs. 4.2 and 4.4). Cumulative frequency data was used for the fit, since binning data for comparison with differential probability distributions can obscure information in the data [38].

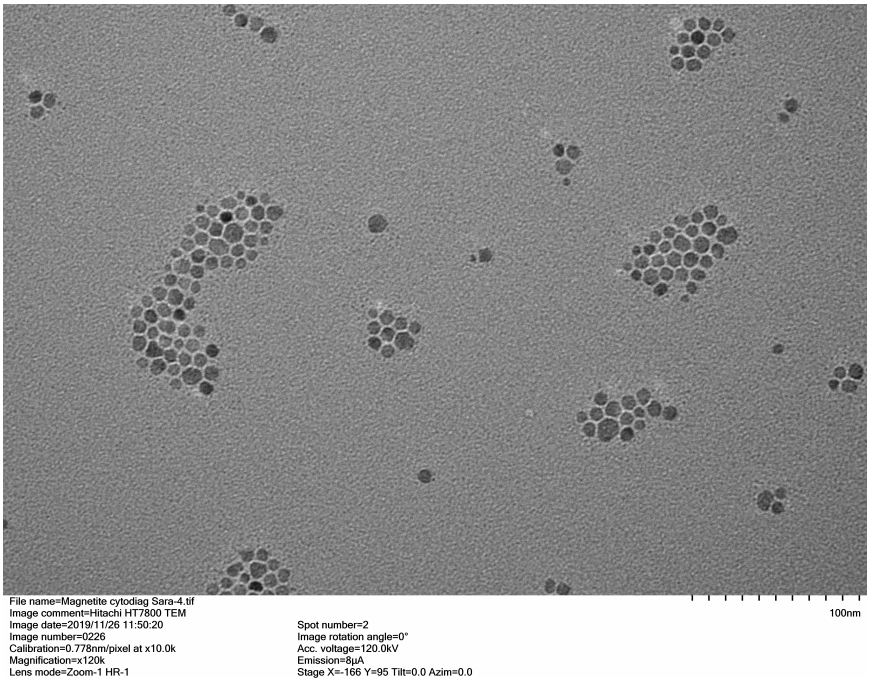

Figure 4.2 TEM image of the commercial iron oxide NPs.

The image processing and analysis method roughly followed that outlined in [38]. The best fitting distribution will be used in later analysis (see sections 5.3, 5.4, and 5.5). The lognormal and Weibull distributions are widely used to characterize nanoparticle size data [38], and are commonly available in off-the-shelf analysis software. The fits of the cumulative number-based frequency data from the diameters output by ImageJ were performed using the Distribution Fitter tool in MATLAB.

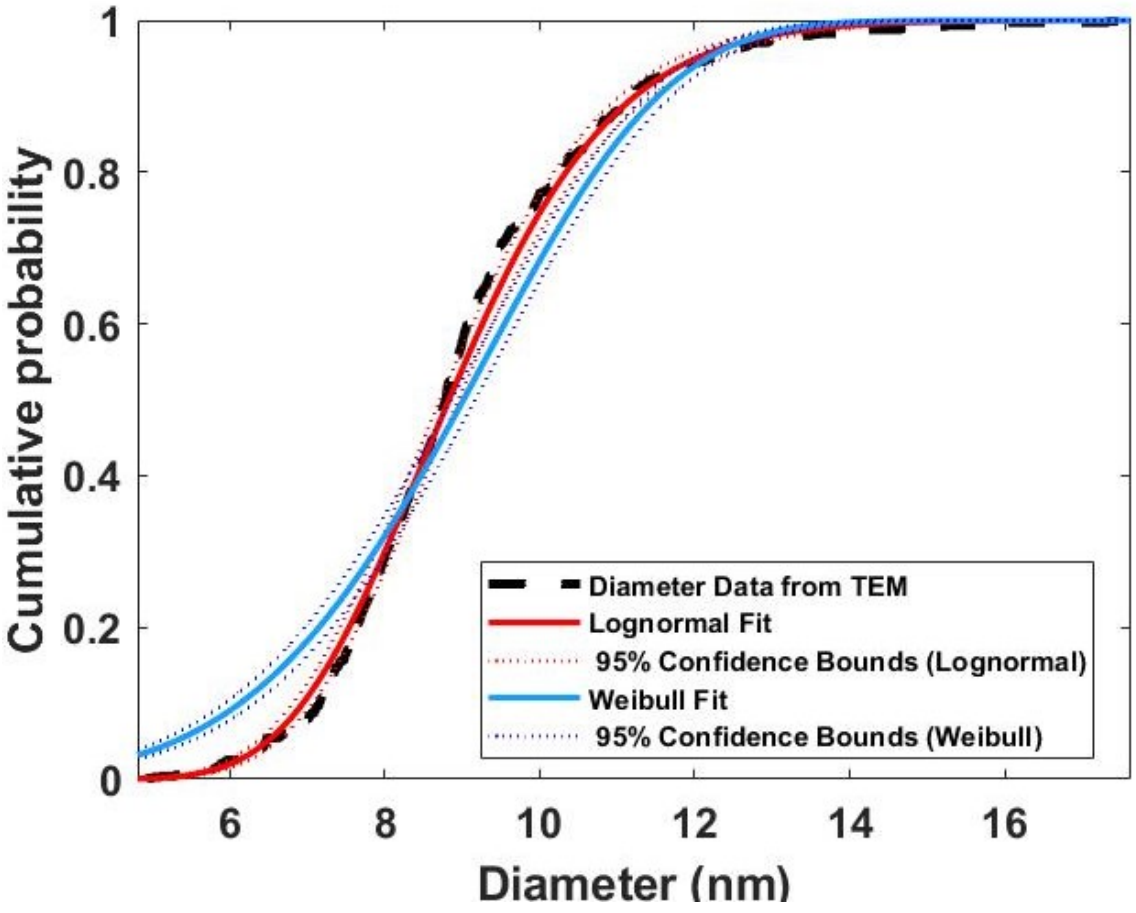

Figure 4.3 Comparison of lognormal and Weibull CDF fits of diameter data.

Despite a better fit of the data from the lognormal distribution (both visually and a higher value of the log likelihood provided by the Distribution Fitter, see Fig. 4.3), the resulting parameters from both the distribution fits were similar: a mean

diameter of 8.5 ± 1.7 nm for the lognormal fit and a mean diameter of 8.9 ± 2.1 nm for the Weibull fit.

Since the nanoparticle size distribution is best described by the lognormal distribution, here we give a description of the lognormal distribution's probability density function (PDF) and its characteristic points: the mean, median, and mode, which will be useful when we analyze the ZFC/FC data in chapter 5. The lognormal PDF of a random variable X is:

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma x} \exp\left(-\frac{(\ln x - \mu)^2}{2\sigma^2}\right) \tag{4.1}$$

[7], where σ and μ are the two parameters of the distribution. If our data is X and $Y = \ln X$, then μ and σ are the mean and standard deviation of Y [7]. However, the mean, median, and mode of X are as follows:

$$\begin{aligned} \langle x \rangle &= \exp\left(\mu + \frac{1}{2}\sigma^2\right) \\ med_x &= \exp(\mu) \\ x_m &= \exp(\mu - \sigma^2) = med_x \exp(-\sigma^2) = \langle x \rangle \exp\left(\frac{-3\sigma^2}{2}\right) \end{aligned} \tag{4.2}$$

[7]. Note the relationships between the mode (x_m) and the other descriptive points. The values of the mean ($\langle x \rangle$) and standard deviation (σ_x) of X can be used to find μ and σ (called the "dimensionless spread" of the distribution).

In addition to the measurements to determine particle size, small sections of polymerized nanoparticle samples were also used for TEM imaging to determine the relationship, if any, between nanoparticle concentration and interparticle separation (see Fig. 4.4). The image analysis was again performed using ImageJ. Prior to imaging with TEM, the nanoparticle in polymer samples were smoothed with a microtome. All imaging and sample preparation for imaging was performed at Clemson University.

One of the results of the ImageJ particle analysis is the x,y coordinates of the centroid of each detected particle. The coordinate data was loaded into MATLAB

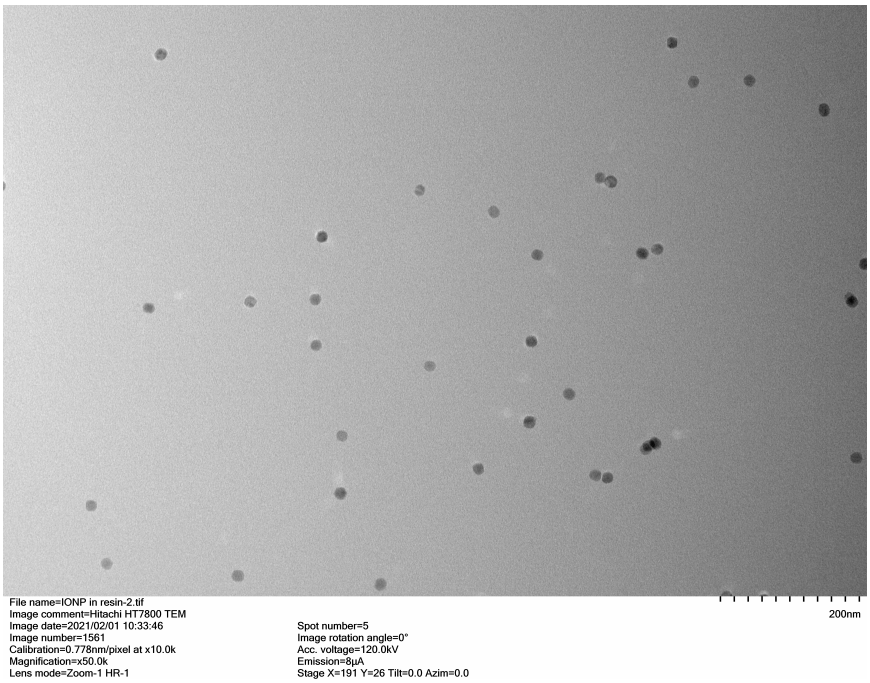

Figure 4.4 TEM image of the commerical iron oxide NPs in polymer.

to compute the average distance between all the particles in each sample, based on multiple images per sample. The results of this analysis are given in Fig. 4.5 with the expected value based on the concentration indicated with a black, dashed line.

Despite the changes in concentration, the average nanoparticle center-to-center separation for each sample was found to be comparable (between 40 and 50 nm) and had no relationship with concentration, with the exception of one sample which had an average separation of 85 nm (see Fig. 4.5). The outlier had fewer particles in each image, which resulted in a much larger standard deviation.

In addition, we can ask how many of the nanoparticles are touching in each sample, since that will likely lead to interacting nanoparticles. The relative number of touching particles in each sample is given in Fig. 4.6. This is the relative number of NPs whose separation from their neighbor is less than or equal to the average NP

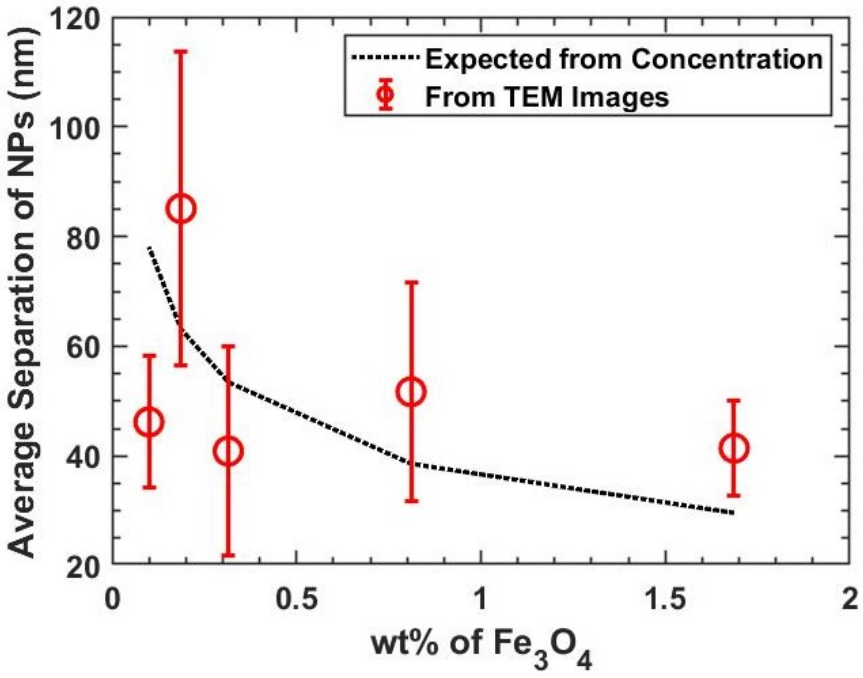

Figure 4.5 Plot of average NP separation in polymer versus wt% of Fe_3O_4 (red, solid) with standard deviation of the average given as error bars. The expected value from the concentration is the black, dashed line.

diameter. This counts all instances of touching particles, so if a particle touches two neighboring particles (a "trimer"), it is counted twice. It is "relative," since we divide the total count of touching particles by the number of particles in an image, so we can directly compare across samples without automatically having a higher count for samples whose images simply include more particles.

It is clear in Fig. 4.6 that the sample with the lowest relative number of touching particles is the 0.25x sample, which is also the sample with much higher average separation in Fig. 4.5. On the other hand, the 0.5x sample has a much larger relative number of touching particles, but an average separation (Fig. 4.5) that is comparable to the 2x, 1x, and 0.125x samples' separations. This suggests that in the 0.5x sample, the particles are very clumped together, but the clumps are separated

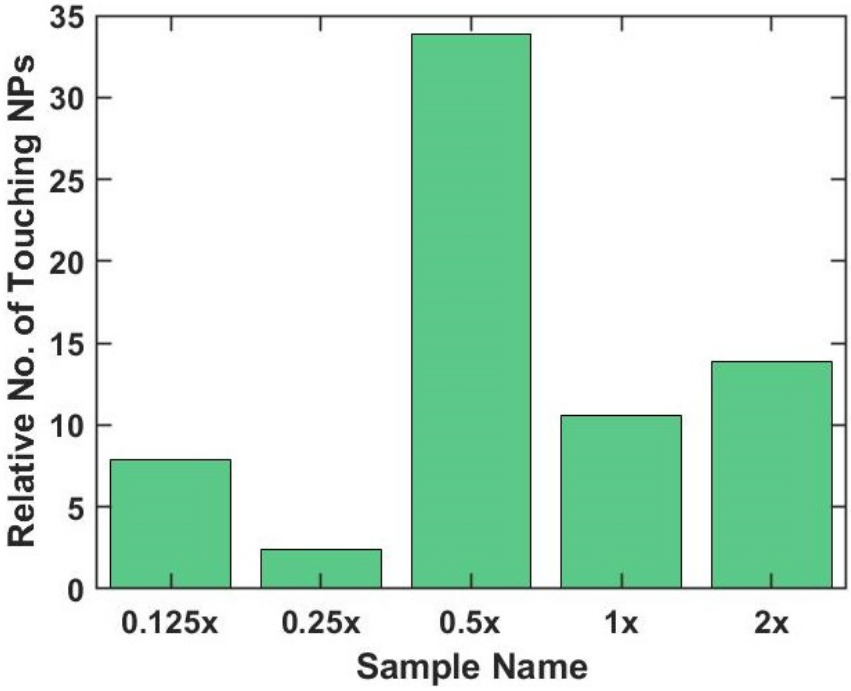

Figure 4.6 Relative number of touching particles for each sample.

enough that the average separation is still roughly equal to that of other samples with fewer touching particles. The 2x, 1x, and 0.125x samples have more similar relative numbers of touching particles and average separations. We would expect the average separation to decrease with increasing concentration and the ideal relative number of touching particles should be 0, so these values are unexpected.

The TEM images were taken on planed surfaces parallel to the diameter, so it is possible that the variation in nanoparticle separation with concentration occurs in the plane parallel to the length of the cylinder. Unfortunately, there was not enough remaining sample to determine this. Another possible explanation is that the nanoparticles settle while the polymer cures, since it takes several hours. It could be that the regions of polymerized sample that were imaged had similar NP separation, but that other regions of the sample would be vastly different, depending on their

relative height during curing. Since the remainder of our samples was too small to test this hypothesis, it must be left to future work to determine the exact nature of nanoparticle separation in polymerized samples.

Now that we have discussed sample preparation and characterizing the size distribution and separation of the NPs, we will use this information to discuss the magnetic interactions in the polymerized samples in the next section.

4.3 Nanoparticle Interactions

Since we have presented information on the separation of the nanoparticles, a logical question to ask is: to what extent do the nanoparticles interact in the samples? The nanoparticle samples we measure in this work are composed of many particles in a confined volume, so it is possible that interactions will affect measurements with small fields. Our analysis of the ZFC/FC data in chapter 5, taken with small fields, assumes non-interacting particles, so it's useful to attempt to quanitfy these interactions.

4.3.1 Dipole Interactions in Polymerized Samples

To begin, we use the simplest model of a magnetic interaction between nanoparticles: one magnetized particle produces a magnetic field that is felt by a neighboring particle. Since the situation becomes much more complex when we account for the neighbor's influence back on the first particle, we will stop with the dipole field picture for simplicity.

The equation for the magnitude of the dipole field produced by a single-domain, magnetic nanoparticle is:

$$H_{dipole} = \frac{m}{r^3}\sqrt{3\cos^2\theta + 1} \tag{4.3}$$

[8], where θ is the angle between the nanoparticle moment vector, $\mathbf{m}$, and the distance vector, $\mathbf{r}$, from the nanoparticle to the point of interest. We want to assess the relative strength of the dipole field to the applied field, since we can reasonably assume that when H_{dipole} is more than 10% of our applied field at the location of a neighboring particle, the interactions will be strong enough to alter the measurement. The magnitude of H_{dipole} is largest when $\theta = 0$ or π, so let us take this worst-case scenario.

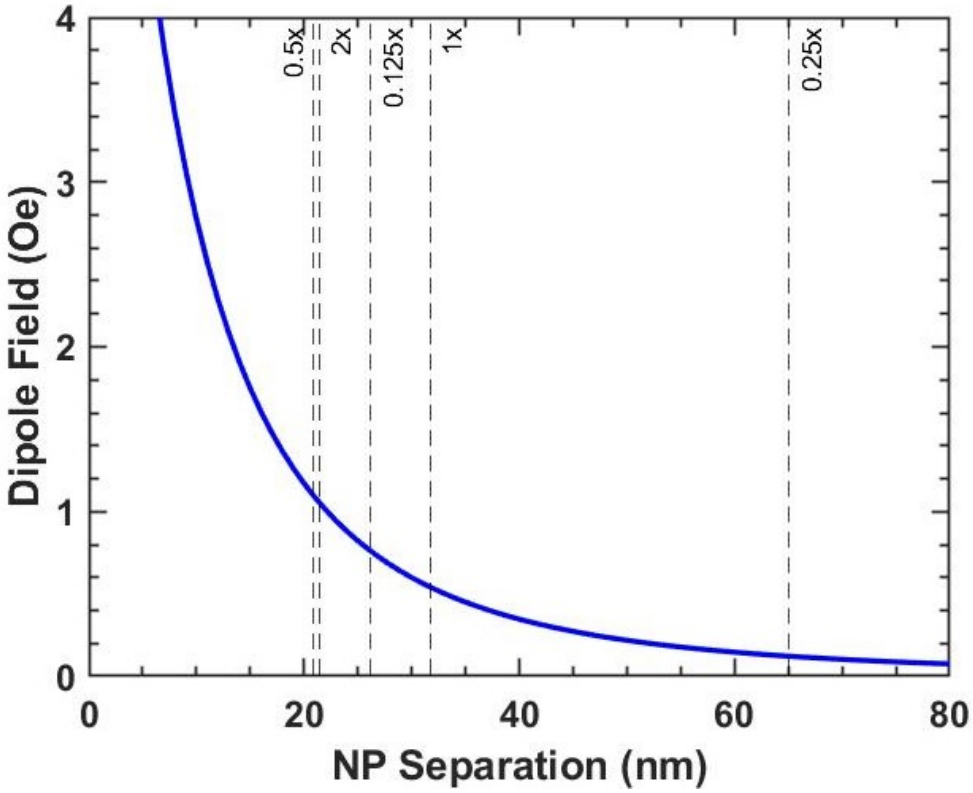

Figure 4.7 Dipole field magnitude from a single NP versus distance. Vertical lines indicate the measured average NP separation in the samples.

This worst-case dipole field magnitude is plotted in Fig. 4.7, which also includes vertical lines at the average separation values for each sample, as found from TEM images of the polymerized samples in the previous section. The sample with

the largest average NP separation from section 4.2 will have the smallest dipole field felt by neighboring particles, as indicated by the vertical dashed line on the far right of the plot.

Table 4.3 Applied fields for ZFC/FC measurements and minimum NP separation to discount dipole interactions.

Applied Field (Oe)	**Minimum NP Separation (nm)**
10	60
25	41
50	32
100	25
200	20

Given the advertised value of 45 emu/g of the saturation mass magnetization of the NPs (Table 4.1) and the average NP size found previously, the moment for a single NP is roughly 7.5×10^{-17} emu. The magnitude of $\mathbf{m}$ is constant, since the particles are single-domain, though a rotation of $\mathbf{m}$ away from the z-axis (the direction of H_{appl}) due to the dipole field means that our measured signal will be lower than expected. In the case when the dipole field is aligned with the applied field ($\theta = \pi/2$), even though the magnitude of H_{dipole} is less, it will still cause the measurement to be inaccurate, since neighboring particles will feel an increased total field. Using our worst-case scenario with Eq. 4.3 and our condition of 10% of the applied field, we can find the minimum separation of the particles at which we may say our interactions are "small." We do this for each applied field used for the ZFC/FC measurements discussed in more detail later. These results are given in Table 4.3.

If we compare the necessary separations to Figs. 4.5 or 4.7, it's immediately obvious that the only sample with high enough measured average NP separation to have "small" interactions for all field values ($>$ 60 nm, based on our previous criteria) is the 0.25x sample. This sample's expected average separation based on

concentration (63 nm) is also greater than the minimum distance required for "small" interactions for all fields. The other samples' average separation values from the TEM images are such that we can assume that measurements on these samples with 10 and 25 Oe will be impacted by the NP interactions.

4.3.2 Interaction Temperature Measurement

In addition to the calculation of dipole field given in the preceding pages, we can attempt to measure the strength of the NP interactions experimentally. One method that is commonly used to approximate NP interaction is the "interaction temperature measurement." This is a rough characterization of the strength of the interactions, expressed as a temperature, that can be included in the calculation of the anisotropy constant. The anisotropy constant that includes interaction temperature is called the effective anisotropy constant, K_{eff}.

The interaction temperature, T_0, also called the "paramagnetic Curie point," is found using the Curie-Weiss law:

$$\frac{1}{\chi_0} = \frac{T - T_0}{C} \tag{4.4}$$

[8], where χ_0 is the initial magnetic susceptibility, C is the Curie constant, and T is temperature. Based on Equation 4.4, the interaction temperature is the T-axis intercept of the inverse χ_0 versus temperature plot (Fig. 4.8) [25]. This additional parameter, when used in conjunction with T_B, lets us calculate the effective anisotropy constant, as in [25], as follows:

$$K_{eff} = \frac{k_B(T_B - T_0)}{V_m} ln\left(\frac{\tau_m}{\tau_0}\right). \tag{4.5}$$

The measurement process to determine the interaction temperature involves many low-field moment versus field measurements at various temperatures, which result in mostly linear plots like the unsaturated region of Fig. 2.2. The slope of each magnetization versus field measurement is the magnetic susceptibility, χ_0, at that

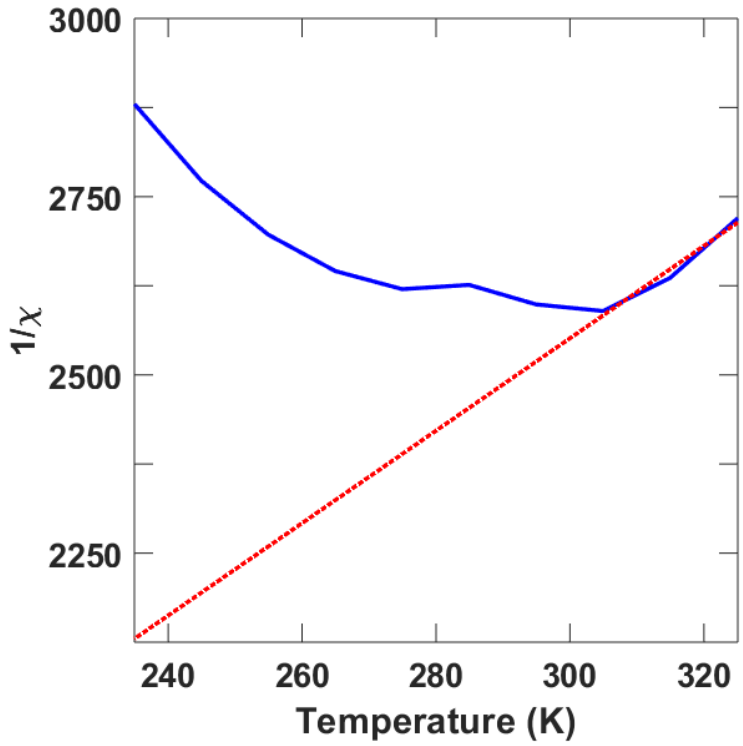

Figure 4.8 Plot of inverse susceptibility versus temperature for a sample of CoNi nanoparticles (blue). Includes a fit of the high-temperature, paramagnetic region with the Curie-Weiss law (red, dashed).

temperature. The inverse of the susceptibility is then plotted versus temperature (blue, solid curve in Fig. 4.8). For superparamagnetic samples, there will be a high-temperature region (much higher than the blocking temperature) of the inverse susceptibility plot that is linear and monotonically increasing with T, indicative of paramagnetic response to the field [8]. This portion of the data is fit with a line (red, dashed line in Fig. 4.8), and the temperature-axis intercept of this fit line is the interaction temperature, T_0 [25].

For our immobilized iron oxide NPs, we measured the samples between -50 and +50 Oe at temperatures ranging from 160 K to 395 K. We included a higher density of points between 330 K and 395 K, since this method of finding the interaction

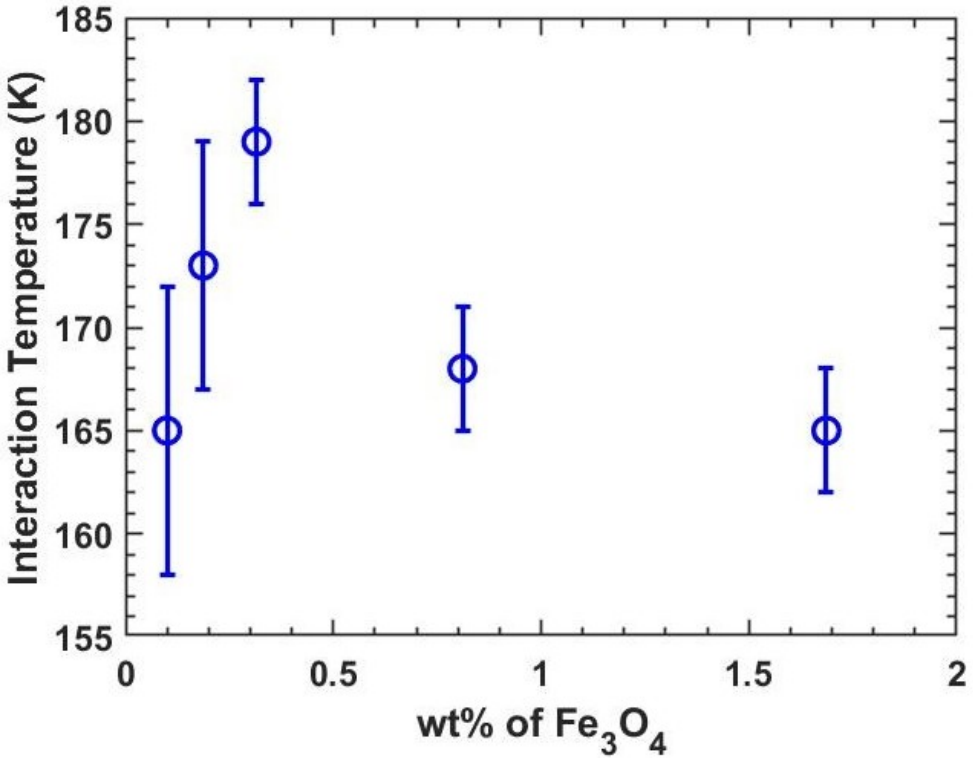

Figure 4.9 Plot of interaction temperature versus the wt% of Fe_3O_4 in the sample.

temperature is very sensitive to noise, and this temperature region is the most clearly linear for all of our samples.

It is interesting to note that there was no clear relationship between interaction temperature and concentration for our samples (see Fig. 4.9), despite the fact that we would expect the interaction strength to increase with increasing concentration. However, the largest value of T_0 does correspond to the sample that had a much higher relative number of touching NPs (Fig. 4.6), which is reasonable for a sample with many clusters of tightly grouped particles. The other values of T_0 do not fit with expectations, based on the separation data, however.

Other studies that measured similar particles have reported $T_0 = 42$ K for similar systems [25], which is much lower than all our results in Fig. 4.9. This

may be simply because the interaction temperature is a rough characterization of interaction strength, but it is not exact. Additionally, the last step of fitting the $1/\chi_0$ plot is highly variable, since it is far from the temperature axis, and thus is sensitive to noise and to the points selected for the fit. The inverse susceptibility is not entirely linear at high temperatures for real samples, so T_0 changes drastically depending on the points included in the fit and the SNR of the measurement. It may also be that even our lowest concentration samples were still populated with enough nanoparticles to have large interactions, which may indicate that still lower concentrations should be examined in the future. As we will see in the later sections of chapter 5, this interaction has a noticable effect on the calculation of T_B, despite the fact that the often quoted result of interaction temperature was unable to capture the changes in interaction strength with changes in concentration.

Now that we have examined the size distribution, NP separation, and the interactions in our polymerized NP samples, we move in the next section to discussing the main magnetic measurement: the Zero Field-Cooled/Field-Cooled measurement.

4.4 The ZFC/FC Measurement

As mentioned in previous chapters, NP response to an applied field is important for many applications and for probing the deeper physics of nanoscale magnetism. A popular characterization technique that yields information about the anisotropy, size distribution, and magnetic interactions between particles in a NP sample is the zero-field cooled/field-cooled (ZFC/FC) moment versus temperature measurement. The names "zero-field cooled" and "field-cooled" refer to the preparation of the sample, the measurements themselves are performed *after* the cooling stage while the sample is warmed, but data from these initial cooling stages is also presented for comprehensiveness.

In ZFC/FC, the sample is first cooled to a low temperature with no applied field, warmed with a small applied field, and then finally cooled and warmed again in the same small applied field. In this work, particles are cooled from room temperature to 6 K and the small fields applied are 10, 25, 50, 100, and 200 Oe. The measurements were performed with the same Quantum Design VSM option of the PPMS described previously. As with any measurement, there is inherent uncertainty in the experimental quantities which must be accounted for before quoting results. When taking magnetic measurements, it is important to correctly determine the field applied to the sample, since this will determine the sample's magnetic response.

The field in the superconducting coils is measured by the PPMS software from the current applied to the Nd wire. This can be inaccurate due to trapped flux in the Nd wires after taking the magnet to high fields (≈ 1 T), due to defects in the material that results in pinned domains [36]. This uncertainty can be roughly 20 Oe when the field is near 0. To decrease the trapped flux before strictly low-field measurements, we minimize the trapped flux by taking the field up to 2 T and oscillating back to 0, which reduces the uncertainty in the field to roughly 3 Oe [36].

For superparamagnetic particles, during the initial cooling, the net magnetic moment is nearly zero, since all particle moments are randomly oriented. This is the "initial cooling" curve in Fig. 4.10. As the temperature decreases, this net moment is " frozen in." Once the small magnetic field is applied (curve labeled "ZFC" in Fig. 4.10), the net moment remains smaller until the temperature reaches a point where the thermal energy of the particles is sufficient to overcome the energy barrier and their moments flip to align with the field and cause an increase in the signal, as discussed in chapter 1. The temperature at which a nanoparticle's moment flips is its blocking temperature. For a collection of nanoparticles, the temperature at which the most particle moments flip at once (creating a large increase in the measured moment signal) is called the mode blocking temperature, T_B. Once all nanoparticle moments

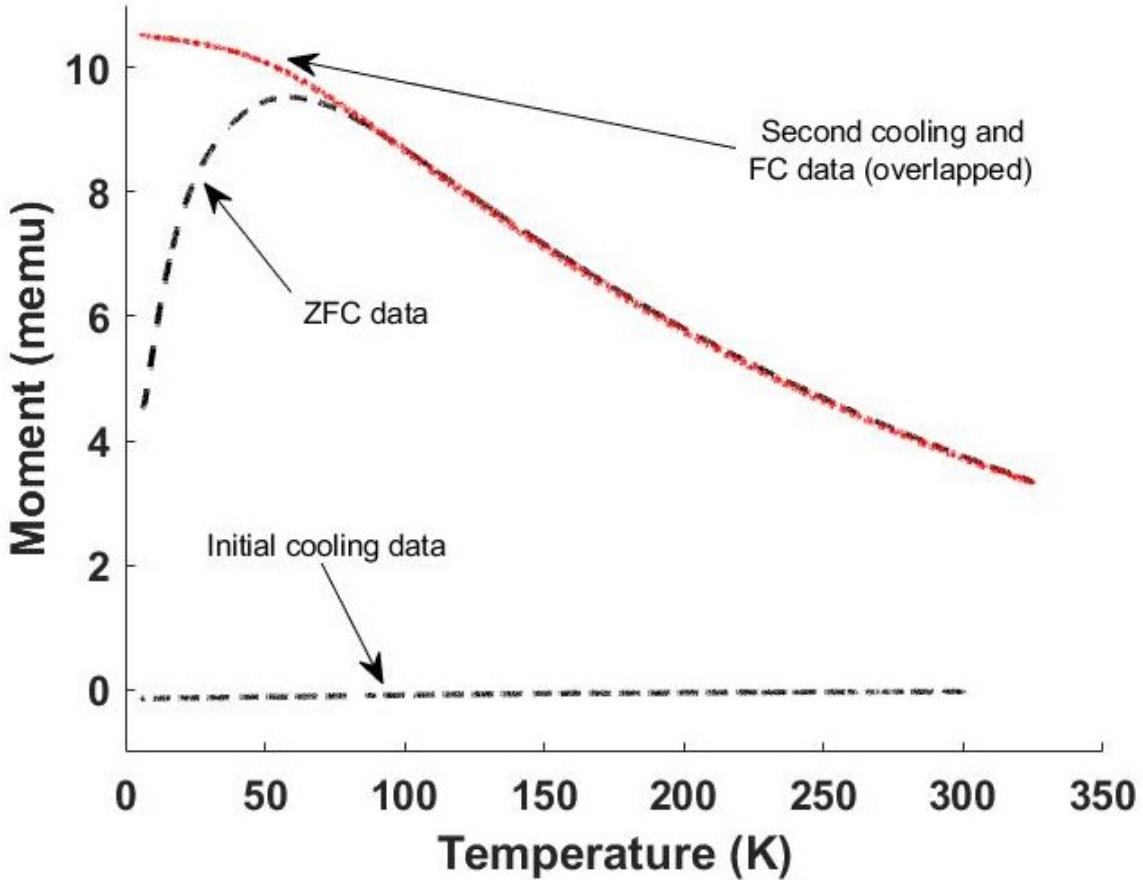

Figure 4.10 ZFC/FC data for an iron oxide NP sample measured with 200 Oe applied field.

have flipped to align with the field, the signal is peaked. After this peak, the signal will gradually decrease as the thermal energy allows the nanoparticle moments to become randomly oriented again. After reaching the maximum temperature (325 K), the sample is then cooled again (labeled "second cooling" in Fig. 4.10) while the field is still applied and the FC curve results from measuring another warming sequence in the field after this.

We can use the Néel-Arrhenius Law from section 1.2 to find the blocking temperature when the attempt time is equal to the measurement time, τ_m. The

mode blocking temperature is related to the mode volume through the Néel relaxation equation as follows:

$$T_B = \frac{KV_m}{k_B} ln\left(\frac{\tau_m}{\tau_0}\right) \tag{4.6}$$

[24], where τ_m is computed from the warming rate during data collection, τ_0 is the characteristic time for the material (typically 10^{-9} s), K is the anisotropy, V_m is the mode volume, and k_B is Boltzmann's constant. This equation will be expanded upon when we discuss various methods to analyze the ZFC/FC data in chapter 5, but for now Eq. 4.6 will suffice to give the reader a picture of the physics of the measurement.

From Eq. 4.6, we can find the anisotropy constant for a sample, given we can find the mode blocking temperature, or vice versa. Several methods to extract these parameters from the data are given in chapter 5 as well as results.

Now that we have described our samples and the measurement process, in the next chapter we will discuss the process of analyzing our data with several methods and give the results of that analysis. Finally in chapter 6, we will give a comparison of the methods and suggestions for future work.

Chapter 5

Data Analysis

In the following sections, we will outline four analysis methods to determine the blocking temperature of a given NP sample based on ZFC/FC data from measurements on the five NP samples discussed in section 4.1. The first two analysis methods rely on inflection points of the data (sections 5.2 and 5.3) and the final two involve fitting the ZFC portion of the data (sections 5.4 and 5.5). The measurements were performed with fields of 10, 25, 50, 100, and 200 Oe. We will present results of the analysis, but will hold off the comparison of results between methods until chapter 6. Given that the anisotropy constant of bulk Fe_3O_4 is 13.5 kJ/m^3 [24] and our mode diameter is 8.5 nm (calculated from the mean and Eq. 4.2), we expect to see a blocking temperature of 12 K across all samples and field values (from Eq. 1.3), if blocking temperature is independent of field and concentration. We will call this the expected or "bulk" blocking temperature. As we present results, keep in mind these values.

5.1 Traditional Determination of Blocking Temperature

There is a persistent misunderstanding about the location of the blocking temperature found via the ZFC/FC measurement. The easiest and unfortunately the least accurate way that some researchers determine blocking temperature is to simply locate the peak of the ZFC curve. This however, leads to large, sometimes extremely large, error in the resulting anisotropy constant [3, 24].

One explanation for why this method is so inaccurate is that it draws an equivalency between the shape of the ZFC curve for a real collection of nanoparticles and the theoretical ZFC curve for an ideal, monodisperse sample whose nanoparticle moments flip instantaneously to align with the external applied field. In the ideal case, the particles are uniformly blocked (or their moments are frozen in random orientation) below T_B, resulting in a nearly net zero magnetization, then once the ideal particles reach T_B, their moments simultanously flip, resulting in a maximum measured signal which then drops off as the particles continue to warm and the thermal energy allows the nanoparticle moments to move and orient randomly again (Fig. 5.1). This is described mathematically for a collection of monodisperse particles with randomly aligned easy axes as follows:

$$\frac{M_{ZFC}}{M_s} = \begin{cases} \frac{\mu_0 M_s H v}{3K}, & T < T_B \\ \mathcal{L}\left(\frac{\mu_0 M_s H}{k_B T}\right), & T > T_B \end{cases} \tag{5.1}$$

[24], where M_s is the saturation magnetization, H is the applied field, v is the particle volume, T is temperature, K is the anisotropy constant, and $\mathcal{L}$ is the Langevin function: $\mathcal{L}(x) = \coth(x) - 1/x$. See Fig. 5.1.

In the ideal case, the peak occurs at the blocking temperature where the function is also discontinuous. However, in real samples, measured ZFC curves are continuous and rounded off compared to those generated by Eq. 5.1. This rounding off is due to several factors, including the polydispersity of the sample and the stochastic nature of moment flips, both of which will be discussed in following sections. The result of this is that the real blocking temperature does not correspond to the peak, but to a lower temperature to the left of the peak [3, 24].

This method of taking the peak of M_{ZFC} has been widely used, but yields a much higher blocking temperature. The discrepancy between the anisotropy constant found for NP samples using this erroneous T_B and that of bulk material has in some cases been explained by surface effects in the NPs, whose surface does comprise a

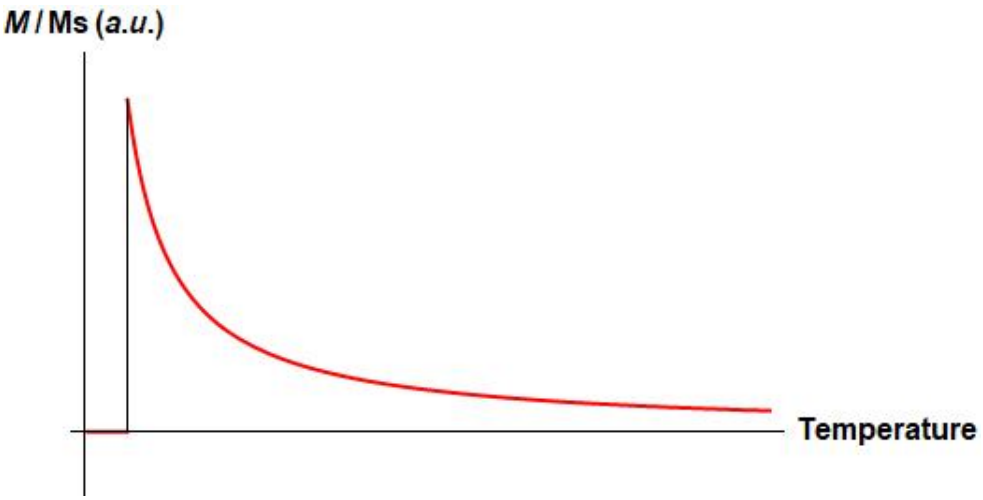

Figure 5.1 Plot of the piecewise, ideal definition of M_{ZFC}. The magnetization starts at nearly 0, then increases instantaneously at T_B, then decays back to zero with increasing temperature.

much larger portion than in bulk samples. However, the anisotropy constant found using other, more accurate methods agrees much better with bulk values [24]. In the following sections of this chapter, we will present some of those more accurate methods and discuss the resulting blocking temperatures.

5.2 The "Ugly" Method

The so-called "ugly" method for determining blocking temperature involves taking the inflection point (IP) of the ZFC branch of the measured signal. The inflection point indicates the point at which the most individual nanoparticle moments are aligning with the external field simultaneously, leading to the largest increase in moment. As shown in section 4.2, the nanoparticles measured here have a dispersion of sizes, and later methods will deal with this more directly, but for the moment we will simply suppose that there is also some distribution of blocking temperatures. Thus, the inflection point represents the most common value of the blocking temperature,

or roughly the mode of that distribution. We will use the mode blocking temperature as the characteristic blocking temperature for a given sample moving forward.

In the case of the ZFC branch, the inflection point is also the maximum of dM/dT, since there is a change in the concavity of the data. The derivative of our data was calculated by taking local linear fits in the neighborhood of each point and using the slope as the derivative at that point. Since the data is noisy, the local derivative data will also be noisy. By increasing the width of the points included in the linear fit, it is possible to reduce this noise, but in doing so, we increase the uncertainty in temperature, which must be included in the error for the resulting inflection point value. Once we find the derivative, we then fit its peak with a parabola, and the inflection point is the location of the vertex of the parabola. This again introduces error, which can be calculated from the variances of the fit coefficients. The location of the vertex of the parabola with equation $y = ax^2 + bx + c$ is:

$$x_0 = -\frac{b}{2a}, \tag{5.2}$$

and we can use this equation and error propogation to account for the error of the fit coefficients in our calculation of the inflection point (see Fig. 5.2).

Now we must ask, how good of an approximation of the blocking temperature is the inflection point? From [3], this is still only an estimation of T_B, which is scaled by the dimensionless spread parameter of the lognormal distribution discussed in section 4.2, σ [3]. However, we can assess how good an approximation it will be, from the relationship between the inflection point and the "true" average blocking temperature. The ratio of the inflection point (IP) to that of the "true" average blocking temperature follows a universal curve found in [3]:

$$\frac{IP}{\langle T_B \rangle} = 1 - 0.21\sigma - 0.79\sigma^2, \tag{5.3}$$

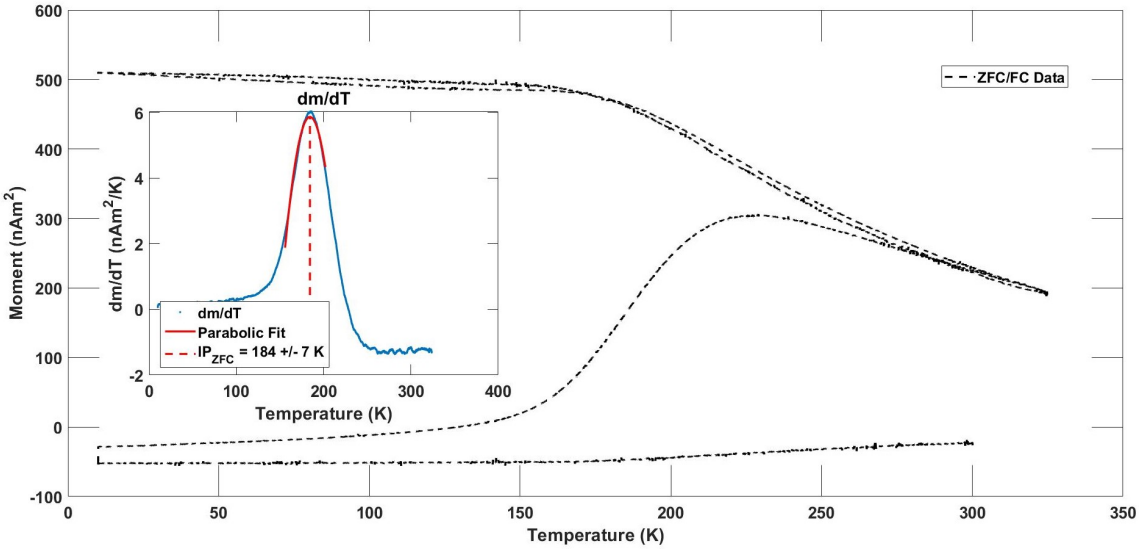

Figure 5.2 Plot of ZFC/FC data from a sample of CoFeMn NPs made at Clemson University. Inset is the derivative of the moment with respect to temperature, fitted with a parabola near the inflection point.

in which the error for the numerical values is 0.02. From this and Eq 4.2, we can find a relationship between the inflection point and the "true" mode blocking temperature:

$$T_{B,mode} = \langle T_B \rangle \exp\left(\frac{-3\sigma^2}{2}\right) = \frac{IP \exp\left(\frac{-3\sigma^2}{2}\right)}{1 - 0.21\sigma - 0.79\sigma^2} \tag{5.4}$$

Using our values found in section 4.2, we find that the mode blocking temperature is 0.98 times the inflection point temperature. Since the uncertainty in temperature that arises when taking the derivative is larger than 2% for all of our measurements, we will simply state the inflection point value as the estimate of the mode blocking temperature for this analysis method.

Now let us discuss the resulting inflection points found with the ugly method. There is no clear relationship between the IP and concentration (as shown in Fig. 5.3), except that it is generally lower for higher concentration. While we cannot determine the exact nature of the relationship from this plot between IP and concentration, we can say that the IP does not remain constant across all concentrations. It's also

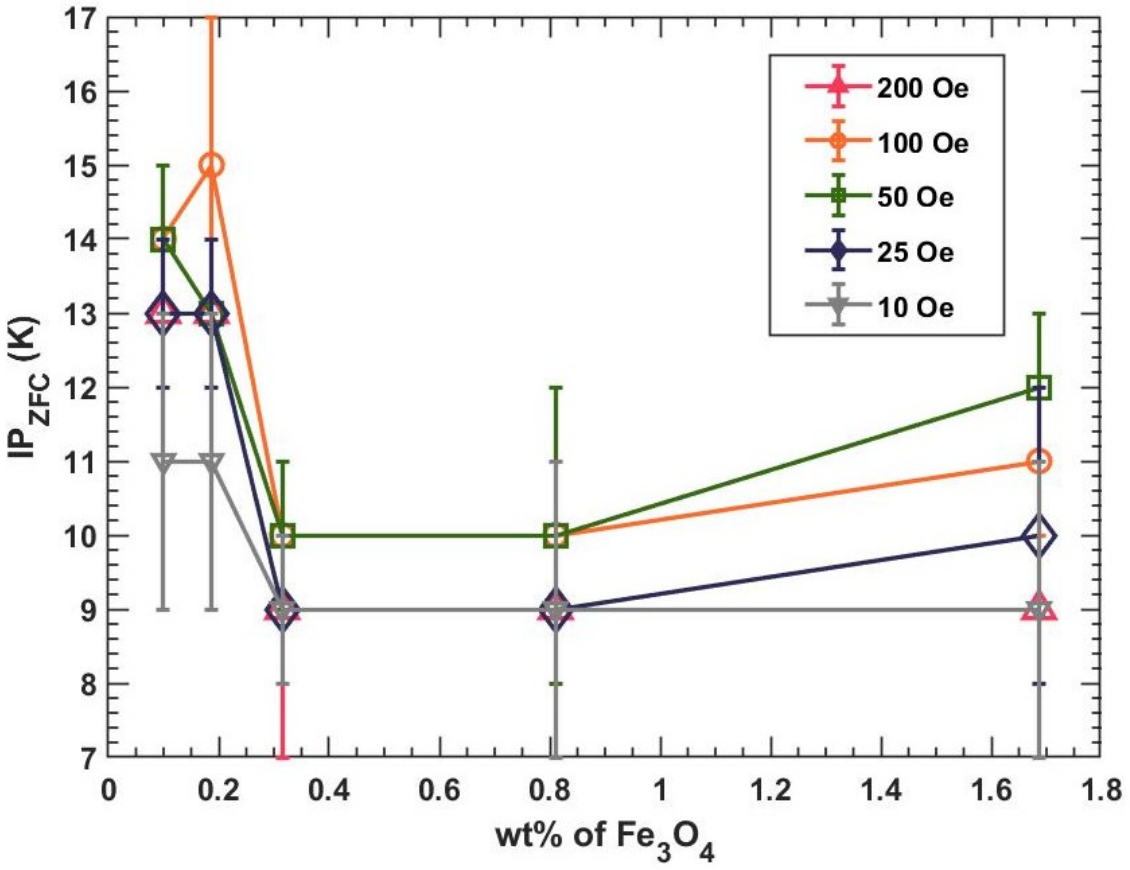

Figure 5.3 IP of M_{ZFC} plotted versus concentration for all applied fields. Error bars represent the uncertainty in T from taking the derivative and error from fitting near the inflection point.

important to recall our "bulk" T_B is 12 K, and there are only a few combinations of concentration and field where the IP is near that value. We will now look at the relationship between IP and applied field, to see if there is a clearer picture of the dependence of T_B on H.

In Fig. 5.4, there is a distinction between the three higher concentration samples (2x, 1x, and 0.5x) and the two lower concentration samples (0.25x and 0.125x). The IP is higher universally for the lowest two concentration samples, but especially so for high fields. The IP is below our expected value of $T_B = 12$ K for all the measurements on 1x and 0.5x. Those IPs closest to the expected value are for the

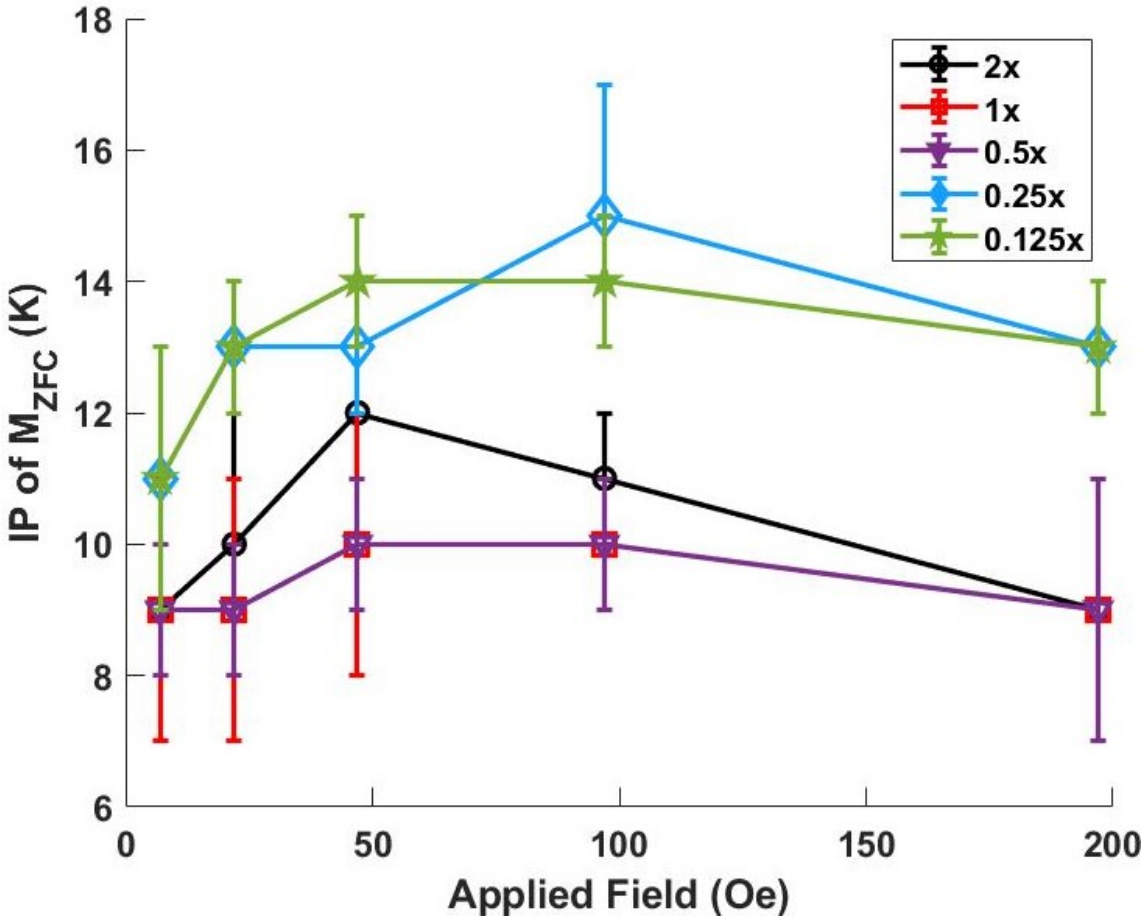

Figure 5.4 IP of M_{ZFC} plotted versus applied field for all concentrations. Error bars represent the uncertainty in T from taking the derivative and error from fitting near the inflection point.

lower-field measurements on 0.25x and 0.125x and surprisingly the middle range field measurements of the 2x sample. Again, the exact relationship between IP and field isn't clear, but the three higher concentrations' IPs have a flatter relationship with applied field, which is expected if it is true that T_B doesn't depend on H.

However, the lower concentrations' IPs show more curvature, which runs counter to that assumption. Both these samples, 0.25x and 0.125x, should have high enough NP separation distances in the polymer (the expected points in Fig. 4.5) that their interactions would be small enough to be negligible for every field measured (see

section 4.3.1). This change in curvature will be discussed again in future sections. In the meantime, we can say that there seems to be a distinction in the behavior of the IP versus field plots between the higher and lower concentration samples, and that IP is not constant with applied field. This is our first evidence that T_B may not in fact be independent of field and concentration. In the next section we will expand on this inflection point method to include the FC portion of our data.

5.3 The "Better" Method

The second analysis technique used in this work is based on the "good" method described in [3]. Of the three methods described in that paper, the authors came to the conclusion that the previous method of finding the inflection point of the ZFC curve was "ugly," but passable. The "bad" method was mentioned in section 5.1 as taking the peak of the ZFC data for T_B, which is highly inaccurate.

In the good method, the FC magnetization data is subtracted from the ZFC magnetization data and then the result is differentiated with respect to temperature. The derivative is then compared with a distribution of blocking temperatures, calculated from the distribution of volumes measured by TEM and using the relationship between blocking temperature and volume, given in Eq. 4.6. The value of K was determined to be the one that "maximized the coincidence" between the derivative and the distribution. The assumption of this model is that only the spread in volumes creates a dispersion in the blocking temperatures.

The "better" method employed here is a shortened version of this technique, in which we will stop short of comparing the derivative and the distribution. We instead examine only the maximum of the derivative (calculated in the same manner as in the previous section), which will correspond to the mode of the distribution in our case, because as discussed in 4.2, our nanoparticle volumes (and thus our blocking temperatures, see Eq. 1.3) follow a lognormal distribution.

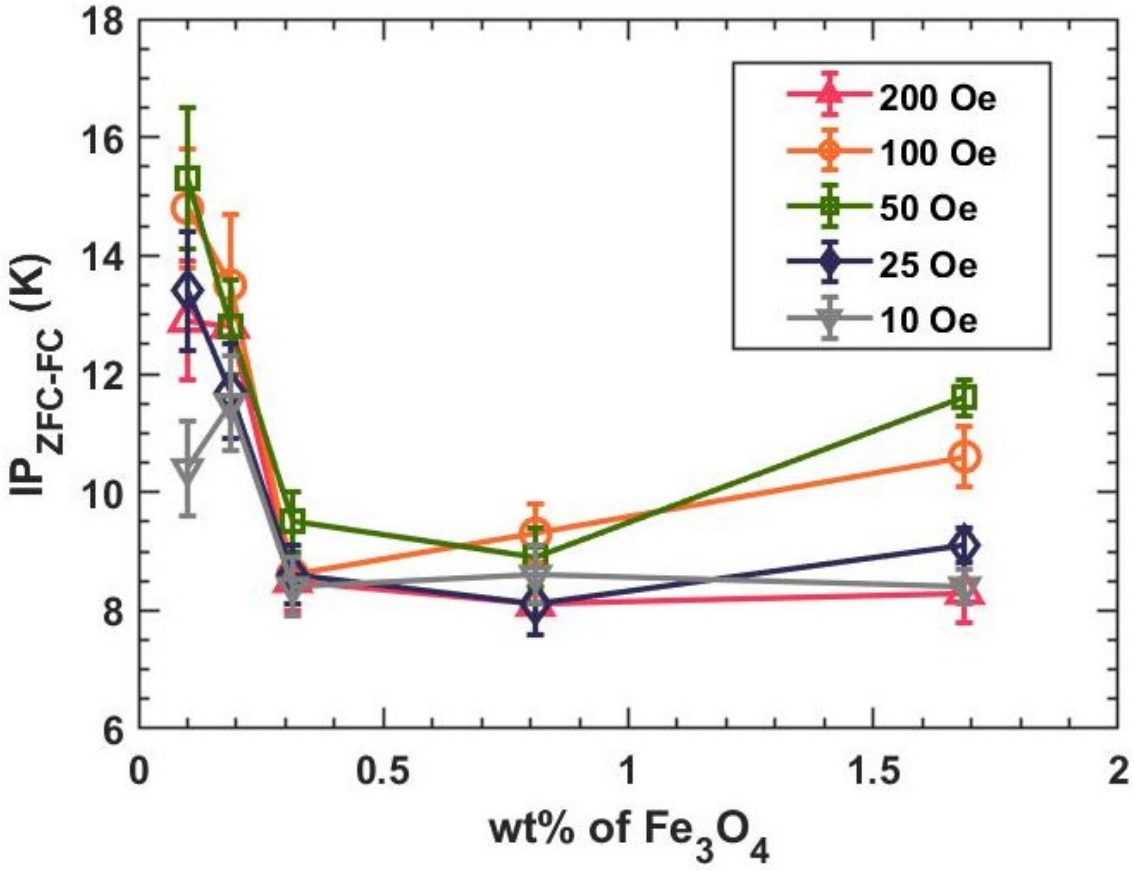

Figure 5.5 IP of M_{ZFC-FC} plotted versus concentration for all applied fields. Error bars represent the uncertainty in T when finding the derivative.

In Fig. 5.5, there is a clearer relationship between IP and concentration than in the previous section: as the concentration increases, the blocking temperature generally tends to decrease and then levels off with respect to increasing concentration for all but the 50 and 100 Oe values for the 2x sample. Again, we would expect the inflection point to be flat with respect to concentration if it were independent of concentration. The values closest to the expected $T_B = 12$ K are those for the lower concentrations (especially the second lowest, 0.25x, sample) and the 50 Oe measurement of the 2x sample. The lower concentrations agreeing most with the expected value is understandable, since we might assume, based on the expected average NP

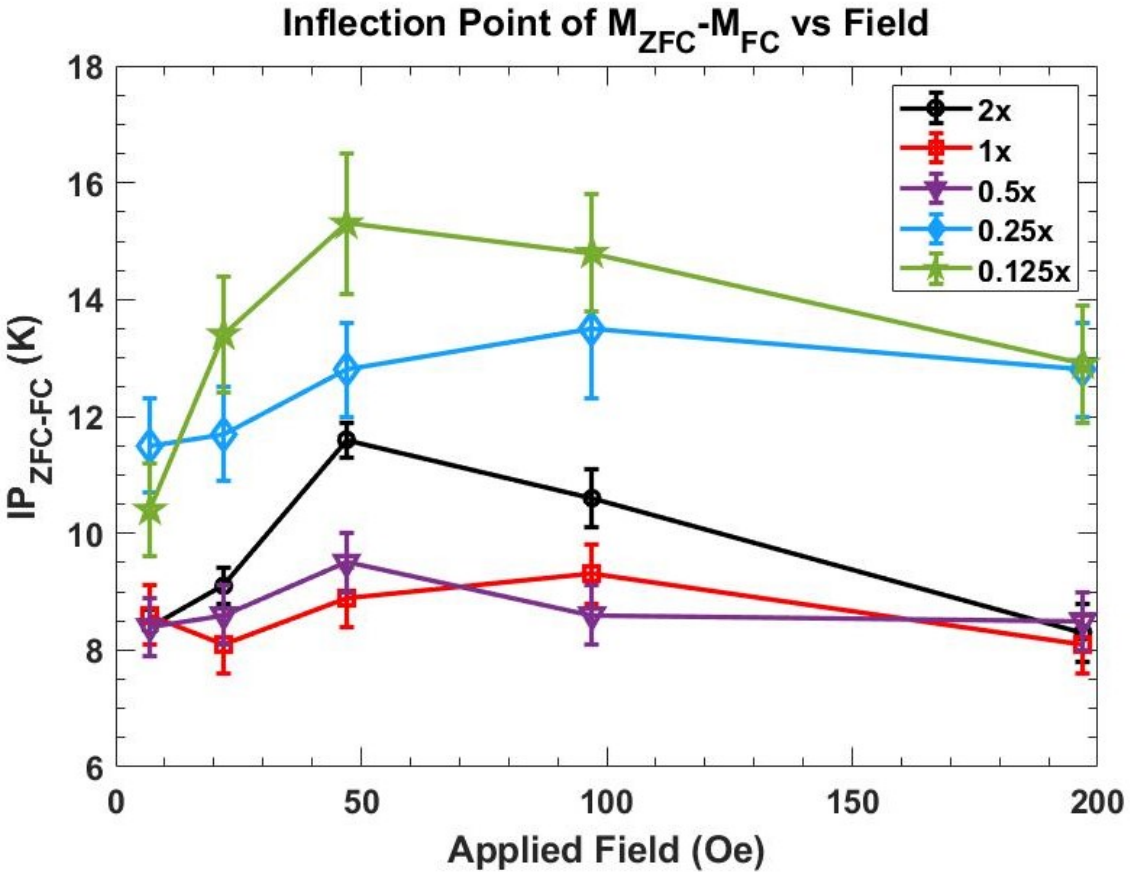

Figure 5.6 Resulting blocking temperatures from the "better" analysis method for all concentrations plotted versus appled field. Error bars represent the uncertainty in T when finding the derivative.

separation, that strength of dipole interactions in those NP samples is weaker and will thus effect the measurement less. The stronger dipole interactions in higher concentration samples may tend to increase the effects of the external field, thus aligning the NP moments more rapidly during the ZFC portion of the measurement and lead to the observation of a lower T_B. This trend is evident for most of the higher concentration data.

Moving on to Fig. 5.6, we can see that there is a separation between the inflection points versus applied field for the three highest concentration samples and

the two lowest concentration samples, as was the case in section 5.2. There is some curvature with respect to applied field, which is most evident for the 0.125x and 2x samples. This curvature of T_B versus H has been noted in [4], but with increasing curvature with larger concentration. However, this analysis relied on results of the traditional (or "bad") determination of T_B. The increase in curvature with concentration is attributed to increasing interactions [4], which we have examined in section 4.3.1.

In addition, the IPs are higher at all field values for the 0.125x and 0.25x samples, which again could be suggestive of lower dipolar interaction strength in those samples (see section 4.3.1). The 0.25x sample shows the best agreement with the expected T_B of 12 K, which is surprising, since we would expect the lowest concentration sample to be least altered by NP interactions, if that is the source of the discrepancy between the given IPs and the expected value.

We will compare these results directly to other methods in section 6.1, but in the following section we will present the first of two methods to fit the ZFC portion of the data to extract the mode blocking temperature.

5.4 The "Beyond the Blocking Model" Method

The previous methods of approximating the blocking temperature rely on the inflection point of the data collected, either for simply the ZFC curve or for the difference between the ZFC and FC curves. The "Beyond the Blocking Model" method directly fits the ZFC magnetization with an analytic expression that takes into account not only the polydispersity of the nanoparticle volumes, but also the probability of a given nanoparticle moment to flip at a given temperature [24]. In other words, in addition to the previous model's assertion that there is a spread in blocking temperatures with a spread of volumes, this model also takes into account the natural dispersion of blocking temperatures that occurs even in a monodisperse

sample. This dispersion in T_B results from the finite probability of a nanoparticle moment to flip at a given temperature. The equation for the magnetization of the FC portion of the curve is quasi-analytic and involves a double integral, which makes fitting this portion of the data directly both more complex and resource-intensive. For the purposes of this work, only the ZFC curves have been fit, and the corresponding FC curves are generated from the model using the resulting fit parameter, T_B. The data fitting was performed with Mathematica. A full description of the model and comparisons to data can be found in [24]. Here we will briefly describe the model and give the relevant equations.

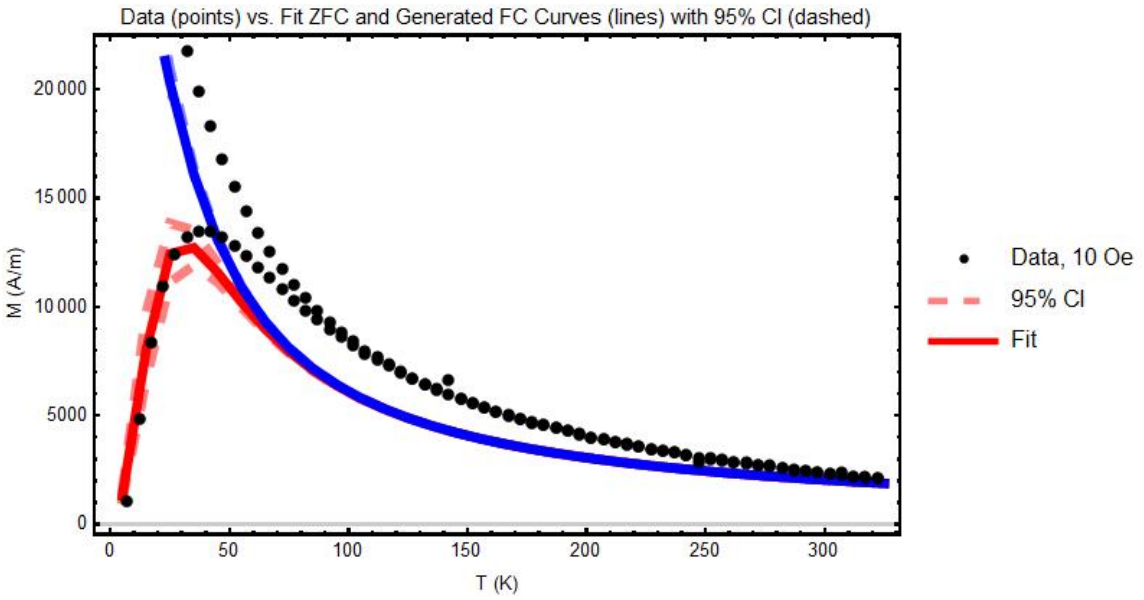

Figure 5.7 Result of fitting low-concentration (0.125x) and low-field ZFC data with the analytic expression.

The model assumes a low applied field, which must satisfy $\mu_0 M_s H \ll K$, which is satisfied best by our lower field values. The analytic expression for the ZFC

volume magnetization is:

$$M_{ZFC}(T) = \frac{M_s}{\langle V \rangle} \int_0^\infty V f(V, T_B) \left[\frac{\mu_0 M_s H}{3K} (1 - P(T,V)) + \mathcal{L}\left(\frac{\mu_0 M_s H V}{k_B T} \right) P(T,V) \right] dV \tag{5.5}$$

[24]. We can convert Eq. 5.5 into an integral over blocking temperatures, rather than volumes, using Eq. 1.3. In addition we can use Eq. 1.3 to solve for the anisotropy constant, K, in terms of T_B and substitute it into 5.5 to give us an equation that depends only on mode blocking temperature and various known quantities. The resulting expression for the ZFC magnetization is:

$$M_{ZFC}(T) = \frac{M_s}{\langle T_B \rangle} \int_0^{T_B} \theta f(\theta, T_B) \left[\frac{\alpha}{3\ln(\tau_m/\tau_0) T_B} (1 - P(T,\theta)) + \mathcal{L}\left(\frac{\alpha}{T_B T} \right) P(T,\theta) \right] d\theta$$
$$+ \frac{M_s}{\langle T_B \rangle} \int_{T_B}^\infty \theta f(\theta, T_B) \left[\frac{\alpha}{3\ln(\tau_m/\tau_0) T_B} (1 - P(T,\theta)) + \mathcal{L}\left(\frac{\alpha}{T_B T} \right) P(T,\theta) \right] d\theta \tag{5.6}$$

[24]. $\alpha = \mu_0 M_s V_m H / k_B$, where V_m is the mode NP volume, M_s is the saturation volume magnetization, μ_0 is the permeability of free space, H is the applied field (3 Oe less than the field set by the PPMS, see section 4.4), and k_B is Boltzmann's constant. We already determined that $\ln(\tau_m/\tau_0) = 26.5$ in section 1.2. In Eq. 5.6 $P(T,\theta)$ is the probability of a nanoparticle moment with blocking temperature θ to flip at temperature T, and $f(\theta, T_B)$ is the distribution of blocking temperatures with mode T_B.

The distribution of blocking temperatures is directly related to the distribution of volumes through Eq. 1.3. Thus, for our purposes (see section 4.2), $f(\theta, T_B)$ is a lognormal probability density function as follows:

$$f(\theta, T_B) = \frac{1}{\sigma\theta\sqrt{2\pi}} \exp\left(\frac{\left(\ln(\theta/T_B) - \sigma^2\right)^2}{2\sigma^2} \right) \tag{5.7}$$

Finally, we can use Eq. 5.6 to fit the measured ZFC data with T_B as the fitting parameter.

The fits were also characterized by their R^2 value, to give a reasonable goodness-of-fit to compare across concentrations and field values. We will present several results

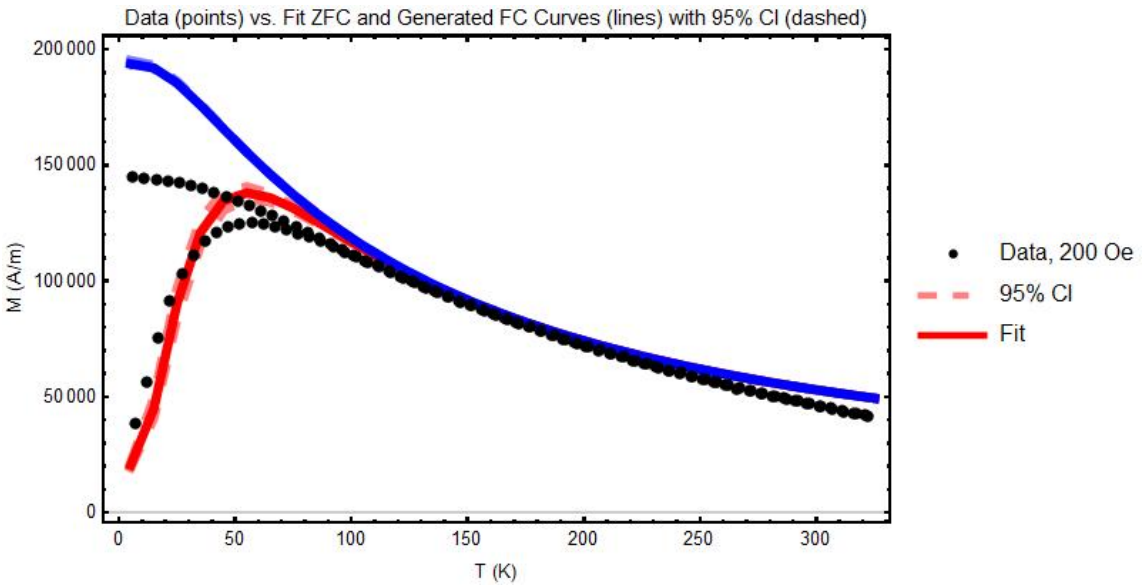

Figure 5.8 Result of fitting low-concentration (0.125x) and high-field ZFC data with the analytic expression.

of fitting the ZFC portion of the curve and then discuss the goodness-of-fit parameters across all the samples to assess the model's application to various combinations of measurement parameters.

For the lowest concentration sample (0.125x), we can see in Figs. 5.7 and 5.8 that the model describes the higher field measurement more accurately at least by eye for the ZFC portion, but the opposite is true for the FC portion. As a reminder, the FC expression was not used for fitting, so the solid and dashed FC curves have been generated from the resulting T_B from the ZFC fit. This result is unexpected, since the model is built on the assumptions of low-field and non-interacting NPs, which is assumed to be true for the lowest concentration samples.

For the highest concentration sample (2x), we can see the opposite result as for the lowest concentration sample (Figs. 5.9 and 5.10): the highest field measurement fit does not agree as well as the lowest field measurement fit for the ZFC data. The FC portions do not agree well for either field measurement, but this is not accounted

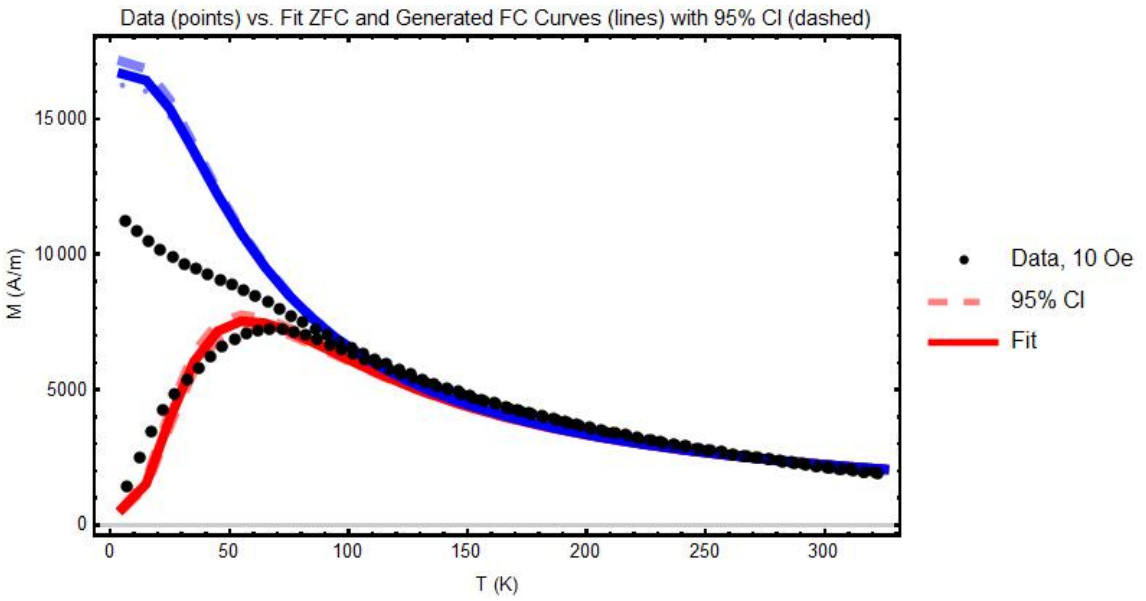

Figure 5.9 Result of fitting high-concentration (2x) and low-field ZFC data with the analytic expression.

for in our goodnss-of-fit. The decrease in goodness-of-fit of the ZFC expression with increasing applied field is the expected trend for the model. As is shown in Fig. 5.11, the two highest concentration samples (1x and 2x) follow this expected trend, but the opposite is true for the three lowest concentration samples (0.5x, 0.25x, and 0.125x).

For the 10 Oe measurement fit, the goodness-of-fit decreases away from 1 almost monotonically with decreasing concentration, except for the fact that the 0.25x sample and 0.125x sample are out of order (Fig. 5.11). One possible explanation for this is that the lower concentration samples have lower SNR, especially for the low-field measurements, and thus fitting those measurements may be more difficult, even when the model, by definition, is more applicable to those measurement parameters.

The 0.25x sample also achieves the R^2 value closest to 1 of all the samples for the highest field measurement. If we refer back to Fig. 4.5, we see that it was the 0.25x sample that had the highest average NP separation of the polymerized samples. It may be that the 0.25x sample behaves more like a lower concentration than the

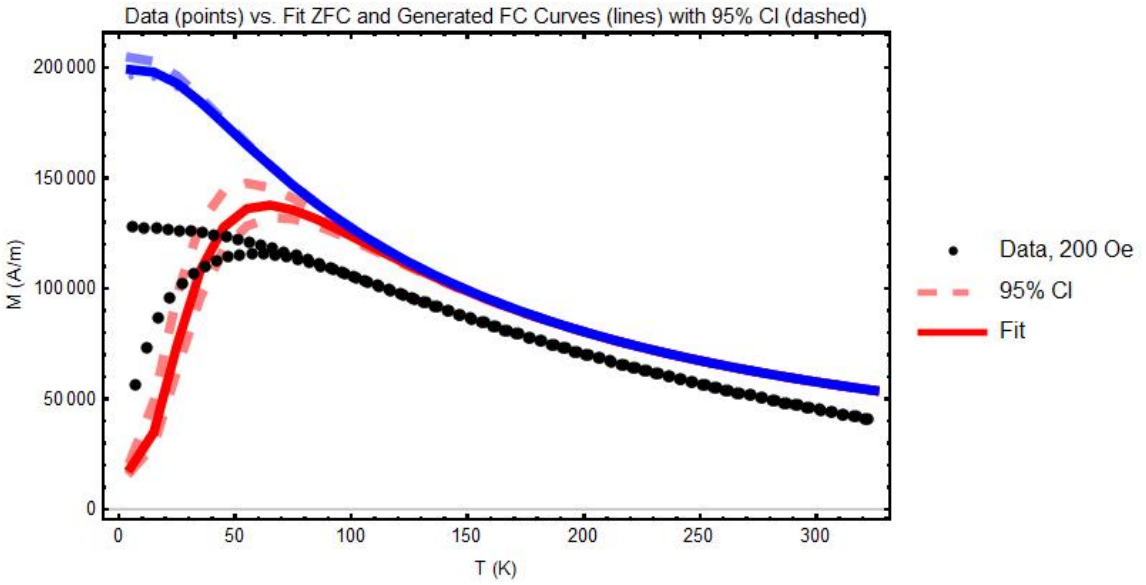

Figure 5.10 Result of fitting high-concentration (2x) and high-field ZFC data with the analytic expression.

0.125x sample because of its relatively larger average separation. Since the increasing R^2 value with increasing field for these samples is contrary to what is expected of the model, let us compare the resulting mode blocking temperatures from fitting across all samples to see if there are any other observations we can make about trends with concentration and applied field.

We can see in Fig. 5.12 that the mode blocking temperatures are lowest for the three lowest concentration samples, and relatively similar across the lower field values (10, 25, and 50 Oe). However, there is a dramatic increase in T_B with concentration for the two higher concentration samples, opposite to the general trend of the previous models. This is unexpected if we attribute a lower T_B to increased dipolar interactions in the sample. The lower concentration samples' results are closest to our expected "bulk" T_B, and especially so for the 25 and 50 Oe measurements.

In contrast to the previous analysis methods, when plotted versus applied field, the blocking temperatures resulting from fitting is drastically larger for the

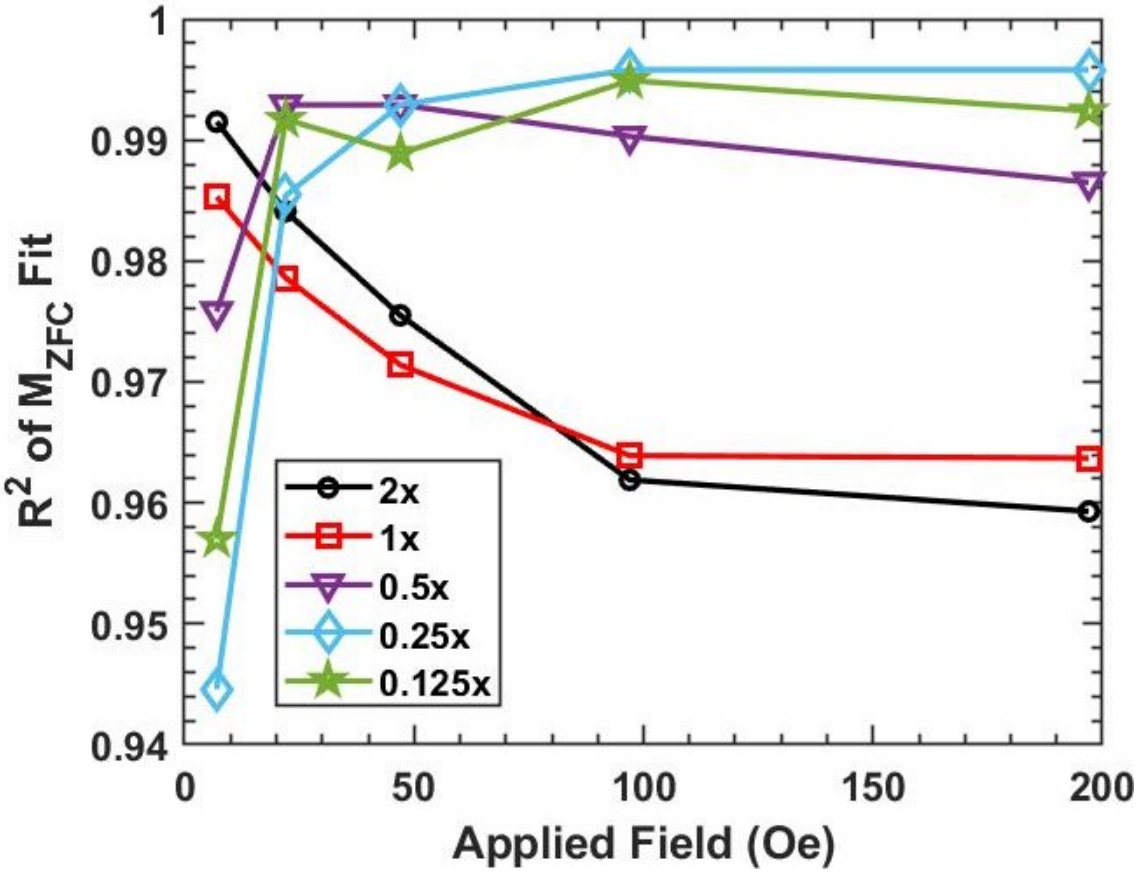

Figure 5.11 Resulting R^2 value from fitting with the analytic expression versus applied field. Error bars represent error from fitting.

two highest concentration samples and show much higher curvature (Fig. 5.13). The lower three concentration samples are flatter with applied field, and are grouped more closely in value. As mentioned in the previous section, there has been demonstration of an increase in curvature of the T_B versus applied field plots with increasing concentration [4]. Although those blocking temperatures were taken from the peak of the ZFC data, which is not a reliable way to find T_B, we do see a similar trend in Fig. 5.13.

From this behavior, it seems that there is a particular threshold at which the concentration of nanoparticles becomes high enough to introduce large enough

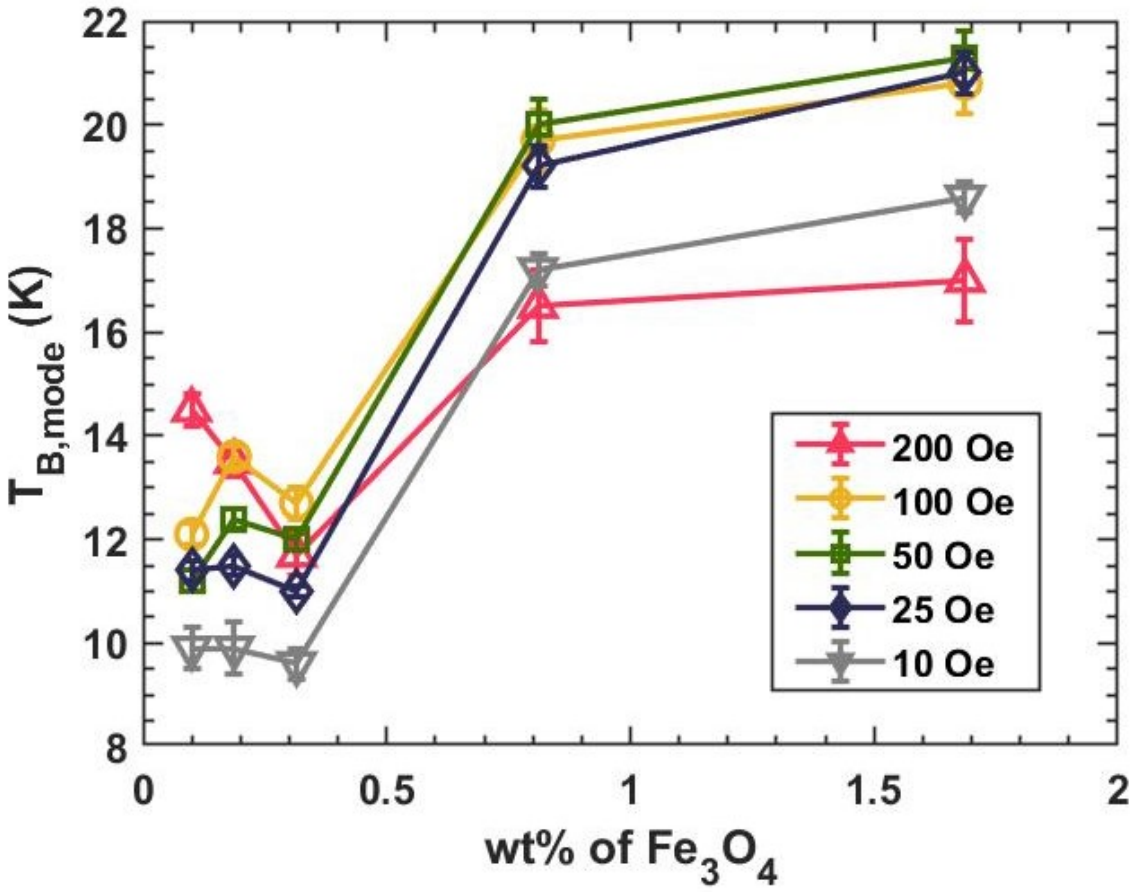

Figure 5.12 $T_{B,mode}$ from fitting, plotted versus NP concentration for all field values. Error bars represent error from fitting.

interactions that the model leads to very different predictions. The expected NP separation of the three lowest concentration samples is >50 nm, which should not interfere with any measurements with fields larger than 25 Oe (see section 4.3.1). Unfortunately, the theoretical description of this particular model has not yet been expanded to include interacting particles, but these results suggest a strong need for such an extension for concentrations above 0.32 wt% (0.5x sample). We will compare the methods directly in section 6.1, but in the next and final section in this chapter, we will attempt an extension of the Beyond the Blocking Model to correct for field strength.

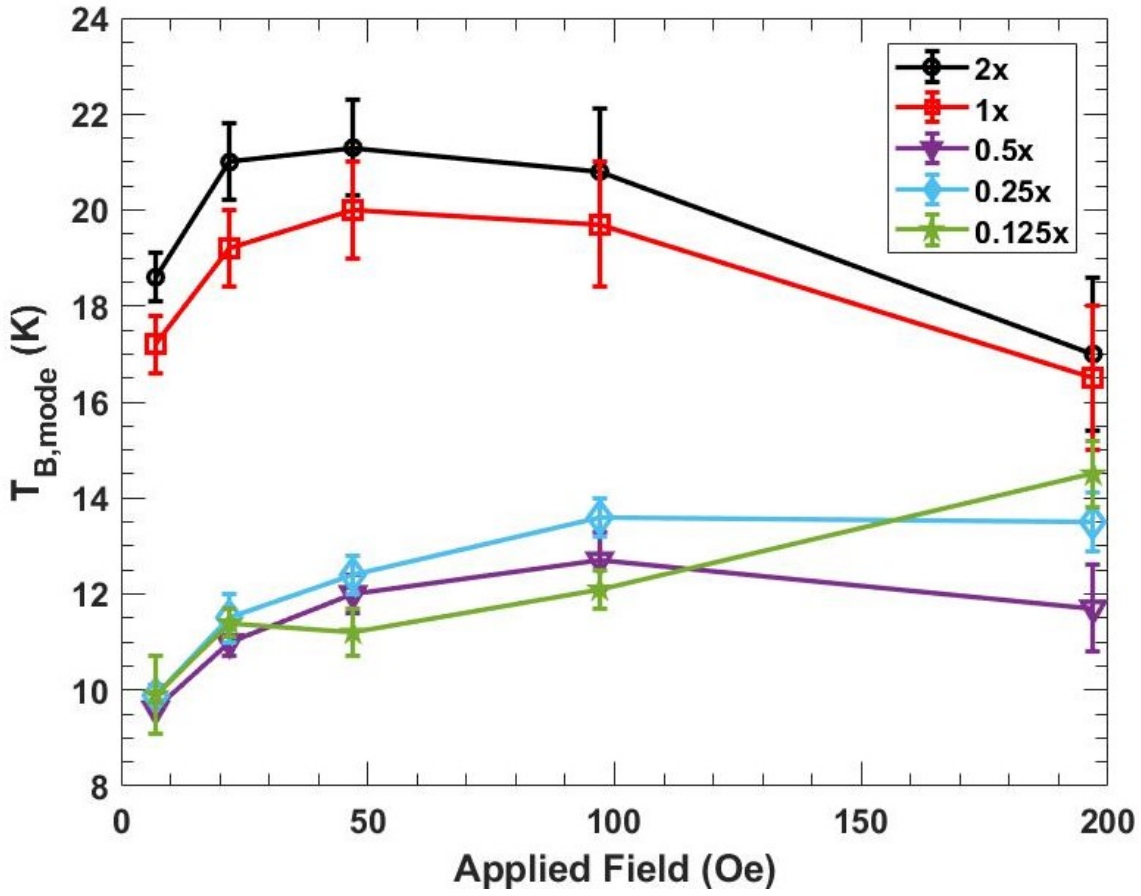

Figure 5.13 Result of fitting ZFC data with the analytic expression. The fit parameter is the mode blocking temperature (y-axis), plotted versus the applied field for all concentrations.

5.5 The Field-Corrected Analytic Method

As seen in section 5.4, the fit of M_{ZFC} with the given analytic expression still shows variation of T_B with applied field. A full extension of the model to account for the applied field strength is underway, but in the meantime we can apply a slight alteration to the analytic model to see if we can easily correct this variation of T_B with field. We will refer to the model in this section as the "field-corrected" analytic model.

For the field-corrected model, we will include a simple adjustment to the barrier height in the Néel-Arrhenius Law (Eq. 1.3). In our samples which have randomly oriented easy axes, the barrier height changes from KV to $KV(1-h)^2$, where h is the ratio of the applied field H to the anisotropy field, $H_k = 2K/(\mu_0 M_s)$ [4]. The reader can refer back to Eq. 5.6 for the full expression for M_{ZFC}. This correction does not make any assumptions about the strength of the applied field, unlike the previous model, which assumes a small field.

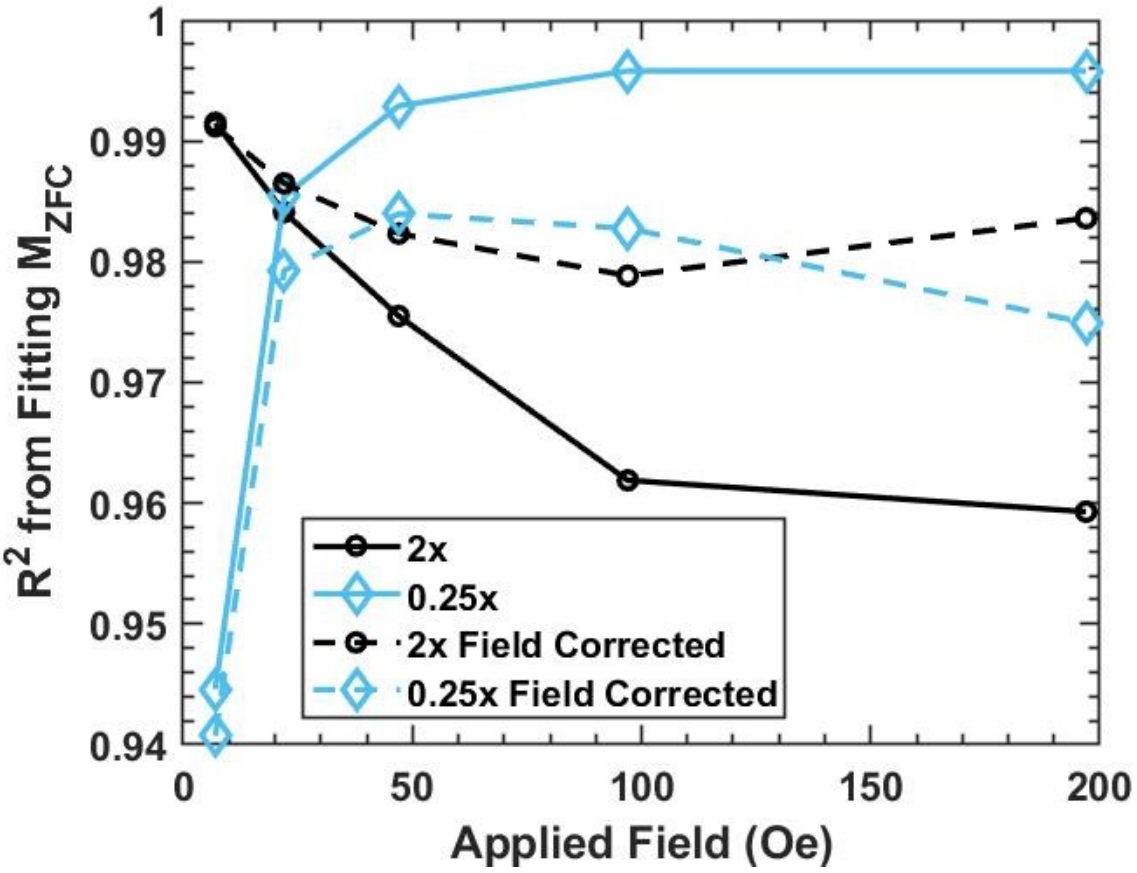

Figure 5.14 Comparison of R^2 values of the analytic fit models for 2x and 0.25x samples.

In the field-corrected model, using mode blocking temperature as the fitting parameter becomes more difficult mathematically. For further simplicity, we have

used K as the fitting parameter and used that result to find T_B from Eq. 1.3 with the altered barrier height. Interestingly, since this model includes a field dependence in T_B, our expected, "bulk" value is no longer simply 12 K, but instead is a range of values between roughly 12 K and 8 K which decreases monotonically with increasing applied field. The expected values for T_B change slightly depending on the saturation magnetization of the sample, but in our case this only leads to a difference of about 0.2 K with the largest difference of M_s.

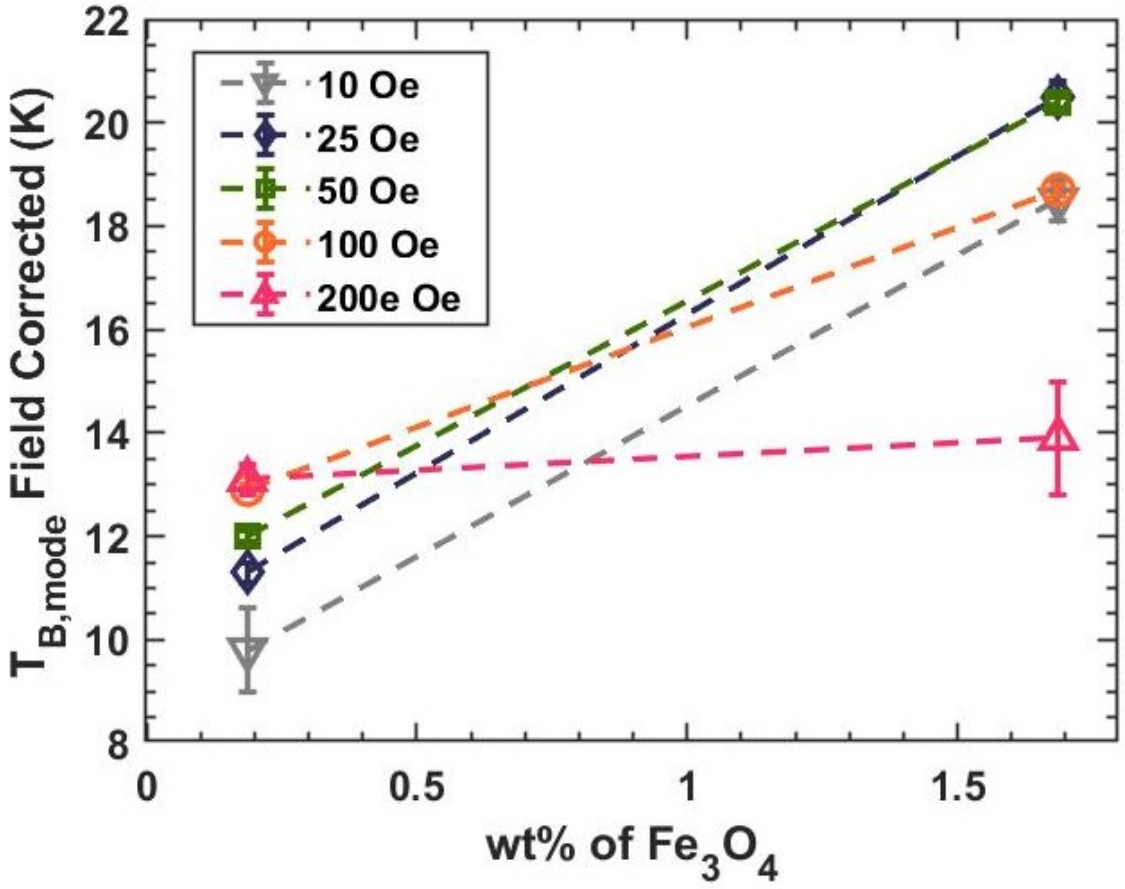

Figure 5.15 Plot of mode blocking temperature versus concentration for the field-corrected fit from measurements on the 2x and 0.25x samples. $T_{B,mode}$ is calculated from the fitting parameter, K, and error bars represent error from fitting.

The extremes of our previous analytic fit model were the 2x sample and the 0.25x sample, which had the most variation in R^2 values between the lowest and highest applied fields (Fig. 5.11). We can attempt our adjustment of the model for these samples to see if the fit improves and see if the resulting parameters' relationship to field changes with the field-corrected model.

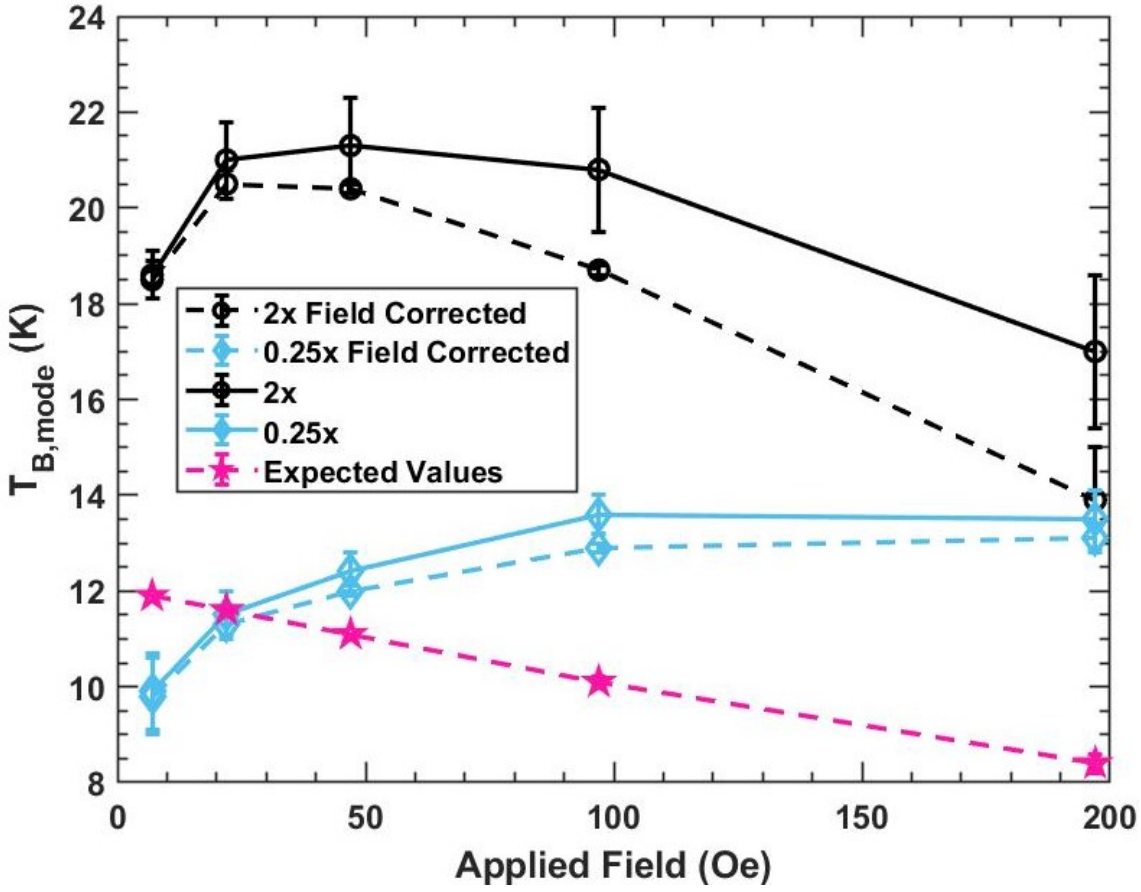

Figure 5.16 Plot of mode blocking temperature versus applied field for the field-corrected and original analytic fits for the 2x and 0.25x samples. Error bars represent error from fitting.

In Fig. 5.14, the goodness-of-fit parameter becomes closer to 1 for the 2x sample across all of the higher field values, which is the desired result of including the barrier height adjustment. However, R^2 decreases away from 1 across all fields for

the 0.25x sample. It should also be noted that with the field-corrected model, the R^2 value for fields 25 Oe and above is much more similar for the two samples. The large divergence in goodness-of-fit that was the case for nearly all of the measurements in the original analytic fit method only remains for the 10 Oe measurement. This suggests that the method performs roughly as well for these two samples, except in the 10 Oe case.

Now let us examine how the field-corrected blocking temperature changes with concentration. This plot is less illustrative than for previous sections, since this analysis focuses on only two samples, but it is still useful to note the general upward trend of T_B with concentration, for all fields except 200 Oe where the plotted line is flat (Fig. 5.15). The increase of T_B with increasing concentration was also apparent for the previous analytic method results, but it is opposite that of the IP methods in sections 5.2 and 5.3.

Moving now to discuss the trend of T_B with applied field, we have Fig. 5.16. We can see that there was little change in the predicted blocking temperatures for the 10 and 25 Oe measurements. This agreement is to be expected, since those fields are considered "small" and should be within the scope of both theoretical descriptions, as mentioned in the definition of the original analytic model. Interestingly, the only good agreement between the calculated values and our expected values of T_B is for the 25 Oe measurement of the 0.25x sample for both analysis methods. This is the sample that showed the best average NP separation and the fewest number of touching NPs (section 4.3.1). Even though the blocking temperatures of the field-corrected method are less for the higher field values, they still do not agree well with our expected values for fields over 25 Oe. This is much more apparent for the 2x sample. The calculated blocking temperatures are almost overlapping for 200 Oe in the field-corrected method, which we saw in Fig. 5.15 as a flat line for 200 Oe, but they are still far from the anticipated blocking temperature for that field.

Now that we have used several models to analyze our ZFC/FC data, and we have found the blocking temperaures resulting from each, we will move on to comparing all the methods directly and providing recommendations to researchers based on our results.

Chapter 6

Conclusions and Suggestions for Future Work

In the final chapter, we will compare the results found in chapter 5 for blocking temperature and also compare the resulting anisotropy values for some of the samples. After that, we will give final thoughts to researchers who interact with this data and analysis. Finally in section 6.2, we will discuss some possible extensions of this work (both custom thin film media for self-assembly and ZFC/FC measurements on nanoparticle samples).

6.1 Comparison of ZFC/FC Results

We have seen the results of these methods individually, and the reader will have noted that the results are somewhat mixed. Now we will compare the resulting blocking temperatures and anisotropy values more directly to see how well they agree with each other and with the expected bulk value of $K = 13.5$ kJ/m^3. It is important to note that because of the relationship between T_B and K in Eq. 1.3 and the field-corrected version, these two are not orthogonal parameters. Therefore, when we saw changes in T_B with field and concentration in the previous sections, that will also lead to a change in K with these measurement parameters. In this section we will call the resulting K's the "effective anisotropy values" and denote them as K_{eff} since characterizing them as constants is misleading.

As an aside, this K_{eff} is not the same as that given in Eq. 4.5 in section 4.3.2. If we were to use that equation with our values of T_B from any analysis method and

the values of T_0 we found earlier, it would yield negative values of K_{eff}, which is unphysical.

First, we will compare the predictions as in [3], by taking the ratios of the resulting blocking temperatures. The resulting blocking temperatures from the inflection point methods (both the "ugly" and "better" methods), agree reasonably well for all samples and field values (see Fig. 6.1), as their ratios are roughly equal to 1 universally.

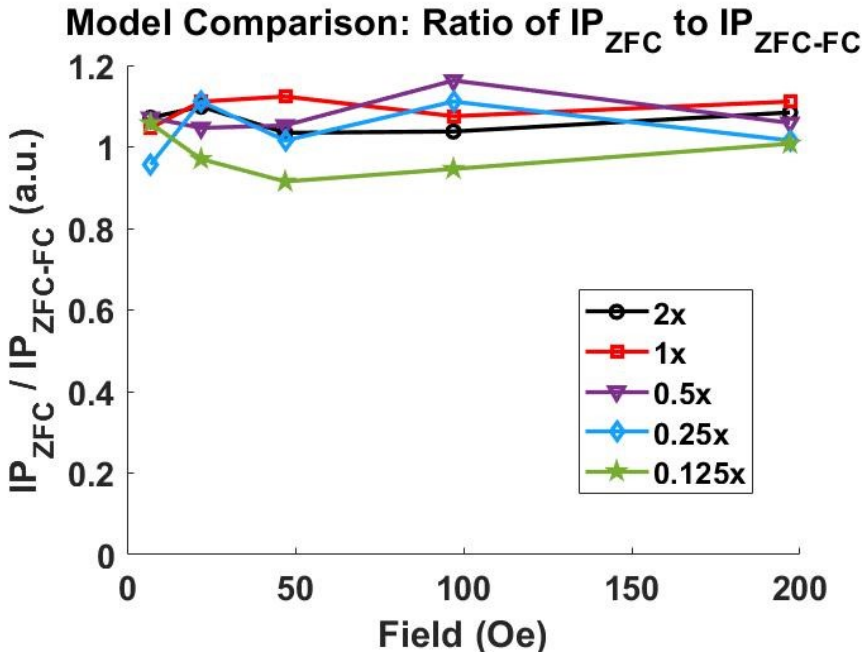

Figure 6.1 The ratio of the inflection point of the ZFC curve to the inflection point of ZFC minus FC.

Since the two IP models agree well, we will only compare the "better" method results to those of the analysis methods using fitting, for simplicity. In Fig. 6.2, we see that the ratio of the inflection point to the mode blocking temperature from fitting without correcting for field (section 5.4) depends strongly on concentration, but is relatively flat for all fields, except in the case of the 0.125x concentration sample. The field correction increases the ratios slightly for both the 0.25x sample and the 2x

sample (dashed lines), but this only improves the agreement for the 2x sample, since the ratio for the 0.25x sample is already close to 1 for all fields 25 Oe and higher. The 0.25x sample had the highest average nanoparticle separation as shown in section 4.2, and its expected average separation should lead to small dipole interactions for 25 Oe measurements and higher, as discussed in section 4.3.2. This good agreement range of fields for the 0.25x sample are also the measurement parameter combinations which resulted in the best R^2 value from the analytic fit of M_{ZFC} in section 5.4.

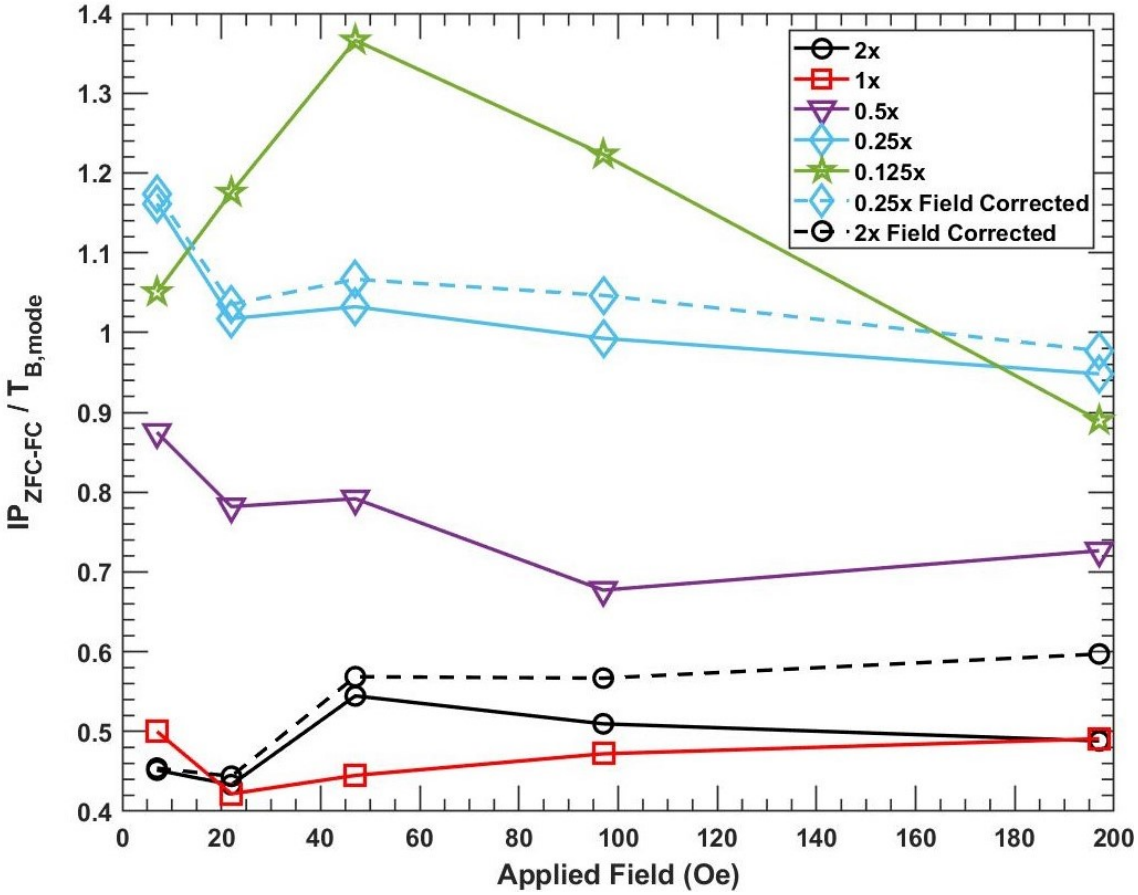

Figure 6.2 The ratio of the inflection point of ZFC-FC to the mode blocking temperature from fitting, plotted versus applied field.

However, while agreement between the models is heartening, it does not tell us which of the predictions is correct. We can now compare the resulting effective

anisotropy values for our extremes discussed in section 5.5 (the 0.25x and 2x samples) to the bulk to see if there are any that agree particularly well with the accepted experimental bulk value and check if there are any other indications of which measurement parameters yield expected results.

In Fig. 6.3, we can see that the analytic models' predictions do not agree well with the bulk value for any applied field with the highest concentration. These models are explicitly built on the assumption that the NPs do not interact, so this result is not particularly suprising if the results from section 4.3.1, that showed higher concentration leads to larger NP interactions, is correct. The IP methods' predicted effective anisotropy values are closer to the bulk value, particularly so for the 50 Oe measurement for this sample.

For the 0.25x sample (Fig. 6.4), we see that the effective anisotropy values are much closer to the bulk value for all measurement parameters and analysis methods, except the field-corrected result with 200 Oe field. It is worth noting at this point that because the field-corrected method used K_{eff} as the fit parameter, it was allowed to vary, so even though we saw lower values of T_B which were closer to the bulk value, the anisotropy value did not improve. Again, we will mention that these are not orthogonal parameters, but it is possible that by letting K_{eff} vary in the fit, we instead prioritized T_B remaining flat with field, instead of K_{eff}. Instead letting T_B vary as the fit parameter could be a more correct implementation of the field-corrected analytic fit in the future.

Even very recent works in this area do not take into account the effects of field strength and concentration, and these results refute the common assumption that field strength and concentration do not affect the results of ZFC/FC measurements. There is pervasive misunderstanding of the analysis of ZFC/FC data which must be corrected to move forward in the quest to accurately characterize NP systems with this measurement. As recently as spring of 2021, a paper submitted for publication

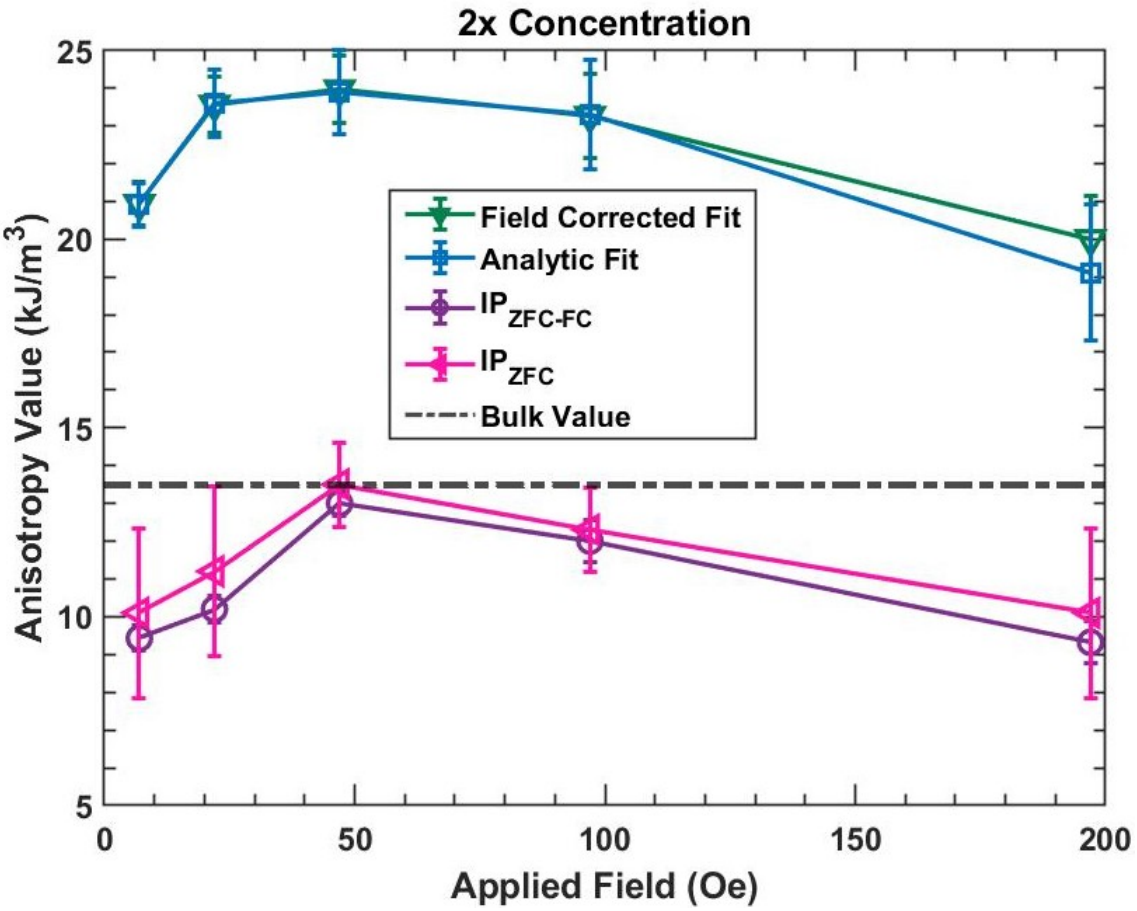

Figure 6.3 Comparison of the anisotropy values from all four analysis methods for the 2x sample versus applied field. Bulk value for K_{eff} is shown as a dashed line.

by researchers at Clemson University, myself, and my advisor received skeptical feedback from a reviewer about our choice of the "ugly" method to determine blocking temperatures. This is despite the method having been published more than 5 years ago [3] and other subsequent works reiterating the same issues with the traditional use of the ZFC peak for T_B [24].

In the results presented here, the agreement of all the methods over certain field values and for certain concentrations is a strong suggestion that regardless of the analysis method, these are desireable parameters for experiment. Overall, the inflection point methods and all fitting methods are appropriate and agree well for

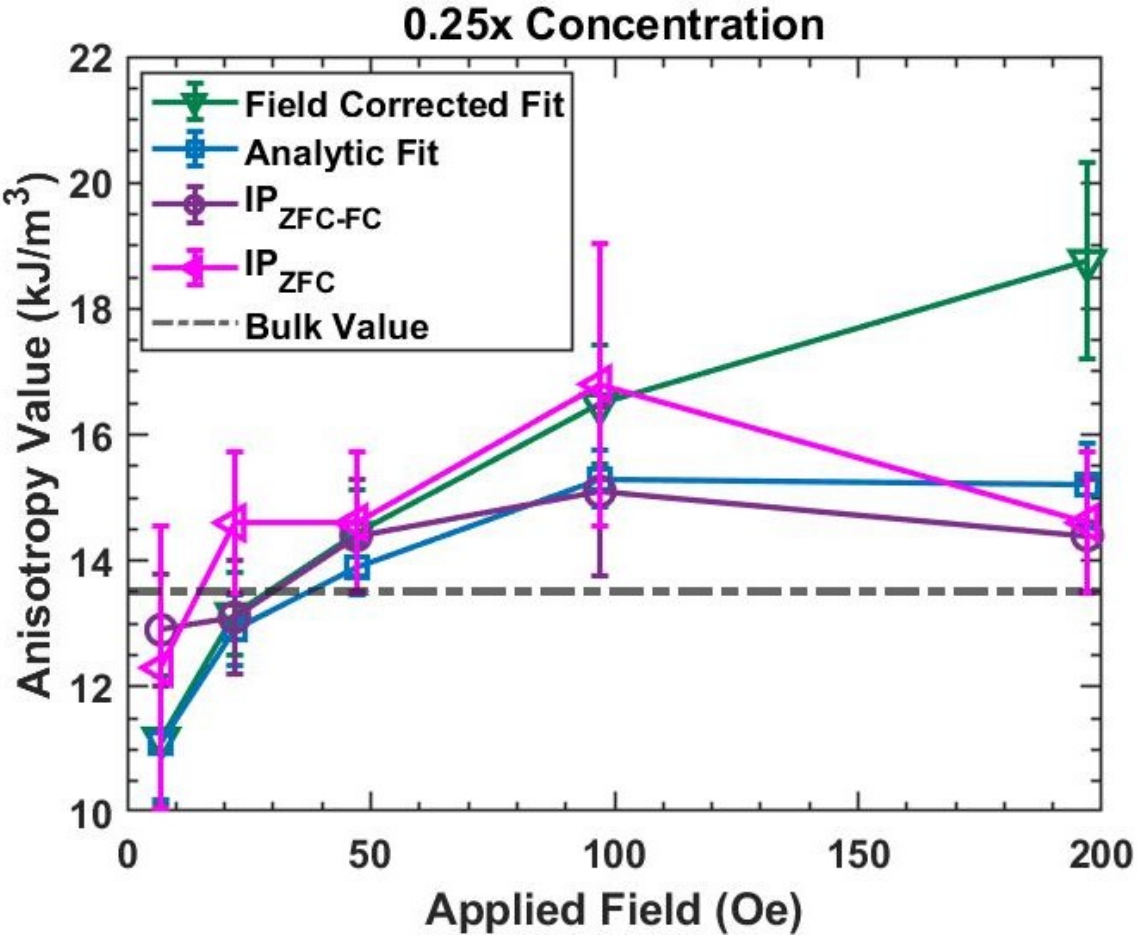

Figure 6.4 Comparison of the annisotropy values from all four analysis methods for the 0.25x sample versus applied field. Bulk value for K_{eff} is shown as a dashed line.

the 0.25x concentration and 25 and 50 Oe field measurements. The resulting values from those combinations of parameters in turn correspond most closely to the expected bulk values of T_B and K. It is my recommendation, based on this work, that experimentalists should prepare their samples and perform measurements with a concentration of 0.2 wt% or less of NPs and field values of at most 50 Oe to achieve the most accurate results. Although the SNR of a measurement will increase with

increased concentration, it is necessary to balance this with the changes that interactions can have on the results.

When considering concentration, it will also be important to confirm with imaging that the average NP separation in the polymer is sufficient (>60 nm) that interactions can be ignored. This work shows, by using standard, commercial nanoparticles and careful measurement technique, that both applied field strength and concentration have strong impact on the results of ZFC/FC measurements. Iron oxide nanoparticles in particular are used for various medical applications, including MRI contrast, and proposed uses for cancer treatments are being studied [32, 6]. The accurate characterization of these particles is vital if their intended uses could impact patient outcomes.

6.2 Suggestions for Future Work

In an effort to guide researchers who will expand on this work in the coming years, I present here a few suggestions as jumping off points.

It would be useful in the future to perform additional measurements to confirm the anisotropy values presented in the previous chapter, because, as a "constant," it should only be dependent on the material, which is not the case for any of the analysis methods or across the majority of measurement parameters presented. Some possibilities for this include using the Law of Approach to saturation to fit the magnetization versus field measurements for the particles in their superparamagnetic state [37], Dynamic Magnetic Susceptibility (DMS) measurements as a function of temperature [25], Mössbauer spectroscopy [28], or measuring the magnetization relaxation and applying the Néel-Arrhenius Law [9].

Particles characterized with ZFC/FC or one of the aforementioned measurements could also be utilized for magnetic field-driven self-assembly to assess the im-

pacts of K_{eff} on the resulting NP patterns and the dynamics of assembly [46]. This would provide further ability to tune the self-assembly process and resulting shapes.

Another clear next step is to extend the analytic model to account for particle interactions, since the concentration of NPs causes wide variation in the predicted blocking temperature and anisotropy values. The ZFC/FC measurement has also been performed on granular magnetic media, in which the grains are typically close-packed in the recording layer, being separated by only nanometers, which we saw in section 4.3.2 can lead to strong interactions. ZFC/FC has been used to characterize granular media already [31, 51], so this is an established technique that may be useful to examine our custom media for self-assembly. Some work has been done on the effects of concentration in these systems in experiment, but there is opportunity to use the Beyond the Blocking Model template to expand our theoretical description to interacting systems and apply it to granular media characterization.

Additionally, most of chapter 3 focused on the results of measuring magnetic response of custom media with a crystalline SUL and its associated challenges with the VSM. However, since an amorphous SUL of CoFeB may be better suited for writing with a commercial write head, it will be necessary to characterize these systems in greater detail to move forward with their use in self assembly. Once the appropriate SUL is chosen, there are many options for oxides to co-sputter with the recording layer material, CoCrPt, or it may be useful to sputter the recording layer in an oxygenated environment. These are all useful starting points for further exploration of custom magnetic media to expand on the results provided here.

Bibliography

[1] D. A. Allwood, Gang Xiong, M. D. Cooke, and R. P. Cowburn. Magneto-optical Kerr effect analysis of magnetic nanostructures. *Journal of Physics D: Applied Physics*, 36(18):2175–2182, 2003.

[2] William Fuller Brown. Thermal fluctuations of a single-domain particle. *Phys. Rev.*, 130:1677–1686, jun 1963.

[3] I. J. Bruvera, P. Mendoza Zélis, M. Pilar Calatayud, G. F. Goya, and F. H. Sánchez. Determination of the blocking temperature of magnetic nanoparticles: The good, the bad, and the ugly. *Journal of Applied Physics*, 118(18):1–7, 2015.

[4] R. W. Chantrell, N. S. Walmsley, J. Gore, and M. Maylin. Theoretical studies of the field-cooled and zero-field cooled magnetization of interacting fine particles. *Journal of Applied Physics*, 85(8 II A):4340–4342, 1999.

[5] G. Choe, E.N. Abarra, B.R. Acharya, Z.D. Yang, M. Zheng, K.E. Johnson, and J.N. Zhou. Role of Oxygen Incorporation in Co-Cr-Pt-Si-O Perpendicular Magnetic Recording Media. *IEEE Transactions on Magnetics*, 40(4):2498–2500, 2004.

[6] Ki Young Choi, Hyunjin Chung, Kyung Hyun Min, Hong Yeol Yoon, Kwangmeyung Kim, Jae Hyung Park, Ick Chan Kwon, and Young Jeong. Self-assembled hyaluronic acid nanoparticles for active tumor targeting. *Biomaterials*, 31:106–114, 2009.

[7] Edwin L Crow and Kunio Shimizu. *Lognormal distributions: theory and applications*. M. Dekker, 1988.

[8] B D Cullity and C D Graham. *Introduction to Magnetic Materials*. John Wiley & Sons, Inc, 2nd edition, 2009.

[9] D. Dickson, N.M.K. Reida, C. Hunta, H.D. Williams, M. El-Hilo, and K. O'Grady. Determination of f0 for fine magnetic particles. *Journal of Magnetism and Magnetic Materials*, 125(3):345–350, 1993.

[10] C Dolbashian, B L Chavez, M Bauer, M Budi, J S Andrew, and T M Crawford. Magnetic properties of aligned multiferroic janus nanofiber agglomerates measured with the scattered magneto-optical kerr effect. *Journal of Physics D: Applied Physics*, 53(19):195002, mar 2020.

[11] Joshua B Edel, Alexei A Kornyshev, and Michael Urbakh. Self-Assembly of Nanoparticle Arrays for Use as Mirrors, Sensors, and Antennas. *ACS Nano*, 2013.

[12] George Francis Fitzgerald. On the Rotation of the Plane of Polarization of Light by Reflection from the Pole of a Magnet. *Proceedings of the Royal Society of London*, 25:447–450, 1876.

[13] E P Furlani. Analysis of particle transport in a magnetophoretic microsystem. *J. Appl. Phys*, 99:24912, 2006.

[14] Heng Gong, Maithri Rao, David E Laughlin, and David N Lambeth. Highly oriented NiFe soft magnetic films on Si substrates. *Journal of Applied Physics*, 85(8):5750–5752, 1999.

[15] J R Henderson and T M Crawford. Repeatability of magnetic-field driven self-assembly of magnetic nanoparticles. *J. Appl. Phys*, 109:7–329, 2011.

[16] Amelie Heuer-Jungemann, Neus Feliu, Ioanna Bakaimi, Majd Hamaly, Alaaldin Alkilany, Indranath Chakraborty, Atif Masood, Maria F. Casula, Athanasia Kostopoulou, Eunkeu Oh, Kimihiro Susumu, Michael H. Stewart, Igor L. Medintz, Emmanuel Stratakis, Wolfgang J. Parak, and Antonios G. Kanaras. The role of ligands in the chemical synthesis and applications of inorganic nanoparticles. *Chemical Reviews*, 119(8):4819–4880, 2019.

[17] Y. Inaba, T. Shimatsu, T. Oikawa, H. Sato, H. Aoi, H. Muraoka, and Y. Nakamura. Optimization of the SiO2 content in CoPtCr-SiO2 perpendicular recording media for high-density recording. *IEEE Transactions on Magnetics*, 40(4 II):2486–2488, 2004.

[18] H S Jung. Improvement of Magnetic Properties of Granular Perpendicular Recording Media by Using a FCC Nonmagnetic Intermediate Layer With Stacking Faults, 2007.

[19] HS Jung, U Kwon, and M Kuo. Effect of Oxygen Incorporation on Microstructure and Media Performance in CoCrPt–SiO 2 Perpendicular Recording Media. *Magnetics, IEEE*, 43(2):615–620, 2007.

[20] Toshiaki Keitoku, Jun Ariake, Naoki Honda, and Kazuhiro Ouchi. Preparation of Co-Cr-Pt alloy film with high perpendicular coercivity and large negative nucleation field. *Journal of Magnetism and Magnetic Materials*, 235:34–39, 2001.

[21] Kim Kong Tham, Shintaro Hinata, Ryosuke Kushibiki, Shin Saito, and Tanaka KK Kikinzoku Kogyo. B 2 O 3 : Grain boundary material for high-K u CoPt–oxide granular media with low degree of intergranular exchange coupling. *Japanese Journal of Applied Physics*, 55:07MC06, 2016.

[22] R Lavrijsen, P V Paluskar, C T J Loermans, P A Van Kruisbergen, J T Kohlhepp, H J M Swagten, B Koopmans, and E Snoeck. Magnetism in Co80-xFexB20: Effect of crystallization. *Journal of Applied Physics Journal of Applied Physics Applied Physics Letters*, 109(102):093905, 2011.

[23] Daishun Ling, Michael J. Hackett, and Taeghwan Hyeon. Surface ligands in synthesis, modification, assembly and biomedical applications of nanoparticles. *Nano Today*, 9(4):457–477, 2014.

[24] K. L. Livesey, S. Ruta, N. R. Anderson, D. Baldomir, R. W. Chantrell, and D. Serantes. Beyond the blocking model to fit nanoparticle ZFC/FC magnetisation curves. *Scientific Reports*, 8(1):11166, dec 2018.

[25] Lorena Maldonado-camargo, Mythreyi Unni, and Carlos Rinaldi. Biomedical Nanotechnology. In S. Petrosko and E. Day, editors, *Magnetic Characterization of Iron Oxide Nanoparticles for Biomedical Applications*, volume 726, pages 47–71. Humana Press, New York, NY, 2011.

[26] Jim J Miles, David Mca Mckirdy, Roy W Chantrell, and Roger Wood. Parametric Optimization for Terabit Perpendicular Recording. *IEEE Transactions on Magnetics*, 39(4):1876–1890, 2003.

[27] Douglas C. Montgomery. *Design and analysis of experiments*. John Wiley & Sons, New York, 4th edition, 1997.

[28] Steen Mørup. Magnetic hyperfine splitting in mössbauer spectra of microcrystals. *Journal of Magnetism and Magnetic Materials*, 37(1):39–50, 1983.

[29] W. Mróz, S. Burdyńska, A. Prokopiuk, M. Jedyński, B. Budner, and M. L. Korwin-Pawlowski. Characteristics of carbon films deposited by magnetron sputtering. *Acta Physica Polonica A*, 116(SUPPL.):120–122, 2009.

[30] F. E. Neumann. Allgemeine gesetze der inducirten elektrischen ströme. *Annalen der Physik*, 143(1):31–44, 1846.

[31] W. C. Nunes, L. M. Socolovsky, J. C. Denardin, F. Cebollada, A. L. Brandl, and M. Knobel. Role of magnetic interparticle coupling on the field dependence of the superparamagnetic relaxation time. *Physical Review B - Condensed Matter and Materials Physics*, 72(21):1–4, 2005.

[32] Q A Pankhurst, J Connolly, S K Jones, and J Dobson. Applications of magnetic nanoparticles in biomedicine. *Journal of Physics D: Applied Physics*, 36:167–181, 2003.

[33] S H Park, D H Hong, and T D Lee. Effect of MgO and on the microstructure and magnetic properties of CoCrPt- oxide perpendicular recording media Effect of MgO and Al 2 O 3 on the microstructure and magnetic properties of CoCrPt-oxide perpendicular recording media. *Journal of Applied Physics*, 97:10–106, 2005.

[34] S. N. Piramanayagam. Perpendicular recording media for hard disk drives. *Journal of Applied Physics*, 102(1):011301, 2007.

[35] S N Piramanayagam, C K Pock, L Lu, C Y Ong, J Z Shi, and C S Mah. Grain size reduction in perpendicular recording media with oxide-based intermediate layers. *Citation: Appl. Phys. Lett*, 89, 2006.

[36] Quantum Design. *Using PPMS Superconducting Magnets at Low Fields*, 2009.

[37] M. Respaud. Magnetization process of noninteracting ferromagnetic cobalt nanoparticles in the superparamagnetic regime: Deviation from Langevin law. *Journal of Applied Physics*, 86(1):556–561, 1999.

[38] Stephen B. Rice, Christopher Chan, Scott C. Brown, Peter Eschbach, Li Han, David S. Ensor, Aleksandr B. Stefaniak, John Bonevich, András E. Vladár, Angela R.Hight Walker, Jiwen Zheng, Catherine Starnes, Arnold Stromberg, Jia Ye, and Eric A. Grulke. Particle size distributions by transmission electron microscopy: An interlaboratory comparison case study. *Metrologia*, 50(6):663–678, dec 2013.

[39] Anup G Roy and David E Laughlin. Effect of seed layers in improving the crystallographic texture of CoCrPt perpendicular recording media Grain size reduction in perpendicular recording media with oxide-based intermediate layers

Effect of seed layers in improving the crystallographic text. *Journal of Applied Physics*, 911(79):8076–8078, 2002.

[40] Rujun Tang and Pin Ho. Growth of Ru-SiO2 underlayer for Co72Pt28 -SiO2 nanocomposite films. *Physica Status Solidi (a)*, 211(5):1162–1167, 2014.

[41] Charbel Tannous and R. Lawrence Comstock. Magnetic Information-Storage Materials. In *Springer Handbook of Electronic and Photonic Materials*, pages 1185–1223. Springer International Publishing, Cham, 2017.

[42] Kim Kong Tham, Ryosuke Kushibiki, and Tomonari Kamada. B 2 O 3 : Grain boundary material for high-K u CoPt-oxide granular media with low degree of intergranular exchange coupling Related content Effects of utilizing a granular nucleation layer on magnetic properties and microstructure of CoPt-B 2 O 3 granular. *Japanese Journal of Applied Physics*, 2016.

[43] Kim Kong Tham, Ryosuke Kushibiki, Tomonari Kamada, Shintaro Hinata, and Shin Saito. Reduction of Intergranular Exchange Coupling for CoPt-B2O3 Granular Media by Employing a RuCoCr-Oxide Buffer Layer With Oxide of Various Melting Points. *IEEE Transactions on Magnetics*, PP:1–4, 2018.

[44] Shan X. Wang and A. M. (Aleksandr Markovich) Taratorin. *Magnetic information storage technology*. Academic Press, San Diego, 1999.

[45] Roger Wood. Future hard disk drive systems. *Journal of Magnetism and Magnetic Materials*, 321(6):555–561, 2009.

[46] L Ye, B Qi, T Pearson, Y Cordeau, O T Mefford, and T M Crawford. Real time monitoring of superparamagnetic nanoparticle self-assembly on surfaces of magnetic recording media. *J. Appl. Phys*, 115:17–513, 2014.

[47] Longfei Ye, Tanner Pearson, Cory Dolbashian, Philip Pstrak, A. R. Mohtasebzadeh, Ben Fellows, O. Thompson Mefford, and Thomas M. Crawford. Magnetic-Field-Directed Self-Assembly of Programmable Mesoscale Shapes. *Advanced Functional Materials*, 26(22):3983–3989, 2016.

[48] Binu C. Yeldose and B. Ramamoorthy. Characterization of DC magnetron sputtered diamond-like carbon (DLC) nano coating. *International Journal of Advanced Manufacturing Technology*, 38(7-8):705–717, 2008.

[49] Hua Yuan and David E Laughlin. Microstructure and magnetic properties of perpendicular media with reduced grain size. *Journal of Applied Physics*, 105:7–707, 2009.

[50] Suxia Zhang, Nq Wang, Huijun Yu, Yaming Niu, and Changqing Sun. Covalent attachment of glucose oxidase to an Au electrode modified with gold nanoparticles for use as glucose biosensor. *Bioelectrochemistry*, 2005.

[51] Y. D. Zhang, J. I. Budnick, W. A. Hines, C. L. Chien, and J. Q. Xiao. Effect of magnetic field on the superparamagnetic relaxation in granular Co-Ag samples. *AIP Conference Proceedings*, 72(16):2053–2055, 1998.

Appendix A

Additional Media Layers and Samples

In this appendix, we will first describe additional components of magnetic recording media not described in chapters 2 and 3. Then we will discuss additional samples and some preliminary data from magnetic measurements. The latter is an aside to section 6.2, as this futher work is incomplete, and future researchers are encouraged to expand upon this information for further exploration of custom magnetic media for use with self assembly.

A.1 Nonmagnetic Interlayers and Seed Layers

The seed layer, directly above the SUL, is a polycrystalline layer designed to set the structure for the following interlayer(s). The seed layers and interlayers positioned between the SUL and recording layers serve to decouple the magnetic materials and to act as templates for the crystalline structure of the recording layer, which is critical for media performance. The size of the grains in the interlayer provide a guide for the grains of the recording layer [35]. The grain size in the interlayer(s) can also be tuned with the same techniques as for the recording layer, namely co-sputtering the interlayer material with an oxide material and in an oxygenated environment.

Common seed and interlayer materials include Ta, Ti, NiW, Ru, and Ru alloys [39, 40, 35]. The desired structure of the Ti and Ru-based interlayers is 0001, to promote c-axis out-of-plane in the Co grains of the recording layer, and Ta or other non-hcp materials can also help align the c-axis of the recording layer crystals out-of-plane by wetting the substrate before recording layer deposition. Several layers of

these materials in combination can also be utilized to further enhance desired growth properties in the granular media layer [49].

A.2 Substrate

Typical commercial media uses a glass or Al substrate [18]. We have employed both Si (111) to set the texture of the subsequent layers and soda lime glass. Glass substrates are preferable for samples that use an amorphous SUL as the first layer, as with CoFeB, which will be discussed at greater length in section A.4. However, a crystalline substrate like Si is preferable for an NiFe SUL, which requires that the magnetocrystalline easy axis be in-plane, as is the case for the (111) orientation. We have achieved coercivities on the order of 1 kOe with both combinations of substrate and SUL materials. The samples discussed in chapter 3 used an Si substrate, but the additional samples presented in section A.4 use predominantly glass substrates. There are several advantages to the glass substrate: not only are glass wafers much lower in price than single crystal Si, but the cleaning process is less intensive, since the subsequent layers do not need to grow epitaxially.

A.3 Protective Layers

Most commercially available perpendicular recording media is coated with diamond-like carbon (DLC), and then with some kind of bonded lubricant to facilitate smooth travel of the head during recording and reading [18]. Currently, the lubricant layer is removed before recording patterns, because it collects dust while outside of the hard drive case, which causes poor head motion in our stage recording setup.

However, based on cursory AFM data, the commercial heads used in this work (removed from HGST drives) can cause scratches up to 100s of nanometers deep in sputtered custom media without a protective layer. Thus coating with a hard material, such as DLC, is essential for testing and performance of the films. DLC can

be formed by RF magnetron sputtering (a type of sputtering like that described in section 3.1, but with an RF power supply) from a carbon target at room temperature in an Ar environment [29, 48].

A.4 Additional Samples and Design of Experiment

Using a factorial design of experiment, I have systematically varied the seed layer thickness and material; SUL thickness and material; interlayer thickness, material, and oxide content; recording layer thickness and oxide content; and the annealling time and temperature of custom media samples, to obtain over 150 thin film samples divided into 13 subsets. The order of sample deposition was randomized to avoid systematic effects to the magnetic properties, such as changes to the deposition rate as the target material is degraded after hours of use. The factorial design allows us to study the effect of changes to multiple factors and the interaction between those factors [27]. The primary effect studied to date has been recording layer coercivity, as measured by a VSM. Some of these results will be presented, but the full analysis of this experiment is incomplete as of the publication of this dissertation.

A.4.1 Soft Magnetic Underlayer Variations

Initially NiFe was used as the SUL, since it is more cost effective and easier to work with than other common SUL materials. However, NiFe requires a crystalline substrate and must have the correct orientation to be an effective SUL. CoFeB, in contrast has a much higher saturation moment and can be an amorpous layer, which is more easily achievable [22]. Unfortunately, CoFeB is slightly more difficult to sputter, since boron is nonmetallic. Because the custom media must be compatible with commercial heads in the current set-up, we transitioned to a CoFeB SUL for its larger saturation moment and easier structural requirements. An amorphous CoFeB SUL has better in-plane anisotropy than crystalline CoFeB, unlike the NiFe SUL

discussed in chapter 3, which has increased saturation magnetization with increasing crystalinity in the (111) orientation.

Sputtering the CoFeB SUL on glass and using low annealing temperatures avoids creating a magnetocrystalline easy axis to compete with the shape anisotropy of the thin film, which tries to keep the magnetization of the layer in-plane [22]. The in-plane anisotropy of the SUL allows for easier writing with a perpendicular recording head by guiding the magnetic flux to the return pole.

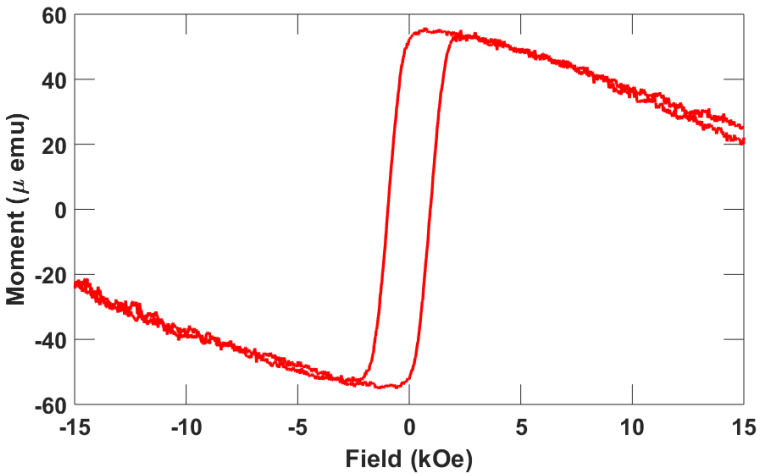

Figure A.1 Hysteresis loop of a sample with Ta/Ru/Ru/CoCrPt-B_2O_3 on a soda lime glass substrate.

The VSM data in Figs. A.1 and A.2 demonstrate that we have the capabilities to fabricate samples using an amorphous substrate (soda glass) which have coercivities comparable to the coercivities found in samples with our crystalline SUL nonmagnetic substitution (see section 3.4). Thus we are confident that an amorphous SUL, like CoFeB, will not hinder growth of the recording layer grains, provided the seed and interlayers between the SUL and recording layer are chosen to promote the 0001 crystal structure in the recording layer required for perpendicular recording.

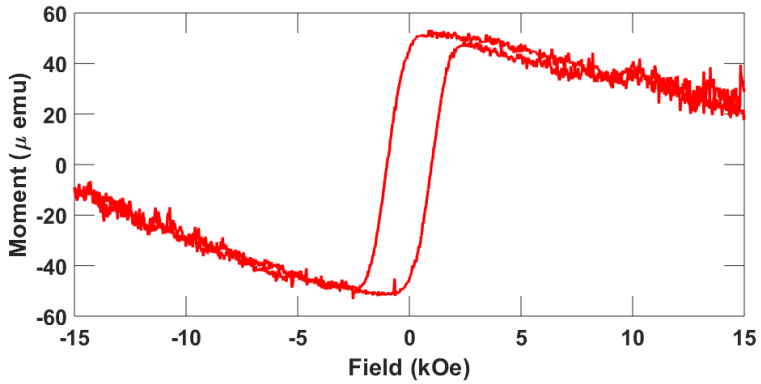

Figure A.2 Hysteresis loop of a sample with Ta/Ru/Ru-B_2O_3/CoCrPt-B_2O_3 on a soda-lime glass substrate.

A.4.2 Grain Boundaries and Size

Grain size and segregation have a large effect on the performance of magnetic media. Tuning these properties can be achieved with several techniques, including by altering the nonmagnetic interlayers and seed layer below the magnetic recording layer or by promoting the formation of an oxide boundary around the magnetic grains.

Deposited directly above the SUL are the seed and interlayers (Fig. 2.3). The seed layer sets the stage for the crystal structure of subsequent layers. Ru has been chosen as the first interlayer material, since its crystalline structure and lattice parameter have been shown to increase alignment of the recording layer out-of-plane [39, 40, 35]. Material candidates have been chosen such that their lattice properties promote c-axis out-of-plane in the first interlayer of Ru. Ta, Ni, W, Cu have cubic structure, which when grown in the 111 orientation, provide a guide for the 0001 structure of the first interlayer. The first interlayer of Ru is still smooth because of the low deposition pressure of less than 10 mTorr. The first Ru interlayer sets the hcp crystal structure for the next Ru layer and in turn, the CoCrPt recording layer. This structure, with the c-axis out-of-plane, increases the coercivity and remnant

magnetization of the recording layer [26]. A larger remnant magnetization yields a larger field from the magnetized grains, which will change the force experienced by NPs above the grains during self-assembly.

The topmost interlayer grains provide a template for the recording layer. Thus by reducing the grain size in the interlayer, we can promote larger or smaller grains in the recording layer. Proposed options for the interlayer material include Ru, and Ru alloys [40, 35].The second Ru layer is also co-sputtered with an oxide (either SiO_2 or B_2O_3) at an Ar pressure greater than 10 mTorr, which promotes the granular texture [42]. The amount of oxide can be varied to tune the Ru grain size, creating a template for the recording layer grains. The isolated grains of Ru promote isolated grains of the Co-alloy, which reduces intergranular exchange [40, 49]. Reduced intergranular exchange allows for easier grain switching and sharper transitions between recorded bits, and will yeild a higher coercivity of the recording layer [34]. Higher coercivity can increase the remnant magnetization, and thus the stray field of the cusom media. In addition to changing grain size, segregating the grains with an oxide boundary can change both their size and switching properties. We can adjust granular spacing in the interlayer by cosputtering with an oxide, such as B_2O_3, in turn we can tune grain size and segregation in the CoCrPt recording layer.

We can segregated and change the size of grains in the recording layer by co-sputtering with an oxide material or in an oxygenated environment. One effective oxide material for segregating grains in the CoCrPt recording layer of the media is SiO_2, which can be co-sputtered with the recording material [33]. Although we have not used this technique, grain segregation can be increased by reactive sputtering of the recording material and SiO_2 in an oxygen-rich environment [5, 19]. Even without an additional oxide material, reactive sputtering with O_2 during deposition of the recording layer can increase the segregation by making the Cr boundaries of the grains more robust [34], and I recommend to future researchers that they expand on

this work by sputtering the recording layer in an oxygenated environment to tune grain size and segregation.

A.4.3 Preliminary Results of Additional Samples

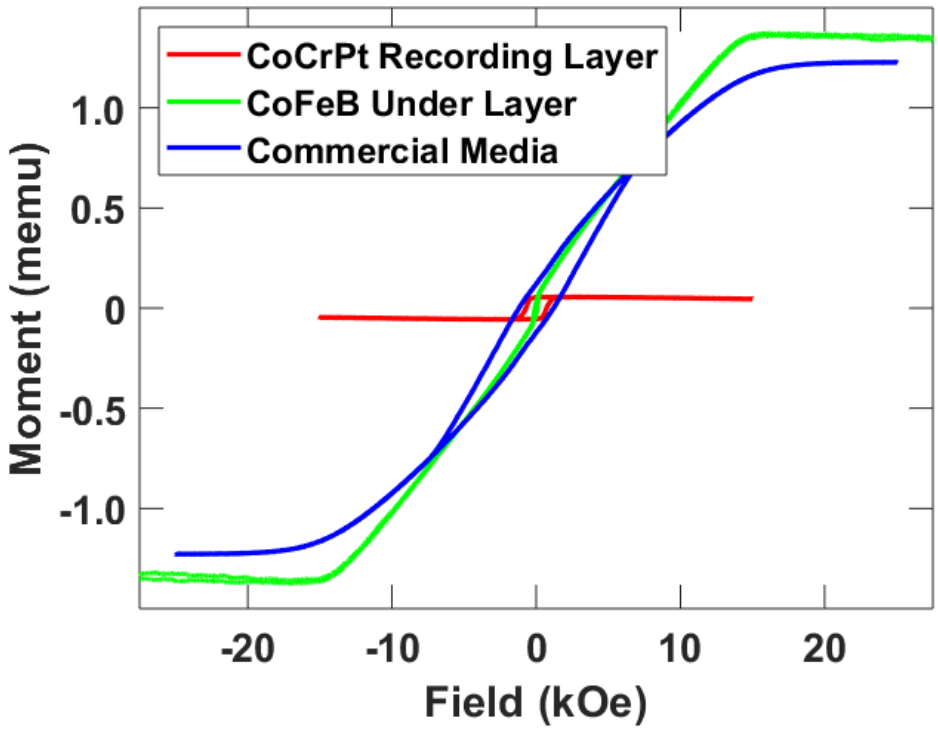

Figure A.3 Comparison of commerical PRM (blue) with a CoFeB SUL sample (green) and a CoCrPt recording layer-only sample (red). Note the similar saturation moments of the SUL-only and commercial samples.

As mentioned previously, we have made samples with nearly 1 kOe coercivity with an amorphous substrate and SUL (CoFeB) (Figs. A.1, A.2). In Fig. A.1, the sample is composed of Ta seed layer and two Ru interlayers on a soda lime glass substrate with a 10 nm CoCrPt recording layer co-sputtered with $\sim 10\%$ B_2O_3. After deposition of those layers, the sample was annealed at 100°C for 25 minutes. The

saturation moment is 68 μemu, coercivity is approximately 960 Oe, and remnant moment is 52 μemu, yielding a squareness of $m_r/m_s \approx 0.76$.

In Fig. A.2, the sample includes a Ta seed layer, an Ru interlayer deposited at a pressure below 10 mTorr, and an Ru interlayer deposited at a pressure above 10 mTorr and co-sputtered with approximately 10% B_2O_3 on a soda lime glass substrate. The recording layer is 10 nm of CoCrPt co-sputtered with $\sim$ 10% B_2O_3. The sample was annealed at 450°C for 5 minutes after deposition. The saturation moment is 55 μemu, coercivity is approximately 1 kOe, and remnant moment is 45 μemu, yielding a squareness of $m_r/m_s \approx 0.82$.

The samples in these subsets of our DOE have not been fully characterized, and doing so could yield interesting information about material choices and deposition conditions, which may influence the properties of custom magnetic media which are vital for self-assembly, namely coercivity, remnance, and squareness. The majority of samples remaining for future characterization utilize a CoFeB SUL with various seed and interlayers. Samples with these components individually show more similarity to commercial media (which is necessary to record the custom media with a commercial head). See Fig. A.3. However, a single sample with the combination of properties has not been realized. Future researchers are encouraged to use this section as a starting point for creating better tuned custom magnetic media for use in self-assembly.

Appendix B

Code for Data Analysis

B.1 Ugly ZFC/FC Analysis

```
% DESCRIPTION:
% Basic analysis of ZFC VSM data in '.dat' file format.
% Methods taken from 'Determination of the blocking ...
  temperature of magnetic nanoparticles:
% The good, the bad, and the ugly' Bruvera, et al. ...
  Journal of Applied
% Physics 2015. DOI: 10.1063/1.4935484
% This is roughly the 'ugly' method.

% NOTE TO USERS (SELECTING FILES):
% When selecting files, select only the "..._zfc.dat" ...
  files.
% This analysis requires only the ZFC data, so the ...
  "..._fc.dat" files are
% unnecessary. However, if the data is in a single ...
  file, the program will
% select only the necessary portion. You be prompted to ...
  input the
```

```
% number of header lines in your data files (typically ...
    30),
% not including the line with variable names.

% NOTE TO USERS (ORGANIZING YOUR DATA IN FOLDERS):
% You will need to change the initial filepath in the ...
    first line
% of code to whatever folder holds your data folders ...
    for analysis. For example, my
% data files are separated into individual folders for ...
    each sample, and
% then those folders are put into the ...
    C:\Users\Fitz\Documents\MATLAB\data\
% folder for analysis. This is the format assumed by ...
    the datFileLoader
% function. If all your data files are in a single ...
    folder, you will need to
% specify the parent folder above that single folder in ...
    the first line of code
% or you will need to update the datFileLoader function,
% found near the bottom of this code with all other ...
    function definitions.

% NOTE TO USERS (RESULTS):
% There are two values calculated for the Blocking ...
    Temperature.
```

```
% The first is found from the parabolic fit of the ...
    derivative
% near the inflection point. This value, even though it ...
    will
% have a larger error, is more accurate. The Blocking
% Temperature found from the maximum of the derivative
% will have a lower error, but it may be significantly ...
    different
% from the other T_B calculation. If they are ...
    drastically different
% I recommend increasing your smoothing width.
```

B.1.1 Importing data from file, preprocessing, and finding ramp rate of temperature during measurement.

```
cd C:\Users\Fitz\Documents\MATLAB\data\

numberOfHeaderLinesInput = ...
    str2double(convertCharsToStrings(inputdlg(['Input ...
    number of ' ...
     'header lines in the data file'],'Header Lines')));

selectedFileData = ...
    datFileLoader(numberOfHeaderLinesInput);

Moment = selectedFileData.Moment_emu_;
Temp = selectedFileData.Temperature_K_;
Field = selectedFileData.MagneticField_Oe_;
```

```
Time = selectedFileData.TimeStamp_sec_;

[FWTemp,FWMoment,FWTime] = ...
    selectFieldWarmedData(Moment,Temp,Field,Time);
% This function selects only the 'field warmed' data, ...
    which is typically
% referred to at the 'ZFC' curve.

indexAt280K = find((FWTemp > 280.0),1,'first');
indexAt270K = find((FWTemp > 270.0),1,'first');
tempRampRate = abs( (FWTemp(indexAt270K) - ...
    FWTemp(indexAt280K)) / ...
      (FWTime(indexAt270K) - FWTime(indexAt280K)) );
```

B.1.2 Plotting the full file data, m(T):

```
zfcfcFigure = figure('Name','ZFC/FC');
set(zfcfcFigure,'Visible','on')
axes('FontWeight','bold','FontSize',16,'TickLength',[0.03 ...
    0.035])
box on
hold on
plot(Temp,(10^-3)*Moment,'k--','LineWidth',2)
title('ZFC/FC')
xlabel('Temperature (K)')
ylabel('Moment (Am^2)')
hold off
```

B.1.3 Calculate the derivative of m with respect to T by linear fit near each point:

```
smoothingWidthInput = ...
    str2double(convertCharsToStrings(inputdlg(['Input an ...
    integer ' ...
     'for the smoothing width. (Recommended between 5 and ...
        25). ' ...
     'Larger values will smooth noise in the derivative ...
        plot, but will also increase ' ...
     'the error in final calculated Blocking ...
        Temperature.'], 'Smoothing Width')));

dMdTemp = localLinearFitDerivative(FWMoment,FWTemp, ...
smoothingWidthInput);
```

B.1.4 Finding TB from the maximum of the derivative (compare with the value found from the parabolic fit below)

```
[maxDeriv,maxDerivIndex] = max(dMdTemp);
TBfromMaxOfDerivative = FWTemp(maxDerivIndex);

TBuncertaintyFromDerivative = ...
    smoothingWidthInput*tempRampRate;
```

B.1.5 Plotting dm/dT and letting user select a range of points to fit with inverse parabola to find inflection point (TB)

```
dmdTempFig = figure('Name','dm/dT versus ...
    Temperature','Units','normalized', ...
     'OuterPosition',[0.1,0.1,0.8,0.8]);
```

```
set(dmdTempFig,'Visible','on')
axes('FontWeight','bold','FontSize',16,'TickLength',[0.03 ...
    0.035])
box on
hold on
plot(FWTemp,(10^-3)*dMdTemp,'.','LineWidth',2)
title('dm/dT')
xlabel('Temperature (K)')
ylabel('dm/dT (Am^2/K)')
ylim([(min(dMdTemp)*0.0015) inf])
axesValues = gca;
axesValues.Position(4) = 0.65;
annotation('textbox',[0.1, 0.9, 0.1, 0.1],'String', ...
    {'1. Click the low temperature/start point for ...
       parabolic fitting (to the left of the peak)', ...
    '2. Hit ENTER,','3. Click the higher temperature/end ...
       point (to the right of the peak)', ...
    '4. Hit ENTER ...
       again.'},'BackgroundColor','w','FitBoxToText','on')
hold off

% The following section allows the user to place ...
    datatips in the plot
% for fitting near TB.
datacursormode on
dcm_obj = datacursormode(dmdTempFig);
set(dcm_obj,'UpdateFcn',@myupdatefcn)
```

```
pause
info_struct = getCursorInfo(dcm_obj); % Exports the ...
   cursor info to workspace
if isfield(info_struct, 'Position')
  lowTempStartPointForFit = info_struct.Position;
end

% The following section allows the user to select the ...
   end point of the data
% to be used to find TB (should be at a higher ...
   temperature than previous
% point selected).
datacursormode on
dcm_obj = datacursormode(dmdTempFig);
set(dcm_obj,'UpdateFcn',@myupdatefcn)
pause
info_struct = getCursorInfo(dcm_obj); % Exports ...
   cursor info to workspace.
if isfield(info_struct, 'Position')
  highTempEndPointForFit = info_struct.Position;
end

dMdTempFitRangeFromUser = (FWTemp >= ...
   lowTempStartPointForFit(1)) & ...
    (FWTemp <= highTempEndPointForFit(1) & ...
       ~isnan(dMdTemp));
```

B.1.6 Fitting the selected points with a parabola and calculating TB

```
[parabolicFitdMdTemp,parabolicFitStats] = ...
    polyfit(FWTemp(dMdTempFitRangeFromUser), ...
     dMdTemp(dMdTempFitRangeFromUser),2);
[parabolaForPlot,parabolaErrorForPlot] = ...
    polyval(parabolicFitdMdTemp, ...
     FWTemp(dMdTempFitRangeFromUser),parabolicFitStats);
TBfromParabolicFit = -parabolicFitdMdTemp(2) / ...
    (2.0*parabolicFitdMdTemp(1));
% This calculated value of TB is the temperature (x) ...
    axis location of the peak of the
% fitted parabola.
```

B.1.7 Calculating the Error in TB (as found from the peak of the parabolic fit)

```
% NOTE TO USERS (ERROR ESTIMATION)
% The following error calculation is from both the ...
    uncertainty in
% TB introduced when we calculated dMdTemp, but also ...
    the error from the
% parabolic fit. To find the error in TB found from the ...
    parabolic fit,
% we need to use the struct 'parabolicFitStats', which ...
    contains fit error information.
% We can find the covariance matrix for the fit ...
    coefficients,
% which contains the variances of the fit coefficients.
```

```
% Those variances can then be used to find the error in ...
   TB with
% an error propogation calculation. (See the MATLAB ...
   documentation on the
% outputs of 'polyfit' for more about this.)

R = parabolicFitStats.R;
normOfResiduals = parabolicFitStats.normr;
degreesOfFreedom = parabolicFitStats.df;

covarianceMatrix = (inv(R) * inv(R)') * ...
   normOfResiduals^2 / degreesOfFreedom;

varianceOfFitCoefficients = diag(covarianceMatrix);

Coefficient1 = parabolicFitdMdTemp(1);
Coefficient2 = parabolicFitdMdTemp(2);

Coefficient1StandardError = ...
   sqrt(varianceOfFitCoefficients(1));
Coefficient2StandardError = ...
   sqrt(varianceOfFitCoefficients(2));

TBtotalError = Coefficient1StandardError * ...
   abs(Coefficient2 / (2*Coefficient1^2) ) + ...
    Coefficient2StandardError * abs(1.0 / ...
       (2*Coefficient1)) + TBuncertaintyFromDerivative;
```

B.1.8 Plotting the derivative, the parabolic fit (in the selected region), and TB from the parabolic fit.

```
dmdTempFig2 = figure('Name','dm/dT versus Temperature');
set(dmdTempFig2,'Visible','on')
axes('FontWeight','bold','FontSize',16,'TickLength',[0.03 ...
    0.035])
box on
hold on
plot(FWTemp,(10^-3)*dMdTemp,'.')
plot(FWTemp(dMdTempFitRangeFromUser), ...
(10^-3)*parabolaForPlot,'-r','LineWidth',2)
title('dm/dT')
xlabel('Temperature (K)')
ylabel('dm/dT (Am^2/K)')
ybottom = ylim;
axesValues = gca;
axesValues.Position(4) = 0.65;
TBline = [ybottom(1), ...
    polyval(parabolicFitdMdTemp,TBfromParabolicFit)];
plot([TBfromParabolicFit ...
    TBfromParabolicFit],(10^-3)*TBline,'r--','LineWidth',2)
legend('dm/dT','Fit','T_B from ...
    fit','Location','northeast')
TBstring = {strcat('T_B from the parabolic fit is: ...
    ',num2str(TBfromParabolicFit),'+/-', ...
     num2str(TBtotalError),'K'),strcat('T_B from the ...
        maximum of the derivative is: ', ...
```

```
    num2str(TBfromMaxOfDerivative),'+/-', ...
    num2str(TBuncertaintyFromDerivative),'K')};
annotation('textbox',[0.1,0.9,0.1,0.1],'String',TBstring)
hold off
```

B.1.9 Function definitions

```
function fileData = datFileLoader(numberOfHeaderLines)
    % numberOfHeaderLines is the input number of lines ...
      for the readtable
    % function to skip when importing data (it is ...
      typically 30 for this VSM
    % data) not including the line with the variable ...
      names.
    folderList = dir;
    folderNames = {folderList.name};
    [folderIndex,~] = listdlg('PromptString','Select ...
      Data Folder Name', ...
       'SelectionMode','single','ListString',folderNames);

    cd(folderNames{folderIndex});
    fileList = dir('*.dat');
    fileNames = {fileList.name};
    [selectedFile,~] = listdlg('PromptString','Choose ...
      .dat file to analyze', ...
       'SelectionMode','single','ListString',fileNames);
```

```
    fileData = ...
        readtable(fileList(round(selectedFile)).name, ...
          'HeaderLines',numberOfHeaderLines, ...
          'ReadVariableNames',true);
end

function [newTempData,newMomentData,newTimeData] = ...
    selectFieldWarmedData(M,T,H,timeData)
    % M is the input moment data
    % T is the input temperature data
    % H is the input applied field data
    % timeData is the timestamp data from the VSM

    maxT = max(T);
    maxH = max(H);

    indicesOfNonzeroField = H > maxH/2; % finds point ...
        at which field is turned on
    nonzeroFieldPoint = ...
        find(indicesOfNonzeroField,1,'first');

    newTempData = T(nonzeroFieldPoint:end); % includes ...
        only temperature data from
    % when field is applied
    newMomentData = M(nonzeroFieldPoint:end);
    newTimeData = timeData(nonzeroFieldPoint:end);
```

```
    % the following lines are in case the ZFC and FC ...
       data are in the same
    % file, this will remove the FC portion.
    indexOfMaxTemp = newTempData > (maxT - 1.0); % ...
       finds point where temperature
    % is 1 K less than the max
    highTempPoint = find(indexOfMaxTemp,1,'first');

    newTempData = newTempData(1:highTempPoint);
    newMomentData = newMomentData(1:highTempPoint);
    newTimeData = newTimeData(1:highTempPoint);
end

function momentDerivative = ...
   localLinearFitDerivative(M,T,smoothingWidth)
    % M is the input moment data
    % T is input temperature data
    % smoothingWidth determines number of points on ...
       either side of data point
    % to fit when finding local derivative
    momentDerivative = ones(numel(T),1);
    for k = 1:numel(T)
        if k<(smoothingWidth+1)    % case if data is ...
           at beginning of T,
            % prevents negative indices
            a = k;
            b = k+2*smoothingWidth;
```

```
        elseif k>(numel(T)-(smoothingWidth+1)) % case ...
           if data is at end of T,
            % prevents indices larger than numel(T)
            a = k-2*smoothingWidth;
            b = k;
        else    % otherwise the data to fit is ...
           smoothingWidth pts to either
            % side of point in question
            a = k-smoothingWidth;
            b = k+smoothingWidth;
        end
        localFit = polyfit(T(a:b),M(a:b),1);
        momentDerivative(k) = localFit(1);
    end
end

function output_txt = myupdatefcn(~,event_obj)
  % ~            Currently not used (empty)
  % event_obj    Object containing event data structure
  % output_txt   Data cursor text
  pos = get(event_obj, 'Position');
  output_txt = {['x: ' num2str(pos(1))], ['y: ' ...
     num2str(pos(2))]};
end
```

B.2 "Better" Analysis Method of ZFC/FC Data

```
% DESCRIPTION:
```

```
% analysis of ZFC/FC curve data, method from:
% 'Determination of the blocking temperature of ...
  magnetic nanoparticles:
% The good, the bad, and the ugly' I. J. Bruvera, et al.
% http://dx.doi.org/10.1063/1.4935484
% This is an abridged version of the "good" method.

% NOTE TO USERS (ORGANIZING YOUR DATA IN FOLDERS):
% You will need to change the initial filepath in both ...
  lines
% immediately preceeding the 'datFileLoader' function call
% to whatever folder holds your data folders for ...
  analysis. For example, my
% data files are separated into individual folders for ...
  each sample, and
% then those folders are put into the ...
  C:\Users\Fitz\Documents\MATLAB\data\
% folder for analysis. This is the format assumed by ...
  the datFileLoader
% function. If all your data files are in a single ...
  folder, you will need to
% specify the parent folder above that single folder in ...
  the first line of code
% or you will need to update the datFileLoader function,
% found near the bottom of this code with all other ...
  function definitions.
```

```
% NOTE TO USERS (FILE SELECTION):
% The first file you'll be asked to select
% is the ZFC portion, the second is the FC portion of ...
   the data.
```

B.2.1 Loading ZFC measurement data and remove initial cooling data:

```
cd C:\Users\Fitz\Documents\MATLAB\data\

numberOfHeaderLinesInput = ...
   str2double(convertCharsToStrings(...
inputdlg(['Input number of ' ...
    'header lines in the data file'],'Header Lines')));

selectedFileData = ...
   datFileLoader(numberOfHeaderLinesInput);

Mzfc = selectedFileData.Moment_emu_;
Tzfc = selectedFileData.Temperature_K_;
Hzfc = selectedFileData.MagneticField_Oe_;
Timezfc = selectedFileData.TimeStamp_sec_;

[Tzfc,Mzfc,Timezfc] = ...
   selectFieldWarmedData(Mzfc,Tzfc,Hzfc,Timezfc);

indexAt280K = find((Tzfc > 280.0),1,'first');
indexAt270K = find((Tzfc > 270.0),1,'first');
```

```
tempRampRate = abs( (Tzfc(indexAt270K) - ...
    Tzfc(indexAt280K)) / ...
     (Timezfc(indexAt270K) - Timezfc(indexAt280K)) );
```

B.2.2 Importing the FC measurement data:

```
cd C:\Users\Fitz\Documents\MATLAB\data\
selectedFileData = ...
    datFileLoader(numberOfHeaderLinesInput);

Mfc = selectedFileData.Moment_emu_;
Tfc = selectedFileData.Temperature_K_;
%Hfc = selectedFileData.MagneticField_Oe_;
%Timefc = selectedFileData.TimeStamp_sec_;

[~, minTpt] = min( abs(min(Tzfc) - Tfc));

Mfc = Mfc(1:minTpt); % removing the final portion of ...
    the data
Tfc = Tfc(1:minTpt); % Which is the warming period ...
    after both
%Hfc = Hfc(1:minTpt); % the ZFC and FC routines are ...
    finished
```

B.2.3 Convert to volume magnetization and subtract Mfc from Mzfc

```
properties = convertCharsToStrings(inputdlg(...
{'Input mass of sample in grams', ...
     'Density in grams/cc'},'Mass and Density',2));
```

```
volume = ...
    str2double(properties(1))/str2double(properties(2));

Mzfc = Mzfc/volume*1000.0;
Mfc = Mfc/volume*1000.0;

a = numel(Mzfc);
%b = numel(Mfc);

subM = zeros(a,1);
for i = 1:a
    [~, ptIdx] = min(abs(Tzfc(i) - Tfc));
    subM(i) = Mzfc(i) - Mfc(ptIdx);
    Temp = Tzfc;
end

[subM,Temp,~] = cleanData(subM,Temp,zeros(numel(Temp),1));
```

B.2.4 Plotting Mzfc, Mfc, and Difference:

```
figure("Name","ZFC/FC and Difference Plot")
axes('FontWeight','bold','FontSize',16,'TickLength',[0.03 ...
    0.035])
box on
hold on
plot(Tzfc,Mzfc,'r','LineWidth',2)
plot(Tfc,Mfc,'b','LineWidth',2)
plot(Temp,subM,'k-','LineWidth',2)
```

```
legend('ZFC','FC','ZFC - FC','Location','east')
title('ZFC, FC, and Difference')
xlabel('Temperature (K)')
ylabel('Magnetization (A/m)')
hold off
```

B.2.5 Calculating the derivative of the subtracted magnetization:

```
smoothingWidthInput = ...
    str2double(convertCharsToStrings(inputdlg(['Input an ...
    integer ' ...
     'for the smoothing width. (Recommended between 5 and ...
        25). ' ...
     'Larger values will smooth noise in the derivative ...
        plot, but will also increase ' ...
     'the error in final calculated Blocking ...
        Temperature.'], 'Smoothing Width')));


dSubM = ...
    localLinearFitDerivative(subM,Temp,smoothingWidthInput);


[~,maxDerivIndex] = max(dSubM);
TBMode = Temp(maxDerivIndex);


TBuncertainty = smoothingWidthInput*tempRampRate;
```

B.2.6 Plotting the derivative:

```
plotpts = Temp<300.0;
dSubMplot = figure("Name","d(m_zfc - m_fc)/dT");
```

```
axes('FontWeight','bold','FontSize',16,'TickLength',[0.03 ...
    0.035])
box on
hold on
plot(Temp(plotpts),dSubM(plotpts))
%title('Good Method for finding TB')
xlabel('Temperature (K)')
ylabel('d(ZFC - FC)/dT (A/m/K)')
ybottom = ylim;
axesValues = gca;
axesValues.Position(4) = 0.65;
TBline = [ybottom(1), dSubM(maxDerivIndex)];
plot([TBMode TBMode],(10^-3)*TBline,'r--','LineWidth',2)
legend('d(Mzfc - Mfc)/dT','TBmode','Location','northeast')
TBstring = {strcat('T_B from the maximum of the ...
    derivative is: ', ...
     num2str(TBMode),'+/-',num2str(TBuncertainty),'K')};
annotation('textbox',[0.2,0.9,0.1,0.1],'String',TBstring)
hold off
```

B.2.7 Function definitions:

```
function fileData = datFileLoader(numberOfHeaderLines)
    % numberOfHeaderLines is the input number of lines ...
      for the readtable
    % function to skip when importing data (it is ...
      typically 30 for this VSM
```

```
    % data) not including the line with the variable ...
       names.
    folderList = dir;
    folderNames = {folderList.name};
    [folderIndex,~] = listdlg('PromptString','Select ...
       Data Folder Name', ...
        'SelectionMode','single','ListString',folderNames);

    cd(folderNames{folderIndex});
    fileList = dir('*.dat');
    fileNames = {fileList.name};
    [selectedFile,~] = listdlg('PromptString','Choose ...
       .dat file to analyze', ...
        'SelectionMode','single','ListString',fileNames);

    fileData = ...
       readtable(fileList(round(selectedFile)).name, ...
           'HeaderLines',numberOfHeaderLines, ...
           'ReadVariableNames',true);
end

function [newTempData,newMomentData,newTimeData] = ...
   selectFieldWarmedData(M,T,H,timeData)
    % M is the input moment data
    % T is the input temperature data
    % H is the input applied field data
    % timeData is the timestamp data from the VSM
```

```
    maxT = max(T);
    maxH = max(H);

    indicesOfNonzeroField = H > maxH/2; % finds point ...
        at which field is turned on
    nonzeroFieldPoint = ...
        find(indicesOfNonzeroField,1,'first');

    newTempData = T(nonzeroFieldPoint:end); % includes ...
        only temperature data
    % from when field is applied
    newMomentData = M(nonzeroFieldPoint:end);
    newTimeData = timeData(nonzeroFieldPoint:end);

    % the following lines are in case the ZFC and FC ...
        data are in the same
    % file, this will remove the FC portion.
    indexOfMaxTemp = newTempData > (maxT - 1.0); % ...
        finds point where temperature
    % is 1 K less than the max
    highTempPoint = find(indexOfMaxTemp,1,'first');

    newTempData = newTempData(1:highTempPoint);
    newMomentData = newMomentData(1:highTempPoint);
    newTimeData = newTimeData(1:highTempPoint);
end
```

```
function [AClean,BClean,CClean] = ...
    cleanData(AData,BData,CData)
     % removes missing/NaN values by row for all ...
        relevant data
     % most helpful for touchdown centering points ...
        (moment will be
     % NaN, so this removes the whole row of data for ...
        that point
     cleanTable = rmmissing(table(AData,BData,CData));
     AClean = cleanTable.AData;
     BClean = cleanTable.BData;
     CClean = cleanTable.CData;
end

function momentDerivative = ...
    localLinearFitDerivative(M,T,smoothingWidth)
     % M is the input moment data
     % T is input temperature data
     % smoothingWidth determines number of points on ...
        either side of data point
     % to fit when finding local derivative
     momentDerivative = ones(numel(T),1);
     for k = 1:numel(T)
          if k<(smoothingWidth+1)    % case if data is ...
             at beginning of T,
              % prevents negative indices
```

```
            a = k;
            b = k+2*smoothingWidth;
        elseif k>(numel(T)-(smoothingWidth+1)) % case ...
           if data is at end of T,
            % prevents indices larger than numel(T)
            a = k-2*smoothingWidth;
            b = k;
        else    % otherwise the data to fit is ...
           smoothingWidth pts
            % to either side of point in question
            a = k-smoothingWidth;
            b = k+smoothingWidth;
        end
        localFit = polyfit(T(a:b),M(a:b),1);
        momentDerivative(k) = localFit(1);
    end
end
```

B.3 Loading Program for ZFC/FC Data

```
% This program loads VSM data from a ZFC/FC measurement ...
  and exports only
% the relevant columns of data (moment, moment error, ...
  magnetic field, and
% temperature) into a new file. This program also ...
  allows averaging over
% temperature ranges to allow for easier fitting with ...
  other software
```

```
% (e.g. Mathematica).
% NOTE: output files are csv format
```

B.3.1 Importing data:

```
% ***** This section imports .dat files from VSM ...
    measurement *****
cd C:\Users\Fitz\Documents\MATLAB\data\
listopt = dir; % lists folders and files in current ...
    directory
lo = {listopt.name}; % holds names of folders and ...
    files in current directory
[fld,tf] = listdlg('PromptString','Select Data Folder ...
    Name','SelectionMode', ...
     'single','ListString',lo);
% fld is the index of user selection in lo character ...
    vector,
% tf is 0 if user doesn't select/x's out/hits cancel, 1 ...
    otherwise

cd(lo{fld}); % changes directory to user selected folder
fileInfo = dir('*.dat'); % lists .dat files in current ...
    directory
fi = {fileInfo.name}; % character vector - holds names ...
    of files in current directory
[datFile,tf2] = listdlg('PromptString','Choose .dat ...
    file to analyze', ...
     'SelectionMode','single','ListString',fi);
```

```
% user dialog to select .dat file (datFile is the index ...
   of selection in fi charachter vector)
% tf2 is 0 is user doesn't select/x's out/hits cancel, ...
   1 otherwise

n = round(datFile);

fileData = readtable(fileInfo(n).name,'HeaderLines', ...
30,'ReadVariableNames',true);
% reads data from user-selected file and makes table of ...
   values with
% variable names from header line
% note: the header is different for the different types ...
   of VSM data,
% these have 30 header lines
% and the 31st is the variable names
M = fileData.Moment_emu_; % full moment data
Merr = fileData.M_Std_Err__emu_; % moment error
T = fileData.Temperature_K_; % full tempurature data
H = fileData.MagneticField_Oe_; % field data

cleanFileData = rmmissing(table(M,Merr,T,H));
M = cleanFileData.M;
Merr = cleanFileData.Merr;
T = cleanFileData.T;
H = cleanFileData.H;
```

```
isFC = questdlg('Is this FC data?');

initialCoolingCut = questdlg('Would you like to remove ...
   initial cooling data?');

switch initialCoolingCut
    case 'Yes'

        switch isFC
            case 'Yes'
                [~,startPt] = min(T);
                T = T(startPt:end);
                M = M(startPt:end);
                Merr = Merr(startPt:end);
                H = H(startPt:end);
            case 'No'
                idxNZH = H > max(H)/2.0;
                startPt = find(idxNZH,1,'first'); % ...
                   finds point at which field is first ...
                   non-zero

                deltaMinitial = M(startPt) - M(startPt-1);

                T = T(startPt:end); % includes only ...
                   temperature data from when field is ...
                   applied
```

```
                M = M(startPt:end); % includes only ...
                    moment data for when field is applied
                Merr = Merr(startPt:end);
                H = H(startPt:end); % removes the ...
                    points where H is essentially 0
        end
end
```

B.3.2 Removing the background contribution:

```
% the next section allows the user to select a ...
   background file to remove
% (e.g. the signal from only the sample holder and ...
   polymer) if it has been
% measured separately

background = questdlg("Would you like to remove ...
   background signal?");

switch background
    case "Yes" % executes if the user selects 'yes' in ...
       background removal dialog box
        cd C:\Users\Fitz\Documents\MATLAB\data\
        listopt = dir; % lists folders and files in ...
           current directory
        lo = {listopt.name}; % holds names of folders ...
           and files in current directory
```

```
[fld,tf] = listdlg('PromptString','Select Data ...
   Folder Name', ...
    'SelectionMode','single','ListString',lo);
% fld is the index of user selection in lo ...
   character vector,
% tf is 0 if user doesn't select/x's out/hits ...
   cancel, 1 otherwise

cd(lo{fld}); % changes directory to user ...
   selected folder
fileInfo = dir('*.dat'); % lists .dat files in ...
   current directory
fi = {fileInfo.name}; % character vector - ...
   holds names of files in current directory
[datFile,tf2] = listdlg('PromptString','Choose ...
   .dat file to analyze', ...
    'SelectionMode','single','ListString',fi);
% user dialog to select .dat file (datFile is ...
   the index of selection in fi charachter vector)
% tf2 is 0 is user doesn't select/x's out/hits ...
   cancel, 1 otherwise

n = round(datFile);

fileData = readtable(fileInfo(n).name, ...
'HeaderLines',30,'ReadVariableNames',true);
```

```
% reads data from user-selected file and makes ...
   table of values with
% variable names from header line
% note: the header is different for the ...
   different types of VSM data,
% these have 30 header lines
% and the 31st is the variable names
bgM = fileData.Moment_emu_; % background ...
   moment data
bgMerr = fileData.M_Std_Err__emu_; % ...
   background moment error
bgT = fileData.Temperature_K_; % background ...
   tempurature data
bgH = fileData.MagneticField_Oe_; % ...
   background data

cleanBackgroundFileData = ...
   rmmissing(table(bgM,bgMerr,bgT,bgH));
bgM = cleanBackgroundFileData.bgM;
bgMerr = cleanBackgroundFileData.bgMerr;
bgT = cleanBackgroundFileData.bgT;
bgH = cleanBackgroundFileData.bgH;

switch initialCoolingCut
    case "Yes"
        switch isFC
            case "Yes"
```

```
            [~,startPt] = min(bgT);
            bgT = bgT(startPt:end);
            bgM = bgM(startPt:end);
            bgMerr = bgMerr(startPt:end);
            bgH = bgH(startPt:end);
        case "No"
            idxNZH = bgH > max(bgH)/2.0;
            startPt = ...
                find(idxNZH,1,'first'); % ...
                finds point at
            % which field is first non-zero

            %deltaMinitial = bgM(startPt) -...
                bgM(startPt-1);

            bgT = bgT(startPt:end); % ...
                includes only temperature
            % data from when field is applied
            bgM = bgM(startPt:end); % ...
                includes only moment data
            % for when field is applied
            bgMerr = bgMerr(startPt:end);
            bgH = bgH(startPt:end); % ...
                removes the points where H is ...
                essentially 0
    end
end
```

```
j = numel(bgM);
k = numel(M);

for i = 1:k
    [~, bgpt] = min(abs(T(i) - bgT)); % finds ...
       the nearest temp value
    % in the background temp data to the ith ...
       temp value in the
    % full sample temp data, diff is how close ...
       the values are,
    % bgpt is the index in bgT

    % the next section fits the background data ...
       for the nearest points to bgpt
    if bgpt == 1 % case for first index
        [bgFit, bgFitStats] = polyfit(bgT(bgpt ...
           : (bgpt + 1)), ...
            bgM (bgpt : (bgpt + 1)), 1);
    elseif bgpt == j % case for last index
        [bgFit, bgFitStats] = polyfit(bgT((bgpt ...
           - 1) : bgpt), ...
            bgM ((bgpt - 1) : bgpt), 1);
    else
        [bgFit, bgFitStats] = polyfit(bgT((bgpt ...
           - 1) : (bgpt + ...
            1)), bgM ((bgpt - 1):(bgpt + 1)), 1);
```

```
            end

            M(i) = M(i) - (bgFit(1) * T(i) + ...
                bgFit(2)); % subtracts the background ...
                from the moment
            % bgFit(1) is the slope, bgFit(2) is the ...
                intercept
        end
    end
```

B.3.3 Averaging over temperature ranges to reduce points for fitting:

```
    answerReducePoints = questdlg(['Would you like to ...
        reduce the ' ...
        'number of data points?'])

    switch answerReducePoints
        case "Yes"
            pointSpread = str2double(convertCharsToStrings( ...
                ...
                inputdlg(['How large (in K) would you like ...
                    the ' ...
                'temperature steps to be?'], ...
                'Temperature Point Separation')));
            tempPts = ceil(min(T)):pointSpread:floor(max(T));
            newM = zeros(numel(tempPts),1);
            newMerr = zeros(numel(tempPts),1);
            newT = zeros(numel(tempPts),1);
```

```
newH = zeros(numel(tempPts),1);
for a=1:numel(tempPts)
    Trange = T( T>=(tempPts(a)-0.5) & ...
       T<=(tempPts(a)+0.5) );
    if isempty(Trange)
        newM(a) = 'NaN';
        newMerr(a) = 'NaN';
    else
        Mrange = M( T>=(tempPts(a)-0.5) & ...
           T<=(tempPts(a)+0.5) );
        MerrRange = Merr( T>=(tempPts(a)-0.5) ...
            & ...
             T<=(tempPts(a)+0.5) );
        [fitForReducePts,fitStrucForReducePts] =...
             polyfit( ...
        Trange,Mrange,1);
        [newM(a),fitError] = ...
            polyval(fitForReducePts, ...
             tempPts(a), fitStrucForReducePts);
        nearestIndices = dsearchn([Mrange ...
            Trange], ...
             [newM(a) tempPts(a)]);
        newMerr(a) = ...
            MerrRange(nearestIndices(1)) + ...
            fitError;
    end
```

```
            HRange = H( T>=(tempPts(a)-0.5) & ...
                T<=(tempPts(a)+0.5) );
            newH(a) = sum(HRange)/numel(HRange);
        end
        M = newM;
        Merr = newMerr;
        H = newH;
        T = tempPts';
end
```

B.3.4 Output data to CSV file:

```
names = ["Moment", "MomentError", "Temperature", "Field"];

tab = table(M, Merr, T, H, 'VariableNames', names);

fileInput = inputdlg("What would you like to name the ...
    output file?");
filename = strcat(fileInput, '.csv'); % adds file ...
    extension
% the above is still a cell, so indexing below calls ...
    the character
% vector in the cell

cd C:\Users\Fitz\Documents\MATLAB\data\CSVs\
writetable(tab, filename{1,1})

clear fileData cleanFileData cleanBackgroundFileData
```

B.4 Calculating Interaction Temperature from VSM Data

```
% NOTE TO USERS (DATA FILES):
% Check your data files for the number of header lines.
% If you have multiple files to concatenate,
% select the high temperature data first,
% assuming your data goes from high temp to low.


% NOTE TO USERS (SELECTING INVERSE SUSCEPTIBILITY DATA ...
  FOR FIT):
% The data tips for the fit of 1/chi vs T should be placed
% in the paramagnetic region of the plot, which is a ...
  linear
% region with positive slope. Should be on the right in ...
  the plot
% or possibly the whole plot. This slope and region are ...
  variable
% depending on the sample properties, so it may look ...
  simply flat
% compared to large negative slope on the left. If the ...
  sample
% blocking temp is too high, then there will be no ...
  visible paramagnetic
% region in VSM data, because the highest available ...
  temp is 400 K
% with the VSM currently.
```

B.4.1 Importing Data:

```
cd C:\Users\Fitz\Documents\MATLAB\data\

numberOfHeaderLinesInput = ...
    str2double(convertCharsToStrings(inputdlg(['Input ...
    number of ' ...
     'header lines in the data file'],'Header Lines')));

selectedFileData = ...
    datFileLoader(numberOfHeaderLinesInput);

properties = convertCharsToStrings(inputdlg({'Input ...
    mass of sample in grams', ...
     'Density in grams/cc'},'Mass and Density',2));
volume = ...
    str2double(properties(1))/str2double(properties(2));

M = selectedFileData.Moment_emu_/volume;
Temp = selectedFileData.Temperature_K_;
Field = selectedFileData.MagneticField_Oe_;

[M,Temp,Field] = cleanData(M,Temp,Field);

clear selectedFileData

answeraddfile = questdlg(['Do you want to select ...
    another mvH small field measurement' ...
```

```
    ' file to concatenate with the previous file?']);
answerAddFCfile = questdlg(['Do you want to add an FC ...
    file to add DC susceptibility data' ...
    ' to the calculation of Chi?'])
```

B.4.2 Adding additional data file (optional):

```
switch answeraddfile
    case 'Yes' % This executes if user decides to add ...
        additional data
        % (separate VSM file, will concatenate ...
            vertically with the second file on bottom)

        cd C:\Users\Fitz\Documents\MATLAB\data\

        additionalFileData = ...
            datFileLoader(numberOfHeaderLinesInput);

        M2 = additionalFileData.Moment_emu_/volume;
        Temp2 = additionalFileData.Temperature_K_;
        Field2 = additionalFileData.MagneticField_Oe_;

        [M2,Temp2,Field2] = cleanData(M2,Temp2,Field2);

        M = vertcat(M, M2);
        Temp = vertcat(Temp, Temp2);
        Field = vertcat(Field, Field2);
```

```
        clear M2 Field2 Temp2
end
demagN = ...
   str2double(convertCharsToStrings(inputdlg(['Input the ...
   demagnetization ' ...
    'factor (positive value, SI units) for your sample. ...
       If you plan to ignore' ...
    ' demagnetization effects, put 0.'])));
Field = Field - 4.0*pi*demagN*M;

tempSpace = ...
   str2double(convertCharsToStrings(inputdlg(['Input the ...
   ' ...
    'smallest step in temperature between consecutive ...
       moment versus ' ...
    'field measurements: '],'Temperature Spacing')));

cutPts = findSteps(tempSpace,Temp);

fieldValueInput = ...
   str2double(convertCharsToStrings(inputdlg(['Input a ...
   field value' ...
    ' slightly less than your maximum field. This will ...
       set the range of moment' ...
    ' versus field data to fit for slope.'],'Field Range ...
       Selection')));
```

```
invChi = zeros(numel(cutPts),1);
chi = zeros(numel(cutPts),1);
avgT = zeros(numel(cutPts),1);
chiErr = zeros(numel(cutPts),1);
invChiErr = zeros(numel(cutPts),1);
standardErrT = zeros(numel(cutPts),1);
colorvect = hsv(numel(cutPts));
mark = {'.','^','o','x','+','pentagram','*','square', ...
    ...
        '>','hexagram','diamond','>'};
legName = strings(1);

figure('Name','Moment versus Field at Various Temperature')
hold on
axes('FontWeight','bold','FontSize',16,'TickLength',[0.03 ...
    0.035],'NextPlot','add')
box on

for p = 1:(numel(cutPts)+1)
    if p == 1
        cutField = Field(1:cutPts(1));
        cutM = M(1:cutPts(1));
        cutTemp = Temp(1:cutPts(1));
    elseif p == (numel(cutPts)+1)
        cutField = Field(cutPts(p-1):end);
        cutM = M(cutPts(p-1):end);
        cutTemp = Temp(cutPts(p-1):end);
```

```
else
    cutField = Field( (cutPts(p-1)+1) : cutPts(p) );
    cutM = M( (cutPts(p-1)+1) : cutPts(p) );
    cutTemp = Temp( (cutPts(p-1)+1) : cutPts(p) );
end

[cutM,cutTemp,cutField] = ...
   cutByField(cutM,cutTemp,cutField,fieldValueInput);

[mHFitPoly,mHFitStats] = polyfit(cutField,cutM,1);
[mHPolydata,mHErr] = ...
   polyval(mHFitPoly,cutField,mHFitStats);

testp = mod(p,3); % selection to plot only certain ...
   lines/data for use with marker type,
% otherwise not enough options

chi(p) = mHFitPoly(1); % chi is the slope of the ...
   mvH data fit line, units: m^3
% (for m in A*m^2 and field in A/m)
invChi(p) = 1.0/mHFitPoly(1); % inverse chi is the ...
   reciprocal of the slope

degreesOfFreedom = numel(cutField); % number of ...
   degrees of freedom,
% i.e. number of data points in fit
```

```
sigmaM = cutM - mHPolydata; % difference between ...
   the acutal moment values
% and y-values of the fit
avgField = sum(cutField)/degreesOfFreedom; % ...
   average field for the
% selected data range, should be near zero

sigmaM2 = zeros(degreesOfFreedom,1);
sigmaField2 = zeros(degreesOfFreedom,1);
for idx = 1:numel(cutField)
    sigmaM2(idx) = sigmaM(idx)^2; % square of the ...
       difference
    % between actual moment and y-vals of fit
    sigmaField2(idx) = (cutField(idx) - ...
       avgField)^2; % square of
    % difference between field vals and avg field
end

chiErr(p) = sqrt( sum(sigmaM2) / ((degreesOfFreedom ...
   - 2) * (sum(sigmaField2) )) );

invChiErr(p) = chiErr(p) / ( chi(p) )^2; % error ...
   propogation from chi to 1/chi

avgT(p) = mean(cutTemp);

for indx = 1:numel(cutTemp)
```

```
        standardErrT(p) = sqrt( sum( (cutTemp(indx) - ...
            avgT(p))^2 ) / ...
              (degreesOfFreedom - 1) ...
                 )/sqrt(degreesOfFreedom); % standar ...
                 error of temperature average
    end


     if testp == 1
         plot(cutField,cutM,mark{round(p/2.0)}, ...
                        'MarkerEdgeColor', ...
                        colorvect(p,:))
         plot(cutField,mHPolydata,'Color',colorvect(p,:), ...
            ...
                        'HandleVisibility','off')
         % 'HandleVisibility' prevents curves from ...
            being labeled in legend
         plot(cutField,mHFitdata,'Color',colorvect(p,:), ...
            ...
                        'LineWidth',2)
         legName = ...
            strcat(legName,num2str(round(avgT(p))),' K');
     end
end
ylabel('Moment (A/m)')
xlabel('Magnetic Field (Oe)')
ylim([min(M) max(M)])
xlim([-40.0 40.0])
```

```
legend(legName)
hold off
```

B.4.3 Removing the Background Signal (from Polymer and Sample Holder):

```
removeBackground = questdlg('Do you want to select a ...
    file for background signal to remove?');

switch removeBackground
    case 'Yes' % This executes if user decides to ...
        remove background signal from data (separate VSM ...
        file)

        cd C:\Users\Fitz\Documents\MATLAB\data\

        backgroundFileData = ...
            datFileLoader(numberOfHeaderLinesInput);

        polymerProperties = ...
            convertCharsToStrings(inputdlg({'Input ...
            polymer mass for this sample (g)','Polymer ...
            density (g/cc)'},'Polymer Mass and Density'));
        polymerVol = str2double(polymerProperties(1)) ...
                /str2double(polymerProperties(2));
        % calculates volume of polymer in the ...
            particular sample from
        % mass/density
```

```
bgM = ...
    backgroundFileData.Moment_emu_/polymerVol; ...
    % converts to volume magnetization
bgTemp = backgroundFileData.Temperature_K_;
bgField = backgroundFileData.MagneticField_Oe_;

clear backgroundFileData

[bgM,bgTemp,bgField] = ...
    cleanData(bgM,bgTemp,bgField);

bgcutPts = findSteps(tempSpace,bgTemp);

invbgChi = zeros(numel(bgcutPts),1); % ...
    allocates space for chi values for background
bgchi = zeros(numel(bgcutPts),1);
bgavgT = zeros(numel(bgcutPts),1);

for g = 1:numel(bgcutPts)
    if g == 1       % selects field, temp, and ...
        moment data to use for fitting
        bgcutField = bgField(1:bgcutPts(1));
        bgcutM = bgM(1:bgcutPts(1));
        bgcutTemp = bgTemp(1:bgcutPts(1));
    else
```

```
    bgcutField = bgField( ...
        (bgcutPts(g-1)+1) : bgcutPts(g) );
    bgcutM = bgM( (bgcutPts(g-1)+1) : ...
        bgcutPts(g) );
    bgcutTemp = bgTemp( (bgcutPts(g-1)+1) ...
        : bgcutPts(g) );
end

bglowField = bgcutField < -30.0; % sets ...
    field cutoff values for fitting/finding ...
    slope
bghighField = bgcutField > 30.0;

bgstartField = find(bghighField,1,'last');
bgendField = find(bglowField,1,'last');

bgcutField = ...
    bgcutField(bgendField:bgstartField);
bgcutM = bgcutM(bgendField:bgstartField);
bgcutTemp = ...
    bgcutTemp(bgendField:bgstartField);

bgcutFieldfit = [ones(numel(bgcutField),1) ...
    bgcutField];

bgmHFit = bgcutFieldfit\bgcutM; % linear ...
    regression fit of moment versus field data
```

```
            bgchi(g) = bgmHFit(2);

            bgmHFitdata = bgcutFieldfit*bgmHFit;

            bgavgT(g) = sum(bgcutTemp)/numel(bgcutTemp);
        end

        chiActual = zeros(numel(avgT),1);
        invChiActual = zeros(numel(avgT),1);
        for h = 1:numel(avgT)
            chiActual(h) = (chi(h) - bgchi(h));
            invChiActual(h) = 1.0/chiActual(h);

            clear bgTemp bgM bgField
        end
    case 'No'
        chiActual = chi;
        invChiActual = zeros(numel(avgT),1);
        for h = 1:numel(avgT)
            invChiActual(h) = 1.0/chiActual(h);
        end
end
```

B.4.4 Adding the FC file, if using it for additional susceptibility data (not recommended):

```
switch answerAddFCfile
```

```
case 'Yes' % This executes if user decides to add ...
   additional FC data

    cd C:\Users\Fitz\Documents\MATLAB\data\

    additionalFileData = ...
        datFileLoader(numberOfHeaderLinesInput);

    MFC = additionalFileData.Moment_emu_/volume;
    MerrFC = ...
        additionalFileData.M_Std_Err__emu_/volume;
    TempFC = additionalFileData.Temperature_K_;
    FieldFC = additionalFileData.MagneticField_Oe_;

    cleanFileData = ...
        rmmissing(table(MFC,MerrFC,TempFC,FieldFC));
    MFC = cleanFileData.MFC;
    MerrFC = cleanFileData.MerrFC;
    TempFC = cleanFileData.TempFC;
    FieldFC = cleanFileData.FieldFC;

    clear cleanFileData additionalFileData

    tempFCstart = str2double(convertCharsToStrings( ...
        ...
              inputdlg(['Input the temperature' ...
```

```
        ' above which the ZFC and FC data overlap ...
            (in K): ']))));

    MFC = MFC(TempFC >= tempFCstart);
    MerrFC = MerrFC(TempFC >= tempFCstart);
    TempFC = TempFC(TempFC >= tempFCstart);
    FieldFC = FieldFC(TempFC >= tempFCstart);

    chiDC = MFC./FieldFC;
    chiDCerr = MerrFC./FieldFC;

    invChiDC = zeros(numel(chiDC),1);
    invChiDCerr = zeros(numel(chiDC),1);
    for b = 1:numel(chiDC)
        invChiDC(b) = 1.0/chiDC(b);
        invChiDCerr(b) = chiDCerr(b) / ( chiDC(b) ...
            )^2;
    end

    invChiActual = vertcat(invChiActual,invChiDC);
    invChiErr = vertcat(invChiErr,invChiDCerr);

    avgT = vertcat(avgT,TempFC);

    errorbar(avgT,invChiActual,invChiErr)
end
```

B.4.5 Fitting the inverse chi vs T plot using user-selected start and end points:

```
fitInverseChiPlot = questdlg(['Would you like to fit ...
   the inverse chi ' ...
    'vs T plot for user selected points?']);

switch fitInverseChiPlot
    case 'Yes'
        inverseChiFig = figure('Name','Inverse Chi ...
           versus Temperature');
        set(inverseChiFig,'Visible','on')
        hold on
        axes('FontWeight','bold','FontSize',16, ...
                'TickLength',[0.03 0.035],'NextPlot',"add")
        box on
        axesValues = gca;
        axesValues.Position(4) = 0.65;
        annotation('textbox',[0.1, 0.9, 0.1, ...
           0.1],'String', ...
            {'1. Click the low temperature/start point ...
               for fitting', ...
            '2. Hit ENTER,','3. Click the higher ...
               temperature/end point', ...
            '4. Hit ENTER again.'}, ...
            'BackgroundColor','w','FitBoxToText','on')
        errorbar(avgT,invChiActual/(4*pi), ...
                invChiErr/(4*pi),'o','MarkerFaceColor','b')
```

```
xlabel('Temperature (K)')
ylabel('(\chi_0)^{-1}')
hold off

% the following section allows the user to ...
   place a datatip in the plot of
% the inverse chi data for the low-temperature ...
   end of the data to be analyzed to
% find T0.
datacursormode on
dcm_obj = datacursormode(inverseChiFig);
set(dcm_obj,'UpdateFcn',@myupdatefcn)
pause
% Export cursor to workspace
info_struct = getCursorInfo(dcm_obj);
if isfield(info_struct, 'Position')
  disp('Clicked position is')
  disp(info_struct.Position)
  chiStartPt = info_struct.Position;
end

% The following section allows the user to ...
   select the end point of the data
% to be fit (should be the largest temperature ...
   value measured, close to 400
% K with the VSM, if the paramagnetic region is ...
   on the right side of the plot)
```

```
datacursormode on
dcm_obj = datacursormode(inverseChiFig);
% Set update function
set(dcm_obj,'UpdateFcn',@myupdatefcn)
% Wait for the user to click
pause
% Export cursor to workspace and allow the ...
   program to use the values
info_struct = getCursorInfo(dcm_obj);
if isfield(info_struct, 'Position')
  disp('Clicked position is')
  disp(info_struct.Position)
  chiEndPt = info_struct.Position; % this ...
     holds the x and y values of selected end point
end

fitRange = avgT >= chiStartPt(1) & avgT <= ...
   chiEndPt(1); % logical array of indices between
% the start point and end point selected by ...
   user - includes the start and
% end points

chiForFit = invChi(fitRange);
avgTForFit = avgT(fitRange);

% the following section fits the inverse chi ...
   versus T plot in
```

```
% approximately the paramagnetic region and ...
   finds the temp axis
% intercept with error (this is the interaction ...
   temperature)
[chiTfit2,chiTfitStats] = polyfit(avgTForFit, ...
   chiForFit,1);

slope = chiTfit2(1); % slope of fit line from ...
   polyfit
yint = chiTfit2(2); % y intercept of fit line

R = chiTfitStats.R;
df = chiTfitStats.df;
normr = chiTfitStats.normr;

covarianceOfFitCoefficients = (inv(R)*inv(R)') ...
   * normr^2 / df;
% matrix of the covariances for the ...
   coefficients of fit line
variancesOfFitCoefficients = ...
   diag(covarianceOfFitCoefficients);
% variances of the coeffs themselves are the ...
   diag elements
slopeErr = sqrt(variancesOfFitCoefficients(1)); ...
   % error in the slope coefficient
yintErr = sqrt(variancesOfFitCoefficients(2)); ...
   % variance in the y intercept
```

```
        T0 = - yint / slope; % based on Rinaldi paper, ...
           T0 is x-intercept
        T0Err = T0*sqrt( (yintErr/yint)^2 + ...
           (slopeErr/slope)^2 )/sqrt(df) + ...
           max(standardErrT);
        % the first term in T0Err is the standard error ...
           of the calculated
        % T0 from the fit coefficients, the second is ...
           the standard error
        % from the averaging over the temperature points

        % CALCULATING THE R^2 FOR THE LINEAR FIT
        R2 = 1 - (normr/norm(chiForFit - ...
           mean(chiForFit)))^2;

        [chiTfitVals2,delta] = ...
           polyval(chiTfit2,avgT,chiTfitStats);
        % delta gives the std error of the predicted ...
           y-vals
    end
```

B.4.6 Plotting the fit:

```
    switch fitInverseChiPlot
        case 'Yes'
            inverseChiWithFitFig = figure('Name','Inverse ...
               Chi versus Temperature');
```

```
set(inverseChiWithFitFig,'Visible','on')
hold on
axes('FontWeight','bold','FontSize',16, ...
        'TickLength',[0.03 0.035],'NextPlot',"add")
box on
axesValues = gca;
axesValues.Position(4) = 0.7;
resultT0string = strcat('The interaction ...
   temperature is ',num2str(T0), ...
    '+/-',num2str(T0Err),' K');
resultFitString = strcat('The R-squared value ...
   for the linear fit is ',num2str(R2));
annotation('textbox',[0.1, 0.9, 0.1, ...
   0.1],'String', ...
    {resultT0string,resultFitString}, ...
    'BackgroundColor','w', ...
    'FitBoxToText','on')
errorbar(avgT,invChiActual/(4*pi), ...
        invChiErr/(4*pi),'o', ...
        'MarkerFaceColor','b')
plot(avgT,chiTfitVals2/(4*pi),'r:','LineWidth',2)
plot(avgT,(chiTfitVals2+2*delta)/(4*pi), ...
   'r.','LineWidth',1)
plot(avgT,(chiTfitVals2-2*delta)/(4*pi), ...
   'r.','LineWidth',1)
% adds fit line and 95% prediction interval to ...
   plot
```

```
        xlabel('Temperature (K)')
        ylabel('(\chi_0)^{-1}')
        hold off
        hold off
end
```

B.4.7 Function definitions:

```
function fileData = datFileLoader(numberOfHeaderLines)
    folderList = dir;
    folderNames = {folderList.name};
    [folderIndex,~] = listdlg('PromptString','Select ...
       Data Folder Name', ...
        'SelectionMode','single','ListString',folderNames);

    cd(folderNames{folderIndex});
    fileList = dir('*.dat');
    fileNames = {fileList.name};
    [selectedFile,~] = listdlg('PromptString','Choose ...
       .dat file to analyze', ...
        'SelectionMode','single','ListString',fileNames);

    fileData = ...
       readtable(fileList(round(selectedFile)).name, ...
        'HeaderLines',numberOfHeaderLines, ...
        'ReadVariableNames',true);
end
```

```
function [AClean,BClean,CClean] = ...
   cleanData(AData,BData,CData)
    % removes missing/NaN values by row for all ...
       relevant data
    % most helpful for touchdown centering points ...
       (moment will be
    % NaN, so this removes the whole row of data for ...
       that point
    cleanTable = rmmissing(table(AData,BData,CData));
    AClean = cleanTable.AData;
    BClean = cleanTable.BData;
    CClean = cleanTable.CData;
end

function stepPts = findSteps(spacing,TData)
    % spacing is the smallest change in temperature ...
       between
    % consecutive moment vs field measurements
    % TData is temperature data
    % stepPts will be a list of indices where T changes by
    % over 90% of the spacing value
    stepPts = false(numel(TData),1);
    for k = 1:numel(TData)
        if k == numel(TData)
            stepPts(k) = false;
        else
```

```
            if ( TData(k) > (TData(k+1) + ...
               0.9*spacing) )|( TData(k) < (TData(k+1) ...
               - 0.9*spacing) )
                stepPts(k) = ~stepPts(k);
            end
        end
    end

    stepPts = find(stepPts);
end

function [newMag,newT,newH] = ...
   cutByField(Mag,T,H,fieldValue)
    % Mag, T, and H are the magnetization or moment, ...
       temperature, and field
    % data. They should already be separated by the ...
       steps in temperature.
    % newMag, newT, and newH will be smaller sections, ...
       within the field
    % range specified by the user as 'fieldValue'
    newMag = Mag( (H >= -fieldValue) & (H <= ...
       fieldValue) );
    newH = H( (H >= -fieldValue) & (H <= ...
       fieldValue) );
    newT = T( (H >= -fieldValue) & (H <= ...
       fieldValue) );
end
```

```
% ***** this function is used in the user selection of ...
    fit points *****
function output_txt = myupdatefcn(~,event_obj)
  % ~            Currently not used (empty)
  % event_obj     Object containing event data structure
  % output_txt    Data cursor text
  pos = get(event_obj, 'Position');
  output_txt = {['x: ' num2str(pos(1))], ['y: ' ...
     num2str(pos(2))]};
end
```

B.5 Hysteresis Data Analysis

```
% plots and normalizes (if selected) VSM and MOKE ...
    hysteresis data
% user should put data to be analyzed into a separate ...
    folder in 'data' folder in MATLAB directory

% can select multiple files, but only one background file
```

B.5.1 Read Files and Allocate Space

```
cd C:\Users\Fitz\Documents\MATLAB\data\

numberOfHeaderLinesInput = ...
    str2double(convertCharsToStrings(inputdlg(['Input ...
    number of ' ...
     'header lines in the data file'],'Header Lines')));
```

```
[selectedFileData,selectedFileNames] = ...
    datFileLoaderMulti(numberOfHeaderLinesInput);
```

B.5.2 The first switch case is VSM and second is MOKE data:

```
answerMOKE = questdlg('Is this multirun MOKE data?');

switch answerMOKE

    case 'No'    % THIS CASE IS FOR VSM DATA

        numFiles = numel(selectedFileData);

        colorVector = hsv(numFiles);

        mSat = zeros(numFiles,1);
        Hc = zeros(numFiles,1);
        mRem = zeros(numFiles,1);
        squareness = zeros(numFiles,1);

        MvHFig = figure('Name','Magnetization vs Field')
        figure(MvHFig)
        axes('FontWeight','bold','FontSize',16, ...
                'TickLength', ...
            [0.03 0.035],'NextPlot','add')
        box on
```

```
demagN = ...
    str2double(convertCharsToStrings(inputdlg([ ...
     'Input the demagnetization factor ' ...
     '(positive value, ' ...
     'SI units) for your sample. If you plan to ...
        ignore demagnetization' ...
     ' effects, put 0.'])));

answerBG = questdlg(['Does the data need to ' ...
     'have a background contribution removed?']);

answerNormalize = questdlg('Does the data need ...
    to be normalized?');

answerParamagnetic = questdlg('Is the sample ...
    paramagnetic?');

answerSPMcheck = questdlg(['Are these VSM mvH ...
    measurements at various T to check for' ...
     ' SPM behavior?']);

for n = 1:numFiles

     properties = ...
        convertCharsToStrings(inputdlg({'Input ...
        mass of sample in grams', ...
```

```
    'Density in grams/cc'},'Mass and ...
        Density',2));
volume = str2double(properties(1) ...
    )/str2double(properties(2));

fileData = selectedFileData{n};

M = fileData.Moment_emu_/volume;
Merr = fileData.M_Std_Err__emu_/volume;
T = fileData.Temperature_K_;
H = fileData.MagneticField_Oe_;

cleanFileData = rmmissing(table(M,Merr,T,H));
M = cleanFileData.M;
Merr = cleanFileData.Merr;
T = cleanFileData.T;
H = cleanFileData.H;

clear cleanFileData

H = H - 4.0*pi*demagN*M;

[~,minHpt] = min(H);

Mdescending = M(1:minHpt);
Mascending = M((minHpt + 1):end);
Hdescending = H(1:minHpt);
```

```
Hascending = H((minHpt + 1):end);

% MAXIMUM FIELD FITTING FOR PARAMAGNETIC SLOPE

maxInitFieldEl = (H > (0.75*max(H))) & ...
    (find(H) < 0.5*numel(H));

MaxInitFieldDataToFit = H(maxInitFieldEl);

%numMaxInitFitPts = ...
    length(MaxInitFieldDataToFit); % counts ...
    points to be fit
maxInitX = ...
    [ones(length(MaxInitFieldDataToFit),1) ...
    MaxInitFieldDataToFit];
maxInitY = M(maxInitFieldEl);

maxInitFit = maxInitX\maxInitY; %first ...
    element is intercept, second element is ...
    slope



maxFinFieldEl = (H > (0.75*max(H))) & ...
    (find(H) > 0.5*numel(H));

MaxFinFieldDataToFit = H(maxFinFieldEl);
```

```
%numMaxFinFitPts = ...
    length(MaxFinFieldDataToFit); % counts ...
    points to be fit
maxFinX = ...
    [ones(length(MaxFinFieldDataToFit),1) ...
    MaxFinFieldDataToFit];
maxFinY = M(maxFinFieldEl);

maxFinFit = maxFinX\maxFinY; % first ...
    element is intercept, second element is ...
    slope


% MINIMUM FIELD FITTING FOR PARAMAGNETIC SLOPE

MinFieldDataToFit = H(H < 0.85*min(H));

numMinFitPts = length(MinFieldDataToFit); %...
     counts points to be fit
minX = [ones(numMinFitPts,1) ...
    MinFieldDataToFit]; % y data for fit ...
    (field data,
% with y-inercept column)
minY = M(H < 0.85*min(H));

minFit = minX\minY;
```

```
fitCalcY = minX*minFit;

mSat(n) = minY(1) - minFit(2)*minX(1,2); % ...
   more accurate calculation of ms
% than just intercept

field = H;

switch answerBG
    case 'Yes'
        cd C:\Users\Fitz\Documents\MATLAB\data\

        numberOfHeaderLinesInput = ...
           str2double( ...
               convertCharsToStrings( ...
               inputdlg([ ...
            'Input number of ' ...
            'header lines in the data ...
               file '],'Header Lines')));

        selectedBackgroundFileData = ...
           datFileLoader( ...
            numberOfHeaderLinesInput);
        bgproperties = ...
           convertCharsToStrings(inputdlg( ...
            {'Input mass of sample in ...
               grams', ...
```

```
                'Density in grams/cc'},'Mass and ...
                    Density',2));
                bgvolume = str2double(properties(1 ...
                ))/str2double(properties(2));

                bgM = ...
selectedBackgroundFileData.Moment_emu_/ ...
                    bgvolume;
                bgMerr = ...
selectedBackgroundFileData.M_Std_Err__emu_;
                bgT = ...
selectedBackgroundFileData.Temperature_K_;
                bgH = ...
selectedBackgroundFileData.MagneticField_Oe_;

                cleanFileData = ...
                    rmmissing(table(bgM,bgMerr,bgT,bgH));
                % Missing entries will usually only ...
                    be in M, so in
                % order to maintain the same length ...
                    for all data arrays, we
                % remove the missing lines from a ...
                    table, rather than
                % individual arrays. Also won't ...
                    work on the whole 58
                % column table, since lots of the ...
                    entries are blank (e.g.
```

```
% comments, etc.)
bgM = cleanFileData.bgM;
bgMerr = cleanFileData.bgMerr;
bgT = cleanFileData.bgT;
bgH = cleanFileData.bgH;

clear cleanFileData

[~,minBgHpt] = min(bgH);

bgMdescending = bgM(1:minBgHpt);
bgMascending = bgM((minBgHpt + ...
    1):end);
bgHdescending = bgH(1:minBgHpt);
bgHascending = bgH((minBgHpt + ...
    1):end);

Mdescending = ...
    subtractBackground(Mdescending, ...
      Hdescending, ...
      bgMdescending,bgHdescending);
Mascending = ...
    subtractBackground(Mascending, ...
      Hascending, ...
      bgMascending,bgHascending);

M = vertcat(Mdescending,Mascending);
```

```
            clear bgM bgH bgMerr bgT ...
                bgMascending bgMdescending ...
                bgHascending bgHdescending
            clear selectedBackgroundFileData

            figure(MvHFig)
            yaxname = 'Magnetization (emu/cc)';
        case 'No'
            figure(MvHFig)
            yaxname = 'Magnetization (emu/cc)';
    end

    switch answerParamagnetic
        case 'Yes'
            muVolInput = convertCharsToStrings ...
                (inputdlg(['Input ...
                    atomic/molecular/NP' ...
                ' radius in cm']));
            muVol = 4*pi/3.0 * ...
                (str2double(muVolInput))^3;

            MsBulk = str2double( ...
                convertCharsToStrings( ...
                inputdlg(['Input the ' ...
```

```
    'magnetization for the bulk ...
       material of your sample in ...
       emu/g' ...
    ]))))*str2double(properties(2));

HforLangevinFit = ...
   Hdescending(Mdescending > 0);
MforLangevinFit = ...
   Mdescending(Mdescending > 0);
xForLangevinFit = ...
   MsBulk*muVol*HforLangevinFit/( ...
    1.38*10^-16*mean(T));



fitOps = fitoptions('Method', ...
    'NonlinearLeastSquares',...
   'Lower',0,...
   'Upper',Inf,...
   'StartPoint',1);
fitTypeDef = fittype( ...
    'Ms*(coth(x)-1.0/(x))', ...
    'independent',{'x'}, ...
    'coefficients', ...
    {'Ms'},'options',fitOps);
magnetizationFit = ...
   fit(xForLangevinFit, ...
    MforLangevinFit,fitTypeDef)
```

```
        MagFitLangFigure = figure('Name', ...
            'Magnetization Fit with Langevin ...
               Function')
        figure(MagFitLangFigure)
        axes('FontWeight','bold', ...
            'FontSize',16,'TickLength',[0.03 ...
               0.035], ...
            'NextPlot','add')
        box on
        hold on
        plot(magnetizationFit, ...
            xForLangevinFit,MforLangevinFit)
        %xlim([min(xForLangevinFit) ...
           max(xForLangevinFit)]);
        hold off

end

switch answerSPMcheck
    case 'Yes'
        figure(MvHFig)
        %avgT = sum(T)/numel(T); % finds ...
           average temperature over the mvH ...
           curve
        M = M/abs(mSat(n)); % m/ms for ...
           y-axis data, ms is the saturation
```

```
        % moment of the room temp data
        field = field/mean(T); % H/average ...
            temperature for x-axis data
        % aka 'universal curve'
        xlim([-75.0 75.0]); % this might ...
            need to be updated, depending on ...
            field values
    case 'No'
        figure(MvHFig)
        xlim([min(H) max(H)]);
end

switch answerNormalize
    case 'Yes'
        figure(MvHFig)
        M = (M - minFit(2)*H) / ...
            abs(minFit(1));
        % normalized moment data from fit ...
            of saturation near max field value
        yaxname = 'M/M_s (a.u.)';
    case 'No'
        figure(MvHFig)
        yaxname = 'Magnetization (emu/cc)';
end

figure(MvHFig)
hold on
```

```
plot(field,M, ...
    'Color',colorVector(n,:),'LineWidth',2)

maxPts = H > (max(H)*0.9999);
startPt = find(maxPts,1,'first');

field = H(startPt:end);
moment = M(startPt:end);

pts = length(M);
    for i = 1:(pts-1)
        if (M(i) > 0) && (M(i+1) < 0)
            x1 = [ones(2,1) [H(i) H(i+1)]'];
            y1 = [M(i) M(i+1)]';
            y2 = x1\y1;

            Hc(n) = y2(1)/y2(2); % ...
                calculates the coercivity ...
                from fit line
        elseif (M(i) > 0) && (M(i+1) ==...
            0)
            Hc(n) = H(i+1);
        end
    end

for j = 1:(pts-1)
    if (H(j) > 0) && (H(j+1) < 0)
```

```
            x3 = [ones(2,1) [H(j) H(j+1)]'];
            y3 = [M(j) M(j+1)]';
            yN3 = [M(j) M(j+1)]';
            y4 = x3\y3; % Calculates remnant ...
                moment
            yN4 = x3\yN3; % calculates ...
                squareness of normalized loops ...
                (Mr/Ms, Ms=1 for M)

            mRem(n) = y4(1);
            squareness(n) = yN4(1);
        elseif (H(j) > 0) && (H(j+1) == 0)
            mRem(n) = M(j+1);
            squareness(n) = M(j+1);
        end
    end
end

xlabel('Field (kOe)')
ylabel(yaxname)
prompt = cell(numFiles,1); % allocates space ...
   for plot names
legendTitle = 'Name Plots for Legend';
selectedFileNames = cell2mat(selectedFileNames);
for v = 1:numFiles
    prompt{v} = selectedFileNames(n); % sets ...
       prompt of naming dialog box to file name
```

```
    end
    legendNames = inputdlg(prompt,legendTitle);
    % prompts user to input plot names for each ...
       file to be plotted
    legend(legendNames,'Location','northwest')
    hold off

case 'Yes'  % THIS CASE IS FOR MOKE DATA

    answer2 = questdlg('Do all of your MOKE runs ...
       need to be normalized?');

    fileInfo = dir('*_Ch1'); % lists information ...
       about files in current folder
    numFiles = numel(fileInfo(:,1)); % counts ...
       number of files in selected folder

    colorVector = winter(numFiles); % makes all ...
       MOKE data (multiruns of same sample) plots ...
       blue-green

    % The following lines open the first file data ...
       as a test, and
    % calculates the number of points per run (that ...
       way user doesn't
    % have to input/guess anymore
    testFile = readtable(fileInfo(1).name, ...
```

```
    'Headerlines',2,'ReadVariableNames',false)
testFile.Properties.VariableNames = {'Field','V'};
runPts = numel(testFile.V);

%runPts = input('Number of Points per Run: ');
M = zeros(runPts,numFiles);

axes('FontWeight','bold', ...
    'FontSize',16,'TickLength',[0.03 ...
       0.035],'NextPlot','add')
box on
hold on
for n = 1:numFiles % iterates through all ...
   files in folder
    fileData = readtable(fileInfo(n).name, ...
        'HeaderLines',2,'ReadVariableNames',false);
                % reads data from nth file, ...
                   skips first 22 lines of text,
                % TODO name variables manually
    fileData.Properties.VariableNames = ...
       {'MagneticField_Oe_',...
        'Signal_V'};

    switch answer2
        case 'Yes'

            maxField = max(H);
```

```
        maxFieldEl = H > (max(H) - ...
            0.1*max(H));
% creates logical array for elements that ...
    are within 1% of the maximum field
% i.e. the points where the moment is ...
    saturated

        MinFieldDataToFit = H(maxFieldEl);
% makes array of field data near max field ...
    saturation points

        numMaxFitPts = ...
            length(MinFieldDataToFit); % ...
            counts points to be fit
        minX = [ones(numMaxFitPts,1) ...
            MinFieldDataToFit]; % y data for ...
            fit
        % (field data, with y-inercept column)
        minY = fileData.Signal_V(maxFieldEl);

        minFit = minX\minY;
        fitCalcY = minX*minFit;

        M(:,n) = (fileData.Signal_V - ...
            minFit(2)*H) / minFit(1);
        % normalized moment data from fit ...
            of saturation near max field value
```

```
            % this moment data is stored in the ...
              n'th column

            yAxLabel = 'M/Ms (a.u.)';

        case 'No'
            M(:,n) = fileData.Signal_V;
            % M is just signal data, doesn't ...
              need to be normalized from questdlg
            % data is appended to M as-is, this ...
              might be computationally taxing ...
              or unnecessary

            yAxLabel = 'Lock-in output (V)';
    end

    figure(1)
    plot(H,M(:,n),'Color',colorVector(n,:))
    % 'Color' command will set line color for ...
       each plot
end

field = H;

avgM = sum(M,2)/numFiles; % returns a column ...
   vector that is
% the sum of all normalized moment data
```

```
% divided by the number of runs
plot(H,avgM,'r','LineWidth',2)
yticklabels({})
xlabel('Field (Oe)')
ylabel(yAxLabel) % changes y axis label based ...
   on if data is normalized or not

% The following section CENTERS THE AVERAGED ...
   MOKE SIGNAL ABOUT
% X-AXIS
maxFieldAvg = max(field);
maxFieldElAvg = field > (maxFieldAvg - ...
   0.25*maxFieldAvg);
% creates logical array for elements that are ...
   within certain % of the maximum field
% i.e. the points where the moment is saturated ...
   (might need to be changed depending
% on sample/measurement properties

MaxFieldDataToFitAvg = field(maxFieldElAvg);
% makes array of field data near max field ...
   saturation points using logical array ...
   maxFieldElAvg

MaxMomDataToFitAvg = avgM(maxFieldElAvg);
% makes array of averaged moment data near max ...
   saturation using same logical array as above
```

```
MaxMomVal = sum(MaxMomDataToFitAvg)/numel( ...
    MaxMomDataToFitAvg);
% averaged saturation value of averaged signal ...
  to find upper edge of hysteresis loop

minFieldAvg = min(field);
minFieldElAvg = field < (minFieldAvg - ...
  0.25*minFieldAvg);
% creates logical array for elements within ...
  certain % of min field
% lower saturated curve

MinMomDataToFitAvg = avgM(minFieldElAvg);
% makes array of averaged moment data near min ...
  saturation
MinMomVal = sum(MinMomDataToFitAvg)/numel( ...
    MinMomDataToFitAvg);
% averaged saturation value of averaged singla ...
  to find lower edge of hysteresis loop

midPtAvg = (MinMomVal + MaxMomVal)/2.0;
% finds midpoint of averaged MOKE hysteresis loop
avgMCentered = avgM - midPtAvg;
% subtracts calculated midpoint from all data ...
  to center the loop about y = 0 (x-axis)
```

```
answer3 = questdlg(['Do you want to ' ...
    'normalize the averaged MOKE data?']);

    switch answer3
        case 'Yes' % averaged MOKE data will be ...
            normalized, must also be made center ...
            around y = 0
                      % which VSM data doesn't ...
                         need. to do this we ...
                         average the moment values ...
                         near
                      % the max and min ...
                         saturation points
                      % then average those two ...
                         values to find midpoint
                      % then subtract that number ...
                         from all averaged signal
                      % (previous section) to ...
                         find avgMCentered, then
                      % normalize the centered data
                normAvgM = avgMCentered/(MaxMomVal -...
                    midPtAvg);

    end

    pts = length(avgMCentered);
        for i = 1:(pts-1)
```

```
            if (avgMCentered(i) > 0) && ...
               (avgMCentered(i+1) < 0)
                x1 = [ones(2,1) [field(i) ...
                   field(i+1)]'];
                y1 = [avgMCentered(i) ...
                   avgMCentered(i+1)]';
                y2 = x1\y1; % finds eq of line ...
                   between points
                % above and below zero crossing ...
                   of moment vs field

                Hc = y2(1)/y2(2); % calculates ...
                   the coercivity from fit line
            elseif (avgMCentered(i) > 0) && ...
               (avgMCentered(i+1) == 0)
                Hc = field(i+1);
            end
        end

hold off

figure(2) % creates separate figure for the ...
   normalized, averaged data
axes('FontWeight','bold', ...
    'FontSize',16,'TickLength',[0.03 0.035], ...
       'NextPlot','add')
box on
```

```
        hold on
        plot(field,normAvgM,'r','LineWidth',2)
        %yticklabels({})
        xlabel('Field (Oe)')
        ylabel('M/Ms (a.u.)')
        hold off

end
```

B.5.3 Function definitions:

```
function [fileData,fileNames] = ...
    datFileLoaderMulti(numberOfHeaderLines)
      % numberOfHeaderLines is the input number of lines ...
        for the readtable
      % function to skip when importing data (it is ...
        typically 30 for this VSM
      % data) not including the line with the variable ...
        names.
      folderList = dir;
      folderNames = {folderList.name};
      [folderIndex,~] = listdlg('PromptString','Select ...
        Data Folder Name', ...
         'SelectionMode','single','ListString',folderNames);


      cd(folderNames{folderIndex});
      fileList = dir('*.dat');
      fileNames = {fileList.name};
```

```
    [selectedFiles,~] = listdlg('PromptString','Choose ...
       .dat file to analyze', ...
        'SelectionMode','multiple','ListString',fileNames);

    fileData = cell(numel(selectedFiles),1);
    for n = 1:numel(selectedFiles)
        fileData{n} = readtable(fileList( ...
                round(selectedFiles(n))).name, ...
            'HeaderLines',numberOfHeaderLines, ...
            'ReadVariableNames',true);
    end
end

function fileData = datFileLoader(numberOfHeaderLines)
    % numberOfHeaderLines is the input number of lines ...
       for the readtable
    % function to skip when importing data (it is ...
       typically 30 for this VSM
    % data) not including the line with the variable ...
       names.
    folderList = dir;
    folderNames = {folderList.name};
    [folderIndex,~] = listdlg('PromptString', ...
        'Select Data Folder Name', ...
        'SelectionMode','single','ListString',folderNames);

    cd(folderNames{folderIndex});
```

```
    fileList = dir('*.dat');
    fileNames = {fileList.name};
    [selectedFile,~] = listdlg('PromptString', ...
        'Choose .dat file to analyze', ...
        'SelectionMode','single','ListString',fileNames);

    fileData = ...
       readtable(fileList(round(selectedFile)).name, ...
        'HeaderLines', ...
        numberOfHeaderLines,'ReadVariableNames',true);
end

function noBgM = ...
   subtractBackground(Moment,X,BackgroundMoment,BackgroundX)
    % Moment is the input moment or magnetization data
    % X is the independent parameter from the moment ...
       measurement (usually
    % either field or temperature)
    % BackgroundMoment is the moment or magnetization ...
       of the background
    % BackgroundX is the indep. param. from the ...
       background measurement
    % which must correspond to the same units as X
    % noBgM will be the new moment/magnetization data ...
       without the
    % background signal
    k = numel(Moment);
```

```
    b = numel(BackgroundMoment);
    noBgM = zeros(k,1);
    for c = 1:k
        [~, bgpt] = min(abs(X(c) - BackgroundX));

        if bgpt == 1 % case for first index
            bgFit = polyfit(BackgroundX(bgpt:(bgpt + ...
                1)), ...
                 BackgroundMoment(bgpt:(bgpt + 1)), 1);
        elseif bgpt == b % case for last index
            bgFit = polyfit(BackgroundX((bgpt - ...
                1):bgpt), ...
                 BackgroundMoment((bgpt - 1):bgpt), 1);
        else
            bgFit = polyfit(BackgroundX((bgpt - ...
                1):(bgpt + 1)), ...
                 BackgroundMoment((bgpt - 1):(bgpt + ...
                    1)), 1);
        end

        noBgM(c) = Moment(c) - (bgFit(1) * X(c) + ...
            bgFit(2));
        % subtracts the background from the moment
        % bgFit(1) is the slope, bgFit(2) is the intercept
    end
end
```