Modeling Magnetorheological Fluids

CHEM 26800 Computational Chemistry and Biology

Sprint 2014

Joshua Miller

Abstract

The purpose of this project was to create a simple model of a magnetorheological fluid in order to examine the structural trends as a function of magnetic to non-magnetic particle ratio and external field strength. The rheological properties were examined by testing the ability of a massive projectile to penetrate the fluid.

Introduction

Magnetorheological fluids, similar to their ferrofluid cousins, have the ability to be controlled by magnetic fields. Composed of a micrometer sized magnetic particles suspended in a non-magnetic medium, the application of an external magnetic field induces two conformational changes. A magnetic particle experiences an interaction force with those surrounding it, but the random alignment of the magnetic poles prevents this force from creating a global structure amongst them. With the application of an external magnetic field, the torsion aligns the magnetic moments along the field. This uniformity causes the magnetic pole interactions to create chain-like structures amongst the magnetic particles. These chain structures increase the apparent magnitude of the fluid and may cause the fluid to become a viscoelastic solid. This property gives MR fluids applications in vehicle shock absorbers, prosthetics, and impact resistant materials. The goal of creating a model of MR fluids was to look at trends in its rheological properties by examining the viscosity under varying magnetic field strengths as well as varying magnetic to non-magnetic particle ratios.

The simulation was written in C and is available at `github.com/millerjs/MagRheol`. There is a preprocessor definition that relies on `GNU_SOURCE` and therefore the code has only been tested on a GNU linux machine. In order to run the simulation, the code must be compiled. This can be accomplished by running `make` in the root directory of the software. To run the software, specify a configuration file as such: `./magrheol mag.config`. The first 6 columns of the output specify the checkpoint number, the current time, the internal energy, and the $\hat{x}, \hat{y}, \hat{z}$ components of the average magnetic moment. The code also outputs

a log file `magrheol.log` and three different data files at each checkpoint interval containing information about (1) all the particles (2) the magnetic particles and (3) the projectile into the directory `checkpoints`.

Fluid Model

The model presented on this paper is based on the work of Y. Ido, T. Inagaki, and T. Yamaguchi[1]. For simplicity, the model fluid is composed of uniform, hard, spherical particles suspended in a carrier fluid. The magnetic particles are approximated to be dipoles. Particle parameters used in this simulation mirror those in the Ido simulation. The magnetic material is approximated to be $ZnFe_2O_4$ in particles of radius 1.5μm with a density of $\rho = 2.5 \times 10^{-4}$ kg m^{-3}. The non-magnetic particles maintain the same properties aside from the lack of magnetic dipole moment.

Domain and Magnetic Field

This simulation ignores the effect of the fluid on the magnetic field, and the field strength and direction is considered constant throughout the domain. The domain was constructed by selecting the number of particles used in the simulation, a cubic number for convenience, and using this to determine the domain size. For example, for a simulation with 343 particles, the domain volume was approximated as

$$\begin{aligned}
\mathrm{V} &= \frac{\text{no. of particles}}{\text{particle dens}} \\
\text{particle dens} &= \frac{\rho}{m_{\text{particle}}} \\
m_{\text{particle}} &= m_{\text{molecule}} \times \frac{\text{molecules}}{\text{particle}} \\
m_{\text{molecule}} &= \frac{m_{ZnFe_2O_4}}{N_A} \\
\frac{\text{molecules}}{\text{particle}} &\approx \frac{r^3_{particle}}{r^3_{molecule}}
\end{aligned}$$

This calculation provided for a cubic domain on the order of 40μm in length. The external magnetic field was $-H\hat{z}$ across the domain. During the initial creation of the model the boundary condition `PERIODIC` was applied to all faces of the domain. All distances calculated for directions with periodic faces used the minimum image convention. I found that the periodic boundary condition was especially important for the faces perpendicular to magnetic field. This not only allowed the fluid to behave as if it were in a larger domain but also prevented the truncation of particle chains near the boundaries that would be seen in a non-physically small domain from a lack of dipole attraction across the border. This condition is necessary because the dipole interaction force was not mirrored across

a reflecting boundary condition in this simulation. The `REFLECTING` boundary condition was created simply by reflecting the velocity of particles crossing the face.

The code reads in a parameter config file to avoid compilation for every new initial setup. Included below is a sample config file:

```
# magrheol config file
# maxt in 1e-4 seconds
maxt=                 12.
checkpoint_interval=  50
dt=                   .001
# Particle counts/ratio
npart=                343
ratio=                .5
# dimensions in micrometers
X=                    4.223095e+01
Y=                    4.223095e+01
Z=                    4.223095e+01
# Magnetic field and dipole
# mTesla
H=                    70
MU=                   .01
```

Forces/Torques

Due to the reduction of the magnetic particles to spheres with dipole moments, the forces needed to simulate the fluid included F_{rep}, the physical repulsion force, $F_{\text{Dipole-Dipole}}$, the dipole interaction between magnetic particles, and F_{drag}. Because the particles are on the micrometer scale, the random force due to Brownian motion was considered negligible. The forces used were as derived in the Ido paper[1]. The repulsion force was based on the DLVO theory which is derived from the van der Waal's attraction and the Double Layer force caused by charge dissociation at particle surfaces.

$$F_{rep} = \frac{3\mu_s^2}{64\pi\mu_0 R^4} e^{-\kappa S} \frac{\vec{r}}{||\vec{r}||}$$

where S is the distance between the surfaces of the two particles [1] and is the inverse Debeye screening length, with a value provided in the Ido paper[1]. The Dipole-Dipole interaction force was

$$F_{Dipole} = \frac{3}{4\pi\mu_0||\vec{r}||^4}\left[(\mu_i \cdot \mu_j)\frac{\vec{r}}{||\vec{r}||} - 5(\mu_i \cdot \vec{r})(\mu_j \cdot \vec{r})\frac{\vec{r}}{||\vec{r}||} + ((\mu_i \cdot \vec{r})\mu_j + (\mu_j \cdot \vec{r}\mu_i))\,\frac{1}{||\vec{r}||}\right].$$

[1] This repulsion convention was novel to me. I found it extremely interesting because of its generality over surface topologies.

The drag force was applied with Stokes' drag,

$$F_{drag} = -6\pi\eta r\vec{v},$$

where the particle radius was used for the Stokes radius. In order for the particles to align to the magnetic field, the must experience a torque. The torque on a magnetic dipole in an external field is

$$T_H = \vec{\mu} \times \vec{H}.$$

Furthermore each magnetic particle experiences a torsion force from the dipoles around it,

$$T_{Dipole} = \frac{1}{4\pi\mu_0||\vec{r}||^3}\left(\mu_i \times \mu_j - \frac{3}{||\vec{r}||^2}(\mu_j \cdot \vec{r})\mu_i \times \vec{r}\right).$$

Evolving the state

Initially, populating to domain with particles was done randomly. The random placement of the particles started the simulation with too high of an initial energy because of perticle overlap. Monte Carlo placement of the particles was ruled out as unnecessarily costly. Therefore, the particals in all simulationes are initialized along an equally spaced grid. Whether or not a particle is a dipole is determinite by linear random sampling with the given ratio ϕ.

Both the positions and the orientation of the dipole moment were evolved using Verlet integration. Because of the long evolution time and the small time step that were necessary, I decided to parallelize the verlet algorithm in order to utilize any available cores. I created a pool of `pthreads`. Each thread performs $\frac{1}{N}$th of the force and angle verlet integration where N is the number of threads set to be used. The inclusion of this scheme necessitated an update in the integration. Due to the race condition introduced by the threading, the data belonging to each thread was not guaranteed to be at the right time step when viewed by another thread. To handle this, the data was stored in pre-allocated temporary arrays. Each thread utilized a thread barrier to wait until all threads had updated their positions to a temporary memory location before the master thread performed calculations involving projectile position and performed output writes to checkpoints. After the thread barrier was reached, the variables pointing to the (old and new) position and dipole vectors were rotated. Using 16 cores I was able to achieve approximately a 1200% speedup. Even with this parallelization, simulations were relatively costly, making it difficult to verify parameters such as the time step and gather quantitative data on the rheological properties of the fluid.

In order to test the rheological properties of the fluid, an attempt was made to accelerate a massive particle into the domain, propelled by a small force. This process is akin to measuring the viscosity of a fluid by calculating the rate of descent of an object under a gravitational field. The strength of the external force on the projectile was determined empirically by testing which forces provided stable motion at a reasonable time step while still eventually overcoming

any Brownian opposition force on the projectile by the simulation particles. The intent was to use the time required for the particle to travel through the domain was used as a metric of the rheological properties given a magnetic field strength and particle ratio. However, this metric was unreliable due the random nature of the particles and the comparatively similar size of the projectile and the particles. Instead the k-means clustering algorithm was used to analyze the chaining characteristics of the fluid.

Results

Figures 1 & 2 demonstrate the initial state $t = 0$ms and the final state $t = 2.0$ms for a simulation with 343 particles.

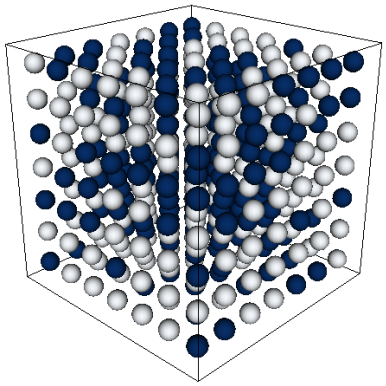

Figure 1: Initial Setup with magnetic (blue) and non-magnetic (white).

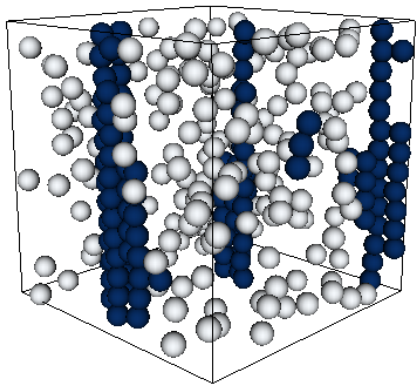

Figure 2: Final state after $t = 2.0$ms. Magnetic (blue) and non-magnetic (white).

In order to quantitatively measure the convergence pattern of the chaining effects the k-means clustering algorithm[4] was used to ascribe a chain to the magnetic particle locations projected onto X-Y plane (as it is perpendicular to the magnetic field). For a simulation with a cubic domain of side length 42.2μm, there can be 14 molecules chained along the magnetic field direction in the domain. Therefore, for 343 particles, we would expect to see approximately 13 clusters. Figure 4 below demonstrates the distortion (the displacement of each point to the centroid it has been classified to) of the points after applicaiton of the k-means algorithm as a function of time.

An exponential function was applied using a linear error regression of the form Distortion $= Ae^{-Bt^C} + D$. For this analysis we are interested in the decay power C. For the examples above, the decay rate $C = 1.54$ and 2.15 for field strengths of $H = 1$mT and 1000mT respectively.

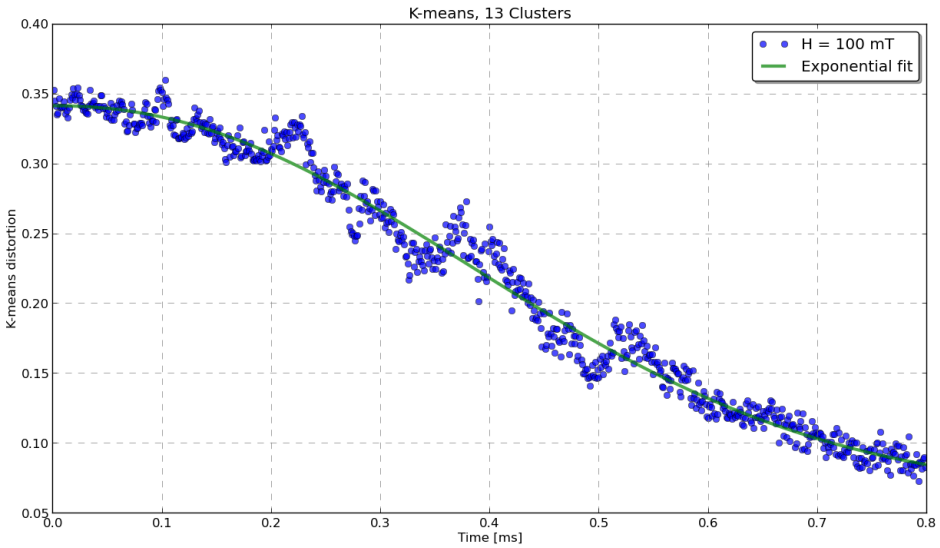

Figure 3: K-Means clustering distortion over time, $K = 13$, $H = 1$ mT, $dt = 1 \times 10^{-6}$ ms

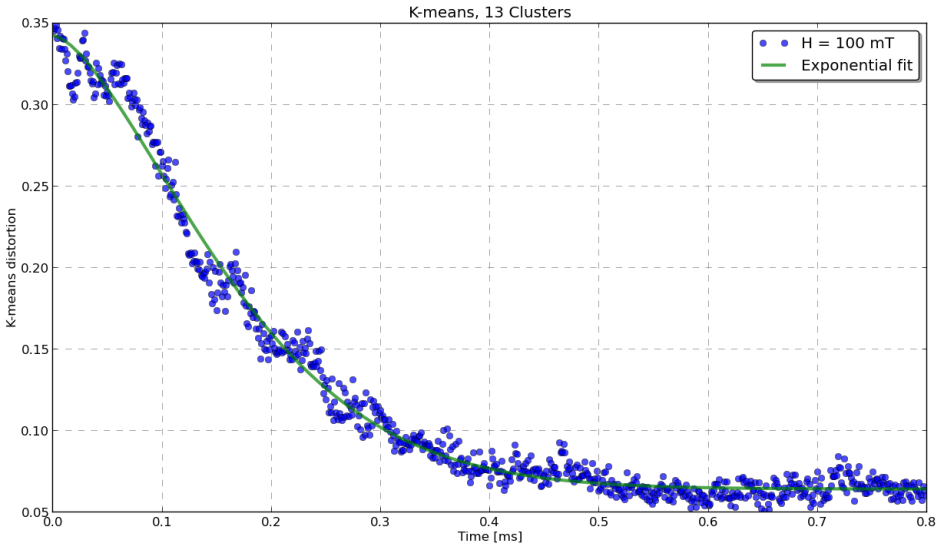

Figure 4: K-Means clustering distortion over time, $K = 13$, $H = 1000$ mT, $dt = 1 \times 10^{-6}$ ms

Figure refrates

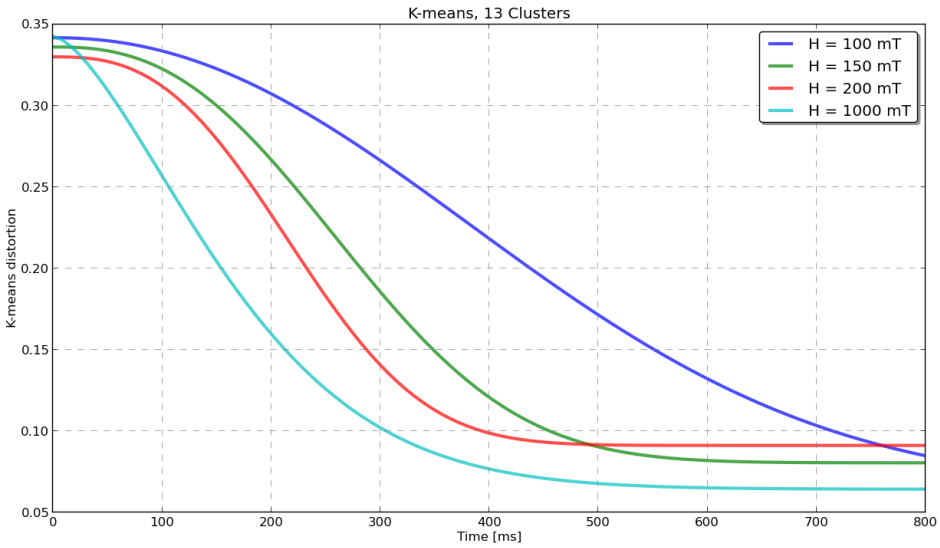

Figure 5: Clustering rate comparison for various magnetic field strengths

Conclusions

In general, the most challenging part of this project was debugging. A large difficulty arose in finding an error in indexing which wrote the Dipole-Dipole force to the wrong particles. Given that the forces used were derived in the Ido paper[1] the only additional force required was the general stokes viscosity.

References

[1] Y. Ido, T. Inagaki, and T. Yamaguchi. *Numerical simulation of microstructure formation of suspended particles in magnetorheological fluids.* Journal of Physics: Condensed Matter. IOP Publishing. 15 July 2010.

[2] Spinks, Joseph Michael. *Dynamic simulation of particles in a magnetorheological fluid.* PhD diss., Monterey California. Naval Postgraduate School, 2008.

[3] ViSit Visualization, animation and analysis tool. `https://wci.llnl.gov/codes/visit/`

[4] SciPy k-means. `http://docs.scipy.org/doc/scipy-0.13.0/reference/generated/scipy.cluster.vq.kmeans.html`. The Scipy community. Copyright 2008-2009.