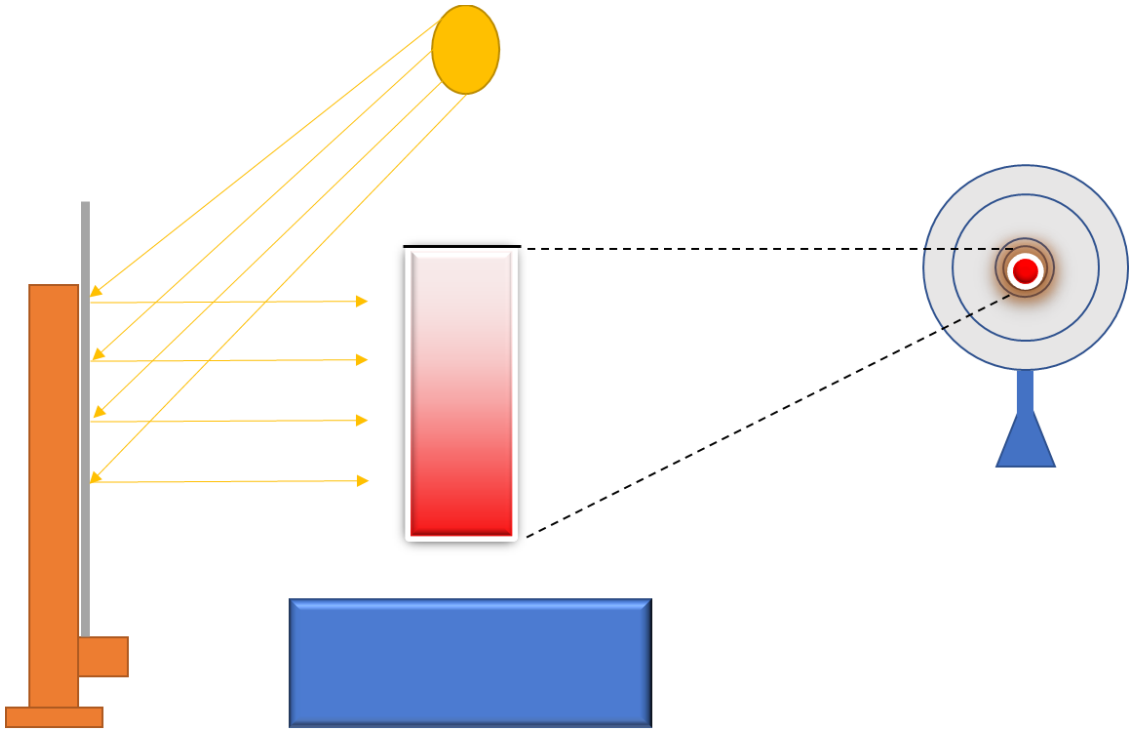
Master's Thesis

Optical Analysis and Theoretical Modelling of Colloidal Maghemite Sedimentation in Magnetic Field



Utrecht University

Physical & Colloid Chemistry

Author:

Richard Koschny

Supervisors:

Alex van Silfhout

Ben Erné

Monday, 24th of June 2019

Abstract

Superparamagnetic fluids are colloidal suspensions of ferri- or ferromagnetic nanoparticles in a liquid solvent. For applications of these magnetic fluids, it is essential that sedimentation of the suspended particles in an external magnetic field is slow. This Master's Thesis examines the sedimentation of suspended nanoparticles of a superparamagnetic fluid in a magnetic field, both in theory and experiment. Two simple models to predict the behavior of superparamagnetic nanoparticles in a magnetic field are developed, a sedimentation-diffusion equilibrium model (I) and a dynamic model (II) respectively. Both calculate the forces acting on the nanoparticles and evaluate concentration profiles from this. Model I evaluates the concentration profile at sedimentation-diffusion equilibrium. Model II calculates dynamically the concentration profiles in time between the start of the sedimentation and as the magnetic fluid approaches the sedimentation-diffusion equilibrium. Additionaly, an optical detection technique which measures the absorption of visual light of the superparamagnetic fluid to capable of measuring the nanoparticle concentration profiles in time was developed. Experiments were performed on nitrate coated maghemite superparamagnetic fluids in a magnetic field. The results obtained from the two models were compared with one another and with the concentration profiles obtained from the experimental setup.

The optical setup to monitor sedimentation in a magnetic field yielded promising results. The experimental equilibrium and time-dependent concentration profiles were qualitatively similar but not quantitatively the same as predicted by our theoretical calculations.

Index

1: Introduction

1.1 Context

In this Master's thesis theoretical models were developed together an optical experimental setup to analyze the sedimentation of maghemite nanoparticles in a superparamagnetic fluid. Here the International Union of Pure and Applied Chemistry's (IUPAC) definition of nanoparticles is used, namely a "particle of any shape with dimension in the 1 x 10-9 and 1 x 10-7 m range" (Vert et al. 2012). Of particular interest to this thesis are superparamagnetic (maghemite) nanoparticles of roughly 10 nm and smaller, which behave as single-domain non-interacting magnetic particles. Paramagnetism is caused by the presence of unpaired electrons in the (maghemite) nanoparticles. In the absence of an external magnetic field H (in A/m) the (maghemite) nanoparticles will on average not be magnetized. Thermal fluctuations randomize the spin orientations. Yet, in the presence of a magnetic field (H), parallel alignment of the electrons' spins will occur in the nanoparticles, which in turn results in the magnetic moments aligning in the derection of the magnetic field aligned (see figure 1). The magnetization M (in A/m) in an external magnetic field (H) is given by $H = \chi * M$, where χ displays the magnetic susceptibility of the nanoparticles (Luigjes 2012). The magnetization (M) of superparamagnets is much larger than that showcased by paramagnets (higher magnetic susceptibility), hence the name superparamagnetism (Marolt 2014).

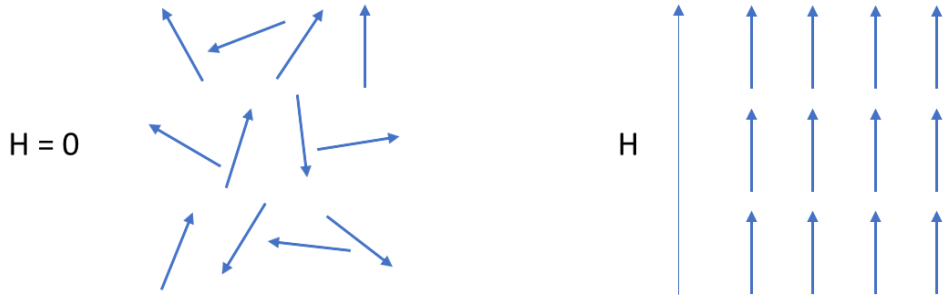


Figure 1: Schematic illustration of (super)paramagnetism, in which the arrows represent the directionality of the magnetic moments of the nanoparticles when no external magnetic field is present (to the left) and when an external magnetic field is present (to the right) the magnetic field is represented by the large arrow (adapted from (Luigjes 2012))

Superparamagnetic fluids were first developed by Stephen Papell of the National Aeronautics and Space Administration (NASA) in the 1960s as a way to manipulate a carrier liquid in zero gravity (1963). Superparamagnetic fluids have strongly paramagnetic properties under ambient temperature and virtually any

temperature between the freezing and boiling points of the solvent (Rosensweig 1982). This makes superparamagnetic fluids useful in a manifold of potential applications. Examples include but are not limited to: zero-leakage rotary shaft seals used in cd drives, coolant for loudspeakers, magnetic ink in ink-jet printers (Bailey 1983, Rosensweig 1982). Superparamagnetic fluids are also used in magnetic levitation (magneto-Archimedes levitation); applications include microplastic recycling or separation of diamonds from gangue material (Fay et al. 1975, Vatta 2005, Kimura et al. 2005, van Silfhout et al. 2019).

In spite of superparamagnetic (iron oxide) nanoparticles being toxic in high concentrations (Singh et al. 2010), sufficiently small (iron oxide) nanoparticles (smaller than 10 nm in diameter) suspended in water are quickly removed from vital organs and tissue (Liu et al. 2013). Therefore, aqueous superparamagnetic fluids can also be used in biochemical applications such as contrast enhancement in magnetic resonance imaging (MRI), selective separation of nucleic acids, water treatment, targeted drug delivery and in the catabolism of tumours by hyperthermia (Horák et al. 2007, Pankhurst et al. 2003).

In this research, the magnetic fluid analysed consisted of superparamagnetic maghemite (iron oxide) nanoparticles suspended in water. An advantage of using suspended maghemite (and magnetite) nanoparticles in magnetic fluids compared to for instance other iron oxide nanoparticles is their high saturation magnetization (60-80 Am^2kg^{-1} for maghemite and 92-100 Am^2kg^{-1} for magnetite) (Cornell et al. 2003). For iron oxide superparamagnetic nanoparticles the saturation magnetization is usually 20 to 50 percent lower than the bulk saturation magnetization (Horák et al. 2007). At room temperature maghemite is ferrimagnetic. If the maghemite nanoparticles are small enough for every particle to contain one magnetic domain (smaller than roughly 10 nm in diameter) the nanoparticles are superparamagnetic (Coey et al. 1972, Cornell et al. 2003). Maghemite nanoparticles suspended in water are therefore a viable candidate for superparamagnetic fluids, even at high temperatures.

1.2 Aim of research

The aim of this research is to better understand the process of sedimentation of nanoparticles in a superparamagnetic fluid in an external magnetic field and to develop a technique for superparamagnetic fluid stability testing in a magnetic field. Sedimentation of nanoparticles in magnetic field results in lower efficiency of the magnetic fluid in applications. The research question to be answered is therefore the following: Can the behavior of superparamagnetic nanoparticles in magnetic field be predicted using a simple model? In order to obtain a theoretical grasp of the sedimentation process in a magnetic field, two computational models were

developed with the general purpose programming language python (shown in appendices IV and VII); the sedimentation-diffusion equilibrium model (I) calculates the equilibrium profile of sedimentation from the force balance at equilibrium; the dynamic model (II) determines the development of the concentration profiles in time. In equilibrium a concentration gradient will have set in, with the highest average concentration being present at the bottom of the fluid column and the lowest average concentration being present at the top of the fluid column. Letting a sufficient amount of time elapse in model II should make the calculated concentration profile converge with the concentration profile calculation with model I. Both of these models will be explained further in chapter 2. In chapter 3, the methods used in this research will be explained. In chapter 4 the results obtained from the methods in chapter 3 will be shown. In chapter 4, also the two models will be compared to the results obtained from experiments. Chapter 5 gives the conclusions and outlook for further research.

1.3 Sedimentation of Superparamagnetic Nanoparticles in a Magnetic Field

In a magnetic field the superparamagnetic particles will sediment, resulting in a decrease of the efficiency of the superparamagnetic fluid in applications.

In order to simplify the calculations, in the model described in this thesis nanoparticles are assumed to be non-interacting ideal particles. Therefore, agglomeration of nanoparticles in a magnetic field will not be included in the model. This could be a considerable limitation, since it has been shown that in a magnetic field, nanoparticles do indeed agglomerate, forming dipolar structures such as chains, wormlike clusters or flux-closure rings, especially in aqueous superparamagnetic fluids in strong magnetic fields, even out of field (de Gennes et al. 1970, Peterson et al. 1977, Philipse et al. 2002, Planken et al. 2009). Also, in this simplified model, interactions between the superparamagnetic nanoparticles will be taken to be exactly zero. In other words, the nanoparticles are assumed to be hard spheres with no electrostatic charge and the Vanderwaals forces between spheres is assumed to be negligible. The height of the fluid column is also taken to be small, so that a density gradient due to gravity in height is avoided. This holds when $h < \frac{k_b T}{v \, \Delta\varrho \, g}$ (de Gennes et al. 1970), with k_b the Boltzmann constant (1.38065e^{-23} J/K), T the absolute temperature taken to be 293 K, $\Delta\varrho$ the mass density of the maghemite nanoparticles which is taken to be 4.9 g/cm^3 (Ramos Guivar et al. 2014), g the gravitational constant 9.81 m/s^2. R the largest particle radius is taken to be 8.25 nm, so that: $v = \frac{4}{3}\pi R^3 = 2.4e^{-24}\, m^3$. Therefore, for $h < 3.5\, cm$ a density gradient in height is negligible. Since the height of the fluid column in the

experiments described later was not much more than 1.0 cm, the assumption of no gradient in height due to gravity holds.

Due to the small width of the fluid column, the magnetic field lines running through the liquid column are taken to run parallel to one another. The magnetic field is dependent on height above the magnet as given by equation (1).

$$B(h) = \frac{\mathrm{M}}{2}\left(\left(\frac{h+d}{\sqrt{(h+d)^2 + R^2}}\right) - \left(\frac{h}{\sqrt{h^2 + R^2}}\right)\right) \quad (1)$$

Here B(h) displays the magnetic field strength in T, M the magnet's (internal) remanent magnetization, z the distance from the magnet, d the magnet's thickness and R the magnet's radius (van Silfhout et al. 2019). In figure 2 the magnetic field strength as a function of distance from the magnet given by equation (1) is shown for a neodymium magnet used in the experiments discussed in chapter 3.1. It can be seen that the strength of the magnet decreases when the distance from the magnet is increased.

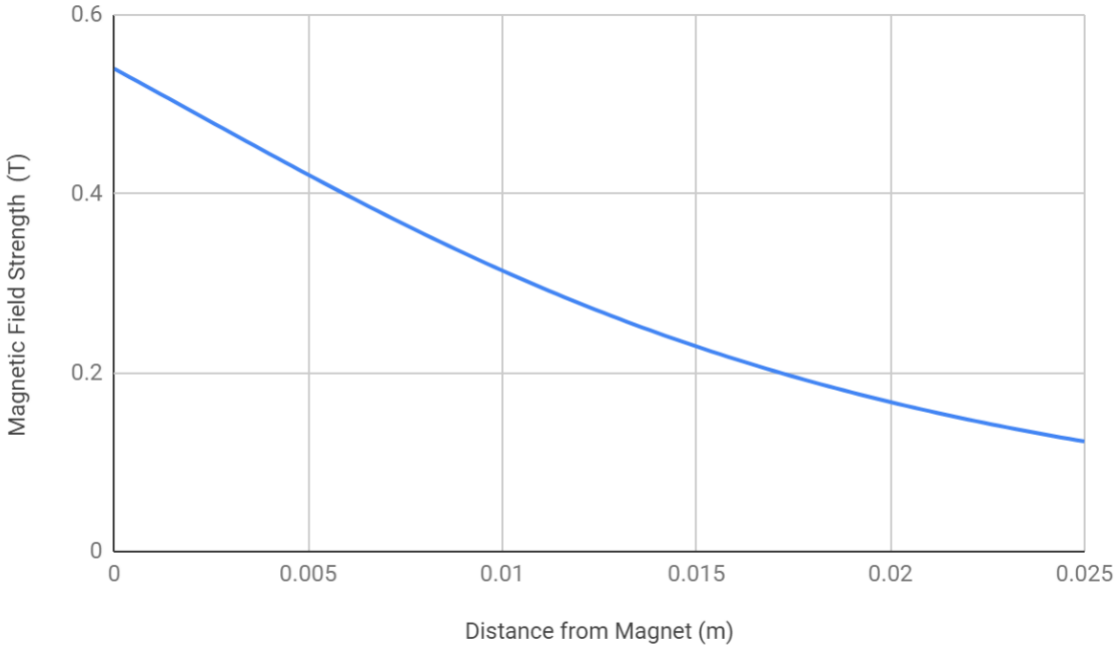


Figure 2: Magnetic field strength according to magnetic characterization theory of a neodymium magnet with a radius of 2.25 cm, a thickness of 3.0 cm and a remanent magnetization of 1.35 T

2: Theoretical Models

In this chapter, two theoretical models are discussed which were used to provide a theoretical framework of the sedimentation of superparamagnetic nanoparticles in magnetic fluids. The first model is the sedimentation-diffusion equilibrium model (I). In the sedimentation-diffusion equilibrium model (I) the concentration profile at sedimentation-diffusion equilibrium is determined from the force balance at equilibrium. The dynamic model (II) describes the development of concentration profiles in time.

2.1 Sedimentation-Diffusion Equilibrium Model

In the sedimentation-diffusion equilibrium model (I), three forces are taken to act on the superparamagnetic nanoparticles, the magnetic force (F_{mag}) and the gravitational force (F_{grav}) are acting in the same direction, the diffusion force (F_{dif}) is acting in the opposite direction of the other two forces (see figure 3). Unlike in the dynamic model (II), as explained in the next section, at sedimentation-diffusion equilibrium there will be no drag force (F_{drag}). This is the case since the average velocity of the nanoparticles at equilibrium is zero. Furthermore, at all points in the vertical direction the three forces are in equilibrium, i.e. equation 2 holds:

$$F_{mag} + F_{dif} + F_{grav} = 0 \qquad (2)$$

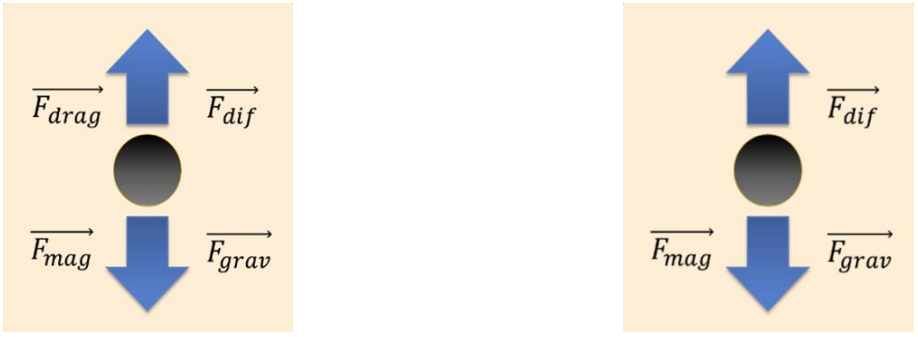


Figure 3: Left: Forces on a sedimenting superparamagnetic nanoparticle, right: Forces on superparamagnetic nanoparticle at sedimentation-difussion equilibrium

Since this model is to be compared to experimental results, where the fluid is contained in rectangular cuvettes (8cmx1cmx0.2cm) with a stopper, also in model I the superparamagnetic nanoparticles are taken to be present in a rectangular fluid column. This column is divided into identical volume elements (slabs) of height dh (see figure 4). The magnetic liquid column starts at a distance of 0.010 meters from the magnet and is 0.010 m in height. The forces on the superparamagnetic

nanoparticles are taken not to differ within the slabs but only vary between slabs. Moreover, it is assumed that the magnetic field lines are close to parallel, so that only the vertical component of the magnetic field has to be taken into account. This simplification can be made since the width of the slabs and the height of the fluid column are sufficiently small and a strong (neodymium) magnet was used. In the experimental setup the liquid column than should be placed as close to the magnet as possible, since the curvature of the field lines increase the further one gets from the magnet. Furthermore, the diffusive force is taken as an average of a slab, the precision of this force at a given height in the fluid column depends on the size of the height (dh); a smaller dh equates to a higher precision.

In the experimental setup the width of the cuvette is not very large (1 cm) and is located above the center of the magnet during the experiment, so that indeed the magnetic field lines run approximately parallel. In the python code the thickness of the slab is represented by the variable called 'step size', this variable is equivalent in order of magnitude to the height of a single pixel of the PNG files obtained in the experimental setup. A smaller value for the step size variable would increase accuracy, but it also would increase the calculation time.

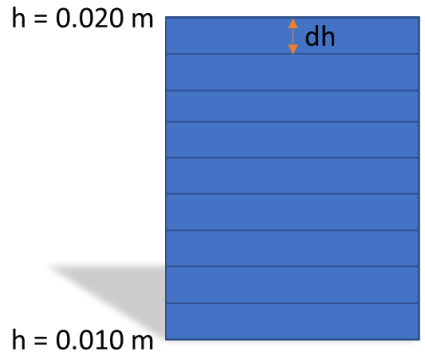


Figure 4: Liquid column of superparamagnetic fluid

Each of the forces acting on the superparamagnetic nanoparticles will now be examined in greater detail. Firstly, the gravitational force (F_{grav}) acting on a single nanoparticle is shown:

$$F_{grav} = \Delta m\, g = \left(\rho_w - \rho_{np}\right)\frac{1}{6}\pi D^3\, g \tag{3}$$

Where $\frac{1}{6}\pi D^3$ is an expression of the volume of the nanoparticles, with D being the diameter of the nanoparticle, ρ_w is the mass density of the solvent, ρ_{np} the mass

density of the nanoparticles and g the gravitational acceleration. Secondly, the magnetic force acting on a single nanoparticle is given:

$$F_{mag} = \mu_{np} \frac{dB(h)}{dh} = \frac{1}{6}\pi D^3 M_b \, L(\xi\,(\mathrm{B(h)}))\frac{dB(h)}{dh} \quad (4)$$

To get from the first expression of the magnetic force (F_{mag}) in equation (4) to the second expression, we assume that the B-field is largely given by the external field H, $B \approx \mu_0 H$, where μ_0 is the permeability in vacuum. In other words, the magnetization M of the sample is negligible compared to the external field. Here, μ_{np} is the magnetic dipole moment of a nanoparticle of diameter D in Am^2, $\frac{dB(h)}{dh}$ is the difference in field strength over height, M_b the saturation of the bulk magnetic material in A/m, and $L(\xi\,(\mathrm{B(h)}))$ the Langevin function as a function of the B-Field, which itself is a function of the distance h from the magnet. The Langevin function describes the magnetic polarization of the superparamagnetic nanoparticles due to the B-field. Since the nanoparticles experience Brownian rotation, the orientation of the nanoparticles is determined by the interplay between the magnetic energy and the thermal energy of the nanoparticles. In the high temperature limit the nanoparticles will not have a preferred dipole orientation, whereas in the strong field limit the nanoparticles are oriented in the direction of the magnetic field (Boon 2018). The Langevin function is given by: $L(\xi\,(B)) = \coth(\xi) - 1/\xi$, where $\xi(B) = \frac{\pi D^3\, M_b\, B(h)}{6 k_b T}$ (Berret et al. 2007).

The diffusive force (F_{dif}) results from differences in the vertical direction in the osmotic pressure exerted by the nanoparticles. The assumption is made that only the suspended superparamagnetic nanoparticles move due to the presence of the magnet and the solvent (water) molecules to be not affected by the presence of the magnet. Due to the osmotic pressure, superparamagnetic nanoparticles will preferentially move from slabs with a high superparamagnetic nanoparticle concentration to slabs with a low superparamagnetic nanoparticle concentration. If we assume that the superparamagnetic nanoparticles behave ideally and therefore do not interact with each other; the osmotic pressure can be given by Van 't Hoff's law:

$$\Pi = \frac{n\,R\,T}{V} = C\,R\,T = c\,k\,T \quad (5)$$

Here n is the amount of suspended nanoparticles in moles, R the gas constant, T the absolute temperature, V the volume and C displays the number concentration;

dividing C with Avogadro's number ($c = \frac{C}{N_A}$) gives the molar concentration. From equation (5), the vertical gradient in osmotic pressure than becomes:

$$\frac{\mathrm{d}\Pi}{\mathrm{d}h} = \frac{\mathrm{dc}}{\mathrm{d}h}\, k\, T \tag{6}$$

Here, $\frac{\mathrm{dc}}{\mathrm{d}h}$ is the concentration gradient; k the Boltzmann constant (1.38064852e^{-23} m^2 kg s^{-2} K^{-1}) and T the absolute temperature in Kelvin (Philipse 2011). F_{dif}, the diffusive or chemical force on a nanoparticle than becomes:

$$F_{dif} = \frac{d\Pi\,\mathrm{A}}{n} = \frac{d\Pi}{dh}\frac{V}{n} = \frac{\mathrm{dc}}{\mathrm{d}h}\frac{k\,T}{\mathrm{c}} \tag{7}$$

Here, $\frac{\mathrm{d}\Pi}{\mathrm{dh}}$ is the change of osmotic pressure over height in Pa/m given by equation (6), V the volume of the slab in m^3, n the number of suspended nanoparticles in moles, k the Boltzmann constant, T the absolute temperature in Kelvin and c the molar concentration.

At equilibrium the concentration of nanoparticles decreases when one goes from bottom to top, therefore the diffusive force is opposed to the gravitational and magnetic force. Filling in (3), (4) and (7) into the force balance equation (2) gives equation (8) with which the concentration of all slabs can be calculated, see Appendix I for more details.

$$V\, M_b\, L\left(\xi\left(\mathrm{B(h)}\right)\right)\frac{dB(h)}{dh} + \frac{dc}{dh}\frac{kT}{c} + \left(\rho_{np} - \rho_w\right)Vg = 0 \tag{8}$$

The Equilibrium profiles obtained in this model can now be plotted and compared to the dynamic model (II) and the experiments, which is described in the Results section.

2.2 Dynamic Model

The calculations for the dynamic model (II) are similar to the sedimentation-diffusion equilibrium model (I) in that at every point in time all forces acting on the superparamagnetic nanoparticles are in equilibrium. The main difference in model (II) is that a fourth force has to be added to the force equilibrium equation (2), namely the drag force F_{drag}. This results in equation (9), which holds for all slabs at all points in time.

$$F_{drag} + F_{mag} + F_{dif} + F_{grav} = 0 \tag{9}$$

The drag force exerted on spherical particles with a small Reynolds number (i.e. particles experiencing laminar flow) in a viscous fluid can be equated to Stokes' law (Cole et al. 2008, Planken et al. 2009). This force is displayed in equation (10), where R is the hydrodynamic radius of the superparamagnetic nanoparticles, η the viscosity of the solvent (water), and v the velocity of sedimentation (directed downwards) of the superparamagnetic nanoparticles. This force is negative, since it is directed in the direction opposite to F_{grav} and F_{mag}.

$$F_{drag} = -6\,\pi\,\eta\,R\,v \tag{10}$$

Writing out all the forces at every time step in model II yields:

$$6\,\pi\,\eta\,R\,v + \frac{4}{3}\pi R^3 M_{sat}\,L(B)\frac{dB}{dh} + \frac{dc}{dh}\frac{kT}{c} + \left(\rho_{np} - \rho_w\right)\frac{4}{3}\pi R^3 g = 0 \tag{11}$$

Initially, the nanoparticles sediment in the regime where $\frac{dc}{dh} \approx 0$ and $v \neq 0$:

$$v = \frac{4\,(\rho_{np} - \rho_w)\,R^2 + 4M_b\,R^2\,L(\xi\,(\mathrm{B}))\frac{dB}{dh} + 3\frac{dc\,kT}{dh\,cR}}{18\eta} \tag{12}$$

The density of superparamagnetic maghemite nanoparticles is given by: $\rho_{np} = 4900\,\mathrm{kg/m^3}$, the density of water is given by $\rho_w = 998\,\mathrm{kg/m^3}$, $\mathrm{T} = 293\,K$, and $B(h)$ follows equation (1). Obtaining the equilibrium distribution with model II is done by taking a dynamic approach in which the sedimentation process is divided into equal timesteps, the final concentration profile can then be taken to be equal to the equilibrium profile, since in the last steps the concentration profile does not change that much anymore. At equilibrium $v = 0$, then the force balance simplifies to:

$$\frac{4}{3}\pi R^3 M_{sat}\, L(\xi\,(\mathrm{B}))\frac{dB}{dh}+\frac{dc}{dh}\frac{kT}{c} + (\rho_{np}-\rho_w)\frac{4}{3}\pi R^3 g = 0 \tag{13}$$

This equation is equivalent to the force balance equation in model I. The degree with which the final profile in model II will agree with the equilibrium profile obtained from model II depends on the number of time steps taken and the size of the individual time steps. This is due to the way in which the concentration profiles in time are calculated. The change of the concentration profile is calculated iteratively in a two-step process. Firstly, the velocity with which the nanoparticles sediment at the start of a time step is calculated from the force balance. Secondly, the concentration of the new profile is calculated from the distance the nanoparticles will travel in that time step if the velocity remains constant for the entirety of the time step. Thereby it is assumed that the diffusive force does not change during the duration of the time step. However, this is not the case in the experimental situation. The concentration profile changes continuously and not discretely, as is calculated through this model. Also, for the entirety of the first time step the diffusive force is taken to be zero. A more continuous behavior can be approximated if the time steps are taken to be as small as possible, for instance 100 ms. The concentration profiles calculated with model II should have slightly higher concentrations at the bottom half of the liquid column and slightly lower concentrations at the top half of the liquid column, as the diffusive force is underestimated (for the complete description of the computation code see appendix VI through VIII)

3: Experimental Methods

In this chapter the experimental methods that have been applied to evaluate sedimentation under a magnetic field are described. Firstly, the optical setup to monitor magnetic sedimentation is described with which the dynamic development of the concentration profiles in time towards sedimentation-diffusion equilibrium is analyzed. Secondly, the Chemical Synthesis of the superparamagnetic maghemite nanoparticles suspended in water is given. Thirdly, the size fractionation is evaluated, which was done to obtain fractions with different average particle diameters. Fourthly, the vibrating-sample magnetometry (VSM) analysis technique is described. This is followed by a description of the Analytical Ultracentrifugation (AUC) technique. Finally, Transmission Electron Microscopy (TEM) is described.

3.1 Optical Setup to Monitor Magnetic Sedimentation

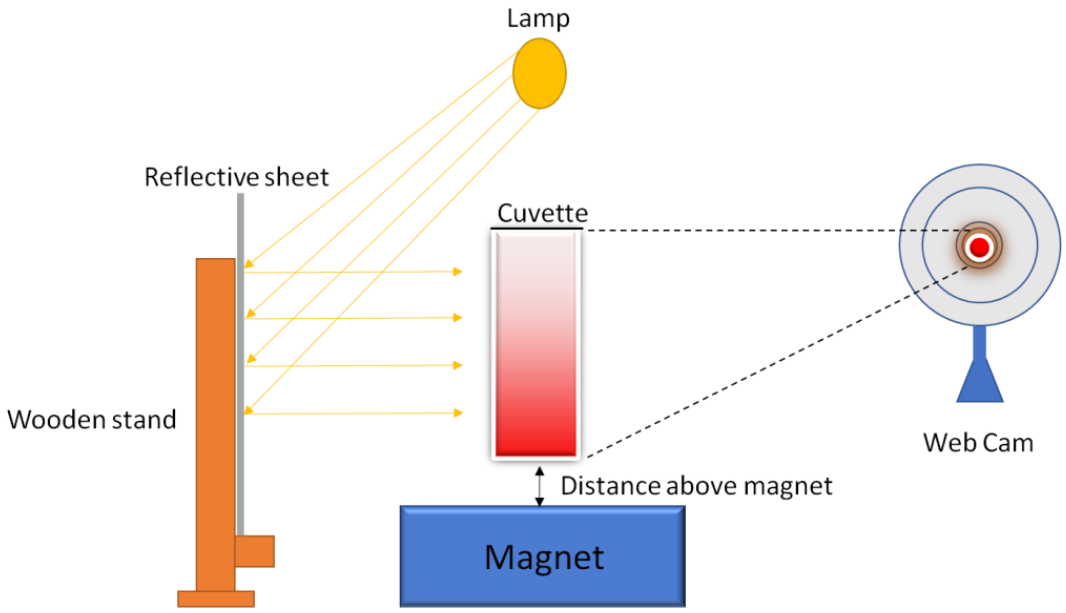


Figure 5: Setup magnetic sedimentation experiment

In the magnetic sedimentation experiment, sedimentation in time of superparamagnetic nanoparticles can be examined in the presence of a magnetic field. In this setup light is emitted by a 270 K, 806 lumen LED (Philips Lighting) and reflected by a white reflective piece of paper (A4), which was held in place by a wooden stand. The top of the cuvette is covered, which prevents light passing through the fluid sample from the top. The reflected light passes through a superparamagnetic fluid sample (in a cuvette with an optical pathlength of 2 mm), which is placed a distance of 1.0 cm above a circular neodymium magnet with a

radius of 2.25 cm, a thickness of 3.0 cm and an internal remanent magnetization of 1.35 T (obtained from SuperMagnete, Gottmadingen, Germany). The transmitted light is detected by a logitech HD 720p web cam, yielding PNG files of the development of the concentration profile through time. The setup is displayed in figure 5.

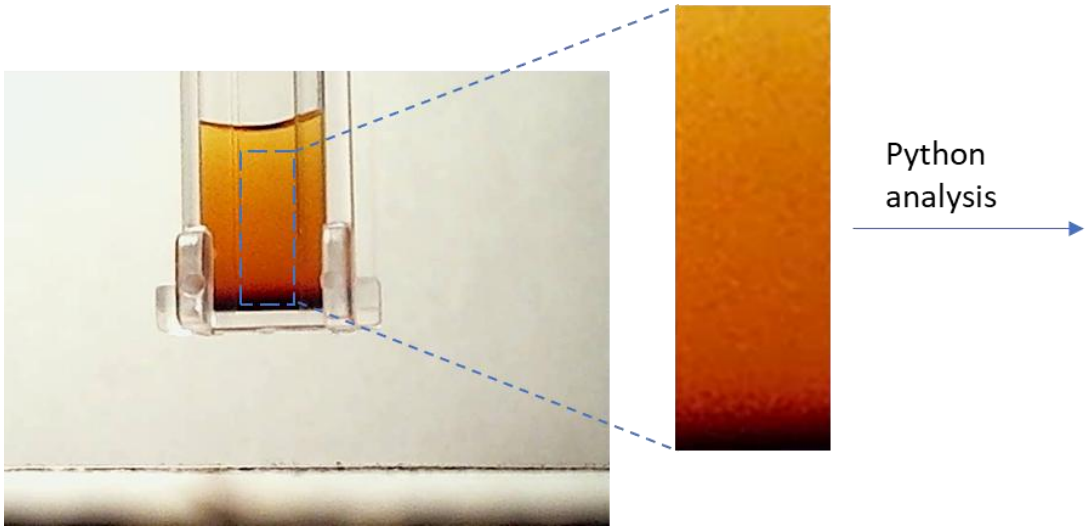


Figure 6: Cropping the desired part for sedimentation analysis of a PNG image file obtained with the optical setup

The pictures are obtained by using the Debut Video Capture Software from NCH Software, the settings used in evaluating the samples are given in Appendix I. Before the concentration profiles can be obtained, a cropped file containing only the pixels necessary for the determination of nanoparticle settling needs to be created. This is done via the Icrop function in the python code shown in appendices VII and VIII and has been shown visually in figure 6. Where on the left a PNG file obtained from the optical experiment is shown with an enlarged picture of the cropped liquid column to its right.

Solvent backflow, which always takes place in a closed vessel where particles settle, (Planken et al. 2012) has been neglected in the calculations, which would give a retarding effect to sedimentation. However, it must be acknowledged that in actuality the velocity of sedimentation behaves very much differently than in this simplified model. Due to conservation of volume flux, the downward movement of the nanoparticles is accompanied by an upward flux of fluid. Moreover, the settling nanoparticles drag down adjoining solvent, creating an inaccessible (for the center of nanoparticles) shell around the nanoparticles, which causes an upward flux of solvent that is equal to the downward flux of settling adjoining solvent nanoparticles (Batchelor 1972).

The effect of backflow on sedimentation is felt mostly close to the walls of the containers (Dogic et al. 2000). Therefore, the crop has been chosen to not include the sides with a width of approximately 4 mm. Also a couple of pixels at the bottom of the fluid column have not been included in the crop, this due to the reflection of light that was observed occurring here. The very top of the fluid column was also not included in the cropped image, since due to evaporation of solvent images taken at later moments in time showed a decrease in the height of the fluid column compared to earlier images. Therefore, the crop has to be chosen such that the top will still be part of the fluid column for images taken at later moments in time.

The PNG files obtained in the way shown above are given in values of the RGB color model. In the further analysis only the red values will be evaluated to obtain the concentration profiles. Both the green and blue values can become close to zero or zero for very dark colors and are therefore not suited for obtaining concentration profiles (the log_{10} of zero is undefined). Also, blue and green light are scattered more than red light due to scattering being dependent on the inverse of the wavelength to the fourth power ${}^{1}/_{\lambda}{}^{4}$ (Seinfeld 2016). Therefore, using red light exclusively in concentration profile determination results in a greater dynamic range (compared to using blue or green light), since relatively more light will be transmitted as compared to scattered (than would be the case for green and blue light). Obtaining the concentration values from the red values is done by filling in the Lambert-Beer law (Swinehart 1962), shown below:

$$A = \varepsilon\, c\, d = -log_{10} \frac{I(h)}{I_0(h)} \tag{14}$$

The symbol A represents the absorbance, ε the molar extinction coefficient of the superparamagnetic fluid, c the molar concentration, d the optical pathlength (distance the light travels through the superparamagnetic fluid), the symbol I the intensity of the outgoing light (obtained by the measurement) and I_0 the intensity of the ingoing light, which is determined by a single measurement of the solvent without the suspended superparamagnetic nanoparticles.

At sedimentation-diffusion equilibrium the concentration distribution of suspended nanoparticles in a gravitational field can be determined by the barometric profile, i.e. the concentration decreases exponentially from the bottom of the fluid column to the top as a consequence of osmotic pressure following van Hoff's law in a gravitational field (Piazza et al. 2012). Suspended superparamagnetic nanoparticles in a magnetic field will not follow the barometric profile, since the strength of the magnetic field decreases according to equation (1)

with an increase of the distance between the nanoparticles and the magnet (see chapter 1.3).

3.2 Chemical Synthesis

Superparamagnetic nitrate-stabilized maghemite nanoparticles suspended in water are obtained via a synthesis according to Massart (1981). First, magnetite is formed from the coprecipitation of iron(II) and iron(III) salts with NaOH solution. For this 3.46 g of $FeCl_3 \cdot 6H_2O$ and 1.28 g of $FeCl_2 \cdot 4H_2O$ in 80 mL of demiwater, and 2.0 g NaOH is dissolved in 20 mL demiwater. Whilst stirring, NaOH solution is added to the iron salt solution; magnetite is formed immediately. Stirring is continued for 3 more minutes. Whilst holding a handheld magnet under the solution, the supernatant is poured off. This is followed by redispersing the magnetite in 16 mL 2M HNO_3 and stirring for 5 minutes. A solution of 3.40 g of $Fe(NO_3)_3 \cdot 9H_2O$ in 24 mL demiwater is added. Whilst stirring, the solution is refluxed at 90°C for 1 hour. The magnetite oxidizes to maghemite. After washing twice with 40 mL of 2M HNO_3, the desired maghemite nanoparticles are obtained, which are redispersed in demiwater.

3.3 Size Fractionation

From the synthesis according to Massart (1981) a rather polydisperse system is obtained. In order to obtain fractions with a different mean diameter for the nanoparticles a size sorting procedure according to Lefebure et al. is performed to obtain fractions bbb, btb and tbt (Lefebure et al. 1998, Boon 2018). It was observed that fraction bbb contained nanoparticles with the highest mean diameter of the three fractions, fraction tbt contained nanoparticles with the lowest mean diameter, btb's mean diameter fell between the mean diameter of the other two fractions (see chapter 4.1). In this section the procedure of size fractionation to obtain these three fractions is described.

Firstly, 10 mL of 65% nitric acid was diluted with 75 mL demiwater, forming a 85 mL stock solution. 2 mL of this stock solution was then added to 10.5 mL of the superparamagnetic fluid obtained from the synthesis. A dark brown precipitate is formed. A magnet was held for approximately 5 minutes under the solution, followed by a decantation, where the top fraction (t) resulted in a solution of 8.5 mL and the bottom fraction (b) contained 4 mL. Due to the charges of the maghemite nanoparticles being screened by the electrolyte solution, the nanoparticles clustered. The bottom fraction (b) obtained in the previous decantation step was redispersed with demiwater to a final solution of 15 mL. To the top fraction 15 mL demiwater

was added together with 2.5 mL of the nitric acid stock. After having kept t on top of a magnet for 5 minutes, decanting resulted in a top fraction of 17 mL and a bottom fraction (tb) of 8 mL. The top fraction was discarded and 12 mL of demiwater and 5 mL of nitric acid stock solution were added to tb. After 5 minutes on top of a magnet, decanting tb yielded a top fraction (tbt) of 17 mL and the bottom fraction (tbb) containing 8 mL. From both fractions the liquid was decanted, leaving only the precipitate. The tbb fraction contained little nanoparticles and was discarded.

4 mL of nitric acid stock was added to the bottom fraction from the first decantation. Consecutively it was placed on a magnet and was decanted into two fractions, the top fraction (bt) contained 15 mL and the bottom fraction 4 mL (bb). The bt fraction was redispersed with demiwater to 20 mL. To the bb fraction 11 mL demiwater and 4 mL nitric acid stock solution were added. This fraction was decanted, yielding a top fraction (bbt) and a bottom fraction (bbb). From both fractions all the liquid was decanted and only the precipitate was kept. The bbt fraction contained little precipitate and was discarded. To bt 5 mL nitric acid was added. Again, bt was held on top of a magnet for 5 minutes, followed by decantation in which the btb fraction contained 8 mL and the btt fraction contained 17 mL. After the decanting process only for btb sufficient precipitate was obtained, the remaining btt precipitate was discarded. All three viable fractions (tbt, btb and bbb) were dispersed in demiwater and stored for further analysis. For the complete fractionation scheme see figure 7.

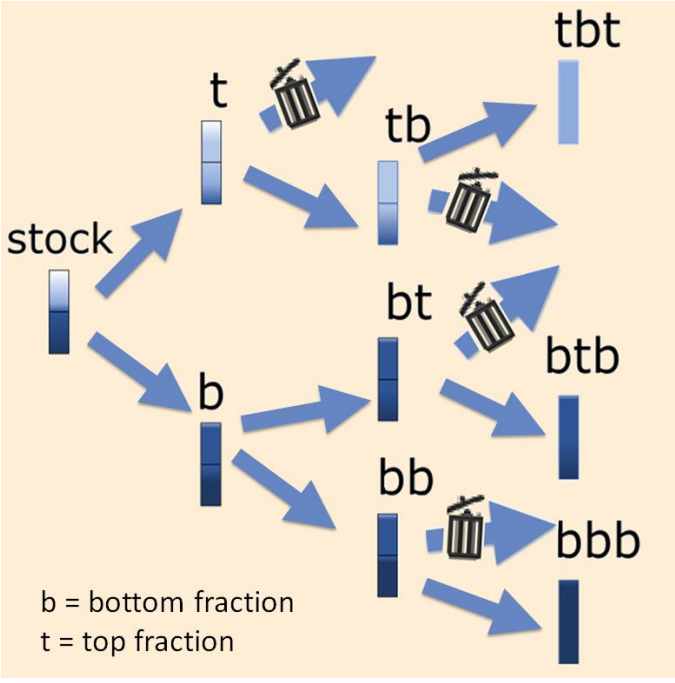


Figure 7: Fractionation protocol. The discarded fractions are indicated with a trash can symbol. Only the tbt, btb and bbb fractions were kept.

3.4 Vibrating-Sample Magnetometry

An EZ-9 vibrating sample magnetometer (VSM) from MicroSense was used to measure the magnetization of the superparamagnetic fluid samples in a magnetic field according to Van Silfhout et al. (2019). The superparamagnetic fluid sample was placed in a uniform magnetic field, resulting in an induced magnetization in the sample. In the VSM, the sample was vibrated sinusoidally perpendicular to the magnetic field, such that it experienced an oscillating external magnetic field and the magnetization of the sample according to the applied magnetic field was measured (Foner 1959, Foner 1981). From the plot of the magnetization in a magnetic field, the magnetic saturation and the initial magnetic susceptibility are determined of the samples (stock, bbb, btb and tbt obtained from the fractionation). A Langevin function was also plotted to the magnetization to obtain a measure for the nanoparticle size.

3.5 Analytical Ultracentrifugation

Analytical Ultracentrifugation (AUC) is a high-resolution technique in determining the sedimentation velocity and sedimentation equilibrium of colloidal suspensions. Specifically, in this research the superparamagnetic ferrofluid was analyzed using a Beckham Coulter Proeteome Lab XL-A/XL-I analytical ultracentrifuge. Analytical ultracentrifugation has been applied a lot since the beginning of the last century in the detection and quantitative analysis of biomacromolecules (Cole et al. 2008). Yet it is also very useful in the analysis of other colloidal systems (Planken et al. 2012}). Three optical systems are available for the analytical ultracentrifuge (absorbance, interference and fluorescence) (Cole et al. 2008). For this research only the absorbance mode is of interest. In this mode the amount of light absorbed by the superparamagnetic fluid in the ultracentrifuge is measured, a measure for the concentration of colloidal particles (see equation 14). 3mm epon centerpieces were used, which were filled with the superparamagnetic fluid. Quartz windows were used, since the AUC measurement was run in absorbance mode.

Two complementary views of dispersion behavior are available for AUC. Via sedimentation velocity (SV) hydrodynamic properties such as size and shape of the nanoparticles can be obtained and via sedimentation-diffusion equilibrium (SE) thermodynamic properties are provided about the suspension (i.e. it determines the concentration distribution after equilibrium is reached). The requirements to keep in mind for successfully making measurements using this technique are threefold. Firstly, the sample to be analyzed has to be distinguishable optically from the solvent. Secondly, the sample has to sediment or float with a detectible rate under a centrifugal field that is experimentally achievable. Finally, the sample should be chemically compatible with the sample cell and the optical systems (i.e. it should

not react with the experimental setup) (Cole et al. 2008). The results that can be obtained from AUC depend on the purity of the sample. Among others AUC can be used to measure the molecular weight, association state, equilibrium constants for reversibly interacting systems and hydrodynamic shape information (Planken et al. 2012, Cole et al. 2008).

Size dependent light scattering can take place if large agglomerations of nanoparticles occur, i.e. Mie scattering of colloids can potentially cause further complications if measuring in absorbance mode (XL-A), depending on the size of the colloids in the superparamagnetic fluid. This light-dependent scattering needs to be corrected for. Mie correction can potentially increase uncertainties or artifacts in the relatively small particle size range (in Mie scattering the amount of scattering of the particle depends on the direction of the incident light). Mie scattering will only take place in particles with sizes above a tenth of the wavelength (λ) of the incident light (Bohren et al. 2008, Costello et al. 2007). In XL-A the wavelength of the incident light was between 360 and 500 nm, which makes mie scattering relevant for particles of diameters above 36-50 nm. Yet, since the nanoparticles in the superparamagnetic fluid analyzed in this research have been shown (from the TEM measurements) to be in the size range of roughly 1-20 nm in diameter. In this limiting case, where the nanoparticles are much smaller than λ, the Rayleigh scattering model can safely be used without losing accuracy. Only when larger clusters form will Mie scattering effects become significant (Seinfeld 2016).

It is known that magnetic colloids display spontaneous structure formation, even in absence of a magnetic field (de Gennes et al. 1970, Planken et al. 2009). The tendency to form agglomerates will be reduced when the suspension is sufficiently dilute. This is the case since agglomeration under a magnetic field is governed by van der Waals attractions (Rosensweig 1982). In analyzing the absorption profiles obtained from AUC, Mie scattering will be neglected. The data obtained from AUC was analyzed with the Sedfit software package developed by Dr. P. Schuck (2000).

Similar to the optical setup the Absorbance is measured by the analytical ultracentrifuge. It followed from the Lambert-Beer law that absorbance is proportional to concentration. By obtaining a measure for the change of concentration in time and using the Lamm equation, the velocity of sedimentation can be obtained (Ralston 1993). The Lamm equation describes diffusion and sedimentation under ultracentrifugation of a solute. It is given below:

$$\frac{\partial c}{\partial t} = D\left[\left(\frac{\partial^2 c}{\partial r^2} + \frac{1}{r}\left(\frac{\partial c}{\partial r}\right)\right] - s\omega^2\ \left[r\left(\frac{\partial c}{\partial r}\right) + 2c\right]\right. \qquad (15)$$

Here c represents the solute concentration, $\frac{\partial c}{\partial t}$ the change in concentration over time, r the distance of the settling particle from the center of rotation, D the diffusion constant which depends on the effective size of the nanoparticles, $s = \frac{u}{\omega^2 r}$ is the sedimentation coefficient, where u is the velocity with which the nanoparticles sediment and ω the angular velocity of the rotor (Mazumdar 1999, Ralston 1993).

The Lamm equation cannot be solved analytically, yet approximate numerical solutions of the Lamm equation can be used to describe the AUC magnetic sedimentation data (Cao et al. 2005). The Lamm equation has been shown to be viable for modeling polydisperse systems, provided the particle density is known (Schuck 2000).

Analytical Ultracentrifugation is a good technique for analyzing sedimentation in a gravitational field and thereby also gives clues on how the fluid might behave in a magnetic field. Here the gravitational acceleration (g) is replaced by ωr^2, with ω the angular velocity of the rotor of the analytical ultracentrifuge and r the distance of the sample from the axis of rotation (Luigjes et al. 2012). Yet caution needs to be taken with the AUC results, since they do not show how superparamagnetic fluids behave in a magnetic field, due to different forces being present in a magnetic field. T magnetic field strength decreases differently with height compared to gravitational sedimentation (since the magnetic field strength follows equation (1)).

3.6 Transmission Electron Microscopy

In Transmission Electron Microscopy (TEM) is a powerful imaging technique, in which a beam of electrons is transmitted through a sample. The transmission of electrons is measured and an image of the sample is obtained. TEM is capable of achieving significant higher resolution than conventional light microscopes, due to the smaller de Broglie wavelengths of electrons compared to the wavelengths of light (Williams 2009). Since the resolution of TEM is in the sub-nanometer range it can be used in determining the size of maghemite nanoparticles in superparamagnetic fluids. 6 images were made for all the fractions obtained from the fractionation (see 3.3) with the Philips Tecnai 10 Transmission Electron Microscope. The clearest two TEM images were chosen for every fraction and the diameter of roughly 120 nanoparticles was measured for every image using the imageJ image processing program.

3.7 Dynamic Light Scattering

Another way in which the size of the nanoparticle can be determined is by using Dynamic Light Scattering (Erné 2018). The size distributions were determined in this way with the Zetasizer Nano-ZS from Malvern Instruments. The measurements were conducted at 25 °C. The samples were analyzed in DTS1070 cuvettes from Malvern Instruments.

4: Results

In this chapter the experimental results obtained from the different techniques are being discussed. Firstly, the characterization of the nanoparticles is elaborated upon. The size of the nanoparticles was determined with the help of the transmission electron microscopy (TEM) images. The magnetic saturation of the nanoparticles is shown from vibrating sample magnetometry (VSM). Analytical Ultracentrifugation experiments are also discussed. Finally, both the results obtained from the sedimentation-diffusion model (I) and the dynamic model (II) for monodisperse and polydisperse superparamagnetic fluids respectively are compared to the experimental results from the optical setup.

4.1 Particle size determination

To obtain a measure of the size of the nanoparticles transmission electron microscopy (TEM) was conducted. From the TEM images large clusters of maghemite nanoparticles can be seen, regardless whether the nanoparticle is located in the top or bottom fraction of the fractionation (see figure 7). Yet due to the nature of TEM (drying artifacts created by evaporation of solvent) this cannot be adequately taken as a proof that clusters actually exist in situ. In figure 8 TEM images for the tbt and bbb fraction are shown.

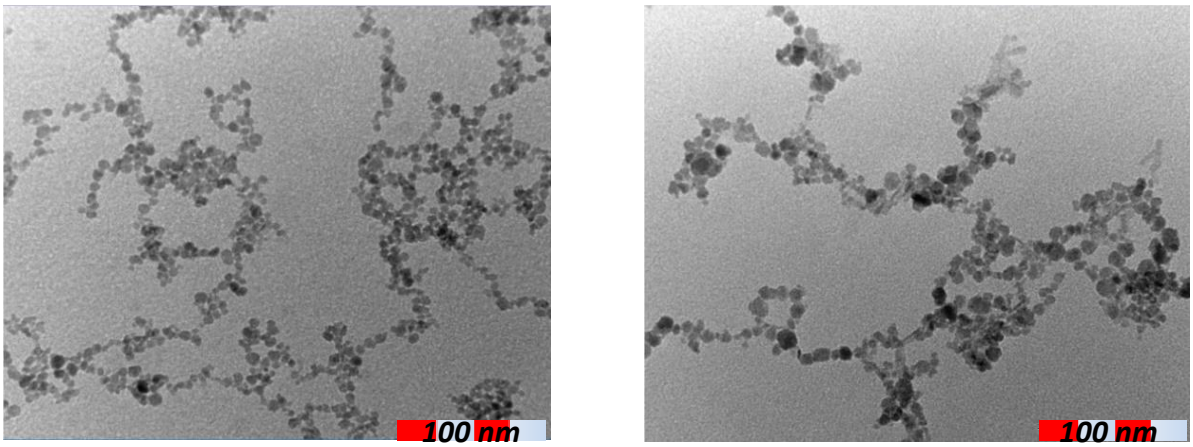


Figure 8: Left: Tbt/top fraction, right: Bbb/bottom fraction taken with TEM

The graphs below (figures 9-12) show the measured radii of the individual maghemite nanoparticles; the diameters have been sorted into bins, with the numbers indicating the starting value of the bin (so in bin 4, the radii of the maghemite nanoparticles are between 4 and 5 nm).The stock refers to the suspension before fractionation. The bbb fraction is the lowest fraction, btb the

middle fraction and tbt the top fraction obtained from the fractionation (as described in chapter 3.3 and illustrated in figure 7). In table 1 the median diameter, mean diameter and standard deviation of the fractions are shown. Table 2 displays the pH of the fraction solutions that were measured using a pH meter. All the pH values measured are around 2 pH. This is much lower than the isoelectric point (IEP) of maghemite, which lies at a pH of 6.7±0.2 (Parks 1965). Therefore, the nanoparticles will be cationic (Lefebure et al. 1998).

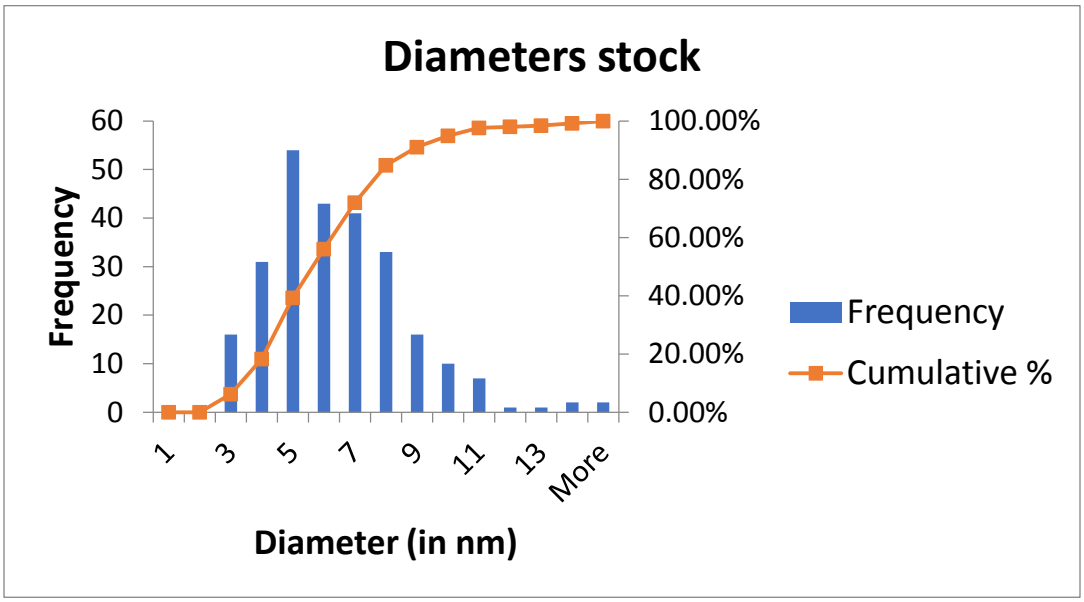


Figure 9: Diameters of nanoparticles from TEM image of the stock fraction

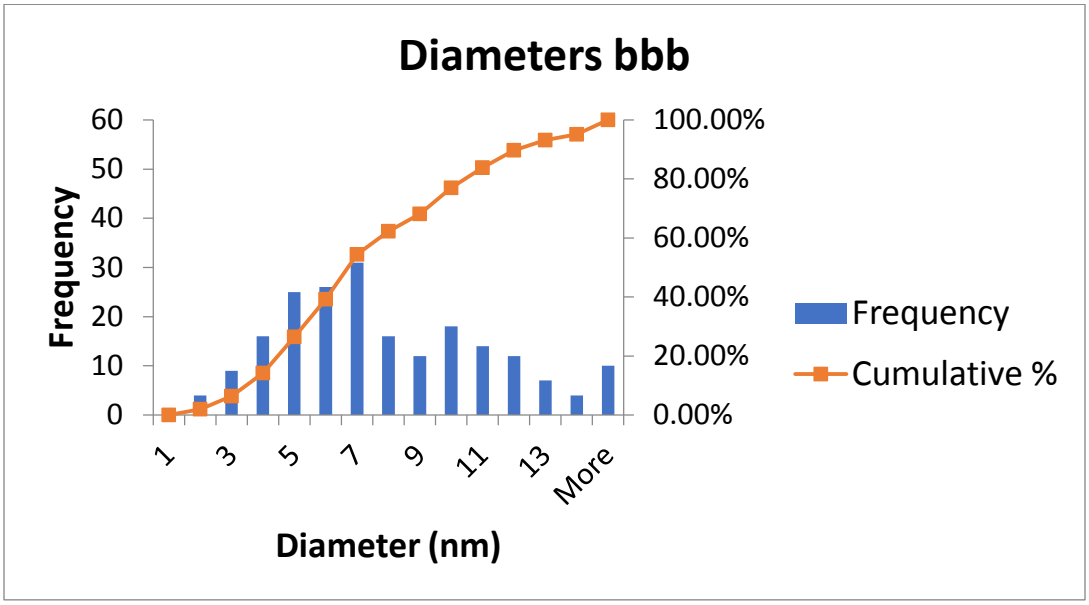


Figure 10: Diameters of nanoparticles from TEM image of the bbb/bottom fraction

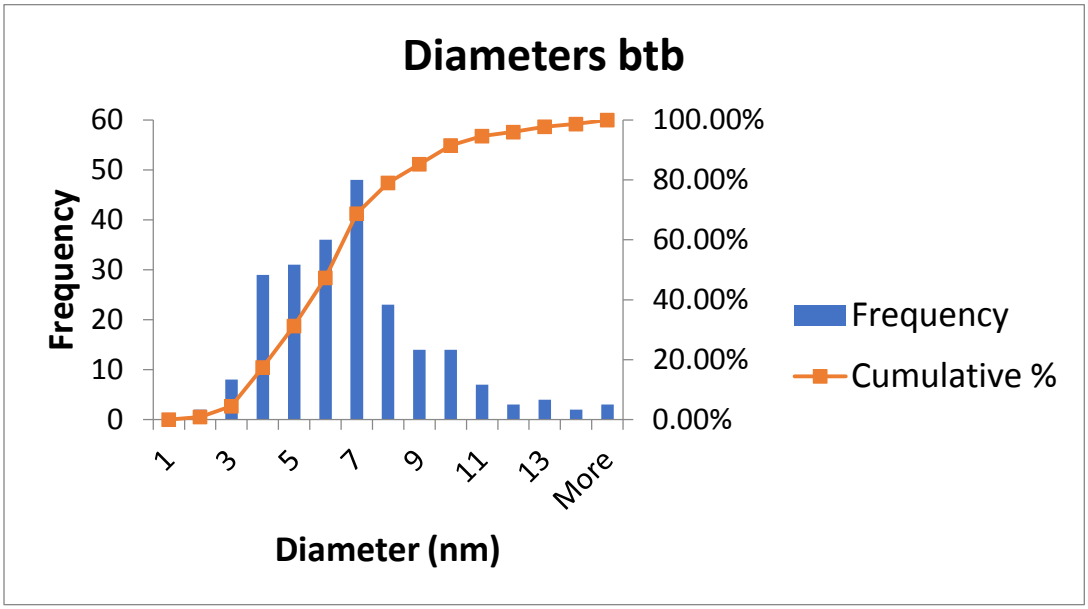


Figure 11: Diameters of nanoparticles from TEM image of the btb/middle fraction

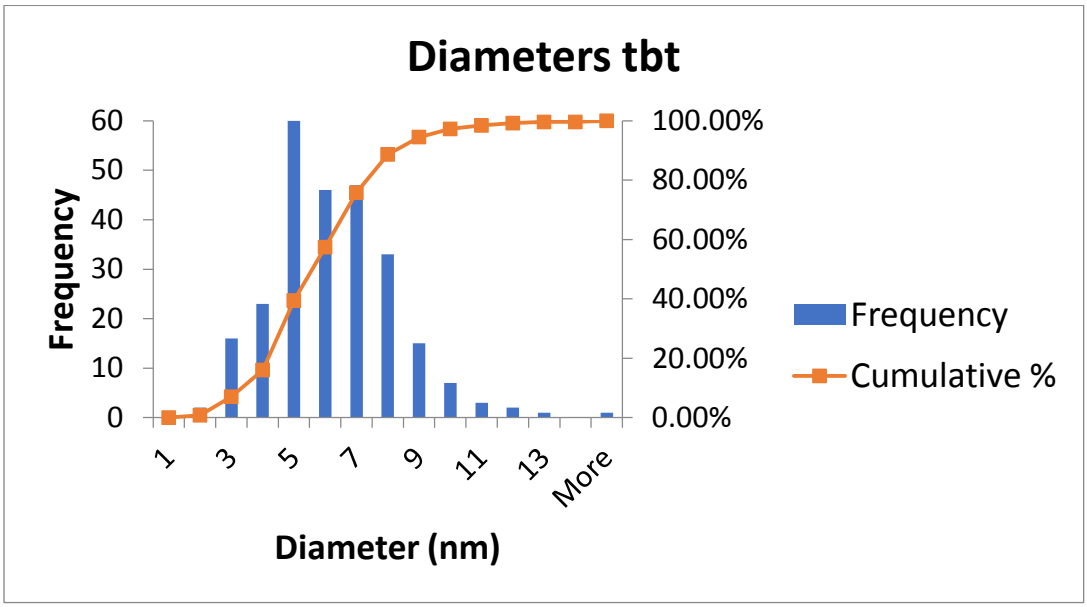


Figure 12: Diameters of nanoparticles from TEM image of the tbt/top fraction

Fraction	Median Diameter (nm)	Mean Diameter (nm)	Standard Deviation (nm)
Stock	5.5	6.0	2.3
Bbb/bottom	6.7	7.6	3.6
Btb/middle	6.1	6.4	2.8
Tbt/top	5.6	5.7	2.0

Table 1: Median Diameter, Mean Diameter and Standard Deviation of the fractions obtained via the fractionation in 3.3

Fraction	pH
Stock	2.42
Bbb/bottom	2.09
Btb/middle	1.98
Tbt/top	1.97

Table 2: pH of the fractions

In figure 13 the particle radii found in the three fractions (bbb, btb, tbt) are compared. The boxes are showing the radii between the first and third quartile and the cross the mean diameter. It can be seen that the lower fractions have a higher median and average particle radius and the higher fractions have a lower median and average particle diameter. The total distribution range is greatly reduced and the median diameter is smaller in the tbt and btb fractions compared to the bbb fraction. From this it can be concluded that despite not making the ferrofluid lose its polydispersity much, the fractionation method can be used to divide the stock into fractions with differing median diameters.

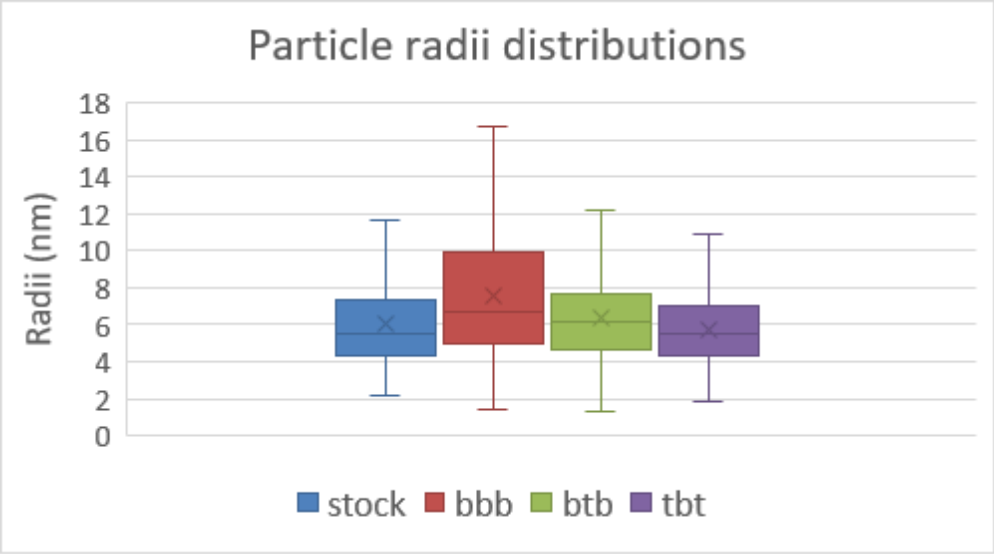


Figure 13: Particle Diameters distribution of the bbb/bottom, btb/middle and tbt/top fractions after fractionation and the stock sample

Also with the Zetasizer a measure for nanoparticle size was obtained. For all the fractions and the stock sample three measurements were conducted, what is displayed in figure 14 is the average of the three measurements for the four samples. From this it is seen again that the bottom most fractions contains the biggest particles, the middle. All samples remain polydisperse (showing a large tail in the distribution).

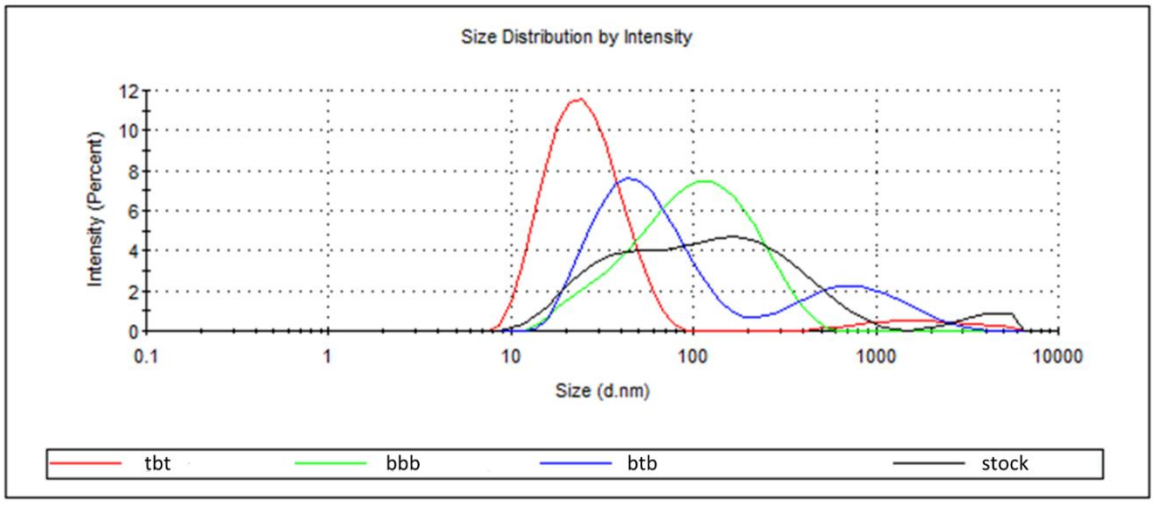


Figure 14: Radii of nanoparticles according to zetasizer for the bbb/bottom (green), btb/middle (blue), tbt/top (red) fractions and the stock sample (black)

Below the particle diameter distributions obtained via TEM for the bbb and btb fractions have been plotted next to the respective log-normal distribution that is used in the evaluation of the polydisperse models.

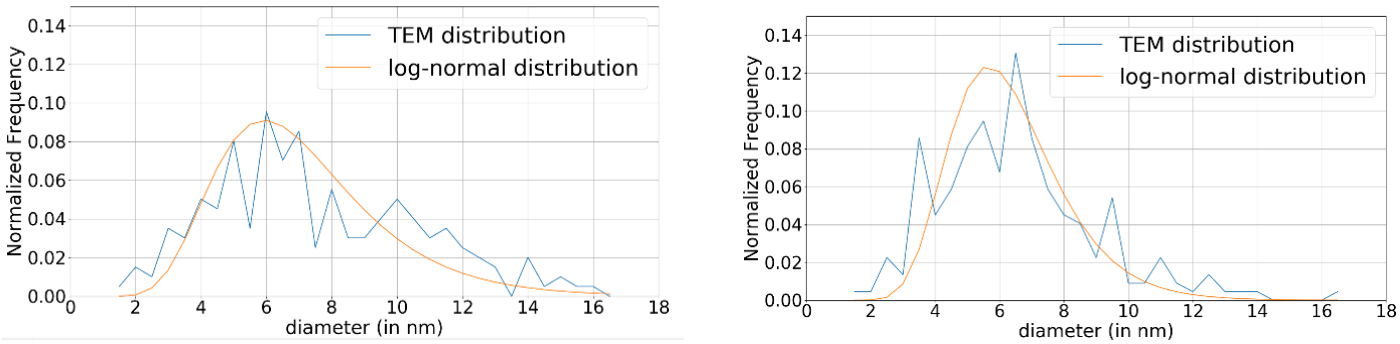


Figure 15: Left: bbb/bottom fraction TEM distribution and log normal distribution, Right: btb/middle fraction TEM distrubtion and log normal distribution

4.2 Vibrating Sample Magnetometry

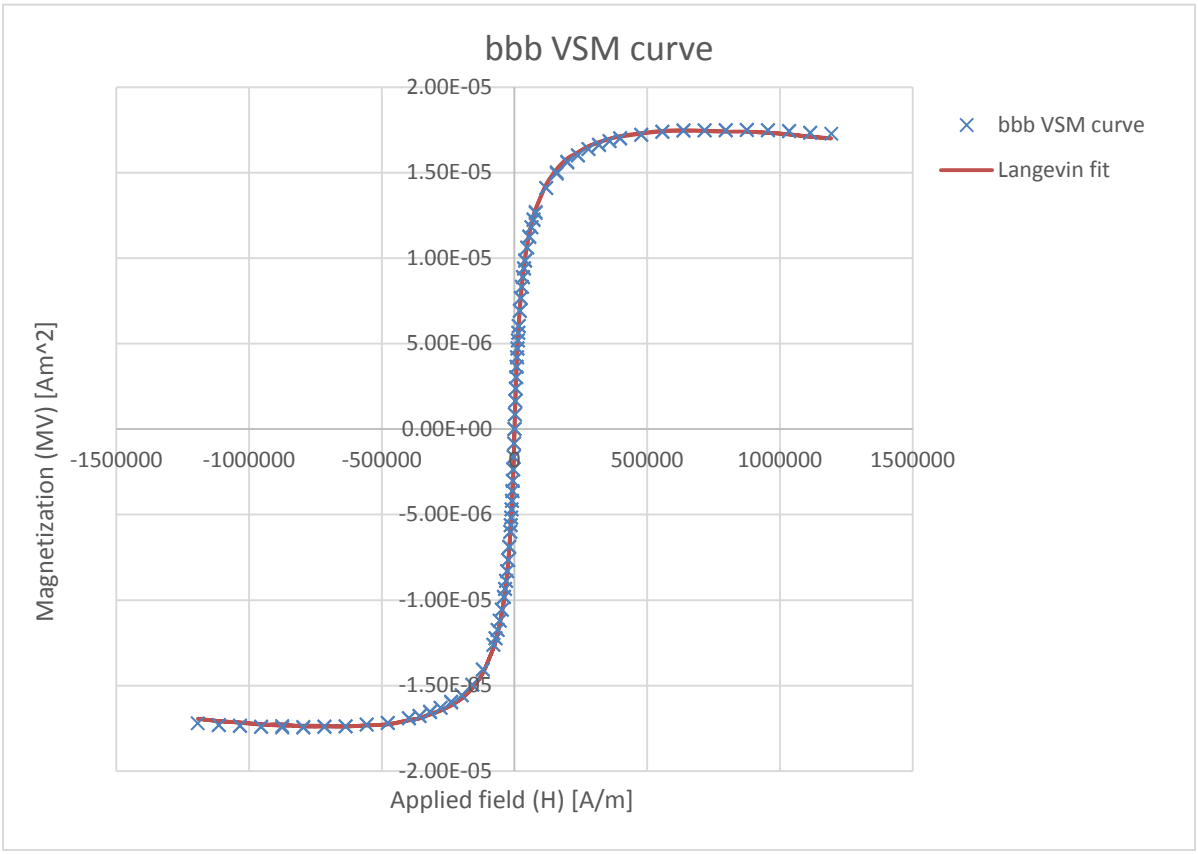
In Figure 16 the results obtained from vibrating sample magnetometer (VSM) can be seen for the bbb/bottom and btb/middle fractions and in figure 17 the VSM results for the tbt and stock fractions can be seen. It can be used to determine the magnetization under an external magnetic field. From these, it becomes clear that the magnetization saturation of the material does not differ greatly between the bbb and btb samples and being only 5.000e^{-6} Am^2 higher than the magnetization saturation of the tbt fraction. The stock sample's magnetization saturation is even lower than the bbb and btb samples, being less than a third of the magnetization saturation of those samples. In all four curves no hysteresis is seen; this is characteristic of superparamagnetism, as there will be no magnetization if the magnetic field is switched off (Marghussian 2015).

The slope of the VSM curve at the origin gives the initial magnetic susceptibility at zero field times the volume ($\chi V = {}^{MV}/_{H}$) of the superparamagnetic nanoparticles. A division of χV by the volume of the sample (100 µL) gives the magnetic susceptibility (χ). The magnetic susceptibilities are shown in table 3 for all four samples. The highest χ is the stock, with bbb/fraction having the steepest slope (highest χ) of the fractions and the lowest χ is of the tbt/top fraction. The magnetic susceptibility was calculated by determining the slope of the VSM curve at the origin.

Fraction	Initial magnetic susceptibility ($v\chi$)	Initial magnetic susceptibility (χ)
Stock	$5.34e^{-9}$	0.0534
Bbb/bottom	$5.30e^{-10}$	0.00530
Btb/middle	$2.27e^{-10}$	0.00227
Tbt/top	$9.53e^{-11}$	0.000953

Table 3: Magnetic susceptibility (χ) of the samples analyzed with vibrating sample magnetometry (VSM)

By fitting a Langevin function (discussed in chapter 2.1) to the VSM curve a measure of particle size can be obtained. Interestingly, VSM curves of the bbb, btb and tbt samples are described easily with a Langevin function were the median diameter of the particles is set to 3.8 nm (the saturation magnetization of the bulk M_b was set to 430000 A/m and the temperature 293 K). This is smaller than the median diameters determined from the TEM images. This might be caused by amorphous surface layer around the crystalline core of the nanoparticle (Ahrentorp et al. 2010, Gazeau et al. 1997), which was probably caused by the coprecipitation of iron(II) and iron(III) salts to form the magnetite in the synthesis (see chapter 3.2). The fact that the median diameter of the nanoparticles of all fractions is the same, it seems that the crystalline cores of the nanoparticles stay the same during fractionation. The fractions only differ in the size of the surface layers. Close to magnetization saturation the stock sample cannot be readily fitted to the Langevin function (with a particle diameter of 3.8nm). Also, the magnetization saturation is higher than would seem from looking only at the Langevin fit in the stock sample.



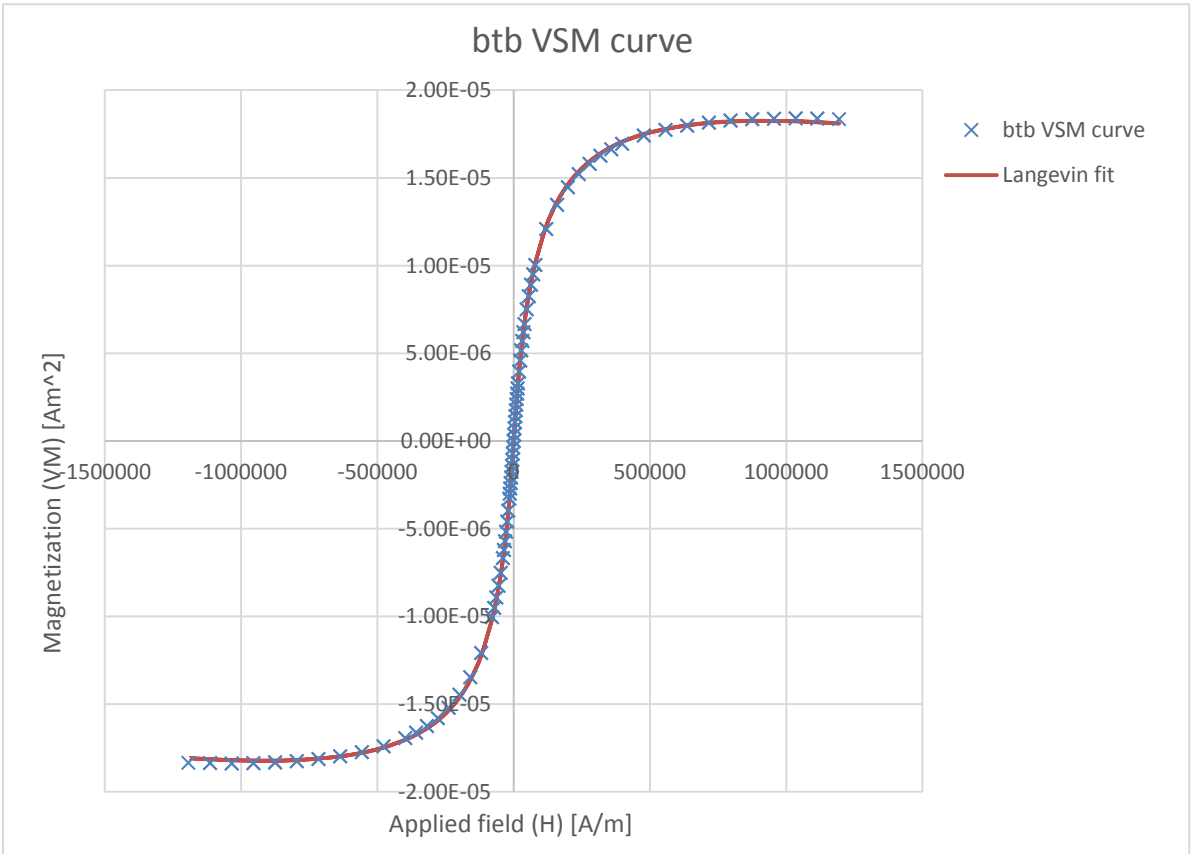
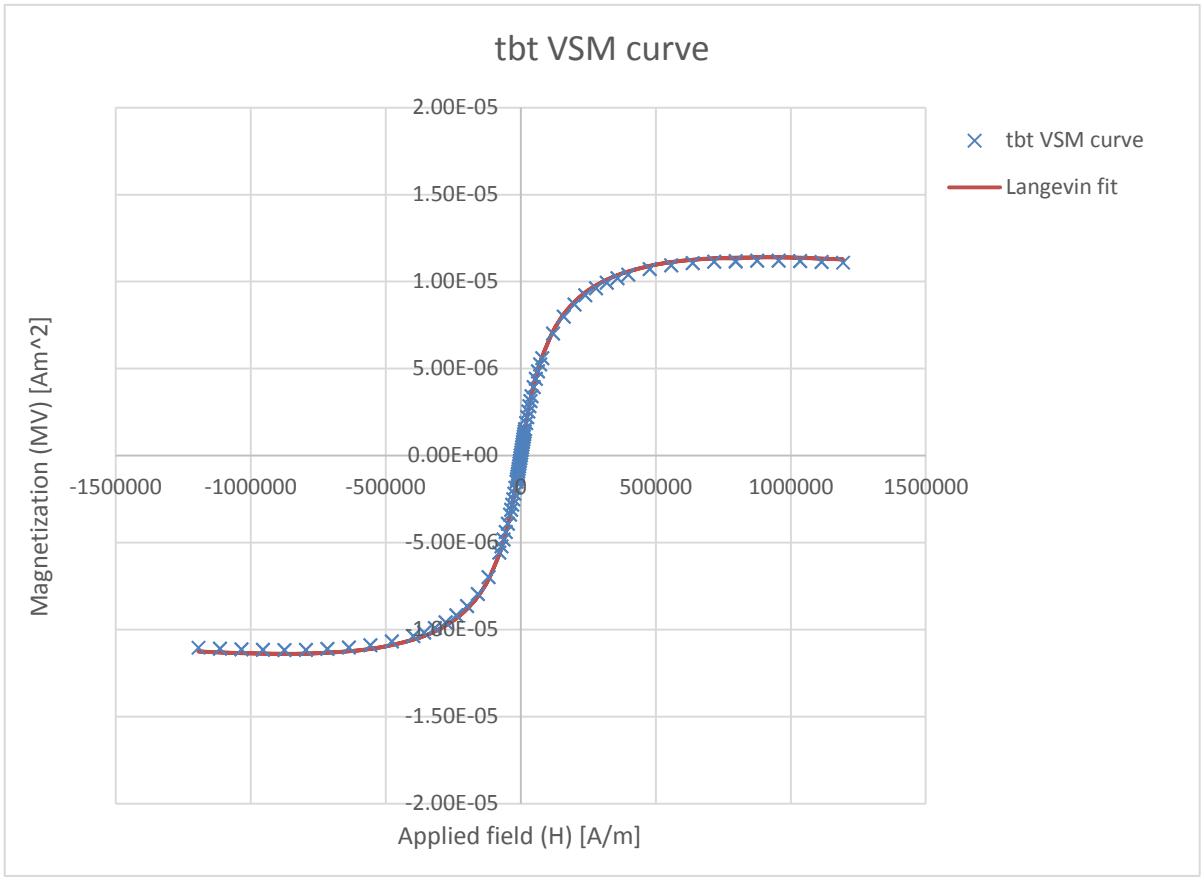


Figure 16: Top: bbb/bottom VSM curve, Bottom: btb/middle VSM curve



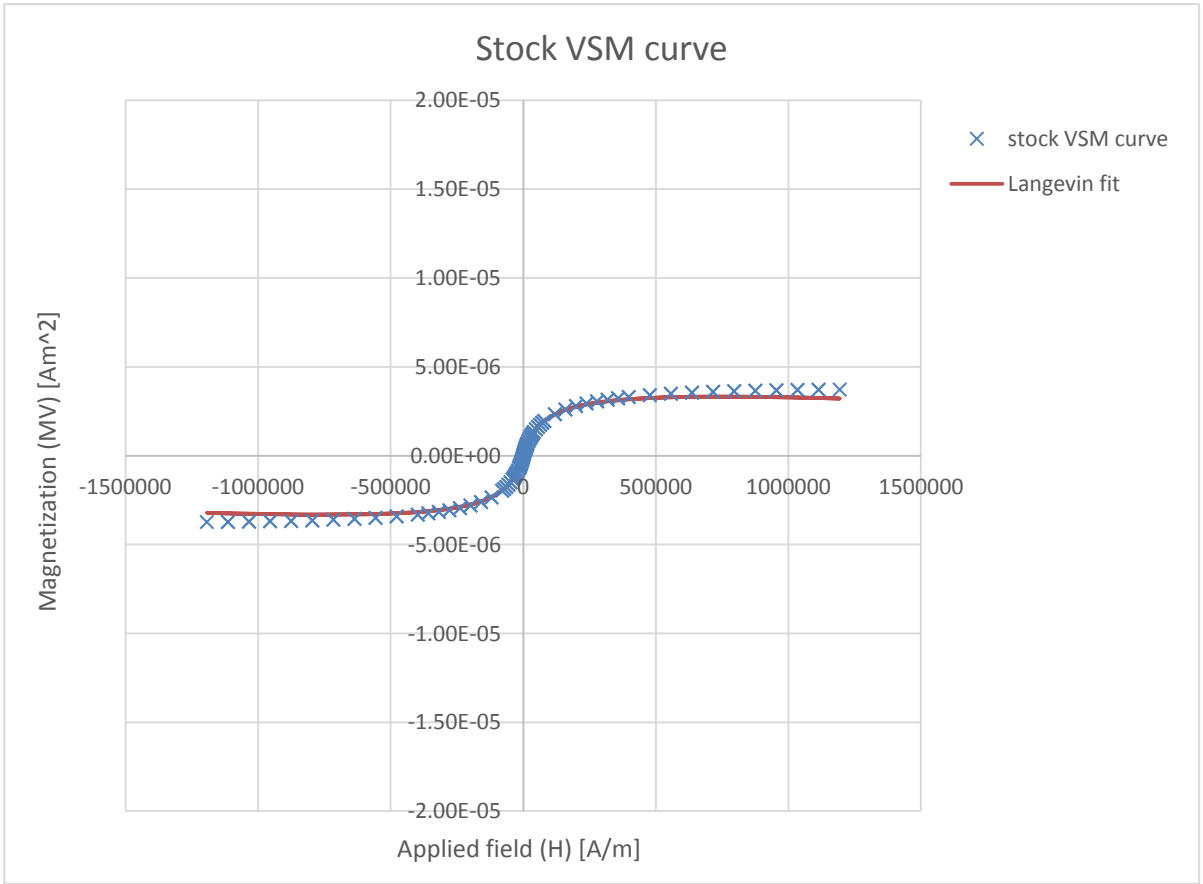


Figure 17: Top: tbt/top VSM curve, Bottom: stock VSM curve

4.3 Analytical Ultracentrifugation

Before the Analytical Ultracentrifugation (AUC) could be started the wavelength (W1) of analysis was chosen. This was done by picking a peak of the emission spectrum of the Xenon flash lamp of the AUC that corresponded with an absorbance of roughly 1.0 as determined by a UV/VIS measurement of the respective sample (Lambda 35 UV/VIS Spectrometer from Parkin Elmer). In appendix II the W1 for all samples is displayed, together with the other settings of AUC analysis.

From the single wavelength AUC analysis the continuous concentration c(s) as function of location was obtained (Absorbance). The rotational speed was set to 3000 rpm for all samples. With this rotational speed the top of the fluid column experiences a force of 593 g and the bottom a force of 724 g. The initial absorbance was set to 1 for all points. On the x-axis of the AUC graphs (figures 18, 18 and 20) the distance from the axis of rotation in centimeters is given. The time between lines is roughly 300 s. The bottom of the fluid column is at a distance of 7.2 cm from the axis of rotation and the meniscus at 5.9 cm. The initial graph (darkest blue) in bbb/bottom (figure 18), btb/middle (figure 19) and stock (figure 20) shows little transmission, only at the very top, just below the meniscus. However after initiating centrifugation the three samples diverge, the bbb fraction shows that large aggregates sink almost instantly to the bottom (second graphs is lower than first graph). For this reason, AUC might not be the right method to investigate this fraction. The stock solution (before fractionation) shows that initially slightly more nanoparticles sink to the bottom than for btb, this is an indication that the stock sample (as well as bbb) contains large agglomerates with a high sedimentation speed.

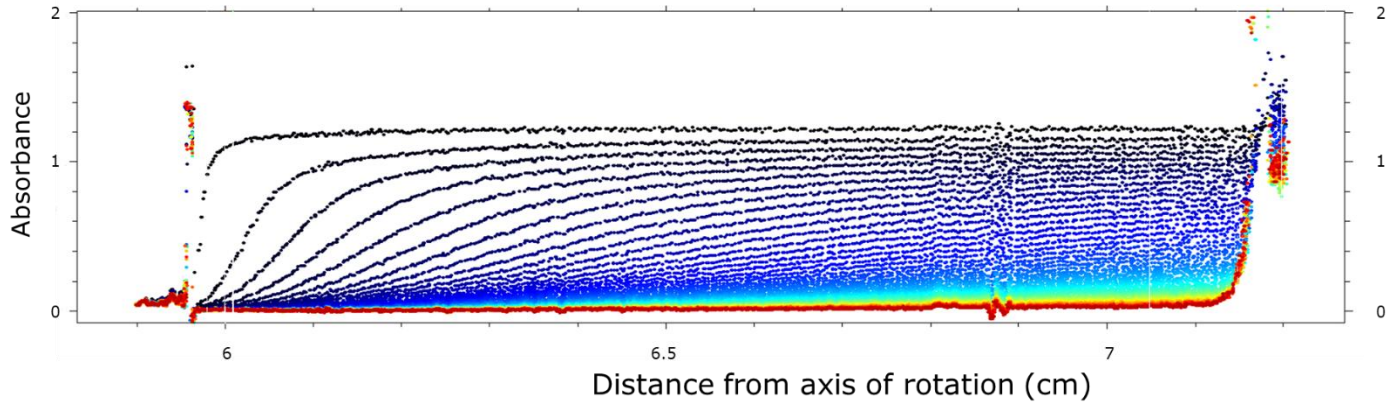


Figure 18: Absorbance bbb/bottom fraction from AUC measurement, blue=intial, red=final

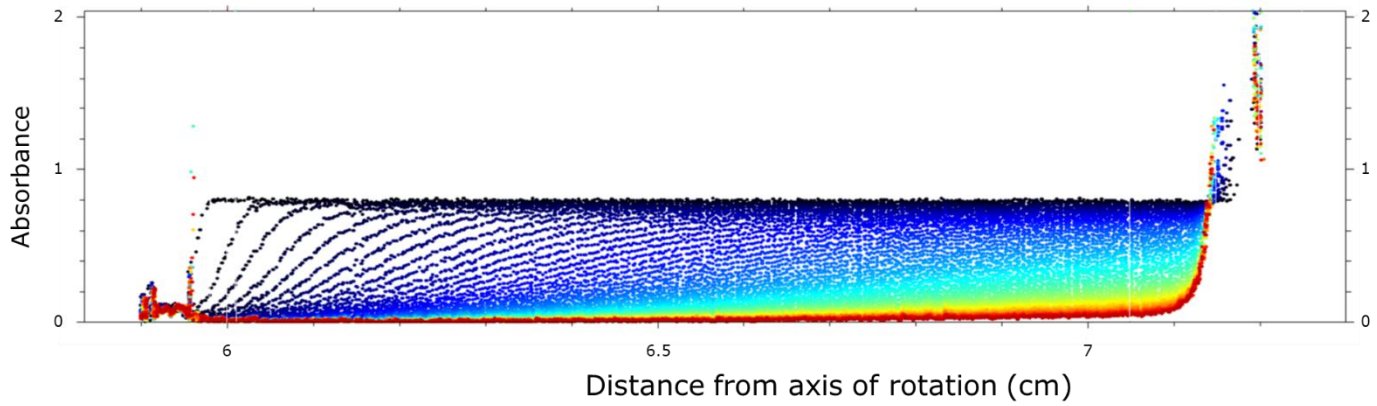


Figure 19: Absorbance btb/middle fraction from AUC measurement, blue=intial, red=final

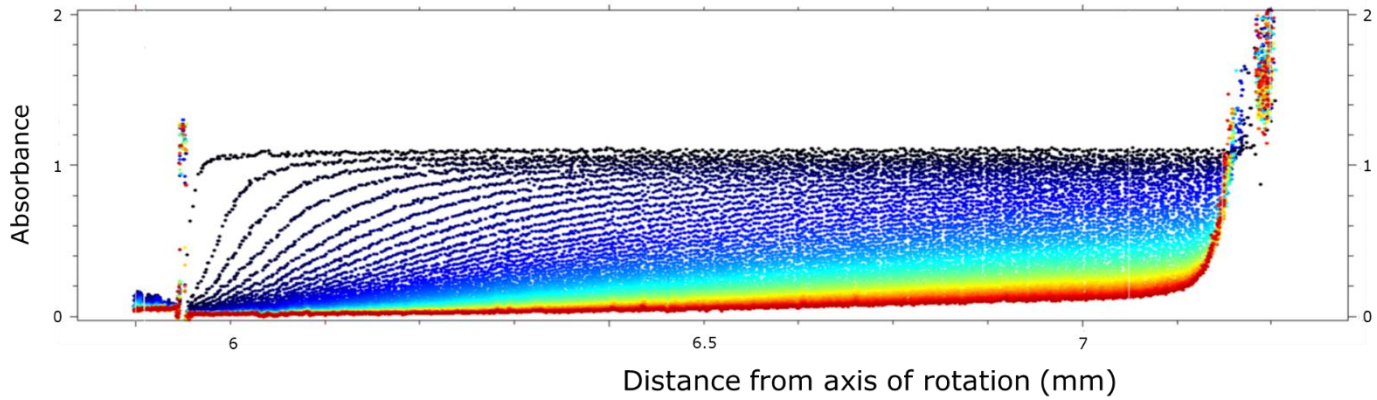


Figure 20: Absorbance Stock from AUC measurement, blue=intial, red=final

4.4 Theoretical Equilibria of Monodisperse Nanoparticles

Calculations of model II are run till a final time of 300 hours. For every hour in the model a save file is created. The diameters of the superparamagnetic nanoparticles calculated were chosen from 1.5 nm to 16.5 nm, with intervals of 0.5 nm. This equates to radii of 0.75 nm to 8.25 nm and intervals of 0.25 nm. The average location of the nanoparticles will initially be at the middle of the fluid column, with a column of height 0.01 m the average particle location at time zero will be at 0.0050 m and will decrease in time, since nanoparticles sediment towards the bottom of the fluid column (see figure 4). In the last couple of hours in the model the concentration profile does not change significantly anymore, a clear indication that sedimentation-diffusion equilibrium has been reached, or is close to being reached. In fact, the concentration profiles do not change much after around 180 hours, for nanoparticles with a high nanoparticle diameter (>10 nm) the profile does not change much after 100 hours, attesting that superparamagnetic fluids that contain larger nanoparticles will reach sedimentation equilibrium faster than fluids

containing smaller nanoparticles. The model was run several times with different values for the time step (dt). Decreasing the dt increases the accuracy of the calculation, but increases the computation time. Small Nanoparticles (below approximately 10 nm) show no big difference in the calculated concentration profiles when dt values differ by just a few seconds. On the other hand, nanoparticles with a large diameter show a significant change in average location at equilibrium (see figure 21). This is to be expected, since larger nanoparticles experience a higher magnetic and gravitational force; more time between diffusive steps will result in increased sedimentation in larger nanoparticles compared to smaller nanoparticles.

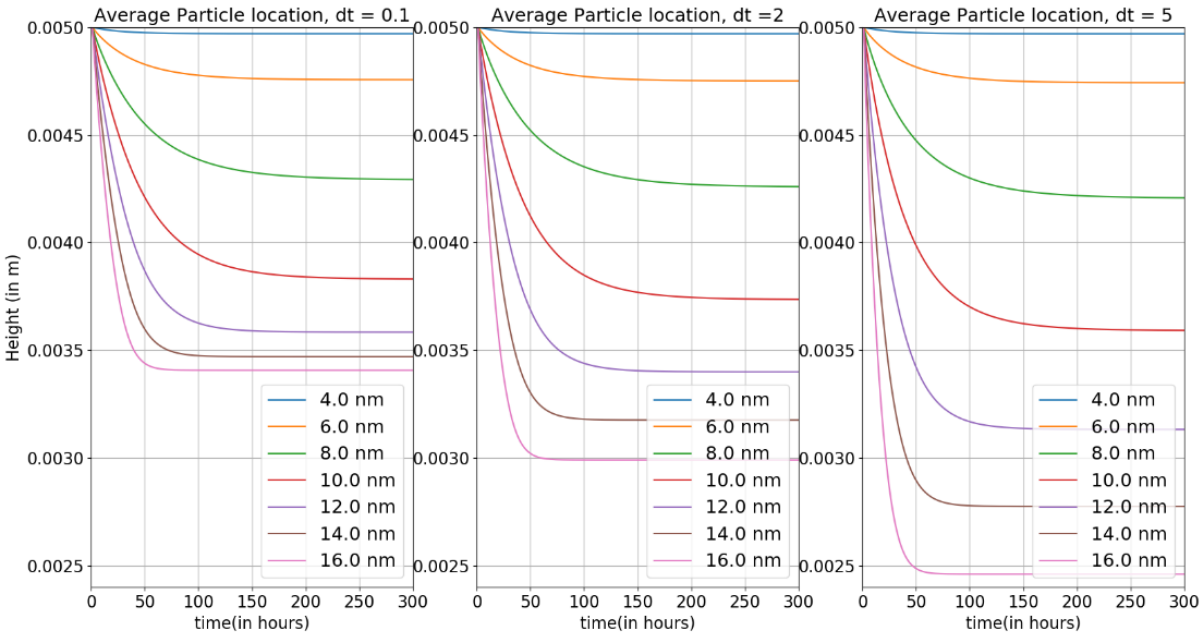


Figure 22: Average particle location of btb/middle fraction in time for time steps of 0.1, 2 and 5 s

After letting the dynamic model (II) run multiple times with different values of the time step between calculations (dt), it became apparent that the change in concentration from initial concentration (c/c_0) obtained from model II varies linearly with dt. This has been illustrated in figure 22 for the case of the concentration calculated for the top row of the magnetic fluid column for monodisperse nanoparticles with diameter 1.5 nm at a time of 300 hours after the beginning of the sedimentation. Here c displays c/c_0 displays the concentration at time 300h over the initial concentration at time t=0, before sedimentation has started. From the linear dependence of c/c_0 on dt one can conclude that it suffices to let the dynamic model run twice (for two different values of dt) to extrapolate the

concentration profile at dt=0 can be extrapolated. Also, since the concentration profile at dt=0.1s do not differ much with dt=0 (by calculating the formula for the linear fit).

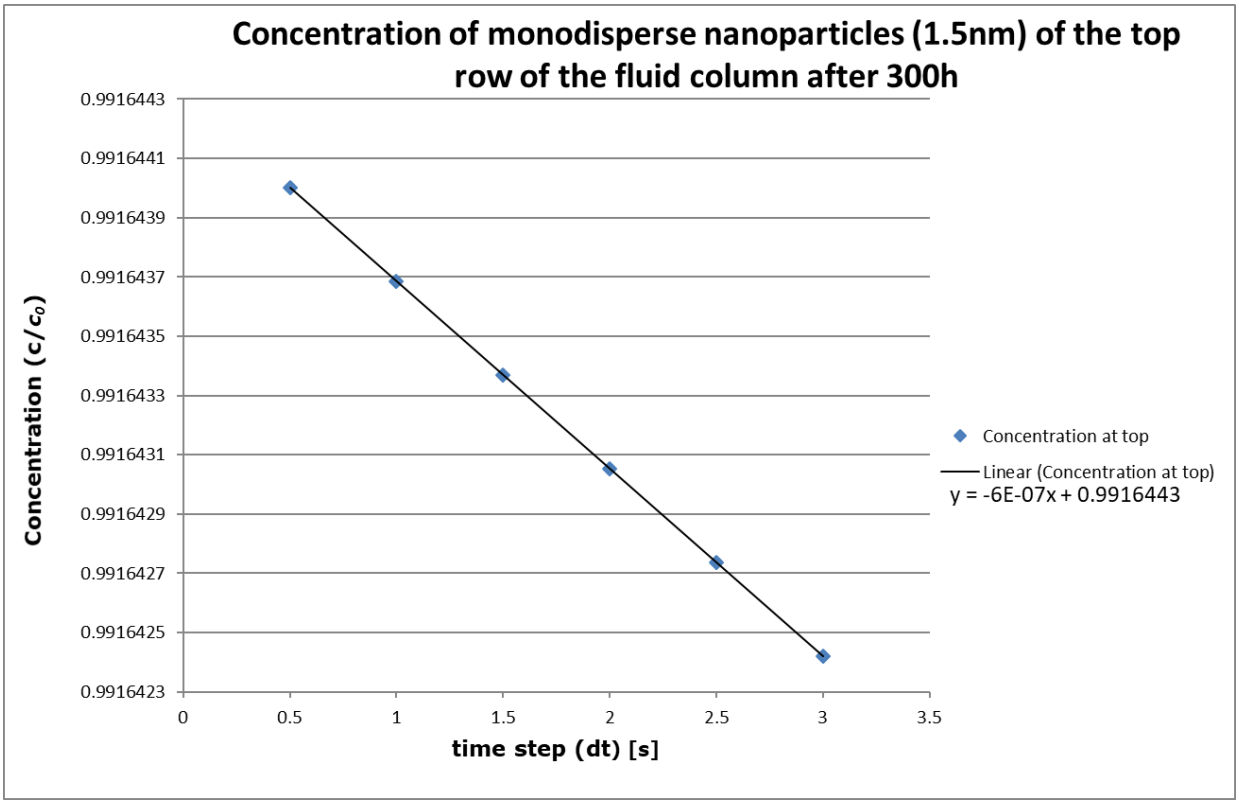


Figure 22: Concentration (c/c_0) of superparamagnetic nanoparticles obtained from model II at the top most row of magnetic fluid column after 300 hours of sedimentation dependent on the length of the time step (dt) in seconds with linear fit.

In figures 23 and 24 the sedimentation behavior of monodisperse nanoparticles is evaluated in terms of change from initial concentration (c/c_0). Where c is the concentration at a specific point in time and c_0 the initial concentration. The initial concentration (c_0) has been chosen to be exactly 1 at all points in height. There the height is the height from the magnet in meters. The liquid column starts at a height of approximately 0.010 m above the magnet and ends at a height of 0.020 m above the magnet (see figure 4). It can be seen again that larger nanoparticles go quicker towards equilibrium than smaller nanoparticles. Also, at sedimentation-diffusion equilibrium for large nanoparticles there will be relatively more nanoparticles at the bottom of the fluid column than is the case for smaller nanoparticles at sedimentation-diffusion equilibrium. At the top of the fluid column larger nanoparticles will show a larger decrease in nanoparticle concentration than smaller nanoparticles at equilibrium. For instance, at equilibrium

for the top of the fluid column nanoparticles with only 4.0 nm (in diameter) the concentration (c/c_0) decreases roughly by 4% and increases by 6% at the bottom of the column. In contrast, for nanoparticles of 8.0 nm (in diameter) the concentration (c/c_0) is more than halved at the top in equilibrium and tripled at the bottom.

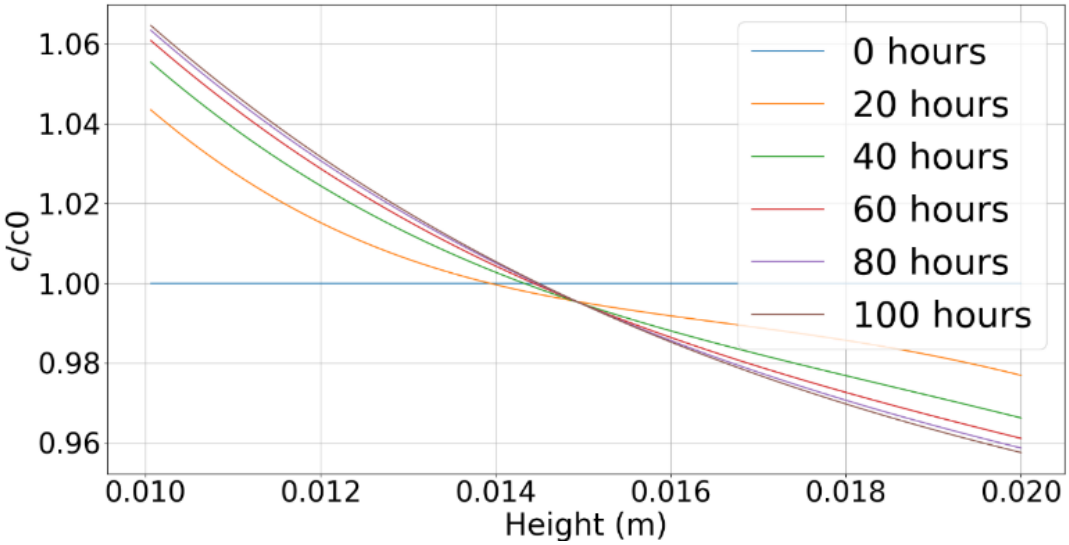


Figure 23: Development of concentration profiles in time with nanoparticles of 4.0 nm in diameter and dt = 0.1 s

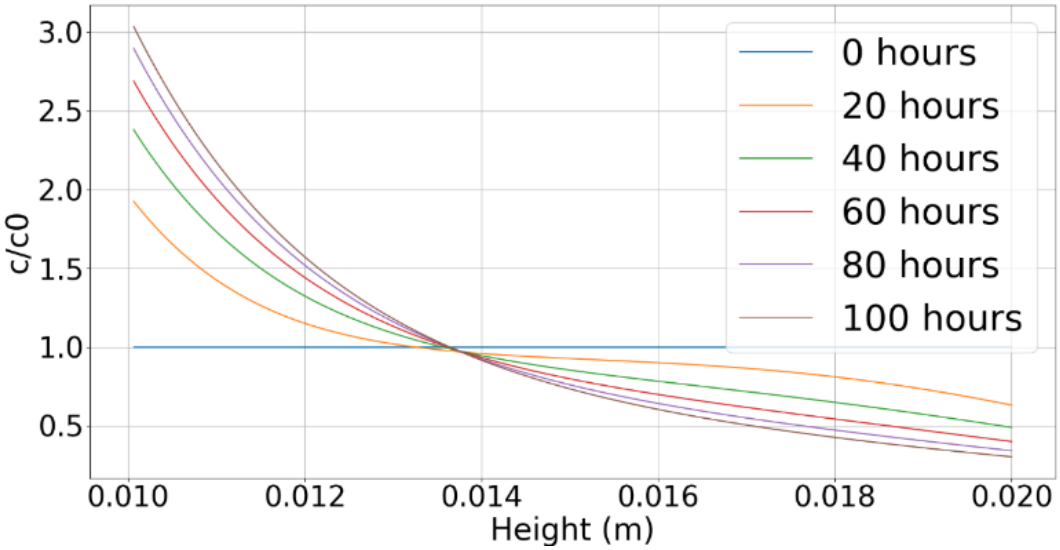


Figure 24: Development of concentration with nanoparticles of 8.0 nm in diameter and dt = 0.1 s

4.5 Theoretical Equilibria of Polydisperse Particles

In this section, the polydisperse case is calculated theoretically for a time step (dt) of 0.1 s and 5 s respectively possessing a log-normal distribution that was determined from the TEM analysis of btb and bbb fractions. Both the dynamic model (I) and the sedimentation-diffusion equilibrium model (II) are shown. The results go until 6 days (144 hours), since the concentration profile does not change considerably after this. In figures 25 and 26 a log-normal function for the btb fraction is chosen and in figure 27 the log-normal function for the bbb fraction is chosen. It can be seen that after 6 days, the calculation using a time step of 100 ms is closer to the equilibrium profile calculated from model (I) than when using a time step of 5 s, indicating that the dynamic model will indeed go towards the sedimentation-diffusion equilibrium profile if evaluated for a large enough. Using even smaller time steps would improve the accuracy of the final graph somewhat, but will go at the cost of calculation time. As is to be expected, the concentration at the bottom is higher for the bbb equilibrium profile compared to the btb equilibrium profile. The concentration at the top of the bbb fraction is not much different from the concentration at the top of the btb fraction.

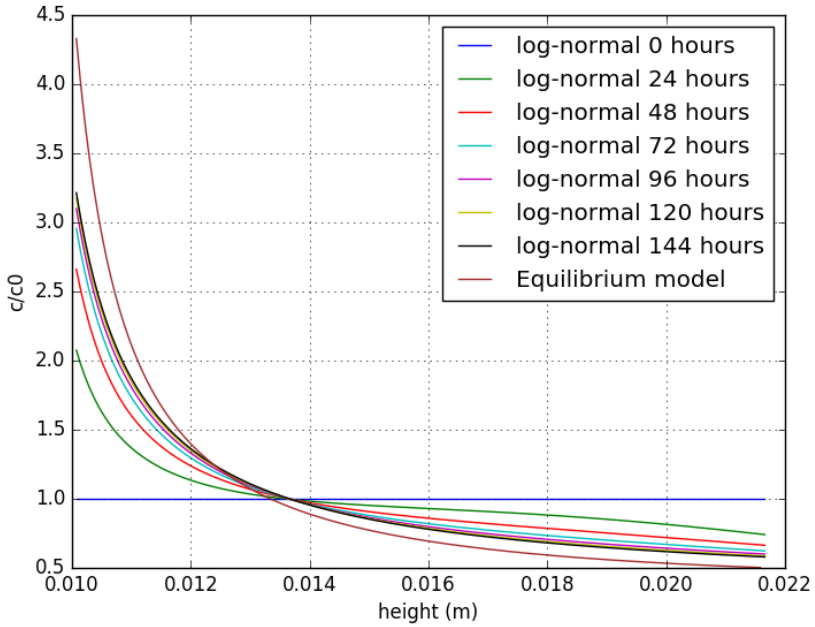


Figure 25: Polydisperse btb/middle faction dynamic concentration profile calculation with dt=0.1 s, the dynamic development and the concentration profile at sedimentation-diffusion equilibrium is shown

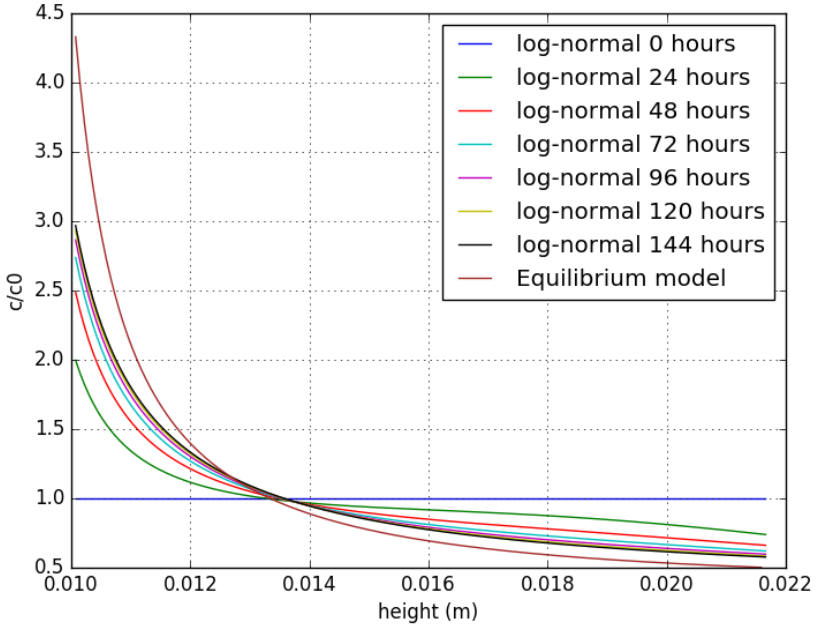


Figure 26: Polydisperse btb/middle fraction dynamic concentration profile calculation with dt=5 s, the dynamic development and the concentration profile at sedimentation-diffusion equilibrium is shown

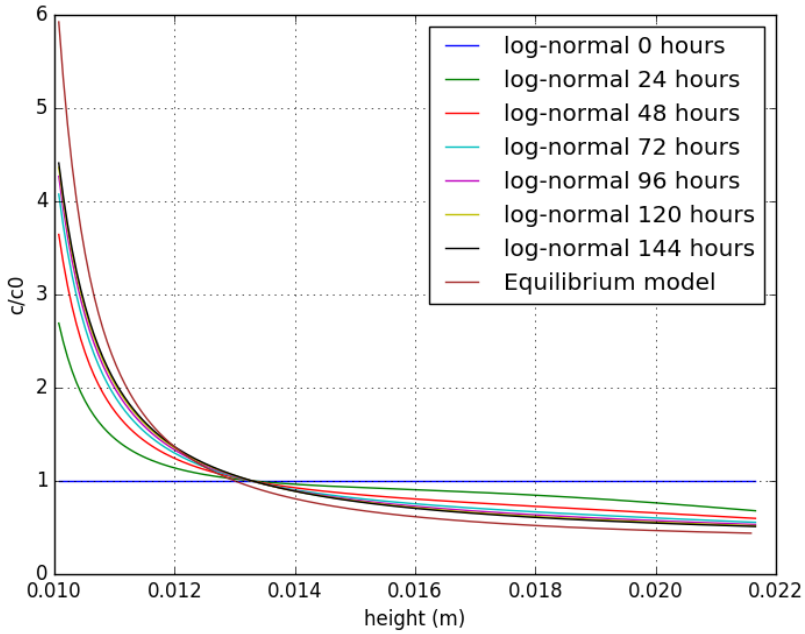


Figure 27: Polydisperse bbb/bottom fraction dynamic concentration profile calculation with dt=0.1 s, the dynamic development and the concentration profile at sedimentation-diffusion equilibrium is shown

Unlike a concentration profile in a gravitational field, the sedimentation-diffusion equilibrium profile does not follow a barometric distribution, i.e. the decrease in concentration from the top to the bottom of the fluid column at equilibrium is not logarithmic. It has been stated already in chapter 3.1 that this is due to the dependency of the magnetic field on the height from the magnet as illustrated in equation (1). This has been exemplified for the btb/middle fraction in figure 28. If the barometric distribution would hold for the equilibrium profile, the graph should be linear in a plot with a logarithmic y-axis. In fact, the decrease is even larger than logarithmic.

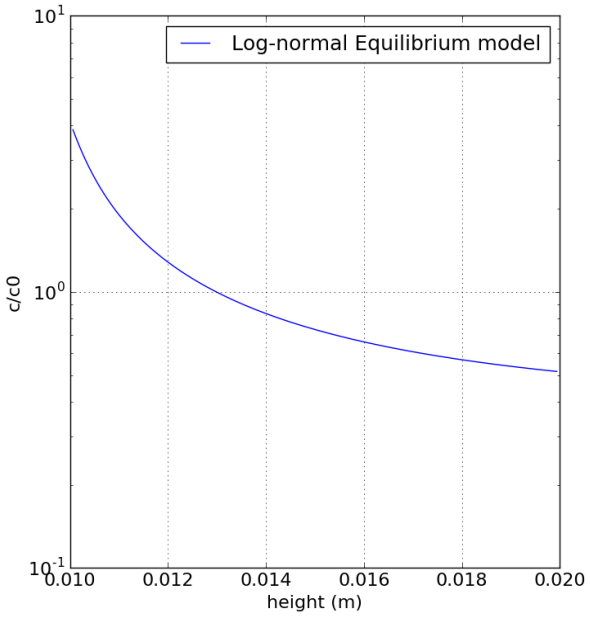


Figure 28: btb/middle fraction sedimentation-diffusion equilibrium concentration (c/c_0) profile with a logarithmic scale in the y-axis (dt=0.1 s)

4.6 Comparison magnetic sedimentation results

Finally, a comparison is made between the experimental and both the dynamic model (I) and the sedimentation-diffusion equilibrium model (II). Below the pictures of the btb sample for three times can be seen. The bottom of the cuvette can clearly be seen to be dark, attesting to the high concentration there.

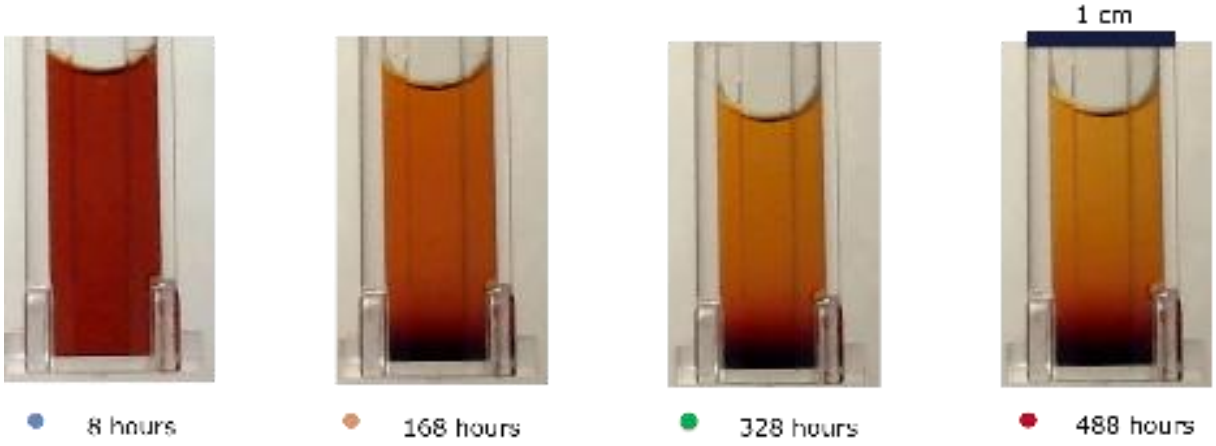


Figure 29: Four time steps after initiating the sedimentation of the analyzed btb/middle fraction sample

As can be seen in the comparison between the results from btb and the dynamic model (in figure 30), the shape of the sedimentation profile generally agrees with theoretical model. Here the height again displays the height from the magnet in meters. In the experimental graph the data does not start at a height of exactly 0.010 m, since a few pixels at the bottom of the liquid column were not included in the crop (as discussed in chapter 3.1). c/c_0 indicates the concentration over the initial concentration. The experimental profiles show far higher sedimentation to the bottom of the fluid column than what is obtained from the dynamic model with the log-normal distribution as obtained from TEM.

For the bbb fraction sedimentation under a magnetic field (see figure 31) also results in a larger concentration at equilibrium at the bottom of the fluid column than obtained from the dynamic model. In this case the concentration profile looks less similar to the profile from the dynamic model than was the case for the btb fraction. The concentration increases more rapidly when approaching the bottom of the column than it does in the experimental results from the btb fraction. A larger agglomeration of nanoparticles in the bbb fraction could be the reason for this. Larger nanoparticles have a bigger impact on sedimentation than what is obtained from the log-normal functions due to them having a higher tendency to coagulate and form larger nanoparticles in a magnetic field (Relle et al. 1999), resulting in the shape seen in figure 31.

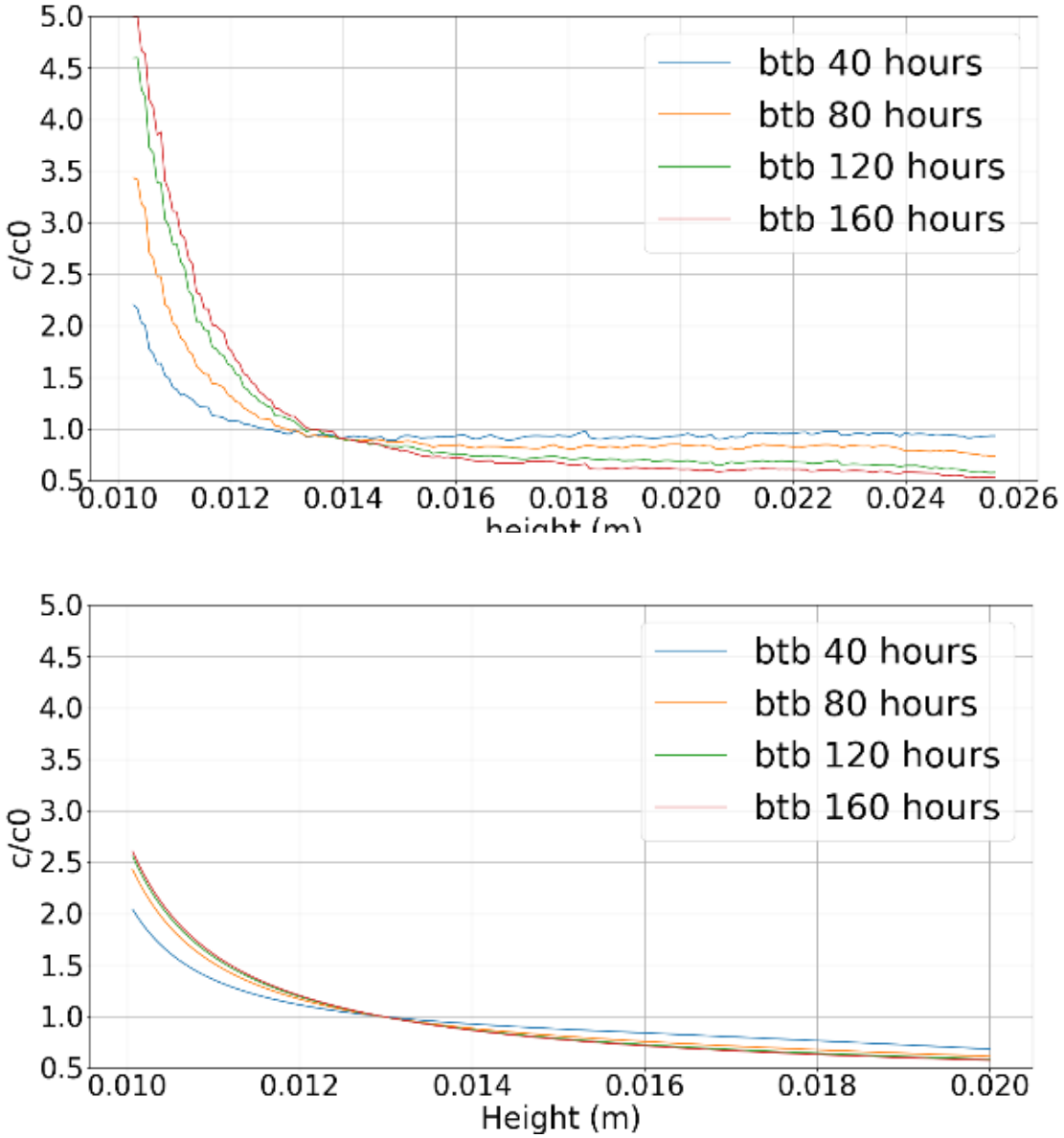
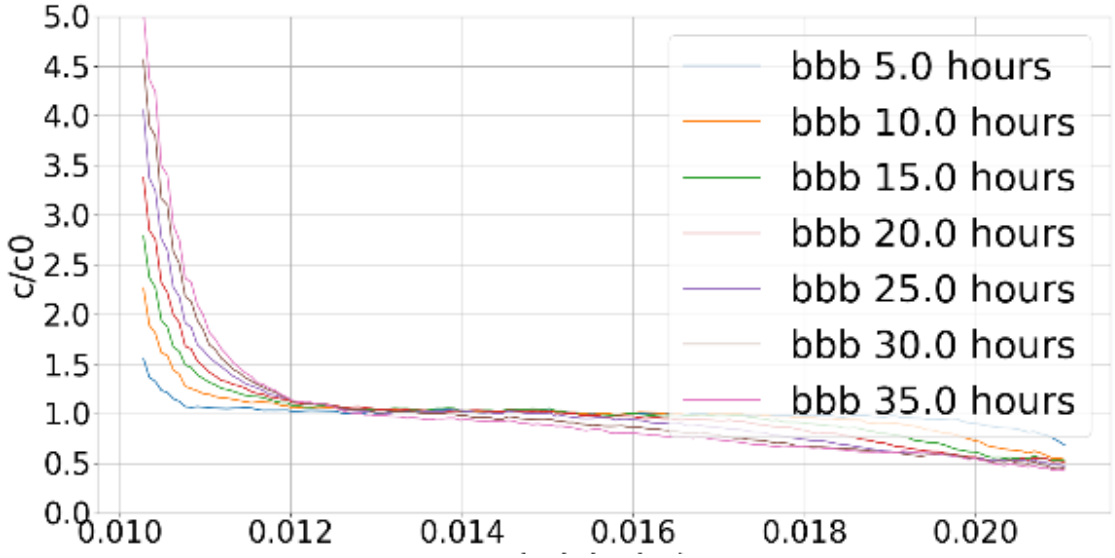


Figure 30: Top: experimental results for btb/middle fraction, bottom: dynamic development of concentration profiles for btb/middle fraction (dt=0.1)



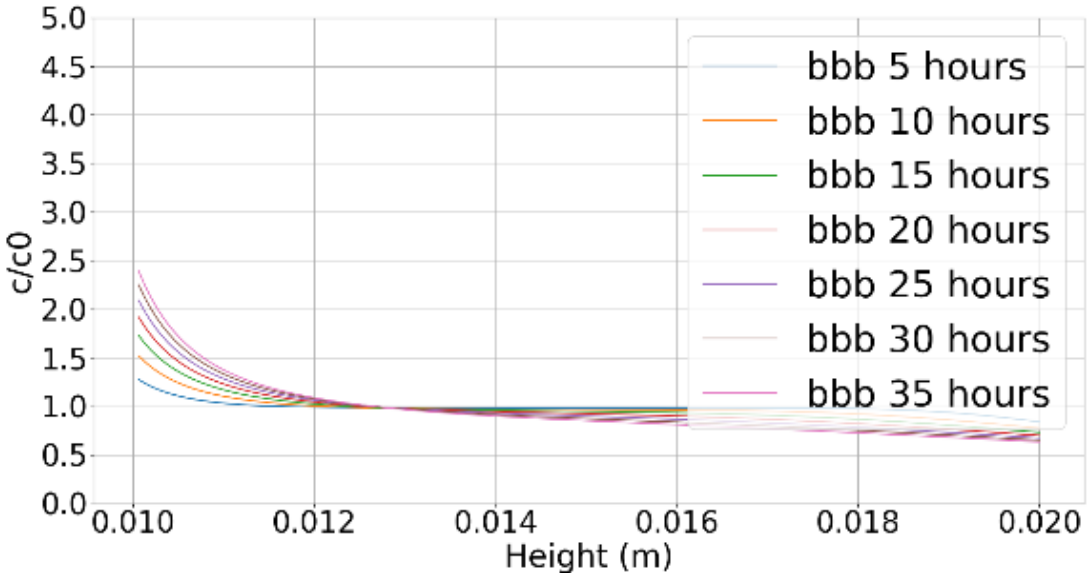


Figure 31: top: experimental results for bbb/bottom fraction, bottom: dynamic development of concentration profiles for bbb/bottom fraction (dt=0.1)

5: Conclusion and Outlook

The general shape of the equilibrium concentration profile obtained from the transmission experiments agrees with the theoretical models. Yet, the transmission experiment concentration profiles close to equilibrium do not agree quantitatively with the concentration profiles obtained from the dynamic model. This could be due to the assumption of ideal nanoparticles in the model. Interactions between superparamagnetic nanoparticles could lead to an increased sedimentation velocity due to agglomeration of nanoparticles. The equilibrium concentration profile will have a higher concentration (c/c_0) at the bottom of the cuvette than is predicted by the ideal model, since the presence of larger nanoparticles is increased. Also, evaporation of solvent will have an effect on the experimental samples, this was especially seen in the final few images taken, where the superparamagnetic fluid as a whole actually became darker and the concentration profile saw an increase in c/c0 at the top of the cuvette. Therefore it was difficult to pinpoint the exact time at which equilibrium is reached in the experiments. If the cuvette would be sealed airtight, this should not be an issue anymore.

From the fractionation according to Lefebure et al. (chapter 3.3) samples were obtained with a differing average nanoparticle diameter according to TEM analysis. Also according to Transmission Electron Microscopy (TEM) and Dynamic Light Scattering (DLS) measurements, samples obtained from fractionation still showed a large polydispersity. The VSM analysis yielded the same median diameter for all three fractions tested (bbb/bottom, btb/middle and tbt/top). The discrepancy between TEM and VSM in this regard is presumably caused by the nanoparticles being composed of a crystalline core around which an amorphous layer of maghemite atoms is located. This amorphous layer is quantitatively different between the fractionated samples. The sedimentation-diffusion equilibrium model (I) yielded concentration profiles that were in accordance with the concentration profiles of the dynamic model (II) at large times after inset of sedimentation. Unlikethe equilibrium concentration profile for sedimentation in a magnetic field, the equilibrium concentration profile of sedimentation in a magnetic field does not decrease logarithmically in height. From the optical setup, concentration profiles in time could be obtained. In shape they looked similar to concentration profiles obtained with the theoretical models. Yet they did not correspond quantitatively to the results from the models.

By using only the red sensors in of the webcam a lot of data created by the camera is not used. Therefore, repeating this experiment with a monochrome camera would be more efficient, since the exact color of the sample is irrelevant for the evaluation. Also, degree of redness is not a perfectly accurate measure for concentration in the f used. Using a true monochromatic light source, such as a red laser would solve this problem, since the red values are measuring a large range of

wavelengths and not just a single wavelength. Also, to reduce the agglomeration of nanoparticles, one can think of adding a salt solution to the superparamagnetic fluid. Since, the salt ions will screen the charges of the nanoparticles (Luigjes 2012). Another improvement would be coating the (maghemite) nanoparticles with a bulky surfactant, such as citrate or a polymer like polyethylene glycol (PEG). This ensures that the nanoparticles disperse more easily in water and also less likely to agglomerate.

In conclusion, this research provides for a proof-of-concept and a better understanding of the theoretical aspects behind sedimentation of superparamagnetic nanoparticles in a magnetic field. But, the concentration profiles obtained from the models are still too dissimilar to conclude that a simple force-balance model is adequate to predict the behavior of superparamagnetic nanoparticles in a magnetic field. A more complex model is required to better explain magnetic sedimentation. The optical experimental setup does provide for a simple, usable method to conduct sedimentation experiments in situ.

6: Bibliography

1. F. Ahrentorp, A. P. Astalan, C. Jonasson, J. Blomgren, B. Qi et al. Sensitive High Frequency AC Susceptometry in Magnetic Nanoparticle Applications. AIP Conference Proceedings. 1311(213): 213-223, 2010.

2. R. L. Bailey. Lesser Known Applications of Ferrofluids. Journal of Magnetism and Magnetic Materials. North-Holland Publishing Company, 39: 178-182, 1983.

3. G. K. Batchelor. Sedimentation in a Dilute Dispersion of Spheres. Journal of Fluid Mechanics, Cambridge, United Kingdom. 1972.

4. J. F. Berret, O. Sandre, A. Mauger. Size distribution of superparamagnetic particles determined by magnetic sedimentation. Langmuir. 23: 2993 – 2999, 2007.

5. C. F. Bohren, D. R. Huffman. Absorption and Scattering of Light by Small Particles. Wiley-VCH Verlag GMBH & co. KGaA. 2008.

6. W. Boon. Concentration Profiles of Ferrofluids in Magnetic Fields. Master's Thesis. 2018.

7. W. Cao, B. Demeler. Modeling Analytical Ultracentrifugation Experiments with an Adaptive Space-Time Finite Element Solution of the Lamm Equation. Biophys. J., 89(3): 1589–1602, 2005.

8. J. M. D. Coey, and D. Khalafalla. Superparamagnetic γ-Fe_2O_3. Physica Status Solidi A, 11: 229-241, 1972.

9. J. Cole, J Lary, T. Moody, T, Laue. Analytical Ultracentrifugation: Sedimentation Velocity and Sedimentation equilibrium. Methods in Cell Biology. 84:143 – 179, 2008.

10. R.M. Cornell, U. Swertmann. The Iron Oxides. Structure, Properties, Reactions, Occurrences and Uses. 2003.

11. M. J. Costello, S. Johnsen, K. O. Gilliland, C. D. Freel, W. C. Fowler. Predicted Light Scattering from Particles Observed in Human Age-Related Nuclear Cataracts using Mie Scattering Theory. Investigative Ophthalmology & Visual Science. 48(1): 303-312, 2007.

12. Z. Dogic, A. P. Philipse, S. Fraden, J. K. G. Dhont. Concentration-Dependent Sedimentation of Colloidal Rods. Chem. Phys. 113(28): 8368-8380, 2000.

13. B. Erné. Lecture notes on Colloidal Analysis Techniques. Nanomaterials Science Master Program, Graduate School of Natural Sciences, Utrecht University. 2018

14. H. Fay, J. M. Quets, H. Hatwell. US Patent No. US05645016. Union Carbide Corp. Apparatus and process for the separation of particles of different density with magnetic fluids, 1975.

15. S. Foner. Versatile and Sensitive Vibrating-Sample Magnetometer. Review of Scientific Instruments. 27(261), 1956.

16. S. Foner. Review of Magnetometry. IEEE 17(6): 3358 – 3363, 1981.

17. F. Gazeau, E. Dubois, M. Hennion, R. Perzynski, Y. Raikher. Quasi-elastic neutron scattering on γ-Fe_2O^3 nanoparticles. Europhysics Letters. 40(5): 575-580, 1997.

18. P.G. de Gennes, P.A., Pincus. Pair Correlations in a Ferromagnetic Colloid. Physik der Kondensierten Materie. 11(3): 189 – 198, 1970.

19. D. Horák, M. Babič, H. Macková, M. J. Beneš. Preparation and properties of magnetic nano- and microsized particles for biological and environmental applications, J. Sep. Sci. 30: 1751-1772, 2007

20. T. Kimura, S. Mamada, M. Yamato, K. Kitazawa. US Patent No. US6902065B2. Japan Society for the Promotion of Science. Method for separation of plastic mixtures based on magneto-Archimedes levitation, 2005.

21. S. Lefebure, E. Dubois, V. Cabuil, S. Neveu, R. Massart. Monodisperse magnetic nanoparticles: Preparation and dispersion in water and oils. Journal of Materials Research, 13(10):2975–2981, 1998.

22. G. Liu, J. Gao, H. Ai, and X. Chen. Applications and Potential Toxicity of Magnetic Iron Oxide Nanoparticles. Small. 9(9-10): 1533, 2013.

23. B. Luigjes. Synthesis and Analytical Centrifugation of Magnetic Colloids. Dissertation. Utrecht University. 2012. J. Phys.: Condes. Matter.

24. B. Luigjes, D. M. E. Thies-Weesie, A. P. Philipse, B. H. Erné. Sedimentation equilibria of ferrofluids: I. Analytical centrifugation in ultrathin glass capillaries. 24(24): 1-8, 2012.

25. V. Marghussian. Magnetic Properties of Nano-Glass Ceramics. Nano-Glass Ceramics. Processing, Properties and Applications. 181- 223, 2015.

26. M. Marolt. Superparamagnetic Materials. Seminar I_b. Faculty of Mathematics and Physics, University of Ljubljana. 2014

27. R. Massart. Preparation of aqeous magnetic liquids in alkaline and acidic media. IEEE Transactions on Magnetics, 17(2):1247 – 1248, 1981.

28. J. Mazumdar. An Introduction to Mathematical Physiology and Biology. Cambridge University Press, Cambridge, United Kingdom. 1999.

29. Q. A. Pankhurst, J. Connoly, S. K. Jones, and J. Dobson. Applications of Magnetic Nanoparticles in Biomedicine. Journal of Physics D: Applied Physics. 36: 167-181, 2003.

30. G. A. Parks. The isoelectic points of solid oxides, solid hydroxides, and aqueous hydroxo complex systems. Chemical Reviews. ACS Publications. 1965.

31. E. A. Peterson, D. A. Krueger. Reversible, Field Induced Agglomeration in Magnetic Colloids. Journal of Colloid and Interface Science. 52(1): 24-34, 1977.

32. R. Piazza, S. Buzzacaro, A. Secchi. The unbearable heaviness of colloids: facts, surprises and puzzles in sedimentation. Journal of Phyics: Condensed Matter. 24(28): 1-13, 2012.

33. A. P. Philipse, D. Maas. Magnetic Colloids from Magnetotactic Bacteria: Chain Formation and Colloidal Stability. Langmuir. 18(24): 9977-9984, 2002.

34. A. P. Philipse. Notes on Brownian motion. Nanomaterials Science Master Program, Graduate School of Natural Sciences, Utrecht University. 2011.

35. S. Papell. US Patent No. US3215572A. National Aeronautics and Space Administration. Low viscosity magnetic fluid obtained by the colloidal suspension of magnetic particles, 1963.

36. K. L. Planken, M. Klokkenburg, J. Groenewold, A. P. Philipse, Ultracentrifugation of Single-Domain Magnetite Particles and the De Gennes-Pincus Approach to Ferromagnetic Colloids in the Dilute Regime. The Journal of Physical Chemistry. 113(12): 3932–3940, 2009.

37. K. L. Planken, H. Cölfen. Analytical Ultracentrifugation of colloids. Nanoscale. 2:1849– 1869, 2012.

38. G. Ralston. Introduction to Analytical Ultracentrifugation. Beckman, Sydney, Australia. 1993.

39. J. A. Ramos Guivar, A. Isaías Martínez, A. Osorio Anaya, L. De Los Santos Valladares, L. León Félix, A. Bustamante Dominguez. Structural and Magnetic Properties of Monophasic Maghemite (γ-Fe2O3) Nanocrystalline Powder. Advances in Nanoparticles. 3: 114-121, 2014.

40. S. Relle, S. B. Grant, T. Costas. Diffusional coagulation of superparamagnetic particles in the presence of an external magnetic field, Physica A: Statistical Mechanics and its Applications. 270(3-4): 427-443, 1999.

41. P. C. Rem, S. P. M. Berkhout. US Patent No. US2016288137, NL patent No. NL2011559. Urban Mining Corp B.V. Improved Magnetic Density Separation Device and Method, 2006.

42. P. C. Rem, S. P. M. Berkhout. US Patent No. US20110042274A1. NL Patent No. NL2001322A. Urban Mining Corp B.V. Method and apparatus for the separation of solid particles having different densities, 2010.

43. R. E. Rosensweig, Magnetic Fluids. Scientific American. 247(4): 136-145, 1982.

44. W. B. Russel, D. A. Saville, W. R. Schowalter. Colloidal Dispersions. Cambridge University Press. Cambridge, United Kingdom. 1999.

45. P. Schuck. Size-Distribution Analysis of Macromolecules by Sedimentation Velocity Ultracentrifugation and Lamm Equation Modeling. Biophys. J., 78(3): 1606-1619, 2000.

46. J. H. Seinfeld, S. N. Pandis. Atmospheric Chemistry and Physics: From Air Pollution to Climate Change. Wiley. 2016

47. A. van Silfhout, B. H. Erné. Magnetic detection of nanoparticle sedimentation in magnetized ferrofluids. Journal of Magnetism and Magnetic Materials. 472: 53-58, 2019.

48. N. Singh, G. J. S. Jenkins, R. Asadi, and S. H. Doak. Potential Toxicity of Superparamagnetic iron oxide nanoparticles (SPION). Nano Reviews. 1(1): 5358, 2010.

49. D. F. Swinehart. The Beer-Lambert Law. J. Chem. Educ. 39(7): 333, 1962.

50. I. Vatta. Floating Diamonds with nanomagnetic particles, Macroml. Symp. 225: 221-228, 2005.

51. M. Vert, Y. Doi, K.-M. Hellwich, M. Hess, P. Hodge, P. Kubisa. M. Rinaudo, F. Schué. Terminology for biorelated polymers and applications (IUPAC Reccomendations 2012). Pure Apl. Chem. 84(2): 377-410, 2012.

52. D. B. Williams, C. B. Carter. Transmission Electron Microscopy. Springer, Huntsville, USA. 2009.

Acknowledgments

Firstly, I would like to thank Alex van Silfhout for being my daily supervisor during the duration of my Master Thesis research. I would also like to thank him for doing the VSM experiments and making the TEM images of my superparamagnetic fluid sample. Secondly, I would like to thank Dr. Dominique Thies-Weesie for her help, expertise and patience with the Analytical Ultracentrifuge. Also, I would like to thank Dr. Ben Erné for his ideas for improving the setup of the webcam experiments and valuable feedback. Moreover, I would like to thank Professor Albert Philipse for being my second examiner. Lastly, I would like to thank all staff and students at FCC for the interesting discussions during the coffee breaks and the conducive and curious research and learning environment.

Appendix I: Device Settings Optical Setup to Monitor Magnetic Sedimentation

Here the device settings that were chosen in the Debut Video Capture Software (from NCH Software) for the conduction of the optical experiment to monitor magnetic sedimentation (see chapter 3.1). The samples tbt, btb and bbb were the fractions obtained from fractionation according to chapter 3.3.

Device Settings

Brightness	141
Contrast	32
Saturation	32
Sharpness	60
White Balance	2800
Backlight Compensation	1
Gain	25
Powerline Frequency	50 (Hz)

Camera Control

Zoom	3
Focus	70
Exposure	-6
Pan	0
Tilt	0
Low Light Compensation	on

Figure I.1: Device Settings and Camera Control for Sample tbt in Optical Setup, an image is taken every 12 hours

Device Settings

Brightness	119
Contrast	32
Saturation	32
Sharpness	54
White Balance	2800
Backlight Compensation	1
Gain	26
Powerline Frequency	50 (Hz)

Camera Control

Zoom	3
Focus	65
Exposure	-6
Pan	0
Tilt	0
Low Light Compensation	off

Figure I.2: Device Settings and Camera Control for Sample btb in Optical Setup, an image is taken every 8 hours

Device Settings

Brightness	119
Contrast	32
Saturation	32
Sharpness	54
White Balance	2800
Backlight Compensation	1
Gain	16
Powerline Frequency	50 (Hz)

Camera Control

Zoom	3
Focus	80
Exposure	-6
Pan	0
Tilt	0
Low Light Compensation	off

Figure I.3: Device Settings and Camera Control for Sample bbb in Optical Setup, an image is taken first every 20 min, after 48 hours and 40 minutes, every 4 hours

Appendix II: Settings Analytical Ultracentrifugation Experiments

Here the settings that were applied in conducted the analytical ultracentrifugation (AUC) experiments are displayed. Three samples were analyzed, namely the bbb fraction and btb fraction obtained from fractionation (as discussed in chapter 3.3) and the stock solution that had not undergone fractionation.

Sample	Path-length Cell	Center-piece	Window	W1 (nm)	Rmin (cm)	Rmax (cm)	Rotation (rpm)	Temperature (K)
bbb	3 mm	Epon	Quartz	485	5.9	7.2	3000	293
Btb	3 mm	Epon	Quartz	485	5.9	7.2	3000	293
Stock	3 mm	Epon	Quartz	397	6.0	7.2	3000	293

Figure II.1: Single wavelength scan

In figure II.1 W1 displays the wavelength of analysis for the single wavelength AUC measurement. Centerpiece gives the material the center piece is made of (in all cases Beckham Coulter epon Charcoal filled 3 mm centerpieces are used). A pathlength of 3 mm allows for a volume of 100 µL for both sample and reference to be inserted. Rmin and Rmax give the length from the axis of rotation that is to be analyzed. The reference in all cases was the solvent (demiwater). Below the UV/VIS measurements that were conducted to determine W1 are shown.

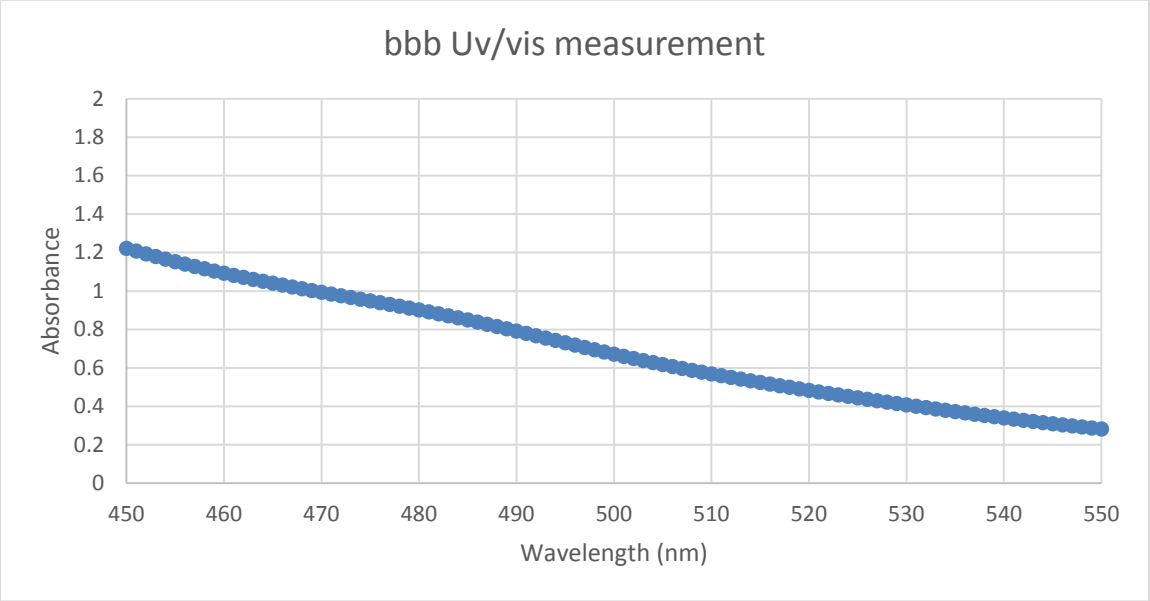


Figure II.2: Absorbance as a function of incident wavelength of the bbb fraction obtained from UV/VIS measurement

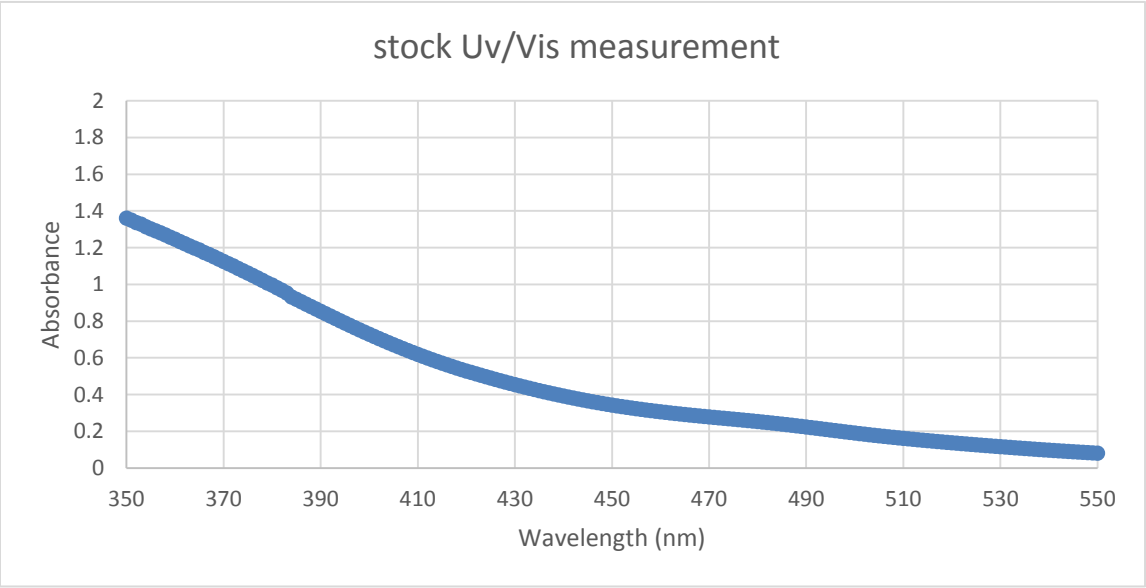


Figure II.3: Absorbance as a function of incident wavelength of the stock obtained from UV/VIS measurement

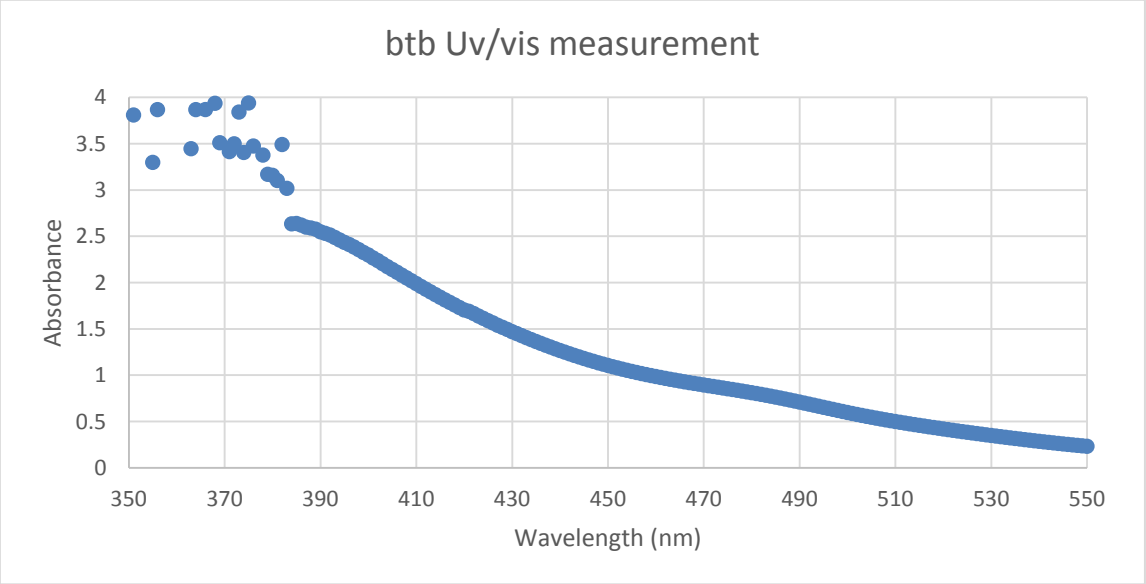


Figure II.4: Absorbance as a function of incident wavelength of the btb fraction obtained from UV/VIS measurement

Appendix III: Calculations for Concentration Profile in the Sedimentation-Diffusion Equilibrium Model

Here the exact procedure of how the concentration profile in the sedimentation-diffusion equilibrium model (I) is obtained is evaluated. Below the equation for the force balance in model I is shown again:

$$V\left(\left(\rho_{np}-\rho_{w}\right)g+\ M_{sat}\ L(B)\frac{dB}{dh}\right)=\ \frac{dc(n)}{c(h=n)}\frac{kT}{dh} \qquad \text{(III.1)}$$

Here the volume of the nanoparticles is given by $V=\frac{1}{6}\pi D^3$, where D is the diameter of the nanoparticles. The column is divided into identical volume elements (slabs) of height dh. In I.1, n is the slab number, such that $c(n=0)$ is the concentration at the highest slab and $c(n=1)$ is the concentration at the slab just below the highest slab, etc.

In order to obtain the concentration of superparamagnetic particles at every point in height at equilibrium, first the concentration at the highest volume element (h = 0) is defined to be equal to 1 ($c(n=0)=1$), the relative concentration at any height at equilibrium can then be calculated by filling in the force-balance equation (III.1). The value of $\frac{dc(n)}{c(h=n)}$ can then be obtained by solving:

$$\frac{dc(n)}{c(h=n)}=(\ V\left(M_{sat}\ L(B)\frac{dB}{dh}+\left(\rho_{np}-\rho_{w}\right)g\right))/(\frac{kT}{dh}) \qquad \text{(III.2)}$$

Taking $c(h=n)=1$ will allow for the calculation of $dc(n)$. It only becomes a matter of filling in (III.3) to obtain: $c(h=n+1)$, which is the concentration of superparamagnetic particles at one height element lower than $c(h=n)$.

$$c(h=n+1)=\ c(h=n)+dc(n) \qquad \text{(III.3)}$$

After calculating the concentration $c(h=1)$, the procedure is repeated to calculate the concentration for $h=2$. By repeating this procedure concentrations at all heights can be calculated

Appendix IV: Simulation in Time

With this code the concentration profile in time for a single particle diameter is calculated, the output is an excel file. The code was written in Python version 2.7, available at http://www.python.org.

```
#-------------------------------------------------------------------------------
import numpy as np
import time
import csv
#-------------------------------------------------------------------------------
# Here the magnetic field H (in Tesla) of the magnet is evaluated according
# to Van Silfhout and Erné, 2019
# Where, ms is the magnet's internal remanent magnetization in Tesla
# d is the thickness of the magnet in meters
# and R is the radius of the magnet in meters
#-------------------------------------------------------------------------------
def field(z):
    ms, d, R = 1.35, 0.03, 0.0225
    H = ms/2*((z + d)/np.sqrt((z+d)**2+R**2)-(z/np.sqrt(z**2+R**2)))
    return H
#-------------------------------------------------------------------------------
# Here the Langevin function is evaluated
# With T the temperature in Kelvin
# kb the boltzmann constant in m^2 kg s^-2 K^-1
#-------------------------------------------------------------------------------
def Langevin(ms, R, B):
    T = 293
    kb = 1.38064852e-23
    Langevin = [None]*(len(B) - 1)
    for i in range(len(B) - 1):
        zeta = (np.pi*((2*R)**3)*ms*(B[i]))/(6*kb*T) #since zeta is a large number,
coth
        if zeta < 14:
            coth = (np.exp(zeta) + np.exp(-zeta))/(np.exp(zeta)-np.exp(-zeta))
        elif zeta >= 14:
            coth = float(1)
        L = coth - 1/zeta
        Langevin[i] = L
    return Langevin
#-------------------------------------------------------------------------------
# Here a new concentration profile is evaluated
# with kb the boltzmann constant in m^2 kg s^-2 K^-1
# rho_w the density of water in kg/m^3
# rho_np the density of maghemite/iron oxide in kg/m^3
# g the gravitational acceleration in m/s^2
# T the temperature in Kelvin
# M_sat the magnetic saturation of the fluid in A/m
# viscocity_water the dynamic viscosity of water in Pa*s at 293 Kelvin
#-------------------------------------------------------------------------------
def next_prof(R, B, c, delta_t, step_size):
    kb = 1.38064852e-23
```

```
    rho_w = 997.
    rho_np = 5240.
    g = 9.81
    T = 293
    M_sat = 4.3e5
    viscosity_water = 1.0016e-3
    Fgrav = (4/3)*np.pi*(rho_np - rho_w)* R**3*g
    L = Langevin(M_sat, R, B) #Langevin function
    f = 6*np.pi*viscosity_water*R #friction factor
    v = [None]*(len(c)-1)
    for i in range(len(c) - 1):
        Fmag = (4/3)*np.pi*R**3*M_sat*L[i]*((B[i+1]-B[i])/step_size)
        Fdif = ((c[i+1]-c[i])/step_size)*(kb*T)/(c[i+1])
        v[i] = (Fgrav + Fmag - Fdif) / f
    new_c = c
    for i in range(len(c)-1):
        delta_c_fraction = (v[i] * delta_t) / step_size
        new_c[i] = new_c[i] - (c[i] * delta_c_fraction)
        new_c[i+1] = new_c[i+1] + (c[i] * delta_c_fraction)
    return new_c
#------------------------------------------------------------------------------
# Here the simulation, which evaluates consecutive concentration profiles
# is evaluated, h returns the heights and data_array the concentration profiles
# with interval determining what the time interval is between saved profiles
# h_bottom is the distance from the magnet to the bottom of the fluid column
# h_top is the distance from the magnet to the top of the fluid column
# bin_number is the number of bins in height between h_bottom and h_top
#------------------------------------------------------------------------------
def simulation(R, dt, t_max):
    interval = 3600.
    h_bottom = 0.01
    h_top = 0.02
    bin_number = 159
    bin_size = (h_top - h_bottom) / bin_number
    t_steps = int(t_max / dt)
    data_array = np.zeros([int(t_max/interval),bin_number+1])
    h = [h_top - (i * bin_size) for i in range(bin_number)]
    B = [None] * bin_number
    for i in range(bin_number):
        B[i] = field(h[i])
    profile = [1 for i in range(bin_number)]
    for i in range(t_steps):
        profile = next_prof(R, B, profile, dt, bin_size)
        if i in range(0,t_steps,int(interval/dt)):
            data_array[int(i/(interval/dt)),0] = i * dt
            data_array[int(i/(interval/dt)),1:] = profile
    return h, data_array
#------------------------------------------------------------------------------
# Here the obtained concentration profiles are written into a csv file
# with location determining where the file will be written to
# CAUTION: !!! Check the location parameter before starting the evaluation
# otherwise files might be overwritten, leading to data loss !!!
#------------------------------------------------------------------------------
def write(location, data_array, t_max_hour):
    with open(location, mode='wb') as write_file:
```

```
    # use in word binary mode to prevent an empty row being put after every written
row
        csv_writer = csv.writer(write_file, delimiter =',')
        profile = [-3600]
        for i in range(159):
            profile.append(1)
        csv_writer.writerow(profile)
        for i in range(0, t_max_hour, 1):
            csv_writer.writerow(data_array[i])

    write_file.close()
#-------------------------------------------------------------------------------
# Here the evaluation is started
# where dt determines the interval in seconds between force-balance calculations
# NOTE: !! the smaller dt, the more accurate the results
# but also the longer the computation time, values of dt should always be
# below 10 seconds, else  !!
# t_max_hour determines the maximum simluation time in hours
# t_max_second determines the maximum simulation time in seconds
# R_number determines the amount of nanoparticle radii used in the calculation
# R_start determines the smallest radius in meters to be used in the calculation
# R[i] gives the radii of the nanoparticles in meters to be evaluated
# location determines where the files will be written to
#-------------------------------------------------------------------------------
def go():
    start = time.time()
    dt = 5
    t_max_hour = 301
    t_max_second = t_max_hour * 3600
    R_number = 31
    R_start = 0.75e-9
    R_interval = 0.25e-9
    R = [None for y in range(R_number)]
    for i in range(0, len(R), 1):
        R[i] = R_start + i*R_interval
    for j in range(0, len(R), 1):
        h, data_array = simulation(R[j], dt, t_max_second)
        location = 'F:/Richard/test/'+ str(dt) + '/equilibrium_dynamic_'+ str(R[j]) \
    +'_nm_dt_'+ str(dt) +'.csv'
        write(location, data_array, t_max_hour)
    end = time.time()
    print('Computation time ' + str(end - start) + ' seconds')
    return
#-------------------------------------------------------------------------------
go()
#-------------------------------------------------------------------------------
```

Appendix V: Equilibrium Profile

With this code the sedimentation-diffusion equilibrium profile in time is calculated, the output is an excel file. The concentration profiles at equilibrium are plotted. The code was written in Python version 2.7, available at http://www.python.org.

```
#-------------------------------------------------------------------------
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.pylab as pylab
params = {'legend.fontsize': 18,
          'figure.figsize': (25, 10),
         'axes.labelsize': 16,
         'axes.titlesize': 18,
         'xtick.labelsize': 16,
         'ytick.labelsize': 16}
pylab.rcParams.update(params)
#-------------------------------------------------------------------------
# Here the magnetic field H (in Tesla) of the magnet is evaluated according
# to Silfhout and Erné, 2019
# Where, ms is the magnet's internal remanent magnetization in Tesla
# d is the thickness of the magnet in meters
# and R is the radius of the magnet in meters
#-------------------------------------------------------------------------
def field(z, M, d, R):
    H = M/2*((z + d)/np.sqrt((z+d)**2+R**2)-(z/np.sqrt(z**2+R**2)))
    return H
#-------------------------------------------------------------------------
# Here the Langevin function is evaluated
# With T the temperature in Kelvin
# kb the boltzmann constant in m^2 kg s^-2 K^-1
#-------------------------------------------------------------------------
def Langevin(ms, D, B):
    T = 293.
    kb = 1.38064852e-23
    Langevin = [None]*(len(B) - 1)
    for i in range(len(B) - 1):
        zeta = (np.pi*(D**3)*ms*B[i])/(6*kb*T) #since zeta is a large number, coth
        if zeta < 14:
            coth = (np.exp(zeta) + np.exp(-zeta))/(np.exp(zeta)-np.exp(-zeta))
        elif zeta >= 14:
            coth = float(1)
        L = coth - 1/zeta
        Langevin[i] = L
    return Langevin
#-------------------------------------------------------------------------
# Here the distribution of the particle diameters, according to a log-
# normal function is evaluated
#-------------------------------------------------------------------------
def log_normal_distribution(D, Dmedian, s):
    beta = np.sqrt(np.log(1+s**2))
    log = 1E-9/(np.sqrt(2*np.pi)*beta*D)*np.exp(-((np.log(D/Dmedian))**2/(2*beta**2)))
    return log
```

```
#-------------------------------------------------------------------------------
# Here the log_normal distribution according to the measurements from the
# TEM measurement are evaluated (btb, bbb and tbt respectively)
# d_number determines the amount of nanoparticle diameters used in the calculation
# d_median determines the median diameter in nm of the log-normal distribution
# s determines the standard deviation in nm of the log-normal distribution
#-------------------------------------------------------------------------------
D = []
log_distribution = []
d_number = 31
d_median = 6.1 #btb
# d_median = 6.7 #bbb
# d_median = 5.6 #tbt
s = 0.42 #btb
#s = 0.5 #bbb
# s = 0.36 #tbt
log_total = 0 #initializing
log_distribution = [None for y in range(d_number)]
D = [None for y in range(d_number)]
log_distribution_norm = [None for y in range(d_number)]
for i in range(d_number):
    D[i] = (1.5 + 0.5*i)
    log_distribution[i] = log_normal_distribution(D[i], d_median, s)
    log_total += log_distribution[i]
for i in range(d_number):
    log_distribution_norm[i] = log_distribution[i]/log_total
#-------------------------------------------------------------------------------
# Here the magnetic forces at all heights on the nanparticles are calculated
# at equilibrium through the magnetic field strenght (B)
# msat_material determines the magnetic saturation of the fluid in A/m
# d_magnet determines the thickness of the magnet in meters
# R_magnet determines the radius of the magnet in meters
# M_magnet determines the magnet's internal remanent magnetization in Tesla
# h_bottom is the distance from the magnet in meters to the bottom of the fluid column
# h_top is the distance from the magnet in meters to the top of the fluid column
# bin_number is the number of bins in height between h_bottom and h_top
# L represent the Langevin function
# V represnts the volume of the nanoparticles
#-------------------------------------------------------------------------------
msat_material = float(4.3e5)
d_magnet = float(0.030)
R_magnet = float(0.0225)
M_magnet = 1.35
h_bottom = 0.01
h_top = 0.02
bin_number = 159 # number of bins (pixels)
bin_size = (h_top - h_bottom) / bin_number
distance_from_magnet = [h_top - (i * bin_size) for i in range(bin_number)]

B = np.zeros(bin_number)
for i in range(bin_number):
    B[i] = field(distance_from_magnet[i], M_magnet, d_magnet, R_magnet)

L = [[] for i in range(len(D))]
for i in range(len(D)):
```

```
    L[i] = Langevin(msat_material, D[i]*(10**(-9)), B)

V = np.zeros(len(D))
for i in range(len(D)):
    V[i] = (1/6.) * np.pi * (D[i]*(10**(-9)))**3

F_mag = [[] for j in range(len(D))] #creats an empty list of lists
for j in range(len(D)):
    lang = L[j]
    for i in range(bin_number - 1):
        dB = B[i] - B[i-1]
        dBdh = dB / bin_size
        force = lang[i - 1] * msat_material * V[j] * dBdh
        F_mag[j].append(force)
#-------------------------------------------------------------------------------
# Here the gravitational force on the nanoparticles at equilibrium is calculated
# with kb the boltzmann constant in m^2 kg s^-2 K^-1
# rho_w the density of water in kg/m^3
# rho_np the density of maghemite/iron oxide in kg/m^3
# g the gravitational acceleration in m/s^2
# T the temperature in Kelvin
# F_grav represents the gravitational force
#-------------------------------------------------------------------------------
kb = 1.38064852e-23
T = float(293)
rho_w = float(997)
rho_np = float(5240)
g = 9.81
F_grav = np.zeros(len(D))
for i in range(len(D)):
    F_grav[i] = (rho_np-rho_w) * V[i] * g
#-------------------------------------------------------------------------------
# Here the equilibrium concentration profiles are calculated from the force
# equations given in appendix I
# dcdh_c represents dc(n)/c(h=n) given in equation I.2
# concentration represents magnetic fluids containing monodisperse nanoparticles
# concentration_lognormal represents the concentration profile of amagnetic
# fluid with polydisperse nanoparticles according to the log_normal function
# determined above
#-------------------------------------------------------------------------------
dcdh_c = [[] for j in range(len(D))]

for j in range(len(D)):
    f_mag = F_mag[j]
    for i in range(bin_number - 1):
        force = f_mag[i] + F_grav[j]
        dcdh_c[j].append(force/(kb*T))

concentration = np.ones([len(D),bin_number])

for j in range(len(D)):
    for i in range(bin_number - 1):
        concentration[j,i+1] = concentration[j,i] + ((dcdh_c[j][i] *
concentration[j,i]) * bin_size)
```

```
for j in range(len(D)):
    conc = concentration[j]
    ratio = sum(conc[1:]) / len(conc[1:])
    for i in range(bin_number):
        concentration[j][i] = concentration[j][i] / ratio


concentration_lognormal = [0 for i in range(bin_number)]
for i in range(bin_number):
    for j in range(len(D)):
        concentration_lognormal[i] += concentration[j][i] * log_distribution_norm[j]
#-------------------------------------------------------------------------------
# Here the equilibrium concentration profile, as well as the log-normal
function of the polydisperse particle distribution are plotted
#-------------------------------------------------------------------------------
plt.subplot(1,2,1)
for i in range(0,13,1):
    plt.plot(distance_from_magnet[1:],concentration[i][1:],label = str(D[i]) + ' nm
diameter')

plt.plot(distance_from_magnet[1:],concentration_lognormal[1:],label = 'Log-normal
Equilibrium model')

plt.xlabel('height (m)')
plt.ylabel('c/c0')
plt.legend(loc=1)
plt.grid()

plt.subplot(1,2,2)
plt.plot(D,log_distribution_norm, label ="Normalized log-normal distribution")
plt.legend(loc=1)
plt.xlabel('Diameter (nm)')
plt.ylabel('P')
plt.grid()

plt.show()
#-------------------------------------------------------------------------------
```

Appendix VI: Average Particle Location in Time

Here the average particle location in time is calculated. The code was written in Python version 2.7, available at http://www.python.org.

```
#-------------------------------------------------------------------------------

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import csv
import os
import re
import matplotlib.pylab as pylab
params = {'legend.fontsize': 18,
          'figure.figsize': (25, 10),
         'axes.labelsize': 16,
         'axes.titlesize': 18,
         'xtick.labelsize': 16,
         'ytick.labelsize': 16}
pylab.rcParams.update(params)
#-------------------------------------------------------------------------------
# This turns numbers into integers instead of strings, other characters remain
# strings
#-------------------------------------------------------------------------------
def tryint(s):
    try:
        return int(s)
    except:
        return s

def numerical_key(s):
    return [tryint(c) for c in re.split('([0.00-9.00]+)', s)]

def get_namelist(location):
    namelist = []
    for file in os.listdir(location):
        if file.endswith('.csv'):
            namelist.append(file)
            namelist.sort(key = numerical_key)
    return namelist

def log_normal_distribution(D, Dmedian, s):
    beta = np.sqrt(np.log(1+s**2))
    log = 1E-9/(np.sqrt(2*np.pi)*beta*D)*np.exp(-((np.log(D/Dmedian))**2/(2*beta**2)))
    return log
#-------------------------------------------------------------------------------
# Here the TEM distribution is inputted
# file_dmeters gives the values for the diameters of the TEM evalution
# simulation_location gives the location of the simulation data
# max_time_hours gives the total time of evaluation in hours
# h_bottom is the distance from the magnet in meters to the bottom of the fluid column
# h_top is the distance from the magnet in meters to the top of the fluid column
# bin_number is the number of bins in height between h_bottom and h_top
```

```
# d_number determines the amount of nanoparticle diameters used in the calculation
# d_median determines the median diameter in nm of the log-normal distribution
# s determines the standard deviation in nm of the log-normal distribution
#-------------------------------------------------------------------------------
def TEM(file_dmeters, simulation_location, max_time_hours):
    dmeter = pd.read_excel(file_dmeters)
    #Bin = dmeter["Bin"]
    bin_number = 159 # number of pixels
    d_number = 31 # number of diameters
    h_bottom = 0.01 #m
    h_top = 0.02 #m
    bin_size = (h_top - h_bottom) / bin_number
    h = [h_top - (i * bin_size) for i in range(bin_number)]
    log_distribution = [None for y in range(d_number)]
    #d_median = 6.1 #btb
    d_median = 6.7 #bbb
    # d_median = 5.6 #tbt
    #s = 0.42 #btb
    s = 0.5 #bbb
    #s = 0.36 #tbt

    D = [None for y in range(d_number)]
    for i in range(0, d_number, 1):
        # in nm
        d = 1.5 + i * 0.5
        D[i] = d
        log = log_normal_distribution(d, d_median, s)
        log_distribution[i] = log

    fr = [None for y in range(d_number)]
    freq = dmeter["Frequency"]
    for i in range(1, d_number + 1, 1):
        freqa = freq[i]
        fr[i -1] = freqa
    c = 0
    fr_total = 0
    log_total = 0
    concentration = [None for y in range(d_number)]
    for i in range(0, d_number, 1):
        concentration[i] = 0.5*D[i]*fr[i] # radius = 0.5*diameter
        c += concentration[i]
        fr_total += fr[i]
        log_total += log_distribution[i]
   # mu = c/fr_total #

    fr_norm = [None for y in range(d_number)]
    concentration_norm = [None for y in range(d_number)]
    for i in range(0, d_number, 1):
        fr_norm[i] =(float(fr[i])/float(fr_total))
        concentration_norm[i] = (concentration[i]/c)

    log_distribution_norm = [None for y in range(d_number)]
    for i in range(d_number):
        log_distribution_norm[i] = log_distribution[i]/log_total
```

```
    namelist = get_namelist(simulation_location)
    full_namelist = [simulation_location + n for n in namelist]

    read = [[] for y in range(d_number)]
    for i in range(d_number):
        with open(full_namelist[i], mode='r') as read_file:
            csv_reader = csv.reader(read_file, dialect ='excel')
            for row in csv_reader:
                read[i].append(row)

            r_norm = [[0 for y in range(bin_number)] for x in range(max_time_hours +
1)]


    for k in range(d_number):
        reading = read[k]
        for j in range(0, max_time_hours+1, 1):
            r = reading[j]
            r_n = r_norm[j]
            for i in range(bin_number):
                r_n[i] += (float(r[i+1]) * log_distribution[k])/log_total
    return h, dmeter, D, fr_norm, r_norm, log_distribution_norm, read, namelist
#-------------------------------------------------------------------------------
#
# r_norm represents the normalized concentration at all heights in time
# location gives the location to which the data will be written
# max_time_hours gives the total time of evaluation in hours
#-------------------------------------------------------------------------------
def write(r_norm, location, max_time_hours):
    with open(location, mode='wb') as write_file:
    # use in word binary mode to prevent an empty row being put after every written
row
        csv_writer = csv.writer(write_file, delimiter =',')
        for i in range(0, max_time_hours, 1):
            csv_writer.writerow(r_norm[i])

    write_file.close()

#def go():
#file_dmeters = "C:/Users/4118308/Documents/excel_files/Diameter_L1_btb.xlsx"
max_time_hours = 301
#file_dmeters = "O:/BETA/Instituut/Debye/FCC/FCC/2.
Students/Richard/Excel/Diameter_L1_btb.xlsx"
file_dmeters = "F:/Richard/Diameter_T1_bbb.xlsx"
#simulation_location = "F:/Richard/Dynamic/excel/dt_5_300h/"
simulation_location = "O:/BETA/Instituut/Debye/FCC/FCC/2.
Students/Richard/Excel/dt_5_300h/"
#simulation_location1 = "F:/Richard/Dynamic/excel/dt_0.1/"
simulation_location1 = "O:/BETA/Instituut/Debye/FCC/FCC/2.
Students/Richard/Excel/dt_0.1/"
#simulation_location2 = "F:/Richard/Dynamic/excel/dt_2/"
simulation_location2 = "O:/BETA/Instituut/Debye/FCC/FCC/2.
Students/Richard/Excel/dt_2/"
```

```
h, dmeter, D, fr_norm, r_norm, log_distribution_norm, read, namelist =
TEM(file_dmeters, \
                                                        simulation_location,
max_time_hours)
h_1, dmeter1, D1, fr_norm1, r_norm1, log_distribution_norm1, read1, namelist1 =
TEM(file_dmeters, \

simulation_location1, \

max_time_hours)
h_2, dmeter2, D2, fr_norm2, r_norm2, log_distribution_norm2, read2, namelist2 =
TEM(file_dmeters, \

simulation_location2, \

max_time_hours)
#location = 'log_normal\equilibrium_dynamic_0.75_nm_to__dt_5_norm.csv'
#write(r_norm, location, max_time_hours)
"""
plt.plot()
plt.subplot(1, 2, 1)
plot = plt.plot(D, fr_norm, label="TEM distribution")
plt.axis([0, 18, 0, 0.15])
plt.ylabel('Normalized Frequency')
plt.xlabel('diameter (in nm)')
plt.grid()

 plt.subplot(1,2,1)
plt.plot(D, log_distribution)
plt.axis([0, 10, 0, 0.4])

plt.subplot(1,2,1)
plt.plot(D, log_distribution_norm, label="log-normal distribution")
plt.legend()
"""
"""
plt.subplot(1, 4, 1)
for i in range(0, 145, 24):
    plot = plt.plot(h, r_norm[i], label = str(i)+' hours')
plt.xlabel('Height above the magnet (in m)')
plt.ylabel('c/c0')
plt.legend()
plt.grid()
plt.show()"""
#-----------------------------------------------------------------------------
#
# with max_time determining the duration of the evaluation in seconds
# interval determining what the time interval is between saved profiles in seconds
# h_bottom is the distance from the magnet in meters to the bottom of the fluid column
# h_top is the distance from the magnet in meters to the top of the fluid column
# bin_number is the number of bins in height between h_bottom and h_top
#
#-----------------------------------------------------------------------------
max_time = 302*3600
interval = 3600
```

```
h_bottom = 0.01 #m
h_top = 0.02 #m
bin_number = 159
bin_size = (h_top - h_bottom) / bin_number

plt.subplot(1, 3, 3)
for k in range(5, 31, 4):
    profile = read[k]

    time = [None]*(int(max_time/interval))
    for i in range(0, int(max_time/interval), 1):
        time[i] = i + 1
        concentration_average = [None for y in range(0, int(max_time/interval), 1)]
    C_initial = 0
    c_initial = profile[0]
    for j in range(1, bin_number + 1, 1):
        C_initial += float(c_initial[j])*h[j-1]
    for i in range(0, int(max_time/interval), 1):

        c = profile[i]
        C = 0
        for j in range(1, bin_number + 1, 1):
            C += float(c[j])*h[j-1]
        concentration_average[i] = C/(C_initial)*0.5*h_bottom

    plot = plt.plot(time, concentration_average, label = str(0.5*k+1.5) +' nm')
plt.title("Average Particle location, dt = 5")
plt.axis([0, 300, 0.0024, 0.0050])
plt.xlabel('time(in hours)')
plt.legend(loc=4) #lower-right location of legend
plt.grid()

plt.subplot(1, 3, 1)
for k in range(5, 31, 4):
    profile = read1[k]

    time = [None]*(int(max_time/interval))
    for i in range(0, int(max_time/interval), 1):
        time[i] = i + 1
        concentration_average = [None for y in range(0, int(max_time/interval), 1)]
    C_initial = 0
    c_initial = profile[0]
    for j in range(1, bin_number + 1, 1):
        C_initial += float(c_initial[j])*h[j-1]
    for i in range(0, int(max_time/interval), 1):

        c = profile[i]
        C = 0
        for j in range(1, bin_number + 1, 1):
            C += float(c[j])*h[j-1]
        concentration_average[i] = C/(2.39)*0.5*h_bottom

    plot = plt.plot(time, concentration_average, label = str(0.5*k+1.5) +' nm')
plt.title("Average Particle location, dt = 0.1")
plt.axis([0, 300, 0.0024, 0.0050])
```

```
plt.ylabel('Height (in m)')
plt.xlabel('time(in hours)')
plt.legend(loc=4) #lower-right location of legend
plt.grid()

plt.subplot(1, 3, 2)
for k in range(5, 31, 4):
    profile = read2[k]

    time = [None]*(int(max_time/interval))
    for i in range(0, int(max_time/interval), 1):
        time[i] = i + 1
        concentration_average = [None for y in range(0, int(max_time/interval), 1)]
    C_initial = 0
    c_initial = profile[0]
    for j in range(1, bin_number + 1, 1):
        C_initial += float(c_initial[j])*h[j-1]
    for i in range(0, int(max_time/interval), 1):

        c = profile[i]
        C = 0
        for j in range(1, bin_number + 1, 1):
            C += float(c[j])*h[j-1]
        concentration_average[i] = C/(2.39)*0.5*h_bottom

    plot = plt.plot(time, concentration_average, label = str(0.5*k+1.5) +' nm')
plt.title("Average Particle location, dt =2")
plt.axis([0, 300, 0.0024, 0.0050])
plt.xlabel('time(in hours)')
plt.legend(loc=4)
plt.grid()
```

Appendix VII: Dynamic Concentration Profiles

Here the concentration profiles in time are calculated and plotted. The code was written in Python version 2.7, available at http://www.python.org.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import csv
import os
import re

import matplotlib.pylab as pylab
params = {'legend.fontsize': 40,
          'figure.figsize': (15, 5),
         'axes.labelsize': 34,
         'axes.titlesize': 34,
         'xtick.labelsize': 32,
         'ytick.labelsize': 32}
pylab.rcParams.update(params)

#-------------------------------------------------------------------------------
# This turns numbers into integers instead of strings, other characters remain
# strings
#-------------------------------------------------------------------------------
def tryint(s):
    try:
        return int(s)
    except:
        return s

def numerical_key(s):
    return [tryint(c) for c in re.split('([0.00-9.00]+)', s)]

def get_namelist(location):
    namelist = []
    for file in os.listdir(location):
        if file.endswith('.csv'):
            namelist.append(file)
            namelist.sort(key = numerical_key)
    return namelist

def log_normal_distribution(D, Dmedian, s):
    beta = np.sqrt(np.log(1+s**2))
    log = 1E-9/(np.sqrt(2*np.pi)*beta*D)*np.exp(-((np.log(D/Dmedian))**2/(2*beta**2)))
    return log
#-------------------------------------------------------------------------------
# Here the TEM distribution is read
# d_number determines the amount of nanoparticle diameters used in the calculation
# h_bottom is the distance from the magnet in meters to the bottom of the fluid column
# h_top is the distance from the magnet in meters to the top of the fluid column
# d_median determines the median diameter in nm of the log-normal distribution
# s determines the standard deviation in nm of the log-normal distribution
```

```
#-------------------------------------------------------------------------------
def TEM(file_dmeters, simulation_location):
    dmeter = pd.read_excel(file_dmeters)
    #Bin = dmeter["Bin"]
    step_number = 155 # number of pixels
    d_number = 31 # number of diameters
    h_bottom = 0.01 #m
    h_top = 0.02 #m
    bin_size = (h_top - h_bottom) / step_number
    h = [h_top - (i * bin_size) for i in range(step_number)]


    log_distribution = [None for y in range(d_number)]
    d_median = 6.1 #btb
    #d_median = 6.7 #bbb
    #d_median = 5.6 #tbt
    s = 0.42 #btb
    #s = 0.5 #bbb
    #s = 0.36 #tbt

    D = [None for y in range(d_number)]
    for i in range(0, d_number, 1):
        # in nm
        D[i] = 1.5 + i * 0.5
        log_distribution[i] = log_normal_distribution(D[i], d_median, s)

    fr = [None for y in range(d_number)]
    freq = dmeter["Frequency"]
    for i in range(1, d_number + 1, 1):
        freqa = freq[i]
        fr[i -1] = freqa
    c = 0
    fr_total = 0
    log_total = 0
    concentration = [None for y in range(d_number)]
    for i in range(0, d_number, 1):
        concentration[i] = 0.5*D[i]*fr[i] # radius = 0.5*diameter
        c += concentration[i]
        fr_total += fr[i]
        log_total += log_distribution[i]
   # mu = c/fr_total #

    fr_norm = [None for y in range(d_number)]
    concentration_norm = [None for y in range(d_number)]
    for i in range(0, d_number, 1):
        fr_norm[i] =(float(fr[i])/float(fr_total))
        concentration_norm[i] = (concentration[i]/c)

    log_distribution_norm = [None for y in range(d_number)]
    for i in range(d_number):
        log_distribution_norm[i] = log_distribution[i]/log_total
#-------------------------------------------------------------------------------
# This gives the location of the folder where the excel files obtained from
# simulation are located
#-------------------------------------------------------------------------------
```

```
    namelist = get_namelist(simulation_location)
    full_namelist = [simulation_location + n for n in namelist]

    read = [[] for y in range(d_number)]
    for i in range(d_number):
        with open(full_namelist[i], mode='r') as read_file:
            csv_reader = csv.reader(read_file, dialect ='excel')
            for row in csv_reader:
                read[i].append(row)

    poly_sed = [[0. for y in range(step_number)] for x in range(302)]

    for k in range(d_number):
        read_monodisperse = read[k]
        for j in range(0, 302, 1):
            r = read_monodisperse[j]
            p_s = poly_sed[j]
            for i in range(step_number):
                p_s[i] += float(r[i+1]) * float(log_distribution_norm[k])
    return h, dmeter, D, fr_norm, poly_sed, read, namelist, log_distribution,
log_distribution_norm

def write(poly_sed, location):
    with open(location, mode='wb') as write_file:
    # use in word binary mode to prevent an empty row being put after every written
row
        csv_writer = csv.writer(write_file, delimiter =',')
        for i in range(0, 301, 1):
            csv_writer.writerow(poly_sed[i])

    write_file.close()

#def go():
#file_dmeters = "C:/Users/4118308/Documents/excel_files/Diameter_L1_btb.xlsx"
file_dmeters = "O:/BETA/Instituut/Debye/FCC/FCC/2.
Students/Richard/Excel/Diameter_L1_btb.xlsx"
#file_dmeters = "F:/Richard/Diameter_T1_bbb.xlsx"
simulation_location = "F:/Richard/Dynamic/excel/dt_0.1/"
#simulation_location = "O:/BETA/Instituut/Debye/FCC/FCC/2.
Students/Richard/Excel/dt_5/"
h, dmeter, D, fr_norm, polydisperse_sedimentation_norm, read, namelist, log,
log_distribution_norm = TEM(file_dmeters, \
                                                    simulation_location)
#location = 'log_normal\equilibrium_dynamic_0.75_nm_to__dt_5_norm.csv'
#write(r_norm, location)

plt.subplot(1,1,1)
#plt.plot(H, Absorption_R_norm, label= "equilibrium btb fraction")
for i in range(0, 161, 40):
    plot = plt.plot(h, polydisperse_sedimentation_norm[i], label = 'btb ' + str(i)+'
hours')
plt.ylim((0.5,2.6))
plt.xlabel('Height (m)')
plt.ylabel('c/c0')
plt.legend()
```

```
plt.grid()
plt.show()
```

Appendix VIII: Experimental Sedimentation Evaluation

This code calculates the concentration profiles in time from the images obtained from the experimental setup. The code was written in Python version 2.7, available at http://www.python.org.

```
import os
import re
import matplotlib.pyplot as plt
from PIL import Image
import numpy as np
import matplotlib.pylab as pylab
params = {'legend.fontsize': 40,
          'figure.figsize': (15, 5),
         'axes.labelsize': 34,
         'axes.titlesize': 34,
         'xtick.labelsize': 32,
         'ytick.labelsize': 32}
pylab.rcParams.update(params)
#-------------------------------------------------------------------------------
# This turns numbers into integers instead of strings, other characters remain
# strings
#-------------------------------------------------------------------------------
def tryint(s):
    try:
        return int(s)
    except:
        return s
#-------------------------------------------------------------------------------
# This defines a key for the list.sort() function that will order the entries
# that contain numbers numerically, else file1, file2, file10 would be
# ordered as: file 1, file10, file2
#-------------------------------------------------------------------------------
def numerical_key(s):
    return [tryint(c) for c in re.split('([0-9]+)', s)]
#-------------------------------------------------------------------------------
# This defines a function that gets the location of .png image files
#-------------------------------------------------------------------------------
def get_namelist(location):
    namelist = []
    for file in os.listdir(location):
        if file.endswith('.png'):
            namelist.append(file)
            namelist.sort(key=numerical_key)
    return namelist
#-------------------------------------------------------------------------------
# This defines a function that opens an image file
#-------------------------------------------------------------------------------
def Iopen(pathname):
    try:
        img = Image.open(pathname)

    except IOError: # Not an image
        print('IOError: Not an image')
```

```
    return img
#-------------------------------------------------------------------------------
# This defines a crop function that reduces the Image file to the pixel values
# between a specified box, located within the crop boundery values: left,
# upper, right and lower.
# left gives the x-value of the left boundary points
# upper gives the y-value of the top boundary points
# right gives the x-value of the right boundary points
# lower gives the y-value of the lower boundary points
# Hence the box will have the shape:
# (left, upper)\-----\(right, upper)
#                   \-----\
# (left, lower)  \-----\(right, lower)
#-------------------------------------------------------------------------------
def Icrop(left, upper, right, lower, Image):
    crop = Image.crop((left, upper, right, lower))
    return crop
#-------------------------------------------------------------------------------
# This defines the lambert-beer function with which Absorption (A)  can be
# calculated from the ingoing intensity (I0) and the outgoing intensity (I1)
#-------------------------------------------------------------------------------
def lambert(I1, I0):
    A = -np.log10(I1/I0)
    #b = 4e-3 #m path length
    #extinction_coefficient =
    #A = extinction_coefficient * c * b
    return A
#-------------------------------------------------------------------------------
# This gives a namelist of the locations for the measurements with intervals
# of 20 minutes between images (full_namelist_20min) and the namelist of the
# locations of the measurements with intervals of 4 hours between images
# (full_namelist_4hours)
#-------------------------------------------------------------------------------
namelist = get_namelist("F:/Richard/Experiments/10oct/")

full_namelist = [None for y in range(len(namelist))]
for i in range(len(namelist)):
    full_namelist[i] = "F:/Richard/Experiments/10oct/" \
    +str(namelist[i])
#-------------------------------------------------------------------------------
# This defines the crop boundaries
#-------------------------------------------------------------------------------
left = 280
upper = 200
right = 330
lower = 420
#-------------------------------------------------------------------------------
# Here the images are opened
#-------------------------------------------------------------------------------
image = [None for i in range(len(namelist))]
for i in range(len(namelist)):
    image[i] = Iopen(full_namelist[i])
#image.show() # This shows the specific image

image_crop = [None for y in range(len(namelist))]
```

```
for i in range(len(namelist)):
    image_crop[i] = Icrop(left, upper, right, lower, \
                    image[i])
#-------------------------------------------------------------------------------
# Below the images are split into the red blue and green parts of the image
#-------------------------------------------------------------------------------
R = [None for y in range(len(namelist))]
G = [None for y in range(len(namelist))]
B = [None for y in range(len(namelist))]
array =  [None for y in range(len(namelist))]
G_array = [None for y in range(len(namelist))]
R_array = [None for y in range(len(namelist))]

for i in range(len(namelist)):
    R[i], G[i], B[i] = image_crop[i].split()
    array[i] = np.array(image_crop[i])
    G_array[i] = np.array(G[i])
    R_array[i] = np.array(R[i])
#-------------------------------------------------------------------------------
# Here an verage result is obtained for every horizontal value
#-------------------------------------------------------------------------------
G_array_mean = [None for y in range(len(namelist))]
R_array_mean = [None for y in range(len(namelist))]
array_mean = [None for y in range(len(namelist))]

for i in range(len(namelist)):
    G_array_mean[i] = G_array[i].mean(axis=1)
    R_array_mean[i] = R_array[i].mean(axis=1)
    array_mean[i] = array[i].mean(axis=1)

#-------------------------------------------------------------------------------
"""
# Here the average results are calculated for G_array if it is split into
# n equal horizontal boxes
#-------------------------------------------------------------------------------
nx = 5
G_array_split_initial = np.split(G_array_initial, nx, axis=1)
d_initial = {}
d_average_initial = {}
for m in range(0, nx):
    d_initial["mean_G_{0}".format(m)] = G_array_split_initial[m].mean(axis=1)
    d_average_initial["average_G_{0}".format(m)] =
np.mean(d_initial["mean_G_{0}".format(m)])

G_array_split_final = np.split(G_array_final, nx, axis=1)
d_final = {}
d_average_final = {}
for m in range(0, nx):
    d_final["mean_G_{0}".format(m)] = G_array_split_final[m].mean(axis=1)
    d_average_final["average_G_{0}".format(m)] =
np.mean(d_final["mean_G_{0}".format(m)])
"""
#np.savetxt("C:/Users/4118308/Documents/Webcam/5oct/G_array_split_mean_1.csv",
G_array_split_mean_1, delimiter=",")
```

```
#np.savetxt("C:/Users/4118308/Documents/Webcam/5oct/G_array_split_mean_2.csv",
G_array_split_mean_2, delimiter=",")
#np.savetxt("C:/Users/4118308/Documents/Webcam/5oct/G_array_split_mean_3.csv",
G_array_split_mean_3, delimiter=",")
#np.savetxt("filename.csv", a, delimiter=",") #This saves array n as a csv file
#-------------------------------------------------------------------------------
# These are th values for the intensity of light through the sovlent (water)
# water for respectively the Red, Green and Blue (RGB) values
#-------------------------------------------------------------------------------
Iwater = np.array([202.725, 186.5, 169.22])

#-------------------------------------------------------------------------------
# Below the absorbance is calculated via lambert-beer
#-------------------------------------------------------------------------------
Absorption = [None for y in range(len(namelist))]
Absorption_R = [None for y in range(len(namelist))]
Absorption_G = [None for y in range(len(namelist))]


for i in range(len(namelist)):
    Absorption[i] = lambert(array_mean[i], Iwater)
    Absorption_G[i] = lambert(G_array_mean[i], Iwater[1])
    Absorption_R[i] = lambert(R_array_mean[i], Iwater[0])

#-------------------------------------------------------------------------------
# Below the normalized dynamic concentration profiles are calculated
#-------------------------------------------------------------------------------
A = [0 for y in range(len(namelist))]

for j in range(len(namelist)):
    absorption_R = Absorption_R[j]
    for i in range(220):
        A[j] += absorption_R[i]/220 #taking the initial average concentration to be
the average of all locations

Absorption_R_averaged = [[None for y in range(220)] for x in range(len(namelist))]
Absorption_R_norm_averaged = [[None for y in range(len(Absorption_R))] for x in
range(len(namelist))]

for j in range(len(full_namelist)):
    absorption_R_averaged = Absorption_R_averaged[j]
    absorption_R = Absorption_R[j]
    for i in range(220):
        absorption_R_averaged[i] = absorption_R[i]/A[j]

Absorption_R_norm = [[None for y in range(220)] for x in range(len(namelist))]
Absorption_R_norm_averaged = [[None for x in range(220)] for y in
range(len(namelist))]
total_averaged = [0 for i in range(len(namelist))]
total = [0 for i in range(len(namelist))]

for j in range(len(namelist)):
    absorption_R_averaged = Absorption_R_averaged[j]
    absorption_R = Absorption_R[j]
    for i in range(len(Absorption_R)):
```

```
        total_averaged[j] += absorption_R_averaged[i]
        total[j] += absorption_R[i]

for j in range(len(namelist)):
    absorption_R_norm_averaged = Absorption_R_norm_averaged[j]
    absorption_R_averaged = Absorption_R_averaged[j]
    absorption_R_norm = Absorption_R_norm[j]
    absorption_R = Absorption_R[j]
    for i in range(220):
        absorption_R_norm_averaged[i] = 220*absorption_R_averaged[i]/total_averaged[j]
        absorption_R_norm[i] = 220*absorption_R[i]/total[j]
#-------------------------------------------------------------------------------
# Here the height for the concentration profiles is evaluated
# pixel_botom indicates the amount of pixels to the bottom of the cuvette
# h_bottom is the distance from the magnet in meters to the bottom of the fluid column
# h_max is the distance from the magnet in meters to the top of the evaluated fluid
column
# bin_number is the number of bins in height between h_bottom and h_top
#-------------------------------------------------------------------------------
pixel_bottom = 8
h_bottom = 0.01
h_max = 0.025365
bin_number = 220

bin_size = (h_max - h_bottom) / bin_number
distance_from_magnet = [h_max - (i * bin_size) for i in range(bin_number)]
pixel_size = 0.022/315 #m
length_number = lower - upper
h_length = length_number*pixel_size
h_lowest_pixel = pixel_bottom*pixel_size + 0.01
H = [None for y in range(220)]

for i in range(220):
    H[i] = (h_length-i*pixel_size) + h_lowest_pixel - pixel_size
#-------------------------------------------------------------------------------
# Here the concentration profiles are plotted
#-------------------------------------------------------------------------------
plt.subplot(1,1,1)
for i in range(0, 10, 1):
    plt.subplot(1,1,1)
    plot = plt.plot(H, Absorption_R_norm[i], label="btb " + str(8*i+8) + " hours")
plt.xlabel('height (m)')
plt.ylabel('c/c0')
plt.legend()
plt.grid()
plt.show()
#-------------------------------------------------------------------------------
```