Continuum Modeling of Particle Suspension Conductivity

by

Tyler J. Olsen

Submitted to the Department Mechanical Engineering
in partial fulfillment of the requirements for the degree of

Master of Science in Mechanical Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2015

Author ..... Signature redacted
Department Mechanical Engineering
August 19, 2015

Certified by ..... Signature redacted
Kenneth N. Kamrin
Assistant Professor
Thesis Supervisor

Accepted by ..... Signature redacted
David E. Hardt
Ralph E. and Eloise F. Cross Professor in Manufacturing
Chairman, Committee on Graduate Students

Continuum Modeling of Particle Suspension Conductivity

by

Tyler J. Olsen

Submitted to the Department Mechanical Engineering
on August 19, 2015, in partial fulfillment of the
requirements for the degree of
Master of Science in Mechanical Engineering

Abstract

A suspension of network-forming, electrically conductive particles imparts electrical conductivity to an otherwise insulating medium. This effect can be used to great effect in many industrial applications. The ability to describe these networks and to predict their physical properties is a key step in designing systems that rely on these properties. In addition, many times these networks are suspended in a flowing fluid, which disrupts existing networks and forms new ones. The extra layer of complexity introduced by flow requires more sophisticated tools to model the effect on the network and its properties.

In the first chapter, we derive a model for the full, tensorial effective conductivity of a particle particle network as a function of a local tensor description of the particle network, the "fabric tensor." We validate our model against a large number of computer-generated networks and compare its performance against an analogous existing model in the literature. We show that the model accurately predicts the isotropic magnitude, deviatoric magnitude, and deviatoric direction of a particle network.

In the second chapter, we set out to model the effects of flow on a particle network. We propose two frame-indifferent constitutive equations for the evolution of the fabric tensor. We perform conductivity measurements of real flowing carbon black suspensions and fit our models to the results by using the conductivity model derived in chapter 1. We find that our models are able to reproduce out-of-sample experimental results with a high degree of accuracy.

Thesis Supervisor: Kenneth N. Kamrin
Title: Assistant Professor

Acknowledgments

First and foremost, I would like to thank my advisor Professor Ken Kamrin for his support, mentorship, and (seemingly infinite) patience. His enthusiasm, knowledge, and experience have been invaluable throughout the course of this work. I am excited to be able to continue my work as a Ph.D. candidate under his guidance.

I would also like to thank my labmates Sachith, Ramin, Hesam, Patrick, Chen-Hung, Qiong, and Jake. I couldn't imagine a better group of colleagues with whom to be down in the trenches.

Last, and certainly not least, I would like to thank my wife Sarah. She is always there pushing me when I become complacent, motivating me when I get discouraged, and cooking dinner when I get too caught up in work to feed myself.

Contents

List of Figures

List of Tables

Chapter 1

Modeling Tensorial Conductivity of Particle Suspension Networks

Chapter 1 largely appeared in a publication by Olsen and Kamrin [34]. The chapter here contains additional details.

1.1 Summary

Significant microstructural anisotropy is known to develop during shearing flow of attractive particle suspensions. These suspensions, and their capacity to form conductive networks, play a key role in flow-battery technology, among other applications. Herein, we present and test an analytical model for the tensorial conductivity of attractive particle suspensions. The model utilizes the mean fabric of the network to characterize the structure, and the relationship to the conductivity is inspired by a lattice argument. We test the accuracy of our model against a large number of computer-generated suspension networks, based on multiple in-house generation protocols, giving rise to particle networks that emulate the physical system. The model is shown to adequately capture the tensorial conductivity, both in terms of its invariants

and its mean directionality.

1.2 Introduction

The electrical conductivity of heterogeneous materials has been extensively studied by many different researchers over the years [5, 11, 44, 42, 26]. The literature primarily focuses on heterogeneous materials which are mixtures of two materials that each have different, isotropic electrical conductivities. The most well-known result is that of Maxwell, which is based on an effective-medium approximation for dilute suspensions [30]. Hashin and Shtrikman approached the problem in a different way. Rather than attempt to solve for an exact expression for the effective conductivity of a randomly structured material, they applied a variational method to derive upper and lower bounds on the effective conductivity [22]. They chose to use a variational approach to derive bounds on the conductivity because solving the exact problem for an arbitrarily structured heterogeneous material was analytically intractable. Torquato [44, 42, 43] has studied the effective conductivity problem in great depth. He has improved the bounds laid out by Hashin and Shtrikman, has solved for effective conductivity of a number of different lattice types, and has expressed the exact tensorial effective conductivity in terms of an infinite series of N-point probability functions, which can be used to describe the microstructure of a heterogeneous material. The particular case of a suspension consisting of a conductive particle network within an insulating medium has been considered theoretically, to our knowledge, in one existing study [26]. The approach they take assumes a spatially homogeneous potential gradient field imposed upon the structure, leading to a model for the conductivity that can be proven to be an upper bound.

Much of the aforementioned work is concerned with the isotropic conductivity of heterogeneous materials. In this work, we aim to model the full tensorial conductivity, with a focus on suspended networks of conductive particles. These particle networks are of practical importance, especially in flowable battery technology currently un-

der development by the Joint Center for Energy Storage Research (JCESR) [14]. In these batteries, a conductive, flowing suspension of carbon black forms an integral component of the system, see Figure 1-1(a). It has been shown in related systems [25] that shearing flows induce anisotropy in a contact network of suspended particles, as pictured in Figure 1-1(b). In instances where suspension conductivity arises from particle-particle contacts, this structure anisotropy should give rise to conductivity anisotropy. It is this behavior that we seek to describe. It has been shown experimentally that the electrical conductivity of a suspension is highly sensitive to shear rate[2], dropping by several orders of magnitude as shear rate increases. From this observation and the evidence of particle microstructure changing in shearing flow, we deduce that a suitably chosen description of the particle network should be sufficient to predict the electrical conductivity of a suspension.

In the granular media literature, a great deal of attention has been given to describing the structure of the contact network between particles. Perhaps the simplest structural measure for such a network that includes anisotropy is the *fabric tensor* [33, 31, 37]. While more complex structural measures exist, such as pair- and higher-order particle correlation functions[44], whose use could enable greater accuracy in constructing a conductivity model, we shall show that a suitable model can be achieved solely in terms of the fabric. Key to our model development is the solution of a simple case, based on a network conforming to a lattice structure. The results instruct the form for a new conductivity model, whose accuracy is then tested against many thousands of random particle networks. To explore a range of particle networks, we describe two distinct algorithms for creating random packings — one for denser packings, and one for more dilute packings that closely resemble those formed by carbon-black — and demonstrate the model's predictive capability against thousands of packings generated from both algorithms.

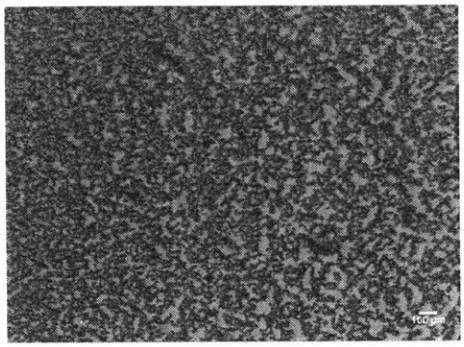
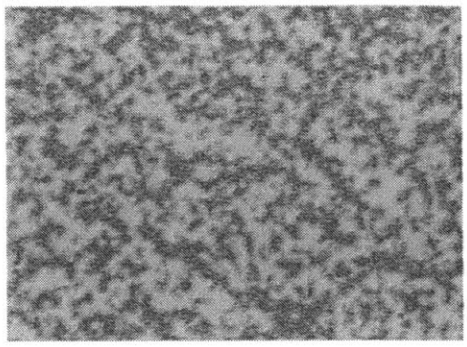
(a)  (b) 

Figure 1-1: (a) Image of a carbon black particle network, an electrically conductive suspension [24]. (b) Image of an effective two-dimenionsional suspension (attractive polystyrene beads on a fluid surface), which has been subjected to shearing. Note the formation of an anisotropic contact network between particles. [25]

1.3 Homogenization

The tensorial form of Ohm's law relates the electric field vector $\mathbf{E}$ to the current density vector $\mathbf{J}$ through a second-order conductivity tensor $\mathbf{K}$, i.e.

$$\mathbf{J} = \mathbf{K}\mathbf{E} \tag{1.1}$$

The conductivity tensor is a symmetric, positive-definite tensor [43]. An effective conductivity for a representative volume Ω of a heterogeneous material must be defined prior to any analytical or numerical work. The effective conductivity of an ergodic medium is defined by

$$\langle \mathbf{J} \rangle = \mathbf{K}\langle \mathbf{E} \rangle \tag{1.2}$$

where $\langle \mathbf{E} \rangle$ and $\langle \mathbf{J} \rangle$ are, respectively, the spatially-averaged electric and current density fields over Ω [43]. To avoid a possibly over-reaching assumption of ergodicity — our tests will be conducted on finite domains — we specify that $\langle \mathbf{E} \rangle$ is imposed by prescribing a linear boundary potential $\varphi(\mathbf{x} \in \partial\Omega) = -\langle \mathbf{E} \rangle \cdot \mathbf{x}$, and that $\langle \mathbf{J} \rangle$ is redefined as the flux that is power-conjugate to $\langle \mathbf{E} \rangle$. That is,

$$\langle \mathbf{E} \rangle \cdot \langle \mathbf{J} \rangle \equiv \frac{1}{V} \int_{\Omega} -\nabla\varphi \cdot \mathbf{j} \, \mathrm{dV} \tag{1.3}$$

where $\mathbf{j}$ is the local current density field. In the ergodic limit of the ensuing analysis, $\langle \mathbf{J} \rangle$ reduces to a standard spatial average.

Assuming that the current density obeys Kirchoff's current law and Ohm's law — respectively, $\nabla \cdot \mathbf{j} = 0$ and $\mathbf{j} = -\sigma \nabla \varphi$ for some non-negative conductivity field $\sigma(\mathbf{x})$ — a symmetric, positive-definite conductivity tensor $\mathbf{K}$ must exist that obeys (1.2). By using the divergence theorem, Eq 1.3 can be transformed into

$$\langle \mathbf{E} \rangle \cdot \mathbf{K} \langle \mathbf{E} \rangle = \frac{1}{V} \int_{\partial\Omega} -\varphi \mathbf{j} \cdot \mathbf{n} \, \mathrm{dA} \tag{1.4}$$

where $\mathbf{n}$ is the outward-pointing normal vector.

We model the particles as perfect conductors, the fluid as a perfect insulator, and we suppose electrical resistance arises only at the contacts between particles. Likewise, the field φ is approximated as a constant within each particle but possibly varying from particle to particle. The above integral can now be broken into a sum of integrals over the boundary. In the locations where the boundary passes through free space (i.e., not a particle), then we know that $\mathbf{j}$ is exactly $\mathbf{0}$. This leaves only the parts of the boundary that pass through particles, which allows us to write the integral over the set of boundary particles B, i.e.

$$\langle \mathbf{E} \rangle \cdot \mathbf{K} \langle \mathbf{E} \rangle = \frac{1}{V} \sum_{i \in B} -\varphi_i \int_{\partial\Omega_i} \mathbf{j} \cdot \mathbf{n} \, \mathrm{dA}. \tag{1.5}$$

where Ω_i is the intersection of the ith boundary particle with $\partial\Omega$, and the potential within particle i, denoted φ_i above, can be brought outside the integral since it is constant within a particle. Although the precise nature of $\mathbf{j}$ is unknown within the particle, the value of the integral $\int_{\partial\Omega_i} \mathbf{j} \cdot \mathbf{n} \, \mathrm{dA}$ is the current that is flowing out of Ω. Denoting this current as I_i^{out} we can write the final expression for the right-hand-side of (1.4),

$$\langle \mathbf{E} \rangle \cdot \mathbf{K} \langle \mathbf{E} \rangle = \frac{1}{V} \sum_{i \in B} -\varphi_i I_i^{out}. \tag{1.6}$$

The three independent components of $\mathbf{K}_e$ can be determined by performing multiple simulations on the same particle network with three non-colinear choices of $\langle \mathbf{E} \rangle$.

By our assumptions for the particle properties, the problem can be reduced further to that of a resistor network. The network is defined by the set of particles acting as the nodes, which are connected by a set of contacts acting as the edges, which carry a resistance R_c. In our model development and simulations, we assume that R_c is a constant at all contacts. In reality, this is not strictly the case since the contact resistance is affected by the contact area between particles and the local structure of the network. This was studied in detail by Batchelor[6]. If the particles in question were Hertzian elastic spheres compressed into a dense granular packing, spatially-fluctuating contact forces arise causing fluctuating contact areas, and this issue may be a factor to consider. However, for the suspensions of interest in this study, the particles actually have an open, fractal structure, and form contacts only due to van der Waals attraction. Absent any information to inform the size of the contact area beside the $\sim$ constant attractive force, we choose to use a constant contact resistance. A schematic of an example network with 9 nodes and 12 edges can be found in figure 1-2. Supposing an N-particle sample and letting $i_{m,n}$ represent the (signed) current flowing from particle m to n, Ohm's and Kirchoff's law can be rewritten in their simpler discrete form,

$$i_{m,n} = \frac{\varphi_m - \varphi_n}{R_c} H_{mn} \tag{1.7}$$

and

$$\sum_n i_{m,n} = 0 \quad \text{for all} \quad m. \tag{1.8}$$

where H_{mn} is the adjacency matrix of the graph formed by the particle network. $H_{mn} = 1$ if there is an edge connecting particles m and n, and zero otherwise. These equations define a sparse linear system that can be solved for the potential at each particle after applying appropriate boundary conditions, which is described in a later section. The resulting linear system is sparse, symmetric, and positive definite, so we used a sparse Cholesky direct solver to compute the potential at each point. A trivial

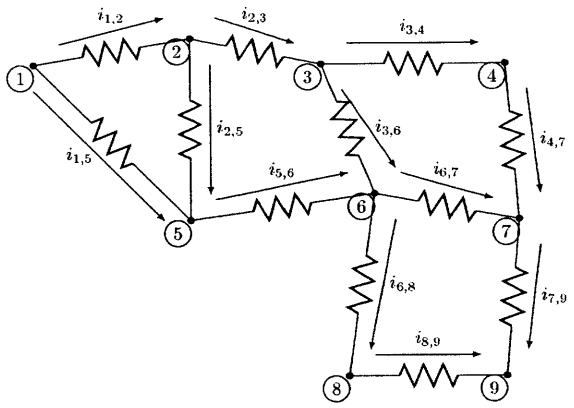


Figure 1-2: Schematic of a small resistor network with nodes and edges labeled according to our conventions.

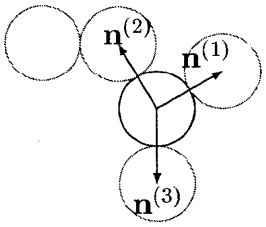


Figure 1-3: Schematic of particles in contact showing contact vectors $\mathbf{n}^{(i)}$.

post-processing step can be performed to compute the current through each contact.

Solving these linear equations for a given particle network enables us to calculate I^{out} in (1.6) and hence the conductivity tensor for the network.

We choose to use the fabric tensor as the measure of the network structure. The particle-level fabric is a local quantity that can be defined for particle p by the relation [37, 33, 31]

$$\mathbf{A^p} = \sum_i \mathbf{n}^{(i)} \otimes \mathbf{n}^{(i)} \tag{1.9}$$

where $\otimes$ denotes the dyadic product, and $\mathbf{n}_i$ is the unit vector connecting particle centroids of the i'th contact on the particle. This is illustrated in Figure 1-3. To homogenize over the entire particle network, or at least meso-sized region of it, the average fabric tensor is defined as the system average of the particle fabric tensors.

$$\mathbf{A} = \frac{1}{N_{\text{particles}}} \sum_{p=1}^{N_{\text{particles}}} \mathbf{A^p} \tag{1.10}$$

The definition of the fabric tensor has some attractive features. It is symmetric and positive-semidefinite, guaranteeing that the eigenvalues are non-negative and that the eigenvectors are orthogonal. These properties are shared by the conductivity tensor $\mathbf{K}$, suggesting the fabric tensor could be an appropriate independent variable in the conductivity's functional form.

An important concept that will be used later is that of a tensor deviator. The tensor deviator is the trace-free part of a tensor, and it is useful to describe anisotropic phenomena since it has the isotropic part removed. It is defined as

$$\mathbf{A}_0 = \mathbf{A} - \frac{1}{d}\mathrm{tr}\mathbf{A}\,\mathbf{1} \tag{1.11}$$

where d is the spatial dimension and $\mathbf{1}$ is the identity tensor.

1.4 Lattice-Reduced Model

We propose an analytical model to elucidate the connection between electrical conductivity and the fabric tensor based on a simplified lattice structure. We will test this model's applicability to random packings in the later sections.

The particles are imagined to live on an idealized infinite, periodic lattice. The lattice is parameterized by a set of numbers that describe the particle size and spacing. These parameters are (1) particle diameter D_p, (2) distance in x-direction between chains d_x, (3) distance in y-direction between chains d_y, (4) distance in z-direction between chains d_z. In 2D, only the first three parameters are used. An illustration of a 2D lattice characterized by these parameters is shown in figure 1-4(a), with its fundamental unit cell shown in figure 1-4(b).

Both the average fabric tensor and effective conductivity can be computed analytically. The average fabric tensor is defined as the spatial average of the fabric tensor

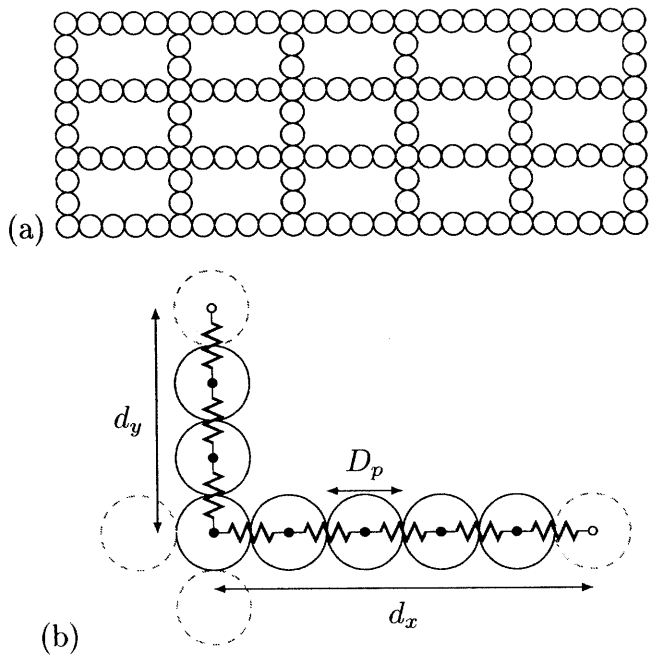


Figure 1-4: Idealized particle lattice and unit cell from which fabric-conductivity relation was derived. (a) Example 2D idealized particle lattice. (b) 2D lattice unit cell and its resistor network analog. Neighboring unit cells are shown in gray dashed lines.

for all of the particles in the unit cell and ultimately results in the formua

$$\mathbf{A} = \frac{2}{N_x + N_y - 1} \begin{bmatrix} N_x & 0 \\ 0 & N_y \end{bmatrix} \tag{1.12}$$

In this expression, the key quantities to recognize are the number of particles in the x-oriented chain, $N_x = d_x/D_p$, and the number of particles in the y-oriented chain, $N_y = d_y/D_p$.

Next, the effective conductivity was derived for the unit cell. To do this, imagine applying an arbitrary voltage difference across the x-oriented and y-oriented chains separately. These voltages are $\Delta\varphi_x$ and $\Delta\varphi_y$, respectively. By applying Ohm's law through the corresponding chains, we can recover the components of the vector form

of Ohm's Law shown in (1.1). For example, for the x-oriented chain

$$j_x = -\left(\frac{1}{N_y R_c}\right)(\nabla\varphi)_x \tag{1.13}$$

with $(\nabla\varphi)_x = \Delta\varphi_x/d_x$. Due to the geometry of the problem, we know that the off-diagonal components of the conductivity tensor $\mathbf{K}$ are exactly zero. Therefore, we can say

$$K_{11} = \frac{1}{N_y R_c}. \tag{1.14}$$

Similarly analysis yields

$$K_{22} = \frac{1}{N_x R_c}. \tag{1.15}$$

Finally, the parameters N_x and N_y can be algebraically eliminated to give the components of $\mathbf{K}$ in terms of the components of $\mathbf{A}$, yielding the tensorial relationship

$$\mathbf{K} = \frac{1}{R_c}\frac{\mathrm{tr}\mathbf{A} - 2}{\det \mathbf{A}}\mathbf{A}. \tag{1.16}$$

We refer to the formula in (1.16) as the "lattice model". A similar analysis can be carried out for a three-dimensional unit cell, which will yield the following expression for the conductivity tensor,

$$\mathbf{K} = \frac{1}{4D_p R_c}\frac{(\mathrm{tr}\mathbf{A} - 2)^2}{\det \mathbf{A}}\mathbf{A}. \tag{1.17}$$

The formulae above apply when $\mathrm{tr}\mathbf{A}-2$ is non-negative. Otherwise the solution is $\mathbf{K} = \mathbf{0}$. The matrix $\mathbf{A}/\det\mathbf{A}$ can be understood as an approximate measure of how much a lattice cell of given perimeter (area) deviates from a square (cube) configuration, distributing proportionally less conductivity in directions where particle chains are more separated and more conductivity in directions where chains are tightly spaced.

Despite its inspiration from the lattice structure, there are several reasons to consider the applicability of the lattice model to more general particle networks. For one, the formula purports codirectionality of the fabric and conductivity, i.e. the deviators

of the two tensors are aligned, implying that the direction of anisotropy of one tensor gives the anisotropy direction of the other, which to a first approximation ought to match the behavior of general particle networks. Second, the results imply that conductivity should vanish when $\mathrm{tr}\mathbf{A} < 2$, which is sensible more generally (though not strictly) because particles in a percolating chain, as needed to conduct current across the sample, must have coordination number at least two. Above this threshold, conductivity increases with $\mathrm{tr}\mathbf{A}$ in line with one's basic intuition for more highly coordinated networks. In reality, islands of monomers, dimers, etc, enable the possibility of conductivity with an average coordination number less than two because conductivity will be nonzero with any percolating chain of particles. However, the geometric assumption underlying this lattice model prevents this from being taken into consideration. This shortcoming of the model is evident in our numerical results in figure 1-9 where nonzero conductivity was observed for a small range of coordination numbers below two.

We are aware of one other fabric-based analytical model for conductive particle networks, which was developed by Jagota and Hui [26]. In their work, a uniformity hypothesis is made with regard to the potential gradient, which results in a conductivity model that is fully linear in the fabric tensor,

$$\mathbf{K} = \frac{N_V D_p^2}{2R_c}\mathbf{A}. \tag{1.18}$$

The above, which can be proven to be an upper-bound on the real conductivity, is for a two-dimensional system and N_V is the particle number fraction (per area in 2D). In the isotropic case, (1.18) reduces to precisely the Hashin-Shtrikman upper bound one finds for the limit of thin, conductive bridges (of net resistance R_c) connecting the centers of contacting particles [22, 43]. The above model differs from ours most notably in that the conductivity is not thresholded by the coordination number, the formula depends explicitly on the particle area fraction as well fabric, and it does not depend on the fabric determinant.

1.5 Numerical Simulation

In order to perform numerical experiments and determine the generality of the lattice model, a large number of random particle networks (packings) must be created. There are a number of methods to do this already in the granular and particulate matter literature. See the references for a broad summary of the currently available granular packing algorithms[4]. Attractive suspensions have been modeled with the Diffusion-Limited Aggregation (DLA) model of Witten and Sander[27]. A common feature of many of the granular statics methods is that they solve force equilibrium equations for a system of particles This was not a feature that was required for this study, so these types of methods were not used, in the interest of saving computational time. Instead, we developed two methods for creating two-dimensional random contact networks of particles, and we tested our model against numerous packings generated by each method. Both methods allow us to influence the resulting anisotropic structure of the packings.

Algorithm 1: Our first packing algorithm was designed to create a dense random contact networks of particles. This is in contrast to a later algorithm, to be described below, which created packings that resulted in much lower-density packings. The dense packings were created by perturbing a 2D hexagonal close-packing of particles. This was achieved by placing points into a triangular lattice, adding random noise to the position of each point, and finally growing each particle as large as possible such that no particles overlapped. Anisotropy can be influenced by shearing the points with an affine transformation $\mathbf{x}' = \mathbf{F}\,\mathbf{x}$ before growing the radii. This process is described in pseudocode below (Algorithm 1.1). An example of the resulting packing overlaid by its analogous resistor network is shown in figure 1-5.

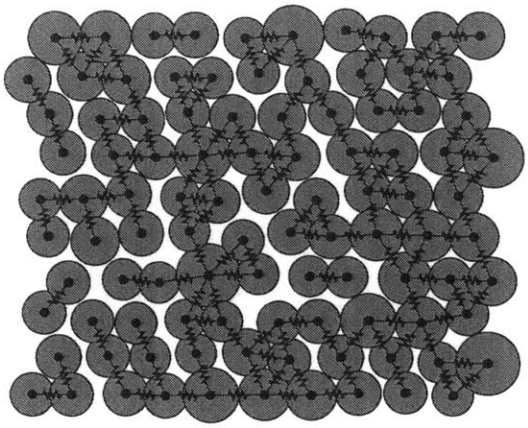


Figure 1-5: Example (using a small number of particles) of a dense particle packing resulting from Algorithm 1.1.

Algorithm 1.1

Seed $L \times L$ box with close-packed points

Perturb points with random noise

Move each point to new location $\mathbf{x}'$ by $\mathbf{x}' = \mathbf{F}\,\mathbf{x}$

while Not all radii frozen **do**

Find smallest distance that any particle can grow

Grow all particles by this amount

Freeze radii of particles that come into contact

end while

Algorithm 2: This procedure was motivated by a need to better understand the conductivity of carbon black suspensions in an insulating medium. The self-attraction carbon black particles leads to fractal particle networks that are electrically percolating at low volume fraction (below 1 vol%)[14].

To produce structures that more closely resemble carbon black suspensions, we developed our second packing algorithm, which is inspired by the "hit-and-stick" behavior of the carbon particles. In addition, the new algorithm is able to include the effects of particle Brownian motion but this is not essential to the algorithm.

First, clusters (single particles at this stage) are seeded randomly into a L^d box, where

d is the number of spatial dimensions. Next, a linear velocity field is imposed directly on each cluster's centroid according to

$$\mathbf{v} = -\mathbf{B}(\mathbf{x} - \mathbf{O}) + \mathbf{v_B} \tag{1.19}$$

where $\mathbf{O}$ is a point in the middle of the original box. This imposed velocity field serves to pull all of the clusters together. The extra term, $\mathbf{v_B}$, is a random velocity due to Brownian motion. The magnitude of this term can be tuned at runtime. The matrix $\mathbf{B}$ is a $d \times d$ matrix that allows us to impose an anisotropic velocity field. This allows us to influence (but not completely impose) the fabric tensor that results from this packing method. After the velocity field is imposed, the particle positions are updated by assuming a time step dt (computed at runtime). Then, the clusters are checked to determine whether any contacts have been made with other clusters. If so, the clusters are cohered into a single cluster for all future steps. This process of imposing velocity, updating positions, and handling contacts is repeated until only a single cluster remains. The process is outlined in pseudocode in Algorithm 1.2. An example of a packing resulting from this process is shown in figure 1-6 and a larger example is displayed in figure 1-8.

The box-counting fractal dimension [16] of the resulting packings was computed in order to determine if they resembled real-life packings found in experiments. The fractal dimension of packings produced by this method is approximately $d = 1.75$. This was compared against the particle network image in figure 1-1. This network has a fractal dimension of approximately $d = 1.7 \pm 0.1$. Uncertainty in the measurement is due to the image processing techniques used to identify particles. Based on these measurements, we are satisfied that this algorithm produces realistic packings, although more detailed correlation function measurements would be needed for a firmer conclusion.

Algorithm 1.2

Seed N clusters (particles) in L^d square
while $N_{Clusters} > 1$ **do**
  Move clusters according to $\mathbf{v} = -\mathbf{B}(\mathbf{x} - \mathbf{O}) + \mathbf{v_B}$
  Locate collisions between clusters
  Combine clusters in contact and recompute centroids
end while

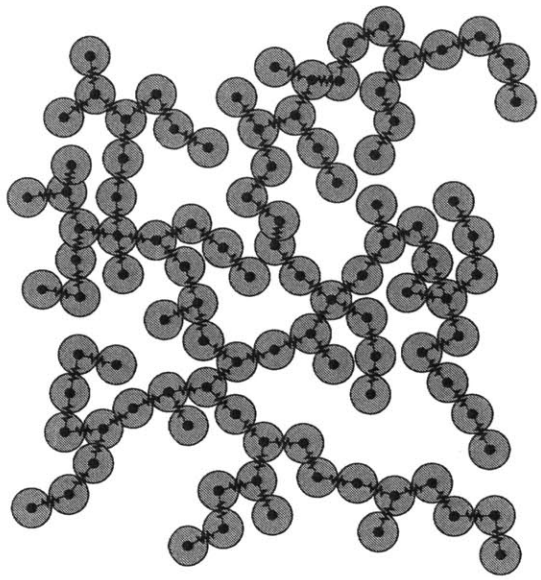


Figure 1-6: Example (using a small number of particles) of a packing resulting from Algorithm 1.2 using a small number of particles.

1.5.1 Aside: Packing dependence on diffusion

While using algorithm 1.2, we noticed that the strength of the diffusion term $\mathbf{v}_B$ had a large impact on the final volume fraction of the resulting packing. To explore this effect in more detail, we defined a non-negative dimensionless parameter F, jokingly called the "fluffy factor." F is defined as

$$F = \frac{D}{R^2|\mathbf{B}|} \tag{1.20}$$

where D is the standard diffusion coefficient for Brownian motion. By examining the definition of F, we see that $F = 0$ corresponds to pure advection by the sink defined by $\mathbf{B}$, and $F \to \infty$ corresponds to the case of pure diffusion. We then performed a series

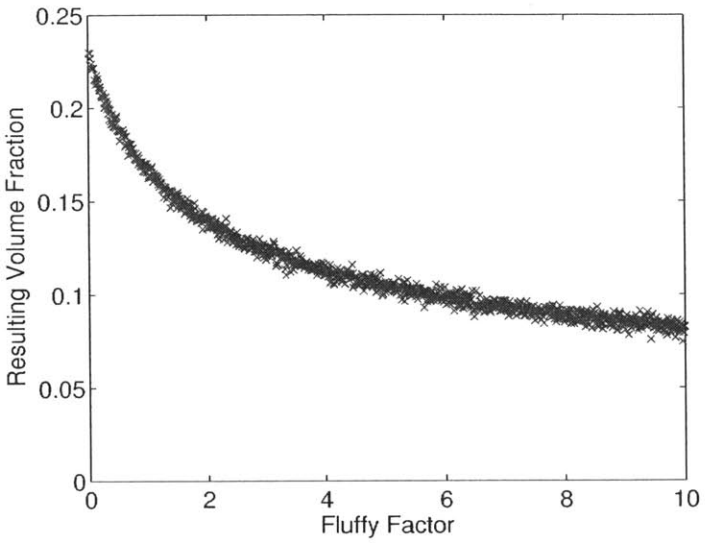


Figure 1-7: Packing volume fraction resulting from algorithm 1.2 as a function of F.

of 10000 three-dimensional simulations spanning $F \in [010]$. The resulting volume fractions were binned by F value and plotted in figure 1-7. The plot shows that the average volume fraction of a packing decreases with F. Additionally, the deviation from the mean behavior is extremely small, on the order of 0.5%, indicating that F is strongly predictive of the volume fraction of a packing.

The utility of this observation rests in the one-to-one correspondence of F and packing fraction. It provides a means of observing an existing packing and inferring the strength of diffusion relative to advection during the formation of the packing. This could, for example, provide some insight into the presence of flow during the formation of a packing of particles for which the diffusion constant is known. Similarly, it can provide an estimate of the diffusion coefficient if the flow conditions during packing formation are known. However it can be used, it is an interesting relationship that merits further study at a later time.

Applying boundary conditions: In order to apply the solution method described above to an arbitrary packing of particles, appropriate boundary conditions must be applied. In these simulations, a prescribed voltage was applied to particles all around the boundary. This process consists of two steps: first, the boundary must be identified,

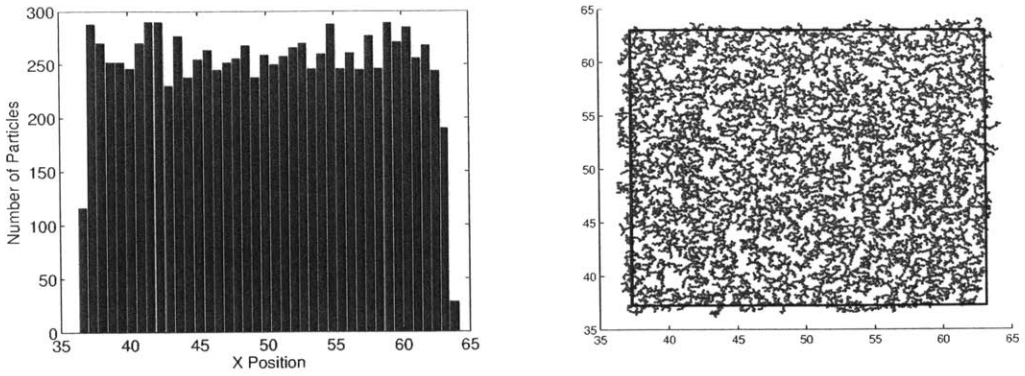


Figure 1-8: Example 10,000-particle packing (from Algorithm 2) with its associated x-position histogram and the boundary selected by the method.

and second, the linear system must be updated to reflect the known voltages.

For the first packing algorithm, identifying the boundary is a trivial process, since the particle locations are known *a priori*. For algorithm 1.2, however, the particle positions are not known. A boundary can be located visually quite easily at the end of the simulation process, but performing this step manually would be prohibitively slow. In order to expedite and automate the simulation process, the following method was devised to locate the boundary.

First, histograms of the particle x and y positions were separately created. To find the "left" and "right" boundaries, denoted x^- and x^+ respectively, the histogram of x positions was thresholded. The value x^- is defined as the smallest x value where the histogram reaches 50% of its maximum value. The value x^+ is defined as the largest x value that meets the same criterion. The top and bottom boundaries, y^+ and y^-, are found in the same manner using the histogram of particle y coordinates. The threshold value 50% was determined emperically to locate the same boundary that one would identify visually. An example packing and its associated x-position histogram is shown below in figure 1-8 to demonstrate the efficacy of the method. Once the location of the boundary has been identified, all particles whose centers fall less than one radius away from the lines are marked as being "boundary particles".

The expression in (1.6) can be computed easily from the solution of the particle net-

work, so by judiciously choosing $\langle \mathbf{E} \rangle$, the components of $\mathbf{K}_e$ can be extracted. In two dimensions, the effective conductivity tensor has three independent components, so three simulations are sufficient to extract all of the components. The K_{11} component can be extracted by setting $\langle \mathbf{E} \rangle = \mathbf{e}_x$. This corresponds to evaluating the integral for an applied boundary voltage of $\varphi = -x$. The remaining tensor components may be similarly extracted by applying specific potential fields at the boundary and evaluating the summation given in (1.6).

1.6 Tests

The previously described packing algorithms and solution procedures for the current/potential have been implemented in Matlab. Algorithm 1.1 was used to create 50,000 separate 400-particle packings. In all of these packings, the F_{11} and F_{22} components of the affine transformation $\mathbf{F}$ equalled 1.0. The F_{12} component that controlled the shearing of the packing ranged between 0 and 0.5 in increments of 0.01. Any particles that were sheared out of the original bounding rectangle were reflected to the other side of the box to return the packing to a rectangular geometry. We find packing fractions in the range $\phi \in [0.65, 0.80]$. Algorithm 1.2 was used to create 10,000 separate 5,000-particle packings. In the $\mathbf{B}$ matrix, the B_{11} component remained 1.0, and the B_{22} component was varied in $[1.0, 1.9]$ in increments of 0.1 to influence the level of anisotropy of the resulting packings. The approximate range of packing fractions we find is $\phi \in [0.45, 0.70]$. After applying the previously described procedure to each packing to obtain the effective conductivity tensor and average fabric tensor for each packing, the data were analyzed to determine how well the results agree with the model's predictions for the isotropic magnitude, the deviatoric magnitude, and the direction of conductivity. If the model is successful, then for any choice of $\mathbf{A}$, the model should match the ensemble average conductivity of all packings having that fabric $\mathbf{A}$. Below, for ease of demonstration, we bin the data based on scalar invariants of $\mathbf{A}$, and show either the ensemble average of the conductivity data at different

choices of those scalars, or simply show scatter-plot comparisons against the full set of tests when it is more illustrative to do so. These tests are described next, and thereafter we shall proceed to show how well the lattice model performs compared to the existing model, equation (1.18).

The isotropic behavior of the conductivity can be investigated by taking the trace of both sides of (1.16). The average coordination number is the most natural independent variable when examining the isotropic behavior, so in addition to taking the trace of both sides of (1.16), both sides were multiplied by $\det \mathbf{A}$ in order to make the right-hand side a single-valued function of $\mathrm{tr}\mathbf{A}$. This results in (1.21).

$$R_c \, \mathrm{tr}\mathbf{K} \, \det \mathbf{A} = (\mathrm{tr}\mathbf{A} - 2) \, \mathrm{tr}\mathbf{A} \tag{1.21}$$

The results of the simulations are scatter-plotted together with the analytical curve given by (1.21) in figure 1-9. It was found that the analytical solution is usually an upper bound on the measured conductivity. This can be explained by the fact that the analytical model was derived from an idealized system where the chains span a unit cell in a straight line. Since the total resistance of a chain is proportional to the number of contacts in the chain, it follows that the shortest chain between any two points is the lowest resistance path, and therefore most conductive. Since the model was derived from a straight-chain idealization, it implies an upper bound on the conductivity. This logic is less valid in low-coordinated systems, which have many disconnected groupings of one or two particles; low-coordinated systems rarely if ever occur from Algorithm 2 or in actual carbon black suspension networks. In this case, the trace of the system's fabric can be less than 2 but percolating chains may still exist to produce small but non-zero conductivity. This effect is evident in the figure in the data of Algorithm 1.

Next, we determine the extent the analytical lattice model predicts the anisotropy of the conductivity. To remove the influence of the isotropic behavior, we take the

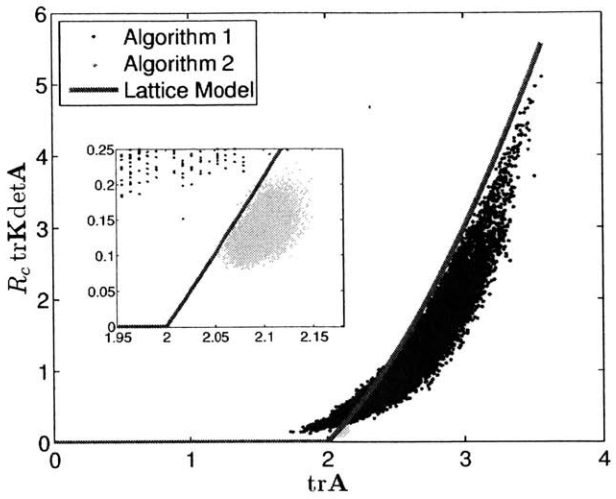


Figure 1-9: Predicted relationship between the (modified) trace of the conductivity and the fabric trace, compared to numerical results of 50,000 packings generated by Algorithm 1 and 10,000 generated under Algorithm 2. Inset is a zoom-in of the vicinity of $\mathrm{tr}\mathbf{A} = 2$.

deviator of both sides of (1.16). In this case, the most natural independent variable is the magnitude of the fabric deviator, so the resulting equation was manipulated to be a single-valued function of this quantity. After manipulation, (1.16) can be written as (1.22).

$$R_c\,\mathbf{K}_0 : \frac{\mathbf{A}_0}{|\mathbf{A}_0|}\left(\frac{\det\mathbf{A}}{\mathrm{tr}\mathbf{A}-2}\right) = |\mathbf{A}_0| \tag{1.22}$$

where a subscript 0 denotes the deviator of the tensor, and the term $\frac{\mathbf{A}_0}{|\mathbf{A}_0|}$ is commonly referred to as the direction or sign of the tensor $\mathbf{A}_0$. The left hand side was plotted against $|\mathbf{A}_0|$ by binning all the test data by $|\mathbf{A}_0|$ and ensemble averaging in each bin. It can be seen in figure 1-10 that, although there is a large amount of noise in the measurements, the model captures the average behavior very closely.

The final prediction that must be examined is the notion of codirectionality. The analytical model in (1.16) predicts that the fabric and conductivity tensors have the same eigenvectors. To examine this, the angle difference between the fabric and conductivity deviators was calculated, which is equivalent to the (signed) angle between the eigenvectors corresponding to the largest eigenvalues of the two tensors, denoted $\mathbf{e}_K$ and $\mathbf{e}_A$. The deviators were chosen because, in 2D, the eigenvector correspond-

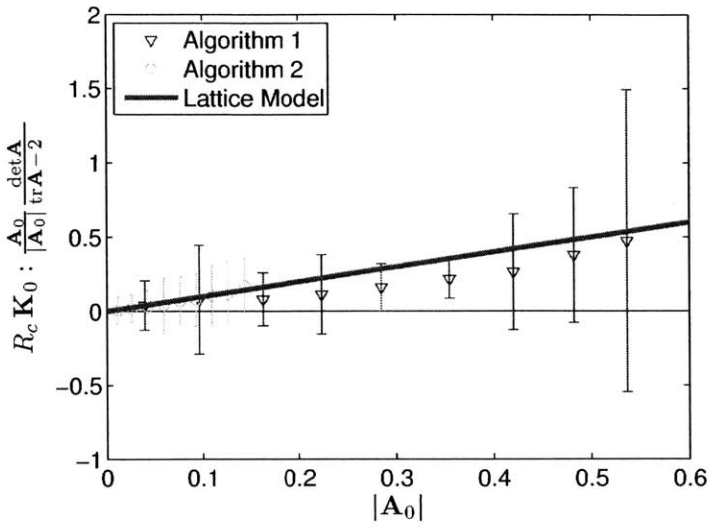


Figure 1-10: Predicted relationship between effective magnitude of the anisotropy of the conductivity and the invariants of the fabric. Error bars show $\pm$ one standard deviation.

ing to the positive eigenvalue can be unambiguously chosen. The probability density function of the angle difference as a function of $\Delta\theta$ is plotted in figure 1-11. It can be seen that this distribution is symmetrically centered around zero, indicating that the fabric and conductivity are strongly codirectional.

Finally, we also compared the lattice model, (1.16), to the existing model by Jagota & Hui[26] shown in (1.18). For the same 60,000 packings generated using both packing algorithms, we computed the relative error of the prediction of the trace and the determinant of the conductivity using each model and plotted the results in figures 1-12 and 1-13. In every case, we found that the new lattice model predictions were closer to the true values from the numerical experiments than the previous model by Jagota & Hui. On the other hand, the Jagota & Hui model maintains a strong upper bound on both invariants of the conductivity tensor, whereas the lattice model is not strictly an upper bound, as previously discussed. In addition, since the isotropic part of the Jagota & Hui model is identical to the Hashin-Shtrikman upper bound on conductivity, Figure 1-12 also indicates that the Lattice Model is falling within this bound. To be more precise, the lattice model prediction for the isotropic part of

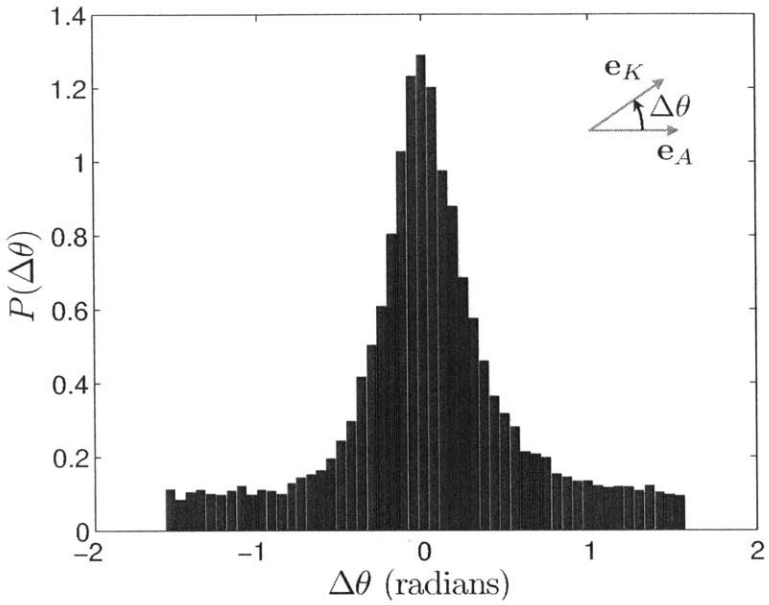


Figure 1-11: PDF of angle differences are distributed around zero, indicating codirectionality of the fabric and conductivity tensors, as predicted by the analytical model, i.e. (1.16).

conductivity is always less than the Hashin-Shtrikman upper bound computed for a given packing.

1.7 Discussion and Conclusions

In this paper we have derived and tested a new model relating the structure of a packing of particles to its tensorial electrical conductivity. The assumptions implicit in the model are that the suspending medium is a perfect insulator and that electrical resistance arises only at particle contacts. The structural measurement used was the fabric tensor, and the model arises from a straightforward analysis of a representative problem involving a lattice structure. The resulting model takes a nonlinear functional form, and was tested multiple ways against numerical simulations of many thousands of random particle packings. The agreement in its predictions of the various scalar properties and tensorial orientation is significant, especially in light of the simplistic nature of the fabric tensor being the sole independent variable for the

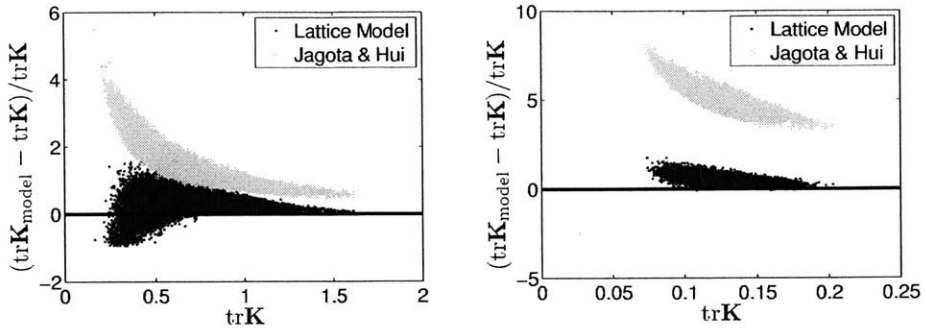


Figure 1-12: Plot of the relative error of the trace of conductivity. (Left) Relative error from packings created with Algorithm 1.1. (Right) Relative error from packings created with Algorithm 1.2.

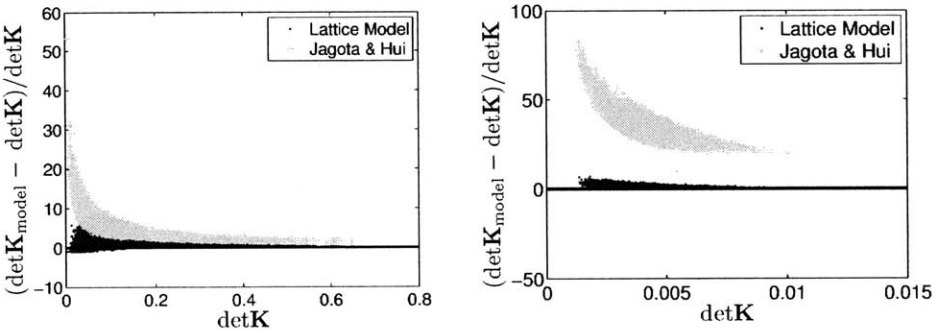


Figure 1-13: Plot of the relative error of the determinant of conductivity. (Left) Relative error from packings created with Algorithm 1.1. (Right) Relative error from packings created with Algorithm 1.2.

model. In our tests, the lattice model's accuracy was shown to be higher than an existing conductivity model, a model which requires more structural input data than the lattice model. While it is definitely possible to write a more accurate model by including dependences on more structural variables — some of our data spread is due to the finite nature of the datasets, but some is surely due to modeling error — the current simplicity of the lattice model is an advantage for its usage in engineering applications involving flowing suspension networks. Modeling frameworks for the evolution of anisotropy tensors in flowing media have been developed over the last decades[20, 17, 35]; keeping our model in terms of fabric, then, suggests a path to the simulation of simultaneous flow and current transfer fields in nontrivial systems by coupling a fabric evolution rule and a rheology with our conductivity model. Such a

capability would be key in the targeted application of modeling flow battery systems, which rely on a flowing conductive suspension that closely resembles the idealized system that we considered. A second direction for future work would be to apply the same idea of simultaneous flow and anisotropy modeling to other transport phenomena. For example, incompressible flow through a deforming granular media would have a similar mathematical formulation, albeit with reversed spatial assumptions since the impermeable grains play the role of the insulating medium here. The fabric tensor could be used as a surrogate to describe the structure of the pore space, which would relate to anisotropic permeability.

1.8 Acknowledgements

The authors acknowledge support from the Joint Center for Energy Storage Research (JCESR), an Energy Innovation Hub funded by the U.S. Department of Energy, Office of Science, Basic Energy Science (BES). The authors declare that there are no conflicts of interest.

Chapter 2

Suspension Microstructure Evolution

2.1 Introduction

Material anisotropy has been an active area of interest in many fields for decades. It plays a critical role in such fields as biomechanics [8, 13], plasticity [17], granular materials [3, 12, 31, 33, 35, 37, 41], liquid crystals [40], and more. Some materials, such as elastic composites, have fixed anisotropy that does not evolve over time. However, other materials may develop anisotropy due to deformation, as in the well-known kinematic hardening plasticity theory of Armstrong and Frederick [17]. Still others may develop anisotropy due to an externally-applied field, such as an electric field. This behavior is typical of liquid crystals [40].

Of particular interest in this study is the flow-induced anisotropy in colloidal suspensions. Suspensions of carbon black, an electrically-conductive form of carbon that has recently found application in a new class of batteries called "flow batteries" [14, 46]. The carbon black creates an electrically conductive network inside the flowing electrolytes of the battery, allowing for much higher reaction rates and overall system efficiency. However, it has been experimentally demonstrated that the networks in these carbon suspensions are highly sensitive to shearing [1, 2, 7, 38]. In

these studies, the conductivity of the carbon network drops precipitously with shear and recovers dynamically when brought to rest. This has serious implications for battery performance if the evolution of network structure and conductivity are not properly handled during design. Recent studies [39] on optimizing the efficiency of a flow battery have neglected the effect of a shear-induced drop in suspension conductivity. In addition to the drop in conductivity, it has been shown that suspensions become anisotropic during shearing flow, which can lead to anisotropic conductivity [25, 32, 45]. In this study, we will develop a frame-indifferent constitutive law for the evolution of a tensor-valued measure of network anisotropy and combine this model with previous conductivity modeling work to make quantitative predictions of conductivity evolution during flow.

2.1.1 Necessary Background

To describe the structure of the particle network in suspension, we use a tensor-valued measure called the "fabric tensor." It was originally devised to describe the contact network inside a dense granular material [31, 33, 37]. The fabric tensor is a second-order tensor that can be defined at the particle level with the relation

$$\mathbf{A}^P = \sum_{i=1}^{N_{contacts}} \mathbf{n}^{(i)} \otimes \mathbf{n}^{(i)} \tag{2.1}$$

where $\otimes$ denotes the dyadic product of contact unit normal vectors $\mathbf{n}_i$. This is illustrated in figure 2-1. Often, however, it is more illustrative to examine the average fabric of a group of particles rather than the particle-level information. The averaging can be done in

$$\mathbf{A} = \frac{1}{N_{particles}} \sum_{i=1}^{N_{particles}} \left(\mathbf{A}^P\right)_i \tag{2.2}$$

This definition of the fabric tensor yields a number of useful properties. First, the trace of $\mathbf{A}^P$ is equal to the coordination number of contacts on a particle. Consequently, tr$\mathbf{A}$ represents the average coordination number, usually denoted Z, of

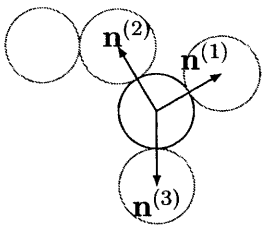


Figure 2-1: Schematic of particles in contact showing contact vectors $\mathbf{n}^{(i)}$.

a group of particles. Second, this definition of $\mathbf{A}$ results in a symmetric, positive semi-definite tensor. This means that the eigenvalues of $\mathbf{A}$ are non-negative, and the eigenvectors are orthogonal. This is appealing, because these properties are shared by the conductivity tensor $\mathbf{K}$.

In previous work [34], we modeled the conductivity tensor of a network as a function of the average fabric tensor $\mathbf{A}$. To do this, we assumed that a particle network could be represented by a regular lattice of particles with the same average fabric tensor. It is simple, then, to compute the effective conductivity and average fabric tensors of the regular lattice in terms of the lattice dimensions. We then inverted the fabric-lattice relationship to obtain the conductivity in terms of the fabric tensor directly. See Olsen and Kamrin 2015 [34] for a more detailed explanation. We found that

$$\mathbf{K} = k_1 \frac{(\mathrm{tr}\mathbf{A} - 2)^2}{\det \mathbf{A}} \mathbf{A}. \tag{2.3}$$

2.2 Evolution Law

Faced with the aforementioned experimental evidence that the contact network of a suspension changes due to flow, we set out to develop a continuum model that can accurately characterize the evolution of the network under arbitrary flow fields. Although the fundamental quantity that we model, the particle network, is composed of discrete units, we make a continuum approximation. In the continuum approximation, quantities at a point really represent averages of quantities, such as velocity or fabric, that are defined discretely at much smaller length scales. This is a valid

approximation since typical applications of these particle networks are several orders of magnitude larger than the constituents of the networks. As a concrete example, the particles of carbon black are $\sim 100nm$, while the features in a flow battery are $\sim 1mm$. Therefore, we can use the language of modern continuum mechanics to describe the evolution of the fabric tensor [19]. We define the velocity gradient $\mathbf{L}$, the stretching (strain-rate) tensor $\mathbf{D}$, and the spin tensor $\mathbf{W}$ below.

$$\mathbf{L} = \frac{\partial \mathbf{v}}{\partial \mathbf{x}} \tag{2.4}$$

$$\mathbf{D} = \text{sym}(\mathbf{L}) = \frac{1}{2}\left(\mathbf{L} + \mathbf{L}^T\right) \tag{2.5}$$

$$\mathbf{W} = \text{skw}(\mathbf{L}) = \frac{1}{2}\left(\mathbf{L} - \mathbf{L}^T\right) \tag{2.6}$$

We postulate a fabric evolution law of the form $\dot{\mathbf{A}} = \mathbf{f}(\mathbf{A}, \mathbf{L})$, where $\dot{\mathbf{A}}$ denotes the material time derivative of $\mathbf{A}$. In order for an evolution law such as this to be indifferent under a change in an observer's frame of reference, the evolution law must be of the form

$$\dot{\mathbf{A}} = \mathbf{WA} - \mathbf{AW} + \hat{\mathbf{f}}(\mathbf{A}, \mathbf{D}) \tag{2.7}$$

where where $\dot{\mathbf{A}}$ denotes the material time derivative of $\mathbf{A}$ and $\hat{\mathbf{f}}$ is an isotropic function of the fabric tensor and the stretching tensor [20]. Using the Caley-Hamilton theorem, Rivlin derived a representation theorem for 3×3 symmetric tensors as a function of two other 3×3 symmetric tensors [36]. Using this, we can write the evolution law as

$$\begin{aligned}\dot{\mathbf{A}} + \mathbf{AW} - \mathbf{WA} =& c_1\mathbf{1} + c_2\mathbf{A} + c_3\mathbf{D} + c_4\mathbf{A}^2 + c_5\mathbf{D}^2 + c_6(\mathbf{AD} + \mathbf{DA}) \\ &+ c_7(\mathbf{A}^2\mathbf{D} + \mathbf{DA}^2) + c_8(\mathbf{AD}^2 + \mathbf{D}^2\mathbf{A}) + c_9(\mathbf{A}^2\mathbf{D}^2 + \mathbf{D}^2\mathbf{A}^2)\end{aligned} \tag{2.8}$$

In this expression, the coefficients c_i are scalar functions of $\mathcal{I}_{\mathbf{A},\mathbf{D}}$, a full set of simul-

taneous invariants of $\mathbf{A}$ and $\mathbf{D}$.

$$\left.\begin{array}{cccc} \mathrm{tr}\mathbf{A}, & \mathrm{tr}\mathbf{A}^2, & \mathrm{tr}\mathbf{A}^3 & \\ \mathrm{tr}\mathbf{D}, & \mathrm{tr}\mathbf{D}^2, & \mathrm{tr}\mathbf{D}^3 & \\ \mathrm{tr}\mathbf{A}\mathbf{D}, & \mathrm{tr}\mathbf{A}^2\mathbf{D}, & \mathrm{tr}\mathbf{A}\mathbf{D}^2, & \mathrm{tr}\mathbf{A}^2\mathbf{D}^2 \end{array}\right\} \tag{2.9}$$

The left-hand side of (2.8) is known as the co-rotational time derivative of the tensor field $\mathbf{A}$. In the solid mechanics literature, it is frequently referred to as the Jaumann rate, and is given the symbol $\mathring{\mathbf{A}}$. In general, the left-hand side can be any objective time derivative of the tensor field, and it can be shown that they are all specializations of the Lie derivative [29]. We chose to use the co-rotational time derivative so that we, as modelers, retain full control over the evolution of $\mathbf{A}$ in its principal frame. Other objective rates, such as the contravariant or covariant time derivatives, contain terms that would contribute in this way [19].

2.2.1 Specialization of Constitutive Equation

The fully general form of the evolution law, shown in (2.8), has a large number of scalar functions that must be specified. In order to reduce the risk of overfitting our model, we chose to set $c_i \equiv 0$ for $i \geq 4$. This leaves us with all of the tensorially linear terms and their associated coefficients.

$$\mathring{\mathbf{A}} = c_1\mathbf{1} + c_2\mathbf{A} + c_3\mathbf{D} \tag{2.10}$$

The task of modeling, therefore, is reduced to choosing the coefficients c_1, c_2, and c_3 in a physically meaningful way.

Physical Intuition and Constraints

By examining the effect of each term on the evolution of the fabric, some intuition and observation can be applied to reduce the space of possible coefficients c_i. There

are some key observations that must be qualitatively matched before any more detailed modeling work can proceed. First, at steady state, the fabric will be positive, isotropic, and unchanging in the absence of flow. This requires that any anisotropy induced by flow relax away over time. Second, contacts are formed on the compressive axis of shearing flow and broken on the extension axis. This was experimentally observed by Hoekstra et al.[25]. Lastly, we observe that the electrical conductivity of a suspension decreases with increasing shear rate. However, the conductivity never reaches zero. Since we have derived a model, (2.3), for electrical conductivity in terms of the fabric tensor, we can use this to put further constraints on the evolution law.

We can translate these observations directly into constraints on the coefficients. The condition that the fabric relaxes to an isotropic steady state in the absence of flow implies the following two constraints.

$$c_1(\mathcal{I}_{\mathbf{A},\mathbf{D}}) > 0 \quad \forall\, \mathbf{A}, \mathbf{D} \tag{2.11}$$

$$c_2(\mathcal{I}_{\mathbf{A},\mathbf{D}}) < 0 \quad \forall\, \mathbf{A}, \mathbf{D} \tag{2.12}$$

If either of these constraints were violated, then the fabric would either decay away to a non-positive isotropic state or diverge. The precise nature of the decay or divergence depends on the functional forms of c_1 and c_2 and will be examined in greater detail in the following sections where these functions are specified.

The sign of c_3 can be identified from the second observation. By examining the pure shear evolution of $\mathbf{A}$ in the principal basis of $\mathbf{D}$, it becomes clear that in order for the observations in Hoekstra[25] and Morris [32] to be correctly represented, c_3 must obey

$$c_3(\mathcal{I}_{\mathbf{A},\mathbf{D}}) < 0 \quad \forall\, \mathbf{A}, \mathbf{D}. \tag{2.13}$$

A third constraint can be formed from the experimental observation that conductivity never entirely disappears, even at high shear rate[2]. Based on the conductivity model assumption in (2.3), this implies that $\mathrm{tr}\mathbf{A}$ remains above 2 at all times. To find the

conditions on the evolution law coefficients that must be true, we take the trace of (2.10) and solve for the steady-state trace of the fabric $\mathrm{tr}\mathbf{A}_{ss}$. The flow is assumed to be incompressible, so $\mathrm{tr}\mathbf{D} = 0$.

$$0 = 3\,c_1 + c_2\,\mathrm{tr}\mathbf{A}_{ss}$$
$$\mathrm{tr}\mathbf{A}_{ss} = -\frac{3\,c_1}{c_2} > 2$$

$$\Longrightarrow\ -\frac{c_1}{c_2} > \frac{2}{3} \quad \forall\ \mathbf{A}, \mathbf{D} \tag{2.14}$$

The final constraint is of mathematical origin. By closely examining the definition of the fabric tensor in (2.1), it quickly becomes evident that no diagonal component of $\mathbf{A}$ can have a negative value. A physical interpretation of this property of $\mathbf{A}$ is that it is not possible to have a negative number of contacts in any principal direction. Consequently, this condition also prevents any negative diagonal values of conductivity, and in so doing prevents negative dissipation, which would be a violation of the second law of thermodynamics. A constraint can be formed from this by examining the steady-state values of the diagonal components of $\mathbf{A}$ in an incompressible, spin-free flow (i.e. $\mathbf{W} = \mathbf{0}$).

$$\mathbf{0} = c_1\mathbf{1} + c_2\mathbf{A} + c_3\mathbf{D}$$
$$\mathbf{A} = -\frac{1}{c_2}(c_1\mathbf{1} + c_3\mathbf{D})$$

Without loss of generality, we can examine the component A_{11} and set it to be ≥ 0.

$$A_{11} = -\frac{1}{c_2}(c_1 + c_3 D_{11}) \geq 0 \tag{2.15}$$

Again, without loss of generality, we can take $\mathbf{D}$ to be

$$\mathbf{D} = \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & -(\lambda_1 + \lambda_2) \end{bmatrix} \tag{2.16}$$

Using the identity $\mathbf{N}_D = \mathbf{D}/|\mathbf{D}|$, where $\mathbf{N}_D$ is a unit tensor giving the direction of $\mathbf{D}$, we can rewrite (2.15) as

$$A_{11} = -\frac{1}{c_2}\left(c_1 + c_3|\mathbf{D}|\frac{\lambda_1}{|\mathbf{D}|}\right) \geq 0 \tag{2.17}$$

Using our knowledge that $-\frac{1}{c_2} > 0$, we can say that the argument inside the parentheses in (2.17) is ≥ 0. Next, we maximize the value of $\lambda_1/|\mathbf{D}|$ to constrain the value of c_3. We can compute $|\mathbf{D}|$ directly from (2.16).

$$|\mathbf{D}| = \sqrt{2}\sqrt{\lambda_1^2 + \lambda_2^2 + \lambda_1\lambda_2} \tag{2.18}$$

By factoring $|\lambda_1|$ out of (2.18), one can minimize the resulting expression in terms of λ_2/λ_1.

$$\min_{\lambda_2/\lambda_1} \sqrt{2}|\lambda_1|\sqrt{1 + \left(\frac{\lambda_2}{\lambda_1}\right)^2 + \left(\frac{\lambda_2}{\lambda_1}\right)} = \sqrt{\frac{3}{2}}\,|\lambda_1| \tag{2.19}$$

This, in turn, maximizes $\lambda_1/|\mathbf{D}|$, thus yielding a constraint on c_3. The resulting expression is

$$\frac{c_1}{c_3} \leq -\sqrt{\frac{2}{3}}\,|\mathbf{D}|\,\mathrm{sign}(\lambda_1) \tag{2.20}$$

Note that the left-hand side of (2.20) is negative. Thus, when $\lambda_1 < 0$—which is the condition for compressive flow in this direction—the condition is trivially satisfied. When $\lambda_1 > 0$, we obtain a constraint on the relative values of c_1 and c_3 in terms of $|\mathbf{D}|$. A simpler statement of this constraint is

$$\left|\frac{c_1}{c_3}\right| \geq \sqrt{\frac{2}{3}}\,|\mathbf{D}|. \tag{2.21}$$

The following sections will present two specializations of the scalar functions c_i that satisfy the model constraints.

Linear Model

The first specialization of the scalar functions c_i was constructed by finding the simplest set of functions that satisfy all constraints given by (2.11), (2.12), (2.13), (2.14), and (2.21). The set of functions chosen is

$$c_1 = (a_1 + a_2|\mathbf{D}|) \tag{2.22}$$

$$c_2 = -(b_1 + b_2|\mathbf{D}|) \tag{2.23}$$

$$c_3 = const < 0 \tag{2.24}$$

where a_1, a_2, b_1, b_2 are positive constants that must be fit to experimental data. The constraints can be applied to these functions in order to derive constraints on the new parameters. The resulting constraints, written in compact form, are

$$a_1 \geq \frac{2}{3} b_1 > 0 \tag{2.25}$$

$$a_2 \geq \frac{2}{3} b_2 > 0 \tag{2.26}$$

$$a_2 \geq -\sqrt{\frac{2}{3}}\, c_3 > 0 \tag{2.27}$$

With these functions for c_1, c_2, and c_3, we can write out the first fully-specified evolution law for the fabric tensor, termed the "Linear Model."

$$\mathring{\mathbf{A}} = (a_1 + a_2|\mathbf{D}|)\,\mathbf{1} - (b_1 + b_2|\mathbf{D}|)\,\mathbf{A} + c_3\mathbf{D} \tag{2.28}$$

The name "Linear Model" communicates the fact that c_1 and c_2 are linear functions of $|\mathbf{D}|$. Later model refinement leads to nonlinear functions, so this distinction is convenient.

By writing (2.28) in a different way, we can assign some physical significance to groups of the parameters. Consider the following grouping:

$$\mathring{\mathbf{A}} = (a_1\mathbf{1} - b_1\mathbf{A}) + (a_2\mathbf{1} - b_2\mathbf{A})\,|\mathbf{D}| + c_3\mathbf{D} \tag{2.29}$$

The first observation is that the first group on the right-hand side of (2.29) is the only part that does not depend on the flow. This term represents the influence of fabric rearrangement due to non-flow-related stimuli, such as thermal interactions. By noting the similarity of the evolution law in the absence of flow and the analogous scalar ordinary differential equation $\dot{A} = a - bA$, we can see that this model predicts an exponential decay of fabric to its steady-state value with a time scale set by $1/b_1$.

The second and third terms in (2.29) reflect the influence of flow on the fabric rearrangement. Similarly to how we examined the evolution behavior under thermal rearrangement, we can note that the model predicts an exponential decay to steady state, even in the presence of flow. However, when the flow terms are dominant, the time scale for this decay is $1/(b_2|\mathbf{D}|)$. The implication of this is that steady state is reached at a different rate when particle motion is driven by flow rather than by thermal motion alone. This is a key prediction of the model that will end up being verified by our experiments. Results demonstrating this phenomenon are shown in a later section.

Next, we take the trace of both sides of (2.29) to gain more insight into the significance of the model parameters. The result, after some rearrangement, is

$$\mathrm{tr}\dot{\mathbf{A}} = b_1\left(\frac{3a_1}{b_1} - \mathrm{tr}\mathbf{A}\right) + b_2|\mathbf{D}|\left(\frac{3a_2}{b_2} - \mathrm{tr}\mathbf{A}\right) \tag{2.30}$$

From this expression, we notice two very important quantities. First is the group $\frac{3a_1}{b_1}$. This is the steady-state value of the trace of $\mathbf{A}$ when there is no flow. We call this the static coordination number, Z_0. Physically, this is the average number of contacts per particle at steady state in the absence of shear. Similarly, we can see the group $\frac{3a_2}{b_2}$ in the second term. By taking the limit of (2.30) as $|\mathbf{D}| \to \infty$, we can see that

this is the value of the trace of **A** at infinite shear. We call this the infinite shear coordination number Z_∞. A quick re-examination of the model constraints (2.25) and (2.26) reveals that these quantities will never fall below 2 with a properly-chosen set of parameters.

Nonlinear Model

The second specialization of the scalar functions c_i was created after analyzing the results from the Linear Model. In the results shown later in figure 2-6, we noticed that the predicted decay back to static steady-state did not agree well with the experimentally observed decay. Motivated by this, we employed the discrete particle-aggregation code that we used in Olsen and Kamrin [34] to probe the nature of the rate of fabric evolution over time in the absence of flow.

In the code, 10,000 spherical particles were seeded into a three-dimensional box with periodic boundaries such that the volume fraction of particles was 1.5%, which was the same volume fraction as our physical experiments. Next, the particles were allowed to diffuse simultaneously. Particles and clusters were assigned velocities such that the distance that a cluster moved in a single time step was drawn from a Gaussian distribution with variance $D\Delta t$, where D is the diffusion coefficient, and Δt is the time step used in the simulation. Any clusters that came into contact were joined permanently, in order to mimic the "hit-and-stick" behavior that is typical in diffusion-limited aggregation [28]. As clusters grew, the diffusion coefficient of the cluster was adjusted according to $D = D_0/N$, where D_0 is the diffusion coefficient for a single particle, and N is the number of particles in a cluster. This scaling arose from the combination of the drag on a rod in Stokes flow [21],

$$F_D \approx \frac{8\pi\mu V a}{\ln(2a/b) + 0.5} \tag{2.31}$$

where a is the rod length and a/b is the rod aspect ratio, and the Stokes-Einstein

relation [15],

$$D = Mk_BT \tag{2.32}$$

where M is the mobility of a particle defined as $M = V/F_D$. Rearranging, we can say that

$$D \sim \frac{k_BT}{\mu a} \tag{2.33}$$

From (2.31), we can say that the mobility is approximately inversely proportional to the length of the rod, with a weak nonlinear correction due to the aspect ratio. Based on (2.32), the diffusion coefficient for a particle cluster is approximately inversely proportional to its length scale. The precise nature of the proportionality is geometry-dependent. We take the length scale associated with our clusters to be the total particle length of the cluster. That is to say, $a = N\,D_p$. Thus, when scaling diffusion coefficients in the code, it would be sufficient to scale by the inverse of the number of particles in a cluster to obtain a qualitatively correct relationship. The code already makes many simplifications from a real physical system, so we feel that a more precise scaling of the diffusion coefficient would not be informative. More detailed calculations would require a more realistic particle solver such as a Stokesian Dynamics code [9, 10].

The code allowed us to examine the evolution of the trace of the fabric tensor over time. This plot, shown below in figure 2-2 shows that the decay to steady state is linear when plotted in log-log space. This conclusively shows that the evolution of $\mathrm{tr}\mathbf{A}$ is not exponential. Rather, it follows a power law that can be fit by computing the slope of the linear portion of the plot. The power law that fits this data is

$$\log(\mathrm{tr}\mathbf{A}_{ss} - \mathrm{tr}\mathbf{A}) = -0.745\log(t) + 3.249 \tag{2.34}$$

Armed with the knowledge that the diffusion-driven part of the fabric evolution results in power-law, not exponential, decay, we set out to modify (2.28) to give the

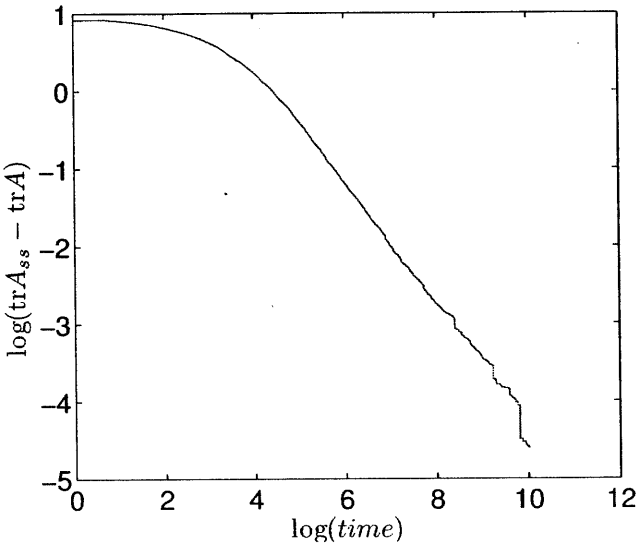


Figure 2-2: Log-log plot of $\mathrm{tr}\mathbf{A}_{ss} - \mathrm{tr}\mathbf{A}$ vs time shows power-law nature of $\mathrm{tr}\mathbf{A}$ decay to steady state.

desired response. Our pre-existing knowledge about the behavior of the Linear Model provided a strong jumping-off point for the new model. The key observation is the form of the non-flow-dependent terms in (2.30). Ultimately, the exponential decay is due to the fact that $\left(\frac{3a_1}{b_1} - \mathrm{tr}\mathbf{A}\right)$ is raised to exactly a power of 1 rather than some general power. To create the desired behavior in the $\mathrm{tr}\mathbf{A}$ evolution, we modified (2.28) to be the following

$$\mathring{\mathbf{A}} = \left(a_1 \left(Z_0 - \mathrm{tr}\mathbf{A}\right)^n + a_2|\mathbf{D}|\right)\mathbf{1} - \left(b_1 \left(Z_0 - \mathrm{tr}\mathbf{A}\right)^n + b_2|\mathbf{D}|\right)\mathbf{A} + c_3\mathbf{D} \tag{2.35}$$

where n is a new model parameter that controls the power-law nature of the $\mathrm{tr}\mathbf{A}$ decay. The parameters a_1, a_2, b_1, b_2, and c_3 all still obey the constraints listed in the Linear Model section in (2.25), (2.26), and (2.27). It can be recognized by comparing the form of (2.35) to the form of (2.10) that

$$c_1 = \left(a_1 \left(Z_0 - \mathrm{tr}\mathbf{A}\right)^n + a_2|\mathbf{D}|\right) \tag{2.36}$$

$$c_2 = -\left(b_1 \left(Z_0 - \mathrm{tr}\mathbf{A}\right)^n + b_2|\mathbf{D}|\right) \tag{2.37}$$

$$c_3 = const < 0 \tag{2.38}$$

In the same way that we call (2.28) the "Linear Model," we call (2.35) the "Nonlinear Model" due to the fact that the scalar functions c_1 and c_2 are now nonlinear functions of $\mathcal{I}_{\mathbf{A},\mathbf{D}}$. It should be noted that the Nonlinear Model reduces to the Linear Model in the case of $n = 0$.

We can now connect the power law observed in figure 2-2 to the new parameter n by solving for the evolution of tr$\mathbf{A}$ in the absence of flow. By taking the trace of (2.35) and rearranging terms, we obtain

$$\mathrm{tr}\dot{\mathbf{A}} = b_1 \left(Z_0 - \mathrm{tr}\mathbf{A}\right)^{n+1} + b_2 |\mathbf{D}| \left(Z_\infty - \mathrm{tr}\mathbf{A}\right) \tag{2.39}$$

where Z_0 and Z_∞ are defined the same as the groups in (2.30). Taking the case where $|\mathbf{D}| = 0$, we now have an ODE that yields a power-law decay to steady state for tr$\mathbf{A}$. Integrating this equation, we obtain a relation for the static recovery of tr$\mathbf{A}$.

$$\left(Z_0 - \mathrm{tr}\mathbf{A}\right) = \left(n\, b_1\, t\right)^{-1/n} \tag{2.40}$$

By plotting (2.40) in log-log space, we notice that the slope of the line would be $-1/n$. Comparing this to (2.34), we can see that $n = 1.34$. This value for n will be used in comparisons to experimental data in upcoming sections, and we show that it produces data fits that are far superior to those of the Linear Model.

Note that there were other ways to achieve the desired ODE for tr$\mathbf{A}$ in the absence of flow. One such way was to simply multiply the entire right-hand-side of (2.28) by $\left(Z_0 - \mathrm{tr}\mathbf{A}\right)^n$. However, this suffers from the unjustifiable weakness that if tr$\mathbf{A} = Z_0$, then nothing—not even flow—can ever change tr$\mathbf{A}$. This is not a feature that we wanted to build into the model, so the only way to keep a functional form similar to (2.28) and reproduce the power-law decay of tr$\mathbf{A}$ was to multiply $\left(Z_0 - \mathrm{tr}\mathbf{A}\right)^n$ into only the non-flow-dependent terms.

2.3 Methods

In this section, we describe the physical setup used to conduct conductivity experiments and the numerical procedures used for fitting the model.

2.3.1 Experimental Setup

The system studied is a carbon black suspension prepared in the absence of any dispersant by mixing carbon black particles (Cabot Vulcan XC72R of specific gravity 1.8) in a light mineral oil (Sigma-Aldrich, specific gravity 0.838, viscosity 20 mPa·s) as described in reference [18] at a weight concentration of 8%w/w. The suspension was sonicated for one hour and mixed vigorously prior to each test to minimize the effects of sedimentation.

Simultaneous rheo-electric measurements were performed using a custom setup on an ARG2 torsional stress-controlled rheometer with a parallel plate geometry. This custom setup uses liquid metal (EGaIn) to create a low-friction continuous electrical connection to the rotating shaft [23]. DC potentiostatic tests with $\phi = 100mV$ were performed using a Solartron SI1287 potentiostat. The plates ($d = 40mm$, $R_a = 0.10\mu m$), acting as a two-electrode system, are coated with gold to reduce contact resistance. All rheo-electric tests were performed at gap $h = 0.75mm$ and $T = 26 \pm 0.3°C$. A schematic of the device used to perform the rheo-electric measurements is shown in figure 2-3.

2.3.2 Experiments Performed

Two sets of experiments were performed using the setup descrbed in section 2.3.1. The first was a set of steady-state current measurements taken at nominal shear rates ranging from $\dot{\gamma}_0 \in [0, 300]\, s^{-1}$. At each shear rate, the current was allowed to equilibrate for ten minutes before the measurement was recorded to ensure that it had

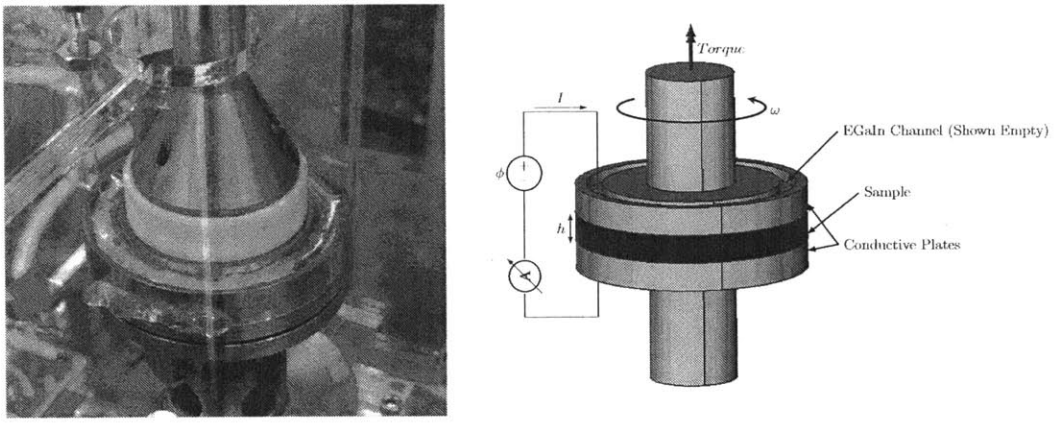


Figure 2-3: Schematic of device used to perform rheo-electric measurements.

relaxed to its steady-state value. The shear rates were swept in descending order to mitigate the effects of complications that arise at low shear rates, below $\dot{\gamma}_0 \sim 20s^{-1}$. A more detailed discussion of these phenomena may be found in Helal et al [23]. The purpose of this test was to provide a set of steady-state current data to which the evolution laws (2.28) and (2.35) could be fit.

The second dataset was a collection of transient ramp tests wherein current data was collected continuously for the duration of the test. Each ramp test is characterized by a starting nominal shear rate $\dot{\gamma}_1$, and ending nominal shear rate $\dot{\gamma}_2$, and a ramp time t_R. For each test, a shear rate of $\dot{\gamma}_1$ was applied for five minutes. Then, the shear rate was linearly ramped from $\dot{\gamma}_1$ to $\dot{\gamma}_2$ over a duration of t_R. Next, the shear rate was held at $\dot{\gamma}_2$ for five minutes. Finally, the plates were abruptly stopped, imposing $\dot{\gamma} = 0s^{-1}$. Current measurements were collected for an additional fifteen minutes. At the beginning of each test, the sample was pre-sheared at $\dot{\gamma} = 100s^{-1}$ for 5 minutes and left to equilibrate for 20 minutes prior to the start of data collection. This ensures that any "memory" of previous tests is removed and that all tests have consistent initial conditions. The purpose of these tests was to demonstrate that the evolution law correctly predicts not only the steady-state current values, but also the transient dynamics of transitions between flow rates.

2.3.3 Model Fitting

The Nonlinear Model has six parameters that govern the fabric evolution, plus one parameter for the conductivity model (2.3). As mentioned before, the Linear Model can be taken as a special case of the nonlinear model, with $n = 0$. These parameters must be chosen in a sensible way before the model may be used to quantitatively predict the evolution of fabric, and therefore conductivity, during flow. These parameters were chosen by `fminunc`, a Matlab nonlinear optimization routine, to minimize the difference between predicted and measured electrical currents. The predicted current is calculated by evaluating the integral

$$I(t) = 2\pi \int_0^R K_{zz}(\mathbf{A}(r,t)) \cdot \frac{\phi}{h} r \, dr \tag{2.41}$$

where K_{zz} is the component of conductivity perpendicular to the plate, ϕ is the applied potential difference across the plates, h is the plate separation, and R is the plate radius. The key part of this expression is that the fabric tensor is a function of radial position; each point along a radius is subjected to a different shear rate due to the applied torsional motion, and thus evolves differently.

The form of (2.41) can be modified slightly in order to solve for the current at steady state for a given nominal shear rate $\dot{\gamma}_0$.

$$I_{ss}(\dot{\gamma}_0) = 2\pi \int_0^R K_{zz}\left(\mathbf{A}_{ss}\left(\dot{\gamma}\right)\right) \cdot \frac{\phi}{h} r \, dr \tag{2.42}$$

In the above equation, $\mathbf{A}_{ss}$ is the steady-state fabric tensor, which is a function of the local shear rate $\dot{\gamma} = (r/R)\dot{\gamma}_0$. Approximating this integral with a Riemann sum requires the computation of $\mathbf{A}_{ss}$ at many values of r. $\mathbf{A}_{ss}$ can be computed using the procedure laid out in algorithm 2.1.

Algorithm 2.1 Computation of $\mathbf{A}_{ss}$ for given simple-shear strain-rate $\dot{\gamma}$

Input: Simple-shear strain-rate $\dot{\gamma}$.

Output: Steady-state fabric $\mathbf{A}_{ss}$

Construct $\mathbf{L} = \dot{\gamma}\, \hat{x} \otimes \hat{y}$.

Define $\mathbf{D} = \frac{1}{2}\left(\mathbf{L} + \mathbf{L}^T\right)$, $\mathbf{W} = \frac{1}{2}\left(\mathbf{L} - \mathbf{L}^T\right)$.

Solve nonlinear equation (2.39) for $\mathrm{tr}\mathbf{A}_{ss}$.

Compute c_1 and c_2 from $\mathrm{tr}\mathbf{A}_{ss}$ and $|\mathbf{D}|$ using (2.36), (2.37).

Use algebraic solution of (2.10) with $\dot{\mathbf{A}} = 0$ to find $\mathbf{A}_{ss}$.

To fit the model, we minimize the squared error between the model prediction and experimental values. The procedure for calculating this error is given in algorithm 2.2.

Algorithm 2.2 Objective function for evolution law optimization routine

Input: a_1, a_2, b_1, b_2, c_3, n, k_1

Output: Squared error = *Score*

Score $\leftarrow 0$

for all $\dot{\gamma}_{0,i} \in$ {Experimental shear rates} **do**

Compute $I_{ss}(\dot{\gamma}_{0,i})$ using (2.42) and algorithm 2.1.

Compute error between model and experiment: $err \leftarrow \left(I_{ss}(\dot{\gamma}_{0,i}) - I_{exp}(\dot{\gamma}_{0,i})\right)^2$.

Score $\leftarrow$ *Score* + *err*

end for

return *Score*

The following algorithm outlines the high-level procedure for fitting the evolution laws (2.28) and (2.35) to a set of experimental data. It relies heavily on the preceeding procedures laid out in algorithm 2.1 and algorithm 2.2.

Algorithm 2.3 Procedure for fitting (2.28) and (2.35) to experimental data

Input: List of $\{\dot{\gamma}_0, I_{exp}\}$ pairs from steady-state experiments

Output: Evolution law coefficients $\{a_1, a_2, b_1, b_2, c_3, k_1\}$

Estimate b_1 based on relaxation time of transient ramp tests.

Choose n to match discrete particle simulations (see section 2.2.1).

Call `fminunc` with an objective function given by algorithm 2.2.

$\{a_1, a_2, b_2, c_3, k_1\} \leftarrow$ Output from `fminunc`.

Verify that constraints (2.25), (2.26), (2.27) are satisfied.

Adjust b_1 estimate and repeat until $\dot{\gamma}_0 = 0$ transient is well-matched.

To simulate the electrical current for a transient test, the evolution law (2.35) must be directly integrated at each point on the radius where the conductivity will be evaluated. To do this, one must first construct a function for the nominal shear rate as a function of time, $\dot{\gamma}_0(t)$. This can be constructed directly from the experimental protocol for the test that you wish to replicate. The evolution law was numerically integrated using the 4th-order Runge-Kutta method implemented in the Matlab function `ode45`. After the fabric is known for all time at each point along the radius, (2.41) can be approximated directly using a Riemann sum to give the predicted current as a function of time. This process was used in the last step of algorithm 2.3 as a way to test the quality of the b_1 estimate.

2.4 Results

2.4.1 Linear Model

Model Parameters

The model-fitting procedure described in section 2.3.3 was carried out using the Linear Model. The resulting parameters are shown below in table 2.1. The parameters given

satisfy the physical constraints of the model given by (2.25), (2.26), and (2.27), so they are considered to be a valid set of parameters.

Table 2.1: Parameters used to fit the Linear Model to the experimental data

a_1	a_2	b_1	b_2	c_3	n	k_1
0.0393	0.0014	0.0133	0.0013	-0.0011	0	0.0241

Steady-state Experiments

The following plots show the results of the linear model fit to the steady-state experiments described in section 2.3.2. The results show that the linear does a good job predicting the current measurements in the experiment. However, the model does undershoot some of the current measurements around the $\dot{\gamma}_0 \in (50, 200)s^{-1}$ range and overshoot beyond that point. Figure 2-4 shows the fit of the Linear Model to experimental data. Figure 2-5 shows the predicted transverse conductivity in a hypothetical simple shear experiment.

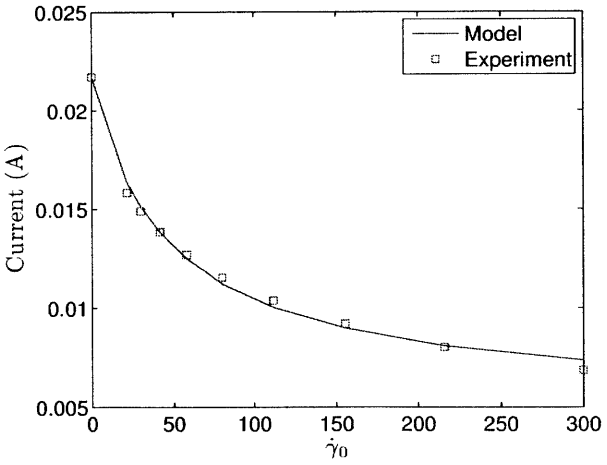


Figure 2-4: Linear Model fit of steady-state current measurements at different nominal shear rates.

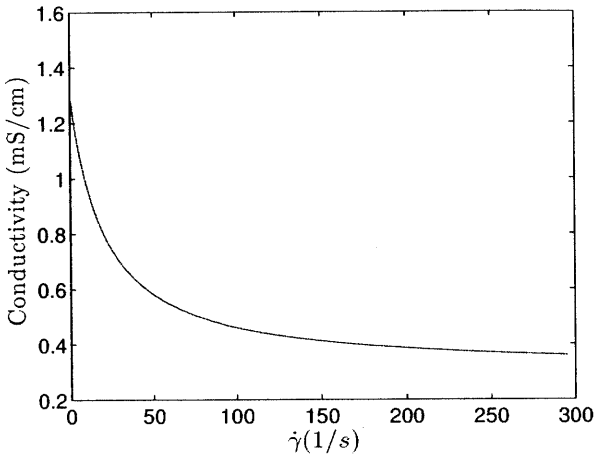


Figure 2-5: Linear Model prediction of steady-state transverse conductivity as a function of shear rate in simple shear.

Transient Ramp Experiments

The Linear Model was tested to see how well it predicted the transient ramp experiments described in section 2.3.2. In these experiments, samples were subjected to the previously-described ramp protocol with the sets of parameters shown in table 2.2. Each experiment was simulated using the Linear Model, and the results are shown in figures 2-6, 2-7, 2-8, 2-9, 2-10, and 2-11. The figures show the current, normalized by the static current measurement, plotted as a function of time. The transitions between different shear rates are quite noticeable from the shape of the curve. The initial plateau is the normalized steady-state current at $\dot{\gamma}_1$ ($\dot{\gamma}_1 = 50s^{-1}$ in all cases). The second plateau is the steady value at $\dot{\gamma}_2$. The transition between the first and second plateaus is the ramp period, and spans a duration of 30, 60, or 300 seconds depending on the test. The region after the second plateau is the abrupt stop to zero shear. It is from this region that the time scale for the relaxation, given by b_1, is determined. See section 2.3.3 for details on this procedure.

The figures show that the Linear model predicts the experiment quite well for most of the curve. The most noticeable breakdown in the quality of the fit is in the last

Table 2.2: Parameter sets for the transient ramp experiments.

$\dot{\gamma}_1\ [s^{-1}]$	$\dot{\gamma}_2\ [s^{-1}]$	$t_R\ [s]$
50	100	30
50	100	60
50	100	300
50	200	30
50	200	60
50	200	300

segment of the experiment when the conductivity is relaxing back to its static value. Here, the shape of the relaxation is not correct. As mentioned in section 2.2.1, the fabric relaxation in the Linear Model is exponential. This is inconsistent with the experiments, and is most noticeable here. The mismatch in these results was the factor that prompted the development of the Nonlinear Model.

The linear model does, however, verify a key prediction from section 2.2.1. Namely, the difference of equilibration times during flow and under static conditions. Recall that the model predicts that the time scale for fabric relaxation under flow-dominated conditions is $1/(b_2|\mathbf{D}|)$, and the time scale for diffusion-driven relaxation is $1/b_1$. The impact of this is most easily seen when $\dot{\gamma}_0$ changes from $50s^{-1}$ to $100s^{-1}$. The conductivity takes much less time to equilibrate in this transition than for the transition to $\dot{\gamma}_0 = 0$.

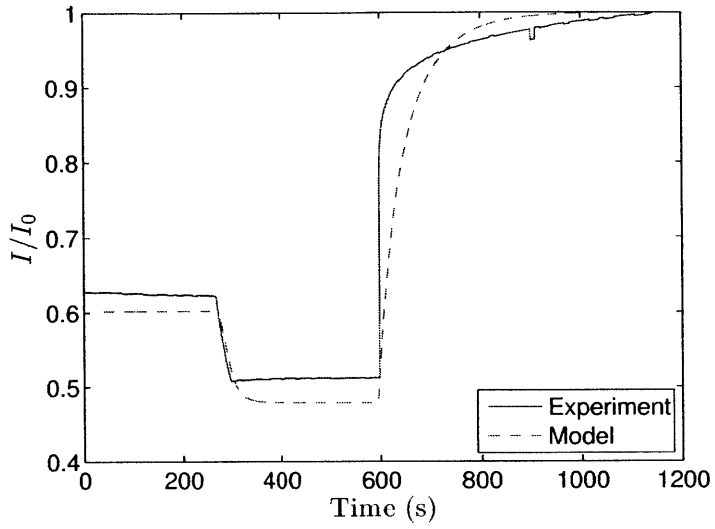


Figure 2-6: Linear Model prediction of normalized current as a function of time during 30 second ramp experiment with $\dot{\gamma}_1 = 50s^{-1}$ and $\dot{\gamma}_2 = 100s^{-1}$.

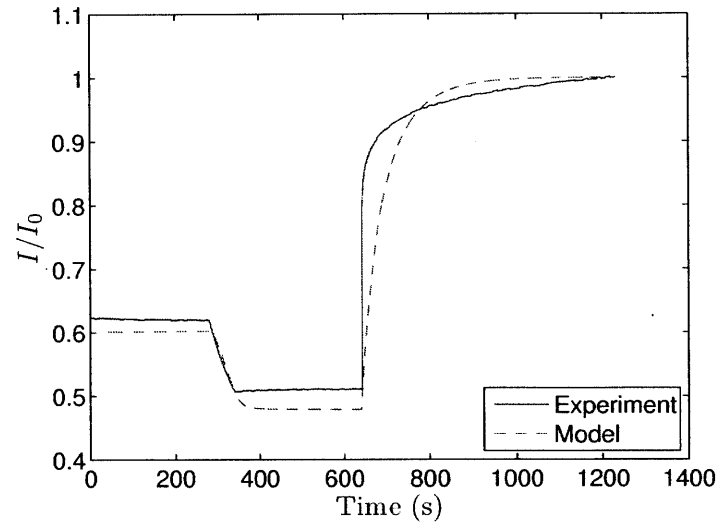


Figure 2-7: Linear Model prediction of normalized current as a function of time during 60 second ramp experiment with $\dot{\gamma}_1 = 50s^{-1}$ and $\dot{\gamma}_2 = 100s^{-1}$.

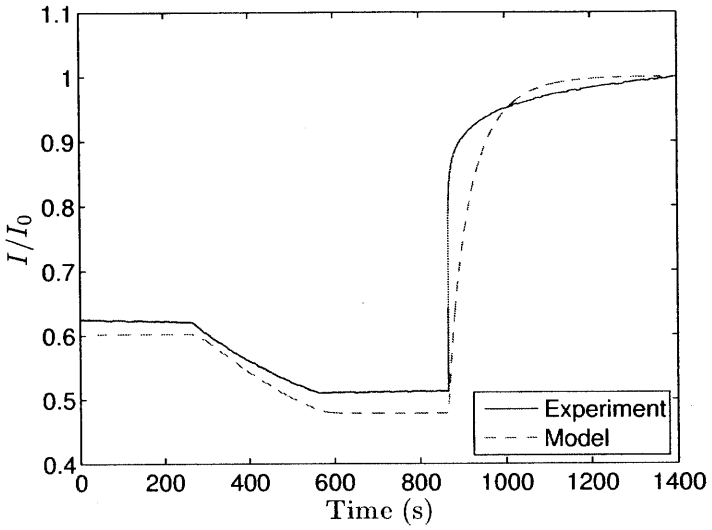


Figure 2-8: Linear Model prediction of normalized current as a function of time during 300 second ramp experiment with $\dot{\gamma}_1 = 50s^{-1}$ and $\dot{\gamma}_2 = 100s^{-1}$.

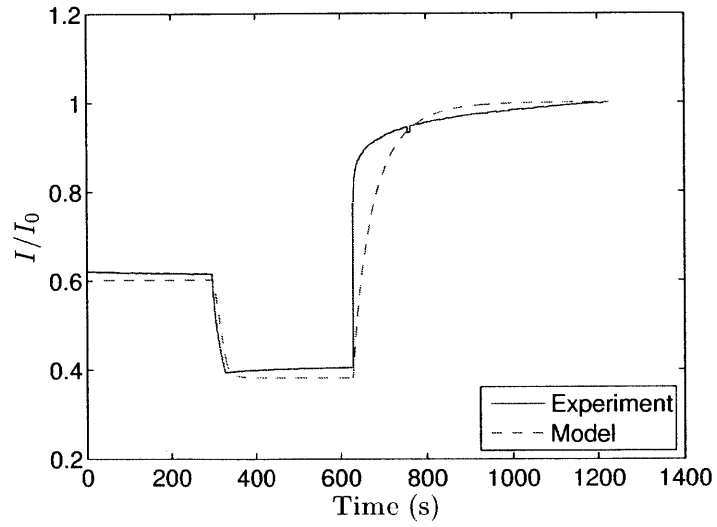


Figure 2-9: Linear Model prediction of normalized current as a function of time during 30 second ramp experiment with $\dot{\gamma}_1 = 50s^{-1}$ and $\dot{\gamma}_2 = 200s^{-1}$.

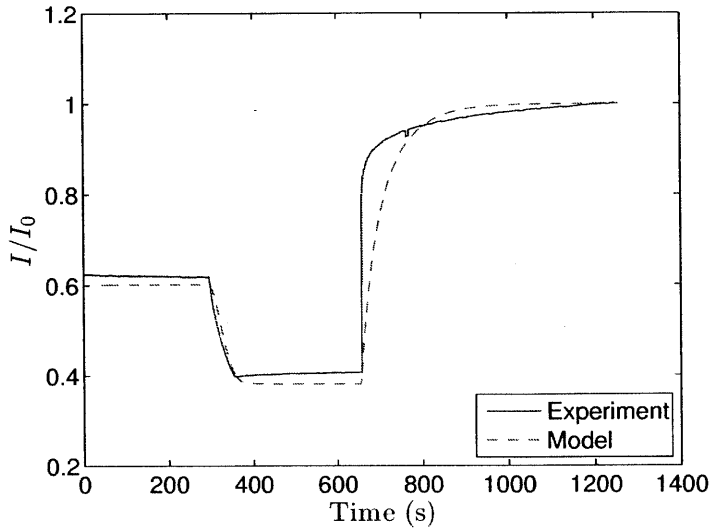


Figure 2-10: Linear Model prediction of normalized current as a function of time during 60 second ramp experiment with $\dot{\gamma}_1 = 50s^{-1}$ and $\dot{\gamma}_2 = 200s^{-1}$.

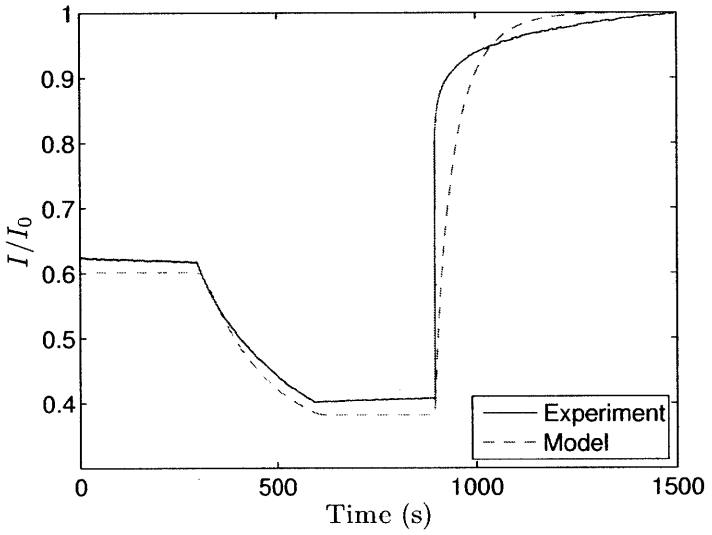


Figure 2-11: Linear Model prediction of normalized current as a function of time during 300 second ramp experiment with $\dot{\gamma}_1 = 50s^{-1}$ and $\dot{\gamma}_2 = 200s^{-1}$.

2.4.2 Nonlinear Model

We also fit the same experimental data with the Nonlinear Model to see the difference in performance between the two models. Parameters found to fit the data are shown

below in table 2.3. Recall that the n parameter was determined *a priori* using out discrete model, so the number of parameters we must determine through experimental fitting is the same as for the Linear Model. As with the Linear Model, the constants satisfy the constraints (2.25), (2.26), and (2.27), so this is a valid set of parameters. These parameters were used to generate all of the following plots for the Nonlinear Model.

Table 2.3: Parameters used to fit the Nonlinear Model to the experimental data

a_1	a_2	b_1	b_2	c_3	n	k_1
0.0137	0.0025	0.0040	0.0028	-0.0020	1.34	0.022

Steady-state Experiments

The following plots show the results of the Nonlinear Model fit to the steady-state experiments described previously in section 2.3.2. Figure 2-12 shows the fit of the Nonlinear Model predicted current to experimental data. Figure 2-13 shows the transverse conductivity predicted in a hypothetical simple shear experiment. The results indicate the Nonlinear Model does an excellent job predicting the steady-state current measurements given a suitable set of parameters. This model succeeds in capturing the steady-state behavior over the entire tested range of $\dot{\gamma}_0$, whereas the Linear Model gets close but is not able to quite capture the right behavior.

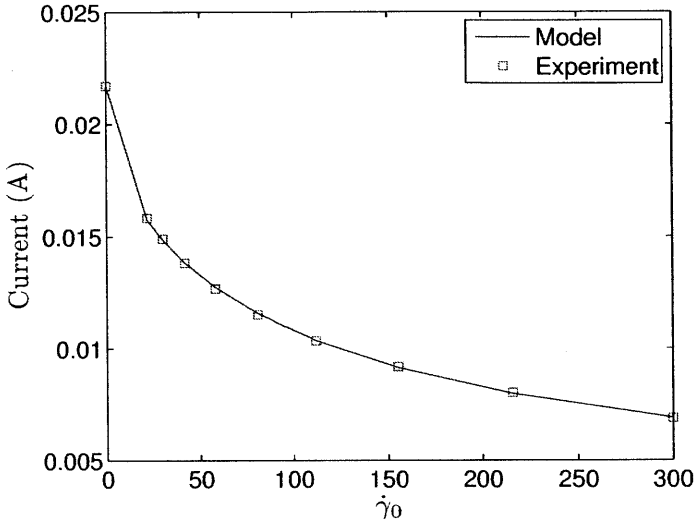


Figure 2-12: Nonlinear Model fit of steady-state current measurements at different nominal shear rates.

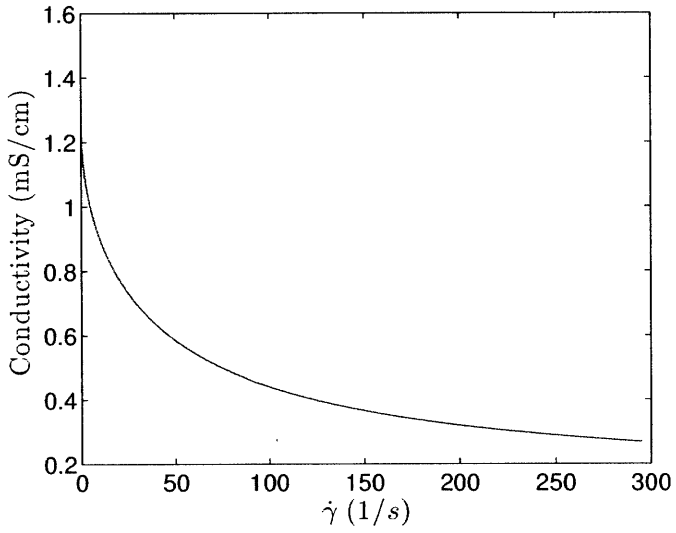


Figure 2-13: Nonlinear Model prediction of steady-state transverse conductivity as a function of shear rate in simple shear.

Transient Ramp Experiments

The Nonlinear Model was tested to determine how well it predicted the transient experiments outlined in section 2.3.2. The datasets used here were the same as those

compared against the Linear Model in section 2.4.1, with the ramp parameters given in table 2.2. Like before, the results shown in figures 2-14, 2-15, 2-16, 2-17, 2-18, and 2-19. show the current normalized by the static current measurement plotted as a function of time.

The figures show that the Nonlinear Model is capable of making accurate, quantitative predictions for the transient behavior of the normalized current. The model and the data are virtually indistinguishable over the majority of the experiment, and the model even reproduces the relaxation behavior correctly. This is a marked improvement over the Linear Model, which proved unable to reproduce this key feature.

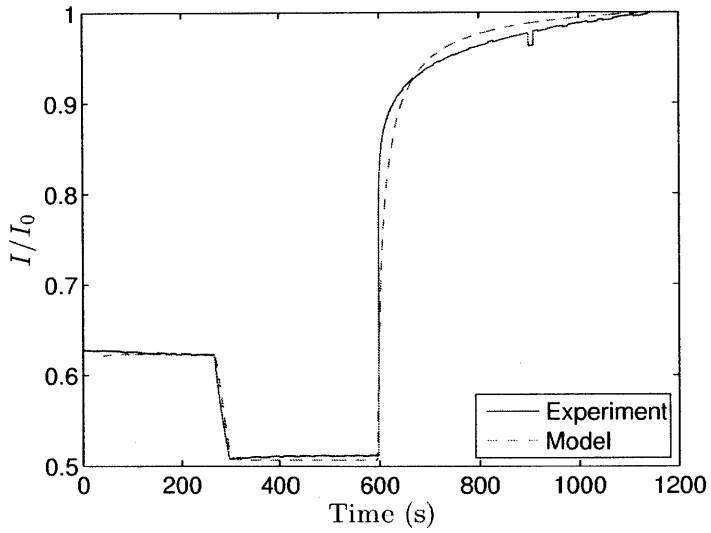


Figure 2-14: Nonlinear Model prediction of normalized current as a function of time during 30 second ramp experiment with $\dot{\gamma}_1 = 50s^{-1}$ and $\dot{\gamma}_2 = 100s^{-1}$.

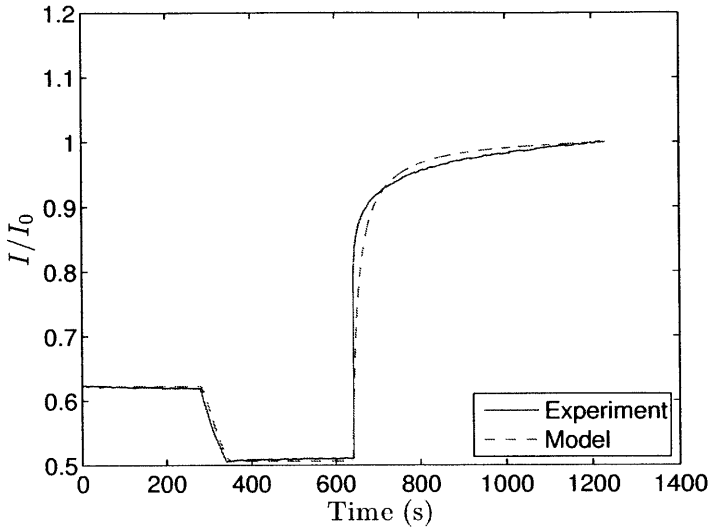


Figure 2-15: Nonlinear Model prediction of normalized current as a function of time during 60 second ramp experiment with $\dot{\gamma}_1 = 50s^{-1}$ and $\dot{\gamma}_2 = 100s^{-1}$.

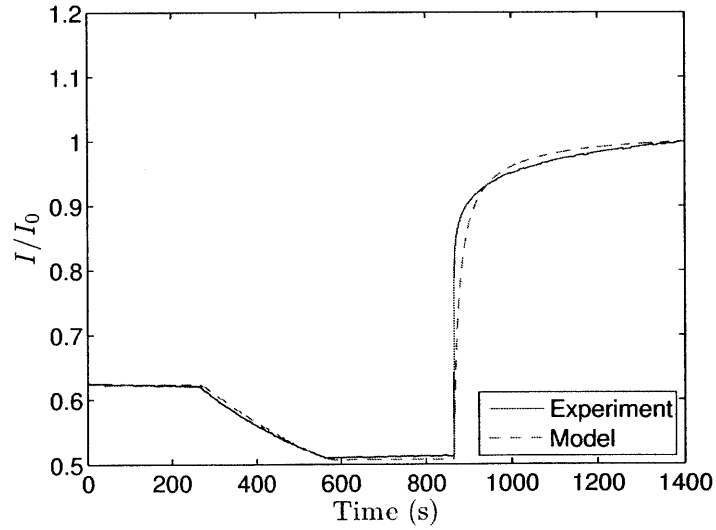


Figure 2-16: Nonlinear Model prediction of normalized current as a function of time during 300 second ramp experiment with $\dot{\gamma}_1 = 50s^{-1}$ and $\dot{\gamma}_2 = 100s^{-1}$.

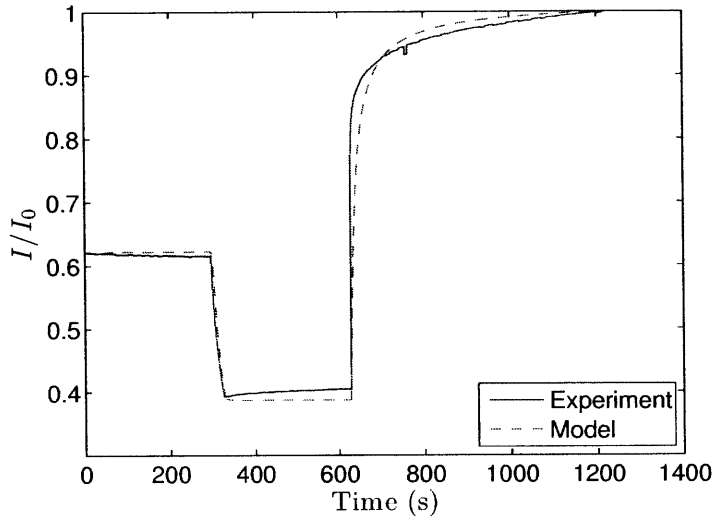


Figure 2-17: Nonlinear Model prediction of normalized current as a function of time during 30 second ramp experiment with $\dot{\gamma}_1 = 50s^{-1}$ and $\dot{\gamma}_2 = 200s^{-1}$.

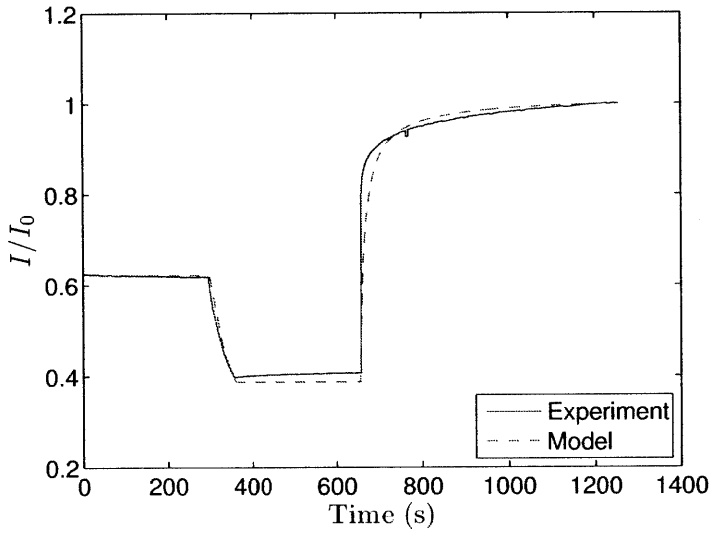


Figure 2-18: Nonlinear Model prediction of normalized current as a function of time during 60 second ramp experiment with $\dot{\gamma}_1 = 50s^{-1}$ and $\dot{\gamma}_2 = 200s^{-1}$.

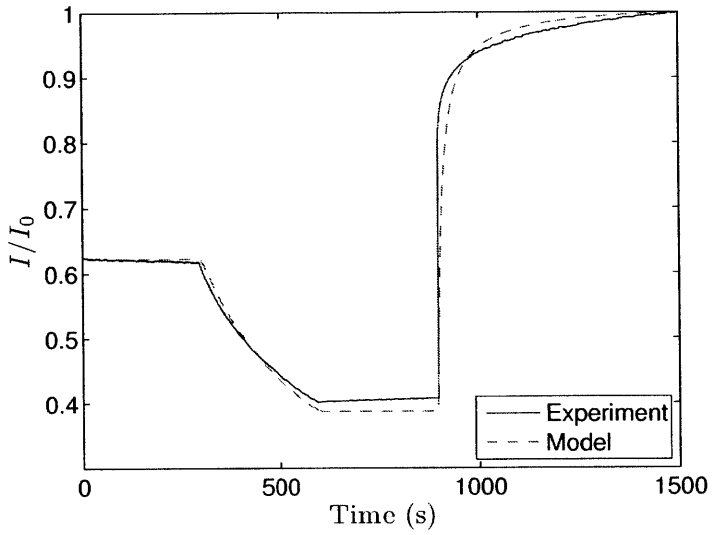


Figure 2-19: Nonlinear Model prediction of normalized current as a function of time during 300 second ramp experiment with $\dot{\gamma}_1 = 50s^{-1}$ and $\dot{\gamma}_2 = 200s^{-1}$.

2.5 Discussion and Conclusions

We have continued our previous work of modeling the electrical conductivity of suspensions. In the previous study, we derived an elegantly simple relationship between the conductivity tensor $\mathbf{K}$ and the fabric tensor $\mathbf{A}$ given, in three dimensions, by

$$\mathbf{K} = k_1 \frac{(\mathrm{tr}\mathbf{A} - 2)^2}{\det \mathbf{A}} \mathbf{A}. \tag{2.43}$$

In this work we have proposed a frame-indifferent constitutive equation for the evolution of the fabric tensor under flow. We proposed that the evolution of fabric is a function of the current fabric state and the local velocity gradient of the flow.

$$\dot{\mathbf{A}} = \mathbf{f}(\mathbf{A}, \mathbf{L}) \tag{2.44}$$

Using standard arguments from continuum mechanics and by making some simplifying assumptions, we arrived at the following general form for the evolution law

$$\mathring{\mathbf{A}} = c_1\mathbf{1} + c_2\mathbf{A} + c_3\mathbf{D} \tag{2.45}$$

where the coefficients c_1, c_2, and c_3 are scalar functions of the joint invariants of $\mathbf{A}$ and $\mathbf{D}$, and $\mathring{\mathbf{A}}$ denotes the co-rotational time derivative of $\mathbf{A}$.

From here, we proposed two evolution laws that are similar in character but yield important differences when simulating experiments. First, we proposed the Linear Model, so named due to the function coefficients' linear dependence on the invariants of $\mathbf{A}$ and $\mathbf{D}$.

$$\mathring{\mathbf{A}} = (a_1 + a_2|\mathbf{D}|)\ \mathbf{1} - (b_1 + b_2|\mathbf{D}|)\ \mathbf{A} + c_3\mathbf{D} \tag{2.46}$$

This model did a good job predicting the steady-state behavior of the suspension conductivity, but its predicted transients did not match experiments very well. The second proposed model is the Nonlinear Model, predictably named due to its function coefficients' nonlinear dependence on the invariants of $\mathbf{A}$ and $\mathbf{D}$.

$$\mathring{\mathbf{A}} = (a_1\,(Z_0 - \mathrm{tr}\mathbf{A})^n + a_2|\mathbf{D}|)\ \mathbf{1} - (b_1\,(Z_0 - \mathrm{tr}\mathbf{A})^n + b_2|\mathbf{D}|)\ \mathbf{A} + c_3\mathbf{D} \tag{2.47}$$

In this model, a single new parameter, n, was introduced with the intent of modifying the transient behavior of (2.46). It was selected to match computer simulations, and it was hoped that this agreement would lead to better prediction of the experiments.

Both models were fit to experimental data using the procedure outlined above in section 2.3.3. While both models do a good job predicting the measured current in the experiments described in section 2.3.2, it is the Nonlinear Model that clearly is better able to capture the dynamics of the system.

2.5.1 Future Work

With the fabric-conductivity and fabric-evolution models in hand, we now have a powerful tool for designing systems which rely on flowing, electrically-active suspensions. The primary use for the models is to design a semi-solid flow battery [14, 24, 39], which depends heavily on such a suspension to transport electrical charge within the flowing electrodes. Given that the shear-induced conductivity loss is quite strong at even moderate shear rates (eg, near 66% drop in conductivity from its static value at $\dot{\gamma} = 150s^{-1}$ in the system studied, as shown in figure 2-13), the particle network evolution can have a severe impact on the performance of the battery during flow.

2.6 Acknowledgements

The authors acknowledge support from the Joint Center for Energy Storage Research (JCESR), an Energy Innovation Hub funded by the U.S. Department of Energy, Office of Science, Basic Energy Science (BES). The authors declare that there are no conflicts of interest.

Appendix A

Selected Source Code: nparticle.cpp

The following is a listing of the main function in nparticle.cpp, the discrete particle aggregation code used throughout this work. Supporting data structures are not included since the main logic of the program can be grasped without the details of the implementation underlying data structures such as the Cluster object or utility routines such as `write_data_file`. Note, this code will not compile without these supporting functions.

```
#include <stdio.h>
#include <stdlib.h>
#include <sys/time.h>
#include <iostream>
#include <vector>
#include <cmath>
#include <signal.h>
#include <string.h>
#include <random>
#include <string>
#include <limits>
```

```
#include "cluster.hpp"
#include "particle_utils.hpp"
#include "PackingParameters.hpp"

using namespace std;

//global stuff
particle_t *P;
PackingParameters params;

//=====================================
void signal_handler(int signum) {
  char buf[128];
  memset(buf, '\0', 128);
  sprintf(buf, "%s_interrupt", params.getPackingFname().c_str
     ());
  vector<contact_t> contacts;
  calculate_contacts(P, params.getNParticles(), contacts);
  write_data_file(P, params.getNParticles(), contacts, buf);
  exit(signum);
}

int main(int argc, char **argv) {


  if(argc >= 2) {
    params.parseConfigFile(argv[1]);
  }
  params.print(cout);
```

```
char imgnamebuf[128];
memset(imgnamebuf, '\0', 128);
char datanamebuf[128];
memset(datanamebuf, '\0', 128);

//set up signal handlers
struct sigaction sa;
sa.sa_handler = &signal_handler;
sa.sa_flags = SA_RESTART;
sigfillset(&sa.sa_mask);
if(sigaction(SIGTERM, &sa, NULL) == -1) {
  perror("error: could not connect SIGTERM");
}
if(sigaction(SIGINT, &sa, NULL) == -1) {
  perror("error: could not connect SIGINT");
}



//
  ==========================================================


P = new particle_t[params.getNParticles()];
vector<Cluster*> clusters;
double L = params.getBoxRatio()*pow(params.getNParticles(),
    1.0/params.getDimensions())*params.getParticleRadius();

//seed rand w/ usec of current time
struct timeval tv;
gettimeofday(&tv, NULL);
default_random_engine generator(tv.tv_usec);
```

```
uniform_real_distribution<double> uniformRand(0,1.0);

//need global bounding box for image writing
double globalxmin = L;
double globalxmax = -1;
double globalymin = L;
double globalymax = -1;
for(int i=0; i<params.getNParticles(); ++i) {
  P[i].pos.x = P[i].pos.y = P[i].pos.z = 0;

  P[i].pos.x = L*uniformRand(generator);

  if(params.getDimensions() >=2)
    P[i].pos.y = L*uniformRand(generator);

  if(params.getDimensions() >=3)
    P[i].pos.z = L*uniformRand(generator);

  P[i].R = params.getParticleRadius();
  clusters.push_back( new Cluster(P, i) );

  globalxmin = (globalxmin < P[i].pos.x) ? globalxmin : P[i
      ].pos.x;
  globalxmax = (globalxmax > P[i].pos.x) ? globalxmax : P[i
      ].pos.x;
  globalymin = (globalymin < P[i].pos.y) ? globalymin : P[i
      ].pos.y;
  globalymax = (globalymax > P[i].pos.y) ? globalymax : P[i
      ].pos.y;
}
```

```
int iter=0;
int imgiter=0;

bool progress[10];
memset(progress, false, 10);
bool movement = true;

// compute "stable" time step
double magB = abs(params.getBxx())+abs(params.getBxy())+abs
    (params.getBxz()) +
   abs(params.getByx())+abs(params.getByy())+abs(params.
      getByz()) +
   abs(params.getBzx())+abs(params.getBzy())+abs(params.
      getBzz());


double dt= numeric_limits<double>::infinity();
if(magB != 0.0)
   dt = params.getParticleRadius()/L/magB;

// use randCoeff (fluffy factor) to compute a real
    diffusion constant

double DiffConst = params.getRandCoeff()*
   params.getParticleRadius()*params.getParticleRadius()*
      magB;

if (DiffConst == 0)
   DiffConst = 1;
```

```
// stable timestep is min of convection and diffusion
   stable steps
dt = min(dt, 0.1*params.getParticleRadius()/DiffConst);

double BrownianVariance = dt*DiffConst;

normal_distribution<double> normalRand(0.0, sqrt(
   BrownianVariance));

vector<string> vtkFnames;
vector<int> vtkTimes;
int nparticles = params.getNParticles();
while(clusters.size() > params.getClusterCutoff() &&
   movement) {

  //check writing image
  if(iter % params.getImageFreq() == 0 && params.
     getWriteImages()) {

    if(params.getOutputMode() == PackingParameters::
       IMAGE_OUTPUT) {
      double diffx = globalxmax-globalxmin;
      double diffy = globalymax-globalymin;
      double mindiff = (diffx < diffy) ? diffx : diffy;
      int maxdim = (params.getImageWidth()>params.
         getImageHeight()) ?
        params.getImageWidth() : params.getImageHeight();

      sprintf(imgnamebuf, "%s_%06d.png",params.
```

```
        getImageBasename().c_str(), imgiter);
    write_image_file(P, params.getNParticles(),
                     params.getImageWidth(),
                     params.getImageHeight(),
                     globalxmin,
                     globalymin,
                     maxdim/mindiff,
                     imgnamebuf);
  }
  else if(params.getOutputMode() == PackingParameters::
     VTK_OUTPUT) {
    sprintf(imgnamebuf, "%s_%06d.vtu",params.
       getImageBasename().c_str(), imgiter);
    vector<contact_t> contacts;
    calculate_contacts(P, nparticles, contacts);

    write_vtk_file(P, nparticles, contacts, imgnamebuf);
    vtkFnames.emplace_back(string(imgnamebuf));
    vtkTimes.push_back(imgiter);

  }
  ++imgiter;
}



//move clusters

movement = false;
double clusterxmin=L;
```

```
double clusterxmax=-1;
double clusterymin=L;
double clusterymax=-1;

for(unsigned i=0; i<clusters.size(); ++i) {
  vector_t v;
  v.x = v.y = v.z = 0;
  vector_t X;
  X.x = (clusters[i]->centroid.x - 0.5*L);
  X.y = (clusters[i]->centroid.y - 0.5*L);
  X.z = (clusters[i]->centroid.z - 0.5*L);

  v.x =
    -params.getBxx()*X.x
    - params.getBxy()*X.y
    - params.getBxz()*X.z
    + normalRand(generator)/dt/sqrt(clusters[i]->getSize
      ());
  if(params.getDimensions() >= 2)
    v.y =
      -params.getByx()*X.x
      - params.getByy()*X.y
      - params.getByz()*X.z
    + normalRand(generator)/dt/sqrt(clusters[i]->getSize
      ());
  if(params.getDimensions() >=3)
    v.z =
      -params.getBzx()*X.x
      - params.getBzy()*X.y
      - params.getBzz()*X.z
```

```
  + normalRand(generator)/dt/sqrt(clusters[i]->getSize
      ());

vector_t dx = vecmult(v, dt);

double magdisp = sqrt(dx.x*dx.x + dx.y*dx.y + dx.z*dx.z
    );
movement = movement || (magdisp > P[0].R/100);
clusters[i]->move(dx);

bool doPeriodicMove = false;
dx.x = dx.y = dx.z = 0;
if(clusters[i]->centroid.x > L) {
  dx.x = -L;
  doPeriodicMove = true;
}
else if(clusters[i]->centroid.x < 0) {
  dx.x = L;
  doPeriodicMove = true;
}
if(clusters[i]->centroid.y > L) {
  dx.y = -L;
  doPeriodicMove = true;
}
else if(clusters[i]->centroid.y < 0) {
  dx.y = L;
  doPeriodicMove = true;
}
if(clusters[i]->centroid.z > L) {
  dx.z = -L;
```

```
      doPeriodicMove = true;
    }
    else if(clusters[i]->centroid.z < 0) {
      dx.z = L;
      doPeriodicMove = true;
    }
    if(doPeriodicMove) {
      clusters[i]->move(dx);
    }

    clusterxmin = (clusterxmin<clusters[i]->getXMin()) ?
       clusterxmin : clusters[i]->getXMin();
    clusterxmax = (clusterxmax>clusters[i]->getXMax()) ?
       clusterxmax : clusters[i]->getXMax();
    clusterymin = (clusterymin<clusters[i]->getYMin()) ?
       clusterymin : clusters[i]->getYMin();
    clusterymax = (clusterymax>clusters[i]->getYMax()) ?
       clusterymax : clusters[i]->getYMax();

  }

  //check collisions
  for(unsigned i=0; i<clusters.size(); ++i) {
    unsigned j=i+1;
    while(j<clusters.size()) {
      if(clusters[i]->intersects(*clusters[j]) ) {
        clusters[i]->merge(*clusters[j]);
        delete clusters[j];
        clusters.erase( clusters.begin()+j );
      }
```

```
      else {
        ++j;
      }
    }
  }


if( params.getImageZoom() &&
    (clusterxmax-clusterxmin) < params.getZoomRatio()*(
      globalxmax-globalxmin) &&
    (clusterymax-clusterymin) < params.getZoomRatio()*(
      globalymax-globalymin)) {
  globalxmax = clusterxmax;
  globalxmin = clusterxmin;
  globalymin = clusterymin;
  globalymax = clusterymax;
}

//check to see if need to write incremental progress file
for(int i=1; i<10; ++i) {
  if(!progress[i-1] &&
     ((double)clusters.size())/params.getNParticles() <
       ((double)i/10.0)) {
    sprintf(datanamebuf, "%s_%02d", params.
      getPackingFname().c_str(), 10-i);
    fprintf(stdout, "writing %s\n", datanamebuf);
    write_data_file(P, params.getNParticles(),
      datanamebuf);
    progress[i-1]=true;
  }
```

```
    }

    ++iter;
  }//end while loop


  //data output (always do this)
  vector<contact_t> contacts;
  calculate_contacts(P, nparticles, contacts);
  write_data_file(P, params.getNParticles(), contacts, params
      .getPackingFname().c_str());

  if(params.getWriteImages() && params.getOutputMode()==
      PackingParameters::VTK_OUTPUT) {
    write_pvd_file(params.getImageBasename(), vtkFnames,
        vtkTimes);
  }

  // ================ CLEANUP ================
  for(unsigned i=0; i<clusters.size(); ++i) {
    delete clusters[i];
  }

  delete P;
  return 0;
}
```

Appendix B

Selected Source Code:

fit_steadyState_nonlinear.m

The following is a listing of the high-level Matlab routine that implements the procedure that fits the fabric evolution law to experimental data. This high-level function relies calls to many lower-level functions whose source codes are omitted for brevity. Comments have been added in the code, denoted by "%", that clarify the meaning of variables or where source code for important functions has been omitted.

```
function [model, model2] = fit_steadyState_nonlinear( fname,
    r_offset, c_offset, set_values, set_mask )

if(nargin < 3)
    fname = '06302015/steadystate_sweep/yI_pairs.csv';

    r_offset = 0;
    c_offset = 0;
    goodrow = 1;
end
```

```
if(nargin < 5)
    b1 = 1/250; %1/75;
    n = 1.34
    set_values = [1,0,0,b1,0,0,n];
    set_mask = [1,0,0,1,0,0,1];
end
set_mask = logical(set_mask);



model = struct('k1',0,'a1',0,'a2',0,'b1',0,'b2',0,'c3',0);
dat = dlmread(fname,',',r_offset, c_offset);
gooddat = dat;%[dat(2,:);dat(goodRow:end,:)];
shearRate = gooddat(:,1);
currentData = gooddat(:,2); %data in file is A
R = 0.02; % plate radius = 2.0cm = .02m
h = 0.00075; % plate gap = 0.75mm = .00075m
phi0 = .1; %V

Io = currentData(1);
k0 = Io*h/(phi0*(pi*R*R));

% Initial guess for
k10 = .001;
a10 = 2;
a20 = 1;
b10 = 1;
b20 = .1;
c30 = -1;
n0 = 0;
```

```
c0 = [k10 a10 a20 b10 b20 c30 n0];
c0(set_mask) = set_values(set_mask);

% Unconstrained optimization (finds good solution w/ good
    starting point)
c = fminunc(@objFunc, c0, opts)


%model.k1 = c(1);
model.a1 = c(2)*b1;
model.a2 = c(3)*b1;
model.b1 = b1;
model.b2 = c(5)*b1;
model.c3 = c(6)*b1;
model.k1 = k0*((model.a1/model.b1)^2)/((3*model.a1/model.b1 -
    2)^2);
model.n = n;

model2.k1 = model.k1;
model2.a1b1 = c(2);
model2.a2b1 = c(3);
model2.b2b1 = c(5);
model2.c3b1 = c(6);
model2.n = n;

    % This objective function is outlined in algorithm 2.2
    function f = objFunc(c)
        dR = R/100;
        r = 0:dR:R;
```

```
c(set_mask) = set_values(set_mask);

%k1 = c(1);
a1 = c(2)*b1;
a2 = c(3)*b1;
b1 = c(4);
b2 = c(5)*b1;
c3 = c(6)*b1;
n = c(7);

k1 = k0*((a1/b1)^2)/((3*a1/b1 - 2)^2);

computedCurrent = zeros(size(currentData));
% for each shear rate, integrate current at each R
    value
for ii = 1:length(shearRate)
    y0 = shearRate(ii);

    Kzzvec = zeros(size(r));
    for ri=1:length(r)
        d = D(r(ri), R, y0);
        y = ydot(r(ri), R, y0);

        % The following 13 lines implement algorithm
            2.1, computing the steady-state fabric
            tensor
        % Compute the nonlinear part of evolution law
        magD = norm(d,'fro');
        if(n == 0 || magD == 0)
            factor = 1;
```

```
        else
            % This function computes trA at steady
                state by solving equation (2.39)
            trAss = calc_trAss(a1,a2,b1,b2,n,magD);
            factor = abs(3*a1/b1 - trAss)^n;
        end

        c1 = C1(a1*factor, a2, d); % c1 = (a1*(Z0 -
            trA)^n + a2*|D|)
        c2 = C2(b1*factor, b2, d); % c2 = -(b1*(Z0 -
            trA)^n + b2*|D|)
        A = Fabric(c1, c2, c3, y); % This function
            solves equation (2.10) at steady state for
             the full fabric
        K = Conductivity(k1, A); % This function
            evaluates the conductivity according to
            the 3D Lattice Model

        Kzzvec(ri) = K(3,3);
    end
    %integrate (phi0/h)*Kzzvec*r*dr*dtheta from [0,R
        ], [0,2pi]
    kzzr = Kzzvec.*r;
    computedCurrent(ii) = (2*pi*phi0/h)*dR*(kzzr(1) +
        2*sum(kzzr(2:end-1)) + kzzr(end))/2;

end

% compute sum of squared squared-differences cost
f = sum((abs(currentData - computedCurrent).^2));
```

```
    end

end

function c = C1(a1, a2, D)
c = (a1 + a2*norm(D,'fro'));
end

function c = C2(b1, b2, D)
c = -(b1 + b2*norm(D,'fro'));
end

function y = ydot(r, R, y0)
y = y0*(r/R);
end

function d = D(r,R,y0)
d = zeros(3);
d(2,3) = y0*r/(2*R);
d(3,2) = d(2,3);
end

function K = Conductivity(k1, A)

K = k1 * A*( (max(0, sum(diag(A))-2) )^2 )/det(A);

end
```

Bibliography

[1] I. Alig, T. Skipa, M. Engel, D. Lellinger, S. Pegel, and P. Pötschke. Electrical conductivity recovery in carbon nanotubeâĂŞpolymer composites after transient shear. Physica Status Solidi (B), 244(11):4223–4226, November 2007.

[2] Takeshi Amari. Flow properties and electrical conductivity of carbon blackâĂŞlinseed oil suspension. Journal of Rheology, 34(2):207, February 1990.

[3] E. Azéma, N Estrada, and F. Radjai. Nonlinear effects of particle shape angularity in sheared granular media. Physical Review E, 2012.

[4] Katalin Bagi. An algorithm to generate random dense arrangements for discrete element simulations of granular assemblies. Granular Matter, 7(1):31–43, January 2005.

[5] GK Batchelor. Transport properties of two-phase materials with random structure. Annual Review of Fluid Mechanics, 1974.

[6] GK Batchelor and RW O'Brien. Thermal or electrical conduction through a granular material. Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences, 1977.

[7] W. Bauhofer, S.C. Schulz, a.E. Eken, T. Skipa, D. Lellinger, I. Alig, E.J. Tozzi, and D.J. Klingenberg. Shear-controlled electrical conductivity of carbon nanotubes networks suspended in low and high molecular weight liquids. Polymer, 51(22):5024–5027, October 2010.

[8] J. E. Bischoff, Ellen M Arruda, and K Grosh. A Microstructurally Based Orthotropic Hyperelastic Constitutive Law. Journal of Applied Mechanics, 69(5):570, 2002.

[9] John F. Brady and Georges Bossis. Stokesian dynamics. Annual review of fluid mechanics, 20(1):111–157, January 1988.

[10] John F. Brady, Ronald J. Phillips, Julia C. Lester, and Georges Bossis. Dynamic simulation of hydrodynamically interacting suspensions. Journal of Fluid Mechanics, 195:257–280, 1988.

[11] H. Cheng and S. Torquato. Effective conductivity of periodic arrays of spheres with interfacial resistance. Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences, 453(1956):145–161, January 1997.

[12] Frédéric da Cruz, Sacha Emam, Michaël Prochnow, Jean-Noël Roux, and François Chevoir. Rheophysics of dense granular materials: Discrete simulation of plane shear flows. Physical Review E, 72(2):021309, August 2005.

[13] Jessica M Deneweth, Scott G McLean, and Ellen M Arruda. Evaluation of hyperelastic models for the non-linear and non-uniform high strain-rate mechanics of tibial cartilage. Journal of biomechanics, 46(10):1604–10, June 2013.

[14] Mihai Duduta, Bryan Ho, Vanessa C. Wood, Pimpa Limthongkul, Victor E. Brunini, W. Craig Carter, and Yet-Ming Chiang. Semi-Solid Lithium Rechargeable Flow Battery. Advanced Energy Materials, 1(4):511–516, July 2011.

[15] Albert Einstein. On the movement of small particles suspended in stationary liquids required by the molecular-kinetic theory of heat. 322:549–560, 1905.

[16] Kenneth Falconer. Fractal geometry: mathematical foundations and applications. John Wiley & Sons, 2013.

[17] Charles O Frederick and PJ Armstrong. A mathematical representation of the multiaxial bauschinger effect. Materials at High Temperatures, 24(1):1–26, 2007.

[18] Vincent Grenard, Thibaut Divoux, Nicolas Taberlet, and Sebastien Manneville. Timescales in creep and yielding of attractive gels. Soft Matter, 10:1555–1571, 2014.

[19] Morton E Gurtin, Eliot Fried, and Lallit Anand. The mechanics and thermodynamics of continua. Cambridge University Press, 2010.

[20] George L. Hand. A theory of anisotropic fluids. Journal of Fluid Mechanics, 13(01):33, March 1962.

[21] John Happel and Howard Brenner. Low Reynolds number hydrodynamics: with special applications to particulate media, volume 1. Springer Science & Business Media, 2012.

[22] Z. Hashin and S. Shtrikman. A Variational Approach to the Theory of the Effective Magnetic Permeability of Multiphase Materials. Journal of Applied Physics, 33(10):3125, 1962.

[23] Ahmed Helal, T. Divous, XW Chen, and G. H. McKinley. In-situ simultaneous rheo-conductivity and rheo-impedance spectroscopy measurements of strongly conductive complex fluids. In Preparation.

[24] Ahmed Helal, Kyle Smith, Frank Fan, Xin Wei Chen, Joao Miguel Nobrega, Yet-Ming Chiang, and Gareth H. McKinley. Study of the rheology and wall slip of carbon black suspensions for semi-solid flow batteries. The Society of Rheology 86th Annual Meeting, October 2014.

[25] H Hoekstra, J Vermant, J Mewis, and GG Fuller. Flow-induced anisotropy and reversible aggregation in two-dimensional suspensions. Langmuir, (11):9134–9141, 2003.

[26] A Jagota and C Y Hui. The Effective Thermal Conductivity of a Packing of Spheres. J. Appl. Mech, (September 1990):789–791, 1990.

[27] TA Witten Jr and LM Sander. Diffusion-limited aggregation, a kinetic critical phenomenon. Physical review letters, 47(19), 1981.

[28] TA Witten Jr and LM Sander. Diffusion-limited aggregation, a kinetic critical phenomenon. Physical review letters, 47(19), 1981.

[29] Jerrold E Marsden and Thomas JR Hughes. Mathematical foundations of elasticity. Courier Corporation, 1994.

[30] James Clerk Maxwell. A treatise on electricity and magnetism, volume 1. Clarendon press, 1881.

[31] Morteza M. Mehrabadi, S Nemat-Nasser, and M Oda. On statistical description of stress and fabric in granular materials. International Journal for ..., 6(November 1980):95–108, 1982.

[32] Jeffrey F. Morris and Bhavana Katyal. Microstructure from simulated Brownian suspension flows at large shear rate. Physics of Fluids, 14(6):1920, 2002.

[33] M Oda, S Nemat-Nasser, and Morteza M. Mehrabadi. A statistical study of fabric in a random assembly of spherical granules. International Journal for Numerical and Analytical Methods in Geomechanics, 6(July 1982):77–94, 1982.

[34] Tyler J Olsen and Ken Kamrin. Modeling tensorial conductivity of particle suspension networks. Soft Matter, 2015.

[35] F. Radjai, J.-Y. Delenne, E. Azéma, and S. Roux. Fabric evolution and accessible geometrical states in granular materials. Granular Matter, 14(2):259–264, March 2012.

[36] Ronald S Rivlin. Further remarks on the stress-deformation relations for isotropic materials. Journal of rational mechanics and analysis, 4(5):681–702, 1955.

[37] M Satake. Constitution of mechanics of granular materials through the graph theory. Continuum Mechanical and Statistical Approaches in the Mechanics of Granular Materials, pages 47–62, 1978.

[38] S.C. Schulz and Wolfgang Bauhofer. Shear influenced network dynamics and electrical conductivity recovery in carbon nanotube/epoxy suspensions. Polymer, 51(23):5500–5505, October 2010.

[39] Kyle C Smith, Yet-Ming Chiang, and W. Craig Carter. Maximizing Energetic Efficiency in Flow Batteries Utilizing Non-Newtonian Fluids. Journal of the Electrochemical Society, 161(4):A486–A496, January 2014.

[40] Michael J Stephen and Joseph P Straley. Physics of liquid crystals. Reviews of Modern Physics, 46(4):617, 1974.

[41] Jin Sun and Sankaran Sundaresan. A constitutive model with microstructure evolution for flow of rate-independent granular materials. Journal of Fluid Mechanics, 682:590–616, July 2011.

[42] S. Torquato. Effective electrical conductivity of twoâĂŘphase disordered composite media. Journal of Applied Physics, 7910(November):3790–3797, 1985.

[43] S. Torquato. Random heterogeneous materials: microstructure and macroscopic properties, volume 16. Springer, 2002.

[44] S. Torquato and G. Stell. Microstructure of two-phase random media. I. The n-point probability functions. The Journal of Chemical Physics, 77(4):2071, 1982.

[45] J Vermant and M J Solomon. Flow-induced structure in colloidal suspensions. Journal of Physics: Condensed Matter, 17(4):R187–R216, February 2005.

[46] Mohamed Youssry, Lénaïc Madec, Patrick Soudan, Manuella Cerbelaud, Dominique Guyomard, and Bernard Lestriez. Non-aqueous carbon black suspensions for lithium-based redox flow batteries: rheology and simultaneous rheo-electrical behavior. Physical chemistry chemical physics : PCCP, 15(34):14476–86, September 2013.