

The Pennsylvania State University

The Graduate School

**COMPUTATIONAL OPTIMIZATION OF MATERIAL PROPERTIES VIA  
A MULTI-FIELD-ASSISTED ADDITIVE MANUFACTURING PROCESS**

A Dissertation in

Mechanical Engineering

by

Denise Widdowson

Submitted in Partial Fulfillment  
of the Requirements  
for the Degree of

Doctor of Philosophy

May 2023

The dissertation of Denise Widdowson was reviewed and approved by the following

Paris von Lockette  
Associate Professor, Mechanical Engineering  
Dissertation Advisor  
Chair of Committee

Zoubeida Ounaies  
Director, Convergence Center for Living Multifunctional Material Systems (LiMC2)  
Professor, Mechanical Engineering

Mary Frecker  
Department Head  
Riess Chair of Engineering, Director of the Penn State Center for Biodevices  
Professor, Mechanical Engineering & Biomedical Engineering

Allison Beese  
Director, Additive Manufacturing & Design Graduate Program  
Associate Professor, Materials Science and Engineering and Mechanical Engineering

## ABSTRACT

Functional additively manufactured parts necessitate variable material properties such as conduction or insulation, stiffness or compliance, or a range of any material response. This range of material responses can be achieved by using multiple materials, mixing materials, or as will be shown in this work, by controllably varying the properties of a particulate filled polymer matrix composite material *via* electromagnetic processing conditions. Past experimental and computational research has shown that, in this fashion, for a fixed pairing of matrix and filler, composite material properties can be tailored not only by changing the constituents' volume fractions, but also by manipulating the microarchitecture that the filler materials self-organize into during the processing event. Organization occurs because applying external electric and magnetic fields causes particles to orient and align themselves as they react both to the external applied field and the stray fields of other nearby particles. These distinct distributions of particles affect the composite's material properties in all domains including elastic, magnetic, and dielectric. Having this range of material properties available within a single print is crucial for the development of a universal 3-D printer. This work will establish the previously undetermined relationship between processing parameters and effective properties to optimize the constituent set for the largest range of possible properties, building a framework for materials design applications.

Consequently, it is of particular interest to materials design to understand the relationship between the applied field and the resulting microarchitectures and thereby the resulting effective properties. This dissertation focuses on the key components to creating a framework that predicts those composite properties from processing parameters and constituent sets, and determines the processing parameters for a desired set of material properties (the inverse

problem). This identification of required processing parameters will be done by investigating the established homogenization techniques and their limits, developing a computational homogenization scheme, and incorporating modeling methods into optimization and data science structures, all to investigate the processing – constituent design space.

In this work, particle dynamics simulations will form the basis for computational modeling methods by determining the particle microarchitectures formed from coupled electric and magnetic processing fields. This simulation will create the representative volume element (RVE), which with proper boundary conditions and sufficient size, will yield bulk material properties (magnetic, elastic, and dielectric) via computational homogenization using finite element analysis (FEA). With an established pathway from processing parameters to composite properties, the system can be optimized for the maximum and minimum properties and find the widest range possible, for any given set of constituents. With data collected, from experiments, computational simulations, and analytical modeling, data science techniques including regressions can be trained to augment the homogenization, and to solve the inverse problem of determining processing parameters for a selected material property.

This work shows the path from processing parameters to effective properties and trains a model to effectively predict properties for use with an optimization scheme. The multi-objective optimization scheme presented here finds the constituent set for which the combined range of elastic, magnetic, and dielectric properties is largest. Finally, the trained model is used to further explore the design space, generate Pareto fronts, and address the inverse problem by determining the processing parameters that would achieve desired properties. It was shown that the volume fraction determines the possible ranges and



tradeoffs between properties. The relative strength and directions of externally applied processing fields were shown to be critical for structures formed and therefore also resulting effective properties. It was determined that for use with optimization or materials design applications data science techniques are crucial for predicting system behavior for the feasibility of exploring the design space. This work presents the developed framework that models structures, determines properties, predicts composite behavior within the design space, and optimizes the constituent sets.

## TABLE OF CONTENTS

List of Figures .....	ix
List of Tables .....	xxiii
<b>Acknowledgments</b> .....	xxiv
<b>CHAPTER 1: Introduction</b> .....	1
1.1 Overview .....	1
1.2 Literature Survey .....	4
1.2.1 Motivation: Multi-Material Additive Manufacturing .....	4
1.2.2 Processing Magneto Active Polymers to Control the Microarchitecture Within .....	9
1.2.3 Estimating Composite Material Properties .....	17
1.2.3.1 Effective Medium Theories (EMT) .....	18
1.2.3.2 Homogenization .....	22
1.2.3.3 Using Artificial Intelligence to Predict Conductivity .....	27
1.2.4 Optimization via Traditional Gradient Based Techniques and Data Science Modeling..	30
1.2.5 Artificial Intelligence and Machine Learning Models for Defining the Design Space....	34
1.3 Research Objectives and Tasks .....	42
1.4 Dissertation Outline .....	44
<b>CHAPTER 2: Methods</b> .....	46
2.1 Homogenization .....	46
2.1.1 Particle Dynamics to Predict Microarchitecture .....	46
2.1.2 Computational Finite Element Homogenization.....	55
2.1.2.1 Elastic Boundary Conditions and Effective Properties.....	57
2.1.2.2 Dielectric Boundary Conditions and Effective Properties.....	61
2.1.2.3 Magnetic Boundary Conditions and Effective Properties.....	65
2.3 Artificial Intelligence Methods to Predict Conductivity .....	68
2.3.1 Path Finding .....	69
2.3.2 Cutoff Distance .....	72
2.3.3 Effective Resistance .....	73
2.2 Estimating Composite Properties using Effective Medium Theory.....	76

2.2.1 Elastic Domain Models .....	76
2.2.2 Dielectric and Magnetic Domain Models .....	88
2.4 Multi-Objective Optimization .....	91
2.4.1 Nested Optimization Structure .....	92
2.4.2 Combined Fitness Function and Pareto Front .....	95
2.5 Artificial Intelligence and Machine Learning Models .....	98
2.6 Summary .....	105
<b>CHAPTER 3: Computational Homogenization Results and Discussion.....</b>	<b>106</b>
3.1 Effective Medium Approaches.....	106
3.2 COMSOL Simulations and comparison to Effective Medium Theory.....	107
3.2.1 Convergence Study on Model Size for Elastic Properties .....	110
3.2.2 Compliance and Moduli versus Volume Fraction.....	118
3.2.3 Homogenized Anisotropic Dielectric Response versus Volume Fraction .....	123
3.3 Comparison of Experimental Results versus Simulation Results .....	128
3.3.1 Homogenized Anisotropic Elastic Response versus Volume Fraction.....	128
3.3.2 Homogenized Anisotropic Dielectric Response versus Volume Fraction .....	133
3.3.3 Homogenized Anisotropic Magnetic Response versus Volume Fraction.....	136
3.4 Particle Spacing and Resulting Model Fidelity.....	139
3.5 Artificial Intelligence Framework for Predicting Conductivity .....	142
3.6 Summary .....	149
<b>CHAPTER 4: Multi-Objective Optimization Results and Discussion .....</b>	<b>150</b>
4.1 Crossover Torque .....	151
4.2 Nested Optimization Framework .....	156
4.3 Multi-Objective Optimization with Integrated Machine Learning .....	163
4.4 Pareto Front .....	174
4.5 Summary .....	180
<b>CHAPTER 5: Machine Learning and the Inverse Problem .....</b>	<b>182</b>
5.1 Machine Learning Model Selection and Training.....	182
5.2 The Inverse Problem .....	188
5.3 Summary .....	192
<b>CHAPTER 6: Conclusion .....</b>	<b>194</b>
6.1 Conclusions .....	194
6.2 Contributions .....	196

6.3 Future Work .....	197
<b>References</b> .....	200

## LIST OF FIGURES

Figure 1-1: Categories of additive manufacturing as defined by ASTM52900:201 [1] with pros and cons of each method [2] .....	2
Figure 1-2: Bi-extruder for FDM multi-material additive manufacturing a) exploded view of the bi-extruder, b) full assembly of the custom bi-extruder, c-i) CAD model of the inter-mixer, c-ii) fabricated inter-mixer, c-iii) inter-mixer assembled into bi-extruder channel [2], [11] .....	6
Figure 1-3: Gradient in platelet concentration and alignment as a result of varied volume fraction mixes and applied magnetic field during printing of the top (I-II) and bottom (III-VI) layer of the structure [18] .....	9
Figure 1-4: Magnetic surface instabilities a) showing field lines that are compressed and uncompressed due to the peaks and valleys in the surface that minimize the magnetic energy as compared to a planar surface b) experimental instability shown for field versus wave number [48] .....	10
Figure 1-5: Bright field optical micrographs of superparamagnetic microspheres, effects of simultaneous electric versus magnetic field of varying strengths, demonstrating the chaining is determined by the direction and relative intensities of the applied fields. [19] .....	11
Figure 1-6: a-b) SEM image showing particle chain and Barium hexaferrite (BHF) stacks inside the chains, respectively, for parallel applied fields and c-d) SEM image showing part of a chain and BHF stacks inside the chains, respectively, for perpendicular applied fields adapted from [20] .....	12
Figure 1-7: Microarchitecture of Barium Hexaferrite in PDMS (a) before the application of any field, (b) after the application of a magnetic field, and (c) after the application of both electric	

and magnetic fields showing the increased ordering present with one and both applied fields, respectively [37].....	13
Figure 1-8: Simulation result for model with 200 particles a) magnetic field only and b) magnetic and electric field, showing the improved chaining for the combined field case [37] .....	13
Figure 1-9: OM/SEM of frequency versus. peak-to-peak voltage showing difference structures as a function of field strength and frequency where increases in electric field improves the chaining response at low frequency, and for higher frequency chaining does not occur [20] .....	15
Figure 1-10: Microscope image at low (a) and high (b) magnification of particle network formed in the presence of perpendicular electric and magnetic fields [19] .....	17
Figure 1-11: Voigt and Reuss model setting upper and lower bounds, respectively, for effective composite properties .....	19
Figure 1-12: Double cell model for the Ponte-Castaneda model adapted from [23].....	23
Figure 1-13: Periodic element visualization showing repeating units to represent translation of boundary conditions to paired faces .....	25
Figure 1-14: Bulk and shear modulus, respectively, versus model size comparing kinematically uniform, statically uniform, and periodic boundary conditions to effective medium bound (Voigt and Reuss) [24] .....	26
Figure 1-15: Diagram of a simple graph with both spanning tree paths as well as all relevant non-looping connective paths (gray lines are segments that are not considered, black lines are segments that do not connect the two terminals, green lines show paths within each graph that	

connect the terminals) to demonstrate the excess segments counted as well as the paths missed by the spanning tree that would contribute to a conductive network .....	29
Figure 1-16: Ideal problem domains for selected data science and AI techniques (Decision Tree, Gaussian Process, Support Vector Machine (SVM), Bayesian Networks, Neural Networks (NN), k Nearest Neighbor (kNN)) .....	35
Figure 1-17: Example of a feed forward neural network (FFNN) showing inputs on the left, a dense layer in the middle, outputs on the right, and an example of the weighted sum of the input or any node's previous layer. ....	36
Figure 1-18: Examples of activation functions for use with a neural network.....	37
Figure 1-19: Distinction between SVM method (purple box) and Neural Network (blue box) of classification showing the method used to draw the decision boundary leading to a more robust model (better avoids misclassification) with the SVM method (adapted from [118]) .....	38
Figure 1-20: a) Ordinary Least Squares (OLS) compared to b) Support Vector Regression (SVR) model (for linear, quadratic and polynomial functions) showing the different approaches to fitting data where the OLS method attempts to minimize the squared error between the fit and the data, as opposed to the SVR which minimizes the weights of the regression terms within specified bounds.....	39
Figure 1-21: Linear Support Vector Regression fit as the allowed error goes to 0 .....	40
Figure 2-1: Idealized ellipsoidal representation of BHF showing major (a/b) and minor (c) axes .....	47

Figure 2-2: magnetic alignment of ellipsoidal particles that are governed by a separation force assuming a spherical shape .....	52
Figure 2-3: Geometric values used to calculate the surface to surface distance between two ellipsoids $dR$ is the minimum distance between the surfaces, $\mathbf{R}$ is the center to center vector $xa$ and $xb$ are points on the surface of the two ellipsoids (A and B, respectively), and $r$ and $s$ are the center points of ellipsoids A and B, respectively.....	54
Figure 2-4: RVE showing particle microarchitecture formed from applying various fields .....	56
Figure 2-5: Scripting workflow from particle dynamics simulation to effective material properties.....	57
Figure 2-6: Load case 1 applied strain (yellow) and resulting strains (purple) for an RVE with periodic boundary conditions.....	59
Figure 2-7: Equivalent circuit for a dielectric material in an experiment test setup showing the applied voltage and resulting current.....	61
Figure 2-8: Theoretical response of a dielectric system whose equivalent circuit can be represented by Figure 2-7, showing the input voltage $V_{in}$ , the total output current $I_{out}$ , the capacitive component of the output current $I_{cap}$ , and the conductive component of the output current $I_{cond}$ .....	62
Figure 2-9: Representation of boundary conditions on the dielectric RVE a) electric field, $E$ , from applied terminal voltage, $V_{in}$ , and ground on the top and bottom faces, respectively b) periodic boundary conditions (pink).....	63



Figure 2-10: Mass magnetizations for Case 3 (Magnetic Field only, 100 particle RVE's) for magnetic vector sum versus full field magnetization response illustrating the effect of the particle interactions accounted for in the COMSOL model but not in the vector sum method .....	68
Figure 2-11: Diagram showing potential pathways between particles that can result in percolating paths used for calculating effective resistance of the RVE from connecting paths by treating the elements of one path as resistors in series and then combining all paths in parallel .....	69
Figure 2-12: a.) Spanning tree representation of a graph b.) Graph showing only paths that span the start and finish nodes.....	71
Figure 2-13: Idealized conductive pathlength through a particle based on the position of neighboring particles.....	74
Figure 2-14: Example of test case and Simulink model of resistors for determining effective composite resistance .....	75
Figure 2-15: Simulink model showing boundary conditions used to determine effective resistance of composite where the composite resistance is shown as $R_{effective}$ for simplicity .	75
Figure 2-16: Diagram of strain and stress assumptions in Voigt and Reuss models respectively.	79
Figure 2-17: Effect of aspect ratio, $w/t$ , on effective modulus, normalized to matrix modulus for the case of ellipsoidal filler . Results show that as $w/t$ increases, $E/Em$ decreases, and the effective modulus become less dependent on volume fraction. ....	81
Figure 2-18: Effective medium models moduli versus volume fraction showing the relationship of effective modulus to volume fraction for Halpin Tsai and Mori Tanaka models and Voigt and Reuss bounds .....	83

Figure 2-19: Double inclusion model diagram, inclusion phase  $\Omega$  with elasticity  $C1$  embedded in the double-cell  $R = \Gamma + \Omega$  with elasticity  $C2$  in infinite medium  $B$  with elasticity  $C$  [140]..... 86

Figure 2-20: Representative results of the two optimization loops shown for purposes of illustration: a) searching for the minimum and maximum effective property by varying representative processing field only yields the range. Dotted lines highlight max and min processing condition locations. The max and min also yield the range of possible properties. b) searching for the maximum range of possible properties by varying the constituent set, e.g. volume fraction. For a given volume fraction, varying processing leads to high and low (and intermediate) values from which a range is found for a given volume fraction. .... 93

Figure 2-21: Workflow of nested optimization scheme showing the computational homogenization, property minimization and maximization, and the optimization of the constituent set to maximize the range of properties possible ..... 94

Figure 2-22: Representation of multi-objective optimization space a) design space contour showing selection of specific constituent sets  $x1(1), x1(2), x1(3)$  and b) plot of property evaluation vs processing parameters showing the minimum and maximum property for a single constituent set by varying processing parameters alone ..... 95

Figure 2-23: Example of error bounds, shown in pink, on a sample regression, represented as a blue dashed line, with data points that fall inside and outside of the bounds. The points outside of the shaded pink acceptable error bounds would be considered in the minimization of the SVR model training ..... 101

Figure 2-24: Support Vector Regression (SVR) of noisy exponential test function (black circles) showing the fitted function and sample error bounds lines to illustrate the objective function SVR uses to adjust the weights. Teal points are support vectors that define the regression .....	101
Figure 2-25: Succession of point selection using SmartSearch with an SVR model (shown in purple) based on largest distance between independent variables and largest error in outputs (shown in teal) to define independent variables for the next test point (shown in blue) .....	102
Figure 2-26: Hybrid optimization workflow (using smart search and gradient based optimization) .....	104
Figure 3-1: Effective medium models for BHF/PDMS from 1 to 10% volume fraction showing the bounds and expected trend of properties.....	107
Figure 3-2: Cases 1-5 microarchitecture for simulations with 100 particles showing the effect of applied fields on the orientation, position, and microarchitecture of the particles .....	110
Figure 3-3: Figures showing microarchitecture versus simulation size where chain formation becomes more well defined with increasing particle count .....	112
Figure 3-4: Convergence study of compliance versus simulation size (particle count) at 10% volume fraction of an electrically aligned case compared to effective medium theory .....	114
Figure 3-5: Dielectric convergence study showing relative permittivity in three directions for a) 1% volume fraction b) 5% volume fraction and c) 10% volume fraction versus particle count (simulation size) and the trend with respect to the Bruggeman effective medium model (dashed red line) for reference to determine convergence at each volume fraction.....	117

- Figure 3-6: Dielectric convergence study showing relative permittivity in three directions for 1%, 5%, and 10% volume fraction versus particle count (simulation size) showing the anisotropy vs model size of each volume fraction tested..... 118
- Figure 3-7: Elastic modulus versus various volume fraction of no field (Case 1, green), electric field only (Case 2, blue), and magnetic field only (Case 3, red) cases having 100 particles each, showing the increased anisotropy for Case 2 which has chaining in the z direction..... 119
- Figure 3-8: Elastic modulus versus various volume fraction of no field (Case 1, green), electric field (Case 2, blue), and magnetic field only (Case 3, red) cases (100 particles) compared to EMT (Halpin Tsai, Mori Tanaka, Reuss Bounds) showing the agreement of the computational data in the Ex/y directions and the increased stiffness as compared to the EMT models in the direction of the particle chaining for Case 2 (Z direction)..... 122
- Figure 3-9: Dielectric response (relative permittivity) versus volume fraction in the x, y, and z directions. Results show the increased relative permittivity for cases 2 and 4 that have dense chains along the electric field dimension, e.g. for  $\epsilon_z$  ..... 125
- Figure 3-10: Ratio of maximum to minimum permittivity for each alignment case at 1, 5, and 10% volume fraction, showing the degree of anisotropy of each case with varying volume fraction, where case 4 has the highest degree of anisotropy in relative permittivity because all of the particles are oriented and aligned in chains ..... 125
- Figure 3-11: DC dielectric Data for 5 cases in the z direction compared to Bruggeman Effective Medium Model for 1%, 5%, and 10% volume fraction..... 127
- Figure 3-12: Experimental values (from separate samples) for elastic modulus in the direction of the applied electric field (z- direction) for compressive testing at 5 and 15 wt% (equivalent

0.91% and 2.84% volume fraction, respectively) for a) Compression Tests (pink box highlights typical range of moduli for pure PDMS as reference, range reported by Wang et al. [149]) and b) Tensile Tests ..... 130

Figure 3-13: Results to compare potentially lower matrix modulus effects on effective elastic results a) Experimental results for compression testing at 5wt% (0.91% volume fraction) b) computation homogenization simulation data from early models with PDMS modulus set to 190kPa at 1%, 5%, and 10% volume fraction ..... 131

Figure 3-14: Elastic modulus in the z-direction for 5 cases a) Experimental tension test data (5wt% and 15wt%, equivalent 0.91% and 2.84% volume fraction, respectively) b) Predicted data from machine learning model trained on simulation data ..... 132

Figure 3-15: Effective dielectric properties a) experimental (5wt%, ~1% volume fraction) showing higher permittivity values for electrically aligned cases (2, 4, and 5) and b) simulation (1,5, and 10% volume fraction) results in the z direction (electric field direction)..... 135

Figure 3-16: a) Remanent mass magnetization and b) Saturation magnetization for Case 1,2,3, and 5 for experimental samples with a target of 15wt% (2.84% volume fraction) filler content normalized to particle mass ( $g^*$ )..... 138

Figure 3-17: Comparison between the in plane and out of plane magnetization presented for a) adjusted experimental values for 5 and 15wt% and b) simulation results for 1,5, and 10% volume fraction ..... 138

Figure 3-18: Particle distributions for calibration term  $\beta$  at 2, 10, and 100 with spherical and ellipsoidal distance calculations for determine the repulsive force between contacting particles ..... 141

Figure 3-19: Representations of all conductive pathways for simple test case a) connect graph showing all conductive pathways and b) equivalent matrix form of the solution .....	145
Figure 3-20: Results from simple test case showing the conductive pathways found by the search algorithm in command line output format .....	145
Figure 3-21: Pathfinding algorithm results showing "all paths" graph, where conductive paths are shown in green between particles whose centers are shown by the red points for an RVE with 100 particles and 10% volume fraction .....	146
Figure 3-22: Simulated resistivity results for 1nm diameter spherical Nickel-Polystyrene composite at 60, 70, and 80 wt% (14%, 20%, and 30% volume fraction, respectively) .....	147
Figure 3-23: Resistivity data from Chen et al. [150] showing the effect of volume fraction on resistivity of various Nickle-Polystyrene composite compositions used as a reference for comparison with simulated conductivity results .....	148
Figure 4-1: Resulting torque from a) polarization from externally applied electric field and b) magnetization from magnetic dipole and applied magnetic field (vector quantities vary with the direction of the applied field with respect to the particle orientation) .....	152
Figure 4-2: Particle distributions for systems with varying combinations of electric and magnetic field applied in the z direction (vertical direction in each figure), showing the competing nature of the torques generated from the two fields to orient the particles with the respective field ....	152
Figure 4-3: Particle distributions example for 1% volume fraction with 100 particles a 250,000 A/m magnetic field in the z-direction and increasing parallel electric field from 0-1e8 V/m	

showing the effect of field strength on microarchitecture and demonstrating the crossover torque .....	154
Figure 4-4: Theoretical electric and magnetic field generated torques vs applied field strength shown for 3 particle angles ( $0, \pi/4, \pi/2$ ) .....	155
Figure 4-5: Results of gradient based optimization for mass magnetization showing max mass magnetization vs applied electric and magnetic fields for all volume fractions, with points of interest marked 1-2) no change in magnetization with increased electric field only 3-4) decreasing maximum magnetization due to volume fraction change (left to right the points here are for 11, 10, 7 and 28% volume fraction where the magnetization decreases with increased volume fraction) 5) highly sampled region to determine maximum which is shown to be at magnetic field only, orienting the particles' magnetizations with the applied field .....	161
Figure 4-6: 4D design space representation of mass magnetization optimization results with applied field along the three axes and color representing volume fraction to demonstrate the exploration of the design space and supplement results in Figure 4-5, with the same points of interest marked for reference purposes .....	161
Figure 4-7: Results of gradient based optimization for mass magnetization showing all magnetizations found vs volume fraction to illustrate the range of possible properties as a function of the constituent set, noting the decrease in range at 28% volume fraction, likely due to lack of particle mobility in the composite to form favorable structures .....	162
Figure 4-8: Design space plot with applied fields on the three axes and volume fraction represented in the color bar, showing initialization points and predicted points where effective property minima and maxima occur .....	165

Figure 4-9: Maximums and minimum of a) modulus b) mass magnetization c) relative permittivity vs volume fraction as calculated by the trained SVR model showing the range of properties used by the objective function in the optimization framework..... 169

Figure 4-10: Predicted property range results for magnetic (orange), elastic (blue), dielectric (green), and the evaluation of the combined objective function (black) versus volume fraction170

Figure 4-11: Design space search results from preliminary runs of the framework (that have since been modified), showing the applied fields on the major axes and the volume fraction in point color, to demonstrate the exploratory nature of the optimization framework when training. The lines indicate the order of the points tested, excluding the original 10 Latin Hypercube sample points to demonstrate the exploratory nature of the machine learning augmented optimization framework ..... 173

Figure 4-12: Design space representation from predicted SVR machine learning model results of a) elastic modulus vs magnetization, b) relative permittivity vs magnetization, c) relative permittivity vs elastic modulus showing the possible ranges of property combinations for 4 volume fractions: 1% (yellow), 5% (purple), 10% (green), and 30% (magenta), showing the tradeoffs between properties ..... 175

Figure 4-13: Design space area representation from predicted SVR machine learning model results of a) elastic modulus vs magnetization, b) relative permittivity vs magnetization, c) relative permittivity vs elastic modulus showing the possible ranges of property combinations for 4 volume fractions: 1% (yellow), 5% (purple), 10% (green), and 30% (magenta), showing the tradeoffs between properties ..... 178



- Figure 4-14: Calculated percent comparison of design space area from Figure 4-13 for a) elastic modulus vs magnetization, b) relative permittivity vs magnetization, c) relative permittivity vs elastic modulus normalized to the minimum area for each property combination showing the range of possible property combinations with change in volume fraction ..... 178
- Figure 4-15: Pareto front from SVR machine learning model predictions of a) elastic modulus vs magnetization, b) relative permittivity vs magnetization, c) relative permittivity vs elastic modulus for 4 volume fractions: 1% (yellow), 5% (purple), 10% (green), and 30% (magenta), showing the tradeoffs between properties..... 179
- Figure 5-1: FFNN trained model vs test data for z direction magnetization as a function of magnetic fields (not shown are the applied electric field and volume fraction that are also independent variables in the design space and the FFNN) ..... 185
- Figure 5-2: FFNN predictive surface (shown with colormap of Z axis value for visual clarity) overlaid on model test and predictive data points, showing the interpolated behavior defined by the kernel selection and availability of test points ..... 185
- Figure 5-3: SVR Trained on 80% of magnetic data from an optimization run, 20% used for testing is shown in the grey and red points, the surface is for 0V/m electric field and 0.1 volume fraction (color scale for visualization of 3D space only)..... 187
- Figure 5-4: SVR Trained on an experimental data set, data points are shown in the grey and predictions in red points, the surface is for 0V/m electric field and 0.1 volume fraction (color scale for visualization of 3D space only)..... 187

Figure 5-5: Diagram showing the workflow of the HashLookUp algorithm, mapping the design space, storing properties as hash keys with corresponding processing condition values for quick access lookup .....	190
Figure 5-6: HashLookUp algorithm search results for solving the inverse problem.....	191
Figure 5-7: HashLookUp algorithm search results for solving the inverse problem.....	192
Figure 0-1: Flow Chart of All Paths Search Algorithm.....	215

## LIST OF TABLES

<b>Table 1: Reinforcement factor equations for various filler types with <math>0 \leq vf \leq 1</math> .....</b>	<b>81</b>
<b>Table 2: Multi-objective optimization fitness function weighting methods [145] .....</b>	<b>97</b>
<b>Table 3: Model Conditions that are additional arguments to be used to reduce run time and improve training efficiency of Neural Networks.....</b>	<b>100</b>
<b>Table 4: Constituent Material Properties.....</b>	<b>108</b>
<b>Table 5: Applied fields for each of the 5 cases shown in Figure 3-2.....</b>	<b>110</b>
<b>Table 6: Bounds on the optimization scheme independent variables (volume fraction and applied electric and magnetic fields). .....</b>	<b>157</b>
<b>Table 7: Top 3 minimum and maximum points found at 10% volume fraction (3 points are shown to demonstrate the multiple processing conditions that yield the same optimum property) The first row for the minimums and maximums each are the respective optimum solution found .....</b>	<b>162</b>

## Acknowledgments

I would first like to thank my advisor, **Dr. Paris von Lockette**, for the support and encouragement throughout my graduate school journey. I am thankful for the challenges he presented me with and for the guidance to overcome them. Anyone that has had the pleasure to work with him can see his dedication to research and his students.

I would like to thank **Dr. Zoubeida Ounaies** and **Dashiell Papula** in the EMC Lab for the collaboration on this project. The lessons I have learned in this joint effort are innumerable. It goes without saying that this project would not have been possible without the partnership between the two labs, which most times felt like one cohesive lab group.

Both Dr. Paris von Lockette and Dr. Zoubeida Ounaies have taken time to teach and mentor me, pushing me to extend my initially intended academic path into a PhD. I would not be here without their beliefs and efforts.

I would like to thank my committee members, **Dr. Mary Frecker** and **Dr. Alison Beese** for their feedback and advice throughout this project. The discussions have elevated the quality of this work and my understanding of the subject.

I am grateful for my lab group and collaborators, including **Christian Bergen**, **Dr. Anil Erol** and **Dr. Md Abdulla Al Masud**. The friendship and teamwork made working with this group an absolute joy.

To my parents and grandparents, thank you for always encouraging me to pursue my dreams and being there through all the ups and downs. Thank you for the time and effort that I know went into supporting this journey. There truly aren't words for the love and support that I have felt throughout my life. Thank you for sparking my interest in science, for keeping my life balanced, and helping me become the person I am today.

Thank you to my family and friends, in particular G.B. Widdowson and A.B. Carson, for being there for me when I needed it.

Finally, I am grateful to my partner, **Christopher Carson**, without whom I would not have considered graduate school. Thank you for believing in me and making this possible. Thank you for the care and support through the most stressful moments and for sharing in the joy of all of the accomplishments.

I would also like to thank and acknowledge the National Science Foundation (NSF CMMI-176218) for funding this work. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the National Science Foundation.

## CHAPTER 1: Introduction

### 1.1 Overview

Additive manufacturing (AM), which can be classified into seven general categories as laid out by ASTM52900:201 [1] (see Figure 1-1), allows for the manufacturing of parts with complex functionality derived from incorporation of multiple materials in a single print. In response to growing demand for functional AM, materials providing a range of functionalities have been used concurrently within a single print to serve an array of applications. These methods include composite inks or filaments with additives to promote electrical conductivity, strength, ductility, magnetic permeability, or dielectric responses and more [2]–[7]. Here, the base material is enhanced with filler materials, producing a composite whose functional properties are tailored to the desired application.

Of the seven types of additive manufacturing some lend themselves to printing multiple materials by using multiple nozzles[8]–[10], or nozzles specially designed for mixing feed materials [11], to selectively deposit material with desired properties on the part at desired locations. This includes material extrusion[8]–[11], material jetting[12], [13], powder bed fusion[2], [14], [15], and directed energy deposition for metals[2], [16]. Another method, vat photopolymerization, proves more challenging in terms of multi-material printing because there is a risk for cross contamination of materials, slowing down the process in the cases where it has been adapted to allow for switching between materials[2], [7], [17]. Sheet lamination and binder jetting are limited in the functionality of the final part due to compatible material limitations [2], [7].

In contrast to these discrete source material printing methods, another approach to achieving multi-material properties in a monolithic print has been to process composite materials as they are printed in order to alter the material's internal microarchitecture and thereby control bulk properties at the point of deposition [18]. Additive manufacturing methods such as material extrusion and material jetting, which have been popular methods for depositing composite materials that contain fiber or particle fillers[2], are amenable to such composite processing. For example, using these composite material approaches, curable magnetorheological fluids have been printed by Kokkinis et al. in the presence of a magnetic field to locally alter the material properties of the source material, achieving distinct changes in material properties with the same source material through changes in processing alone [18]. The ability to use a single set of constituents while still achieving multi-material properties within a print would be a clear advance in the additive manufacturing field.

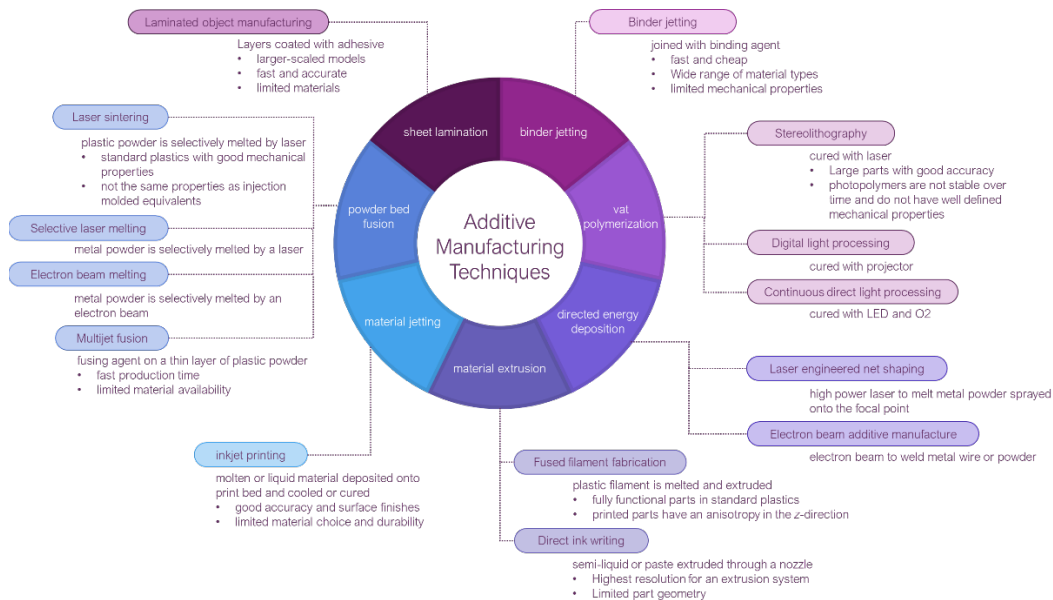


Figure 1-1: Categories of additive manufacturing as defined by ASTM52900:201 [1] with pros and cons of each method [2]

Consequently, this thesis proposes the development and validation of an optimization framework that utilizes the science underlying electromagnetic field processing of polymer matrix composites to investigate how a range of processing conditions applied to a fixed set of constituents affect the microarchitectures formed, and ultimately the material properties of the resulting composite. Electromagnetic field manipulation is chosen given its proven ability to direct particle responses [19]–[22], discussed further in Section 1.2.2. Moreover, the framework seeks to determine the widest range of resulting properties that can be achieved, for a particular particle-matrix material pairing, by varying this processing alone. Determining the set of constituents, which through processing alone yields the widest range of properties across thermal, mechanical, dielectric, etc., would prove an expensive, time consuming endeavor. Historically, effective material property prediction methods have arisen to fill this need, providing estimates of material response via knowledge of the constituents' properties and of the constituents' composite microarchitecture. This is done either via traditional effective medium theory or via homogenization techniques. Effective medium theory started by looking at various filler shapes and orientations for dilute systems and was later expanded to include some particle-particle interactions but was limited in its ability to consider long range order. Homogenization took on this challenge by including distribution functions for particle shape and orientation in an RVE for shape effects, the interactions of particles within an RVE, for short range effects, and consideration for periodic RVE's for interactions on a long range scale [23]. These models consider specific simplified shapes and are limited by the ability to describe the distribution analytically. Computational homogenization allows for shape diversity and any distribution of filler material as well as the ability to integrate processing parameters independent variables to establish the relationship to processing dependent effective properties [24]–[26].

The methods developed can form the scientific basis for a universal printer by determining the processing conditions, for a given constituent set, that can provide the widest range of properties by varying processing alone. This can be accomplished using computational homogenization in conjunction with an optimization scheme such as gradient based optimization which is an efficient algorithm in systems with low degrees of non-linearity such as this. Additionally, Artificial Intelligence and Machine Learning play an integral role in the efficiency and reducing computational cost of searching this design space. A framework that links processing conditions and composite constituents to effective properties is critical in materials design research.

## **1.2 Literature Survey**

### 1.2.1 Motivation: Multi-Material Additive Manufacturing

Multi-material printing has been investigated and realized in two main ways: altering the source materials or altering the material processing environment [2], [18], [27], [28]. Commercially available printers with multiple material wells or reels are commonly used to switch materials during printing[8]–[10]. Alternatively processing conditions can be changed to alter the properties of the material itself, such as by changing the way the layers are built up [27] or by organizing filler material within composites[18], [28]. Both techniques can achieve functional prints with variable material properties but vary in process complexity.

Different additive manufacturing techniques require different approaches to printing with multiple materials. Vat polymerization, for example, is potentially least suited to the task but has been adapted to allow for material changes during printing. The risk when switching between material wells is cross contamination of base materials and therefor slow manufacturing speeds when changing between materials during a print [2], [7]. Material extrusion methods are



frequently used to print with multiple materials with several commercially available multi-material printers [8]–[10]. Custom printers have also been developed to combine multiple printing techniques to expand on multi-material printing options [29]. Given these challenges, adaptations to allow for localized material property control have been made to several printing methods, presented here, to improve the functionality possible with AM parts.

In addition to printing monolithic parts containing segregated regions with different materials, vat photopolymerization has been used to fabricate traditional composite materials, a popular approach to improve on the mechanical properties of materials using well distributed inclusions. Vat photopolymerization offers better resolution in prints as compared to extrusion based methods that are limited by nozzle size and can be an excellent choice for precision prints and incorporating filler materials. Sano et al. have used Stereolithography (SLA), otherwise known as vat photopolymerization, to print fiber reinforced polymer composites to improve the tensile strength and modulus of the base polymer [4]. A Nobel 1.0 SLA printer was used with UV-cure epoxy resin containing glass powder ranging from 5-50 wt% or fibers added at 1wt% to demonstrate the improvements possible with composite materials in SLA printing. The distribution of the glass fibers was shown to be poor, with the majority of the fibers in the first layer but showed 7.2 and 11.5 times the strength and modulus, respectively. The glass powder was uniformly distributed, an advantage in using powder over fibers. However, the increase in strength and modulus at 50wt% were 2.1 and 6.4 times, respectively, over the resin, 2.6 times less than the fiber composite for higher filler content. The shape and distribution of the filler clearly play an important role in the resulting mechanical properties and print quality of such a composite. Alternatively, adding multiple feed materials requires additional modification including the use of multiple vats of resin. When moving a print to a new vat of feed material, it

needs to be cleaned to remove uncured remnants of the previous material [7]. This cleaning needs to happen each time the print is transferred between the various base material vats, a time consuming process. Zhou et al. have developed a method to reduce the surface area exposed to the resin that needs to be cleaned off before switching basins to improve on transition time to address this concern. Even this technique though still requires cleaning between switches that remains the bottleneck [30].

Material extrusion techniques avoid the concerns for cross contamination of vat photopolymerization material wells by selectively depositing the material onto the print platform, while still being able to be used to print composites [2]. Multiple materials can easily be extruded through separate nozzles to build up a multi-material print. Khondoker et al. [11] proposed a bi-extruder capable of extruding two materials through the same nozzle with a static inter-mixer, also helping with bonding of the two materials (see Figure 1-2).

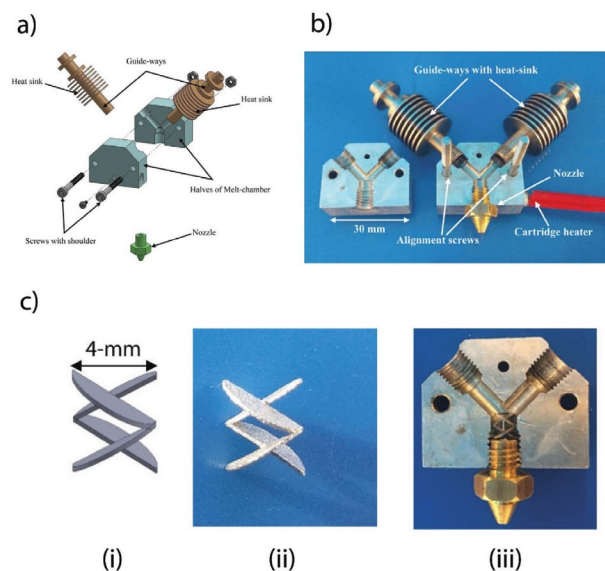


Figure 1-2: Bi-extruder for FDM multi-material additive manufacturing a) exploded view of the bi-extruder, b) full assembly of the custom bi-extruder, c-i) CAD model of the inter-mixer, c-ii) fabricated inter-mixer, c-iii) inter-mixer assembled into bi-extruder channel [2], [11]

Nassar et al. incorporated flexible and conductive material into a print by adapting a commercial printer to allow extrusion of a conductive paste during printing to create flexible electronic components [31]. Multifunctional prints can also be achieved by using filler materials to make composite prints. Gandha et al. [3] and Li et al. [32] used magnetic composite pellets to print bonded magnets that were post processed to align filler material for improve magnetic properties. Using similar techniques, multi-material prints can be achieved with powder bed fusion by selectively placing and fusing powder [15].

Printers combining multiple additive techniques have been developed such as the m4 printer built by Roach et al., which combines fused filament fabrication (FFF), direct ink writing (DIW), ink jetting, and aerosol jetting in addition to robotic pick and place functionality. The development of a system such as the m4 printer hopes to drive more efficient manufacture, such as a soft actuator that was printed in just under 3% of the time it would have with taken with traditional techniques, such as embedded casting of lithography, for such a component. This printer also enables the addition of integrated components and combined techniques such as the inclusion of electronic leads in a flexible print. The use of multiple materials requires consideration in the design for material adhesion, thermal expansion coefficient differences, and interface geometry, all of which can affect the quality and effective material properties of the print. [29]

Functionally graded materials offer additional control over material properties by using single or multiple materials, placed strategically, to achieve a spectrum of tailored properties [33]. With this technique, structure is used to alter effective material properties via voxel-based models [34] or function-based models [35] to define the material distribution throughout the design. Recall that while multiple materials are most often used for multifunctionality, that

multifunctionality is predicted only on multi-material properties, properties which can be achieved manipulating the process-structure-property relationships of single composition materials instead.

Another method to achieve multi-material response involves, tuning properties of single-constituent platform during printing in composite materials by altering the internal structure (or microarchitecture) of the filler material. The effects of electric and magnetic fields, detailed further in Section 1.2, applied to particle suspensions have been studied extensively, showing particle ordering and structuring [19], [22], [36]–[38].

Kokkinis et al. made use of this control over local properties when printing with composite ink. Magnetized platelets suspended in a light-sensitive resin were aligned with a permanent magnet during printing. Curing the resin with the presence of the magnetic field locks in the microarchitecture, shown in Figure 1-3. Tensile tests showed that the alignment of particles parallel and perpendicular to the testing direction increased the elastic modulus by 52% for the parallel case as compared to the perpendicular case, highlighting the anisotropy and therefore range of properties possible from magnetic alignment alone [18].

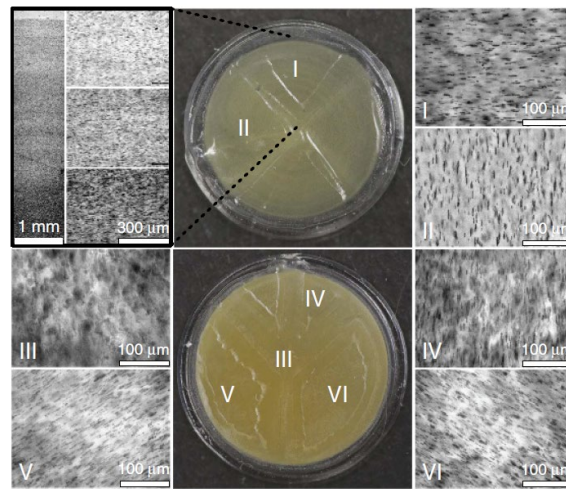


Figure 1-3: Gradient in platelet concentration and alignment as a result of varied volume fraction mixes and applied magnetic field during printing of the top (I-II) and bottom (III-VI) layer of the structure [18]

Multi-field processing , using electric and magnetic fields synergistically, of bulk, filled matrix materials can reduce the complexity of machine setup compared to multi-material prints by simplifying the constituent materials while possibly expanding the range of properties with continuous property control via directed assembly of microarchitecture. Previous works have investigated the structures formed in magnetic and electric fields and have experimentally classified the change in composite property as a result of these microarchitectures [21], [39]–[43]. However, the relationship between material property and applied field(s) is yet to be defined. The methods developed in this proposal can form the scientific basis for a multi-material printer by determining the processing conditions, for a given constituent set, that can provide the widest range of properties by varying processing alone.

### 1.2.2 Processing Magneto Active Polymers to Control the Microarchitecture Within

Material systems consisting of filler material embedded in a matrix can be structurally modified to change several bulk properties, including those in the mechanical, magnetic, and electric domains [44]–[47]. Applying external fields causes particles to form microarchitectures which affect the bulk properties of the composite leading to an anisotropic response. This type of manufacturing will use vat photopolymerization due to the need for particle mobility. The processing and resulting properties of these composites have been extensively studied and characterized, as discussed below.

It is known that application of a static magnetic field on a ferrofluid, which is a colloidal suspension of ferromagnetic particles in a carrier fluid, will result in a change in the internal

structure. On the macro-scale, these particle-particle interactions manifest in observable surface instabilities [48] (see Figure 1-4) or a change in viscosity [49], [50], which forms the basis for magnetorheological fluids

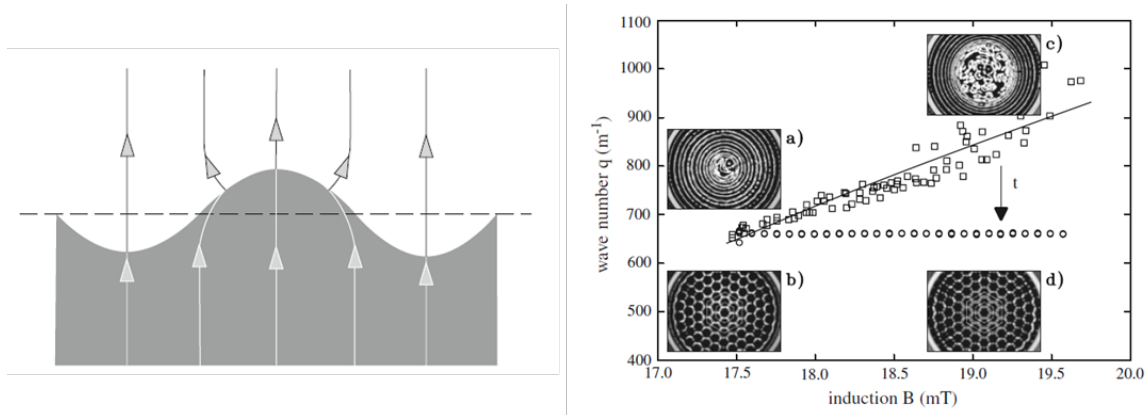


Figure 1-4: Magnetic surface instabilities a) showing field lines that are compressed and uncompressed due to the peaks and valleys in the surface that minimize the magnetic energy as compared to a planar surface b) experimental instability shown for field versus wave number [48]

Studies have also reported that particle polymer colloids can be manipulated to form chains with electric or magnetic fields [19], [22], [36], [38]. Bharti et al. show the effect of simultaneous electric and magnetic fields on superparamagnetic particles in a polystyrene matrix shown in Figure 1-5, where it can be clearly seen that the direction and intensity of the fields with respect to each other directly affect the microarchitectures formed [19]. Furthermore, recent studies have reported that a combination of magnetic and electric fields during the curing process of composites, arranging particles in multiple dimensions, creates a hierarchy of structure that can contain smaller-scale independently oriented structures [37]. Freezing microarchitectures such as these into place by curing the matrix around the particles can create an anisotropic composite whose properties are determined by the constituents and their spatial and orientational distributions. Moreover, this organization, most clearly seen in macroscopic chaining and surface

instabilities, extends to smaller length scales creating electromagnetically driven hierarchical structures.

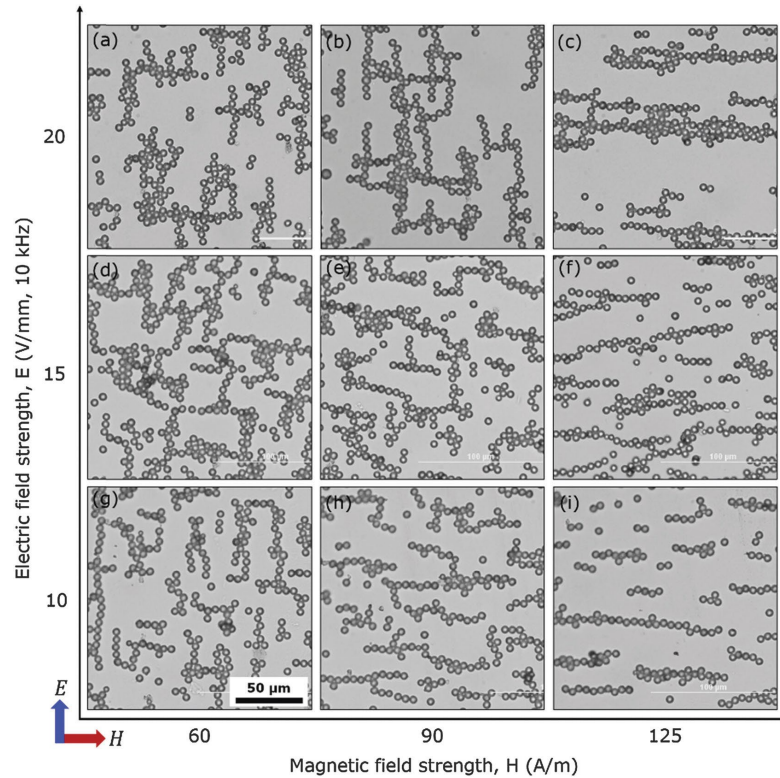


Figure 1-5: Bright field optical micrographs of superparamagnetic microspheres, effects of simultaneous electric versus magnetic field of varying strengths, demonstrating the chaining is determined by the direction and relative intensities of the applied fields. [19]

Recent experimental work has illustrated the ability of electromagnetic fields to drive hierarchical particle self-organization using coupled electric and magnetic fields. An example with parallel electric and magnetic fields is shown in Figure 1-6 a-b), where the platelets are stacked through the magnetic easy axis and stacks align to form longer chains in line with the electric field. This is in contrast to Figure 1-6 c-d) where the stacks of particles are aligned in a perpendicular manner to form chains in the direction of the electric field [20].

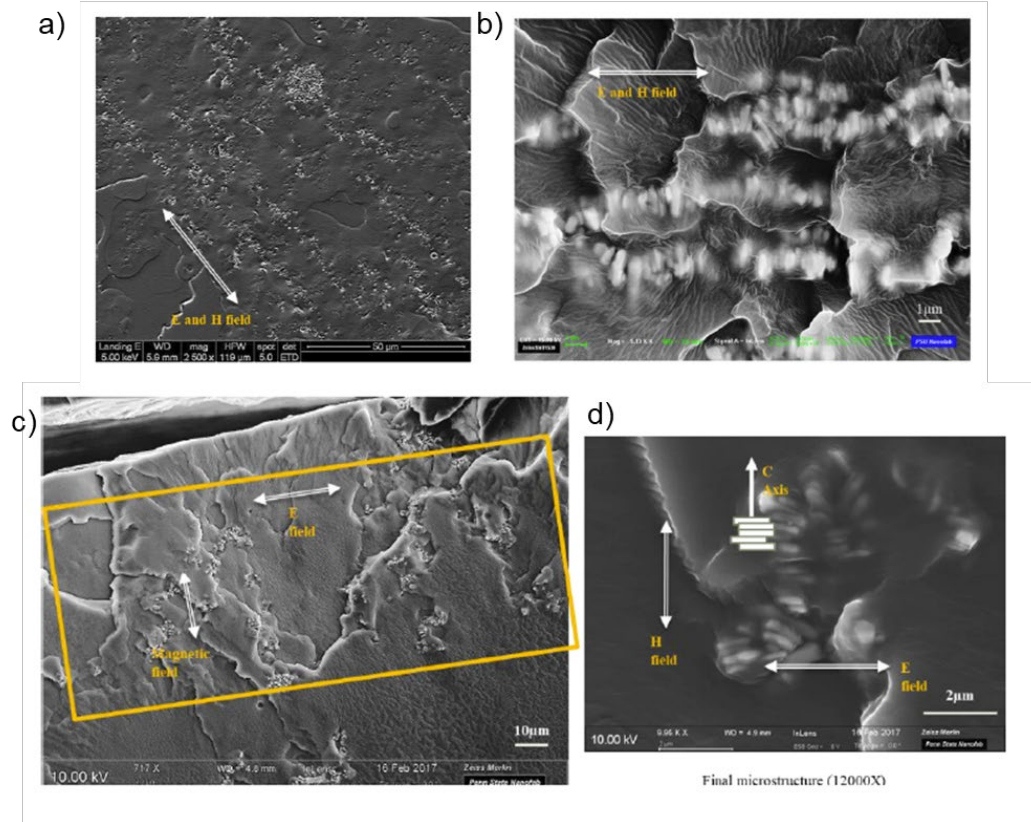


Figure 1-6: a-b) SEM image showing particle chain and Barium hexaferrite (BHF) stacks inside the chains, respectively, for parallel applied fields and c-d) SEM image showing part of a chain and BHF stacks inside the chains, respectively, for perpendicular applied fields adapted from [20]

The microarchitectures formed without a field, with a 0.5T magnetic field, and with a 0.5T magnetic and 4kV/mm electric field were studied experimentally by Masud et al. [37] and are shown in Figure 1-7. Barium hexaferrite (BHF) particles, which are geometrically anisotropic platelets with a magnetic easy axis (and non-zero magnetic remanence) through the thickness, were suspended in heat curable polydimethylsiloxane (PDMS) at 5wt%. The magnetic field causes platelets to orient on their magnetic easy axis [51]. Past works such as by Bharti et al. [19], Bharti and Veleev [38], and Masud et al. [37] have shown application of an electric field to a colloidal suspension to cause filler particles to form chains. An electric field applied in addition to a magnetic field was shown to produce longer chains than the magnetic field alone [37].



Figure 1-8 shows the simulation results and the correlation to experimental findings, confirming the longer chains in the combined field case. Simulations account for dielectrophoresis (DEP), repulsive, hydrodynamic, and magnetic dipole interaction forces [37]. These simulations lay the groundwork for accounting for complexities in microarchitecture needed for assessing composite material properties that traditional effective medium, described further in Section 1.3, does not. The increased thickness of the chains in the experimental results of the magnetic only case (Figure 1-7 b) is not replicated the same way in the models in Figure 1-8 a. However, in Figure 1-8 a the magnetic field only simulations show the shorter chains that are seen experimentally as well in Figure 1-7 b and longer chains in the Figure 1-8 b from the combined applied electric and magnetic fields observed also in Figure 1-7 c experimentally.

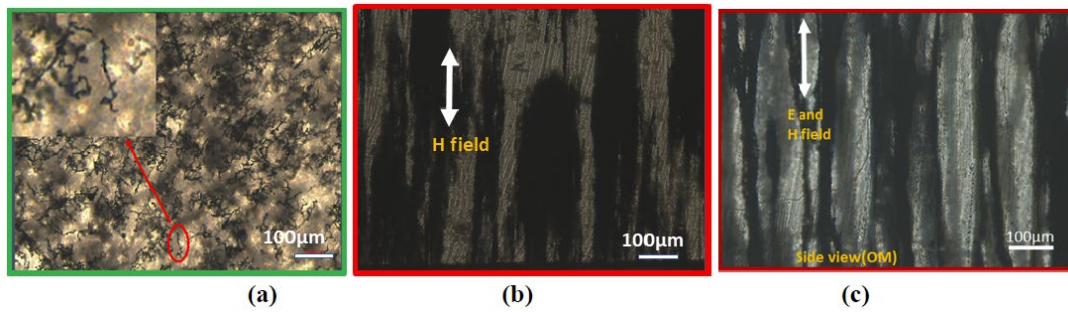


Figure 1-7: Microarchitecture of Barium Hexaferrite in PDMS (a) before the application of any field, (b) after the application of a magnetic field, and (c) after the application of both electric and magnetic fields showing the increased ordering present with one and both applied fields, respectively [37]

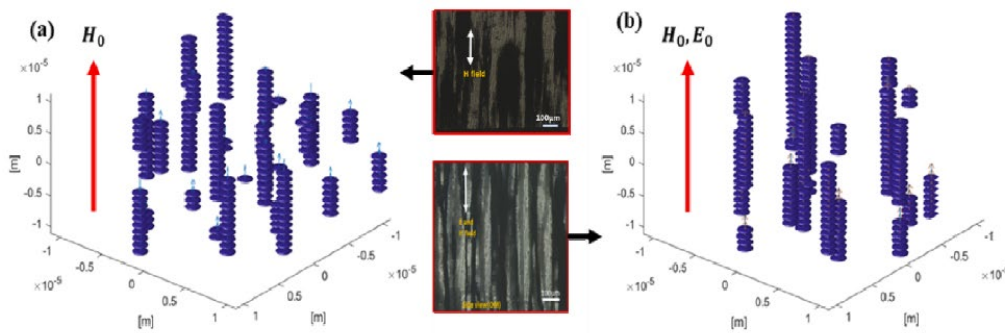


Figure 1-8: Simulation result for model with 200 particles a) magnetic field only and b) magnetic and electric field, showing the improved chaining for the combined field case [37]

Several studies have characterized the anisotropy and properties of magnetic active polymer composites [21], [39]–[42]. Select experimental studies, such as done by Varga et al. [41], show the relationship between elastomeric filler volume fraction and composite properties when processed in an externally applied field. The moduli parallel and perpendicular to the alignment direction were measure across 10, 20, and 30wt% spherical carbonyl iron in PDMS matrix aligned in a 400mT magnetic field. The moduli were measure in and out of plane with the alignment which quantified the anisotropy in aligned composites. Perpendicular to the chains the modulus did not increase outside of the range of experimental error, however, parallel to the chains at 30 wt% iron content the modulus increased by 250% [41]. Despite these promising results of the effects of volume fraction on modulus, the effect of varying the applied fields, and the resulting microarchitectures, on the composite's properties is yet to be defined quantitatively.

Creation of the self-organized structures that lead to changes in properties stems from particles in a colloidal suspension responding to both DC and AC fields in the form of electrophoresis and dielectrophoresis, respectively. Electrophoresis caused by an applied DC field will move charged particles toward the oppositely charged electrode[22], [36]. Applying an AC field results in particle motion if the field is nonuniform, this is known as dielectrophoresis. The induced dipole and field gradient give rise to a dielectrophoretic (DEP) force which is defined as

$$\overrightarrow{F_{DEP}} = 2\pi\epsilon_1 Re|K(\omega)|r^3\nabla E_{rms}^2 \quad (1-1)$$

where  $\epsilon_1$  is the permittivity of the particle,  $r$  is the radius of the particle,  $\nabla E_{rms}^2$  is the gradient of the field squared, and  $K$  is the Clausius-Mossotti function. The Clausius-Mossotti function is

important to understanding self-organization with respect to AC fields because it may cause repulsion or attraction. The function itself is given by

$$Re|K(\omega)| = \frac{\varepsilon_2 - \varepsilon_1}{\varepsilon_2 + 2\varepsilon_1} + \frac{3(\varepsilon_1\sigma_2 - \varepsilon_2\sigma_1)}{\tau_{MW}(\sigma_2 + 2\sigma_1)^2(1 + \omega^2\tau_{MW}^2)} \quad (1-2)$$

$$\tau_{MW} = \frac{\varepsilon_2 + 2\varepsilon_1}{\sigma_2 + 2\sigma_1} \quad (1-3)$$

where  $\varepsilon_1$  is the permittivity of the matrix and  $\sigma_1$  and  $\sigma_2$  are the conductivities of the particle and matrix, respectively. As shown in equation (1-2), the strength of the interaction is governed by the Clausius-Mossotti factor which will vary with frequency. Results of Masud et. al. also highlight the importance of frequency response as seen in Figure 1-9, where field strength improves chaining at low frequencies. Chaining ceased at high frequencies [20].

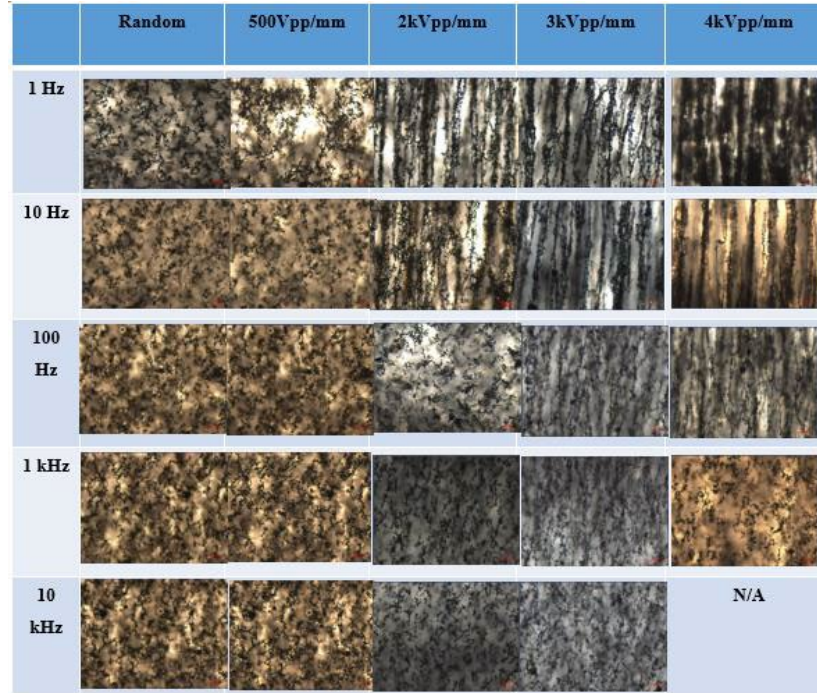


Figure 1-9: OM/SEM of frequency versus. peak-to-peak voltage showing difference structures as a function of field strength and frequency where increases in electric field improves the chaining response at low frequency, and for higher frequency chaining does not occur [20]

Dipoles induced in particles in a uniform field will interact with each other and lead to structuring in the form of chains along field lines. The chaining force is defined as

$$F_{\text{chain}} = -C\pi\epsilon_1 r^2 K^2 E^2 \quad (1-4)$$

where the coefficient  $C$  depends on the distance between the particles, ranging from 3 to greater than  $10^3$  [22], [36]. The benefit of using an AC field is the potential for precise motion control via manipulation of parameters such as field strength and frequency as well as avoiding any water electrolysis [36]. Particles of the same type always align along field lines regardless of the media of the colloid [22].

Similarly for magnetic fields, the force on a magnetic particle with magnetization  $M$  in an applied field  $\mathbf{H}$  is defined as

$$\mathbf{F}_{\text{mag}} = \nabla E_H = -\mu_0 M V (\mathbf{H} \cdot \mathbf{d}) \quad (1-5)$$

where  $\mathbf{d}$  dipole of the particle,  $V$  is the particle volume,  $\mu_0$  is the permeability of free space, and  $E_H$  is the magnetic potential energy due to the applied magnetic field [52]. The magnetic field experienced by any particle is the sum of the external field and the field generated at the location of the particle by the surrounding particles.

Simultaneously, external magnetic and AC electric fields acting on particles can be tuned to control the structures and networks formed. Particles can be coerced into forming directional chains or 2D networks with various field configurations. Figure 1-10 shows an example of the networks that can be formed in a combined field case for spherical particles [19]. The various inputs to a system including frequency of an applied field, dielectric properties of the particles

and the medium, particle size, shape, and volume fraction can be used to tune structures and control bulk material properties [38].

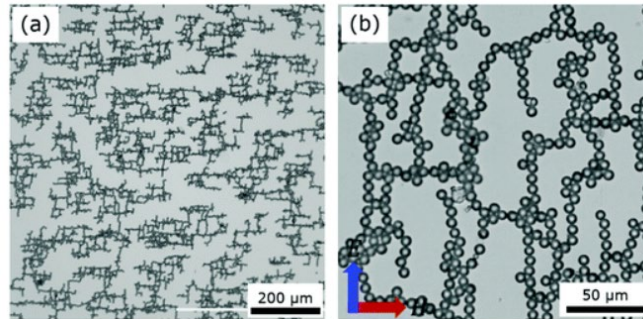


Figure 1-10: Microscope image at low (a) and high (b) magnification of particle network formed in the presence of perpendicular electric and magnetic fields [19]

As has been shown, the application of an external field and characterization of material properties has been studied in the literature. The effect of some input variable like filler volume fraction have also been investigated. The relationship between varying fields and effective composite properties is understood to exist and important to materials design [53], but no quantitative path from processing parameters and constituents to material properties has been established. This crucial step will enable selection of materials and processing setup when manufacturing designs using multi-field processed particle composites. In an AM context, this will allow future part designers to assign desired spatially variable material properties (within a possible range) alongside part geometry, ahead of fabrication on a multi-field processing print platform and provide critical machine parameters for manufacturing of prints.

### 1.2.3 Estimating Composite Material Properties

Estimating properties is important for composite manufacture. Several approaches exist for predicting effective properties based on constituent properties. These approaches can be categorized into effective medium approaches that provide a closed form solution and

homogenization which considers particle distributions with both analytical and computational methods. This section will present traditional effective medium theory and contrast it to homogenization techniques, then close by presenting an AI approach for an additional means of predicting composite properties.

### 1.2.3.1 Effective Medium Theories (EMT)

Effective medium theory and micromechanics models are powerful tools in estimating the properties of various composites from their constituents. Several models, including Mori-Tanaka and Halpin Tsai, take filler shape and orientation into account but do not include effects of long range ordering such as stack or chains formed by the application of external fields. The microarchitecture formed by the applied fields therefore cannot be accounted for with these models. Other analytical homogenization techniques, as detailed below, have considerations for orientation and distribution, but still do not account for stacks and chains in these distributions [23]. Precisely controlling microarchitecture is a crucial component for manipulating the properties of multi-field processed composite materials. The following section lays out basic effective medium theories, highlighting their shortcomings for this project, while concluding with a discussion of computational homogenization and how it can address the required hierarchical ordering.

### Elastic Response

Effective medium theory, often derived from micromechanics models, is a well-accepted method of determining bulk material properties of many types of uniformly distributed composites. Basic models such as the Reuss and Voigt models can be useful for determining bounds on properties. In these approximation simplifying assumptions are made about the

stresses and strains, respectively, to determine the effective modulus,  $E^*$ . In the Reuss model uniform stresses are assumed throughout the volume which results in

$$\frac{1}{E^*} = \frac{v^{(f)}}{E^{(f)}} + \frac{v^{(m)}}{E^{(m)}} \quad (1-6)$$

where  $f$  represents the filler phase,  $m$  represents the matrix phase,  $v$  is the respective volume fraction, and  $E$  is the Young's Modulus of the phase. The Voigt model, in contrast, assumes uniform strain throughout the volume which results in

$$E^* = v^{(f)}E^{(f)} + v^{(m)}E^{(m)} \quad (1-7)$$

While the two equations form the upper and lower bounds on the composite's effective modulus as illustrated in Figure 1-11, neither account for relative spatial ordering of the filler, shape of the filler, or relative orientation of the filler.

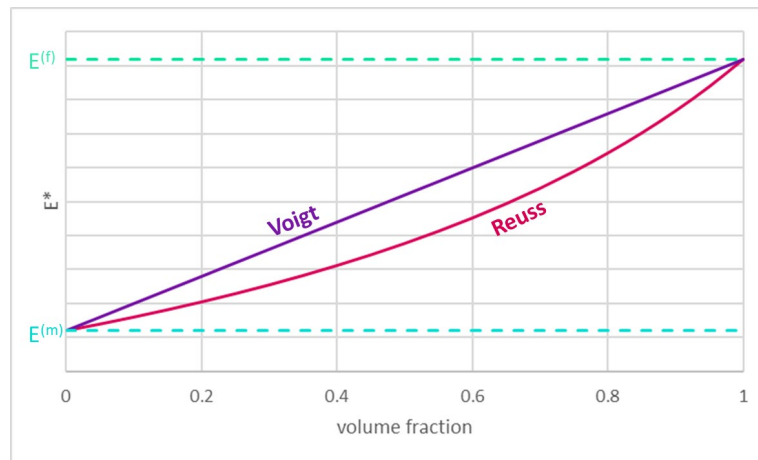


Figure 1-11: Voigt and Reuss model setting upper and lower bounds, respectively, for effective composite properties

Popular higher fidelity methods such as the Mori-Tanaka [23], [54] and Halpin-Tsai [55]–[57] models relate the volume fraction, constituent properties, and filler shape to the

effective composite properties. However they make simplifying assumptions or experimentally defined correlations that do not allow them to compensate for microarchitecture. In one critical assumption, the Mori-Tanaka model, similar to the Eshelby [58] model it is based on, assumes the inhomogeneities are isolated enough to not account for effects of neighboring inclusions, the so-called dilute approximation. This approximation is shared by several prominent models including Eshelby, Mori-Tanaka, and Halpin-Tsai. The Mori-Tanaka model accounts for filler strain separately from matrix strain when determining the average strain on the volume, while the Eshelby assumes the average strain is the strain on the matrix [59]. The Halpin-Tsai model, like both Mori-Tanaka and Eshelby, defines a representative volume looking at a single inclusion in a volume of matrix but has a reinforcement term that can be determined from tabulated equations or experimentally correlated. The reinforcement factor, on its extremum, returns the upper and lower bounds of the Voigt and Reuss model [55], [60]. These models are laid out in further detail in Section 2.2.1.

While these methods account for complex effects of the filler properties, orientation, and/shape on the mechanics, each make simplifying assumptions or experimentally defined correlations that do not allow them to compensate for other aspect of microarchitecture, e.g. relative spatial orientation or ordering of the filler. In one critical assumption, the Mori-Tanaka model, similar to the Eshelby model it is based on, assumes the inhomogeneities are isolated enough to not account for effects of neighboring inclusions, the so-called dilute approximation. This approximation is shared by several prominent models including Eshelby, Mori-Tanaka, and Halpin-Tsai. Given that difference in physical long and short range ordering are expected, these models will not be sufficient to accurately describes the expected microarchitectures.

### Dielectric Response



Similarly, models have been established to predict dielectric properties based on constituent properties. Approximations for relative permittivity have been developed for various filler shapes and orientations. The mean field approximation proposed by Maxwell Garnett [61] served as a basis for low volume fraction approximations, which was further developed for various shapes including spheres by Wagner [62], oriented ellipsoids by Sillars [63], and randomly oriented ellipsoids by Fricke [64], [65]. Bruggeman lays out early models by Maxwell Garnett, Lorentz-Lorenz, Raleigh, and Wiener in addition to presenting his own extension of these models that all assume random spatial distribution but do account for filler shape and orientation [66].

Bruggeman laid out 4 criteria for his models containing fillers of any shape:

- The composite is comprised of 2 homogenous isotropic constituents
- All aggregates are large compared to the atoms or molecules they are made up of
- The aggregates are randomly distributed
- The different constituents are of the same structure (no interspersions)

His model served as a basis for further development [67] by Böttcher [68], Polder and Van Santen [69], and Hsu [70] for spheres, randomly oriented ellipsoids, and oriented ellipsoids, respectively, and could be used to approximate higher volume fraction values by assuming the inclusions were surrounded by an effective medium. The details of these models are presented in Section 2.2.2.

Similar to the elastic models, these effective medium approaches to dielectric homogenization do not have means to account for microarchitecture and are limited to randomly distributed fillers.

### 1.2.3.2 Homogenization

Homogenization aims to address the limitations on effective medium theory by incorporating effects of particle distributions. Approaches to homogenization are either analytical or computational, in both cases a representative volume is used that is prescribed specific boundary conditions. These boundary conditions can either be specific (e.g. stresses or strains for the elastic domain), which necessitate a large enough volume to average the properties, or periodic, which effectively model a continuous solid.

#### Analytic Homogenization

Analytic homogenization techniques use particle distribution functions to account for non-uniform distributions. This necessarily requires the distribution function to be represented by an expression.

Building off of traditional effective medium theory, Ponte-Castaneda expanded the Hashin-Shtrikman model to allow for separation of the spatial and shape distributions of the inclusions, which allows for the evaluation of non-uniformly distributed composites. This is done to estimate the overall moduli of composites, consisting of a matrix with one or more types of inclusions [71]. The separation of these two factors, the inclusion type and the distribution of inclusions, is done by defining a probability density function that defines the probability of an inclusion being centered at a point given the location of another inclusion.

$$\mathbf{L}_c = \mathbf{L}_0 \{ \mathbf{I} - c_1 [\langle \mathbf{A}_1 \rangle^{-1} + c_1 \mathbf{S}^V]^{-1} \} \quad (1-8)$$

where  $\mathbf{L}_c$  is the modulus tensor of the composite,  $c_1$  is the volume fraction of inclusions,  $\mathbf{L}_0$  is the modulus tensor of the matrix,  $\mathbf{L}_1$  is the modulus tensor of the inclusions,  $\mathbf{S}^V$  is the Eshelby tensor

(defined in detail for an ellipsoidal inclusion in Section 2.2.1) for the double cell (see Figure 1-12), and  $A_1$  is the eigenstrain concentration tensor in the inclusion defined as

$$A_1 = -[S^\Omega - (L_0 - L_1)^{-1}L_0]^{-1} \quad (1-9)$$

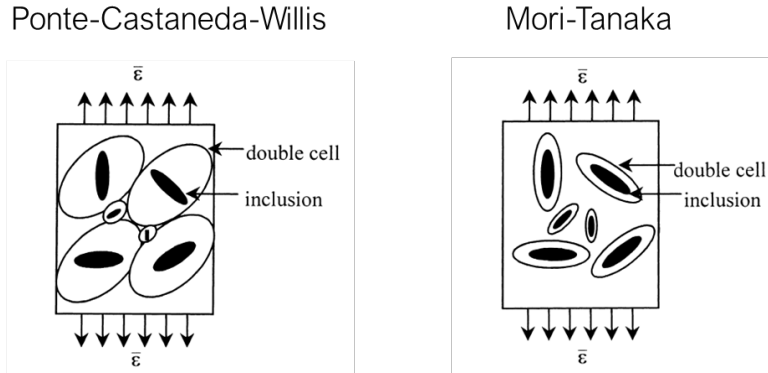


Figure 1-12: Double cell model for the Ponte-Castaneda model adapted from [23]

where  $S^\Omega$  is the Eshelby tensor for the inclusion. It can be seen in Figure 1-12 that the Mori-Tanaka model is a special case of the double inclusion model, where the inclusion and the double cell have the same shape and orientation, whereas for the Ponte Castaneda-Willis (PCW) model the double inclusion model is modified to fix the orientation of the double cell, but the distribution of the inclusions is not. The effective modulus tensor of the composite is therefore dependent on the characterization of the shape and on the distribution function of the filler [23]. In order to make use of this expanded model the distributions must be known. Computational approaches by contrast do not require an expression for the distribution function, but a modeled geometry instead, which is itself the realization of the distribution function.

### Computational Homogenization

Analytical homogenization finds the solution of constraint equations on the representative volume which ensures strain compatibility. Computational homogenization solves these equations *via* discretization of the volume without the need to find a closed form representation

of the filler distribution. Using finite element analysis (FEA) with a computer generated geometry allows for arbitrary shapes, size, spatial and orientational distributions, and material properties.

Geometries can be generated to distribute filler randomly [72]–[75], in an ordered manner [75], [76], as a result of a particle dynamics simulation [19], [26], [77], or experimental reconstruction [78]. Within the physics based methods there are several packing algorithms that consider external forces such as gravity, particle interactions/collisions, and matrix properties such as viscosity. These techniques have been used to model fiber reinforced composites, densification, or even granules in a flow [78]. For analytical homogenization, using particle dynamics simulations to determine the filler distributions as a result of applied electric and magnetic fields, it would be difficult to represent the results in distribution functions required. However, since the particle dynamics simulations return a specific geometry, effectively sampling the distribution, several iterations are required to determine a mean and standard deviation on the resulting material properties.

Once a model is realized, it can be used for homogenization, which is a mathematical approach of estimating the response of a representative volume element (RVE) to prescribed loading conditions on its boundaries which themselves are considered periodic. The solution of the resulting boundary value problem, with respect to its loadings, take the form differential equations governing the response of physics-based conjugate pairs (e.g. stress and strain; electric field and electric displacement; temperature and heat flux, etc.). Applying one side of the pair as a boundary condition and enforcing the appropriate physics on the contents of the RVE, allows solution of the problem with respect to the conjugate, thereby providing the constitutive

relationship between the two. Consequently, homogenization via finite element analysis requires proper selection of boundary conditions.

In the mechanical domain discrete boundary conditions assume the volume to be fixed in space and anchor it so that loads can be applied whereas periodic boundary conditions assume a continuous medium and translate the resulting stresses or strains from one face to another as if the element were repeated on all sides as shown in Figure 1-13.

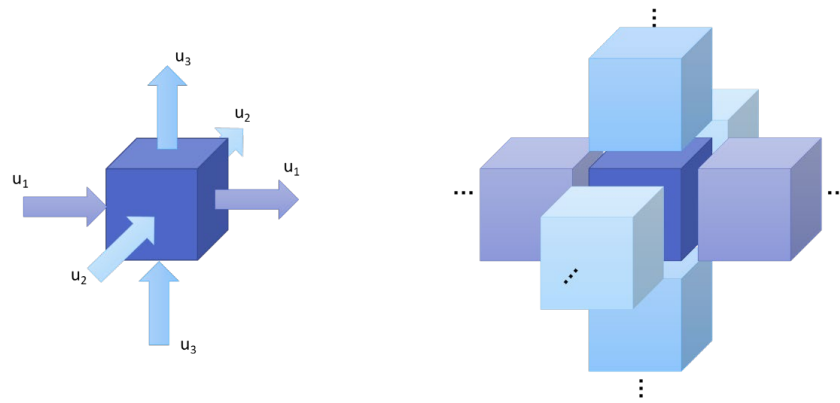


Figure 1-13: Periodic element visualization showing repeating units to represent translation of boundary conditions to paired faces

It has been shown that the periodic boundary conditions yield more accurate results and converge faster [24]. The size of the RVE used is a tradeoff between accuracy and computational cost. Larger models require more resources to converge to a solution but yield more accurate results. Pahlavanpour et al. [72] performed a convergence study on the effective modulus to determine the minimum size of RVE needed. For large discrepancies in constituent moduli larger RVE's were required for convergence. In [72] it was shown that the minimum RVE size needed to be 5% larger when the difference between particle and matrix modulus is increased from approximately 9.5 to 13 times greater. Similar considerations for boundary conditions and RVE size need to be made for other domains such as magnetic and dielectric [79].

Kanit et al. discuss three types of boundary conditions and their effect on the effective moduli of the composite. The volume can be subject to kinematically uniform (imposed displacement), statically uniform (prescribed traction), or periodic boundary conditions. When the volume is small, in this work around 10 particles, the kinematic and statically uniform boundary conditions logically tended toward the Voigt and Reuss bounds which make similar assumptions about stress and strain. The periodic boundary condition showed for sufficiently large volume there is little fluctuation in the computed effective modulus, which is in contrast to the other two methods that were still converging at 100 times larger model sizes [24].

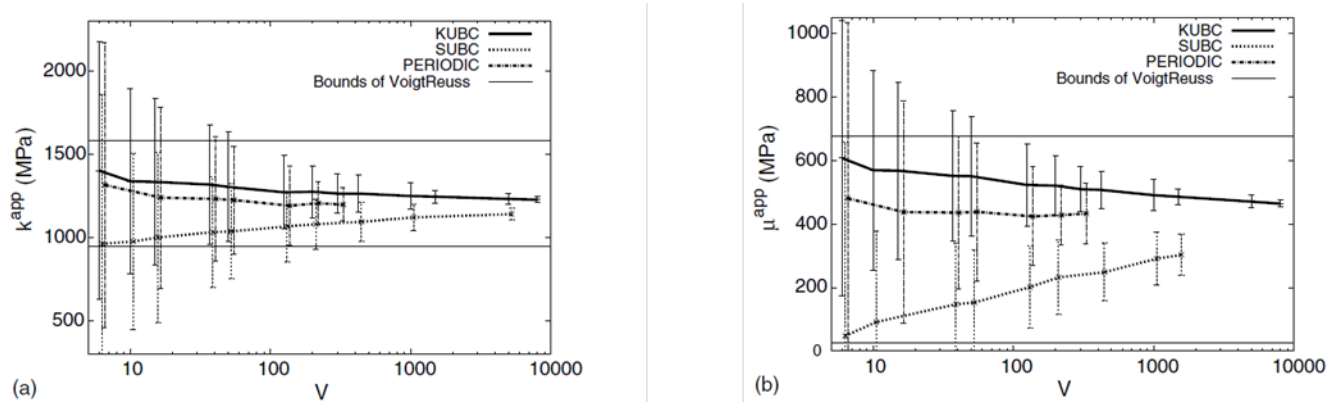


Figure 1-14: Bulk and shear modulus, respectively, versus model size comparing kinematically uniform, statically uniform, and periodic boundary conditions to effective medium bound (Voigt and Reuss) [24]

Several factors contribute to the minimum RVE size needed for accurate results, including particle size, shape, distribution, and the difference in material properties between the particles and the matrix. Several studies have looked at these factors to determine the relationship between effective material properties and simulation size [24], [80]. Moumen et al. found that for ellipsoidal particles the RVE needed to be larger than for spherical particles, also concluding that the models with more heterogeneities needed to be larger, for the same accuracy in effective composite properties [80].

Analytic homogenization expanded the capabilities of traditional effective medium by accounting for non-uniform distributions and long range order. However these methods require knowing the distribution of the particles. Computational homogenization generates composite geometries and solves the defined boundary value problems to determine effective material properties. This is a delicate balance of accuracy and computational expense and is affected by several factors related to the constituents and selected boundary conditions.

#### 1.2.3.3 Using Artificial Intelligence to Predict Conductivity

While computational homogenization methods provide a route to predictions of bulk properties from simulated high fidelity solid geometries, it may be possible to generate accurate predictions of some properties from lower fidelity descriptions of the predicted microarchitecture, e.g. from discrete particle location data alone using methods derived from AI and graph theory. If possible, this would constitute a less computationally expensive approach to computing effective properties. For example, magnetic remanence of hard magnetic materials can be estimated using a simple sum of their magnetization vectors normalized by the total volume in which they are contained [81]. While this does not account for particle-particle induced interactions, it can be used as an approximation where those either do not occur (dilute systems) or when the remanence is less affected by those interactions (high coercivity). Similarly, conductivity can be approximated by the response of an aggregate of discrete actors within appropriate contexts.

Near the percolation threshold conductive-filled composites transition from dielectric response to conductive. The phenomena has been studied in the context of conductive polymer composites (CPCs), which have been extensively studied with special emphasis on this percolation threshold. The percolation threshold is the volume fraction at which enough filler material is present to form conductive pathways through the composite [82]. It has been of

interest to minimize the filler needed to achieve percolation in transparent conductive films to remain transparent [83].

There are several approaches that have been traditionally used to determine the conductivity or percolation threshold of a composite. This includes Monte Carlo percolation threshold methods [79], [84], [85], random resistor networks [86], electrical network modeling [87], finite element modeling, and analytic approaches [88], [89] such as effective medium theory. These models, except finite element methods, typically use a random distribution of filler volumes and look for points of contact to form conductive pathways through a volume. In the case of electrical network modeling these pathways are represented in a matrix of ones and zeros to denote inclusions that are in direct contact. Analytic models have been developed for determining the percolation threshold as a function of filler geometry [89], [90], material properties, and filler orientations [84]. Although the effects of field oriented fillers are considered, the networks and microarchitectures formed are not accounted for in these models. Previous approaches have provided good correlation for networks of uniformly distributed filler composite's percolation thresholds and subsequent conductivity estimates.

The Monte Carlo random walk algorithm is a classic solution for finding conductive pathways and determining percolation [84], [85]. The method can be used to populate electric network matrices [79] or to simulate current flux. Rahatekar et al. used the random walks to move a virtual electron and average the distance it traveled in a given time. When a percolating path existed the electron would travel a large distance compared to those that got trapped in a nonconnected node [91].

Another approach used in previous works is constructing a minimum spanning tree such as done by Dussert et al. [92]. A spanning tree is defined as the minimum edge weights connecting all





Graph search is a category within AI that uses efficient search algorithms that do not require exploring the entire graph to find the shortest paths. However, including non-optimal paths, which would be required to represent a physics based network, does require exploring more of the state space (in the context of computer science, state space refers to the set of all possible configurations of a system, similar to a design space in engineering), and, as a result, is more computationally expensive. Sorting, sampling, and pruning techniques can be incorporated into such an algorithm [93] to improve run times and extend the size of the simulation while minimizing costs.

The graph is established from a discrete distribution of particle locations within an RVE from particle dynamics results. Conductive pathways are found using AI search algorithms, such as A\*, to determine the conductive network. The conductivity of the RVE can then be calculated by finding the effective current and using Ohm's Law with the applied voltage as a low cost method of material property prediction.

#### **1.2.4 Optimization via Traditional Gradient Based Techniques and Data Science Modeling**

At the crux of this work is the need to determine the set of input parameters, processing conditions and constituent set, that produce the widest range of material properties. The problem is located in a large design space, which, even when reduced to recreate experimental findings discussed in Section 3.3, still consists of both the strength and relative orientation of applied electric and magnetic fields (processing conditions) and the volume fraction of the barium hexaferrite filler used (constituent set).

In terms of optimization, we first wish to find both the maximum and minimum values for a property for a given constituent set and set of processing conditions. This dictates the widest property range possible for those conditions for a given property. Next, we wish to

determine, among all possible constituent sets, which set yields the widest possible range (again by varying processing alone). The optimization problem itself relies on a complex objective, the characterization of properties that result from constituents-processing combinations. Similarly complex optimization problems have been studied using both traditional gradient-based methods as well artificial intelligence techniques. The following section discusses several such techniques.

Multi-objective optimizations have been solved using various techniques and algorithms including the broad categories of gradient-based [94], [95] and evolutionary techniques [96]–[98]. Commonly, evolutionary techniques, specifically genetic algorithms, have been used as they are particularly well suited for either problems with a large number of input variables, combinatorial problems, or problems with complex (or not well understood) topologies. The more traditional genetic algorithm techniques can be inefficient when the objective function is computationally intensive such as when an FEM model is required [95] as will be in this work. In recent years, however, incorporating machine learning to improve the efficiency of performance evaluation has been investigated. The remainder of this section addresses how each method fares with respect to the expected multivariate, objectively complex design space of this work.

While genetic algorithms continue to be popular for discontinuous or complex design spaces [99], gradient based methods are efficient and used for materials and electromagnetics optimization and design [94], [95]. A popular method to extend the capabilities of gradient based methods in high dimensional design spaces is the adjoint method. Hughes et al. used the adjoint method to compute the gradient of an objective for design of a non-linear optical switch using only two full-field electric field simulations [95]. Michaels et al. also used a physics based

objective function to run a gradient based optimization scheme, noting that the time it takes to compute the gradient is significant since at least 2 simulations are required [94]. However, the computational expense of an evolutionary technique is significantly larger, since it requires multiple simulations for each generation, and is possibly why gradient based methods are used with physics based objective functions such as those discussed here.

Within data and computer science there are several approaches to optimizing a complex design space which can be classified under two categories, deterministic and stochastic. Deterministic approaches are reproducible when given the same starting parameters. There is no randomness introduced in these algorithms. A classic example of this type of approach is the hill climbing algorithm, recognized in an optimization context as gradient methods but applicable to other computer science and combinatorial problems [93]. Stochastic approaches utilize pseudo-random numbers and will be different each time they are run. Adapting simple models, such as the hill climb algorithm, with stochastic approaches can help avoid finding local minima only, at the cost of some convergence efficiency. Often times algorithms will draw on physical principles, such as in simulated annealing [93], or nature for solutions to complex issues to introduce some randomness in each new generation to find a global minimum [98]. Stochastic approaches can be used in conjunction with gradient based optimization to allow for more exploration of a more complex design space and avoid local minima, without incurring the complete cost of evolutionary techniques, such as genetic algorithms.

On the other hand, artificial intelligence approaches have been used to characterize the response of complex systems [100], [101], enabling the input parameters that produce optimized or target outcome variables to be computed. Furthermore, Wang et al. used an artificial intelligence scheme to generate constitutive laws for the failure of particle composites [102]. In

these previous examples, artificial intelligence optimization techniques have been used to both to predict the outcome of complex systems, and to estimate material response. Consequently, there is evidence that not only could artificial intelligence approaches efficiently determine the optimal constituent set for a given design (e.g. assist with simulation the formation of the microarchitectures in this project) but also relate processing parameters to effective composites properties.

Material modeling, optimization, and design all require a quick computationally inexpensive framework for answering any number of questions and satisfying varying constraints for which AI solutions have been of interest. Current approaches can be viewed as either forward modeling or as attempting to solve the inverse problem. Regressions, neural networks, and policy generating algorithms have all been used in various ways to predict material responses such as stress-strain curves [103], [104] and composite properties for systems that are more complex than effective medium theory accounts for [89], [103], [105]. These modeling approaches are classified as forward modeling, finding properties as outputs from some known input parameters. In contrast, the inverse problem attempts to find a set of independent variables for a desired dependent variable. Current approaches to inverse problems train models and implement guess and check type approaches, made quicker by the model. Inverse problems are an open challenge in the materials design literature [106], [107].

Machine learning has recently been used to attempt to answer both the forward and inverse problems. Chapman and Ramprasad [108] have developed a machine learning model trained to capture the temperature dependence of mechanical properties of platinum to augment dynamics simulations that are costly but more capable than semi-empirical methods. This model was also able to capture larger scale material phenomenon which would have previously been

too computationally expensive to model. Mohammadzadeh et al. [109] used gene expression programming to develop a closed form solution for predicting the compression index of fine grained soils based on other parameters. The model was developed from an experimental database of soil parameters and used to generate surface charts to predict the compression index. Machine learning and optimization have also been combined to make a hybrid model that optimally trains the model such as the neural network trained by Abdolrasol et al. [110] using population based optimization algorithms. Neural networks have been used to improve run times for optimization problems that require many datapoints, such as genetic algorithms [96], and to develop constitutive laws, such as for mechanical systems that can be computationally expensive to solve [102], [111], [112]. There is ample evidence that artificial intelligence and machine learning approaches can augment more traditional gradient and evolutionary optimization frameworks especially in the context of large, complex, computationally intensive design spaces such as the problem considered in this work.

### **1.2.5 Artificial Intelligence and Machine Learning Models for Defining the Design Space**

Artificial Intelligence (AI) and Machine Learning (ML), a subcategory of AI, have grown in popularity in recent years for various applications ranging from classic image classification [113] to creating art [114]. At their base ML models serve one of two primary functions, classification and regression. Within this there are many different models with many methods of application, laid out in Figure 1-16. Classification is useful for many problems such as the classic example of a spam mail filtering that assigns a label, such as spam or not spam to each email, or input data. The use of ML models within materials design requires models that are best used with regressions to predict system behavior in an input-output relationship, rather than assigning a classifier.

Models have also been used for materials design [106], [107]. Herein, materials design is defined as the determination of the parameters needed to produce a material with a set of desired properties. Some of the most common (Decision Tree, Gaussian Process, Support Vector Machine (SVM), Bayesian Networks, Neural Networks (NN), and k Nearest Neighbor (kNN)) have been laid out in Figure 1-16 showing a qualitative comparison of the degree of non-linearity and number of variables for which they are best used [106], [107], [115]–[118].

The independent variables for the multi-objective optimization being considered in this work include the constituents, of which the only variable is volume fraction, and the applied fields which are electric and a magnetic either parallel or perpendicular (total 4 variables) which all range from 0 to a maximum experimentally achievable value. This would be considered a low number of input variables. Furthermore, the system is expected to behave in a similar fashion to effective medium theories, which can largely be expressed as linear combinations of constituent derived parameters. Although the effective medium models cannot capture the effect of the structures in the field processed composites the general trend of the response is expected to be similar and is therefore expected to have a low degree of non-linearity.

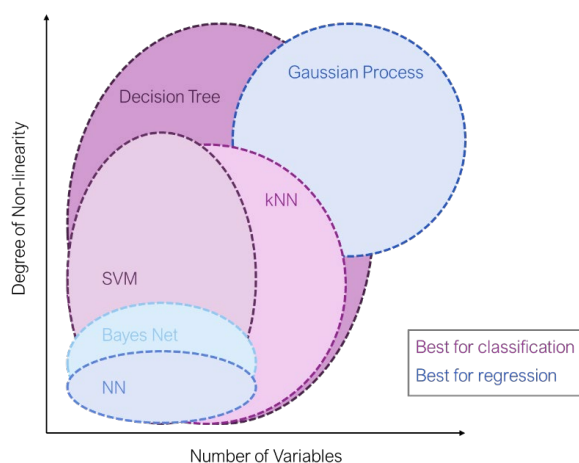


Figure 1-16: Ideal problem domains for selected data science and AI techniques (Decision Tree, Gaussian Process, Support Vector Machine (SVM), Bayesian Networks, Neural Networks (NN), k Nearest Neighbor (kNN))

In the context of regressions of material properties for material science and materials design NN are a common choice. One common type used is the feed forward neural network (FFNN), which uses layers of neurons that each compute an output based on inputs received from the previous layer [119]. An example of these layers is shown in Figure 1-17 with dense layers (all nodes in a layer are connected to every node in the previous layer). These types of models are trained by minimizing error between the prediction and the training data points by adjusting the weights between the layers. Each neuron in the layer is given an activation function, such as those shown in Figure 1-18, that is assigned to the whole layer [120]. The activation function calculates an output based on this function and feeds it to the next layer where the weights for each output determine the sum that is the input for each neuron in that following layer. This continues on to the output of the network.

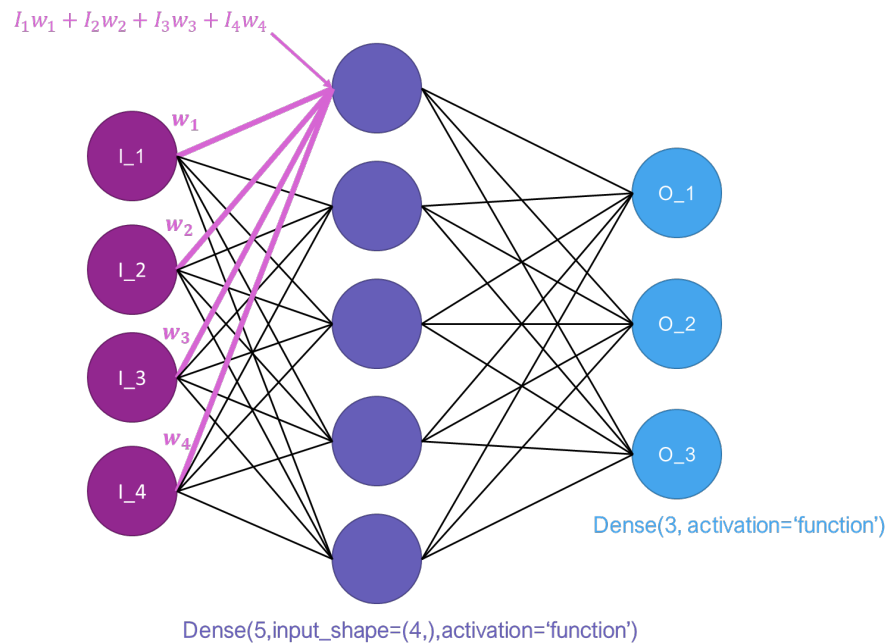


Figure 1-17: Example of a feed forward neural network (FFNN) showing inputs on the left, a dense layer in the middle, outputs on the right, and an example of the weighted sum of the input or any node's previous layer.



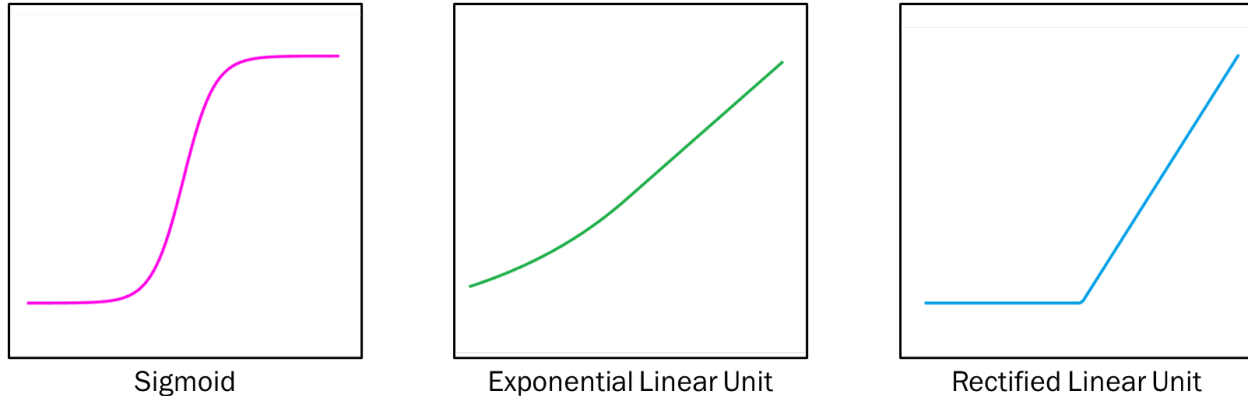


Figure 1-18: Examples of activation functions for use with a neural network

For example, Xiong et al. used a dense layer FFNN to model the relationship between processing parameters and weld bead geometry. The advantage of this approach is there is no need for making any simplifying assumptions as with an analytic approach, and the data can be used to determine correlations between input and output variables. This model was trained with experimental data and was shown to be able to predict bead geometry well and more efficiently than standard regression [121]. One reason this is a more useful approach is the elimination of fine adjustment to the base function because the connections between nodes, when they are not useful, can go to 0 over time, automatically making these adjustments. Once trained models such as this can be used to predict manufacturing outcomes in real time and can be used in control loops to adjust processes. Liu et al. used a numerical model to calculate the material property of a functionally graded material which was fed into a NN to be used for in-situ characterization of material properties. The system consisted of SiC with added carbon inclusions at various volume fractions, which are the independent variables of the system. The model was iteratively trained until converged to model the actual response of the system. This method was found to be effective and stable in characterizing the system, even in the presence of noise [105].

The efficacy of each modeling methodology is highly dependent on the problem it is attempting to solve. Examples of classification approaches are shown in Figure 1-19 for an SVM and NN used to classify a set of data. The SVM identifies the boundary that is the furthest away from all of the data points, whereas the NN adjust the weights of a function to classify the data in the training set. It is seen then that the SVM generates a different boundary from the NN, which in this example can be seen to be more robust against misclassification of future datapoints. It is therefore clear that it is necessary to consider the nature of the data when selecting the model and validation is crucial.

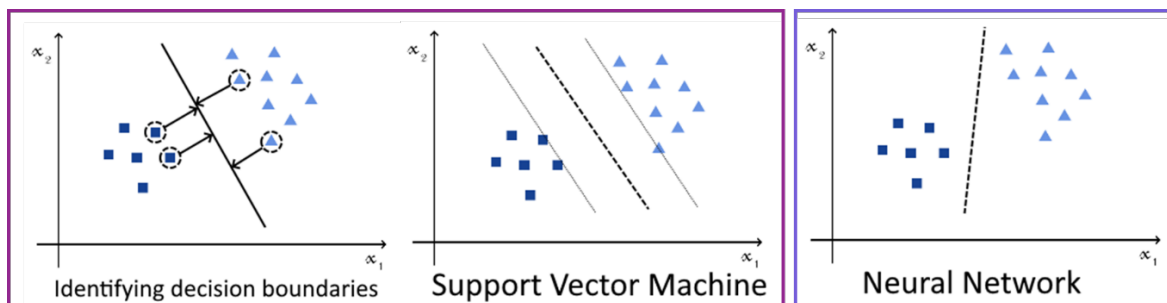
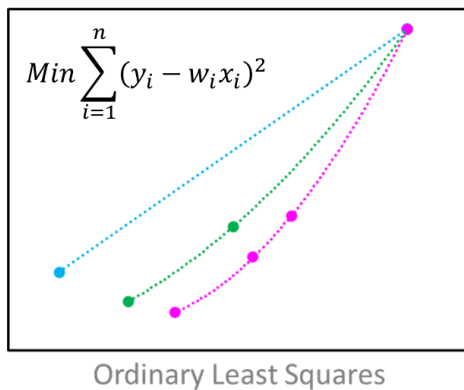


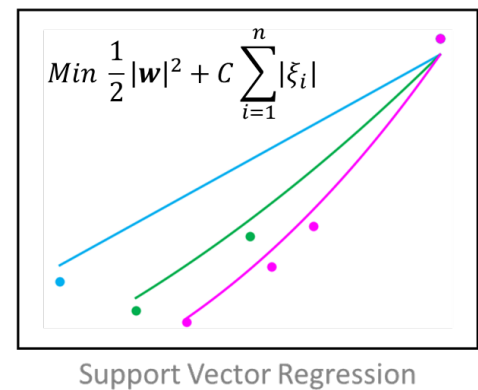
Figure 1-19: Distinction between SVM method (purple box) and Neural Network (blue box) of classification showing the method used to draw the decision boundary leading to a more robust model (better avoids misclassification) with the SVM method (adapted from [118])

Similar to Support Vector Machines, the model can be used for regressions. Support Vector Regressions determine weights by trying to minimize the weight vector while staying within predefined acceptable error bounds (shown in Figure 1-19 as the grey lines), and by extension also minimize the error, or deviation of the true data from the prediction, of the points that do not fall within these error bounds. However, the model does not include a penalty for points that fall within these acceptable error bounds. This error term is important for the quality of fit and is one of the hyperparameters that needs to be set. Figure 1-20 shows a comparison between a classic Ordinary Least Squares regression (Figure 1-20 a) that minimizes the error

between the points and the fit and SVR (Figure 1-20 b). As the error goes to 0 the two solutions converge to the same regression, demonstrated in Figure 1-21. Even though the lines of the SVR fit get closer to the points as the error goes to 0, there is not a concern about the fit simply connecting the points and overfitting the data, due to the nature of the model also minimizing the weights of the function. In this example,  $e_i = (y_i - w_i x_i)^2$  is the value of the error at a given data point  $i$  used for ordinary least squares regression, which is minimized. In both models,  $\mathbf{w}$  is the vector of weights in the fit function. The SVR minimizes this weight vector and the points outside of the allowable error bounds by the amount that they exceed this limit,  $\xi$ , with another hyperparameter,  $C$ , that weights how important this consideration for error is.



a)



b)

Figure 1-20: a) Ordinary Least Squares (OLS) compared to b) Support Vector Regression (SVR) model (for linear, quadratic and polynomial functions) showing the different approaches to fitting data where the OLS method attempts to minimize the squared error between the fit and the data, as opposed to the SVR which minimizes the weights of the regression terms within specified bounds.

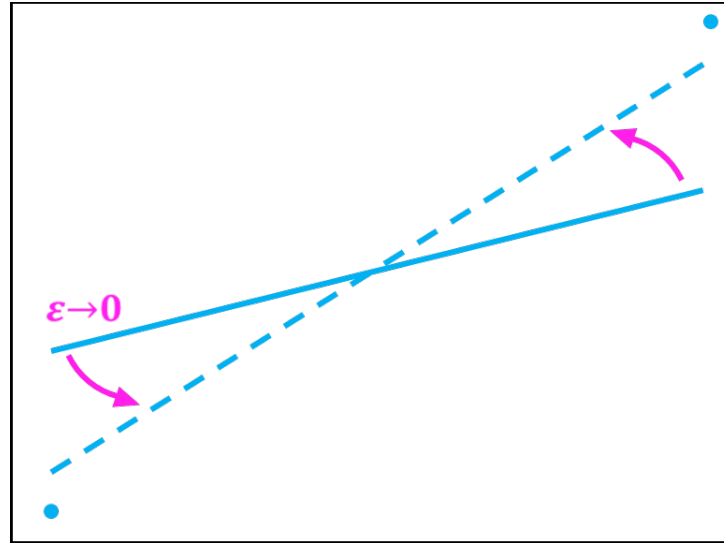


Figure 1-21: Linear Support Vector Regression fit as the allowed error goes to 0

Kurtoglu and Bakbak used SVR to model the shear strength of reinforced aerated concrete slabs, which consists of entrapped air voids in a cement matrix. 271 experimental data points were used in conjunction with several base functions to determine the best hyperparameters for the system. The experimental data was split into a training set to fit the base function and testing set to evaluate the model's performance. The radial basis function was determined to be most effective with a 0.931  $R^2$  value (or correlation coefficient) [122]. This study showed that SVR is a simple yet effective method for predicting material properties.

ML models assume a base function(s), or kernel, and adjust weights to fit the function to the system response. It is logical to conclude that the system would behave in a predictable manner such as described by traditional effective medium theory. This behavior should be taken into consideration when selecting kernels or creating custom kernels for these models. Training a model that provides a closed-form solution to the effective composite properties as a function of processing parameters would solve the materials design problem, and more important to the implementation of multifield processing, it would solve the inverse. Furthermore, the model can

be used in conjunction with a gradient based optimization, including expansion of constraints and objective functions, such as the lowest energy cost or volume fraction or whatever metrics are relevant to the specific design problem. This method maps the behavior of the system to be used to augment expensive computational homogenization for the implementation of this framework to real world additive manufacturing problems.

Recently researchers have been combining optimization and machine learning in two main ways [123]. The first is using a trained model to evaluate the objective function in order to speed up the optimization. The second uses the optimization to effectively train the model, adjusting the hyperparameters of the model itself to reduce error and converge quicker [124]–[126]. A multifidelity method is proposed by Perdikaris et al. which uses low fidelity models to train a model and optimize the hyperparameters which can then be used in training a more computationally expensive model, or experimental values, and eliminate the optimization needed on the hyperparameters with the added expense [127]. A smart sampling algorithm has been proposed by Wang et al. to evaluate the initial dataset and sample new points that perform well on some fitness function [128]. Similar to a genetic algorithm a set is evaluated, and a subset of best performers is used to determine the next set of sampling points. The selection of where to sample however is based on distance in the design space to the best performers. This method still creates a full set of data before training a machine learning model. This begins to touch on the basis of AI search techniques, the balance of exploring the search space to learn, and exploiting the knowledge gained. What has yet to be seen is the evaluation of the training quality of the model as the data is being generated to determine when to stop exploring the design space and start exploiting the model's knowledge.

### 1.3 Research Objectives and Tasks

In this section, a set of research tasks, serving to fulfill objectives that are in turn required to reach the stated goal are outlined. In the work below, barium hexaferrite and polydimethylsiloxane of varying volume fraction mixtures will be used as the composite material system for a multifield processed additive manufacturing application to locally tune properties with a single constituent set by varying processing conditions alone.

#### **Goal:**

Develop an optimization framework that enables the prediction of the Pareto front of the widest range of possible elastic, magnetic, and dielectric properties (objectives) that can be achieved by varying the electromagnetic field processing conditions, alone, for a given constituent set (input variables)

**Objective 1:** Computationally predict composite properties as a function of processing parameters (applied electric and magnetic fields) and constituents by simulating the processing event and computationally characterizing subsequent structures formed

Task 1.1 Improve fidelity of existing particle dynamics simulations, which model the electromagnetic processing event, by addressing artifacts introduced by the assigned minimum separation distance, particle shape-induced field magnetic and electric field effects (demagnetizing and depolarizing fields), and shape-induced hydrodynamic effects

Task 1.2 Develop framework to compute effective properties (mechanical, magnetic, and dielectric) using computational finite element homogenization methods performed

on representative volume elements generated from arrangements of particles predicted in Task 1.1

Task 1.3 Compute conductivity and dielectric properties from arrangement of particles predicted in Task 1.1 using artificial intelligence-based graph exploration techniques

Task 1.4 Validate the properties predicted in Tasks 1.2–1.3 with experimental data taken from the literature and experiments performed by partner labs

**Objective 2:** Computationally determine the widest range possible of the target properties that can be achieved by varying processing conditions (characterized by magnetic and electric field strengths and orientations) alone for a fixed combination of constituents (characterized by particle volume fraction)

Task 2.1 Create an optimization framework to conduct a line search of a constituent set to find the pareto front that defines the maximum and minimum properties of that constituent set, determined from Objective 1, as a function of processing conditions

Task 2.2 Incorporate results of Task 2.1 into a gradient based optimization scheme that determines the Pareto front of the widest range of target properties (objective functions) versus processing conditions and constituents (input variables)

**Objective 3:** Develop a machine learning approach to solve the inverse problem of predicting processing parameters and constituents as a function of composite properties

Task 3.1 Collect data using computational homogenization of Objective 1 to develop machine learning models for the correlation between constituents/processing parameters and composite properties

Task 3.2 Incorporate trained model into the optimization scheme of Objective 2 to improve the efficiency of the computational homogenization framework

Task 3.3 Validate the processing parameters chosen with experimentally measured composite properties

## 1.4 Dissertation Outline

The structure of the remained of this document is as follows.

**Chapter 2** presents the methods for each of the topics presented in this chapter. Homogenization techniques are explored and set up in a similar manner. The development of an AI approach to dielectric homogenization is laid out explaining the basis for the algorithm and a simple test case. Multi-objective optimization and several approaches to AI and machine learning are presented as possible options to incorporate into the optimization scheme to solve the inverse problem. Finally, various effective medium theories are laid out in detail noting assumptions and choices made in setting each up to best model the system in the context of verification.

**Chapter 3** presents the homogenization results and discusses the significance in the findings as well as some possible sources for error. This section lays out the results beginning with effective medium as verification. This is followed by FEA models and experimental results, compared in each section. Finally the AI search conductivity results are presented and compared with experimental results from literature.

**Chapter 4** introduces the multi-objective optimization that utilizes the homogenization framework presented. This include focuses on the results of a traditional gradient based optimization scheme and the drawbacks found with using this method alone. Additional results are presented showing the improvement from integration with machine learning models and an



augmented hybrid style scheme. The details for the machine learning models and the decisions made for the optimization presented here are further discussed in Chapter 5.

**Chapter 5** presents the results from training both Neural Networks and Support Vector Regression models. It presents the components used to integrate this method into the optimization scheme presented in Chapter 4, and the visualization of the trained models from simulation and experimental data.

**Chapter 6** discusses the conclusions from this work, the contributions to the field, and future work that can build off of this foundation.

## **CHAPTER 2:Methods**

This chapter describes the methods used for computational homogenization (from particle dynamics simulations of microarchitecture) to determine bulk material properties, and optimization of the constituent-processing design space to determine which combination provides the widest possible property range. The composite being considered is made of (micro) Barium hexaferrite (BHF) particles in a PDMS matrix. The framework for computational homogenization is also laid out describing the particle dynamics simulations, determination of RVE size, and FEM simulation methods and boundary conditions. An overview of classic effective medium models including equations and comparisons is given as a basis for validation of computational results. Using this framework to strategically explore the design space a machine learning model is developed to address the computational expense of the forward problem and to solve the inverse problem of determining the necessary processing conditions for a desired composite property. With both the forward and inverse problems addressed the best constituent set can be selected and tuning parameters can be set for local tuning of properties.

### **2.1 Homogenization**

#### **2.1.1 Particle Dynamics to Predict Microarchitecture**

The following section has been adapted from [25].

The effective material properties of a multi-field processed composite can be determined through simulations. Particle dynamics simulations can account for effects of applied fields and fluid dynamics to determine the microarchitecture formed during processing. The resulting structures can be modeled in 3D and solved via FEM to determine effective material properties. Key to success in this evaluation is proper definition of loading and efficient use of computational resources.

Colloidal suspension dynamics are governed by thermal, diffusive, electromagnetic, and hydrodynamic principles. However, it has been shown that for micrometer sized particles diffusion and thermal effects can be neglected [52], [129]–[131]. The forces considered acting on the particles in suspension therefore are, dielectrophoretic as shown in equation (1-1), magnetic as shown in equation (1-5), repulsive, and hydrodynamic. The BHF particles used in these simulations, and the experiments from which the simulations are derived, are hexagonal platelets with a permanent magnetic dipole through the face of the plate (e.g. normal to the plate surface). For the purposes of this work, the BHF particles are idealized as ellipsoidal particles (oblate spheroids in particular) due to the fact that this shape has been studied extensively and analytic forms of the electric and magnetic interactions are well understood [81]. The geometric representation of the idealized ellipsoid is shown in Figure 2-1. For the particles considered in this work the major axes,  $a$  and  $b$ , are set to  $0.5\ \mu\text{m}$  and the minor axis,  $c$ , is  $0.15\ \mu\text{m}$  and a permanent magnetic dipole moment in the direction of the  $c$  axis of  $370,000\ \text{A}\cdot\text{m}$ . These values are warranted give that the nominal dimensions of real BHF particles have a mean size of 1 to  $2\ \mu\text{m}$ .

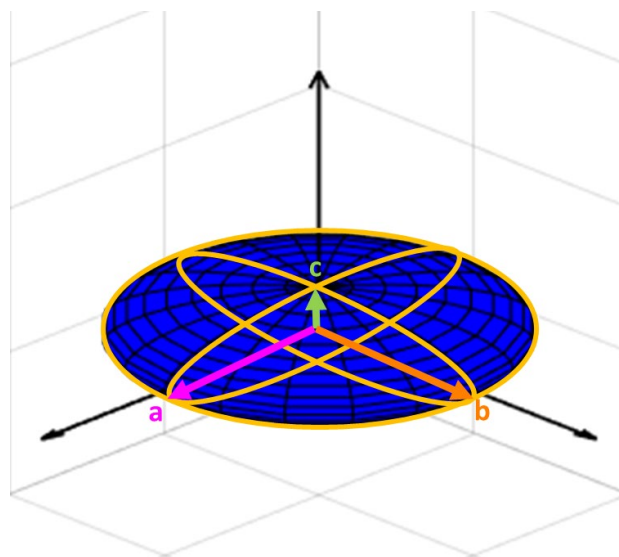


Figure 2-1: Idealized ellipsoidal representation of BHF showing major ( $a/b$ ) and minor ( $c$ ) axes

The field applied to each particle is a combination of the externally applied field and the response and effect of each of the other particles around it to that field. This combined electric and magnetic fields at the  $i^{\text{th}}$  particle are defined as

$$\mathbf{E}_i = \mathbf{E}_0 + \sum_{j=1}^N \frac{1}{4\pi\epsilon_0 r_j^3} [3(\mathbf{p}_j \cdot \hat{\mathbf{r}}_j) \hat{\mathbf{r}}_j - \mathbf{p}_j] \quad (2-1)$$

$$\mathbf{H}_i = \mathbf{H}_0 + \sum_{j=1}^N \frac{MV}{4\pi\mu_0 r_j^3} [3(\hat{\mathbf{e}}_j^3 \cdot \hat{\mathbf{r}}_j) \hat{\mathbf{r}}_j - \hat{\mathbf{e}}_j^3] \quad (2-2)$$

respectively, where  $\epsilon_0$  is the permittivity of free space,  $\hat{\mathbf{r}}_j$  is the unit vector between the  $i^{\text{th}}$  and  $j^{\text{th}}$  particles,  $r_j$  is the magnitude of the distance vector  $\hat{\mathbf{r}}_j$ ,  $\mathbf{p}_j$  is the dipole moment of the particle,  $\hat{\mathbf{e}}_j^3$  is the direction component of the magnetic dipole,  $M$  is the magnetization of the particle, and  $V$  is the volume of the particle. The rotation of each particle is described by the torque defined as [52]

$$(\mathbf{T}_{dep})_i = \mathbf{p}_i \times \mathbf{E}_i \quad (2-3)$$

$$\mathbf{p}_{eff} = \frac{4\pi abc}{3} (\epsilon_2 - \epsilon_1) \mathbf{E}^- \quad (2-4)$$

$$\mathbf{E}^- = \begin{bmatrix} \frac{E_{0,x}}{1 + \left(\frac{\epsilon_2 - \epsilon_1}{\epsilon_1}\right) L_x} \\ \frac{E_{0,y}}{1 + \left(\frac{\epsilon_2 - \epsilon_1}{\epsilon_1}\right) L_y} \\ \frac{E_{0,z}}{1 + \left(\frac{\epsilon_2 - \epsilon_1}{\epsilon_1}\right) L_z} \end{bmatrix} \quad (2-5)$$

$$L_x = \frac{c}{2a} \left[ \frac{\pi}{2} - \frac{c}{a} \right] \quad (2-6)$$

$$L_y = \frac{c}{2a} \left[ \frac{\pi}{2} - \frac{c}{a} \right]$$

$$L_z = 1 - \frac{c}{a} \left[ \frac{\pi}{2} - \frac{c}{a} \right]$$

$$(\mathbf{T}_{\text{mag}})_i = -\mu_0 MV(\mathbf{p}_i \times \mathbf{H}_i) \quad (2-7)$$

The electric field induce dipole moment,  $\mathbf{p}_j$ , is a function of the particle shape. As noted above, the idealized ellipsoid used for the BHF particles have 2 axes of the same dimension, leading to the same depolarizing factor,  $L_x$  and  $L_y$ , in the a and b directions. The shorter minor axis, c, will have a smaller induced field as a result of this shape anisotropy, leading to a preferential orientation of the particles with either the a or b axis aligned with the externally applied electric field.

The ability of the resulting torques to orient the particles with respect to externally applied fields, separate from net forces ordering particles into chains, allows for ordering within those particle chains, which is not possible with spherical particles. Spherical particles have no shape anisotropy, therefore within chains, differentiation of physical alignments of particles are not possible. However, it should also be noted that as mentioned the major axes have the same dielectric depolarizing factors such that both are equally preferential for aligning with the applied electric field. Consequently, these oblate spheroids develop preferential alignments along the applied field direction projected onto their a-b planes with respect to external electric fields. It is the electric field projected along this axis from which torques develop. Further shape anisotropy would change this behavior and the internal hierarchy of the particles within chains.

The development of a preferred dielectric polarization axes and the resulting dielectric torque sits in contrast to the development of magnetic torque. Magnetic torques will always be generated with respect to the fixed magnetic axis of the particle (given the particle is free to rotate). Given that the magnetic axes lie along the minor axis of the particle, and that dielectric polarization develops in the plane of the major axes, the magnetic dipole and induced electric dipole will be orthogonal to each other, thus creating the ability to generate similarly orthogonal controlling torques. The orthogonality of these axes also leads to orthogonal ordering at the chain level as well as through dielectrophoretic and magnetophoretic force responses. In a broader sense, it is the orthogonality of the magnetic and dielectric axes, then, that lead to the ability to generate differentiated hierarchical ordering within chains. It is therefore expected that any particles whose easy magnetic and dielectric axes are orthogonal, and likely those which are simply not collinear, may also be amenable to varying degrees of similar orthogonal control and subsequent hierarchical ordering.

As volume fraction increases some factors such as particle-particle interactions become more important. The particles' mobility generally decreases, and it is more difficult to form structures. The matrix viscosity naturally is unchanged; however, this increase in particle-particle interactions increases the viscosity of the colloid, which forms the basis of magnetorheological fluids. The governing equations dictating forces and torques are necessarily driven by local particle-particle proximity. Equations (2-1)-(2-7) highlight how particle spacing affects the forces and moments acting on particles in the system. As the spacings decrease the influence of the electric or magnetic field between the particles increases and the attractive forces increase, however, the second term in (2-1) and (2-2) contain dependencies on relative particle alignments

that may affect the local fields. The overall effect is to minimize the space between particles and differences in orientation to minimize the energy in the system.

From the literature it can be determined that the Reynolds number is on the order of  $10^{-6}$  [52]. This allows for the use of Stokes drag for determining the hydrodynamic force on the particles. For a given particle the force can be describes as

$$(\mathbf{F}_{drag})_i = -D\dot{\mathbf{x}}_i \quad (2-8)$$

where  $D$  is the drag coefficient and  $\dot{\mathbf{x}}_i$  is the velocity vector of that particle. For the simplified a spherical particle the drag coefficient can be described as

$$D = 6\pi\eta R \quad (2-9)$$

where  $R$  is the hydraulic radius and is the  $\eta$  dynamic viscosity of the fluid. In this work, computations will assume the fluid medium itself maintains a constant uniform viscosity, which is physically reasonable. Particle mobility, however, will be affected by the electromagnetic and physical interaction of particles, both of which are addressed in simulations. Electromagnetic interactions follow equations (2-1) and (2-2), while the physical interaction in simulation are governed by equation 2-8. Additionally particles are subject to a torque causing rotation.

Hydrodynamic torque can be modeled as

$$(\mathbf{T}_{drag})_i = 8\pi\eta R^3\boldsymbol{\omega}_i \quad (2-10)$$

where  $\boldsymbol{\omega}_i$  is the angular velocity of the particle. This angular velocity vector can be defined in terms of the particle orientation vector,  $\mathbf{d}_i$ , and its time derivative,  $\dot{\mathbf{d}}_i$ , as  $\boldsymbol{\omega}_i = \mathbf{d}_i \times \dot{\mathbf{d}}_i$ .

Particle repulsive forces are modeled via traditional hard sphere interactions to maintain some degree of separation between particles centers in a simulation environment [77], [132], [133]

$$(\mathbf{F}_{rep})_i = \sum_{j=1}^N Q * e^{-\beta(\frac{r_m}{2R}-1)} \hat{\mathbf{r}}_j \quad (2-11)$$

where  $Q$  is a variable scaling factor adjusted to ensure stability of the simulations,  $\beta > 0$  is the repelling parameter, and  $\hat{\mathbf{r}}_j$  is the position unit vector between particles  $i$  and  $j$ .

Additional consideration for particle shape will be added to avoid large gaps in particle chains as shown in Figure 2-2 as a result of equation (2-11). This simplification is likely to skew simulation results for composite properties, particularly in the dielectric domain.

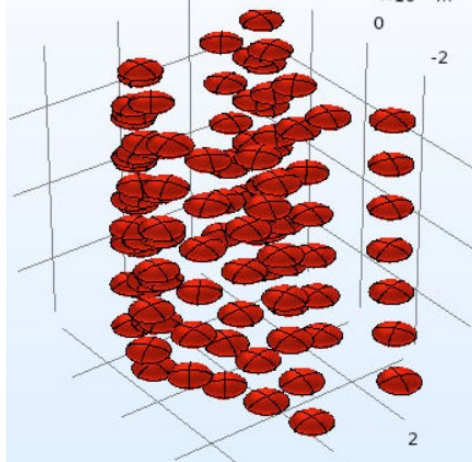


Figure 2-2: magnetic alignment of ellipsoidal particles that are governed by a separation force assuming a spherical shape

There are several other factors in the behavior of these particle systems that exist experimentally and could be revisited to remove additional error. On the microscale forces such as Van der Waals forces, diffusion, Brownian motion, steric repulsion due to surfactants, and



gravity also act on the particles. The degree to which they contribute to the overall motion of the particles in comparison to the applied is small and they are not considered in the models used.

These governing equations, (2-1)-(2-11), are time integrated to solve for final positions and orientations of the particles. Details of the time integration techniques can be found in [26], [52]. The time steps are determined by the magnitude of the forces and torques such that the accelerations can reasonably be linearly approximated. The simulation is then run for a total of 0.5 seconds of in-simulation time.

Particle dynamics simulations provide a path from processing parameters to particle distributions that can be modeled for use with FEA. This type of definition for RVE's provides the largest degree of flexibility with the constituent definition and accounting for the effects of the microarchitecture formed in applied fields, defining a clear pathway to determine material properties from processing parameters.

### *Ellipsoidal Particle Spacing*

To improve simulation fidelity and thereby homogenization results the simplifying assumption of spherical distances between particles was investigated. The spherical distance assumed in equation (2-11) is modified by calculating the surface to surface distance between two ellipsoidal particles. The formula for this has been worked out by Paramonov and Yaliraki [134] using methods described by Cleaver et al. [135]. The ellipsoidal geometric values used to calculate the distance between the two surfaces are laid out as described in Figure 2-3. For the two particles the positions and orientations are known.

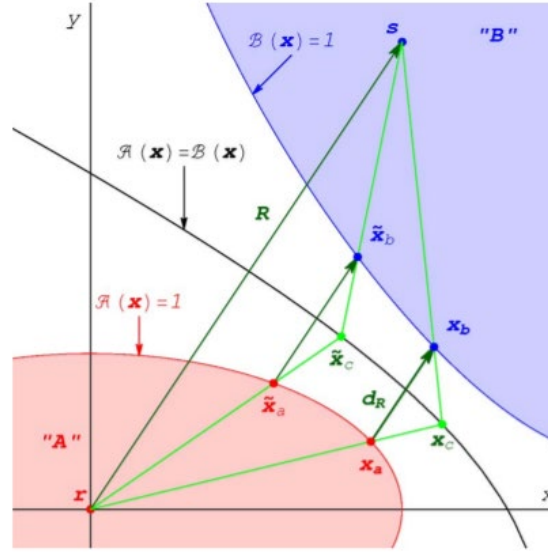


Figure 2-3: Geometric values used to calculate the surface to surface distance between two ellipsoids  $d_R$  is the minimum distance between the surfaces,  $\mathbf{R}$  is the center to center vector  $x_a$  and  $x_b$  are points on the surface of the two ellipsoids (A and B, respectively), and  $r$  and  $s$  are the center points of ellipsoids A and B, respectively

The minimum distance between the two ellipsoidal surfaces is then described as

$$d_R(\mathbf{A}, \mathbf{B}) = R(1 - F(\mathbf{A}, \mathbf{B})^{-\frac{1}{2}}) = R - \sigma_{PW}(\mathbf{A}, \mathbf{B}) \quad (2-12)$$

where  $\mathbf{A}$  and  $\mathbf{B}$  are the ellipsoids,  $R$  is the magnitude of the vector between the two centers,  $F$  is the elliptical contact function (defined in detail in [135]),  $f$  is the result of normalizing  $F$  by the magnitude of  $R$ , and  $\sigma_{PW}$  is the Perram-Wertheim range parameter (which is a function of the orientations of the ellipsoids). The Perram-Wertheim range parameter is further defined as

$$\sigma_{PW}(\mathbf{A}, \mathbf{B}) = \frac{R}{\sqrt{F(\mathbf{A}, \mathbf{B})}} = \frac{1}{\sqrt{f(\mathbf{A}, \mathbf{B})}} \quad (2-13)$$

$$f(\mathbf{A}, \mathbf{B}) = f(\hat{\mathbf{u}}_i, \hat{\mathbf{u}}_j, \hat{\mathbf{r}}_{ij}) \quad (2-14)$$

$$= \frac{1}{4\Delta} \left[ 1 + \frac{c_x(\hat{\mathbf{r}}_{ij} \cdot \hat{\mathbf{u}}_i)^2 + c_y(\hat{\mathbf{r}}_{ij} \cdot \hat{\mathbf{u}}_j)^2 + c_x c_y (\hat{\mathbf{r}}_{ij} \cdot \hat{\mathbf{u}}_i)(\hat{\mathbf{r}}_{ij} \cdot \hat{\mathbf{u}}_j)(\hat{\mathbf{u}}_i \cdot \hat{\mathbf{u}}_j)}{1 - c_x c_y (\hat{\mathbf{u}}_i \cdot \hat{\mathbf{u}}_j)^2} \right] \quad (2-15)$$

$$\Delta = \frac{1}{2}(b_i^2 + b_j^2)$$

$$c_x = \frac{b_i^2 - a_i^2}{b_j^2 + a_i^2} \quad (2-16)$$

$$c_x = \frac{b_j^2 - a_j^2}{b_i^2 + a_j^2} \quad (2-17)$$

where  $a$  and  $b$  are the principal semi axes of the ellipsoids,  $\hat{\mathbf{u}}_i$  and  $\hat{\mathbf{u}}_j$  are the direction unit vectors of the ellipsoids A and B respectively, and  $\hat{\mathbf{r}}_{ij}$  is the vector between the centers of the ellipsoids.

The ellipsoidal (surface to surface) distance is then used to find a repulsive force between the two ellipsoids such that an equilibrium would be found when the particles are just contacting but not overlapping. This is done by scaling the magnetic and dielectric forces between the particles, which is already being calculated as part of the particle dynamics simulations. Similar to the previously established hard particle contact formula, currently assuming spherical distances, an exponential function is used with this newly found ellipsoidal distance to scale the magnetic and dielectric forces with distance such that the repulsion quickly goes to zero as the particles move apart.

$$(\mathbf{F}_{rep})_i = -\mathbf{F}_{mag} e^{-\beta(1-\frac{d\mathbf{R}}{r})} - \mathbf{F}_{diel} e^{-\beta(1-\frac{d\mathbf{R}}{r})} \quad (2-18)$$

where  $\mathbf{F}_{mag}$  is the magnetic force vector between the two particles,  $\mathbf{F}_{diel}$  is the dielectric force vector between the two particles,  $d\mathbf{R}$  is the distance between the two surfaces,  $r$  is the distance between centers, and  $\beta$  is a calibration term as before.

### 2.1.2 Computational Finite Element Homogenization

Once microarchitectures are predicted from particle dynamics simulations, the spatial location and orientation data may be used to construct solid models, shown in Figure 2-4, which are then

analyzed via finite element analysis. This analysis employs periodic boundary conditions on the predicted structures yielding homogenized responses from which effective properties may be drawn.

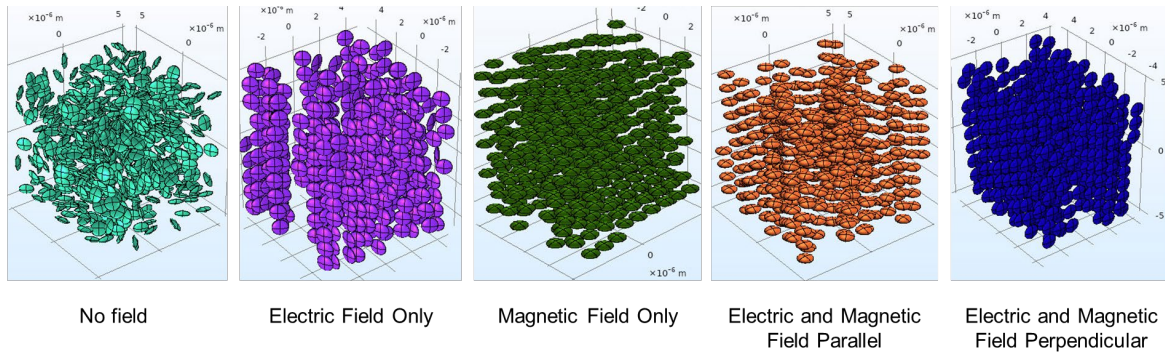


Figure 2-4: RVE showing particle microarchitecture formed from applying various fields

In the mechanical domain compliance entries were used to determine convergence. Once converged, the particle count is fixed, and a sweep of volume fractions can be used to determine the effect of volume fraction on material properties. This can then be compared to analytical solutions, such as the Mori-Tanaka model.

The individual steps and concepts discussed in this section are brought together in a scripted workflow that uses processing parameters and constituent sets as independent variables to determine the microarchitecture of the RVE and determine the effective material properties in the selected domains as detailed in Figure 2-5. In the figure, the framework begins by selecting processing parameters and using the Particle dynamics simulations presented in Section 2.1.1 to determine the field dependent distributions within the RVE. The resulting particle distributions (positions and orientations) are then used to generate a 3-dimensional model which can be homogenized using methods presented in this section. The resulting properties are then used to determine the next set of processing parameters to be tested. The iterations over the various

independent variable combinations are determined by the resulting outputs and objective function, comprising this optimization framework. The framework being developed will be verified and validated for BHF particles in a PDMS matrix. However, as other types of composites, such as spherical particles or fibers, are expected to also follow the general trend of effective medium it is reasonable to expect the same could be done for other systems with the same methods.

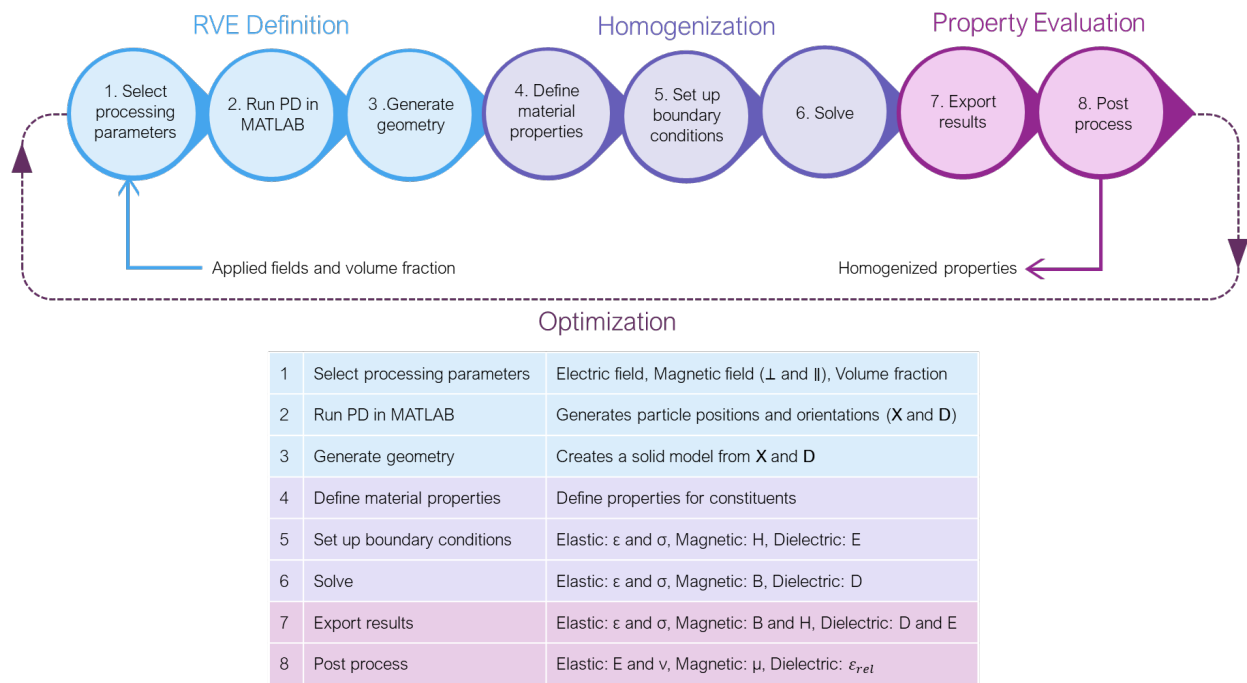


Figure 2-5: Scripting workflow from particle dynamics simulation to effective material properties

### 2.1.2.1 Elastic Boundary Conditions and Effective Properties

In the mechanical domain the stiffness matrix can be populated by isolating each of the stress responses to a prescribed strain case. In this case the composite is assumed to have orthotropic symmetry, such that three normal and three shear cases can be used to determine the non-zero entries of the stiffness matrix. With the chosen periodic boundary conditions, a small strain is

applied to opposite faces, and the stress-strain response is calculated. Each load case isolates a given stress and strain pair to solve the constitutive law shown below.

$$\boldsymbol{\sigma} = \mathbf{C}\boldsymbol{\varepsilon} \quad (2-19)$$

where  $\mathbf{C}$  is the  $6 \times 6$  stiffness matrix,  $\boldsymbol{\sigma}$  is the  $6 \times 1$  stress vector, and  $\boldsymbol{\varepsilon}$  is the  $6 \times 1$  strain vector. We assume an orthotropic response given that alignment along a preferred axis is the ultimate goal with isotropy in the transverse plane an expected outcome. The full tensor for an orthotropic material is

$$\boldsymbol{\sigma} = \begin{bmatrix} C_{11} & C_{12} & C_{13} & C_{14} & C_{15} & C_{16} \\ C_{21} & C_{22} & C_{23} & C_{24} & C_{25} & C_{26} \\ C_{31} & C_{32} & C_{33} & C_{34} & C_{35} & C_{36} \\ C_{41} & C_{42} & C_{43} & C_{44} & C_{45} & C_{46} \\ C_{51} & C_{52} & C_{53} & C_{54} & C_{55} & C_{56} \\ C_{61} & C_{62} & C_{63} & C_{64} & C_{65} & C_{66} \end{bmatrix} \boldsymbol{\varepsilon} \quad (2-20)$$

where the pink entries represent the non-zero terms.

To begin to analyze the RVE, we examine a load case 1 where we apply a small displacement only to a single face of our periodic RVE that yields a macroscopic strain  $\delta_1$ . Note that this necessarily yields a stress  $\sigma_1$  on that face and results in  $\sigma_2 = \sigma_3 = 0$  on unloaded and unconstrained transverse faces. Then by examining only non-zero terms in  $\mathbf{C}$ , equation (2-20) simplifies to

$$\sigma_1 = C_{11}\delta_1 + C_{12}\varepsilon_2 + C_{13}\varepsilon_3 \quad (2-21)$$

$$0 = C_{21}\varepsilon_1 + C_{22}\varepsilon_2 + C_{23}\varepsilon_3 \quad (2-22)$$

$$0 = C_{31}\varepsilon_1 + C_{32}\varepsilon_2 + C_{33}\varepsilon_3 \quad (2-23)$$

$$0 = C_{44}\varepsilon_4 \quad (2-24)$$

$$0 = C_{55}\varepsilon_5 \quad (2-25)$$

$$0 = C_{66}\varepsilon_6 \quad (2-26)$$

where  $\delta_1$  is the imposed small strain, and  $\varepsilon_i \forall i = 2 \dots 6$  are the expected resulting strains on the RVE, and  $\sigma_1$  is the expected stress on the loaded face. This is illustrated in Figure 2-6. Solution of this boundary value problem leads to expressions (if analytical), or values if computational, for  $\sigma_1$  and  $\varepsilon_i \forall i = 2 \dots 6$ .

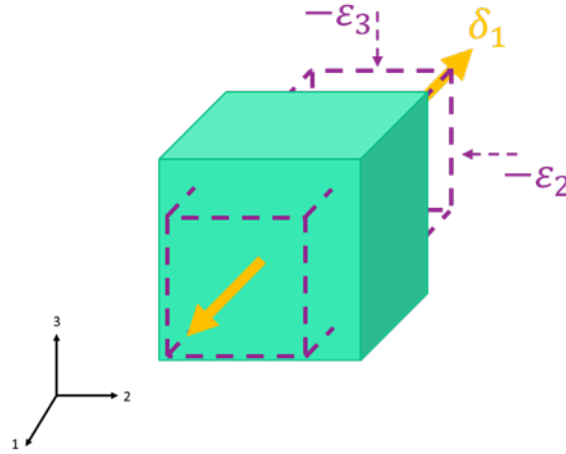


Figure 2-6: Load case 1 applied strain (yellow) and resulting strains (purple) for an RVE with periodic boundary conditions

Looking at equations (2-21):(2-23), there are 3 equations and 9 unknowns. Considering the additional 6 equations from load cases 2 and 3, where  $\delta_2$  and  $\delta_3$  are applied to appropriate faces yielding  $\sigma_2$  and  $\sigma_3$  as the only non-zero quantities, respectively, these three load cases are sufficient to solve the normal response portion of the stiffness matrix, e.g.  $C_{ij} \forall i, j = 1 \dots 3$ .

Similar small strain can be applied, and resulting stresses calculated, for the shear terms to solve

for all 12 unknown non-zero constants equation (2-20). The frequency response in the elastic domain will involve setting up additional load cases to capture time dependent values. This will also require a full response curve of the base materials, specifically the PDMS matrix material.

Solution of the conservation of linear momentum for these boundary value problems is completed in COMSOL. The material response for all of the load cases is reported in a stiffness matrix, which can be inverted to find the compliance matrix, which is used to determine the elastic modulus for comparison with experimental values. The definition of compliance matrix,  $\mathbf{S}$ , with elastic moduli,  $E$ , shear moduli,  $G$ , and Poisson's ratios,  $\nu$ , for an orthotropic material is shown in (2-27).

$$\mathbf{S} = \begin{bmatrix} \frac{1}{E_x} & \frac{-\nu_{xy}}{E_y} & \frac{-\nu_{zx}}{E_z} & 0 & 0 & 0 \\ \frac{-\nu_{xy}}{E_x} & \frac{1}{E_y} & \frac{-\nu_{zy}}{E_z} & 0 & 0 & 0 \\ \frac{-\nu_{zx}}{E_x} & \frac{-\nu_{yz}}{E_y} & \frac{1}{E_z} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{G_{xy}} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{G_{xz}} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{G_{yz}} \end{bmatrix} \quad (2-27)$$

The computational homogenization scheme presented in this section is verified with traditional effective medium theory in Section 3.2.2 and validated with experimental values in Section 3.3.1. The methods presented form the basis for determining the elastic response of multi-field processed composites based on constituents and processing parameters.



### 2.1.2.2 Dielectric Boundary Conditions and Effective Properties

Similarly, dielectric load cases are constructed to determine the relative effective permittivity of the composite material. These load cases were designed to resemble experimental techniques. In each case there is an applied known voltage, and a resulting current is measured that can be post processed to determine dielectric properties for the composite.

Figure 2-7 shows an equivalent circuit for a dielectric material in an experimental test setup. The material can be represented as a capacitor and a conductor in parallel which represent the storage and loss components of the dielectric response. The test includes a known applied voltage and a measured resulting current, both as a function of time. A representative response of this type of system is plotted in Figure 2-8 which shows the input voltage and each component of the resulting current. It can be seen that the capacitive component contributes the phase shift component to the total output current which would not be present in a purely conductive material.

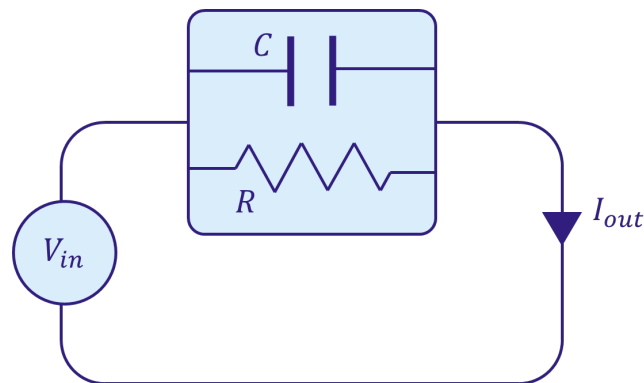


Figure 2-7: Equivalent circuit for a dielectric material in an experiment test setup showing the applied voltage and resulting current

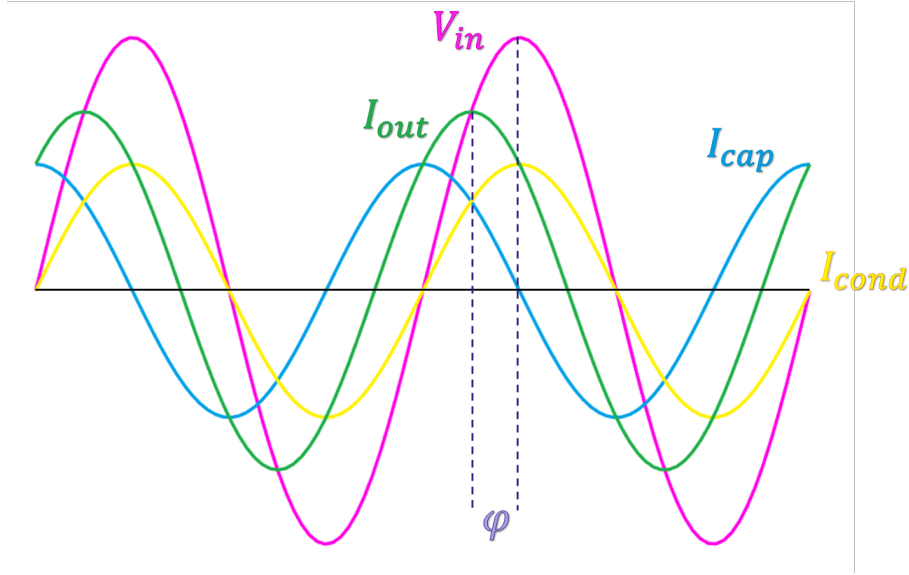


Figure 2-8: Theoretical response of a dielectric system whose equivalent circuit can be represented by Figure 2-7, showing the input voltage  $V_{in}$ , the total output current  $I_{out}$ , the capacitive component of the output current  $I_{cap}$ , and the conductive component of the output current  $I_{cond}$

The details of this response are laid out in Appendix B. From the analysis presented in Appendix B, the phase angle is defined as the arctangent of the ratio of the imaginary,  $Im[I_{out}]$ , and real,  $Re[I_{out}]$ , components of the complex current response to an AC voltage.

$$\phi = \tan^{-1} \left( \frac{Im[I_{out}]}{Re[I_{out}]} \right) \quad (2-28)$$

The phase angle is then used to determine the constants for resistance,  $R$ , and capacitance,  $C$ , using the known frequency,  $\omega$ , the known voltage input amplitude,  $V_{in}$ , and the measure output current amplitude  $I_{out}$ .

$$\frac{1}{R} = \frac{I_{out} \sin(\phi)}{V_{in}} \quad (2-29)$$

$$C = \frac{I_{out} \cos(\phi)}{\omega V_{in}} \quad (2-30)$$

Similarly to the elastic case, we may examine the response of an RVE to load cases that apply current in each of three orthogonal directions, detailed in Figure 2-9. These load cases are again defined with periodic boundary conditions, in this case on 2 pairs of faces shown in Figure 2-9 b in pink. The periodic boundary conditions define voltage continuity across face pairs. The remaining faces, shown in Figure 2-9 a, have an applied voltage,  $V$ , defined on each face. The voltage is defined as a constant value and then varied with a harmonic perturbation in the frequency domain study of the model to evaluate the AC response.

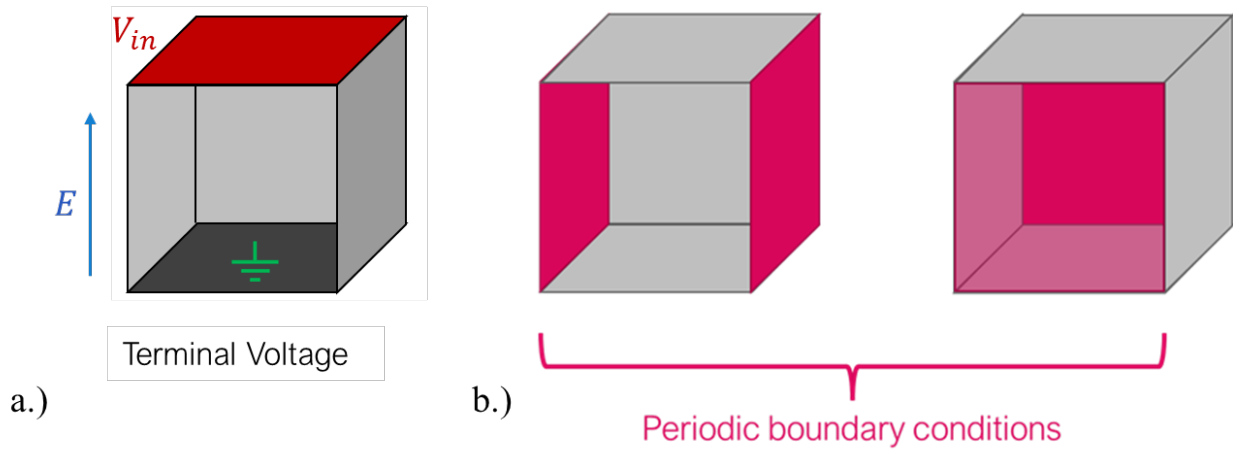


Figure 2-9: Representation of boundary conditions on the dielectric RVE a) electric field,  $E$ , from applied terminal voltage,  $V_{in}$ , and ground on the top and bottom faces, respectively b) periodic boundary conditions (pink)

The constitutive equations for the system use material property definitions for relative permittivity,  $\epsilon_r$ , and conductivity,  $\sigma$ .

$$\mathbf{J} = \sigma \mathbf{E} + j\omega \mathbf{D} \quad (2-31)$$

$$\mathbf{D} = \epsilon_0 \epsilon_r \mathbf{E} + \mathbf{P} \quad (2-32)$$

$\mathbf{J}$  is the complex current density,  $\mathbf{E}$  is the applied electric field,  $\mathbf{D}$  is the electric displacement,  $\omega$  is the frequency of the applied voltage,  $\epsilon_0$  is the permittivity of free space, and  $\mathbf{P}$  is the polarization of the dielectric material. The charge conservation equation is shown in (2-33) and the compatibility equation in (2-34) for the charge density,  $q$ .

$$\nabla \cdot \mathbf{J} = \frac{\partial q}{\partial t} \quad (2-33)$$

$$\mathbf{E} = -\nabla V = \frac{q}{\epsilon_0} \quad (2-34)$$

The current density in the solution is integrated on the terminal face that the voltage was applied on. The resulting current, as mentioned previously, is a complex number for the AC voltage that was applied. This value is then used with the equations (2-28)-(2-30) to determine the effective capacitance and conductivity of the volume. Both of these values are geometry dependent. The RVE dimensions are then used to extract the material properties, permittivity and resistivity, from the capacitance and conductivity.

$$R = \rho \frac{D}{A} \quad (2-35)$$

$$C = \epsilon_r C_0 = \epsilon_r \epsilon_0 (A/D) \quad (2-36)$$

where  $\rho$  is the resistivity,  $\epsilon_r$  is the relative permittivity,  $\epsilon_0$  is the permittivity of free space,  $A$  is the cross sectional area of the face, and  $D$  is the length between the two faces of the RVE.

The boundary conditions shown in Figure 2-9 are used in COMSOL by Nilsson et al. [79] to solve Maxwell's equations and subsequently for conductivity of a composite of short fibers in a DC case and by Lyngdoh et al. for spherical inclusions [136]. Nilsson et al. was not able to correlate FEM results with experimental results due to the computational expense for the fiber volume loading in the domain, which contained thousands of fibers due to aspect ratios and

volume fractions in experimental values, necessitating additional models, such as electrical networks modeling, to correlate to experimental values [79]. However, Lyngdoh et al. were able to achieve good correlation to experimental data found in the literature for spherical inclusions using this method [136], suggesting that an aspect ratio of a filler at (or near) unity can reduce computational costs. In the work of this thesis the aspect ratio of Barium hexaferrite (BHF) is 3 compared to the 11-26 of the fibers in the study done by Nilsson et al. [79], which is expected to reduce the number of particles needed to a computationally feasible amount. Although the methods presented here work for the BHF particles considered in this work, more conductive fillers and higher aspect ratio fillers have been shown in the literature to have complications using similar methods as described. Therefore, an alternative method for calculating conductivity for composites with highly conductive filler material in Section 2.3.

#### 2.1.2.3 Magnetic Boundary Conditions and Effective Properties

Similarly, magnetic models are setup to solve for the effective magnetization,  $\mathbf{M}$ , of the representative volume. The faces here however do not have a periodic boundary condition applied since the field lines would not be permitted to leave the RVE in this case, which introduces a false conservation of flux. In this case then, the RVE is placed inside a sufficiently large volume of air. Gauss' Law of Magnetism is then valid for the boundary of the air domain conserving the flux density,  $\mathbf{B}$ .

$$\nabla \cdot \mathbf{B} = 0 \quad (2-37)$$

A background field,  $\mathbf{H}_0$ , is applied to solve for the resulting magnetic flux density. The magnetic field is then defined by the background field and the magnetic scalar potential, the dependent variable in the solution.

$$\mathbf{H} = -\nabla V_m + \mathbf{H}_0 \quad (2-38)$$

Equation (2-39) shows the constitutive equation is given by Ampere's Law, where  $\mathbf{H}$  is the applied magnetic field, and  $\mu_0$  is the permeability of free space.

$$\mathbf{B} = \mu_0(\mathbf{H} + \mathbf{M}) \quad (2-39)$$

By applying a known  $\mathbf{H}$ , one may solve for the resulting  $\mathbf{B}$  field. Each particle in the volume has a permanent magnetic dipole represented by a magnetic flux conservation condition applied in the direction of a given particle's dipole moment, here normal to its surface. Lastly, from Ampere's Law and the magnetostatic conditions it can be determined that there are no free currents, defined by equation (2-40).

$$\nabla \times \mathbf{H} = 0 \quad (2-40)$$

The field is then integrated over the volume to determine the magnetic dipole moment,  $\mathbf{j}$ , which can be converted to mass magnetization [ $emu/g$ ] (shown in Appendix A) for comparison with experimental results.

$$\mathbf{j}[Tm^3] = \int \mathbf{B} [T] dV [m^3] \quad (2-41)$$

where  $\mathbf{B}$  is the local magnetic flux density and  $V$  is the volume. The dipole moment is calculated using COMSOL, which accounts for the magnetization of the particles and the effect of the magnetization on neighboring particles.

An alternative approach can be used to find an upper bound of mass magnetization by summing the dipole moment vectors of the particles in a volume and normalizing by the mass.

We may start with

$$M_i[emu] = m_i \left[ \frac{emu}{cm^3} \right] * V_i[cm^3] \quad (2-42)$$

where  $m$  and  $V_i$  are the magnetization and volume of the  $i^{th}$  particle, respectively, and  $M_i$  is the resulting dipole moment of the  $i^{th}$  particle. Given that only the particles, and not the PDMS matrix, are affected by the magnetic field, the dipole moment of the entire volume is the sum magnetizations of all,  $N$ , of the dipole moments of the particle within the volume,  $M_{total} = \sum_{i=1}^N M_i$ , which can then be mass normalized.

$$\frac{M_{total}[emu]}{mass_{ptotal}[g]} = \sigma[emu/g^*] \quad (2-43)$$

where  $mass_{ptotal}$  is the total mass of only the particles in the sample composite,  $mass_{ptotal} = \sum_{i=1}^N mass_i$  where  $mass_i$  is the mass of the  $i^{th}$  particle. This method does not account for any particle-particle interactions which is why it is only considered as a bound. The effect of the particle interactions is apparent in Figure 2-10, which shows the full field solution determined via the computational homogenization laid out earlier in this section and the vector sum method. The difference in the mass magnetizations determined with these two methods illustrates the importance of the full scale homogenization technique which will therefore be used in the optimization framework presented in Section 2.4.

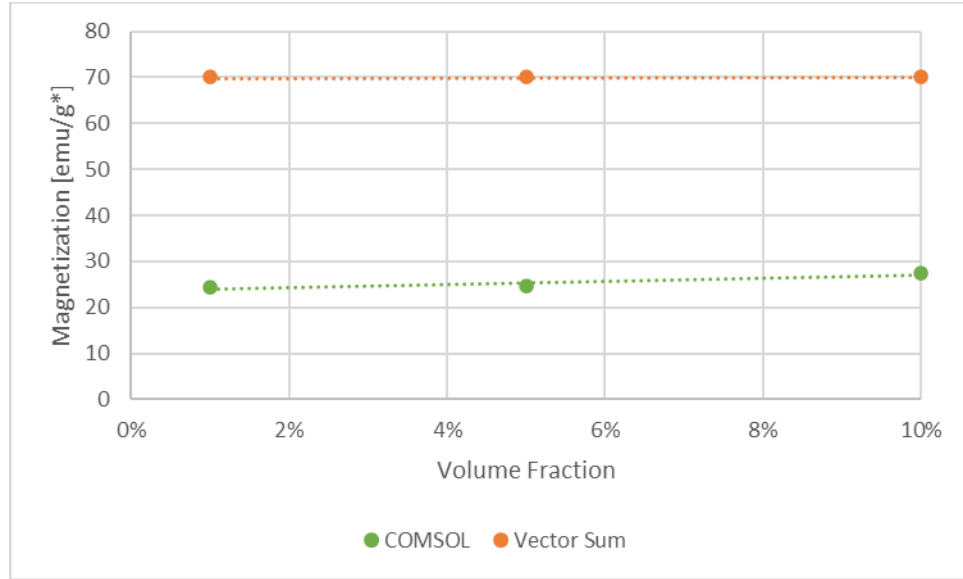


Figure 2-10: Mass magnetizations for Case 3 (Magnetic Field only, 100 particle RVE's) for magnetic vector sum versus full field magnetization response illustrating the effect of the particle interactions accounted for in the COMSOL model but not in the vector sum method

### 2.3 Artificial Intelligence Methods to Predict Conductivity

In this section an alternate approach to COMSOL for computational homogenization for conductivity values is presented. The conductivity,  $\sigma$ , corresponds to the real component of the complex current discussed in Section 2.1.2.2, simplifying the expression to  $J = \sigma E$ . Discrete particle distributions from particle dynamics simulations are used to generate a graph network that is processed using AI search techniques to determine the conductivity of the RVE.

In order to quantify the conductivity of the RVE, each percolating path is treated as a network of resistors in series whose resistances are based on the constituents material properties and the lengths of the paths through them. All percolating paths within the RVE are then treated as resistors in parallel to determine the effective resistance of the composite. By finding all of the conductive pathways through an RVE, a network of resistors can be created to calculate the effective resistance and characterize the conductivity of the composite (see Figure 2-11).



These effective property values can then be compared to experimental data from the literature to validate the model and predict values for future experiments.

This approach to modeling conductivity requires the Path Finding (Section 2.3.1) through the network of nodes, determination of the Cut Off Distance (Section 2.3.2) limiting which nodes may conduct, and calculation of the Effective Resistance (Section 2.3.3), which are discussed in more detail in the following sections.

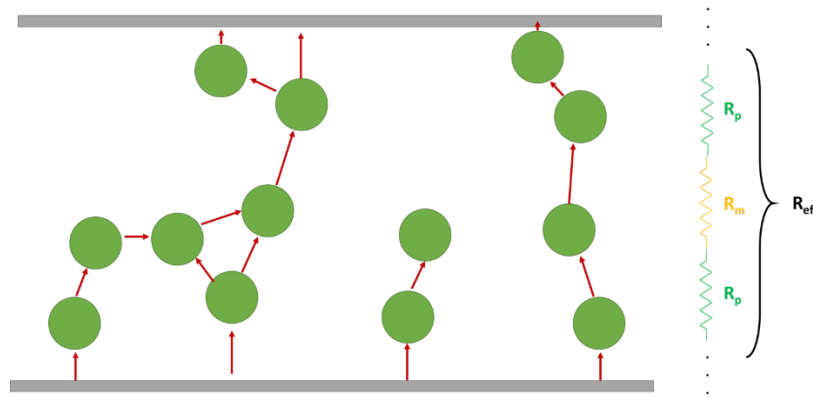


Figure 2-11: Diagram showing potential pathways between particles that can result in percolating paths used for calculating effective resistance of the RVE from connecting paths by treating the elements of one path as resistors in series and then combining all paths in parallel

### 2.3.1 Path Finding

In this section, methods of determining percolation paths are explored. Efficient search of the composite volume for conductive pathways, using AI search algorithms, improves upon previous random walk or exhaustive search methods. The proposed search algorithm, all paths search, uses a modified version of depth first search which is an AI search algorithm for efficiently traversing graphs. The all paths search will not only determine if conductive pathways exist but be used with homogenization techniques for electrical conduction presented in Section 2.3.3 to calculate the effective resistance of the composite. The connected network the all paths search

algorithm builds is the foundation for the framework proposed to determine conductivity for highly conductive or high aspect ratio fillers.

Path finding is a classic AI graph search problem, that can be employed to find conductive networks of particles in a composite. There are different approaches, such as depth first and A\* (a heuristic driven search) algorithms, that can be implemented for different types of graphs to efficiently search the graph for the goal nodes [93]. In the case of finding all conductive pathways through an RVE, it is of no benefit to search the graph for the purpose of exploration, such as breadth first search would do. Therefore a depth first search is implemented to find all possible paths through the RVE. The algorithm is modified to limit path lengths, which was found to be useful during testing, and to not allow looping back to a filler particle that has already been added to the current path.

A test case model, an example problem with a known solution, is shown in Figure 2-12, where a graph is shown with nodes connected in either green (conducting to ground) or black (not connected to ground therefore nonconducting) representing pathways between the start (source) and finish (ground) nodes. The graphs represented in Figure 2-12 are shown analyzed using two methods: a) spanning tree (with connected paths in green) and b) the resulting paths (connecting paths again in green) using all paths search. In Figure 2-12 a, the spanning tree, which is meant to represent resistive pathways with respect to our percolating network, is shown to have several ‘dead’ branches, branches with no terminus at the finish (or ground) node. By contrast the graph shown in Figure 2-12 b shows only paths that connect the two terminals (start and finish) without any ‘dead’ branches. Additionally it finds pathways that connect the start and finish that the spanning tree did not. Consequently, the all paths search method is chosen to

analyze conductive pathways in an RVE. While this example is shown for a finite path through a single RVE, future refinement will include periodic boundary conditions for increased accuracy.

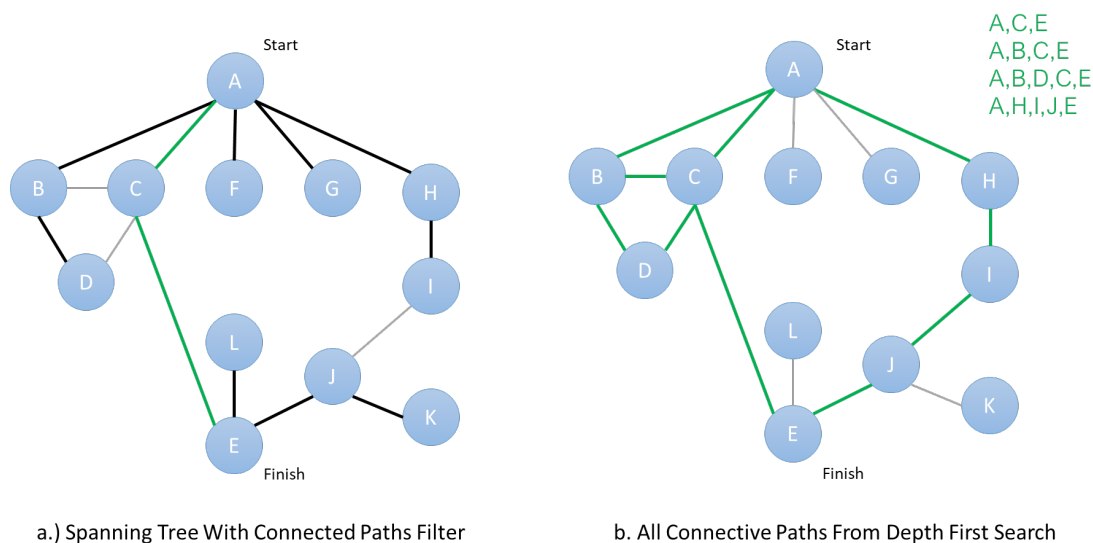


Figure 2-12: a.) Spanning tree representation of a graph b.) Graph showing only paths that span the start and finish nodes

In the “all paths” method, each node that is visited in the search is expanded which returns a set of near-by nodes, as determined by the position in space, the size of the particles, and the maximum electron tunneling distance separating them. Each of these neighbors is then visited during the search. A detailed explanation of the algorithm and code is presented in Appendix C. This type of search builds the graph during the search, only visiting the nodes that are close enough to be a part of a conductive path. This saves in computational costs as compared to constructing a graph and then searching for conductive pathways. When the terminal node is reached the path that was travelled is saved as a series of resistors which allows the effective resistance to be calculated. This algorithm forms a full connected graph, excluding loops, and finds paths that would not have been considered in a minimum spanning tree algorithm.

### 2.3.2 Cutoff Distance

The all paths search method requires use of a cutoff distance that determines suitable neighbors for every filler particle that is computed from the distance between particles and the matrix properties. Details of the cutoff method used herein are given in this section and used to find conductive pathways through the composite. The tunneling resistance through the matrix between two neighboring particles can be described *via* the current density [137]

$$J = \left( 3 * \frac{\sqrt{2m\lambda}}{2d} \right) \left( \frac{e}{h} \right) 2V e^{-4\pi d h \sqrt{2m\lambda}} \quad (2-44)$$

where  $V$  is the voltage,  $e$  is a quantum of electricity ( $1.6 \times 10^{-19}$  C),  $\lambda$  is the barrier height of the matrix [eV],  $h$  is Planck's constant,  $m$  is the mass of an electron ( $9.109 \times 10^{-31}$  kg), and  $d$  is the tunneling distance (distance between conductors). The barrier height is related to the amount of energy required to move an electron through an insulator. Oskouyi et. al. plotted the results of  $\lambda=0.5, 1.0, 1.5$  eV, which showed that an increase in the barrier height decreased the conductivity quickly for a tunneling distance between 0.25 and 0.5 nm, after which the values all converge to 0. This was used to select an initial pass of  $\lambda=1$ , which will be explored further in the future refinement of this algorithm since it may need to be experimentally determined [138]. Equation (2-44) shows that as the distance between particles increases, the current density (which is inversely proportional to the resistance) exponentially decreases. The current density can be plotted vs separation distance between filler particles to select an appropriate cutoff distance. Large cutoff distances will increase the search time, as more and more permutations are possible with more particles included as neighbors for each step in the search. The proper distance for using equation (2-44) is yet to be determined in the literature [139]. For this work a point at

which the increase of cutoff distance is negligible to the effective conductivity will be chosen.

The cutoff distance can also be tuned with experimental results to calibrate the model.

### **2.3.3 Effective Resistance**

The effective resistance of the composite is calculated by constructing a network of resistors representative of the paths found by the all paths search algorithm to connect the filler particles.

This resistor network takes into consideration the contributions from the matrix, the filler particles, and the relative distribution of the filler within the matrix. Using the differential equations previously discussed, and detailed in Appendix B, built into the MATLAB system of equations solvers in Simulink, the effective resistance can be determined for this network. This homogenization technique in conjunction with the all path search algorithm presented in Section 2.3.1 can determine the conductivity for conductive particle composites, even in the cases where the conductivity of the filler is problematically large for traditional FEA techniques.

The resistance of the material is considered to be comprised of the resistance of conducting paths through the matrix, as discussed in Section 2.3, and additional contributions for conduction through the filler particles. The effective resistance through the matrix can be calculated, as discussed for the cutoff distance. The resistance of each particle is a more complex solution that depends on the position of the surrounding particles. The closest points on the surface of the neighboring particles and the corresponding line segment through the middle particle is taken as the effective length through the particle and idealized as a short segment of wire to calculate the conductivity of this section of the total path as is illustrated in Figure 2-13.

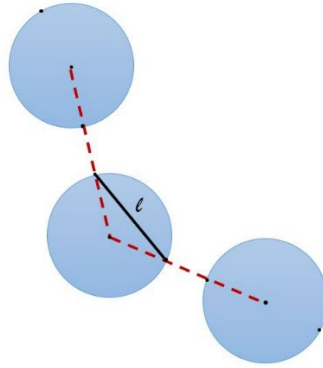


Figure 2-13: Idealized conductive pathlength through a particle based on the position of neighboring particles

The resistances through the matrix and through the filler particles are inputs to the resistor network that use the pathways found in Section 2.3.1. Each path is comprised of several particles and the matrix material between them. Iteration through this chain of components and calculation of the resistance of each segment is used to reconstruct the representative resistor model for the composite. A MATLAB script is used to populate a Simulink model as shown in Figure 2-14. This model is then solved in Simulink by applying a voltage across the top and bottom node which are placeholder components to represent electrical leads in an experimental setup. Figure 2-15 shows a simplified representation of this to illustrate the boundary conditions and the post processing of the model. The Simulink solution determines the current through the system for a known voltage. This can be used with the basic relationship between current, voltage, and resistance to determine the effective resistance of the composite.

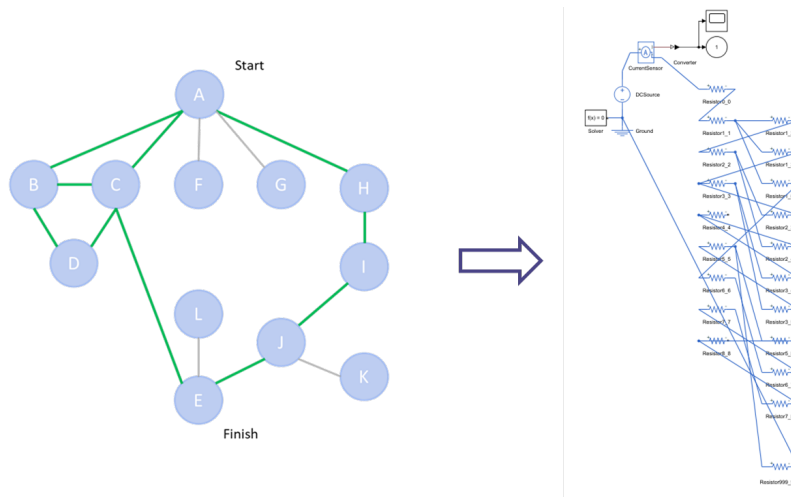


Figure 2-14: Example of test case and Simulink model of resistors for determining effective composite resistance

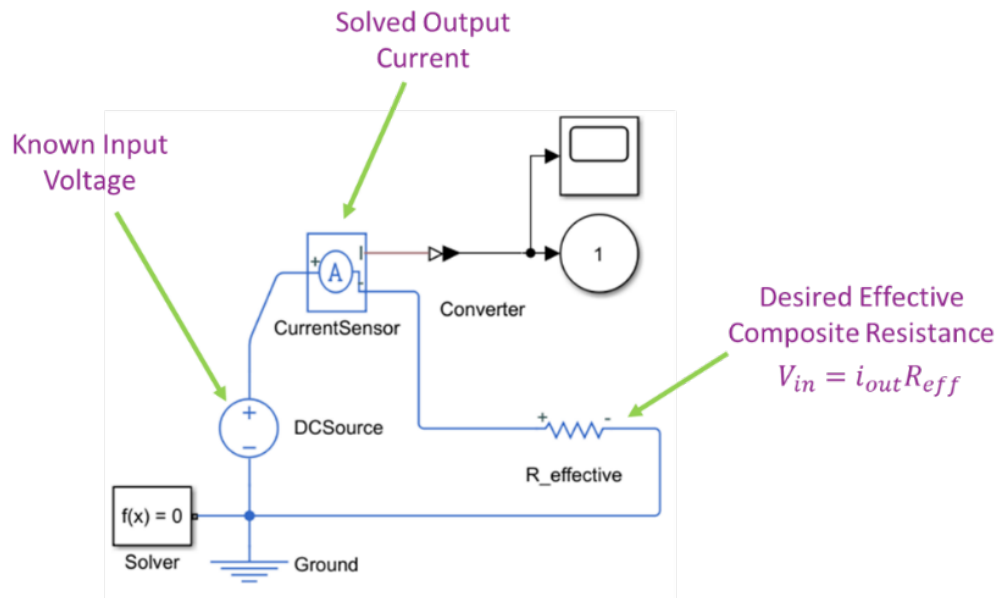


Figure 2-15: Simulink model showing boundary conditions used to determine effective resistance of composite where the composite resistance is shown as  $R_{effective}$  for simplicity

An alternate approach to for computational homogenization for conductivity values was presented, using discrete particle distributions to generate a graph network of resistors to determine the conductivity of the RVE. This method uses AI search techniques to efficiently search for conductive pathways and a representative resistor network to solve for the effective resistance (or conductivity) of the particle filled composite. The framework presented here

expands on the capabilities of computational homogenization to allow conductivity to be determined for composites with highly conductive filler material, previously challenging and computationally expensive with traditional FEA.

## **2.2 Estimating Composite Properties using Effective Medium Theory**

### **2.2.1 Elastic Domain Models**

In order to support rapid advancement in the development of new composite materials with enhanced properties, analytic models of composites with various filler types were developed to quickly predict the performance of such composites in a closed-form solution [140]. These models often build off each other and more interestingly can be related to one another when certain simplifying assumptions are made [23]. For example, simple bounds are established with the early Voigt and Reuss models, which later, self-consistent models fall within. The Halpin-Tsai model gives an estimate for dilute composites with uniformly distributed filler. Eshelby and Mori-Tanaka also assume a dilute composite but address the average strain with different assumptions. The Eshelby model assumes the average strain of the volume to be equal to the average applied strain whereas the Mori-Tanaka model averages the strains for all of the phases in the volume. The double inclusion model builds off of the former two models to improve on the assumptions of average strain and account for particle-particle interactions and becomes the basis for the homogenization schemes of Ponte Castaneda and Willis [23]. In this section the Voigt and Reuss, Halpin Tsai, Eshelby, Mori-Tanaka, and Hashin-Shtrikman, and Double Inclusion Models are presented in further detail showing the assumptions, advantages, and shortcomings of these methods for estimating the effective modulus of composites. The Halpin Tsai model, accounting for shape and orientation of filler particles, will be used to bound the simulation results in Chapter 3 for verification.



### Voigt and Reuss

The Voigt and Reuss model, also referred to as the rule of mixtures, make simplifying assumptions as illustrated in Figure 2-16 on the strain and stress for the RVE and become the upper and lower bounds on modulus, respectively. Using energy methods, the bounds are determined by minimizing the potential and complementary energies for a volume,  $V$ , as defined in equations (2-45) and (2-46), respectively [59].

$$\Pi[u_i] = \frac{1}{2} \int_V C_{ijkl} u_{k,l} u_{i,j} dv - \int_S t_i^0 u_i ds \quad (2-45)$$

$$\Pi[\sigma_{ij}] = \frac{1}{2} \int_V S_{ijkl} \sigma_{kl} \sigma_{ij} dv - \int_S u_i^0 \sigma_{ij} n_j ds \quad (2-46)$$

$C_{ijkl}$  is the stiffness tensor,  $S_{ijkl}$  is the compliance tensor,  $t_i^0$  and  $u_i^0$  are a prescribed traction and displacement boundary conditions, respectively, on the surface  $S$ ,  $\sigma$  is stress, and  $n_j$  is the unit vector normal to the surface. The boundary displacement is defined in equation (2-47), where  $\overline{\varepsilon_{ij}}$  is the average strain on the surface and by the average strain theorem, the average strain in the volume.

$$u_i|_S = \overline{\varepsilon_{ij}} x_j \quad (2-47)$$

To further evaluate the volume the following equilibrium, compatibility, and constitutive laws are used.

$$\sigma_{ij,j} = 0 \quad (2-48)$$

$$\widehat{\varepsilon_{ij}} = \frac{1}{2} (\widehat{u_{i,j}} + \widehat{u_{j,i}}) \quad (2-49)$$

$$\varepsilon_{ij} = S_{ijkl} \sigma_{kl} \quad (2-50)$$

With no prescribed traction the potential and complementary energy expressions reduce to

$$\Pi[u_i] = \bar{\epsilon} \bar{\mathcal{C}}^V \bar{\epsilon} \quad (2-51)$$

$$\Pi[\sigma_{ij}] = 2\bar{\epsilon} \hat{\sigma} - \hat{\sigma} \bar{\mathcal{S}}^R \hat{\sigma} = \bar{\epsilon} \bar{\mathcal{C}}^R \bar{\epsilon} \quad (2-52)$$

$\mathcal{C}^V$  and  $\mathcal{S}^R$  are the Voigt stiffness tensor and Reuss compliance tensor, respectively. The average of these stiffness and compliance tensors is defined as the volume average of the integral over the volume.

$$\bar{\mathcal{C}}^R = [\bar{\mathcal{S}}^R]^{-1} = \left[ \frac{1}{V} \int_V S dv \right]^{-1} \quad (2-53)$$

$$\bar{\mathcal{C}}^V = \frac{1}{V} \int_V C dv \quad (2-54)$$

The expressions from (2-51) and (2-52) are set as the upper and lower bounds, respectively, for the effective compliance of the composite.

$$\bar{\epsilon} \bar{\mathcal{C}}^R \bar{\epsilon} \leq \bar{\epsilon} \bar{\mathcal{C}} \bar{\epsilon} \leq \bar{\epsilon} \bar{\mathcal{C}}^V \bar{\epsilon} \quad (2-55)$$

$$\bar{\mathcal{C}}^R \leq \bar{\mathcal{C}} \leq \bar{\mathcal{C}}^V \quad (2-56)$$

For a composite with volume fraction of filler,  $v_f$ , and matrix,  $v_m$ , integrals presented in (2-53) and (2-54) are then evaluated for the average composite stiffness.

$$\bar{\mathcal{C}}^V = v_f C^f + v_m C^m \quad (2-57)$$

$$\bar{\mathcal{C}}^R = [v_f S^f + v_m S^m]^{-1} \quad (2-58)$$

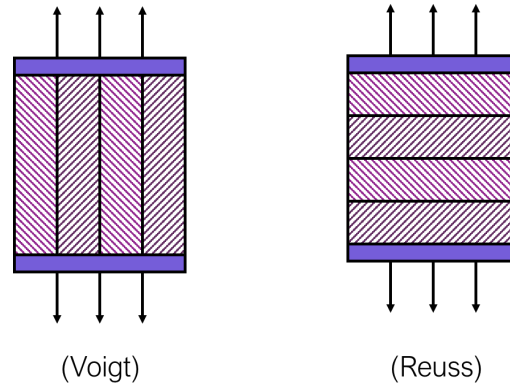


Figure 2-16: Diagram of strain and stress assumptions in Voigt and Reuss models respectively.

The Voigt and Reuss model define the limits of effective modulus of the composite and serve as a first check on any system. Higher fidelity models, such as those of Eshelby, Mori-Tanaka, and Halpin Tsai, can offer further insight into properties based on additional information regarding the filler.

### Halpin Tsai

Compared to Voigt and Reuss, the Halpin Tsai model has additional consideration for filler shape and orientation in an adjustment parameter. This model is a form of the Hill's self-consistent model with simplifications to for engineering analysis of composite materials. The model begins by assuming a cylinder in a matrix with a radial strain applied and a stress in the axial direction to maintain zero strain in the axial direction (plane-strain). It then uses generalized Hooke's law to solve the constituent equations for the volume, shown in Hill's notation in equations (2-59) and (2-60) [55], [57].

$$E_2 = E_m \left( \frac{1 + \xi \eta v_f}{1 - \eta v_f} \right) \quad (2-59)$$

$$\eta = \frac{\frac{E_f}{E_m} - 1}{\frac{E_f}{E_m} + \xi} \quad (2-60)$$

$E_m$  and  $E_f$  are the matrix and filler modulus respectively,  $v_f$  is the volume fraction of the filler, and  $\xi$  is the reinforcement parameter. There have been several investigations into the best reinforcement parameter, which is dependent on the geometry of the filler [60]. On the two extremes,  $\xi = 0$  and  $\xi = \infty$ , the Reuss and Voigt bounds (discussed in Section 1.3) are recovered, respectively [55], [60]. Halpin discusses further variations on the reinforcement factor to account for inclusion geometry, particle packing, and loading conditions. Short, oriented fibers and non-spherical particles all have an aspect ratio term accounted for in the reinforcement factor equation. In the case of ellipsoidal particles the equation for the reinforcement parameter can be written as

$$\xi = 2(a/b) + 40v_f^{10} \quad (2-61)$$

where  $a$  and  $b$  are the major and minor axes of the ellipsoid. Note that the orientation is accounted for in this term in that  $a$  is axially aligned in the volume. Other examples of filler types and corresponding reinforcement factor equations are tabulated in **Table 1**, where  $l$ ,  $w$ , and  $t$  are the length (in the axial 1 direction for fibers or ribbons), width (in the transverse 2 direction), and thickness (in the 3 direction) of the particle, respectively [57]. For the case of oriented ribbons the effect of the aspect ratio is plotted in Figure 2-17. The reinforcement factor's account for shape and orientation of the filler material sets a tighter bound than the Voigt and Reuss models previously presented, and are useful for verifying simulation results. However

the model cannot account for structuring effects on composite properties and is therefore limited to providing bounds for the composites studied in this work.

**Table 1: Reinforcement factor equations for various filler types with  $0 \leq v_f \leq 1$**

Particulate Composites	$\xi_E \cong 2 + 40v_f^{10}$ $\xi_G \cong 1 + 40v_f^{10}$
Voids, foam, or porous solid	2
Oriented continuous fibers	$\xi_E \cong 2 + 40v_f^{10}$ $\xi_G \cong \frac{1}{4 - 3v_m}$
Oriented discontinuous fibers	$\xi_E = 2(l/d) + 40v_f^{10}$ $\xi_G \cong 1 + 40v_f^{10}$
Oriented continuous ribbon or lamellar-shaped reinforcements	$\xi_E \cong 2(w/t) + 40v_f^{10}$ $\xi_G \cong (w/t)^{1.73} + 40v_f^{10}$
Oriented discontinuous ribbon (flake)	$\xi_{E_{11}} = 2(l/t) + 40v_f^{10}$ $\xi_{E_{22}} = 2(w/t) + 40v_f^{10}$

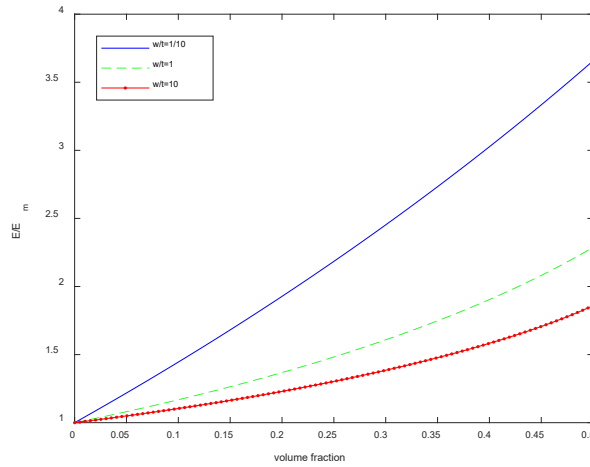


Figure 2-17: Effect of aspect ratio,  $w/t$ , on effective modulus, normalized to matrix modulus for the case of ellipsoidal filler. Results show that as  $w/t$  increases,  $E/E_m$  decreases, and the effective modulus become less dependent on volume fraction.

### Eshelby

For the Eshelby method, the inclusions are considered to be far enough apart that they do not interact with each other and can be treated as single inclusions in a volume of matrix. In other

words, this method assumes a dilute volume fraction of filler material spread uniformly. For this case the average strain is defined as

$$\bar{\epsilon} = \frac{1}{V} \int_V \epsilon dV = \epsilon^0 \quad (2-62)$$

where  $\epsilon^0$  is the applied strain at the boundary,  $\epsilon$  is the strain at a specific position within the volume  $V$ , and  $\bar{\epsilon}$  is the average strain in the RVE. The Eshelby model is a base for models such as the Mori-Tanaka model.

### Mori Tanaka

Although the Mori-Tanaka method also assumes the inhomogeneities are isolated, it accounts for the average strain of all of the phases in the average strain of the volume.

$$\bar{\epsilon} = \frac{1}{V} \int_V \epsilon dV = \sum_{r=0}^N \bar{\epsilon}_r \quad (2-63)$$

$$\bar{\sigma} = \frac{1}{V} \int_V \sigma dV = \sum_{r=0}^N \bar{\sigma}_r \quad (2-64)$$

where  $\bar{\sigma}_r$  and  $\bar{\epsilon}_r$  are the stress and strain tensors averaged over the  $r^{\text{th}}$  inhomogeneity. This leads to the bulk and shear modulus of a composite with spherical inclusions defined as

$$\bar{K} = K_0 + \frac{c_1 K_0 (K_1 - K_0) (3K_0 + 4\mu_0)}{3K_0 + 4\mu_0 + 3(1 - c_1)(K_1 - K_0)} \quad (2-65)$$

$$\bar{\mu} = \mu_0 + \frac{5c_1 \mu_0 (\mu_1 - \mu_0) (3K_0 + 4\mu_0)}{5\mu_0 (3K_0 + 4\mu_0) + 6(1 - c_1)(\mu_1 - \mu_0)(K_0 + 2\mu_0)} \quad (2-66)$$

where  $K_0$ ,  $K_1$ ,  $\mu_0$ , and  $\mu_1$  are the bulk and shear modulus of the matrix and the inclusions, respectively [59].

The Halpin Tsai, Mori Tanaka, and Voigt and Reuss bounds are shown plotted in Figure 2-18 (for ellipsoidal particles where applicable), showing the relationship of effective modulus to volume fraction for each model. Note that when the difference in modulus between the filler and matrix increases this difference in the models becomes amplified, which can be seen in Section 3.1, Figure 3-1.

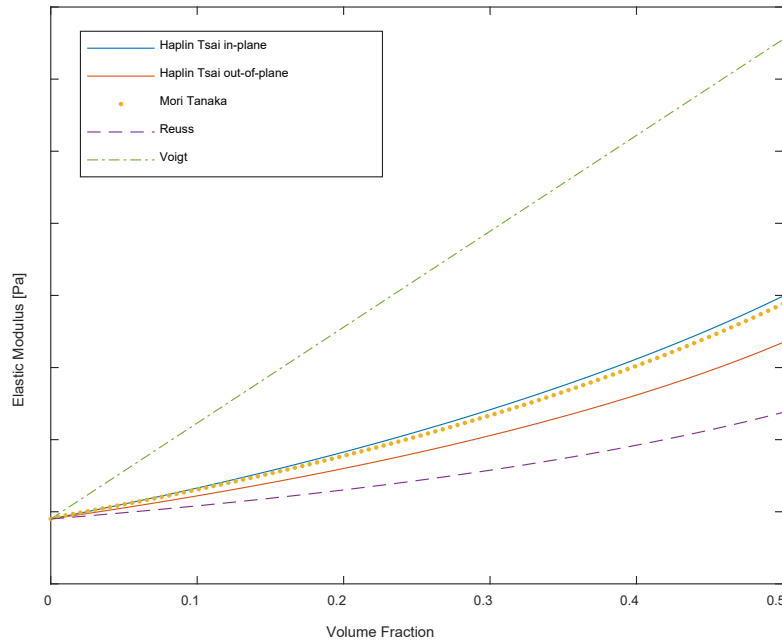


Figure 2-18: Effective medium models moduli versus volume fraction showing the relationship of effective modulus to volume fraction for Halpin Tsai and Mori Tanaka models and Voigt and Reuss bounds

### Hashin-Shtrikman Bounds

Where Voigt and Reuss assumed uniform admissible stress and strain fields, the Hashin-Shtrikman bounds develop a statically admissible stress polarization tensor,  $\boldsymbol{\tau}(\boldsymbol{\chi})$ , a constant

tensor on each inhomogeneity including the matrix. The stiffness tensor of the material is defined as

$$\mathbf{C}(\mathbf{x}) = \mathbf{C}^h + \mathbf{C}^p(\mathbf{x}) \quad (2-67)$$

where  $\mathbf{C}(\mathbf{x})$  is a function of position  $\mathbf{x}$ ,  $\mathbf{C}^h$  is the constant tensor describing a homogeneous material (the matrix without inclusions), and  $\mathbf{C}^p$  is the perturbation from the homogeneous material as a result of the heterogeneities (the difference between the matrix with and without inclusions). The corresponding stress field then becomes

$$\boldsymbol{\sigma}(\mathbf{x}) = \mathbf{C}^h \boldsymbol{\varepsilon}(\mathbf{x}) + \mathbf{C}^p(\mathbf{x}) \boldsymbol{\varepsilon}(\mathbf{x}) \quad (2-68)$$

where the strain field is

$$\boldsymbol{\varepsilon}(\mathbf{x}) = \frac{1}{2} \left( \frac{\partial u_i(\mathbf{x})}{\partial x_j} + \frac{\partial u_j(\mathbf{x})}{\partial x_i} \right) \quad (2-69)$$

where  $u(\mathbf{x})$  is the displacement field. The stress polarization tensor is taken from equation (2-69)

$$\boldsymbol{\tau}(\mathbf{x}) = \mathbf{C}^p(\mathbf{x}) \boldsymbol{\varepsilon}(\mathbf{x}) \quad (2-70)$$

For the bounds it is assumed that the stress polarization tensor is a uniform piecewise function.

This assumes that the stress is a piecewise field, constant on each inhomogeneity,  $r$ , in the domain  $\Omega_r$ .

$$\hat{\boldsymbol{\tau}}(\mathbf{x})|_{\mathbf{x} \in \Omega_r} = \hat{\boldsymbol{\tau}}_r \quad r = 0, 1, 2, \dots, N \quad (2-71)$$

The solution, and algebraic system of equations, of the constant tensor  $\boldsymbol{\tau}_r$  is written as

$$\boldsymbol{\tau}_r = \mathbf{R}_r \bar{\boldsymbol{\varepsilon}} \quad (2-72)$$

which can be used to define the composite effective stiffness



$$\tilde{\mathbf{C}} = \mathbf{C}^h + \sum_{r=0}^N c_r \mathbf{R}_r \quad (2-73)$$

where  $c_r$  is the volume fraction for the  $r^{\text{th}}$  inhomogeneity. The lower bound is found by substituting  $\mathbf{C}^h = \mathbf{C}_0$  and the upper bound by substituting  $\mathbf{C}^h = \mathbf{C}_1$  if  $\mathbf{C}_1 > \mathbf{C}_0$  where 0 and 1 are the matrix and inclusion phase, respectively. These bounds are more restrictive than the Voigt and Reuss bounds but requires the evaluation of a double integral over all inhomogeneities to find  $\bar{\boldsymbol{\varepsilon}}$ , which can be simplified with further assumptions [59].

The models presented deal with dilute systems and do not account for particle-particle interactions but have a relatively easy to use form.

#### Double Inclusion Model

The double inclusion model was proposed to address the interactions between particles by extending the existing Mori-Tanaka and Eshelby models. Consequently, the double inclusion model can be reduced to the Eshelby and Mori-Tanaka models with specific conditions, namely setting the elasticity of the double-cell is set to equal that of the matrix material and excluding effect of other particles on the strain concentration tensor. The focus of the double inclusion model is to more accurately describe the average stress and strain fields, which had been previously expressed as constant throughout the volume [140]. The model considers ellipsoidal inclusions with common aspect ratios contained within the double-cell which contains both the matrix and inclusion phase. This double-cell is then embedded in an infinitely large medium under a uniform strain ( $\boldsymbol{\varepsilon}^\infty$ ) illustrated in Figure 2-19.

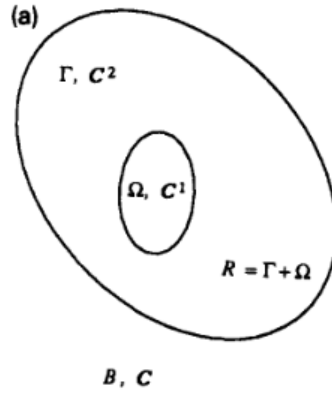


Figure 2-19: Double inclusion model diagram, inclusion phase  $\Omega$  with elasticity  $C^1$  embedded in the double-cell  $R = \Gamma + \Omega$  with elasticity  $C^2$  in infinite medium  $B$  with elasticity  $C$  [140]

The double cells may be aligned or randomly oriented for a given composite. As in typical elasticity problems, we assume a symmetric stress tensor that satisfies conservation of static equilibrium as before,  $\nabla \cdot \sigma = 0$ . Additionally, on the outermost boundary the farfield strain  $\epsilon^\infty$  is applied, with corresponding  $\sigma^\infty$ . On the inner boundary between matrix and inclusions the boundary conditions are not known such that a supplementary problem of a homogeneous volume is used, from which the average strain on the inclusions is estimated. Note that the average strain, 2-56, is a complex function of particle shape and orientation through the Eshelby tensor, but still does not account for interaction of multiple inclusions. The average stress and strain can then be found over all of the inclusions (see equations (2-74) and (2-75), respectively). Hooke's law is then used to describe the effective modulus of the composite (see equation (2-76)).

$$\bar{\sigma} = \langle L + L[(S^V - I)(A_2 + c_1 A_1)] \rangle \epsilon^\infty \quad (2-74)$$

$$\bar{\epsilon} = \langle I + S^V(A_2 + c_1 A_1) \rangle \epsilon^\infty \quad (2-75)$$

$$L_c = \langle L + L[(S^V - I)(A_2 + c_1 A_1)] \rangle \langle I + S^V(A_2 + c_1 A_1) \rangle^{-1} \quad (2-76)$$

where  $\mathbf{L}$  is the modulus tensor of the given phase,  $\mathbf{L}_c$  is the effective modulus tensor of the composite,  $\mathbf{A}_2$  and  $(\mathbf{A}_1 + \mathbf{A}_2)$  are the eigen strain concentration tensors of the matrix and inclusion, respectively;  $\mathbf{I}$  is the unit tensor,  $\mathbf{S}^V$  is the fourth rank Eshelby tensor of the double cell, and  $c_1$  is the volume fraction [23]. The Eshelby tensor contains information relating how the shape and orientation of the filler, with respect to the local strain field, affects the local strain field and thereby the local compliance. For an ellipsoidal inclusion [59] the Eshelby tensor is defined as

$$\begin{aligned}
 S_{1111} &= \frac{3}{8\pi(1-\nu)} a_1^2 I_{11} + \frac{1-2\nu}{8\pi(1-\nu)} I_1 \\
 S_{1122} &= \frac{1}{8\pi(1-\nu)} a_2^2 I_{12} - \frac{1-2\nu}{8\pi(1-\nu)} I_1 \\
 S_{1133} &= \frac{1}{8\pi(1-\nu)} a_3^2 I_{13} - \frac{1-2\nu}{8\pi(1-\nu)} I_1 \\
 S_{1212} &= \frac{a_1^2 + a_2^2}{16\pi(1-\nu)} I_{12} + \frac{1-2\nu}{16\pi(1-\nu)} (I_1 + I_2)
 \end{aligned} \tag{2-77}$$

where  $a_1, a_2, a_3$  are the lengths of the axes of the ellipsoid and  $a_1 > a_2 > a_3$ , the constants  $I_1, I_2, I_{11}$ , and  $I_{22}$  are functions of  $a_1, a_2, a_3$  and elliptical integrals over the volume of the ellipsoid.

$$\begin{aligned}
 I_1 + I_2 + I_3 &= 4\pi \\
 3I_{11} + I_{12} + I_{13} &= 4\pi/a_1^2 \\
 3a_1^2 I_{11} + a_2^2 I_{12} + a_3^2 I_{13} &= 3I_1 \\
 I_{12} &= (I_2 - I_1)/(a_1^2 - a_2^2) \\
 I_1 &= \frac{4\pi a_1 a_2 a_3}{(a_1^2 - a_2^2)(a_1^2 - a_3^2)^{1/2}} \left\{ \frac{a_2(a_1^2 - a_3^2)^{1/2}}{a_1 a_3} - E(\theta, k) \right\} \\
 I_3 &= \frac{4\pi a_1 a_2 a_3}{(a_2^2 - a_3^2)(a_1^2 - a_3^2)^{1/2}} \{F(\theta, k) - E(\theta, k)\}
 \end{aligned}$$

$$\begin{aligned}
F(\theta, k) &= \int_0^\theta \frac{dw}{(1 - k^2 \sin^2 w)^{1/2}} \\
E(\theta, k) &= \int_0^\theta (1 - k^2 \sin^2 w)^{1/2} dw \\
\theta &= \sin^{-1} \left( 1 - \frac{a_3^2}{a_1^2} \right)^{1/2} \\
k &= \left\{ \frac{a_1^2 - a_2^2}{a_1^2 - a_3^2} \right\}^{1/2}
\end{aligned} \tag{2-78}$$

The double inclusion model has been used to study the effective properties of composites by adjusting the surrounding phase, R, to tune the model to experimental results where it was shown that the results are bounded by a minimum of the Mori-Tanaka model [141], [142].

The models presented in this section for the elastic domain vary in complexity and assumptions made, but all assume a uniform distribution of particles and do not have the ability to account for long range order.

### 2.2.2 Dielectric and Magnetic Domain Models

Similar models were laid out for dielectric and magnetic domains. A comparison of dielectric effective medium theory was laid out by Bánhegyi [67], summarized below for the Maxwell Garnett and Bruggeman models, which details the mean field and integration approaches to permittivity approximations of composites with various filler shapes. The following equations are presented for the dielectric domain but have an equivalent form in the magnetic domain.

#### Maxwell Garnett (Mean Field) Approximation

One method for determining the effective permittivity of a heterogenous material is by the definition of the electric displacement field using volume averaging.

$$\langle D \rangle = \varepsilon_0 \bar{\varepsilon} \langle E \rangle \quad (2-79)$$

where  $E$  is the electric field,  $D$  is the displacement,  $\varepsilon_0$  is the vacuum permittivity,  $\bar{\varepsilon}$  is the effective permittivity of the heterogeneous system. This leads to the general expression for the effective permittivity of the heterogeneous system

$$\bar{\varepsilon} = \varepsilon_1 + (\varepsilon_2 - \varepsilon_1) v_2 \frac{\langle E_2 \rangle}{\langle E \rangle} \quad (2-80)$$

where  $\langle E_2 \rangle$  and  $\langle E \rangle$  are the field averages over the volume of the 2<sup>nd</sup> component and whole volume, respectively,  $\varepsilon_1$  and  $\varepsilon_2$  are the relative permittivity of the first and second phase (matrix and filler), respectively, and  $v_2$  is the volume fraction of the second phase. The ratio of the electric fields can be calculated exactly for dilute systems and approximated at higher volume fractions. This leads to the formula for fully aligned (2-81) and randomly (2-82) oriented (dilute) inclusions

$$\bar{\varepsilon} = \varepsilon_1 + (\varepsilon_2 - \varepsilon_1) v_2 \frac{\varepsilon_1}{\varepsilon_1 + (\varepsilon_2 - \varepsilon_1) A_k} \quad (2-81)$$

$$\bar{\varepsilon} = \varepsilon_1 + (\varepsilon_2 - \varepsilon_1) \frac{v_2}{3} \sum_{k=1}^3 \frac{\varepsilon_1}{\varepsilon_1 + (\varepsilon_2 - \varepsilon_1) A_k} \quad (2-82)$$

respectively, where  $A_k$  is the depolarization factor in the  $k$  direction (averaged for the randomly oriented case shown in equation (2-82) using the summation over all 3 directions). The depolarization factor is derived from the shape of the dielectric particle that defines the directional components of the depolarizing field within the volume of the particle. The depolarizing field is a volumetric response to the surface polarization of a dielectric particle in an applied field and works in opposition to the applied external field [143].

For higher volume fractions this becomes

$$\bar{\varepsilon} = \varepsilon_1 \frac{\varepsilon_1(1 - v_2)(1 - A_k) + \varepsilon_2(v_2 + A_k(1 - v_2))}{\varepsilon_1 + A_k(1 - v_2)(\varepsilon_2 - \varepsilon_1)} \quad (2-83)$$

$$\bar{\varepsilon} = \varepsilon_1 \frac{\varepsilon_1(1 - v_2) + \frac{\varepsilon_2 v_2}{3} \sum_{k=1}^3 \frac{\varepsilon_1}{\varepsilon_1 + (\varepsilon_2 - \varepsilon_1)A_k}}{1 - v_2 + \frac{v_2}{3} \sum_{k=1}^3 \frac{\varepsilon_1}{\varepsilon_1 + (\varepsilon_2 - \varepsilon_1)A_k}} \quad (2-84)$$

### Bruggeman Model

Another approach is to model the components of the two-phase mixture as ellipsoidal cells surrounded by an effective medium with the effective relative permittivity,  $\varepsilon_m$ . For dilute system  $\varepsilon_m$  can be replaced with  $\varepsilon_1$  and for non-dilute (higher volume fractions) with  $\bar{\varepsilon}$  itself. For the non-dilute randomly oriented system this results in

$$\bar{\varepsilon} = \varepsilon_1 + (\varepsilon_2 - \varepsilon_1)v_2 \frac{\bar{\varepsilon}}{\bar{\varepsilon} + (\varepsilon_2 - \bar{\varepsilon})A_k} \quad (2-85)$$

$$\bar{\varepsilon} = \varepsilon_1 + (\varepsilon_2 - \varepsilon_1) \frac{v_2}{3} \sum_{k=1}^3 \frac{\bar{\varepsilon}}{\bar{\varepsilon} + (\varepsilon_2 - \bar{\varepsilon})A_k} \quad (2-86)$$

Note that equations (2-85) and (2-86) need to be solved for  $\bar{\varepsilon}$ . This method was first proposed by Bruggeman and was later extended to other shapes by various authors. Bruggeman is also credited with the proposal for integral methods which assume dilute concentrations of filler. For this method the following differential equation is integrated.

$$\frac{\bar{\varepsilon} + (\varepsilon_2 - \bar{\varepsilon})A_k}{\bar{\varepsilon}(\varepsilon_2 - \bar{\varepsilon})} d\bar{\varepsilon} = \frac{dv'_2}{1 - v'_2} \quad (2-87)$$

The results of which is the classic Bruggeman approximation for spheres with  $A_k = 1/3$ .

$$\left[ \frac{\bar{\varepsilon} - \varepsilon_2}{\varepsilon_1 - \varepsilon_2} \right] \left[ \frac{\varepsilon_1}{\bar{\varepsilon}} \right]^{A_k} = 1 - v_2 \quad (2-88)$$

For magnetic permeability the same approach can be used, by setting up the averages of the magnetic flux density,  $B$ , and the magnetic field,  $H$ .

$$\langle B \rangle = \mu_0 \bar{\mu} \langle H \rangle \quad (2-89)$$

where  $\mu_0$  is the permeability of free space, and  $\bar{\mu}$  is the effective permeability. This results in an analogous form to equation (2-80).

$$\bar{\mu} = \mu_1 + (\mu_2 - \mu_1) v_2 \frac{\langle H_2 \rangle}{\langle H \rangle} \quad (2-90)$$

$\langle H_2 \rangle$  and  $\langle H \rangle$  are the magnetic field averages over the volume of the 2<sup>nd</sup> component and whole volume, respectively,  $\mu_1$  and  $\mu_2$  are the relative permeability of the first and second phase (matrix and filler), respectively. Equivalently, the basic Bruggeman model (non-dilute ellipsoidal two-phase cell model) can be used with permeability in the following way [144].

$$\bar{\mu} = \mu_1 + (\mu_2 - \mu_1) v_2 \frac{\bar{\mu}}{\bar{\mu} + (\mu_2 - \bar{\mu}) A_k} \quad (2-91)$$

Note that equation (2-91) needs to be solved for  $\bar{\mu}$ . While these methods have proven adequate at predicting the effective properties of composites where self-organization exists among particles (e.g. ellipsoids with collinear semi-major/minor axes) they do not contain the needed fidelity to address hierarchical ordering (varying relative particle orientations within a chain of collinear particles, for example).

## 2.4 Multi-Objective Optimization

The objective of this work is to determine the maximum range of possible properties for a given constituent set as a function of processing parameters alone. In this way, in a manufacturing context, a single reservoir printing system may draw on a single constituent set, then process material at the point of deposition to access multi-material capabilities. This will be accomplished using a gradient based method, as the design space is not expected to be highly non-linear nor have multiple local minima.

### 2.4.1 Nested Optimization Structure

Two optimization steps will be nested to search the design space. The nested structure of this optimization framework will consist of a primary outer loop and a secondary inner loop. The primary loop will search the design space for the constituent set,  $\mathbf{x}_1$ , with the largest range of effective material properties,  $P_{eff}$ . The secondary loop finds the minimum and maximum property to compute the range of possible properties for a given constituent set, by varying the applied processing conditions,  $\mathbf{x}_2$ . The processing parameters are bounded by the upper and lower limits of the applied fields (magnetic fields  $H_L$  and  $H_U$ , and electric fields  $E_L$  and  $E_U$ ). The formal optimization statement for this framework is laid out in equation (2-92) and illustrated in Figure 2-20.

$$\begin{aligned}
 &\text{Minimize: } f_1(\mathbf{x}_1, \mathbf{x}_2) = - |f_{2a}^*(\mathbf{x}_2|\mathbf{x}_1) + f_{2b}^*(\mathbf{x}_2|\mathbf{x}_1)| \\
 &\text{Subject to: } x_{1L} \leq x_1 \leq x_{1U} \\
 &\text{Where:} \\
 &\quad \text{Minimize: } f_{2a}(\mathbf{x}_2|\mathbf{x}_1) = P_{eff} \\
 &\quad \text{and Minimize: } f_{2b}(\mathbf{x}_2|\mathbf{x}_1) = -P_{eff} \\
 &\quad \text{Subject to: } H_L \leq x_{2,1} \leq H_U \\
 &\quad \quad \quad E_L \leq x_{2,2} \leq E_U
 \end{aligned} \tag{2-92}$$

The objective functions in the secondary loop,  $f_{2a}$  (minimizing properties) and  $f_{2b}$  (maximizing properties) will use the computational homogenization presented in Section 2.1.2 to determine the effective properties of the composite for the given constituents (in this case volume fraction) and varying processing parameters. Note that the objective return from  $f_{2b}$  will be a negative value from the optimization, such that for computing the range this value is added to the minimum  $f_{2a}$ . The objective function in the primary loop,  $f_1$ , will use the maximum and



minimum properties returned from the secondary loop to determine the range. Note that because multiple properties are being evaluated that the property maxima and minima are vectors and that the evaluation of  $f_1$  uses the norm of the range. It is expected that the dielectric, elastic, and magnetic properties will have tradeoffs. For this reason, the properties are maximized and minimized individually such that the returned objective function values from  $f_{2a}$  and  $f_{2b}$  are not necessarily at the same applied fields. This is done to isolate the dependence of the individual properties on other properties when finding the extrema.

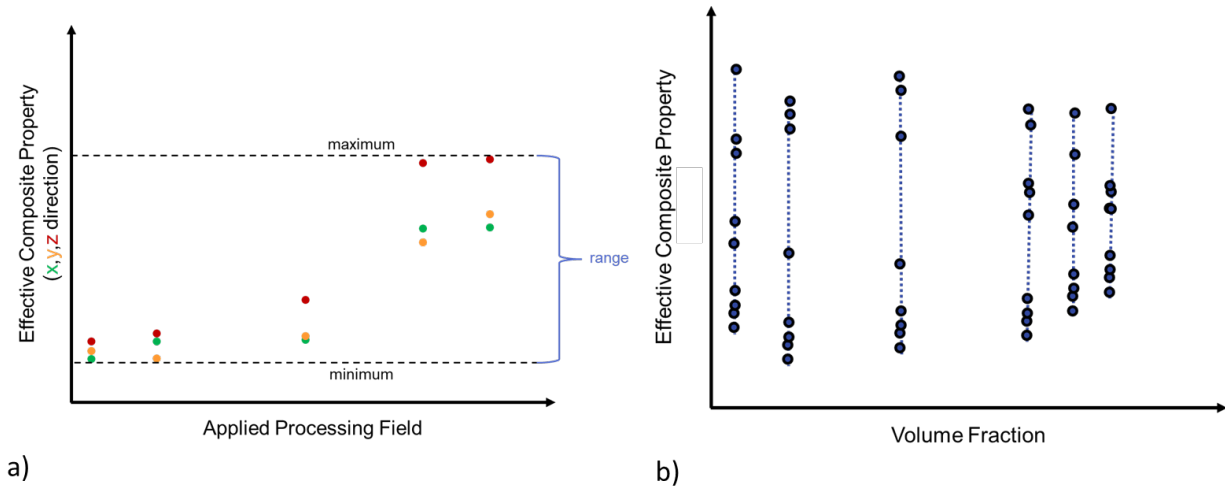


Figure 2-20: Representative results of the two optimization loops shown for purposes of illustration: a) searching for the minimum and maximum effective property by varying representative processing field only yields the range. Dotted lines highlight max and min processing condition locations. The max and min also yield the range of possible properties. b) searching for the maximum range of possible properties by varying the constituent set, e.g. volume fraction. For a given volume fraction, varying processing leads to high and low (and intermediate) values from which a range is found for a given volume fraction.

This process workflow is represented in Figure 2-21. Since the objective function  $f_1$  is dependent on the outputs of  $f_{2a}$  and  $f_{2b}$ , it is necessary that the two be nested in the manner shown. The innermost level shows the computational homogenization for specific constituents and processing fields at each stage of the optimization. The second level out shows the iteration

over the computational homogenization for changing processing parameters and a specified constituent set,  $\mathbf{x}_1$  to evaluate  $\mathbf{f}_{2a}$  and  $\mathbf{f}_{2b}$ . The outermost level is the primary loop which iterates over the property optimization for various constituent sets to find the set with the largest range of properties using the objective function  $f_1$ . The returned values from the optimization framework is the constituent set with the largest range of dielectric, elastic, and magnetic properties and the fields to achieve them.

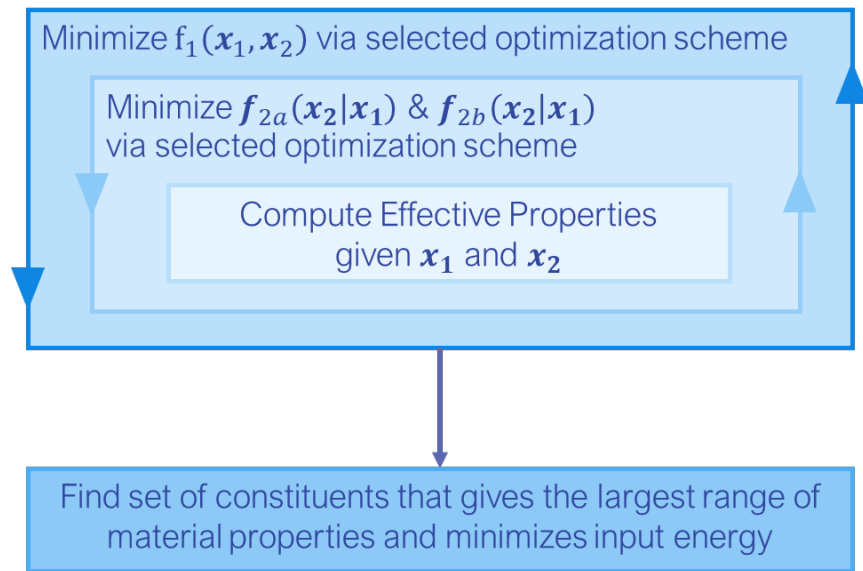


Figure 2-21: Workflow of nested optimization scheme showing the computational homogenization, property minimization and maximization, and the optimization of the constituent set to maximize the range of properties possible

A visual representation of a design space is shown in Figure 2-22, where it can be seen that for each constituent set choice,  $\mathbf{x}_1^{(1)}$ ,  $\mathbf{x}_1^{(2)}$ ,  $\mathbf{x}_1^{(3)}$ , etc. in a), there exists a range of properties as a function of  $\mathbf{x}_2$  in b), for which there exists a minimum and maximum effective material property. The dimension of this representation is reduced for visualization purposes.

The nested optimization scheme presented can be used to search this high order design space to find the set of constituents with the maximum and minimum material properties as to span the largest set of possible composite properties by varying processing parameters alone. The

fitness function, which has to consider various domains (elastic, dielectric, and magnetic), is discussed in the next section.

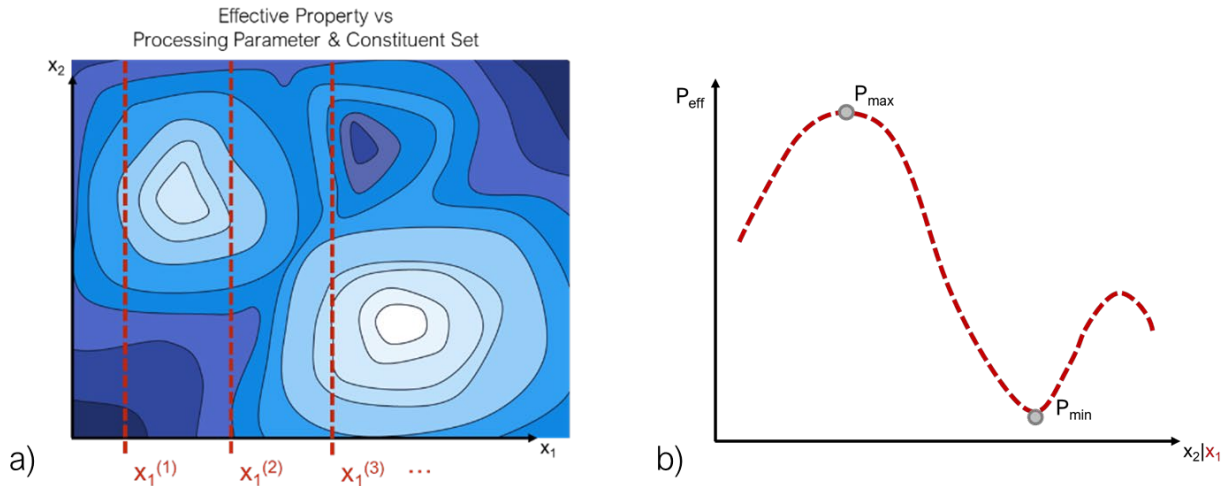


Figure 2-22: Representation of multi-objective optimization space a) design space contour showing selection of specific constituent sets  $x_1^{(1)}$ ,  $x_1^{(2)}$ ,  $x_1^{(3)}$  and b) plot of property evaluation vs processing parameters showing the minimum and maximum property for a single constituent set by varying processing parameters alone

## 2.4.2 Combined Fitness Function and Pareto Front

In order to consider multiple material properties simultaneously, a combined fitness, or objective, function must be defined. The distinctions of the nested combined functions are presented in this section in addition to the resulting design space search. It is expected that the materials properties in the elastic, magnetic, and dielectric domains will not respond the same way to the processing conditions and that there will be tradeoffs to consider. The resulting properties from all 3 homogenizations will be considered in the fitness function when determining the range of properties. Traditionally such a multi-objective fitness function can be handled with weighting the individual components. The resulting optimization scheme will generate a pareto front which can be used in specific design applications.

The nature of how to combine these separate objectives into one function is largely dependent on the application the optimization is being run for. For example in a specific design application the range of elastic modulus may be more important than the dielectric permittivity and would be weighted higher. Alternatively, the range of each individual property may be more important than their combined reach. This is a subtle but important distinction for the framework presented here and can be modified in the future to meet the needs of the specific application. It should be noted that because the properties can be, and in this case are, inversely affected by the structure, this interpretation is significant. For example, the elastic modulus would be higher when the particles are aligned in their major axis to form chains, this however, is less favorable for magnetization. Therefore the two maximums occur at different fields. If maximizing them simultaneously is important, this must be handled differently than maximizing each individual range. For the purpose of this work the “largest range of properties” will be interpreted to mean each individual property’s range.

A machine learning model, presented in further detail in Section 2.5, will be used to augment the design space search to reduce the computational expense of the computational homogenization, particularly of the elastic properties. The model will be used to predict the points for the optimization. The extrema found will then be simulated for with their corresponding processing parameters and the effective properties will be found using the computational homogenization scheme presented in Section 2.1.2. For the presented combined objective function, a total of 6 points (predicted maximum and minimum for dielectric, elastic, and magnetic) will be predicted as the extrema. The 6 points then will be evaluated using the corresponding processing parameters and the computational homogenization scheme from Section 2.1.2. Evaluating 6 points will have a longer run time than if a single set of processing

parameters were return each for a combined maximum and minimum properties, totaling only 2 points to test using computational homogenization. However, finding individual maxima and minima answers the purer form of the question, “which constituent set leads to the largest range of possible properties”, and will generate a pareto front between the individual property values.

Since this work is presenting a general framework for this type of optimization and not designing for a specific application the properties will all be considered with equal weights. There are several ways to apply weighting to the components of a multi-objective fitness function, some of which are tabulated in Table 2. For this framework the weighted exponential sum will be used and simplified by setting the exponents to 1, the result of which is shown in equation (2-93).

**Table 2: Multi-objective optimization fitness function weighting methods [145]**

Bridgman (1922): Product of Powers	Weighted Exponential Sum	Goal Programming
$U = \sum_{i=1}^k [F_i(\mathbf{x})]^{w_i}$	$U = \sum_{i=1}^k w_i [F_i(\mathbf{x})]^p$ $U = \sum_{i=1}^k [w_i F_i(\mathbf{x})]^p$	$U = \sum_{i=1}^k  d_i $ $d_i = F_i - F^*$

For this work, the objective function used is given as

$$f = w_1(Y_{max} - Y_{min}) + w_2(M_{max} - M_{min}) + w_3(\varepsilon_{max} - \varepsilon_{min}) \quad (2-93)$$

where  $w_{1-3}$  are the selected weights for each of the subcomponents of the fitness function,  $Y$  is the elastic modulus,  $M$  is the mass magnetization, and  $\varepsilon$  is the relative permittivity. The minima and maxima are predicted values of the machine learning model presented in Section 2.3. . The weights will be set to sum to 1 and as mentioned previously will all be set equally for this

framework, e.g.  $1/3$ . Additionally multiple starting points will be sampled to ensure that the optimal points found are not local minima.

For this work we will incorporate gradient based methods to produce successive trials points, each requiring the results of computational homogenization, leading to solution of the optimization problem in Section 2.4.1. The analysis will be carried out in Python using the scipy library for the optimization. The presented optimization framework will additionally be augmented by machine learning, also using Python. The presented application uses machine learning in a novel manner, similar to AI learning by using a growing data set as opposed to a previously generated static set to define the previously unknown relationship between processing conditions and effective composite properties.

## **2.5 Artificial Intelligence and Machine Learning Models**

In most search problems there is a tradeoff between exploration and exploitation (using the model to predict the outcome). Finding a balance between these is at the heart of every AI approach. There are two methods for using AI presented in this work that capitalize on this concept differently. One approach is to use the data already collected to train a model on the design space, to address the inverse problem of finding processing parameters to manufacture a material with desired properties. The other is to augment the costly homogenization simulations with a model that, once trained sufficiently will be used to predict the remainder of the properties within the optimization search. Traditional data science applications apply a model to a generated data set, whereas the proposed solution is a novel design space search and simultaneous data generation framework which would apply the model iteratively to a growing data set.

Feed Forward Neural Networks (FFNN) train the model on a set of data that is randomly divided into batches that determine when to update the weights, and epochs that determine how many iterations through the training data set should be completed to train the model fully. Each iteration uses the paired input and output data to update the weights of the network. Early stopping criteria and adjusted learning can be specified such that the learning rate responds to the error in the model and terminates the iterations once the error no longer improves past a specific threshold. The model was tuned using additional arguments laid out in Table 3. Function parameters that specify the model's learning behavior are sometimes referred to as hyperparameters, and can be optimized themselves for efficient model training [146]. The hyperparameters for this model were tuned manually by looking at the resulting data and making adjustments to reduce run time since this is not a process that would need to be repeated many times once tuned for this system. Even with the added parameters for adjusting the learning rate and early termination once no improvement is made, training the neural network can take a few minutes depending on the data. The stochastic nature of the neural network training algorithm can also require rerunning the training if the data random data partitioning results in a poorly trained model, potentially leading to several iterations of training and increasing the run time with each additional iteration. The training described here would need to be repeated whenever a new point is added to the dataset from the optimization framework presented in Section 2.4.

In contrast, Support Vector Regression (SVR) performs an optimization to minimize the regressions coefficients and points outside of the acceptable error range, demonstrated in Figure 2-23. The base function for the regression can be defined by built in regressions (kernel functions) or a custom function. As mentioned previously, this is advantageous because a custom function can be defined to resemble effective medium model and expected system behavior.

**Table 3: Model Conditions that are additional arguments to be used to reduce run time and improve training efficiency of Neural Networks**

<b>Learning Rate Adjustment</b> (Reduce the learning rate if the monitored value plateaus)	<b>Early Stop Criteria</b> (stop if there is no improvement in the results for computational expense reduction)
<ul style="list-style-type: none"> <li>• Monitor loss variable</li> <li>• (Reduction) factor = 0.1</li> <li>• 2 points without change define a plateau, patience = 2</li> </ul>	<ul style="list-style-type: none"> <li>• Monitor loss variable</li> <li>• Minimum change in monitored value to qualify as an improvement, min delta = 1e-5</li> <li>• 9 points without change define a plateau, patience = 9</li> </ul>

In order to verify the model application and behavior a simple exponential function with random noise added was used to setup test cases (sample problems with known solutions to verify the coding framework). A linear base function was selected to observe the SVR behavior easily. Figure 2-24 shows the results of this noisy exponential test set and the linear SVR model with example error bounds drawn in to illustrate the manner in which the model fits the data. Figure 2-24 shows a few selected points from the data set in teal, rather than black. The teal points are the support vectors the model uses to define the regression. They are the points on or outside of the error bounds which need to be minimized along with the weights,  $w_i$ . The figure demonstrates the training method of the SVR (the model's minimization scheme).

Two approaches are developed and presented for incorporating machine learning into an optimization scheme and using a growing data set, optimization based search and SmartSearch. Optimization based search allows the gradient based optimization framework to choose points and learns passively about the system as the optimization progresses. With some convergence criteria, the physics based modeling can be substituted with the regression to predict the system's response and eliminate further computational expense. Points are not spaced out to train the



model most effectively with this method, since the areas of higher sampling are near the minimum of the objective function.

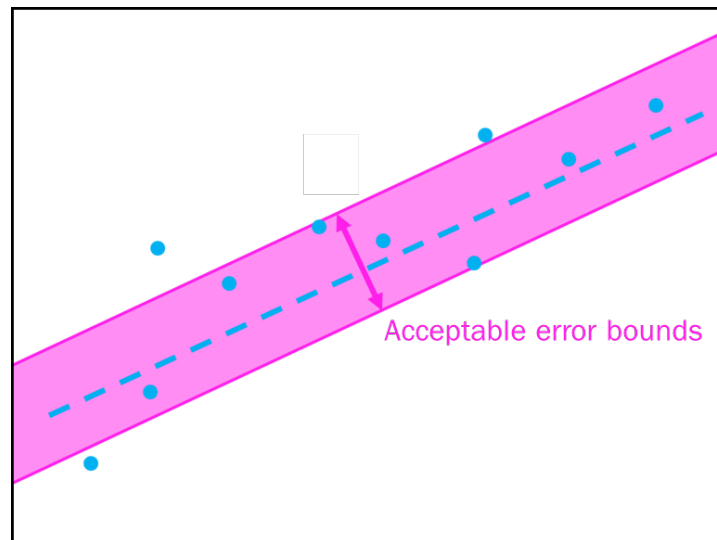


Figure 2-23: Example of error bounds, shown in pink, on a sample regression, represented as a blue dashed line, with data points that fall inside and outside of the bounds. The points outside of the shaded pink acceptable error bounds would be considered in the minimization of the SVR model training

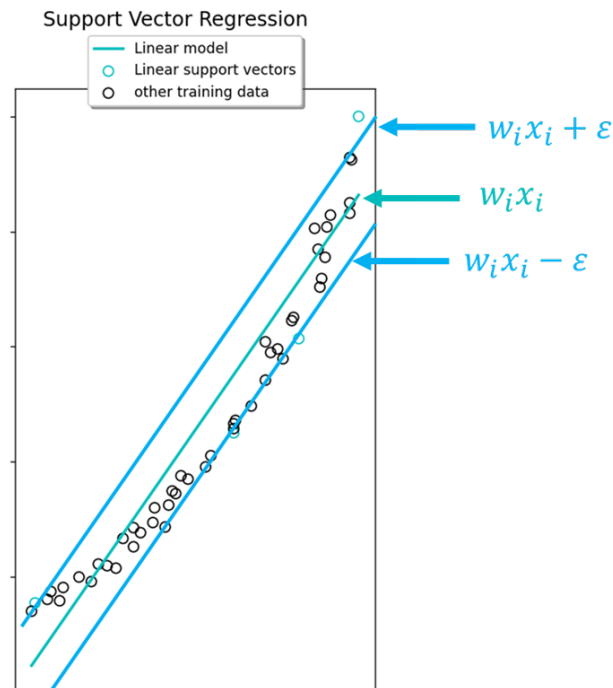


Figure 2-24: Support Vector Regression (SVR) of noisy exponential test function (black circles) showing the fitted function and sample error bounds lines to illustrate the objective function SVR uses to adjust the weights. Teal points are support vectors that define the regression

Alternatively, SmartSearch can learn actively. The machine learning model can be checked at every step and allowed to determine the next test point based off of the fit and spacing between test points. In the smart search algorithm the search areas where independent variables are spaced apart and the error between training data and predictions are large would be searched before points where the increments in independent variables are comparatively small and error is low. This method is demonstrated in Figure 2-25 with a succession of points checked to train the model. The steps in Figure 2-25 show the results of an SVR trained using smart search, indicating the points with the largest error and input spacing. The midpoint of the independent variables of these two points is found and used as the next test point. Such a self-selection method for determining training data has yet to be used with SVR and shows promise for efficiently searching the design space.

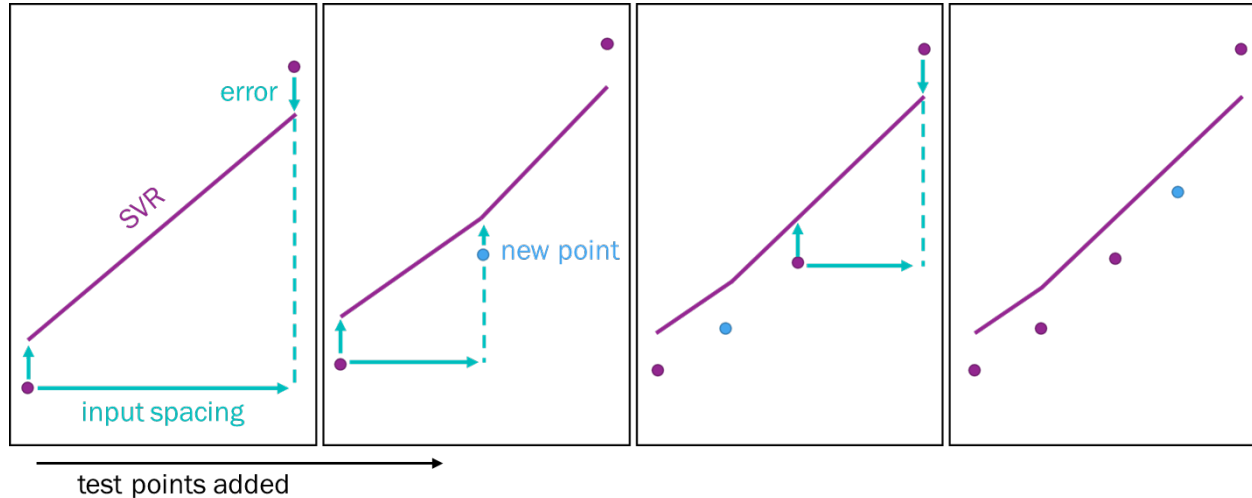


Figure 2-25: Succession of point selection using SmartSearch with an SVR model (shown in purple) based on largest distance between independent variables and largest error in outputs (shown in teal) to define independent variables for the next test point (shown in blue)

Again, here is a choice of algorithms between a focus on exploration (of the design space) versus exploitation (of the model's predictions). The optimization based search collects

data along the way focusing on the exploitation of the model more than the exploration of the design space. However, the exploration is key to a well-trained model that can be exploited effectively. The exploration, of the design space is a cost to pay, but will be an investment in a strong predictive model that reduces the overall computational cost as compared to a less balanced optimization based search method. For this reason the smart search method will be used in conjunction with the gradient based optimization to determine the best constituent set for the widest range of composite properties in a hybrid optimization scheme.

The hybrid optimization scheme will explore the design space to understand the behavior of the system and use this knowledge to optimize the input parameters. In the proposed method, the model will be initialized with test points, the initial exploration upfront cost to this method. The generated SVR model can then be used to evaluate the objective functions of the optimization scheme presented in Section 2.4. After an optimal point is found with the SVR model, the combination of independent variables for this point are then modeled using the particle dynamics and COMSOL simulations to determine the fit between the machine learning model and the simulation results. This is done using a simple error formula, for normalized values, where the simulation results are considered the actual value and the SVR prediction is the experimental value.

$$error = experimental - actual \quad (2-94)$$

This evaluation of the fit of the predictions to the simulation data will be used to determine convergence of the machine learning model to the simulation results. If the error is outside of the bounds the model is retrained with this new point, checked again, and trained for an additional step if the fit is still too poor. This is repeated until the error is acceptably low and the optimum is found. The workflow presented here is laid out in Figure 2-26. Such a process reduces the

number of simulations run when evaluating the optimization function. The proposed scheme uses gradient based optimization, but it should be noted that with the anticipated cost savings, a more expensive optimization scheme, such as genetic algorithms, could feasibly be used.

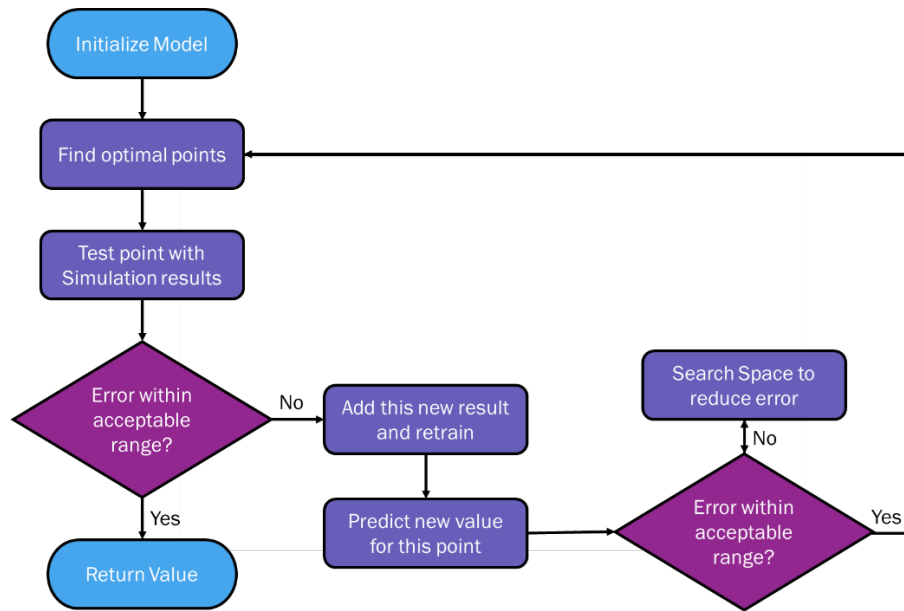


Figure 2-26: Hybrid optimization workflow (using smart search and gradient based optimization)

To further make use of the machine learning model generated through this search, the trained model can be saved and used to solve the inverse problem. This can be done by plotting the relationship between independent and dependent variables, or extracting the coefficients to construct a mathematical representation, and selecting the independent variables that best fit the desired properties or using this model to answer further optimization questions of searching for a desired property within some operating bounds, perhaps at the cost of other properties. The well trained SVR model is capable of reducing the computational expense to that of evaluating a mathematical expression within a search scheme.

## 2.6 Summary

In this chapter methods for analytical and computational homogenization were presented. RVE selection, boundary conditions for each domain, and post processing of results were laid out with particular attention setting up for comparison with experimental results. A homogenization framework for conductivity using AI techniques was developed. The model proposed considers distances between particles and uses efficient data structures to determine effective conductivity. A general framework for optimization and machine learning was presented to improve the computational expense of the number of simulations needed to find the optima. The resulting model simultaneously addresses the inverse problem allowing processing parameters to be determined from desire properties.

### CHAPTER 3: Computational Homogenization Results and Discussion

The previous chapters have presented the historical approaches to homogenization including analytical and computational methods. Although analytic models, previously presented in Section 1.2.3.1, are a computationally inexpensive methods, they do not have the capability to account for changes in processing conditions and the resulting hierarchal microarchitectures. Therefore, computational homogenization is applied to this system, reserving analytical homogenization as a means to verify initial computational results. This chapter presents results from evaluating selected effective medium theories, computational homogenization, and comparison to experimental data. A convergence study and selection of RVE size are also presented. These results were presented, in part, at the ASME Conference on Smart Materials, Adaptive Structures, and Intelligent Systems in 2020, *A Computational Framework for Predicting Properties from Multifield Processing Conditions in Polymer Matrix Composites* [25].

#### 3.1 Effective Medium Approaches

The effective medium models of Chapter 2 were compared to each other with respect to volume fraction to show the bounds and expected trend of properties for changes in volume fraction alone, which can be used as a refence for determining convergence of the computational model. The results for modulus versus volume fraction for the Halpin-Tsai, Mori-Tanaka, and Voigt and Reuss bounds for the BHF and PDMS composites are presented in Figure 3-1. As expected, as volume fraction increases, moduli increase. The Voigt model returns substantially higher moduli because of its uniform strain assumptions especially when compared to the Reuss model at the other extreme, as expected these two present upper and lower bounds. Within these bounds, the longitudinal, z direction, Halpin-Tsai trends larger than Mori-Tanaka which trends larger than

the transverse, x/y direction, Halpin-Tsai. For the convergence study, the Halpin Tsai model will be used as targets for moduli to assess convergence of the elastic properties, since it can account for the transverse and longitudinal moduli.

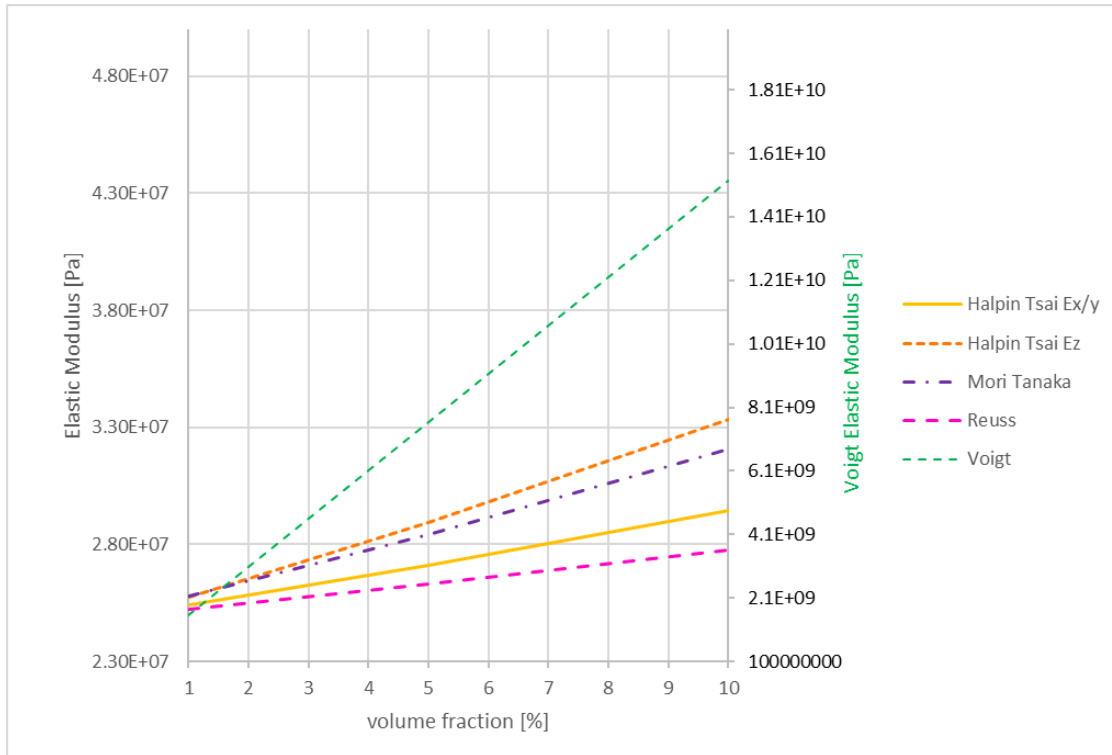


Figure 3-1: Effective medium models for BHF/PDMS from 1 to 10% volume fraction showing the bounds and expected trend of properties

### 3.2 COMSOL Simulations and comparison to Effective Medium Theory

In this section COMSOL simulation results are reported for convergence and comparison to effective medium theory. The Halpin Tsai model is used to aid in the determination of convergence of the mechanical model, as detailed in this chapter. Additionally, their degree of anisotropy as a result of ordering and alignment of particles due to applied fields is presented and discussed for both domains. All constituent material properties used for homogenization are laid out in Table 4. The results for the computational homogenization for elastic and dielectric response are compared to their respective effective medium theories and then presented for the 5

main field cases tested experimentally. The experimental cases are explained in Figure 3-2 and Table 5.

Elastic, dielectric, and magnetic material properties for both Barium Hexaferrite (BHF) and Polydimethylsiloxane (PDMS) are presented in Table 4. The properties laid out here are used in the computational homogenization of the composite RVE's. Note that the elastic modulus,  $E$ , for the PDMS matrix was updated after initial results were obtained. The update of the matrix modulus was made after comparison with experimental values to calibrate the model, and the updated modulus (2.1 MPa) was used to train the machine learning model for which validation is presented in Section 3.3.1. The relative permeability,  $\mu_r$ , for PDMS was set to 1 since it is not a magnetic material. For the BHF the relative permeability was set as 2 with comparison to values from the literature [147], [148]. The permeability of the BHF particles determines the effect of the interaction between particles, since the homogenization evaluated the “as cured” original remanent mass magnetization of the composite. The magnetization of the particles was set to  $370000A/m$  [26], [52]. The dielectric permittivity,  $\epsilon_r$ , set to 2.67 and 32 for the PDMS were BHF, respectively in accordance with values reported by Masud for experimental samples [20].

**Table 4: Constituent Material Properties**

	PDMS Matrix	BHF Particles
Elastic	$E = 270 \text{ kPa}$ (initially) $E = 2.1 \text{ MPa}$ (updated)	$E = 152 \text{ GPa}$
Magnetic	$\mu_r = 1$	$\mu_r = 2, M = 370000A/m$
Dielectric	$\epsilon_r = 2.67$	$\epsilon_r = 32$

Representative results of the particle arrangements resulting from particle dynamics simulations (following methods laid out in Chapter 2.1.1) are shown in Figure 3-2 for the five experimental



cases studied by Masud [20]. Case 1 is the reference case with no field applied. In this case particles align solely due to the magnetic interactions between particles. In Case 2, an electric field is applied, inducing an electric dipole that aligns particle along their major axes in the direction of the applied field. In Case 3, a magnetic field is applied which aligns the magnetic dipole moment of the particles (through their minor axis) with the field. Case 4 uses both electric and magnetic fields, applied perpendicular to one another. The addition of the magnetic field in Case 4, as compared to the electric field only Case 2, introduces additional ordering. Particles will align with their induced electric dipoles on the major axis with the electric field, while also aligning their permanent magnetic dipole with the magnetic field. Like Case 4, Case 5 also uses both electric and magnetic field, however, applied in the same direction, parallel to each other. The resulting alignment and orientation is dependent on the relative strengths of the applied fields, which is further discussed in Section 4.1. The field strengths used, electric and magnetic, in the respective cases are given in Table 4.

The resulting microarchitecture of the 5 main cases of applied fields investigated are shown in Figure 3-2, demonstrated with 100 particles, for which the applied fields are laid out in Table 5. In the table, the field orientation column (column 2) has a symbolic representation of the field direction with colors corresponding to the particles for each case from Figure 3-2 for clear visualization of the 5 cases presented. Columns 3-5 show the magnitude of the components of the electric field vector and columns 6-9 show the magnitude of the components of the magnetic field vector for the applied processing fields for each case. The electric field for Case 5 was reduced compared to the field for Case 2 and Case 4 as a result of analysis of initial runs with the higher field. The applied electric and magnetic fields in Case 5 generate competing torques, further discussed in Section 4.1. The resulting particle orientations are dependent on the relative

strength of the two fields. The strength of the electric field used in the simulations was tuned to form structures that better resemble experimental results for Case 5.

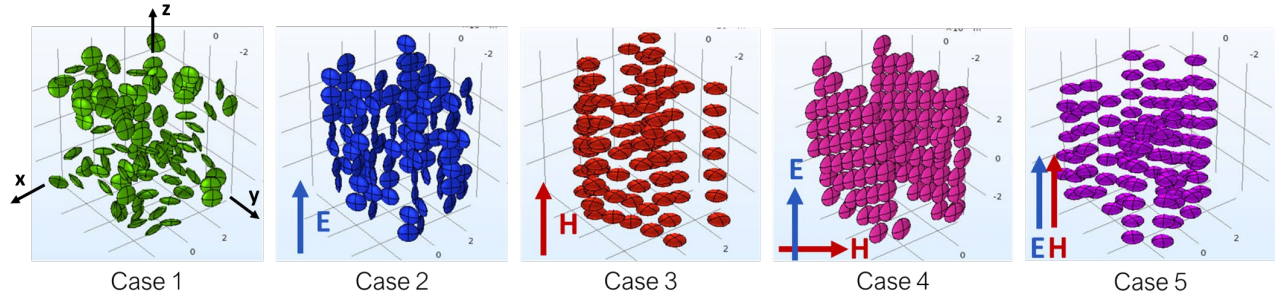


Figure 3-2: Cases 1-5 microarchitecture for simulations with 100 particles showing the effect of applied fields on the orientation, position, and microarchitecture of the particles

Table 5: Applied fields for each of the 5 cases shown in Figure 3-2.

	Field Orientation	Electric Field			Magnetic Field		
		$E_x$	$E_y$	$E_z$	$H_x$	$H_y$	$H_z$
Case 1	—	0	0	0	0	0	0
Case 2	↑	0	0	1e8 V/m	0	0	0
Case 3	↑	0	0	0	0	0	2.5e5 A/m
Case 4	⊥	0	0	1e8 V/m	0	2.5e5 A/m	0
Case 5		0	0	1e6 V/m	0	0	2.5e5 A/m

### 3.2.1 Convergence Study on Model Size for Elastic Properties

In order to determine the appropriate size of the RVE, convergence was observed for a given volume fraction. Convergence is observed in the change in property with respect to model size (particle count). These results are then compared to effective medium theory which can serve as a baseline reference for the model, though the results are not expected to match due to the limitations of effective medium theory, namely particles distribution and concentration, on

system like this one being modeled here. Convergence studies are presented for elastic and dielectric properties.

For the elastic property convergence study 10% volume fraction was used and particle counts were increased from 10 to 500. A 10% volume fraction sample would have decreased particle mobility as compared to a dilute model with, for example, 1% volume fraction, but is not limited beyond the ability to self-organize, and therefore represents a more challenging system in which to form structures (a worst case scenario). Herein, convergence was defined by the change in modulus with simulation size and relationship to the Halpin Tsai and Mori-Tanaka models. The compliance values were not expected to converge to the effective medium models directly since the RVE's exhibit chaining, but the effective medium model values served as a bound for which in-plane (x/y) direction values should fall within and the chaining (z) direction should not. A volume fraction of 10% was chosen to ensure the system is not dilute and particles were able to form fully realized microarchitecture. Similar to Pahlavanpour et al. [72], the difference in the value of physical property found from simulation when compared to the expected value found using effective medium theory was considered when determining if the model was sufficiently large. For this convergence study the electrically aligned case was chosen as it would be the expected worst case scenario for RVE size due to the need to represent the microarchitecture formed. The elastic modulus for the electrically aligned case (Case 2) was considered the worst case because the major axis of the particles aligns with the applied electric field, resulting in longer chains per particle count, and thereby a larger RVE needed to contain the chain.

As per the methods laid out in Chapter 2, the computational homogenization was conducted using the particle distributions found from the particle dynamics simulation. An electric field of  $1e8$  V/m, Case 2, was applied in the z direction of the RVE, with 10% volume

fraction, and increasing particle count from 10 to 500 particles to determine convergence. It is theorized that the structures formed in Case 2 will drive the elastic properties, and that at higher volume fractions larger models will be needed to represent the structures formed, making this the worst case scenario for convergence. The geometry was modeled in COMSOL using the positions and orientations found in the particle dynamics simulations.

The visualization of simulation results of increasing volume of the RVE and microarchitecture represented by them are shown in Figure 3-3, showing the realization of the microarchitecture with increasing model size. Distinct chains can be identified in RVE's with at least 100 particles but start to become more define only with at least 400 particles.

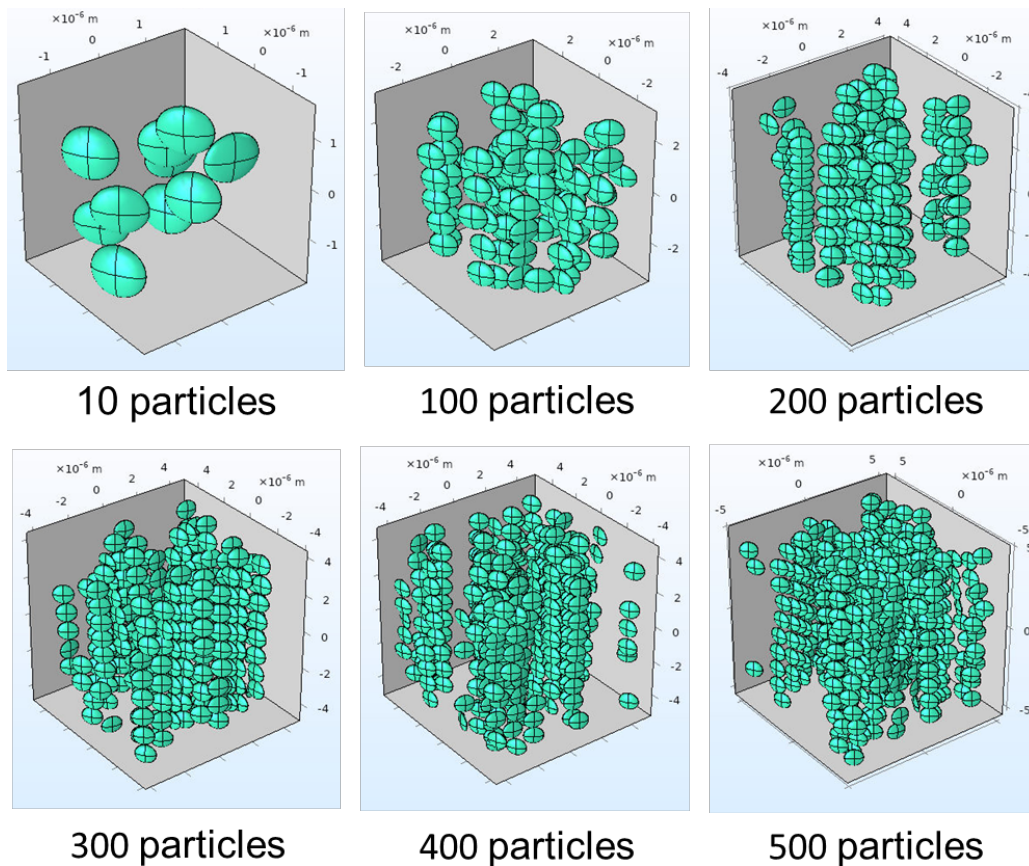


Figure 3-3: Figures showing microarchitecture versus simulation size where chain formation becomes more well defined with increasing particle count

To determine the elastic response, periodic boundary conditions were assigned to all face pairs and using 6 load cases strains are applied to solve the conservation of linear momentum equations, as described in Section 2.2.2.1. Figure 3-4 shows the compliance in the longitudinal and transverse directions of the RVE for various sizes with comparison to effective medium models as reference for convergence. The transverse directions settle in near the Mori-Tanaka model and Halpin-Tsai lower bound, which is in good agreement with the expected behavior since the alignment of the chains is in the longitudinal direction. The chains are expected to be stiffer in the direction of the alignment, which can also be seen in the plot in Figure 3-4 for larger models, with the results exceeding the expected upper bound. The effective medium bounds are expected to be exceeded in this case since there are chains forming that the Halpin-Tsai model does not account for, which would increase the stiffness of the composite as compared to uniformly distributed particles that the model assumes. From the results presented in Figure 3-4 it could be determined, by the settling out of the mean values and the convergence of the in-plane component to within the effective medium bounds (as expected) and the chaining direction (z-direction) component converging to stiffer than the effective medium predictions, that the simulations for the mechanical domain should be at least 400 particles in size. However, due to computational expense smaller simulations, 100 particles, were run to explore the design space, understanding the error associated with these simulation sizes. The larger models, using 400 particles, will be used to find the final optimized values once the smaller models have zeroed in on the solution. The behavior of the system at 100 particles still shows the anisotropy expected for an aligned case and the behavior seen at larger simulation sizes.

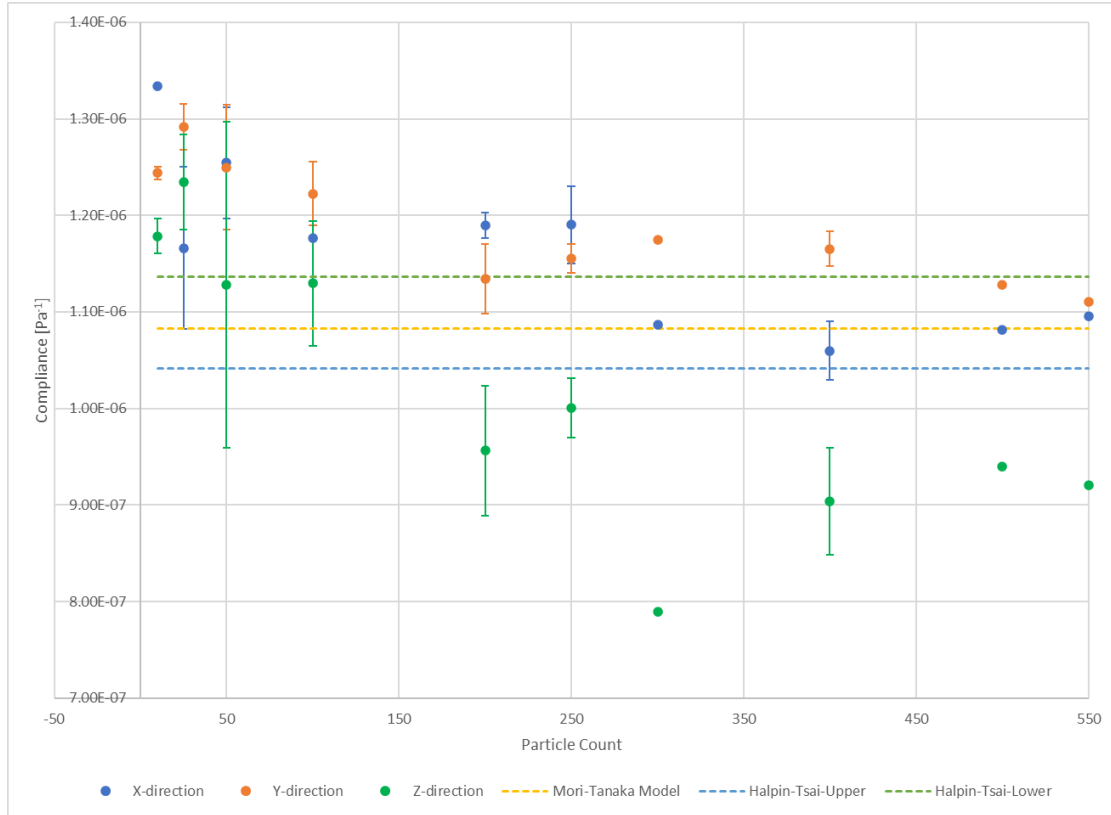


Figure 3-4: Convergence study of compliance versus simulation size (particle count) at 10% volume fraction of an electrically aligned case compared to effective medium theory

For other material responses, convergence needs to once again be considered as it is a function not only of the geometrical representation but the effect that representation has on the specific domain. The dielectric response is compared to the Bruggeman effective medium model to determine convergence and to validate the model. These simulations were evaluated with a DC field. Later models use an AC field for comparison to experimental data, however the trends and behavior of these DC simulations is valuable for observing convergence and the effects of particle spacing.

For this convergence study, the particles in the dynamics simulations were again aligned in a  $1e8$  V/m electric field applied in the z-direction, with 1% volume fraction, to compare a dilute system with the effective medium model that assume a dilute system, and all simulation

and material parameters as previously discussed. Solid models generated from the particle dynamics simulations were used in the periodic computational homogenization analyses discussed in Section 2.2.2 to generate relative permittivity, with the exception that these results were found with a DC field.

For the dielectric properties, the results of the convergence study are presented in Figure 3-5, showing the change in permittivity in the x, y, and z directions for increasing particle count (model size) for 1%, 5%, and 10% volume fraction. The direction of particle alignment is again, like the elastic convergence study, in the z direction. For the 1% volume fraction study (Figure 3-5 a), the x and y direction permittivity is seen to converge and plateau quickly, settling out near 2.72 at 100 particles. It should be noted that the orientation of the particles within the chains is not ordered in Case 2 (unlike Case 4) and that this potentially leads to some degree of variability in the x and y permittivity. The z direction permittivity is in line with the applied electric field and therefore chaining direction, which converges later at 150 particles settling in near 2.8. Comparing to the Bruggeman effective medium model, the x and y directions stay below the model's prediction for spheres. The ellipsoidal particles are narrower in their minor axis than the assumed spheres of the Bruggeman model, and due to the lack of order in the Case 2 minor axis direction (without the application of a magnetic field to align the magnetic dipole moments of the particles) this smaller axis is distributed throughout the x and y directions, corresponding to the reported x and y direction values in Figure 3-5 a. The permittivity in the z direction only corresponds to the Bruggeman value at 10 particles for the 1% volume fraction study, which as seen in Figure 3-3 does not show any chaining or structuring. The computationally computed permittivity exceeds the prediction of the Bruggeman model, once large enough to represent the

resulting microarchitecture of the applied fields, which is expected due to the chaining and close proximity of the particles within the chains in this direction.

Similar trends are shown in Figure 3-5 b and Figure 3-5 c for 5% and 10% volume fraction, respectively. The point where the z direction permittivity exceeds the Bruggeman model is at larger model sizes (particle count) for higher volume fractions, likely due to decrease in particle mobility and resulting difficulty forming structures. Although the absolute value of the permittivity converges only at larger model sizes, such as 200 for the 5% and 300 for the 10%, the anisotropy is consistent from 100 particles for 1% and 5% volume fraction and 150 particles for 10% volume fraction. The average permittivity values from Figure 3-5 are plotted together in a line plot in Figure 3-6 to illustrate the anisotropy trends. From Figure 3-6 it can also be seen that convergence is slower for higher volume fraction, reinforcing the assumptions made for the elastic property convergence study shown previously. The relationship between permittivity and volume fraction is also apparent in Figure 3-6 with the observed increase in maximum permittivity as well as the increase in range between the z direction (alignment direction) and the x and y directions. The structure dependent value that the computational homogenization demonstrates, seen in the anisotropy of the Case 2 values presented here, is necessary for optimizing the constituent set to maximize the range of possible properties, and cannot be predicted with traditional effective medium models.



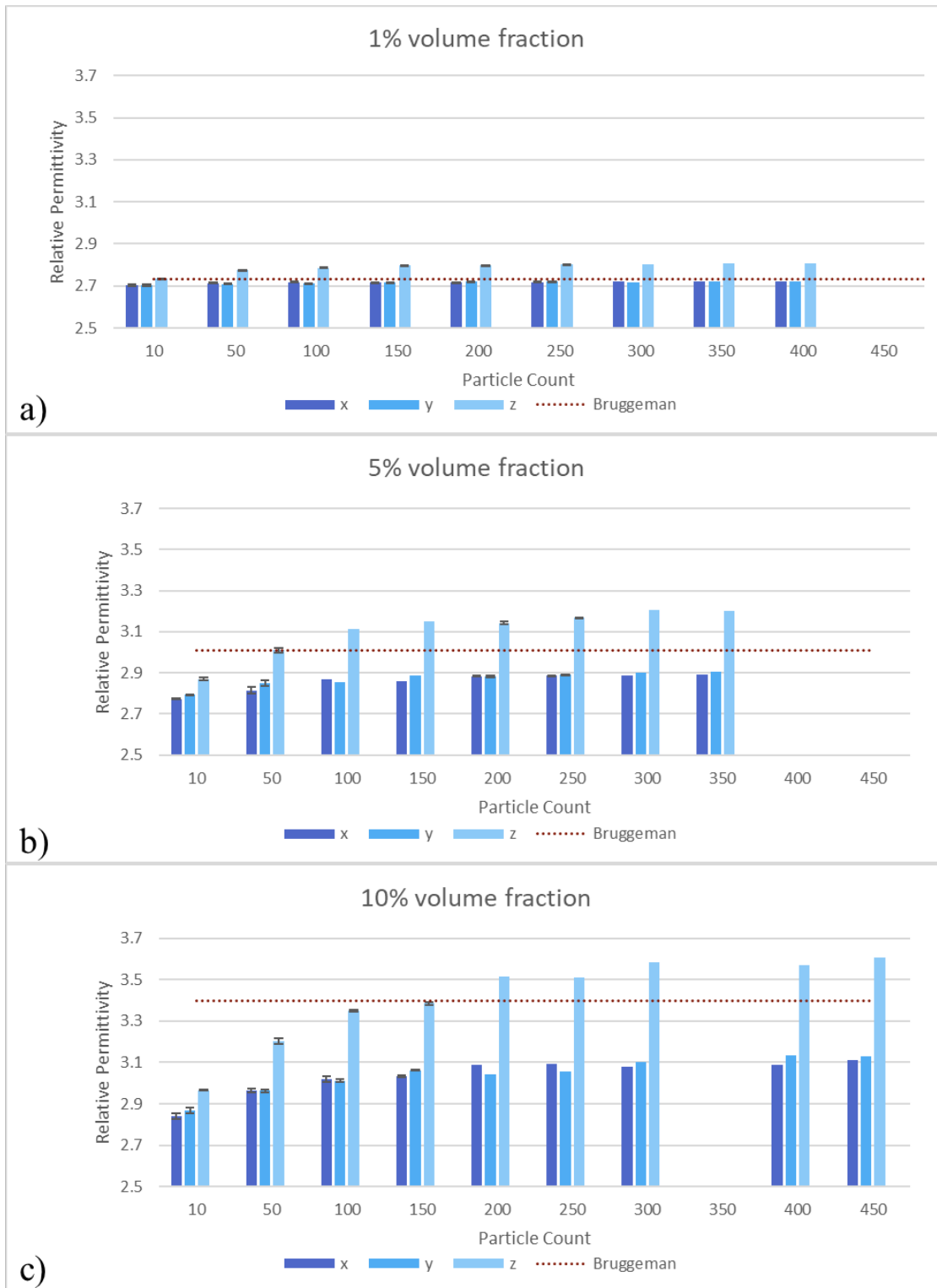


Figure 3-5: Dielectric convergence study showing relative permittivity in three directions for a) 1% volume fraction b) 5% volume fraction and c) 10% volume fraction versus particle count (simulation size) and the trend with respect to the Bruggeman effective medium model (dashed red line) for reference to determine convergence at each volume fraction

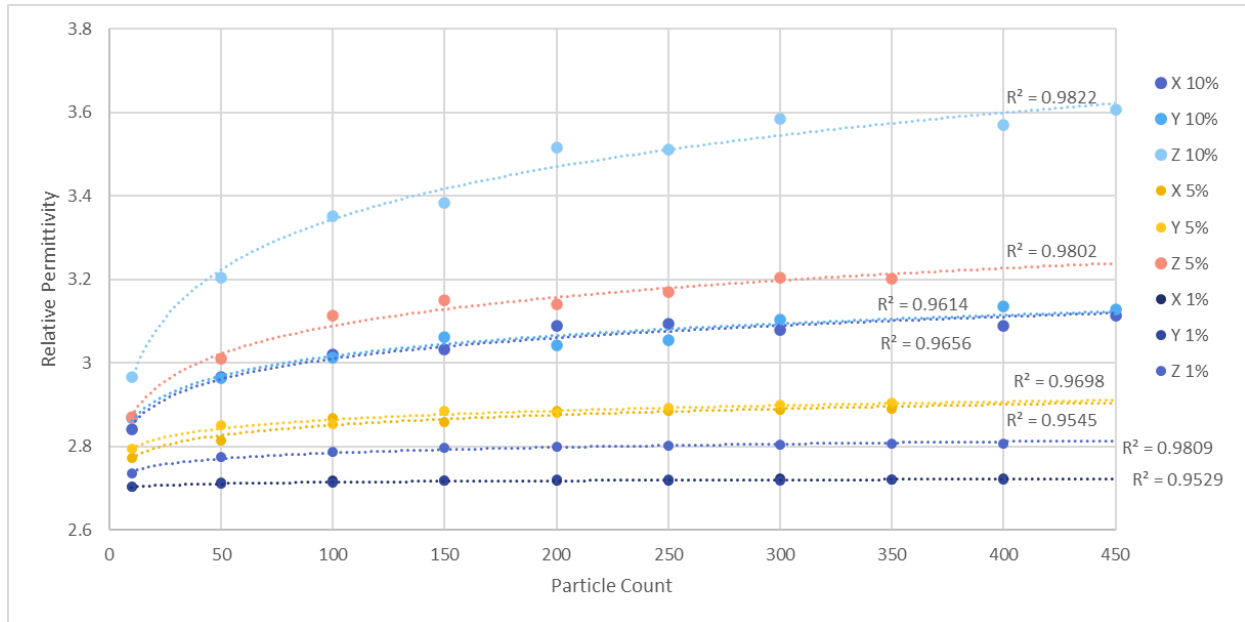


Figure 3-6: Dielectric convergence study showing relative permittivity in three directions for 1%, 5%, and 10% volume fraction versus particle count (simulation size) showing the anisotropy vs model size of each volume fraction tested

For consistency with the elastic properties and an understanding of the expected error, a model with 100 particles will again be used with the dielectric homogenization models. The consistent model size allows for direct comparisons of the three properties investigated for the same RVE.

### 3.2.2 Compliance and Moduli versus Volume Fraction

This section presents effective elastic properties compared to traditional effective medium theory and experimental values. The effect of the changes in particles distribution as a function of applied processing fields captured by the computational homogenization scheme highlights the shortcomings of traditional effective medium theory when attempting to apply it to this type of problem.

Several iterations of each of the no field, the electric field only, and the magnetic field only cases are modeled at 1%, 5%, and 10% volume fraction. The stochastic nature of the particle dynamics simulations leads to variation in results, as shown by the error bars in Figure 3-7. Recall that Case 1 does not have any applied fields, Case 2 has an electric field only (in this case  $1e8V/m$  as stated above for the convergence study), and Case 3 has a magnetic field only (in this case  $250,000 A/m$ ). All other parameters are the same as those used for the results in Figure 3-4. The effect of volume fraction and single fields on mechanical stiffness are shown in Figure 3-7, demonstrating the anisotropy that can be achieved from altering processing conditions alone.

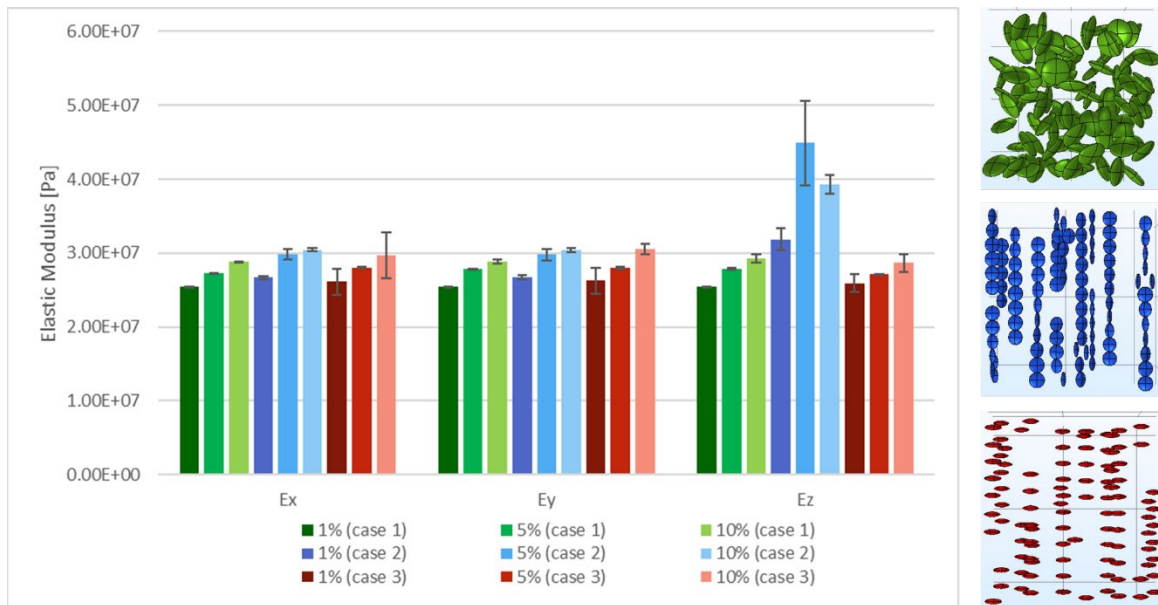


Figure 3-7: Elastic modulus versus various volume fraction of no field (Case 1, green), electric field only (Case 2, blue), and magnetic field only (Case 3, red) cases having 100 particles each, showing the increased anisotropy for Case 2 which has chaining in the z direction

The effect of the chains in Case 2 becomes very apparent in the anisotropy of the modulus, where chaining increases the stiffness of the composite in the direction of chaining. The ranges in error for each case and volume fraction also suggest that the Case 2 (electric field only) at 10% volume fraction should be the worst case scenario for convergence and the study

presented in Section 3.2.1 was accurately chosen as a representative condition. In contrast, Case 1 is isotropic as a result of the uniform distribution of particle spacing and orientation within the RVE, leading to no increase in stiffness in any particular direction over another. Added stiffness in the composite comes from additional (stiffer) filler and the anisotropy, seen in Case 2, from preferentially distributing and orienting that filler within the volume to increase its influence in a given direction over another, such as by chaining. The shape anisotropy of the BHF particles allows for the distributions of particle orientations to contribute to this change in properties. Spherical particles, for example, would be limited to spatial distribution contributions to stiffness, but not to any affects from changes in particle alignments (rotations) within random or aligned chain structures.

A clear (and expected) trend for increasing stiffness with increasing volume fraction can be seen for all cases, noting that the one outlier (5% Case 2 in the z direction) has larger error bars that are likely due to the RVE size. The elastic modulus for the 10% volume fraction Case 2 increased by 23% from the elastic modulus at 1% volume fraction of the same case. This increase is in comparison to a 7% and 10% increase in elastic modulus from 1% to 10% volume fraction for Case 1 and Case 3, respectively, demonstrating the impact of the chaining. The anisotropy for the electrically aligned Case 2 also become apparent and more pronounced with increased volume fraction. The 1% volume fraction Case 2 has a 20% increased stiffness in the chaining direction versus the transverse directions, increasing to 29% increase in stiffness in the chaining direction for 10% volume fraction. The increase in stiffness is expected due to the chaining in the same direction as the increased stiffness, with more or thicker chains in the higher volume fraction models. Though some amount of ordering can be seen in Case 3, the limitation in the results presented here stem from the particle dynamics simulations, which

created a spacing between the particles that works against this effect and could be a possible cause for the lack of anisotropy in the Case 3 data. This spacing can be seen in the Case 3 (red) simulated architecture compared to the Case 2 (green) simulated architecture. The particle collision avoidance method employed in these simulations creates an avoidance distance equivalent to the particle radius (major axis), which works well for Case 2 but not for Case 3 (which should also collapse to near touching chains), as shown. The handling of the repulsive force and changes to particle distributions are addressed in Section 3.4. The distinction between Case 2 and Case 3 using the ellipsoidal particles, as opposed to geometrical isotropic particles, is seen in the orientation of the particle with respect to the applied fields and the resulting hierarchal structuring within chains. The orientation of particles from electric polarization and the permanent magnetic dipole of the BHF particles simulated here adds complexity and control to the structures formed, compared to the spherical particles used in previous work by Bharti et al, previously presented in Section 1.2.2 [19].

Figure 3-8 shows the same computational homogenization data in a scatter plot to compare it to effective medium theory (EMT) models described in Section 2.2.1. Due to the closeness of some of the data the error bars are left out for legibility, but have an average value across all data points of 4% with respect to the average modulus.

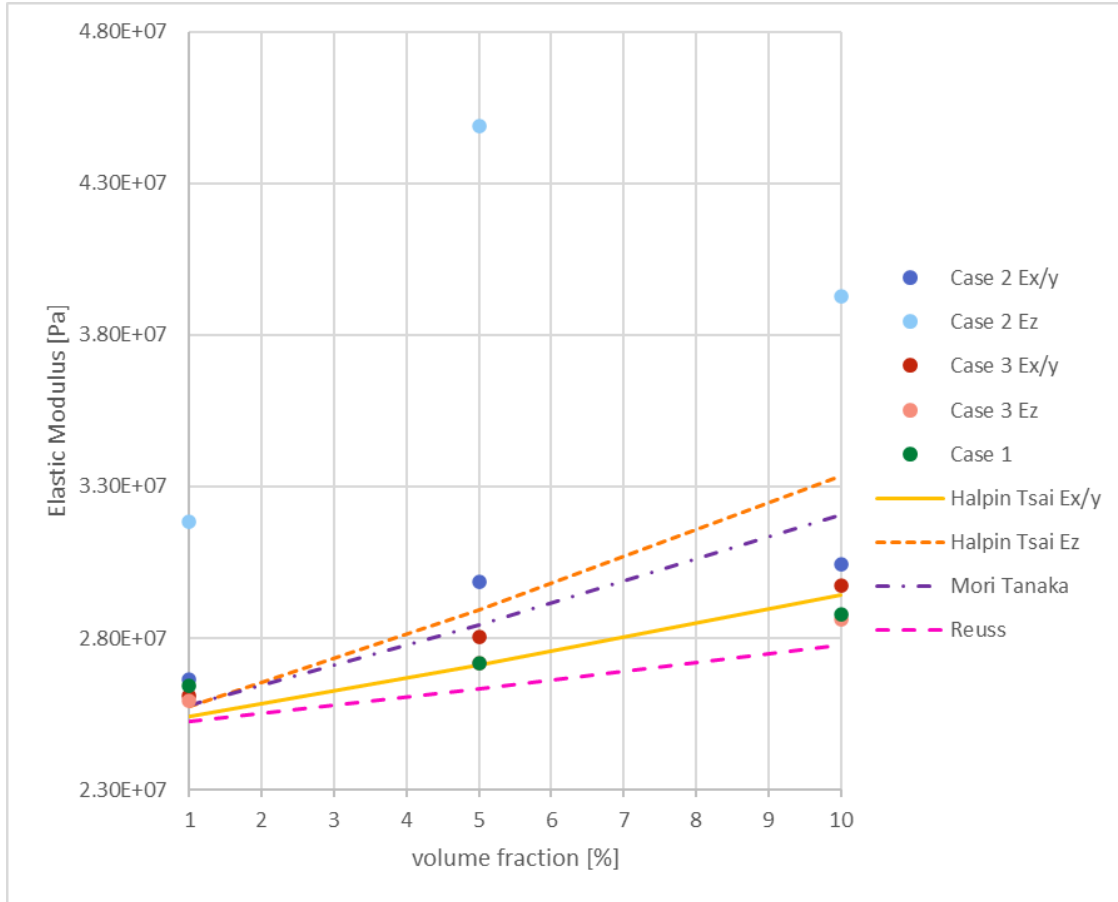


Figure 3-8: Elastic modulus versus various volume fraction of no field (Case 1, green), electric field (Case 2, blue), and magnetic field only (Case 3, red) cases (100 particles) compared to EMT (Halpin Tsai, Mori Tanaka, Reuss Bounds) showing the agreement of the computational data in the Ex/y directions and the increased stiffness as compared to the EMT models in the direction of the particle chaining for Case 2 (Z direction)

Looking at the Case 2 data in Figure 3-8, the anisotropy can be seen again in the in plane (x-y direction) versus out of plane (z direction) stiffness. The Halpin Tsai model, which accounts for particle orientation, but not spatial distribution (chaining), does not recreate this effect, which underlines the need for modeling predicted hierarchies to fully understand the effect of these microarchitectures on the composite property. The Halpin-Tsai model for the x/y direction does show some correlation in trend for volume fraction, particularly with the magnetic only aligned case 3. However there is no factor for microarchitecture, as in chains created by applying

combined external fields, and therefore this method cannot predict composite properties as a result of processing parameters in the same way that the simulations presented herein are able to do.

In this section, the elastic modulus in transverse orientation for three cases (no field (Case 1) , electric field (Case 2) , and magnetic field (Case 3)) were presented and compared. The results showed that chaining due to electric field application increases the stiffness of the composite in the direction of the applied field in the computationally homogenized results. Computational homogenization simulation models were shown to have good agreement with effective medium theory in the case of no field or transverse to the alignment direction, validating the simulations and resulting computational homogenization, at last for those orientations. The hierarchical nature of the alignment in the field direction (e.g. particle orientation within chain orientation) may be a causal factor for the difference between the simulation results and the effective medium results; experimental fabrication and testing will be used as a comparison in Section 3.3 for the moduli in the alignment direction.

### **3.2.3 Homogenized Anisotropic Dielectric Response versus Volume Fraction**

This section presents DC results to compare to effective medium theory. Section 3.3 will address the AC response in comparison to experimental data. The dielectric response here is presented for the 5 cases of applied fields (no field Case 1, electric field Case 2, magnetic field Case 3, electric and magnetic field perpendicular Case 4, and electric and magnetic field parallel Case 5). These results are compared to traditional effective medium theory for 1%, 5%, and 10% volume fraction for 100 particle size simulations using the material properties from Table 4, and methods presented in Section 2.1.2.2 and Section 2.2.2. The difference in DC response presented

here demonstrates the need for computational homogenization to predict the properties of various cases.

The relative permittivity for the 5 main cases are shown in Figure 3-9 and Figure 3-10, demonstrating the effect of the single as well as combined fields and the increase in relative permittivity of cases that have that have chaining induced along the electric field dimension (cases 2 and 4). It should be noted, however, that particle spacing has a fixed minimum in the simulations, which may preference electric field alignment in the plane of the plate and bias these results towards increased permittivity in the electric field direction as opposed to the magnetic field direction. Again, here the effects of the shape anisotropy and the resulting internal hierarchal structuring is seen, particularly in the results for Case 2 and Case 4. Both cases are electrically aligned, however, Case 4 has additional ordering from a perpendicularly applied magnetic field. The ellipsoidal particles in Case 2 have no preferred in-plane orientation once one of the 2 major axes is aligned within the chains in the direction of the applied electric field, resulting in a distribution of orientations of the minor axis. The addition of the perpendicular magnetic field adds a preferential direction for the permanent magnetic dipole, along the minor axis of the ellipsoids, that aligns the particles further within the chains. This additional order reduces the permittivity in the direction of the magnetic field, in Figure 3-9 denoted as the y direction, and increases in in the perpendicular in-plane direction, in Figure 3-9 denoted as the x direction. The additional hierarchal ordering therefore also increases the anisotropy of Case 4 as compared to Case 2, seen in Figure 3-10.



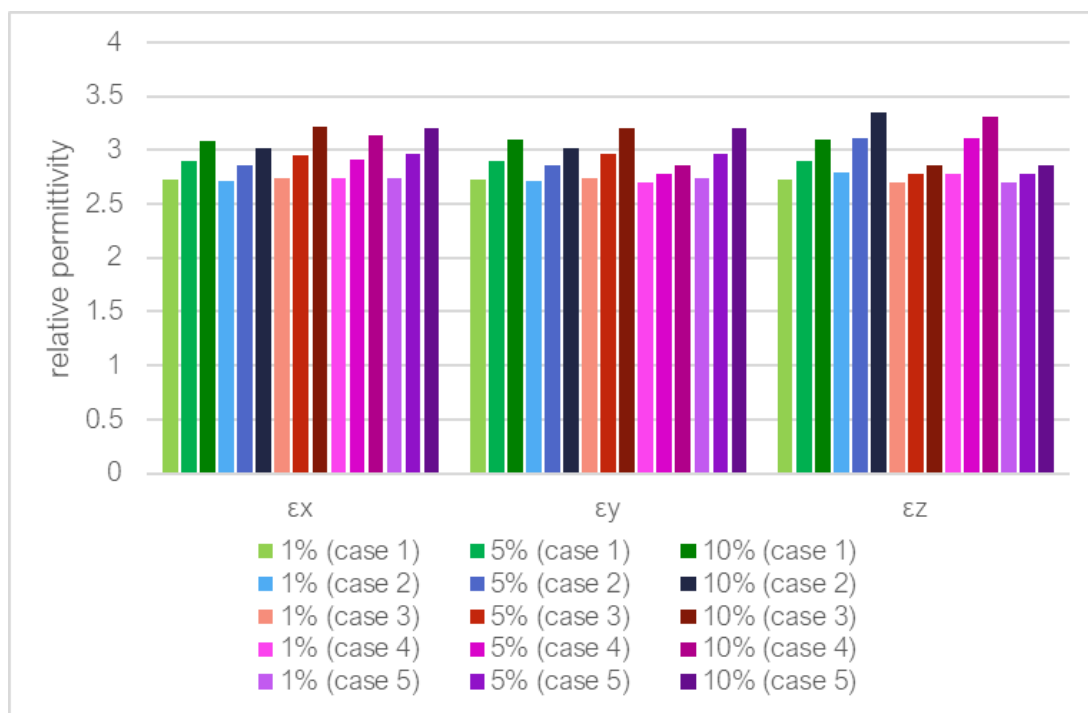


Figure 3-9: Dielectric response (relative permittivity) versus volume fraction in the x, y, and z directions. Results show the increased relative permittivity for cases 2 and 4 that have dense chains along the electric field dimension, e.g. for  $\epsilon_z$

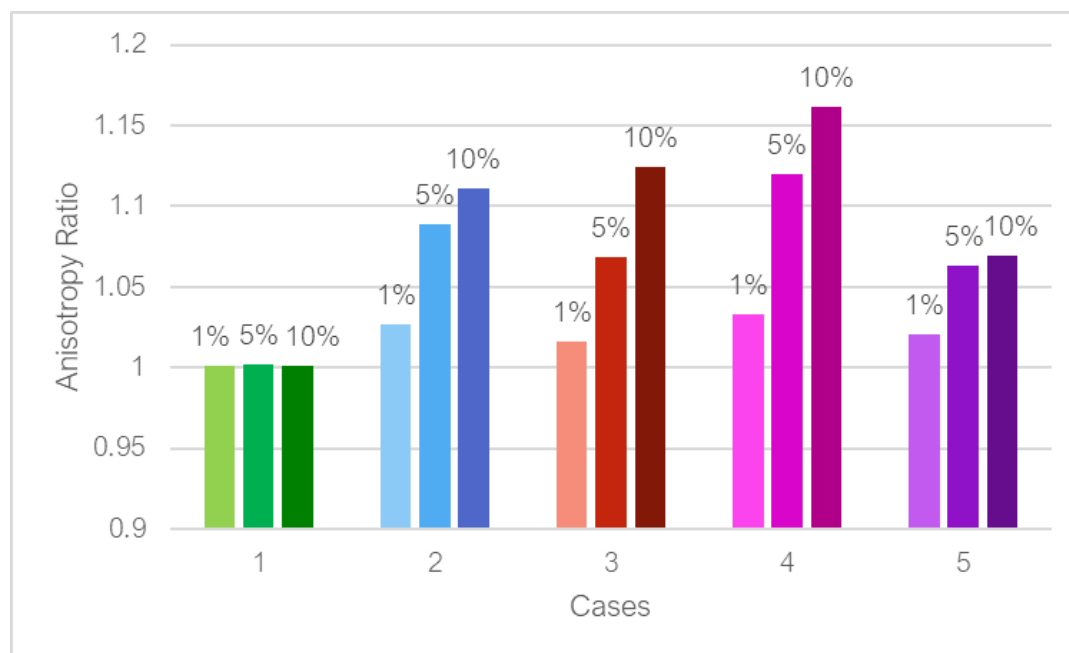


Figure 3-10: Ratio of maximum to minimum permittivity for each alignment case at 1, 5, and 10% volume fraction, showing the degree of anisotropy of each case with varying volume fraction, where case 4 has the highest degree of anisotropy in relative permittivity because all of the particles are oriented and aligned in chains

Anisotropy in permittivity is calculated from simulation results directly by dividing the maximum permittivity by the minimum permittivity for each volume fraction and case,  $\frac{\epsilon_{max}}{\epsilon_{min}}$ . The results are shown in Figure 3-10. The increase in anisotropy of Cases 2-5, ranging from 2% (Case 3 1%) to 16% (Case 4 10%) increase, as compared to the no field case (Case 1) show the effect that any degree of alignment and chaining has in the dielectric domain. As a reminder Cases 2, 4, and 5 have applied electric fields that lead to chain formation and reorient the particles, and Cases 3, 4, and 5 have applied magnetic fields that reorient the particles within the RVE. The increase in anisotropy with volume fraction, particularly for Case 2 and 4, illustrate the effect of particle proximity when the composite is not considered to have a dilute volume fraction, underlining the need for such simulations that effective medium theory would not accurately represent by its own definition. The gap present in Cases 3 and 5 (magnetically aligned) between the particles is an artifact of the simulation as mentioned in Chapter 2 and will have a significant impact on the results in the dielectric domain. The particle spacing causing this separation is further discussed in Section 3.4. It is expected that removing the gap will allow more distinct chain formation and increase the degree of anisotropy as well as the permittivity in the direction of chaining, however the transverse direction in this instance is the plane of the platelet particles, therefore its geometry is also a driver of the anisotropy but here preferencing the transverse plane.

The simulation results of the 5 cases at 1%, 5%, and 15% volume fraction are compared to the Bruggeman Effective Medium Model from equation (2-88) in Figure 3-11. The figure shows the general agreement between the effective medium model and the simulation results in general but does not capture the distinctions in effective properties from different particle distributions. The Bruggeman model used is simplified to assume uniformly distributed spherical

particles. The estimation made by this effective medium model is low for dilute volume fractions, such as 1%, with electric field chaining (Cases 2, 4, and 5), as expected. The Bruggeman model then begins to overestimate the effective permittivity of the model. From the convergence study shown in Figure 3-5 in Section 3.2.1, this is an expected result for the 100 particle size models at higher volume fractions. The higher volume fraction models could require larger model sizes to accurately capture microarchitecture with decreased particle mobility. As shown in Figure 3-11, cases with electric field chaining (Cases 2, 4, and 5) have higher permittivity than the cases without the electric field. The results of case 3 (magnetic field only) are lower than the effective medium model and the remaining cases, including the no field Case 1, due to the spacing between particles, and the lack of chaining in this case. It can be seen then, that the Bruggeman model is useful for verifying the functionality of the simulations but cannot be used for further predictions in variation of permittivity due to changing processing conditions.

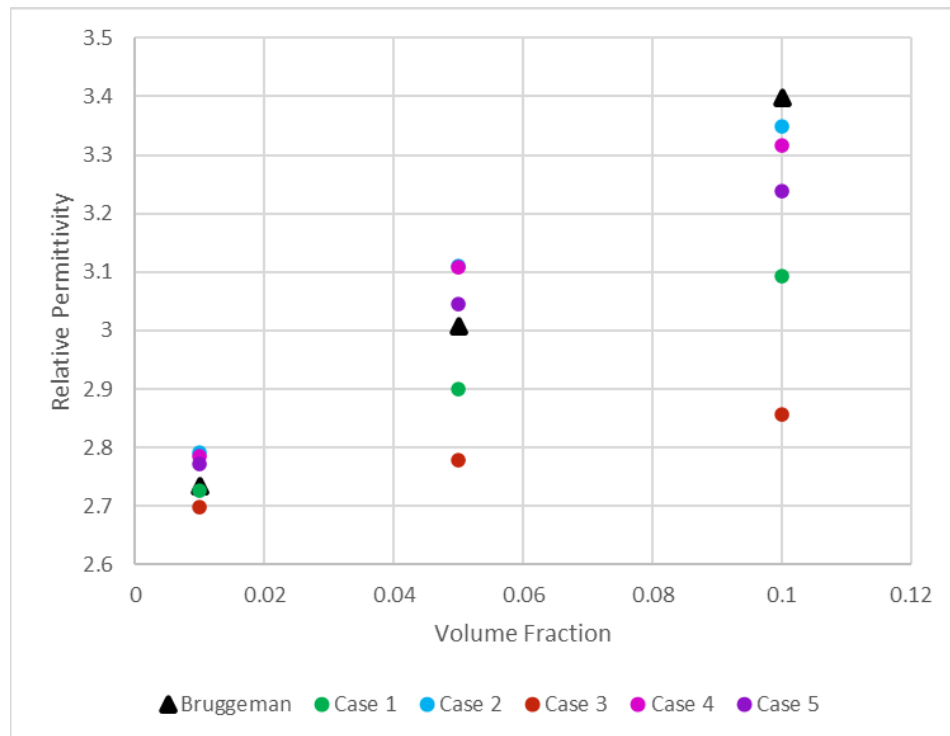


Figure 3-11: DC dielectric Data for 5 cases in the z direction compared to Bruggeman Effective Medium Model for 1%, 5%, and 10% volume fraction

For both the elastic and dielectric domain, the agreement with effective medium theory in the no field case and the transverse direction in aligned cases verifies the model with respect to those theories. The deviation from EMT in the elastic response aligned direction demonstrates the need for computational homogenization for predicting composite properties for varying processing conditions, these hierarchical microarchitectures cause deviation from traditional theory. The trend in the dielectric response matches the expected results, showing higher permittivity values in models with chaining and general agreement with effective medium theory, though distinctions between cases where particle arrangements within chains vary, and that cannot be captured by Bruggeman type models, require computational homogenization.

### **3.3 Comparison of Experimental Results versus Simulation Results**

Thus far computational homogenization results have been compared to analytical models to determine convergence and verify results. Important trends as a result of processing conditions and volume fraction have been presented and discussed in comparison to the effective medium models considered for verification. In the following subsections, the simulation results determined by the computational homogenization methods presented in Section 2.1.2 (and corresponding subsections) for elastic, dielectric, and magnetic properties are presented and compared to experimental results to validate the current computational homogenization model.

#### **3.3.1 Homogenized Anisotropic Elastic Response versus Volume Fraction**

Experimental data is used a reference point for validating simulated results and to understand the behavior of the system being modeled. As mentioned in **Table 4**, the elastic modulus of the PDMS matrix was modified, to 2.1MPa, after initial results were obtained (which were presented in comparison to effective medium theory in Section 3.2.2) to tune the model for comparison to

experimental values. To reduce the computational expense of predicting the effective elastic modulus of the 5 cases (from Table 5), existing simulation results with the updated PDMS modulus (2.1MPa) are used to train a machine learning model to predict the effective elastic properties of the 5 cases of applied fields. Computational homogenization simulation results are presented for the dielectric and magnetic properties in comparison with experimental results. Comparison of the predicted values to the experimental values presented here validate the computational homogenization process for effective elastic properties.

Experimental results for validation are presented in Figure 3-12 for 5wt% and 15 wt% filler content (0.91% and 2.84% volume fraction equivalent) for each of the 5 cases (no field (Case 1), electric field (Case 2), magnetic field (Case 3), electric and magnetic field perpendicular (Case 4), and electric and magnetic field parallel (Case 5)) for tension and compression testing. The y-axis shows the elastic moduli along the tensile or compressive direction determined at approximately 20-23% strain.

A pink box is drawn on Figure 3-12 a to indicate typical values for pure PDMS, which can vary based on curing conditions and the amount of crosslinking [149]. For the 5wt% samples, all of the average values are below the lowest values of the typical PDMS range reported by Wang et al. in their crosslinking study [149]. The PDMS samples tested by Wang et al. use Sylgard 184, manufactured by Dow Corning, with base to curing agent ratios ranging from 5:1 to 33:1, degassing under vacuum, and curing at 65C for 1 hour, to make 2.5-3mm thick samples [149]. The PDMS-BHF samples used for in the experiments for the data presented here also use Sylgard 184 and a 10:1 curing agent ratio, cured at 70C for 12 hours then 120C for 6 additional hours. For a 10:1 curing agent ratio Wang et al. report a 2.61 MPa average elastic modulus for pure PDMS [149].

It would be expected that all cases should be at least as stiff as the pure PDMS samples, however Figure 3-12 a shows that the 5wt% samples are all below the lower limit of the pure PDMS, despite having (stiffer) BHF filler. Earlier simulation runs were conducted with a too low modulus for the PDMS matrix, set at 190kPa, which is shown for comparison with the 5wt% experimental samples in Figure 3-13. Taking into consideration the large error bars in Figure 3-13 a) the behavior of the experimental samples show increased stiffness in the Case 2 and case 5 samples with applied electric field. The experimental results for Case 4 are a bit of an outlier, and only have one sample with no tension data to compare to. It is to be expected that the results in Figure 3-13 b do not show an increased stiffness for Case 3 and Case 5 due to the previously mention spacing concern between particles.

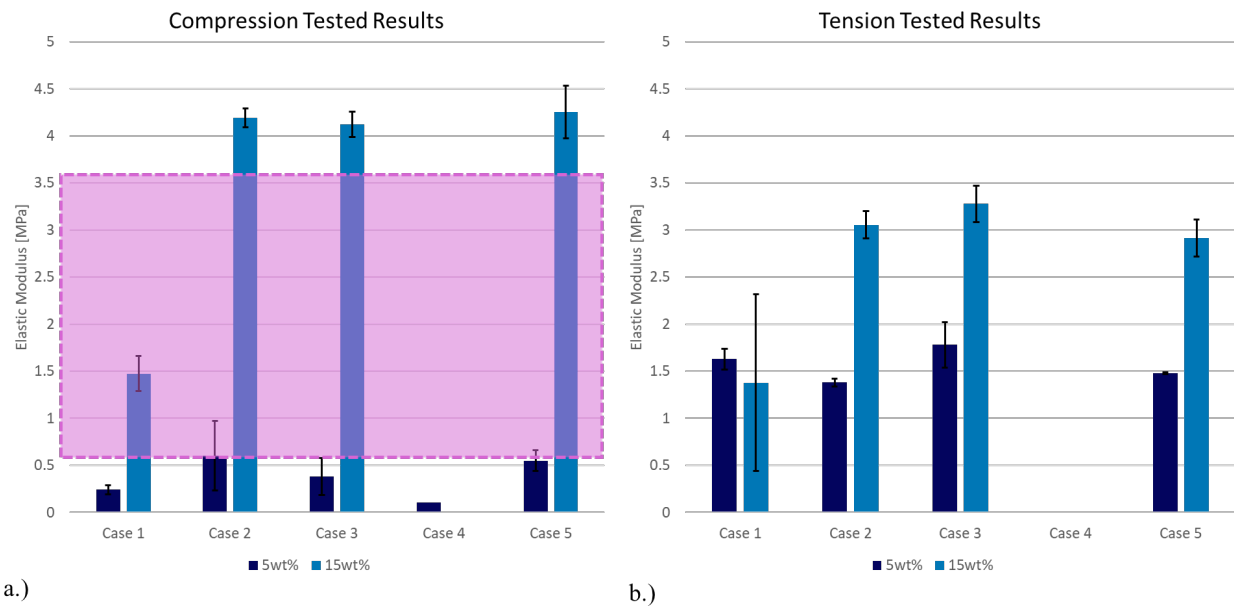


Figure 3-12: Experimental values (from separate samples) for elastic modulus in the direction of the applied electric field (z- direction) for compressive testing at 5 and 15 wt% (equivalent 0.91% and 2.84% volume fraction, respectively) for a) Compression Tests (pink box highlights typical range of moduli for pure PDMS as reference, range reported by Wang et al. [149]) and b) Tensile Tests

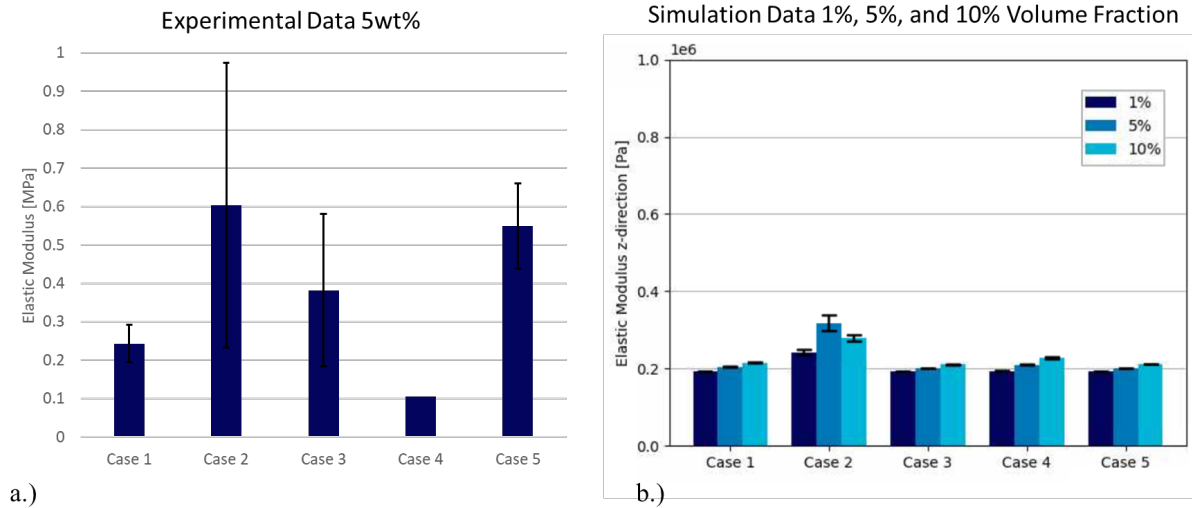


Figure 3-13: Results to compare potentially lower matrix modulus effects on effective elastic results a) Experimental results for compression testing at 5wt% (0.91% volume fraction) b) computation homogenization simulation data from early models with PDMS modulus set to 190kPa at 1%, 5%, and 10% volume fraction

After initial comparison to experimental values, the elastic modulus of the PDMS matrix was adjusted for the computational homogenization to 2.1MPa to tune the model for comparison to the experimental data. Due to the computational expense of the elastic simulations, a machine learning model was trained on simulation data (from the optimization framework presented in Chapter 2 for which results will be presented in Chapter 4). The design space for the machine learning model spans the full electric and magnetic field ranges and 1%-30% volume fraction, initialized with a Latin Hypercube sampling to uniformly test the various parameters' (volume fraction and applied fields) impact on the effective properties. The fit of the model to the training data can be evaluated with an  $R^2$  value, commonly used for regressions. For the machine learning model used to predict the moduli in Figure 3-14, the  $R^2$  value was 0.835, which compared to the 13-14% max error observed in the elastic data presented in Section 3.2.2 is representative of the expected error in the computational homogenization itself. It was estimated that running 3 points for each case and volume fraction would take 720 hours to complete, but could be approximated from data already generated, with the updated PDMS elastic modulus, for

the optimization framework (in Chapter 4) in a matter of seconds. The results of the machine learning model predictions are shown in Figure 3-14 along with the experimental tension data previously presented above in Figure 3-12 b. Understanding that this machine learning model was also trained on simulation results with the particle spacing limited by a spherical distance, the lower results for Case 3 and Case 5 are expected. The no applied field case samples (Case 1) are higher in the machine learning model predictions than the experimental results. However, it should be noted that the error in the experimental results is quite high and could point to non-uniformity, such as agglomeration, that is not seen in the simulated models. On a macro-scale, agglomerations could lead to variation in actual volume fraction when a test sample is cut from the full prepared material. Similarly, better comparison would be possible with a pure PDMS reference value for each sample tested. Interestingly, Case 4 is lower than Case 2, both in the machine learning predictions and the previous simulation results in Figure 3-13, pointing to a potential decrease in stiffness in Case 4 from the thinner chains that are formed with the additional structural ordering within chains as compared to the electric field only Case 2.

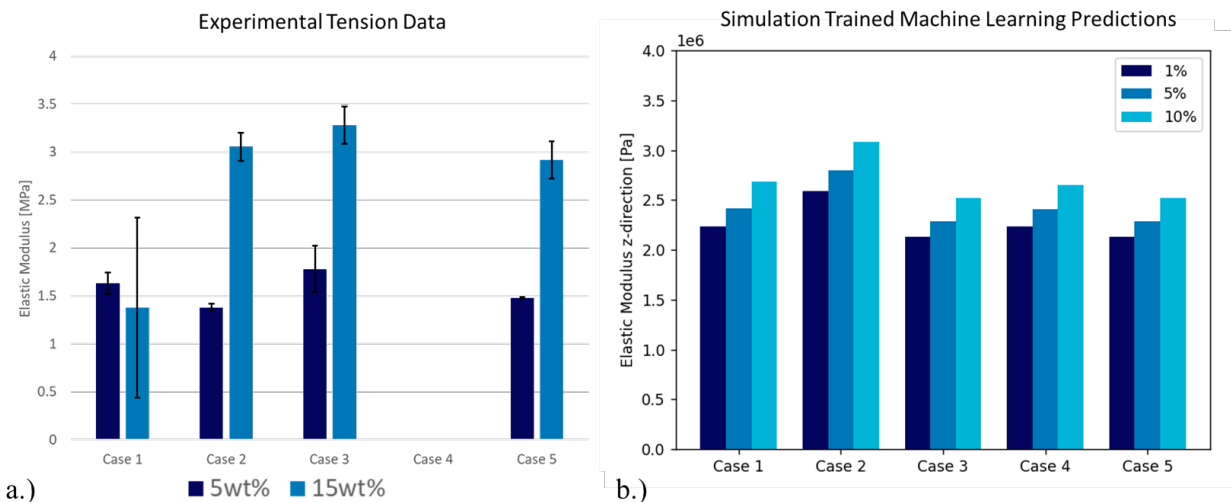


Figure 3-14: Elastic modulus in the z-direction for 5 cases a) Experimental tension test data (5wt% and 15wt%, equivalent 0.91% and 2.84% volume fraction, respectively)  
b) Predicted data from machine learning model trained on simulation data



Comparing the simulation results with the low PDMS modulus from Figure 3-13 and the SVR machine learning predictions in Figure 3-14 shows the relationship between of the properties not only to the matrix modulus itself but the effect of the difference in modulus between the matrix and the filler material. The increased stiffness from the electric alignment of Case 2 has a more pronounced impact in the simulations with the lower matrix modulus than predictions with the stiffer modulus. As the values of the matrix and filler moduli converge the sample would trend to one uniform property independent of structure.

In this section experimental values for elastic modulus of 5wt% and 15wt% were presented and compared to simulated values from the computational homogenization scheme presented in Chapter 2 and a machine learning model trained on the produced simulation results. Modeling efforts show good agreement with experimental values, with known deviations due to particle spacing and possible variation in matrix modulus in experimental samples. These results not only demonstrate the effectiveness of the computational homogenization but also show the efficiency and necessity of using a machine learning model to augment simulations for an optimization framework.

### **3.3.2 Homogenized Anisotropic Dielectric Response versus Volume Fraction**

Using methods presented in Section 3.2, computational homogenization results of dielectric properties for architectures determined from particle dynamics simulations are presented in this section for validation against experimentally determined properties. All 5 cases are presented in this section for dielectric AC response for models with 100 particles and volume fractions at 1%, 5%, and 10%. Experimental filler content is shown at 5wt% and 15wt% which correspond to 0.91% and 2.84% volume fraction, respectively.

Figure 3-15a shows the results of experimental testing for 5wt% (0.91% volume fraction) in the z direction. Due to the thin nature of the samples this is the only direction that can be tested experimentally. The processing cases with chaining due to electric fields (Case 2, 4, and 5) have a 23-84% higher permittivity compared to Case 1. The permittivity for Case 3, with no electric field, is only 11% higher than that of Case 1.

The simulation results in Figure 3-15b show the electric field only case (Case 2) and perpendicular field case (Case 4) have the highest values. Case 3 at 1% volume fraction has one outlier leading to the large error bars, this is likely exacerbated by the spacing between the particles, leading to one erroneous value that is less likely to occur at higher volume fractions. Case 3, which only has a magnetic field, and Case 5, which favors the magnetic field with the relative strengths used, both show lower permittivity in the magnetic field direction. This result is to be expected with the previously presented spherical spacing used in these runs to handle physical collisions between particles. This additional space between particles in chains in Case 5 lowers the values compared to Case 2 and 4, seen in Figure 3-15b. The error bars are seen to collapse with higher volume fraction, likely due to reduced variation in possible particle arrangements from decreased mobility, reinforcing the notion that the permittivity is particularly sensitive to inter-particle spacing. At 10% volume fraction, where the error bars are small, the same trend of improved permittivity in cases with chaining (Case 2 and 4) over Case 1 that was seen in the experiments is seen, understanding that Case 5 has magnetic alignment where the particle spacing issue arises again. Additionally, there exists a trend showing increased permittivity with increased volume fraction in the simulation results that requires additional experimental data for comparison. Future work should also include dielectric loss in experimental results to compare the complex permittivity of the simulations results, laid out in

Section 2.1.2.2. Comparing the dielectric loss for the experimental and simulation results could give additional insight to the variation in magnitude of the results present in in Figure 3-15.

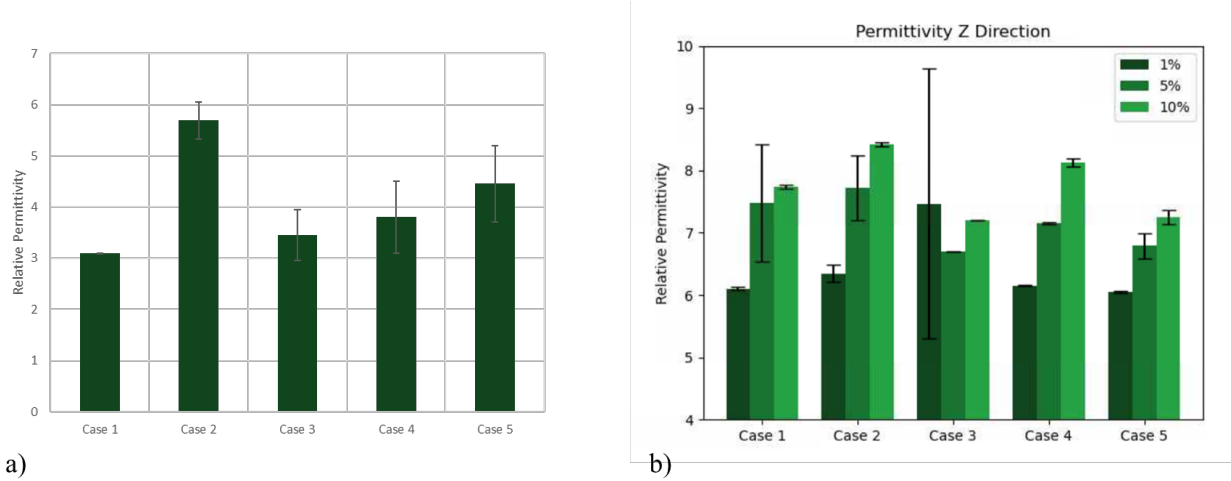


Figure 3-15: Effective dielectric properties a) experimental (5wt%, ~1% volume fraction) showing higher permittivity values for electrically aligned cases (2, 4, and 5) and b) simulation (1,5, and 10% volume fraction) results in the z direction (electric field direction)

Comparing the magnitude of the values between the experimental and simulation results, even for the random distribution of Case 1 it can be seen that the experimental value is lower than the simulation results. The particles are coated in a surfactant to prevent agglomeration which is not accounted for in the simulations, adding an extra layer of nonconductive material with its own permittivity value that could differ from the PDMS. As shown in (3-1) the polarization,  $p_{eff}$  is a function of the difference in permittivity of the matrix and the filler material due to the buildup of charge on the surface of the material [143].

$$p_{eff} = \frac{4\pi abc}{3}(\epsilon_2 - \epsilon_1)E^- \quad (3-1)$$

In equation (3-1)  $a$ ,  $b$ , and  $c$  are the axes of the filler particle, and  $E^-$  is the depolarization field, which is a shape dependent response of the particle to the applied field (further discussed in Section 4.1). The thickness of the sample also plays a crucial role in determining the experimental values, which can be adversely affected by non-uniformity in the thickness of the sample. Particle contact that is unique to experimental samples contributes to conduction or dielectric loss, also lowering the permittivity measured.

Comparison of the experimental and simulation results serve to validate the simulations, outside of instances where known issues with model fidelity intrude. For example, Cases 1, 2, and 4 are able to recreate trends of increased permittivity with chaining for 10% volume fraction. Though magnitudes of permeabilities may differ between simulation and experiment, consistent trends suggests that the mechanisms of property evolution are accurate, and that these differences in absolute magnitude may stem from not accounting for surfactants and other chemical species clouding the pure PDMS-BHF properties used in the simulations.

### 3.3.3 Homogenized Anisotropic Magnetic Response versus Volume Fraction

The computational homogenization presented in Section 2.1.2.3 determines the unpoled magnetic response of a composite. Particle architectures used in the homogenization are determined from simulations containing 100 particles, where the particles and matrix have properties given in Table 4. The simulated values are compared to experimental data for validation. Experimentally the samples are exposed to an increasing magnetic field until the sample saturates (at applied fields on the order of 1.5MA/m) after which the field is reversed to map the hysteresis response, with remanent magnetization determined when the applied field returns to zero. This process poles the sample, resulting in an experimental remnant magnetization in contrast to the simulation's magnetization that does not account for the history

of applied fields. The data presented here focuses on the trends for comparison and validation between the experiments and the simulations.

Experimental samples can vary in actual volume fraction when a section is cut from a whole due to the non-uniformity of the particles throughout the sample. Figure 3-16 shows the remanent magnetization for 5 cases for 5 and 15wt% and saturation magnetization for cases 1,2,3 and 5 with a target filler content of 15wt%. No Case 4 samples had been made for experimental testing at the time of this document. The observed differences in saturation, however, suggest that the filler content varies between the samples. From the vector sum method presented in Section 2.1.2.3, the idealized saturation for fully aligned magnetic domains has been calculated to be 70 emu/g\*. This property is normalized to particle mass and is therefore independent of volume fraction. Discrepancies between target filler content and actual filler content in a sample can occur from non-uniform distribution of filler, which when samples are cut into test specimens can result in an altered filler content for testing. Deviation from the target filler content manifests as reduced or increased saturation when experimental values are mass normalized with the target filler content value. The normalized value and the idealized value for the saturation magnetization can then be used to scale the experimental values to adjust for actual filler content in the prepared sample from the targeted weight percent. The data presented in Figure 3-16 b, experimental saturation magnetization normalized to the target particle mass, is used to adjust the raw data presented in Figure 3-16 a, remanent mass magnetization, to account for these differences in target versus actual filler content. The 5wt% data does not have saturation values available and therefore does not make this adjustment. To compare the remnant magnetization in experiments to the mass magnetization of the unpoled samples in simulation a difference between the z (out of plane) and x/y (in plane) direction components for each is

presented in Figure 3-17, noting that 5wt% and 15wt% are approximately 0.9% and 2.97% volume fraction, respectively.

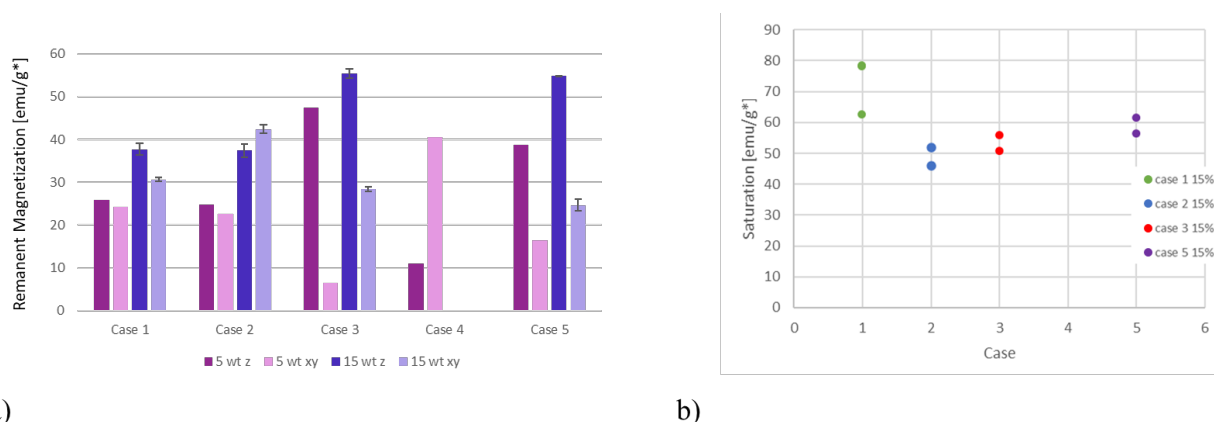


Figure 3-16: a) Remanent mass magnetization and b) Saturation magnetization for Case 1,2,3, and 5 for experimental samples with a target of 15wt% (2.84% volume fraction) filler content normalized to particle mass (g\*)

From Figure 3-17 a and b it can be seen that Cases 1 and 2 are nearly magnetically isotropic in both experiments and simulations compared to Cases 3,4, and 5, which all have some form of magnetic alignment during processing and a resulting enhanced anisotropy here.

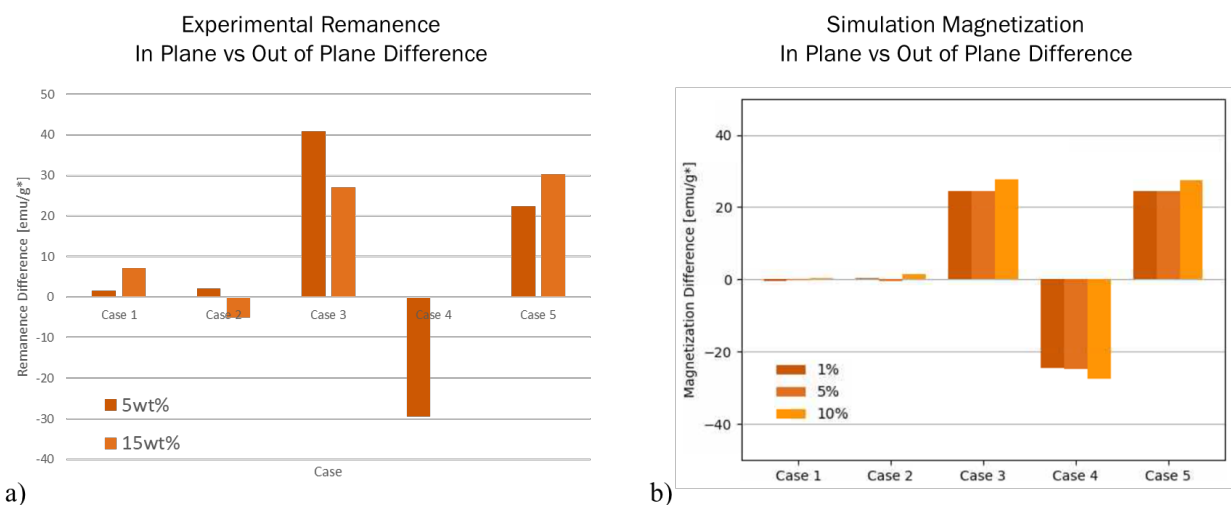


Figure 3-17: Comparison between the in plane and out of plane magnetization presented for a) adjusted experimental values for 5 and 15wt% and b) simulation results for 1,5, and 10% volume fraction

It should be noted that Case 4 is negative due to the x/y (in plane) direction having the applied magnetic field, and the values shown is the difference between the z direction and x/y direction. The varying orientations of the magnetic dipoles in case 1 and 2 (that do not have an externally applied magnetic field during processing) sums to 0 for the overall composite, resulting in an isotropic composite. In the case of the experimental samples for these two cases, some of the domains have been flipped during saturation when running through the full hysteresis loop and remain in a more ordered state than they started in, resulting in an overall remnant magnetization as presented in Figure 3-16a. It should also be noted that the simulations assume a single domain particle with a unidirectional magnetization. The purpose of these simulations is to predict trends in composite properties to maximize the range of properties possible by varying the applied electric and magnetic processing fields. The purpose of this validation is to show that this behavior is also seen in experiments. The simulations predict an anisotropic response in the cases with magnetic field processing (Cases 3, 4, and 5), which is also seen in the experimental results. Additionally the simulations predict a nearly isotropic response in Case 1 and Case 2 that were not processed with a magnetic field, which is again also seen in the experimental data.

### **3.4 Particle Spacing and Resulting Model Fidelity**

Predicted effective properties are a direct result of the particle distributions found from the applied processing fields. In order to improve fidelity and agreement with experimental results, the issue of the spherical, initially employed, spacing was addressed. This section presents the resulting distributions of particles using the ellipsoidal spacing described in Section 2.1.2.5.

Figure 3-18 shows the resulting distributions using the spherical (equation (2-11)) or ellipsoidal (equation (2-18)) distance in the repulsive force for models processed with a  $1\text{e}8$  V/m electric field and a  $250,000$  A/m magnetic field both applied in the  $z$  direction on a 1% volume fraction model with 100 particles. The values of  $\beta$  are varied to see the effect of this calibration term ( $\beta$ ) on the resulting distribution. The results for the spherical distance in Figure 3-18 shows that at lower  $\beta$  values, particularly for  $\beta = 2$ , the minimum spacing between particles tends towards radius of the sphere assumed in the repulsive force calculation of equation (2-11). The vertical gap between particles is essentially the particle's in plane diameter. With larger values of  $\beta$ , the repulsive force decays more quickly as the particles separate, allowing the particles to be closer than their in-plane (major) diameter. Although the closer spacing is an improvement for particles with aligned minor (out of plane) axes, particles would potentially overlap when aligned along their major axes, given the spacing is now less than that distance.

Using the ellipsoidal distance in the repulsive force calculations, the particles' orientations relative to one another are considered when calculating the repulsive force between them, e.g. the proximity of the nearest point on their ellipsoidal bounds. The distributions in Figure 3-18 for the ellipsoidal distance show that particles come closer together when aligned on their minor axes without universally reducing the spacing for all directions and orientations the way the spherical distance calculations do. The  $\beta$  parameter is also shown to have less effect on the results than in the spherical distance case.



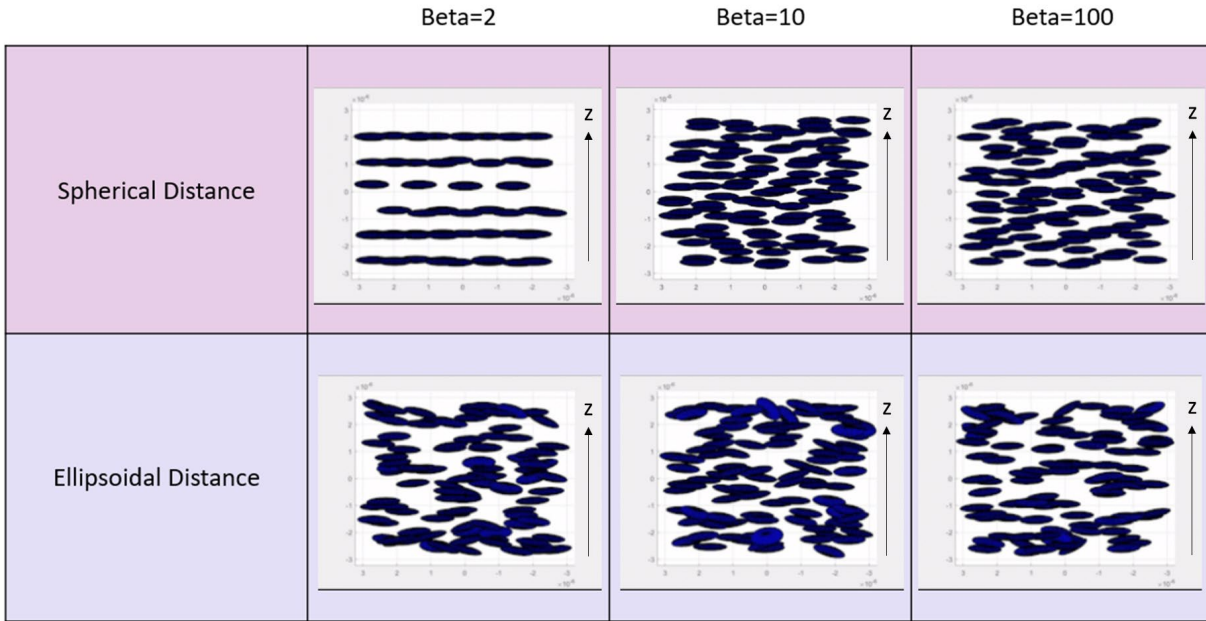


Figure 3-18: Particle distributions for calibration term  $\beta$  at 2, 10, and 100 with spherical and ellipsoidal distance calculations for determine the repulsive force between contacting particles

These distributions are more representative of the way physical particles would interact with each other in a composite and show promise for bridging the gap between simulations and experiments where the spacing is critical to the effective property, such as in Case 3 samples as discussed in previous sections. The way the current particle dynamics simulation handles the particles positions and orientations and determines convergence leads to long run times with this new spacing, and becomes the bottleneck of the optimization framework. Incorporation of this new spacing into the current scheme requires reworking the particle dynamics simulation's handling of convergence.

The effect of the distance and resulting repulsive force calculation on particle distribution was shown. The spherical distance usage results in separation between particles that is not representative of how ellipsoidal particles would interact when coming in contact. The new ellipsoidal distance calculation and proposed repulsive force show improvement in removing

arbitrary spacing between ellipsoidal particles but are more costly to run in the current particle dynamics scheme (going from roughly an hour to more than 12 hours to solve). The change in the repulsive forces alters the way time steps are determined within the particle dynamics simulation. Ultimately, manual calibration or reworking the framework's handling of time discretization and convergence is required to incorporate the ellipsoidal distance force calculation. In the results presented thus far the spherical distance and force calculation is used and will also be used for the optimization framework in subsequent chapters due to the current added computational expense. Revisiting the particle dynamics simulation in future work to modify handling of time steps and convergence will improve this run time and allow the ellipsoidal spacing to be used in calculating repulsive forces between particles, ultimately improving the fidelity of the computational homogenization itself.

### **3.5 Artificial Intelligence Framework for Predicting Conductivity**

The results for the approach to predicting conductivity presented in Section 2.3 are presented here as a reduced computational expense alternative to homogenization using FEA. The search algorithms efficiency and tunability are presented in comparison to an exhaustive search. The search algorithm is verified on a test case, to ensure the functionality of the framework to find conductive pathways as intended. Some sample points are presented to demonstrate the use of the framework in conductivity calculations and the dependency of the results on the physics used. The framework presented here lays the groundwork for homogenizing composites with highly conductive filler materials for which FEA is not well suited.

The goal of an AI pathfinding algorithm is to accurately represent the conductive pathway in an RVE. Graph representations such as the minimum spanning tree only capture the relative distances of the single closest neighbor to each particle, and do not capture the overall

connected nature of the network of filler particles. Similarly a random search such as with commonly used Monte Carlo techniques may miss key pathways, particularly near percolation. While an exhaustive search could be done, it would be limited to very few particles. Using a search algorithm and pruning, or terminating, search paths when there is no reasonable way forward is a powerful way to reduce the computation to a feasible run time. As an example of the computational strength of AI search in this application a volume of 100 particles is considered. Each pathway is a permutation of the particles in the RVE, choosing between 1 and all of the particles. The total number of possible paths to check then would be the sum of all of these permutations,

$$Total\_Paths = \sum_{r=1}^n nP_r \quad (3-2)$$

where n is the total number of particles. Expanding equation (3-2) out with the definition of the permutation formula, this yields equation (3-3).

$$Total\_Paths = \sum_{r=1}^n \frac{n!}{(n-r)!} \quad (3-3)$$

For 100 particles equation (3-3) yields  $3e158$  possible pathways, which is unfeasible to check. If the path length is limited to 22 particles in length this would reduce the total pathways to check to  $8e42$ , still unfeasibly large. If filtering is used and the network were reduced to the 61 particles of the 100 that are found to be within the cutoff distance, and the path length was still set to a maximum of 22, this would reduce the total number of pathways still only to  $3e37$ . In order to reduce the possible pathways enough the path length would have to be limited to 8 or 9 that reduce the total pathways to  $1e14$  and  $6e15$ , respectively. This would still take several hours to days to search through, as compared to the seconds for the AI search technique. The algorithm

presented here, all paths search (APS), is a modified depth first search that does not consider loops, finds nearest neighbors within a defined cutoff distance, and has the option to limit pathlength. APS is integrated into a framework that utilizes object oriented programming to allow for individual particles to have different properties, such as material or geometric properties.

To show the novel nature of this approach as opposed to spanning trees and Monte Carlo methods used previously to determine if percolation was achieved, the search algorithm presented here is demonstrated on an easy to follow graph test case previously presented in Section 2.3 and shown again here with a complimentary matrix form in Figure 3-19. The matrix form shown is used to communicate the connections that have been found in the search algorithm to the network construction function which utilizes MATLAB's Simulink tool. The graph shown has green lines to represent the nodes that connect the start to the finish and would conduct electricity between these, while the gray lines connect nodes but do not conduct electricity between the start and finish nodes and are therefore not considered in the conductive paths results. This example will also ensure that there are no missing paths or early terminations once a path has been found by also requiring backtracking to the start node to continue finding paths. Finally, this test case was constructed specifically to test the functionality of the search algorithm and ensure that there are no loops returned, such as A-C-D-B-C-E, which is verified as shown. The test case ensured the functionality of the search algorithm by testing various failure modes.

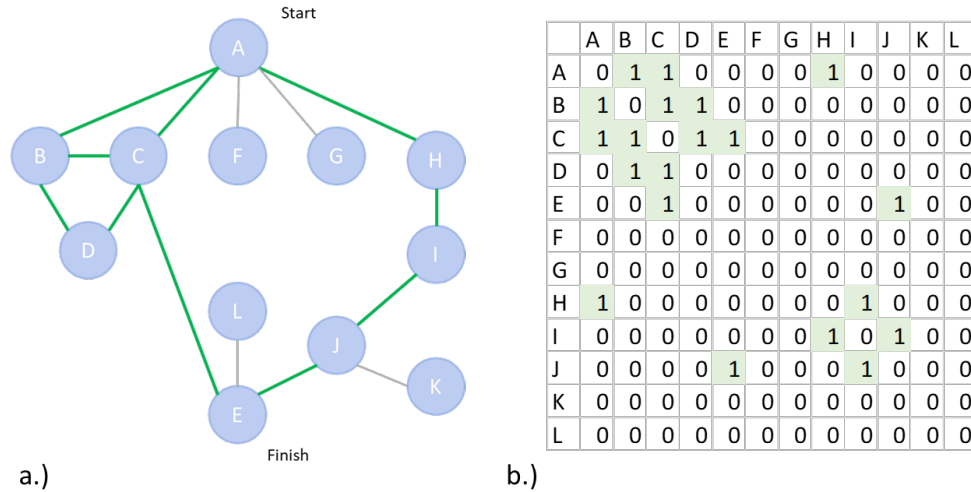


Figure 3-19: Representations of all conductive pathways for simple test case a) connect graph showing all conductive pathways and b) equivalent matrix form of the solution

The results of the search on this test case are shown in Figure 3-20 in command line output form. The algorithm correctly found all of the green paths highlighted in Figure 3-19 while not returning any loops or nodes not connected by green lines. This validates the functionality of the search algorithm on this type of problem. The same framework is then used on 3 dimensional position data for spherical particles to find conductive pathways between electrodes on the top and bottom face of the RVE. The results for the APS framework on a particle network are shown in Figure 3-21.

```
['A', 'B', 'D', 'C', 'E']
['A', 'B', 'C', 'E']
['A', 'C', 'E']
['A', 'H', 'I', 'J', 'E']
connectivityMatrix:
[[0 1 1 0 0 0 0 1 0 0 0 0]
 [1 0 1 1 0 0 0 0 0 0 0 0]
 [1 1 0 1 1 0 0 0 0 0 0 0]
 [0 1 1 0 0 0 0 0 0 0 0 0]
 [0 0 1 0 0 0 0 0 0 1 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0]
 [1 0 0 0 0 0 0 0 1 0 0 0]
 [0 0 0 0 0 0 0 1 0 1 0 0]
 [0 0 0 0 1 0 0 0 1 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0]]
```

Figure 3-20: Results from simple test case showing the conductive pathways found by the search algorithm in command line output format

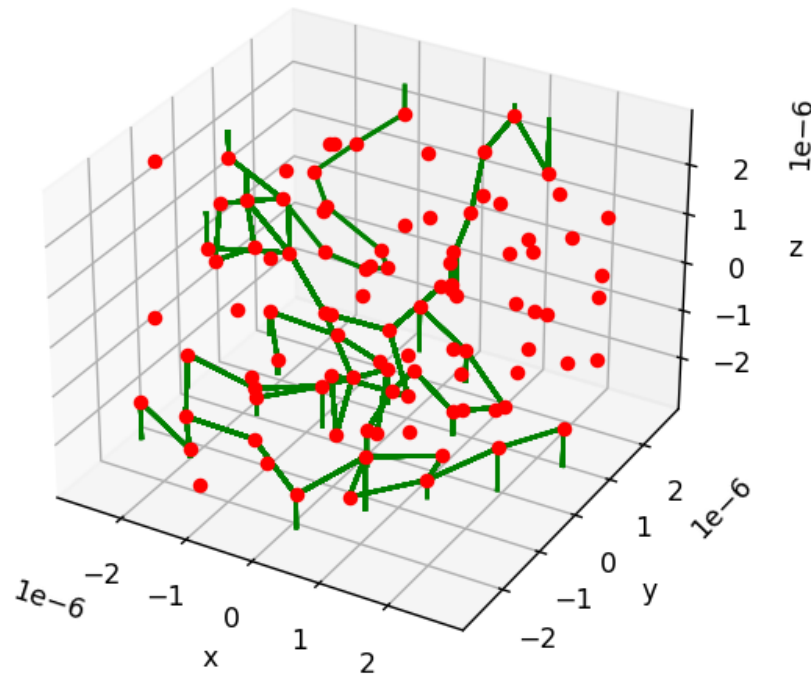


Figure 3-21: Pathfinding algorithm results showing "all paths" graph, where conductive paths are shown in green between particles whose centers are shown by the red points for an RVE with 100 particles and 10% volume fraction

The focus of this comparison is to understand the behavior the framework shows and evaluate the usefulness of the search algorithm incorporated in this workflow. To compare the results of this framework to experimental data a simple case was found in the literature. The samples selected were fabricated by Chen et al. and are Nickle Polystyrene particle composites which were tested for resistivity as a function of volume fraction [150]. For comparison, 14%, 20%, and 30% volume fraction (60, 70, and 80 wt%, respectively) were simulated, assuming a random distribution of particles and 1nm diameter spheres. The conductivity of the particles was set to  $6.4 \times 10^{-8}$  Ohm-m. The conductivity of the Polystyrene used was not indicated in the manufacturer's specifications and estimated at  $1 \times 10^{19}$  Ohm-m. For each volume fraction 10 points are tested, the results of which are plotted in Figure 3-22.

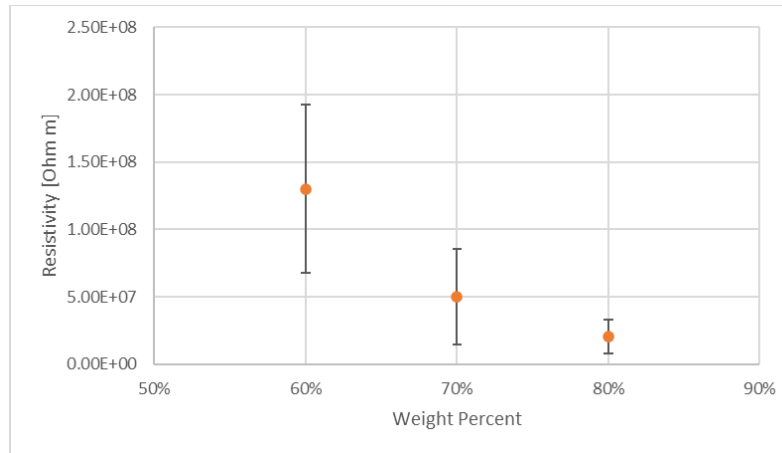


Figure 3-22: Simulated resistivity results for 1nm diameter spherical Nickel-Polystyrene composite at 60, 70, and 80 wt% (14%, 20%, and 30% volume fraction, respectively)

The experimental results from the literature for Nickel powder unaligned composites are shown in Figure 3-23, note this plot is on a log scale in the y-axis. From these results it can be seen that at lower filler content, such as 60 wt% and below, for the powder, the resistivity matches well with the simulated results, on the order of  $1e8$  Ohm-m. The values begin to quickly drop off in the experimental results to  $1e6$  Ohm-m at 70 wt% and below  $1$  Ohm-m at 80 wt%. When examining the image in Figure 3-23 for the 80 wt% particle distribution, it can be seen that the particles are not uniformly distributed and cluster together. It should also be noted that there are no error bars presented on this data to indicate variability between samples.

The simulation results show good agreement with the lower volume fraction cases such as 60 wt%. The effective resistivity found in both cases somewhat diverge at higher filler content, however. It should be noted that some RVE's at 60 wt% did not have a conductive pathway found with the metrics used, suggesting also that this amount of filler content may be near the percolation threshold. In testing of the framework some instances of 80 wt% and higher were also noted to have particle arrangements that formed short circuits. This is an interesting distinction to note, particularly when comparing to the experimental results. The particle clusters

seen in the image in Figure 3-23 suggest a potentially similar mechanism in the experimental results as the short circuit values seen in simulation. These seemingly outlier simulation instances returned values less than 1 Ohm-m. The distributions that resulted in such low resistivity results were found during a separate test set, using different search parameters, from the data presented in Figure 3-22, and could therefore not be included there. It is possible then that the experimental results from Chen et al. show a rapid drop on resistivity with increased filler content because of nonuniform particle distribution and clustering. It would be necessary to study the experimental particle distributions to recreate similar clusters in simulated distributions for better comparison. The simulated results show the effect of increased filler content decreasing the resistivity, suggesting that the path finding algorithm finds connections between particles that are representative of the conductive network in the composites. The model size used was 100 particles and could also be investigated in the future with correlation to experimental distributions to ensure that the model size is able to represent any structuring or clustering found in experimental samples.

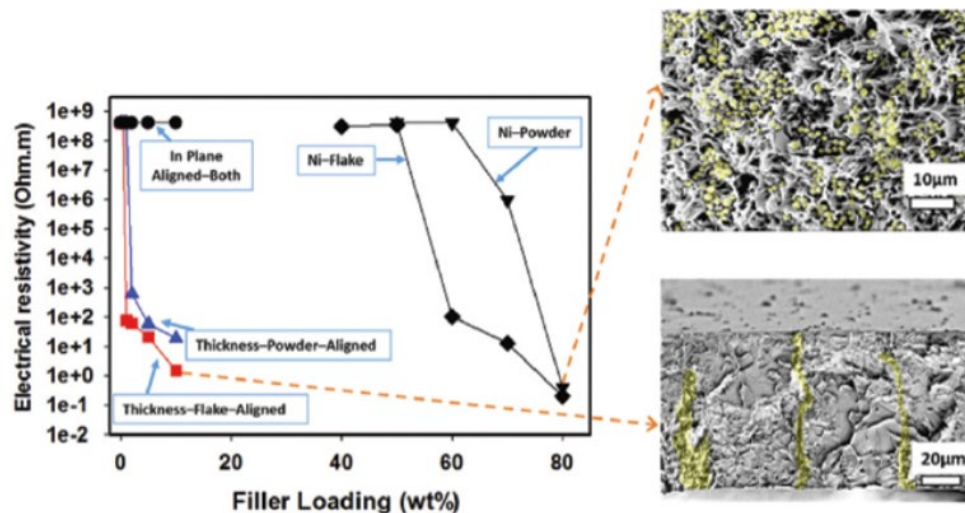


Figure 3-23: Resistivity data from Chen et al. [150] showing the effect of volume fraction on resistivity of various Nickel-Polystyrene composite compositions used as a reference for comparison with simulated conductivity results



The simulated results presented in Figure 3-22 show good agreement with near percolation values and a drop in resistivity by 2 orders of magnitude from 60 to 80 wt% filler content. This framework shows potential for future integration into finding percolation paths and predicting conductivity of composites. The strengths of this framework are that the particles are controlled using individual objects that can be modified to have different sizes and shapes while efficiently searching for physics representative pathways through an RVE.

### **3.6 Summary**

In this chapter the initial results for analytical and computational homogenization were presented and compared. Verification with effective medium theory has been shown for elastic results. Dielectric results have been compared to the Bruggeman model for the DC case. A convergence study was shown to quantify the effect of model size on the effective elastic properties. Elastic, AC dielectric, and magnetic simulation results were presented and validated with experimental data comparison. It was shown that the computational homogenization can determine trends in composite behavior, which is sufficient for optimization where the trends can still point to the design space extrema. However, the model can be further calibrated for specific values with further experimental data. Finally, the results of the novel All Paths Search (APS) framework incorporating AI search into a homogenization scheme for predicting conductivity was presented and compared to data from the literature. The homogenization results presented here are the foundation for the optimization framework presented in Chapter 4.

## CHAPTER 4: Multi-Objective Optimization Results and Discussion

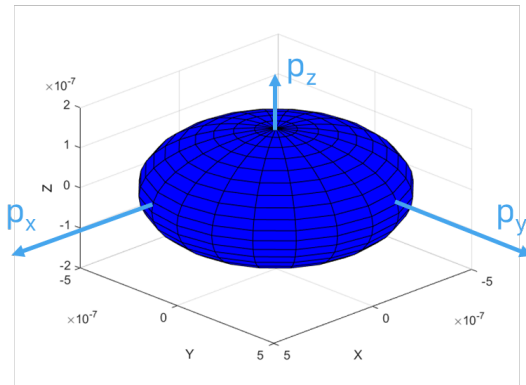
This chapter uses the framework and methods described in Chapter 2 and the homogenization results presented in Chapter 3 to evaluate the multi-objective optimization scheme and determine the constituent set that will provide the largest range of possible properties by varying processing condition alone. The results presented in this chapter underline the need for such a framework that can search and determine the effective properties of multi-field processed composites. The effects of varying the respective orientation and magnitudes of processing fields show the non-trivial nature of the design space. Additionally, results from the optimization scheme in this framework will be shown, both for single properties as well as a combined property multi-objective optimization. Single property objective results were presented, in part, at the ASME Conference on Smart Materials, Adaptive Structures, and Intelligent Systems in 2022, *Multi-Objective Optimization of Predicted Magnetic Properties from Multifield Processing Conditions in Polymer Matrix Particle Composites* [151].

Optimizing the possible range of properties fundamentally seeks to determine how to apply processing fields to achieve desired microarchitectures. Those field, in turn, act on particle to align them in the field direction at the chain and particle level. These spatial rearrangement invariably involve rotations of particles due to the applied electric and magnetic fields. However, these fields may act in concert or in opposition, given the five processing Cases outlined in Section 3.2. The combined action of the two fields, when applied simultaneously, leads to the concept of a 'cross over torque', herein defined as the point at which the applied electric and magnetic field are equal and opposed to one another. The resulting microarchitecture changes on either side of this point, demonstrating the importance of tuning relative field strengths and orientations over the discrete five cases when optimizing the possible range of properties.

## 4.1 Crossover Torque

The relative strength and orientation of the applied fields has been shown to affect the resulting particle distribution. Particularly when considering the electric and magnetic fields applied in the same direction. The shape of the ellipsoidal particle, specifically the shape anisotropy, leads to differences in the components of the polarization vector, which results in a torque on the particle to align with the applied electric field. Similarly, the permanent magnetic dipole of BFH results in a torque on the particle to align the dipole with the externally applied magnetic field. Each field attempts to align the particles as shown in Figure 4-1. When these fields are oriented in the same direction the torques generated oppose each other and the fields compete for preferential orientation. There then exists a point at which these competing torques equal each other, and the system is conditionally stable where small deviation from this point (orientation or field strength) favor one or the other. This point is therefore referred to as the crossover torque. The generated torques are a function of the applied field with respect to the particle, the material properties, and the geometry of the particle itself that are considered in this balance. The effect of the relative strengths of these competing fields drives the resulting particle distribution and therefore effective properties.

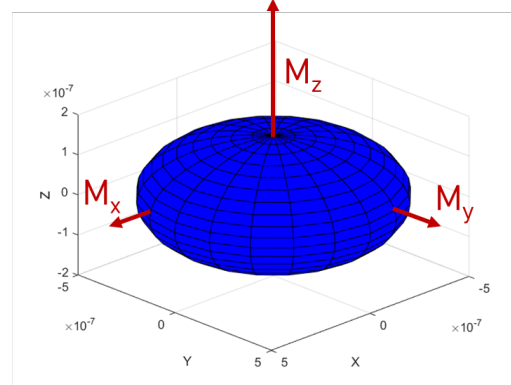
The concept is further illustrated with simulation results in Figure 4-2, showing resulting particle distributions for 1%, 5%, and 10% volume fraction in sub columns with parallel applied magnetic and electric fields ranging in magnitude from 0-500,000 [A/m] down the rows and 0-100 [MV/m] across grouped columns, respectively. The purpose of this figure is to demonstrate the changes in microarchitecture from changes in strengths of the two fields applied in parallel, demonstrating the principle and effects of the crossover torque.



a)

$$\mathbf{T}_e = \mathbf{p}_{eff} \times \mathbf{E}_0$$

Dielectric torque



b)

$$\mathbf{T}_m = \mathbf{M} \times \mathbf{H}_0$$

Magnetic torque

Figure 4-1: Resulting torque from a) polarization from externally applied electric field and b) magnetization from magnetic dipole and applied magnetic field (vector quantities vary with the direction of the applied field with respect to the particle orientation)

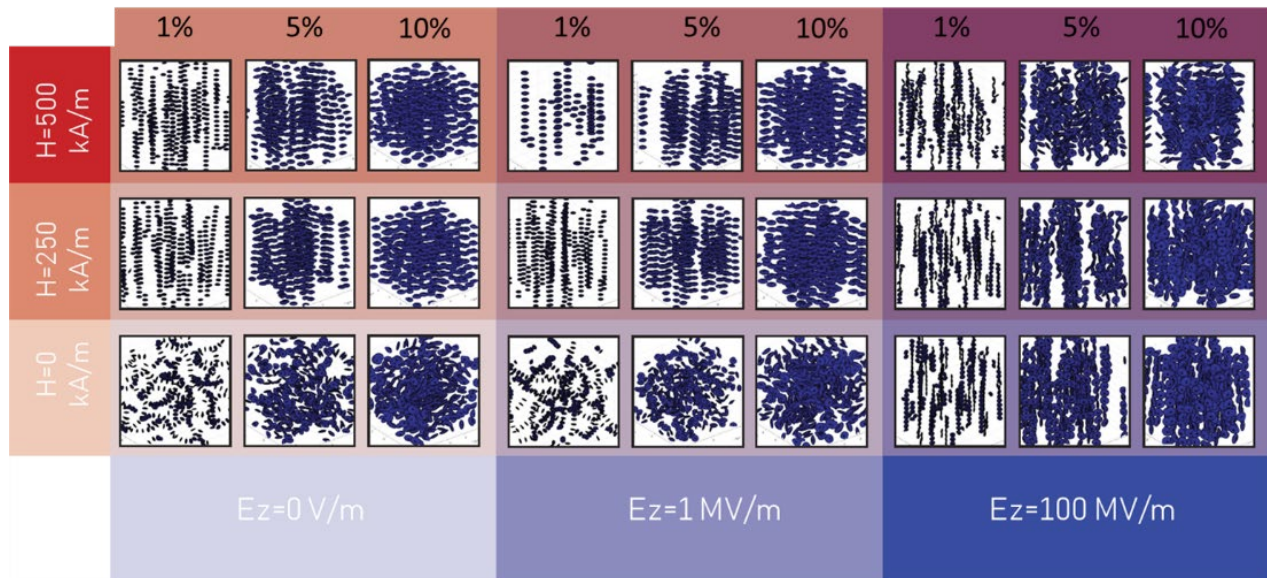


Figure 4-2: Particle distributions for systems with varying combinations of electric and magnetic field applied in the z direction (vertical direction in each figure), showing the competing nature of the torques generated from the two fields to orient the particles with the respective field

An example of this change is seen by observing the middle row at  $H=250,000$  A/m. The example holds for any of the volume fractions; however, 1% volume fraction is easiest to discern the particle orientations. In the first column (no electric field) and 1% volume fraction, for

250,000 A/m magnetic field, the particles align their magnetic dipole moment with the applied magnetic field. In the second grouped column (electric field of 1MV/m), the particle still align their magnetic dipole moment with the applied magnetic field despite the presence of an electric field, due to the magnitudes of the respective fields. In last grouped column (electric field of 100MV/m), for the 1% sub column, the particles now align their major axis with the applied electric field due to the electric polarization induced in the particles by this field, despite the continued presence of the 250,000 A/m magnetic field, a change in the preferred orientation from the applied magnetic field direction toward the electric field direction. These same processing fields and the effect on particle distributions is repeated in Figure 4-3 to illustrate this change in closer detail. The change in particle orientation is clear in Figure 4-3 by increasing the electric field from 1 [MV/m] to 100 [MV/m]. The torque the electric field generated on these particles was strong enough to overcome the torque the magnetic field generates on the particles and the resulting orientations crossed over from magnetically aligned to electrically aligned. At a higher magnetic field of 500,000A/m it can be seen that this crossover is not as strong for the 100MV/m electric field because the larger magnetic field has a stronger torque that is able to compete better with the electric field. For this reason a continuous, rather than discrete, exploration of the design space is necessary, as the 5 general cases manufactured experimentally to date, do not cover the intricacies of the combined fields between the extrema.

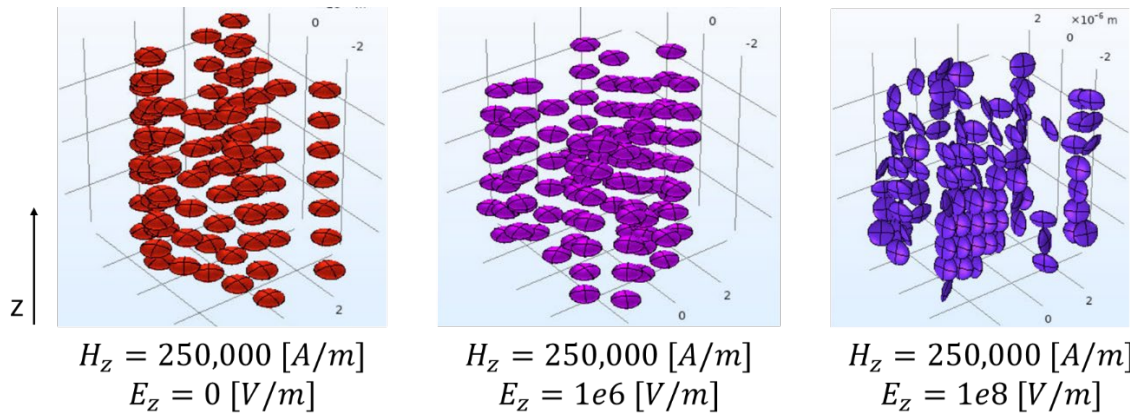


Figure 4-3: Particle distributions example for 1% volume fraction with 100 particles a 250,000 A/m magnetic field in the z-direction and increasing parallel electric field from 0-1e8 V/m showing the effect of field strength on microarchitecture and demonstrating the crossover torque

To better understand the interaction between the effects of the two fields, the generated torques for the two fields can be calculated as a function of field strength and particle orientation to determine where this crossover point is. The resulting torques versus applied field are plotted in Figure 4-4, using equation (4-1) for the magnetic field generated torque and equations (4-2)-(4-5) for the electric field generated torque, using the magnetic dipole moment,  $\mathbf{M}$ , the dimensions of the major axis,  $a$ , and minor axes  $b$ , and  $c$ , of the particles along with the matrix and particle relative permittivity,  $\epsilon_1$  and  $\epsilon_2$ , respectively.

$$\mathbf{T}^m = \mathbf{M} \times \mathbf{H}_0 \quad (4-1)$$

$$\mathbf{T}^e = \mathbf{p}_{eff} \times \mathbf{E}_0 \quad (4-2)$$

$$\mathbf{p}_{eff} = \frac{4\pi abc}{3}(\epsilon_2 - \epsilon_1)\mathbf{E}^- \quad (4-3)$$

$$\mathbf{E}^- = \begin{bmatrix} \frac{E_{0,x}}{1 + \left(\frac{\varepsilon_2 - \varepsilon_1}{\varepsilon_1}\right) L_x} \\ \frac{E_{0,y}}{1 + \left(\frac{\varepsilon_2 - \varepsilon_1}{\varepsilon_1}\right) L_y} \\ \frac{E_{0,z}}{1 + \left(\frac{\varepsilon_2 - \varepsilon_1}{\varepsilon_1}\right) L_z} \end{bmatrix} \quad (4-4)$$

$$L_x = \frac{c}{2a} \left[ \frac{\pi}{2} - \frac{c}{a} \right]$$

$$L_y = \frac{c}{2a} \left[ \frac{\pi}{2} - \frac{c}{a} \right] \quad (4-5)$$

$$L_z = 1 - \frac{c}{a} \left[ \frac{\pi}{2} - \frac{c}{a} \right]$$

It can be seen that when the fields are applied in the same direction ( $\theta = \frac{\pi}{4}$ ) that there is a crossover torque just above  $6 \times 10^7$  [V/m] and 150,000 [A/m] electric and magnetic fields respectively.

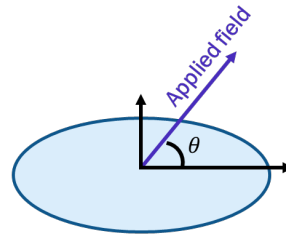
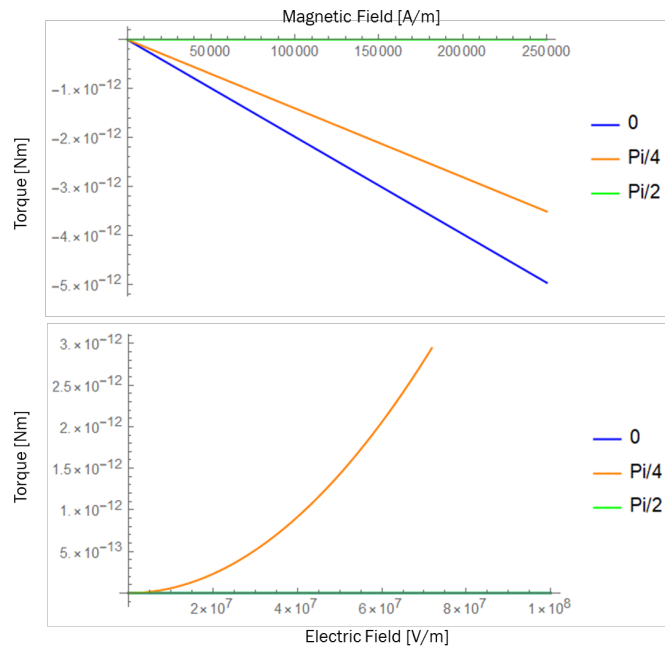


Figure 4-4: Theoretical electric and magnetic field generated torques vs applied field strength shown for 3 particle angles ( $0, \frac{\pi}{4}, \frac{\pi}{2}$ )

It should be noted that these torques are calculated for single particles systems and that nearby particles would have an effect on the actual field seen by the individual particle. It can be assumed however, that this serves as a good approximation for dilute systems and helps to guide experimental values toward a crossover point. What has been shown here is the importance of exploring relative direction and magnitude variations of applied electric and magnetic fields to tune particle distributions and thereby properties for materials design problems.

## **4.2 Nested Optimization Framework**

The design space for this optimization is define by the volume fraction and the applied electric and magnetic fields, for which the bounds are laid out in Table 6. Searching for the optimal properties in this bounded design space is accomplished via the nested optimization framework laid out in Section 2.4.1. In the full scale framework, we seek to optimize the constituent set to maximize the combined range of magnetic, dielectric, and elastic effective material properties. The results presented in this section, however, are for magnetization only. These results for magnetization are used to demonstrate the nested framework and the effect of the constituent set on the range of properties. The range of properties with respect to constituent set is then used to optimize the constituent set itself to maximize the range of possible properties. Later, the combined objective function with elastic and dielectric properties additionally accounted for will be presented in Section 4.3. The magnetic properties are examined alone, specifically without the elastic properties, due to the large computational expense of the elastic homogenization and the lesser expense of the magnetic homogenization. The optimization framework presented in this Section uses traditional gradient based methods to optimize the constituent set using the methods presented in Section 2.4.1 for the nested optimization framework.



The elastic homogenization requires a more efficient search algorithm. Optimization of the combined objective function, presented in Section 4.3, is accomplished by using a machine learning model, namely SVR, to augment the optimization framework for improved efficiency. The results in this chapter underline the improvements found with the hybrid machine learning and optimization model to be presented in Section 4.3.

The bounds for the optimization, equation (2-92), are tabulated in Table 6. It should be noted that the lower bound of volume fraction is set to 1% to avoid issues with the particle dynamics simulation which would be unable to create a volume at 0% volume fraction. The upper bound on volume fraction is set to 30% which exceed the current experimental range, and is a high volume fraction but does not cause issues with the particle dynamics simulations seen at higher volume fractions. A physical limit is approached in initializing the RVE with particles whose collective volume approaches that of the cubic space they occupy, however the simulated limit far exceeds experimental limits and is therefor not an issue for these models. The electric and magnetic field ranges were determined previously to attempt to reproduce experimental particle distributions [26].

**Table 6: Bounds on the optimization scheme independent variables (volume fraction and applied electric and magnetic fields).**

Input Variable	Lower Bound	Upper Bound
Volume Fraction	1%	30%
Electric Field	0 V/m	100 MV/m
Magnetic Field	0 A/m	250,000 A/m

Recall that the optimization seeks to determine the optimal constituent with the largest range of possible properties by varying processing conditions alone. The nested framework uses

the electric and magnetic fields, within in the ranges presented in Table 6, to tune the microarchitecture dependent effective material properties (in this Section, mass magnetization), finding the maximum and minimum properties for a given constituent set. The optimization framework then maximizes the range of properties, from this maximum and minimum set found, by varying the constituent set, in this case volume fraction. The nested optimization framework presented here used 4 independent variables in total, the volume fraction, the electric field, the magnetic field in parallel to the electric field, and the magnetic field perpendicular to the electric field to determine the constituent set with the largest range of possible properties.

The results for this optimization run are plotted in Figure 4-5 where several points of interest are noted. In the Figure, the three axes represent the strengths of the applied processing fields, electric and magnetic, in the simulations along the axes shown. Additionally, the color bar at right expresses the strength of the mass normalized magnetization (by particle mass), the variable be optimized here. Five points are noted on the graph in different regions with respect to the axes; the axes themselves represent the strengths of the applied processing fields  $H_y$ ,  $H_z$ , and  $E_z$  along their respective directions in the simulations. Consequently, the location of any point on the graph expresses its external field processing environment. Points 1 and 2 lie in regions of low processing fields overall. In contrast, point 3 keeps the electric and magnetic fields in the z-direction,  $E_z$  and  $H_z$  low, while the magnetic field along the y-axis,  $H_y$  is increased. From here, point 4 increases the applied electric field,  $E_z$ , to a region that has high perpendicular electric and magnetic fields,  $E_z$  and  $H_y$ . Finally, point 5 is in a highly explored region of the design space with high magnetic field only in the z-direction,  $H_z$ , and low electric field,  $E_z$ . Figure 4-6 is presented as a supplemental figure to Figure 4-5, showing the same three axes with the applied

electric and magnetic fields, but with the volume fraction represented in the color bar for visualization of the 4D design space to aid in interpretation of the optimization results.

Results presented in Figure 4-5 shows how the gradient based optimization explores the design space for the maximum and minimum magnetizations. Recall that the mass magnetization presented is particle mass normalized to remove the effect of differing magnetic content across samples, allowing extraction of only the effects of the structures on the magnetization. It is of no surprise that magnetically aligned particles would yield better mass magnetization as compared to electrically aligned particles, seen between point 1 and 5 in Figure 4-5. It can be seen however that the change in volume fraction shown between the cluster at 3 and the point at 4 from Figure 4-6 is significant, in the change in mass magnetization shown in Figure 4-5 since the mass magnetization decreases from point 3 to 4 despite the addition of the electric field perpendicular to the magnetic field. From the presentation of resulting torques in Section 4.1, the added electric field would aid in orienting the particles in a magnetically favorable orientation for the y-direction in conjunction with the magnetic field.

Figure 4-7 illustrates the results of the framework's search of the design space while varying particle volume fraction; and applied electric and magnetic processing fields. In the figure, the x-axis tracks the volume fraction of the particles modeled in the simulations while the y-axis tracks the resulting mass normalized magnetization as the sole output metric. Recall that the mass magnetization presented in the figure is particle mass normalized to remove the effect of differing magnetic content across samples, allowing extraction of only the effects of the structures on the magnetization. At a given volume fraction, the mass normalized magnetization of multiple discrete volume fractions, sampled within the design space by the framework as it progresses, are presented. The presentation of simulation data in this format results in a vertical

columns of points that show the maximum, the minimum, and thereby the range of intermediary mass magnetizations that can be achieved by varying processing conditions at that volume fraction. To simplify the chart, the external field processing conditions used in the simulation to achieve that point in the design space are not expressed in this figure (however they are shown in Figure 4-5 and Figure 4-6).

Figure 4-7 reports the mass magnetizations plotted versus volume fraction showing the nested nature of the search as well as the range of properties at each volume fraction selected by the outer loop of the optimization framework. In the figure, at a given volume fraction, it can be seen from this figure that the increase in volume fraction at 28% reduces the range of mass magnetization likely due to limited mobility of the particles in such a high volume fraction composite, inhibiting the favorable structures from being formed that contribute to increased mass magnetization in the lower volume fraction composites.

Figure 4-7 also shows that the optimum volume fraction for the largest range of possible properties by varying processing conditions alone is at 10% volume fraction. The minimum and maximum points found across all volume fractions are shown in Table 7. The points are presented in descending order, so that the first minimum and first maximum listed are the respective extrema at 10% volume fraction. It was decided to show 3 points to illustrate the multiple processing fields that can achieve the optimum properties found. Since the final point is selected based on range, the largest and smallest values are considered, but when relating these to input fields for selecting the corresponding processing parameters any of these could be considered to yield the desired properties. Selecting a single optimum point could require additional modification to the objective function to further reduce the processing fields. For example, minimizing the energy consumption would only reduce the valid results to the

processing fields with the lowest possible input energy to achieve the maximum or minimum property.

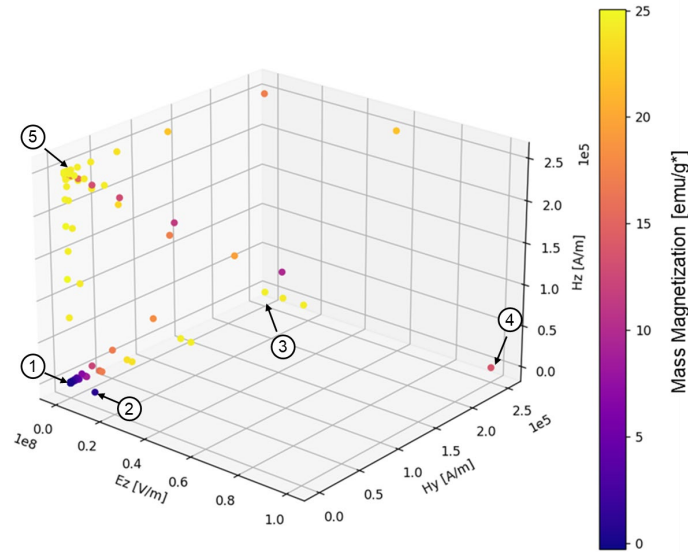


Figure 4-5: Results of gradient based optimization for mass magnetization showing max mass magnetization vs applied electric and magnetic fields for all volume fractions, with points of interest marked 1-2) no change in magnetization with increased electric field only 3-4) decreasing maximum magnetization due to volume fraction change (left to right the points here are for 11, 10, 7 and 28% volume fraction where the magnetization decreases with increased volume fraction) 5) highly sampled region to determine maximum which is shown to be at magnetic field only, orienting the particles' magnetizations with the applied field

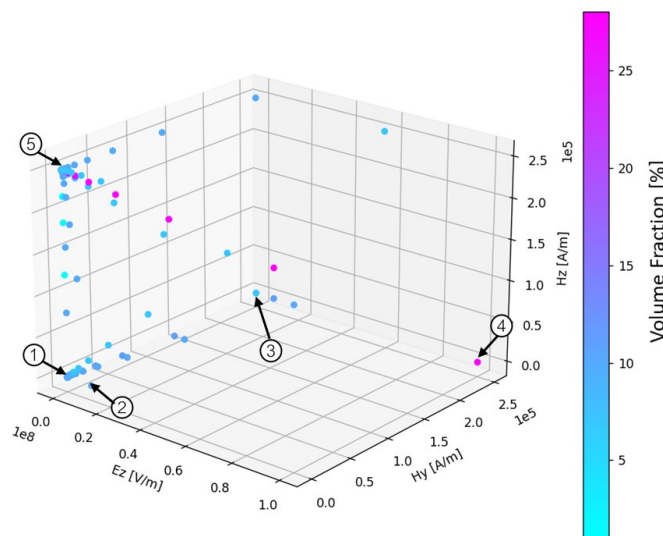


Figure 4-6: 4D design space representation of mass magnetization optimization results with applied field along the three axes and color representing volume fraction to demonstrate the exploration of the design space and supplement results in Figure 4-5, with the same points of interest marked for reference purposes

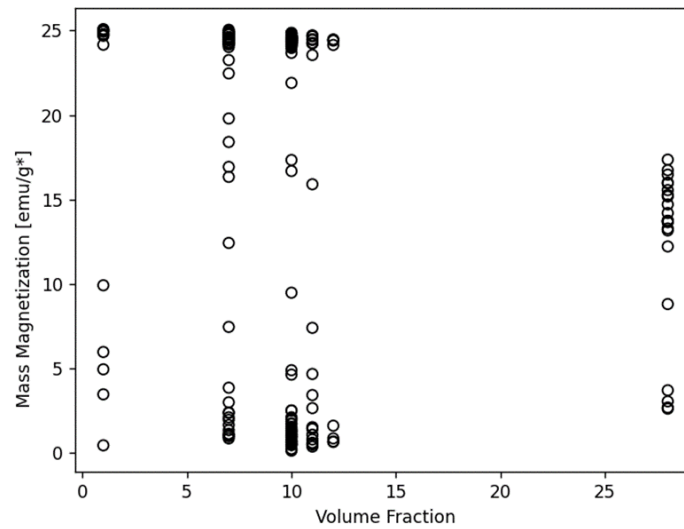


Figure 4-7: Results of gradient based optimization for mass magnetization showing all magnetizations found vs volume fraction to illustrate the range of possible properties as a function of the constituent set, noting the decrease in range at 28% volume fraction, likely due to lack of particle mobility in the composite to form favorable structures

**Table 7: Top 3 minimum and maximum points found at 10% volume fraction (3 points are shown to demonstrate the multiple processing conditions that yield the same optimum property) The first row for the minimums and maximums each are the respective optimum solution found**

	$E_z$	$H_y$	$H_z$	$M_x$	$M_y$	$M_z$
MIN	1000	0	0	-0.429	-0.490	-0.006
	0	0	250000	0.212	0.003	24.340
	0	1000	0	-0.649	-0.046	-0.003
MAX	108500	488	249512	0.110	0.013	24.628
	0	0	250000	0.088	-0.209	24.615
	724	177	249764	-0.076	0.186	24.586

The nested optimization framework presented in Section 2.4.1 was implemented here to find the constituent set (in this case volume fraction) that has the largest range of mass magnetization by varying property conditions alone. The optimization framework demonstrated its ability to search various processing fields and determine effective material properties, using

computational homogenization, to determine the maximum and minimum properties possible for a given volume fraction. Using the extrema found the inner loop (determining the maximum and minimum properties), a range is computed for possible properties for a given constituent set. The outer loop then searches all possible volume fractions for the largest range of possible properties. The optimization framework revealed that there exist multiple processing conditions which can return the maximum or minimum property for a given constituent set. In future work, the objective function can be modified to include additional considerations such as input energy from applying electric and magnetic fields to simultaneously minimize the energy consumption, reducing the number of possible field combinations for achieving a specific property to a single result.

### **4.3 Multi-Objective Optimization with Integrated Machine Learning**

Using the weighted combined objective function and the nested framework presented in Section 2.4 the design space search was able to vary processing fields to maximize and minimize the effective composite properties by varying processing fields, to determine the widest range of possible properties by optimizing the constituent set. The design space consists of 4 independent variables, volume fraction, electric field, and 2 magnetic fields (parallel and perpendicular to the electric field), that can be varied continuously on the bounds presented in Table 6. Each point (set of volume fraction and applied processing parameters) within this design space must be evaluated for the effective properties using the computational homogenization presented in Section 2.1.2, taking approximately 16hrs to determine the elastic, dielectric, and magnetic properties of each point. To minimize the computational expense of finding the maximum range of properties using the optimization framework presented in Section 2.4.2 a machine learning model is implemented to train simultaneously with the design space search for the optima. The

model is used to predict the extrema and efficiently search the full design space for test points (set of volume fraction and applied processing parameters) to evaluate using computational homogenization.

To begin the search of the design space for the optima the SVR machine learning model is initialized with 10 points that sample the design space efficiently. A Latin Hypercube initialization technique was used to sample the design space to train the support vector regression (SVR) machine learning model, presented in Section 2.5, used to find the optimum points in the design space [152]. Further investigation on the quality and efficiency of machine learning models is presented in Section 5.1 where the importance of the distribution of data on the quality of the model is observed. The results presented in this section use the model initialization points to train the SVR machine learning model and predict full design space processing field and volume fraction dependent responses to demonstrate the efficiency that can be achieved with the machine learning augmented optimization framework used.

This augmented model was used to repeat the magnetic optimization presented in section 4.2, the results of which are discussed further in Section 5.1 for determining the appropriate machine learning model. The augmented optimization scheme that was used was able to converge (the SVR machine learning model prediction is within 10% of the simulation results) on a solution in less than  $1/10^{\text{th}}$  the number of test points as the pure gradient based optimization scheme presented in Section 4.2. While the speed of convergence varies expectedly with the initialization points chosen, convergence is still achieved in roughly  $1/10^{\text{th}}$  the number of points when compared to not using a machine learning model with the optimization framework. To eliminate the random sampling a seed was assigned so that the optimization would be deterministic for future runs that may get interrupted and would need to be restarted.



For the optimization results presented in this section the points used are from a single run of the optimization framework to show the predictions of maximum and minimum properties used to compute the ranges of possible properties and the evaluation of the combined property objective function as the optimization framework would on its first run. Figure 4-8 shows the initialization points used in the predicted results (presented in Figure 4-9 and Figure 4-10 for the property ranges and combined property range objective function) and the fields for the maximum and minimum properties. Figure 4-8 shows the distribution of the training data used in the predicted results, both in applied field and volume fraction. The applied fields are, again, on the three axes with the volume fraction represented in the color bar. The machine learning optimization framework is able to predict minima and maxima without highly sampling the region of the design space near the extrema, which is critical in reducing the computational expense of finding the constituent set with the largest range of properties.

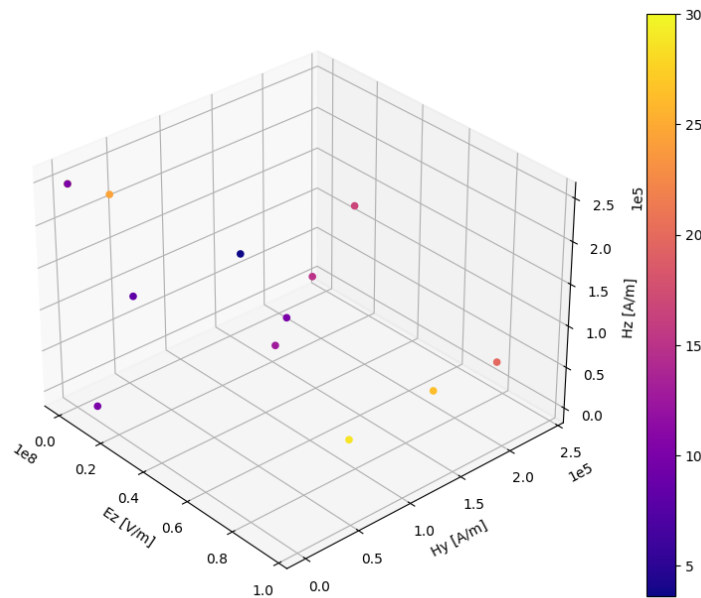


Figure 4-8: Design space plot with applied fields on the three axes and volume fraction represented in the color bar, showing initialization points and predicted points where effective property minima and maxima occur

The trained model is capable of predicting constituent and processing field dependent effective material properties based off of the simulation values in the training set from the initialization. The predictions are used to evaluate the objective functions of the optimization framework to find the maxima and minima that are selected to be verified by subsequent simulations. The SVR machine learning model can predict a full range of responses and is used to predict the ranges of possible properties, which are shown in Figure 4-9. Each figure has the volume fraction along the x axis and the property (elastic, dielectric, and magnetic) along the y axis. The range of volume fractions was discretized and evaluated at each point using the SVR machine learning model. At each volume fraction the gradient based optimization from the inner loop of the optimization framework presented in Section 2.4.1 is used to determine the maximum and minimum properties possible at that volume fraction. The maximum and minimum values found then are used to bound a bar segment for each volume fraction, which makes up the plots shown in Figure 4-9.

Figure 4-9 a shows the range of elastic modulus possible with varying volume fraction, where a steady range is shown with an overall shift to higher moduli with increased volume fraction is observed. The shift in range correlates to expected behavior with effective medium theory, where the case with the lowest possible modulus would necessarily still increase in modulus with increased volume fraction and can be tuned to form structures with a similarly larger maximum modulus. The prediction from the augmented optimization framework cannot be compared to the results from Section 4.2 for the elastic properties, because only magnetization was evaluated in the previous optimization. However, the predictions from the SVR machine learning model correlated to the expected behavior of the computational homogenization shown in Section 3.2.2 where the modulus was seen to increase in both the maximum (seen in the

results for Case 2 with electric field induced chaining) and the minimum (seen in the results for Case 1 with the no field and Case 3 with magnetic field only applied) with increased volume fraction.

For magnetization in Figure 4-9 b, the range is shown to increasingly decrease with volume fraction. When comparing this to the previous optimization results from Section 4.2, the maximum would be at 1% rather than 10% volume fraction as was found previously. However, it should be noted that the ranges in Section 4.2 are a function of the minimum values since the maximums found were near 25 emu/g\* and decreasing with an increase in volume fraction. This minimum dependent range calculation is important to note because the minimums are near 0 emu/g\* and for a random distribution will be a net 0 emu/g\*, possible at any volume fraction. The change in minimum then, shown in Section 4.2 is likely due to error in the computational homogenization itself, and as noted in Table 7 should be considered effectively 0 for near 0 values. Additionally the maximums shown in Figure 4-9 range from 22 emu/g\* to 15 emu/g\*, which is lower from the 25 emu/g\* and 17 emu/g\* found in Section 4.2. The lower values could be a function of both the error in the simulation results and the resulting fit. However, the trend of the maximum magnetization decreasing with increased volume fraction holds. It is therefore reasonable to conclude that the decreasing maximum seen in both the direct optimization, presented in the Section 4.2, and the augmented model, presented in this section, agree on the trend of the mass magnetization range with volume fraction, and that the maximum range for magnetization is at 1% volume fraction as shown in Figure 4-9.

Figure 4-9 c shows the increase in range of permittivity with increased volume fraction. Since the permittivity is highly dependent on the distance between particles, the increase in shown in Figure 4-9 c is intuitive with decreasing space available for particles to separate in at

higher volume fractions. The dielectric results from Figure 4-9 c, like the elastic properties from Figure 4-9 a, can also not be compared to the previous optimization results that only evaluated magnetic properties. The effective medium result presented in Section 3.2.3 for the dielectric response looked at the DC response, whereas the results in Figure 4-9 c are predictions of the AC response. The results presented in Section 3.3.2 in conjunction with the experimental data can be compared to the predictions presented here. The overall range of effective relative permittivity predicted by the SVR machine learning model is shifted higher than the simulation results presented in Section 3.3.2, where average values were between 6 and 8.5 for the relative permittivity. The large error seen in the results for in Section 3.2.2 could lead to a poor fit as a result of uncertainty in the training data. Improvements to the error in computational homogenization could improve this correlation.

The three properties explored in Figure 4-9 can be combined, in their normalized form, to show the behavior of the objective function with increased volume fraction. Figure 4-10 shows the summation of the normalized ranges using equation (2-93), where the largest range for all three properties combined is at 1% volume fraction. For design applications however, it will be important to also consider the absolute properties possible, not only their ranges if for example, a specific value is desired; the range must include that value and a simply large range may not suffice.

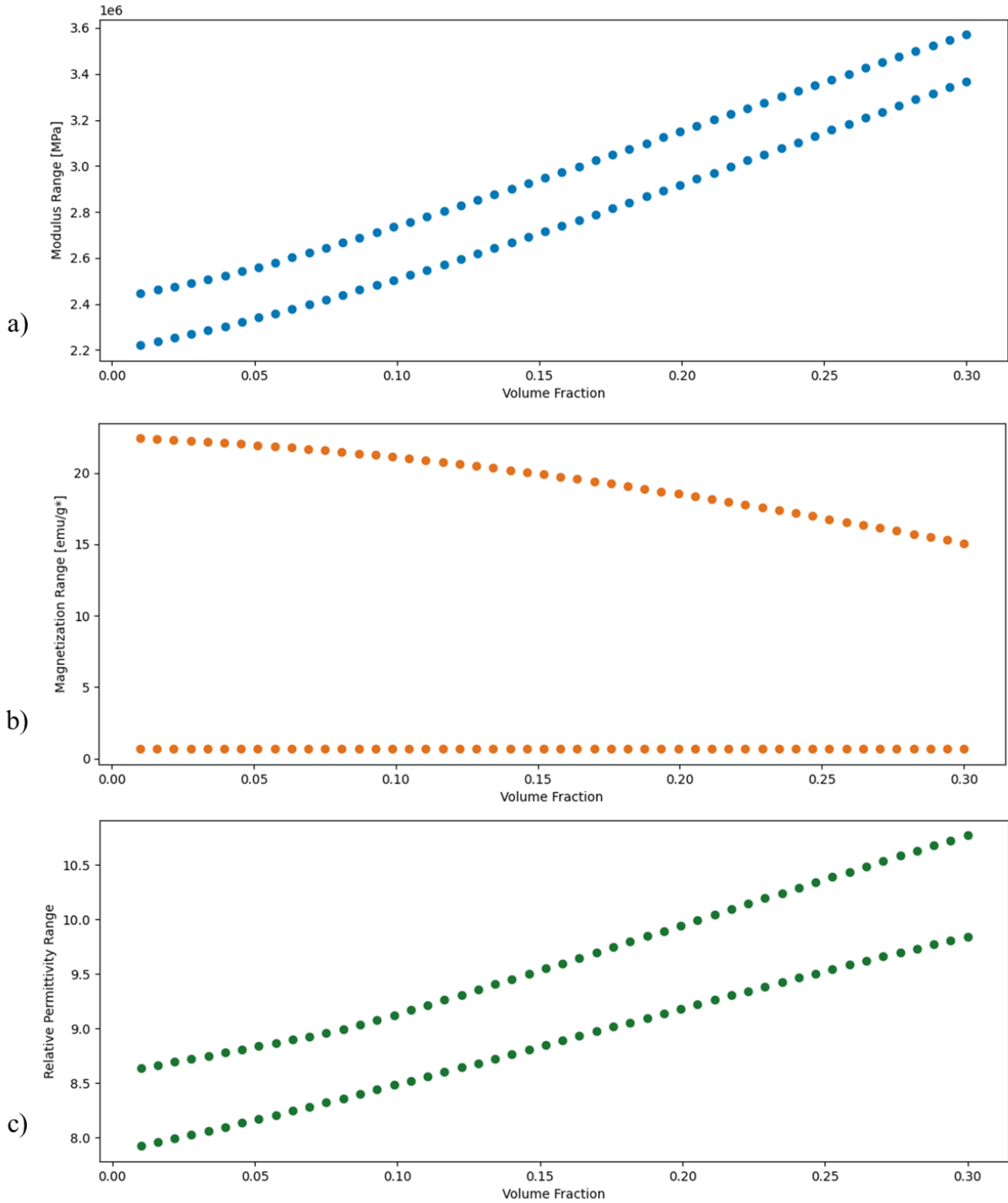


Figure 4-9: Maximums and minimum of a) modulus b) mass magnetization c) relative permittivity vs volume fraction as calculated by the trained SVR model showing the range of properties used by the objective function in the optimization framework

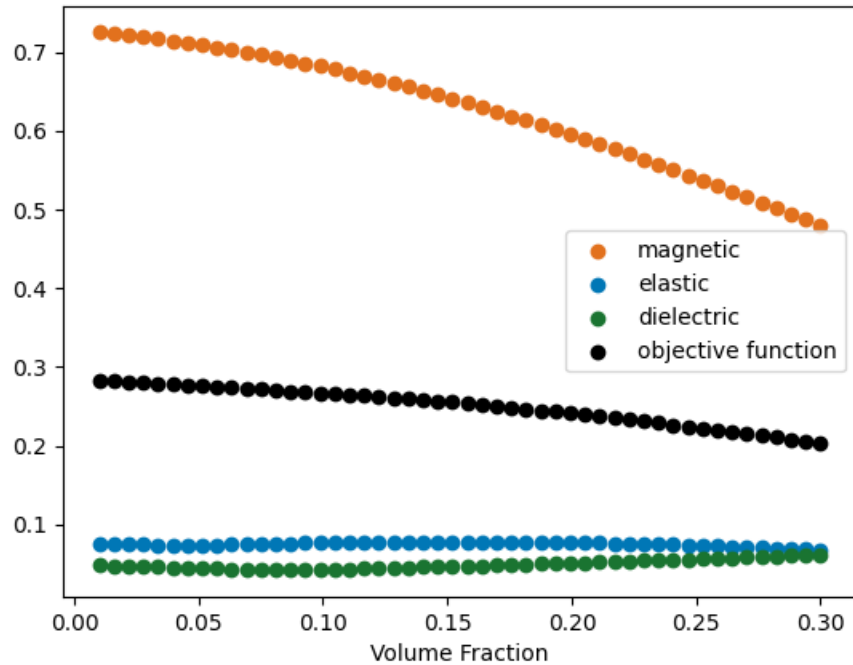


Figure 4-10: Predicted property range results for magnetic (orange), elastic (blue), dielectric (green), and the evaluation of the combined objective function (black) versus volume fraction

The results that have been presented in this section have been shown to agree well with simulations results and the previous magnetic property optimization run in Section 4.2 simply with the initialization points from the Latin Hypercube sampling. The machine learning augmented optimization framework can continue to learn and train while optimizing the constituent set for the maximum range of properties. The performance, and need to generate additional training data, of the SVR machine learning model's predictions to augment the optimization framework is evaluated by an error term. The error evaluates the fit of the SVR machine learning model's base (kernel) function to the tested simulation results within the design space. Similar to the error used to calculate an  $R^2$  value, the error, shown in equation (4-6), is defined as the distance of the prediction to the known value, using the already normalized results of the machine learning model. The simulation result is normalized by the same factor that the training data is normalized with for each property.

$$error = prediction - \frac{simulation\ result}{normalizing\ factor} \quad (4-6)$$

The normalizing factors used were determined by estimating the maximum expected property after several iterations of simulations were run to best scale all of the data, and future data points, between 0 and 1, as required by the machine learning model. The error should be modified for specific applications with consideration of the error in the homogenization framework itself. However, for this demonstration of the augmented framework 0.1 was chosen, when considering the range of normalized results are from 0 to 1, this corresponds to 10%.

Conventional use of data science models, such as the SVR used in this framework, have a given generated data set with input and output values. These values are separated into a training and a test set and used to train and evaluate the model chosen. In this work, a new way of training and evaluating the machine learning model had to be developed to run only the points needed to accurately model the design space to allow for training only as much as necessary, defined by the acceptable error. Using a growing data set, rather than a static one, lead to the development of the SmartSearch algorithm which is implemented when a point is tested with error outside of the allowed limit. The novel search algorithm presented in Section 2.5 is used to handle the design space sampling to minimize the computational expense in the exploration after the initialization of the design space.

Figure 4-11 shows the design space position of the test points chosen throughout the machine learning augment optimization framework with the applied electric and magnetic field on the three axes, and the volume fraction in the color bar. The error calculation in these runs attempted to normalize the already normalized data, resulting in erroneously large values, prompting the optimization framework to continue training. This has since been corrected,

however the behavior of the optimization framework when searching for new points to use for training both via the optimization of the properties and the SmartSearch algorithm discussed in Section 2.5, is demonstrated here. The design space points of this preliminary run are plotted in Figure 4-11 with lines between points showing the search progression through the design space using the SVR machine learning model to augment the optimization framework. The first 10 points from the Latin Hypercube initialization of the data set are shown but not connected with any lines since these are points generated as a set before the optimization begins. The points plotted in Figure 4-11 are the sets of volume fraction and processing conditions that selected by the machine learning augmented optimization framework and are either extrema of any of the three material properties of interest (elastic, dielectric, and magnetic), or points selected for training determined by the SmartSearch algorithm presented in Section 2.5. The points shown in Figure 4-11 are tested using the computational homogenization presented in Section 2.1 and are added to the training set that the SVR machine learning model is fit to. The SVR machine learning model is retrained each time a new data point is added to the training set.

Figure 4-11 shows the exploratory nature of the search, particularly as compared to the results presented in Figure 4-5 which did not utilize the machine learning argument optimization framework. The spread of the data shown in Figure 4-11 is a demonstration of the machine learning augmented optimization framework to only test design space points (set of volume fraction and processing fields) using computational homogenization to check the validity of a found optimum (maximum and minimum) effective property, or to train the model in an area of the design space that is largely unexplored and has large error, to eliminate unnecessary sampling of a small region of the design space as was previously observed in the non-augmented optimization framework in Figure 4-5 presented in Section 4.2.



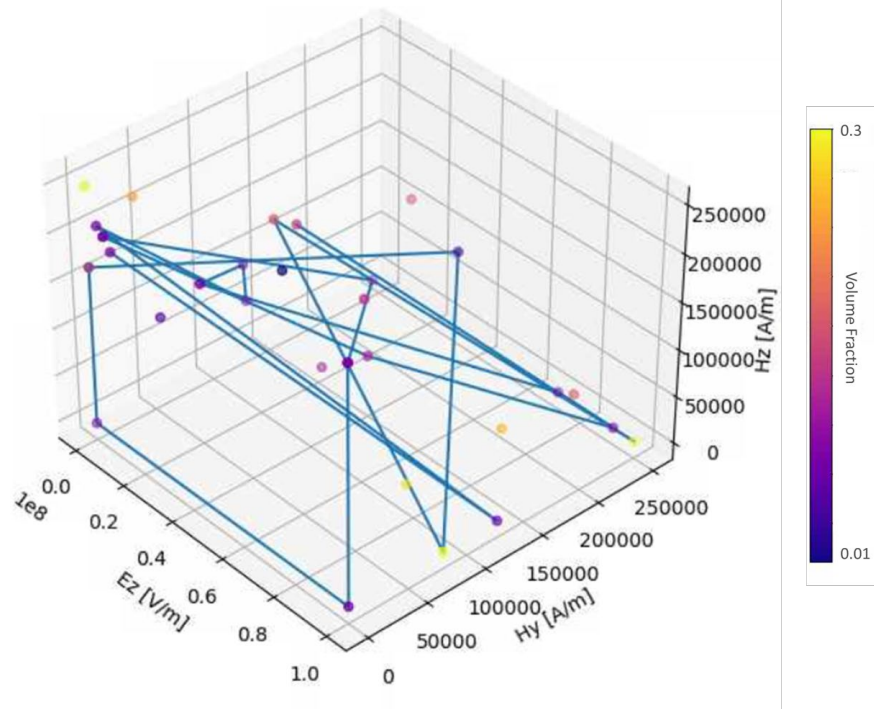


Figure 4-11: Design space search results from preliminary runs of the framework (that have since been modified), showing the applied fields on the major axes and the volume fraction in point color, to demonstrate the exploratory nature of the optimization framework when training. The lines indicate the order of the points tested, excluding the original 10 Latin Hypercube sample points to demonstrate the exploratory nature of the machine learning augmented optimization framework

The good agreement with previous simulation and optimization results of the SVR machine learning model predictions, using only 10 data points, demonstrates the improvement in searching the design space, from the previously presented simple gradient based search of magnetic properties in Section 4.2, which sampled 346 points before converging to a minimum. This is an improvement by an order of magnitude on points sampled, which is critical when considering the run time of the homogenization for elastic properties. Using this hybrid learning model with the novel SmartSearch algorithm presented in Section 2.5 allowed for broad search of this design space. Good agreement between the predicted trends and the previously simulated trends from Chapter 3 were shown. Additionally the predicted magnetization results showed

good agreement in maximum magnetization and change in possible range of properties with previous magnetic only optimization from Section 4.2. Future work will map the development of the results presented here, as additional points are added to the training set from the augmented optimization framework. The trained SVR machine learning model can also be used to display the pareto front of the properties and answer materials design questions.

#### **4.4 Pareto Front**

Due to the expected tradeoffs of microarchitecture driven properties (elastic, magnetic, and dielectric) a Pareto front can be generated between the three properties. Understanding this tradeoff is beneficial for specific design applications that may wish to find a balance between the properties. This section presents the pareto front for this design space using the trained SVR from Section 4.3, with all previously generated points, to demonstrate the behavior. For specific design applications, any points selected from analyzing the pareto front could again be tested using the homogenization simulation in a manner similar to the optimization algorithm, to verify the results for that application. The purpose of exploring the pareto front for this work, however, is to demonstrate the functionality, efficiency, and usefulness of the hybrid machine learning optimization framework presented.

Figure 4-12 shows the predicted properties for a full design space search using the constituent properties laid out in Table 4 and the design space bounds presented for the optimization scheme in Table 6. The design space was linearly discretized into 10 pieces on each of the 4 applied fields, with 4 selected volume fractions to represent the upper (30%) and lower (1%) limit as well as the remaining two volume fractions considered in Chapter 3, (5% and 10%). The interval of discretization can easily be modified to sample more points, however, the discretization used in Figure 4-12 provides a good visual representation of the design space with

1,000 points at each volume fraction. Figure 4-12 shows the predicted properties for each of these points in three subfigures with two properties examined in each along the two axes. The applied fields were discretized and evaluated at 4 volume fractions, 1% (yellow), 5% (purple), 10% (green), and 30% (magenta), generating a data set from which we can visualize Pareto curves.

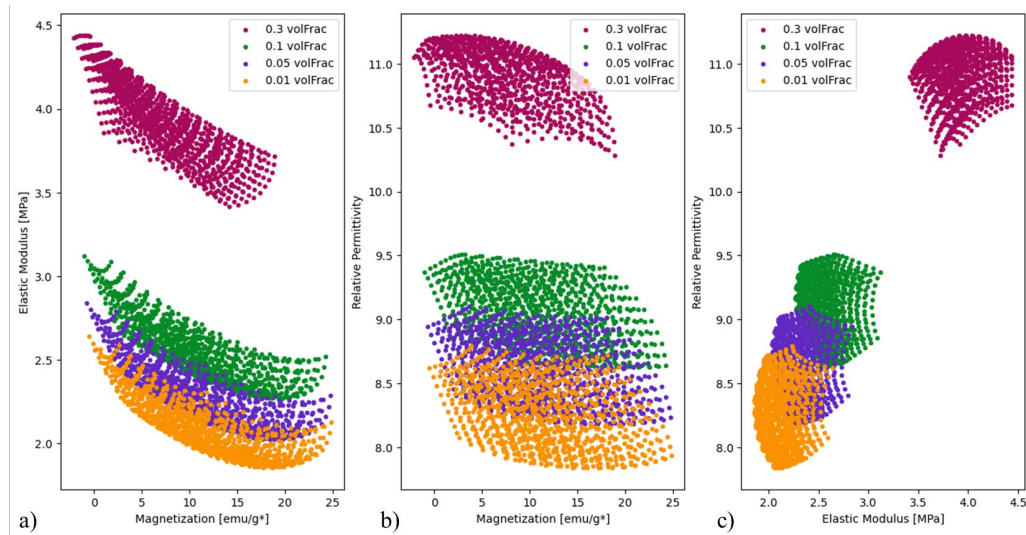


Figure 4-12: Design space representation from predicted SVR machine learning model results of a) elastic modulus vs magnetization, b) relative permittivity vs magnetization, c) relative permittivity vs elastic modulus showing the possible ranges of property combinations for 4 volume fractions: 1% (yellow), 5% (purple), 10% (green), and 30% (magenta), showing the tradeoffs between properties

There are several things to note in Figure 4-12 about the area of the design space (further analyzed in subsequent figures) and the change in properties with change in fields. The design space search was evenly discretized, in the manner discussed previously, for the plotted results. It therefore stands to reason that the areas with high point density have a small gradient of property with respect to field changes and that the less densely filled areas of this mapping have a higher gradient. Additionally, the design space areas shift in their placement within the property space of Figure 4-12, meaning that the change in volume fraction translates the possible

ranges of properties, as seen in the previous section as well. The sharp corners seen in each of the subfigures and volume fractions of the plots in Figure 4-12 are the design space limits, where the electric and magnetic field values are bounded and where the 5 main cases previously presented occur.

Trends in the individual property relationships are also visualized in Figure 4-12. The tradeoff in the elastic modulus and magnetization is shown by the decrease in elastic modulus for increased magnetization in Figure 4-12 a. The chains that increase the stiffness from electric field application do not magnetically orient particles, which can have a random orientation within the chain, producing a lower magnetization for the increased stiffness. The range of elastic moduli is shown to shift, in agreement with previous SVR machine learning model prediction results presented in Section 4.3. The range of magnetizations does not shift, but decreases in width, also seen in Section 4.2 and Section 4.3.

Figure 4-12 b relates the relative permittivity to the magnetization, where the shift in range of permittivity can be seen, increasing with increased volume fraction. Permittivity and magnetization require different microarchitecture for maximizing their respective values. However, both are affected by the particle spacing discussed in Section 3.4. The tradeoff between permittivity and magnetization is seen to swiftly drop off at higher magnetization values and spread to the full range for 30% volume fraction where points are more heavily distributed at higher permittivity within the area for that volume fraction. Increased volume fraction, and resulting reduced particle mobility, seems to favor higher permittivity likely due to particle proximity.

Figure 4-12 c shows the increase in both elastic modulus and relative permittivity with increased volume fraction. Both properties benefit from chaining in the same orientation and can be seen to have little tradeoff between the two.

The ranges of the SVR predicted values show higher maximums than previously presented for the simulation results in Section 3.3.2 which only looked at the 5 main cases. The values presented here explore the full design space, and as mentioned previously, where the corners shown in Figure 4-12 correspond some of these cases (with a few falling within the plotted areas). Predicting only the results for the 5 cases shows relative permittivity maxima around 9.5 for Case 4 in the z direction at 10% volume fraction and minima of 8 for the z direction of Case 3 at 1% volume fraction. The values shown for the simulations of the 5 cases in Figure 3-15 have large error bars and variability at this model size, ranging from 6 to 8.5 in relative permittivity values for the 3 volume fractions and 5 cases presented. This difference in absolute values shows a shift in values (the predicted SVR values being high in both maximum and minimum), potentially due to the error in the simulations used to train the machine learning model and/or from the volume fractions selected by the Latin Hypercube sampling, with approximately 3%, 6%, and 8% being the lowest 3 volume fractions tested each with 20-50% of the full electric field range application. Additional points at low volume fraction and design space bounds, such as zero applied fields, would possibly help to adjust this prediction to better match the simulated values. Although there is a discrepancy between the predictions of the SVR machine learning model, the model shows the effect of volume fraction and applied processing fields on the resulting properties, which is necessary for the optimization scheme.

The area of each design space is further illustrated in Figure 4-13, which outlines the point clusters for legibility. The area of these outlines is computed and tabulated in Figure 4-14,

where the change in possible combinations with respect to volume fraction is made clear. The trend in area shown in Figure 4-14 builds on the ranges of possible properties previously shown in Figure 4-9, showing the ranges of properties with respect to volume fraction and structure.

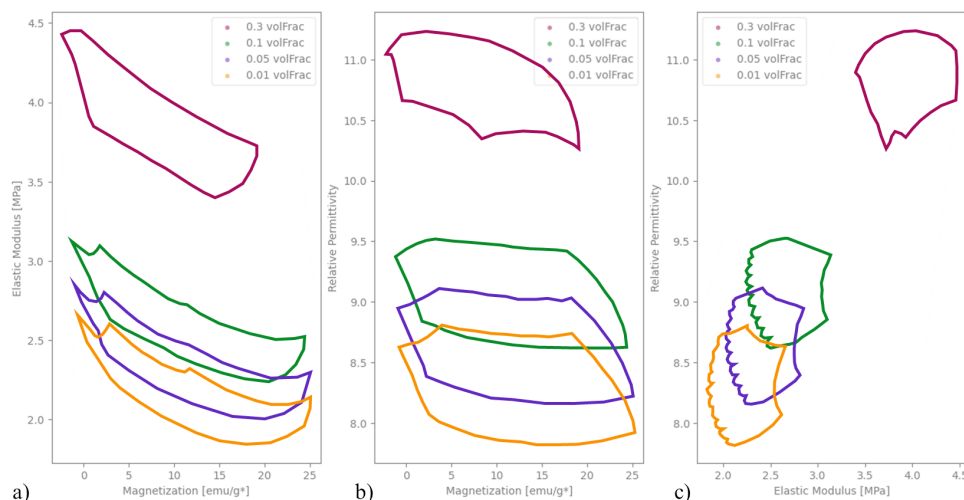


Figure 4-13: Design space area representation from predicted SVR machine learning model results of a) elastic modulus vs magnetization, b) relative permittivity vs magnetization, c) relative permittivity vs elastic modulus showing the possible ranges of property combinations for 4 volume fractions: 1% (yellow), 5% (purple), 10% (green), and 30% (magenta), showing the tradeoffs between properties

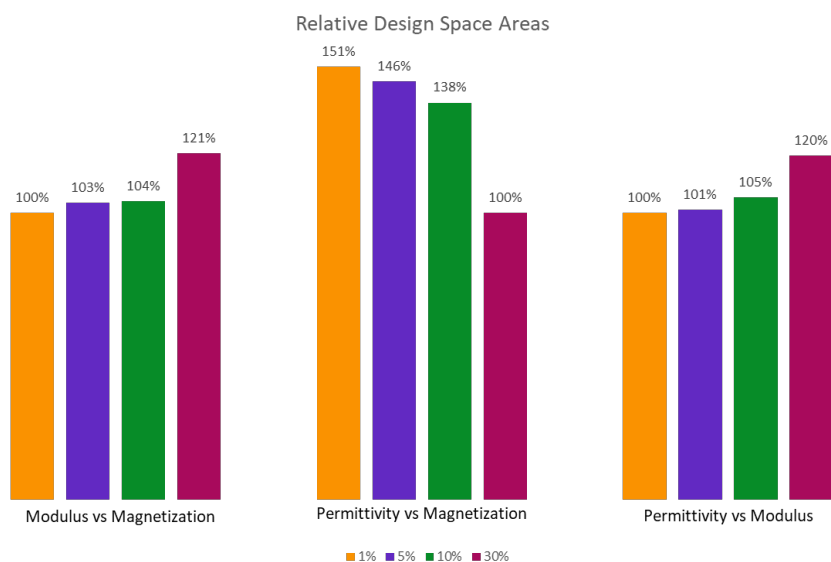


Figure 4-14: Calculated percent comparison of design space area from Figure 4-13 for a) elastic modulus vs magnetization, b) relative permittivity vs magnetization, c) relative permittivity vs elastic modulus normalized to the minimum area for each property combination showing the range of possible property combinations with change in volume fraction

Figure 4-15 shows the overlaid pareto front on the mapped design space, demonstrating the property tradeoffs for maximizing each property with respect to another. The large fronts in Figure 4-15 a and b as compared to the smaller front in Figure 4-15 c is interesting and intuitive. The permittivity and modulus both would improve with particle chaining where spacing is decreased in the direction that the property increases, and both would therefore then also increase with larger volume fractions. The permittivity and elastic modulus would not be opposing properties to be achieved with different structures and therefore applied fields inherently, leading to a smaller pareto front. The tradeoffs between these two properties, elastic modulus and relative permittivity, and the magnetization, however, are apparent in Figure 4-15 a and b. The structures required for maximizing magnetization orient the minor axis of the particles (the magnetic easy axis), directly counter to the orientation and structuring that leads to higher modulus and permittivity values. The design space mappings presented in this section are important when considering the design space needs for a specific design application and will be key components in a design tool.

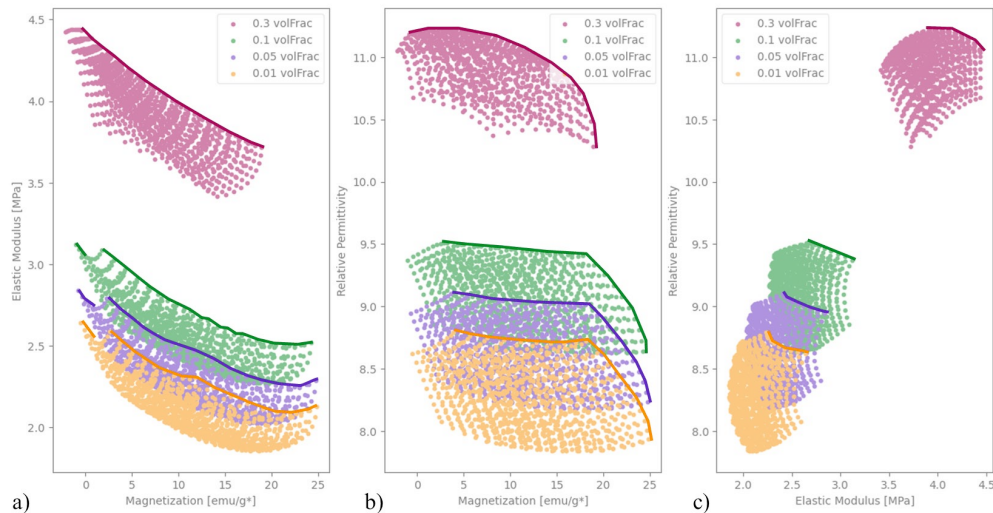


Figure 4-15: Pareto front from SVR machine learning model predictions of a) elastic modulus vs magnetization, b) relative permittivity vs magnetization, c) relative permittivity vs elastic modulus for 4 volume fractions: 1% (yellow), 5% (purple), 10% (green), and 30% (magenta), showing the tradeoffs between properties

## 4.5 Summary

This chapter presented the effect of the relative orientation and strength of the applied fields on particle distributions and resulting properties. To work toward the goal of a universal 3D printer the design space is explored to determine the maximum possible range of properties for a single constituent set. This search is conducted via a nested optimization framework that uses a combined objective function to consider elastic, magnetic, and dielectric properties simultaneously. The computational expense of the simulation and subsequent homogenization is considered and addressed by presenting a machine learning augmented optimization framework that uses the model to select optima and effective training points to test using the computational homogenization presented in Section 2.1.2. The machine learning model trains while being used to search, optimize, and map the design space. The efficiency improvement was demonstrated by the comparison of magnetic results from the single property optimization framework results (using 346 points) in Section 4.2 and the magnetic portion of the augmented optimization framework results, showing good agreement to the single property optimization framework with only 10 points. The trained model was also able to be used to explore the design space and map Pareto fronts to demonstrate the structure driven tradeoffs between properties as well as the decrease in available properties with increased volume fraction. It was also shown that the considerations for optimizations with specific materials design applications should include further parameters in the objective function such as energy consumption, to reduce the number of possible field combinations that will produce the desired properties, and discrete properties, to find design specific feasible working ranges. For the current objective of finding only the constituent set with the largest range of possible properties by varying processing conditions alone, it was found that lower volume fraction has a larger possible range of properties. Having



established the relationship between processing conditions and resulting properties, the next chapter will use this model to address the inverse problem of finding processing conditions for desired target properties.

## CHAPTER 5: Machine Learning and the Inverse Problem

### 5.1 Machine Learning Model Selection and Training

The methods and results discussed in the previous chapter form the basis on which the overarching goal can be achieved, finding the constituent set with the largest possible range of effective elastic, dielectric, and magnetic properties. The computational expense of the simulations required to determine the processing parameter dependent material properties necessitates efficient search, which was achieved in Chapter 4 by augmenting the optimization framework with a machine learning model. The mode chosen was a Support Vector Regression (SVR) model. The model was selected after testing with against the performance of a Feed Forward Neural Network (FFNN), the details of which are presented in this section. Both models were selected for their applications to regression problems. Both machine learning models, SVR and FFNN, are capable of regression and classification with different approaches and solutions. The testing in this chapter is to demonstrate the consideration in ultimately selecting SVR for use with the machine learning augmented framework and showcasing the benefits in the application of mapping the design space in this work.

The machine learning models considered in this work are the Feed Forward Neural Network (FFNN) and Support Vector Regression (SVR) for their applicability to similar problems. Both methods attempt to draw a boundary through a training set to classify data with specific labels or to define a regression equation by updating weights in a base (kernel) function. The manner in which weights are updated and the base (kernel) function is defined differ in the two approaches. Both machine learning models were setup with a test case to verify the functionality of the model to predict known behavior. Experimental and simulation data can be used to then train the machine learning models and map the design space entirely. The two

machine learning models have different training algorithms and are implemented differently in code. Ultimately, SVR is a more versatile model and is easily incorporated into the optimization scheme to augment the design space search and be used to solve the inverse problem.

A FFNN was trained on previously collected magnetization data on each of the 5 main cases to capitalize on the data that has already been collected and to test the machine learning model on the design space the optimization framework searches, the results of which are shown in Figure 5-1. Training the machine learning model on this data allows the model to predict the processing parameter dependent response of the composite. Training data is divided to reserve some data, the test data shown in Figure 5-1, for testing so that the models are not tested on data they have already seen, common in machine learning model approaches. The plot shows the applied magnetic fields in the y and z direction on the x and y axes with the z axis showing the resulting magnetization. Points are shown in black for the test data and the FFNN's prediction in red. The activation functions chosen, as a result of adjusting the hyperparameters manually after training to improve the fit of the function, exponential linear unit (elu), exponential, sigmoid, and rectified linear unit (relu), examples of which were shown in Figure 1-8 and are laid out by equations (5-1)-(5-4) [153], [154].

$$\begin{array}{ll} \text{elu} & y = e^x - 1 \text{ if } x < 0 \\ & y = x \text{ if } x \geq 0 \end{array} \quad (5-1)$$

$$\text{relu} \quad y = \max(0, x) \quad (5-2)$$

$$\text{sigmoid} \quad y = \frac{1}{1 + e^{-x}} \quad (5-3)$$

$$\text{exponential} \quad y = e^x \quad (5-4)$$

The trained model accurately predicts the test data, but due to the lack of sampling between the points that were tested, it cannot accurately predict the behavior of the system there. This is a function of the quality of data used to train the model and the kernel which the model attempts to fit to that data. Adequate spacing of sampling data, such as presented in Section 4.3, using a Latin Hypercube method to sample the design space, addresses this concern. Further testing using larger data sets to attempt to overcome this challenge.

The same hyperparameters were used again on data collected from an optimization run on magnetization, a larger data set, which has better sampling, to attempt to improve the fit of the machine learning model for the full design space. The results are illustrated in Figure 5-2 with a surface plot (for electric field of 0 V/m and 1% volume fraction) of the trained model, with the applied magnetic field in the y and z direction again on the x and y axes and the resulting magnetization on the z axis. The color bar in this plot shows the magnetization as well the better visualize the 3 dimensional space in the figure. It should be noted that the full design space includes the electric field and volume fraction as well, but that these have been collapsed for the plots of this model for simplicity which results in test data (grey points) not on the surface, which is a prediction without electric field. However, for the test points that do not fall on the surface the single predictions for that test data do match the expected results well (<10% error), seen by the red and gray points in Figure 5-2.

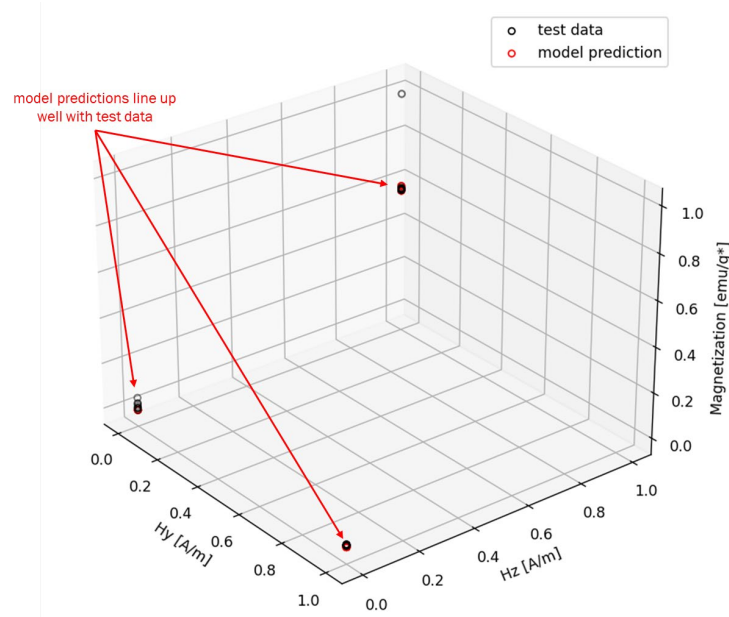


Figure 5-1: FFNN trained model vs test data for z direction magnetization as a function of magnetic fields (not shown are the applied electric field and volume fraction that are also independent variables in the design space and the FFNN)

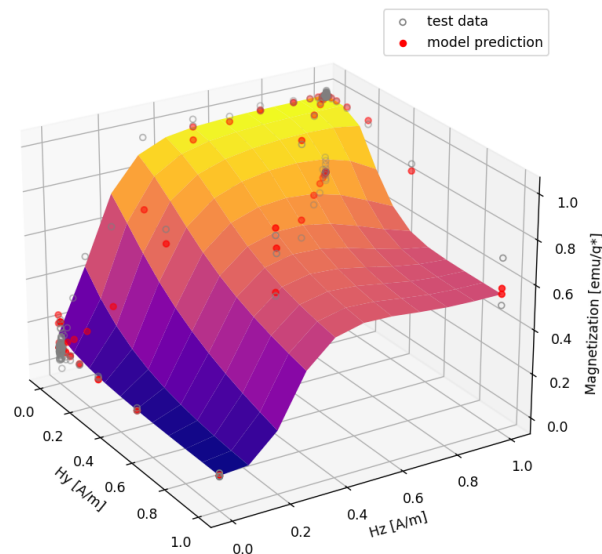


Figure 5-2: FFNN predictive surface (shown with colormap of Z axis value for visual clarity) overlaid on model test and predictive data points, showing the interpolated behavior defined by the kernel selection and availability of test points

One issue that was found with the FFNN is that it is stochastic in nature. The weights in the python TensorFlow package used initialize the weight of the network using random numbers which can lead to the model plateauing with a high level of error remaining. One method that

was used to combat this was shuffling the random seed if the error level was not sufficiently low to accept the model as properly trained and try again. However, this is costly to run, requiring several iterations of comparatively slow training (dozens of iterations can take 10-30 minutes to run) to obtain a usable model.

In contrast the SVR model runs very quickly (running for only a few seconds) and the hyperparameters relate to the fit of the model to the data and the amount of error directly. This could also be optimized in its own scheme but is not costly to run on even high fidelity models. Figure 5-3 shows the results of an SVR trained on the same optimization data used for the FFNN in Figure 5-2, with 20% of the data reserved for testing the model's predictive capabilities. This model is deterministic in nature and is not dependent on the starting state as the FFNN. The results are much more predictable and robust than the FFNN shown above due to the deterministic nature of the training algorithm built into the SVR machine learning model. Additionally, due to its lower computational expense it can also be easily incorporated and optimization scheme to augment the costly physics based simulations and finite element homogenization methods.

SVR is not as dependent on a high volume of training data. A model can therefore also be trained with experimental data. This is shown in Figure 5-4. It is important to note however that due to the small size of the data set, the entire set was used to train the model and is plotted for visualization. This model can be useful as serving as a training base for a hybrid optimization scheme in future work.

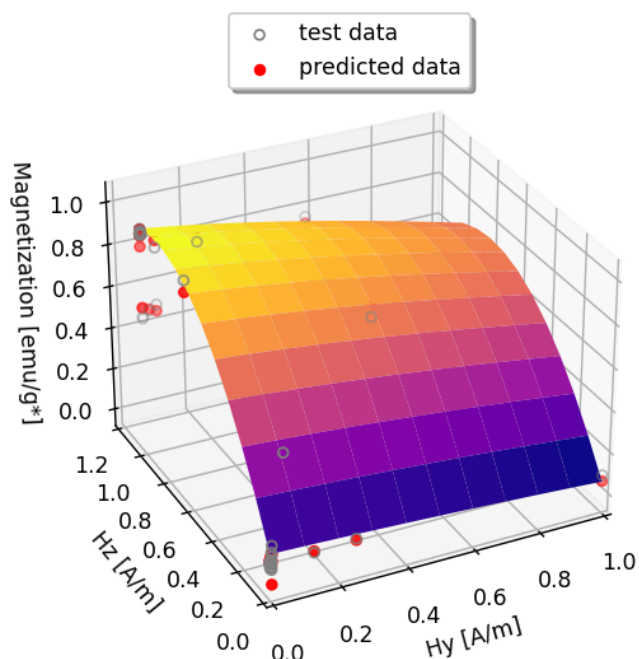


Figure 5-3: SVR Trained on 80% of magnetic data from an optimization run, 20% used for testing is shown in the grey and red points, the surface is for 0V/m electric field and 0.1 volume fraction (color scale for visualization of 3D space only)

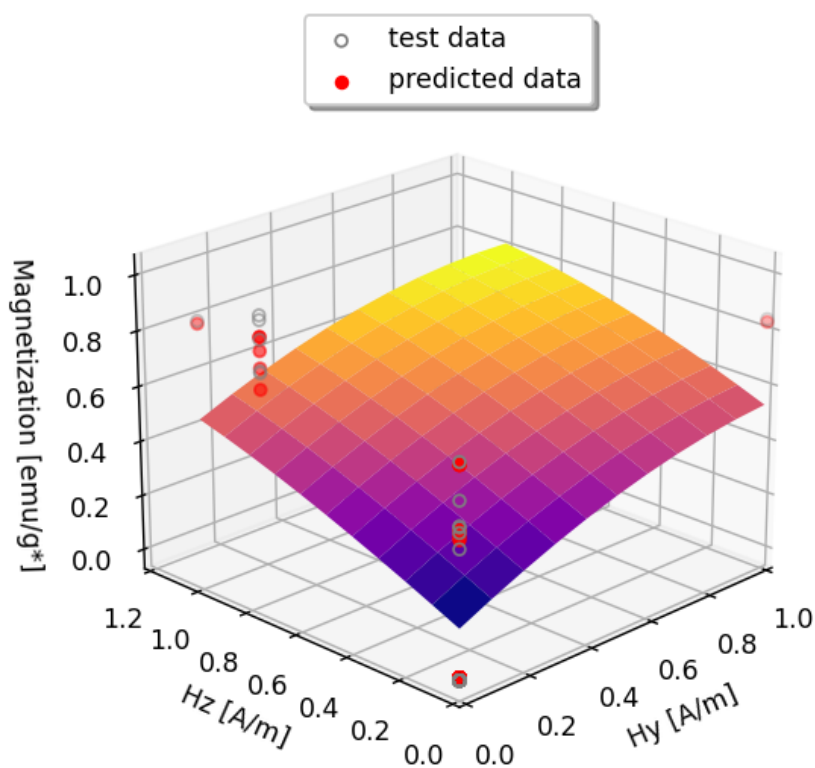


Figure 5-4: SVR Trained on an experimental data set, data points are shown in the grey and predictions in red points, the surface is for 0V/m electric field and 0.1 volume fraction (color scale for visualization of 3D space only)

SVR modeling has been shown to be a better option for the design space considered in this work due to the deterministic nature, low computational expense, and straight forward hyperparameters. The SVR training is integrated into the optimization framework presented in Section 2.4 and used to quickly find maxima and minima to test. The result of this hybrid method is that the number of points is drastically reduced to approximately  $1/10^{\text{th}}$  of the total number of points, which lowers the computational cost of optimizing the system while also building a model that can be used to address the inverse design problem.

## 5.2 The Inverse Problem

Mapping a forward flowing path from processing parameters to effective properties was done in Chapters 3 and 4. Understanding the ranges of properties and selecting a constituent set is useful for understanding the design space. Designing components to be manufactured requires the solution to the inverse problem, finding the processing conditions needed for a target property or combination of properties. Using the machine learning model trained for the optimization framework from Chapter 4, this inverse relationship can be defined. Two approaches are presented in this section. The first approach uses the radial basis function as an analytic solution and attempts to invert the expression to solve for the processing conditions for desired properties. This inversion method is complex for the radial basis (kernel) function used. Therefore, an alternative solution method is presented which uses hash tables to store processing fields for specific properties that is quick to populate and query. Both methods presented use the SVR machine learning model trained in Chapter 4 to explore the design space and the relationship between processing conditions and effective material properties.

The relationship between the independent and dependent variables of the SVR for a radial basis function is defined as



$$\mathbf{y} = \mathbf{C}K(\mathbf{x}_i, \mathbf{x}) + \rho \quad (5-5)$$

where  $\mathbf{y}$  is the output of the model,  $\mathbf{C}$  is the coefficient matrix determined from training the model,  $K$  is the radial basis function with feature vectors  $\mathbf{x}_i$  and intercept  $\rho$  that map the independent variables,  $\mathbf{x}$ , to the outputs. The radial basis function is defined as

$$K = e^{-\gamma|\mathbf{x}_i - \mathbf{x}|^2} \quad (5-6)$$

where  $\gamma$  is the hyperparameter from the model used for error bounds. This relationship shows the similarity between the support vectors and the independent variables to predict the corresponding output value. For the SVR machine learning model trained in Chapter 4 the independent variables are the volume fraction and the processing parameters (4 total); the outputs for the model are the resulting effective properties (9 total).

One approach to the inverse problem is reconstructing this equation from the trained model and solving the expression for the output  $\mathbf{y}$ . The resulting expression is quite complex, as a function of 4 independent variables. For the SVR machine learning model trained in this work, the radial basis (kernel) function has up to 16 coefficients for one output (one property in one direction). A total of 9 radial basis (kernel) functions with varying coefficients represent the full design space, one for each property (elastic, magnetic, dielectric) and direction (x, y, and z directions). This equation was fed into a symbolic math package which ran for hours before terminating with a recursion depth limit error. Inverting this expression is not practical or perhaps even feasible.

Alternatively, the design space can be mapped, as was done in Chapter 4 for displaying the Pareto front and the range of properties using the SVR machine learning model. The properties found in this mapping can then be hashed to store the corresponding fields in a quick

to access list of possible processing parameters for that target property. The process is laid out in a diagram in Figure 5-5 that shows the iterative process to map the full design space and the results being stored in hash tables for quick access data querying. Hash table searches are computationally inexpensive and allow for non-integer indexing of values, useful for storing material properties as keys. Once populated the hash tables can be searched to find all of the possible fields for a desired property and filtered for combinations of properties. The returned processing conditions would report multiple fields and volume fractions if the combination desired can be achieved in multiple ways.

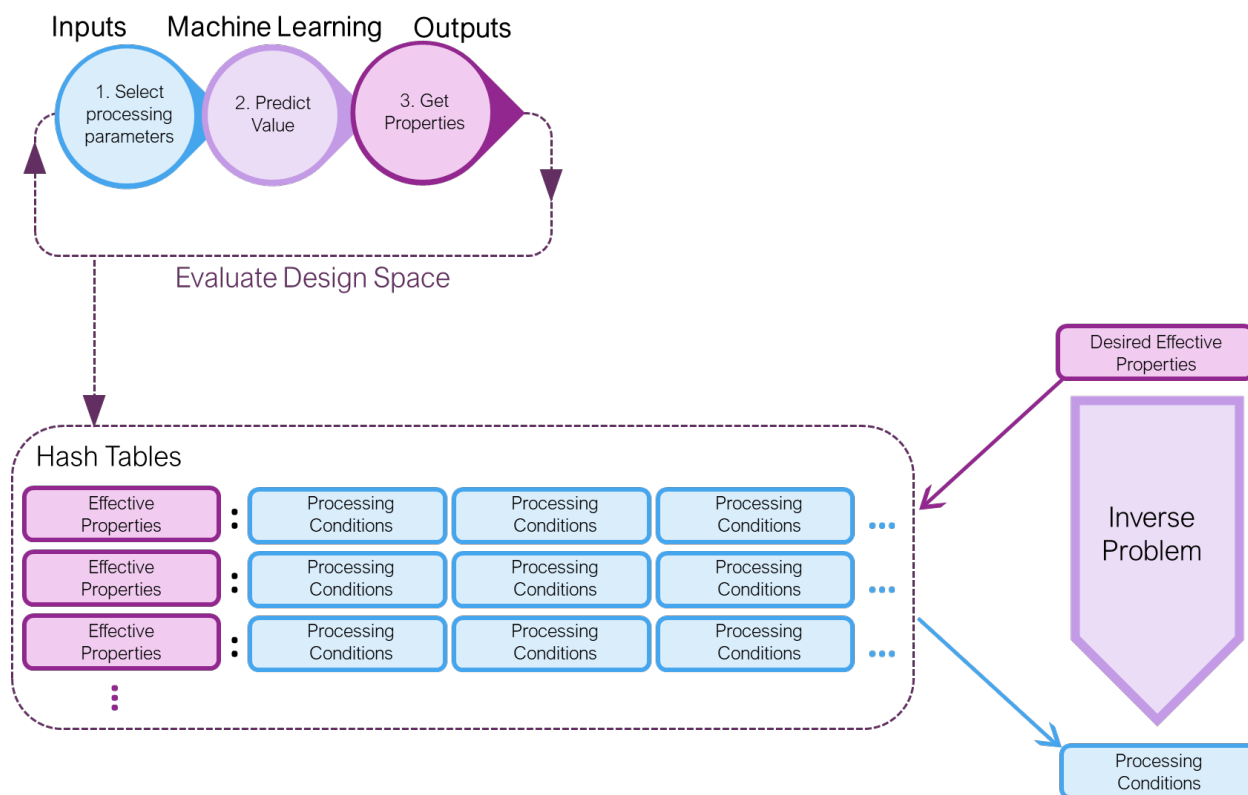


Figure 5-5: Diagram showing the workflow of the HashLookUp algorithm, mapping the design space, storing properties as hash keys with corresponding processing condition values for quick access lookup

An example of the method to find processing conditions and the volume fraction for desired properties is presented in Figure 5-6. For this test case (a problem with a known solution), a set of processing fields and a volume fraction was chosen, and the properties predicted to generate a set of target properties, with known processing conditions, for testing the inverse problem solution method using hash tables (HashLookUp). The targeted properties are 3.7 emu/g\*, 2.77 MPa, and relative permittivity 8.3. Figure 5-6 shows that the method was able to find the (previously known test case solution) set of processing fields that would achieve the targeted properties.

```
Target Properties:
3.7 2.77 8.3
Solution found? -> True
Possible Processing Fields: [[0.01, 1.0, 0.0, 0.0]]
Properties Found:
3.7 2.77 8.3
```

Figure 5-6: HashLookUp algorithm search results for solving the inverse problem

This method was repeated on one of the training points to compare to the known solution. The known test point chosen was 29% volume fraction, 75611216 V/m electric field in the z direction, 103535 A/m magnetic field in the y direction, and 7565 A/m magnetic field in the z direction with found properties (from a single datapoint) 0.28 emu/g\*, 4.2 MPa, and 12.4 relative permittivity. The target properties are taken from the support vector regression prediction of this region of the design space, as would be done in a materials design application, using the pareto fronts, shown in Figure 5-7. The solutions found demonstrate the ability to find multiple fields with the same predicted set of target properties, in addition to the fields found being in the range of the initial known test point processing parameters. The target properties found solutions at the targeted volume fraction with similarly high electric fields in the z

direction, high magnetic fields in the y direction, and low magnetic fields in the z direction as targeted. The discretization of the design space can be increased and the model can be trained with more data in regions of interest to better fit the model to the simulation data bringing the processing fields closer to the simulation averages.

```
Target Properties:
0.8 [emu/g*], 4.2 [MPa], 10.9
Solution found? -> False
Possible Volume Fractions and Processing Fields:
29.0 %, 90000000.0 [V/m], 100000.0 [A/m], 25000.0 [A/m]
29.0 %, 95000000.0 [V/m], 112500.0 [A/m], 37500.0 [A/m]
29.0 %, 100000000.0 [V/m], 137500.0 [A/m], 50000.0 [A/m]
```

Figure 5-7: HashLookUp algorithm search results for solving the inverse problem

The HashLookUp workflow presented in this section addresses the inverse design problem, relating the target properties to the processing parameters required to achieve them. It allows for a combination of parameters to be queried and returns all possible fields for the provided target properties. The HashLookUp method overcomes the issue of inverting a complicated expression in an efficient manner. In future work the algorithm can be incorporated into a user friendly tool along with the design space plots presented in Chapter 4, for use in design applications.

### 5.3 Summary

This section presented the inverse problem and solution methods using a trained machine learning model. The analytic expression (kernel function) relating independent variables to dependent variables is complex and difficult to invert, if possible. As a more efficient and practical approach an iterative search and data structure solution method, utilizing hash tables, was presented. The hash table method allows for multiple solutions to be determined, showing all possible field combinations. The quick solution time using this iterative method makes it

more practical and effective than inverting the kernel function. The solution to the inverse problem presented in this section and solving it efficiently will be important for future work in materials design applications and decisions utilizing multifield processed composites.

## CHAPTER 6: Conclusion

### 6.1 Conclusions

The framework presented in this work has created a previously unestablished relationship for particulate composites processed using coupled electromagnetic fields between processing parameters of composite materials to their effective properties. This framework is the foundation for design and manufacture with a universal 3D printer, to tune the material properties locally without changing the stock material. The framework also demonstrated the ability to optimize the constituent for the largest range of possible properties. A machine learning augmented optimization framework was presented that showed good agreement with previous optimization runs using less than  $1/10^{\text{th}}$  of the number of test points. Furthermore, an inverse path is required to determine the processing parameters needed to achieve a target property or set of properties. This was addressed using the machine learning model incorporated into the optimization scheme. The solution to the inverse problem demonstrated the capability to determine the set of volume fractions and applied processing fields for any property or set of properties. The work presented here sets the basis for future development of additive manufacturing design tools for multi-functional printing.

The relationship between processing parameters and effective material properties was determined using physics based particle dynamics simulations in conjunction with computational homogenization. The effective elastic, magnetic, and dielectric properties of barium hexaferrite (BHF) and polydimethylsiloxane (PDMS) composites processed in different applied electric and magnetic fields were found and compared to classic effective medium theory and experimental values to verify and validate the process. The link to effective properties for field processed composites with varying microarchitecture was previously undefined as a function of

constituents and processing parameters. The newly established homogenization scheme was then used to explore the design space.

An optimization framework was constructed utilizing gradient based techniques to efficiently explore the design space for the processing fields that yielded the maximum and minimum properties. These extrema were then used to find the range of properties possible for a specific constituent set. The framework demonstrated the ability to optimize the constituents for the largest range of combine elastic, dielectric, and magnetic properties. To reduce the computational expense of the combine optimization a machine learning model was used to augment the process, allowing for faster search for the maximum and minimum properties. The results of this model were compared to previous simulation results and experimental values. The trained model also serves as the basis for mapping the full design space, finding the pareto fronts for the three properties examined, and as a starting point for solving the inverse problem.

Important tradeoffs in relative permittivity, elastic modulus, and mass magnetization were shown. It was demonstrated that the microarchitecture formed by the applied electric and magnetic fields determine the effective properties of the composites and critically that the structures that improve one property, such as elastic modulus, may not also improve other properties, such as mass magnetization. The scripts and processing used to present these results are important components for designers when considering properties for specific design applications. It was also shown that the range of properties changes as a function of volume fraction and that for some properties this range may increase while decreasing for others. For example, in the samples run in this work it was seen that the range of magnetization decreases with increased volume fraction while the range of possible relative permittivity values increases. The range of properties, their discrete values, and the tradeoffs between properties are all

necessary considerations for component design and have been demonstrated using the framework presented in this work.

Methods for managing the computational expense of the optimization and integrated computational homogenization have been presented. A new dynamic model for intelligently exploring the design space and exploiting the results for optimization has been presented to most effectively use the cost required to map the design space. The model resulting from this search was also shown to be effective for addressing the inverse problem. An additional function was presented that maps the design space similar to the methods used to find the pareto front, with values stored in hash tables for quick and efficient recall of associated processing fields for target properties. A filtering technique was then used to present the constituent set and processing fields required to achieve a set of desired properties.

This work explored the relationship between processing conditions and effective properties in both the forward and inverse directions as a basis for additive manufacture and materials design. The framework presented here is able to demonstrate the relationship between properties and associated tradeoffs, properties ranges, and optimal points within the defined design space. Efficient algorithms are presented using AI and machine learning methods to augment computationally expensive processes required to explore the continuous design space.

## **6.2 Contributions**

Contributions to the body of knowledge and field of mechanical engineering and material science by this work are:



- Construction of a framework that connects processing parameters to effective composite properties by integrating particle dynamics simulations and computational homogenization into a single workflow
- Construction of a framework to collect, store, and recall datapoints used in optimization and data science schemes to augment the optimization scheme with a growing data set trained machine learning model
- Development of a nested multi-objective optimization framework to search for the constituent set with the largest range of possible properties
- Development of a machine learning augmented multi-objective optimization scheme to reduce computational expense to a feasible runtime (10% of the pure gradient based method), that coupled gradient based optimization with support vector regressions
- Development of an artificial intelligence search algorithm (all paths search a modified depth first search algorithm) to find conductive pathways in a composite RVE to determine percolation and create physics based data structures used to calculate effective conductivity of composites with highly conductive fillers
- Improved fidelity to particle dynamics simulations to account for non-spherical particle contacts by calculating ellipsoidal distances between particles

### **6.3 Future Work**

The first working 3D printer is credited to Charles W. Hull in 1984, commercialized by 3D systems shortly thereafter. The early days of additive manufacturing to follow were defined by academic research, commercial patents, and general inaccessibility to the public. The future of this technology would drastically change starting in 2005 when patents expired. Adrian Bowyer and his team would release open-source printers in 2007 and again in 2009 as part of the RepRap

Project. Community contributions with accessible tools such as Arduino and the code repository GitHub fueled the development that led to the innovation in 3D printing technology available today. [155]

The state of additive manufacturing today is a result of the collaboration and community contribution to the development of 3D printers. Open source printers matured at an accelerated rate due to the availability of the information [155]. This access to information drives so many of the valuable open source tools like Wikipedia [156] and VLC [157]. Open-source hardware, such as the Raspberry Pi, have been used as educational tools to increase proficiency with computers and programming [158]. The community around such coding projects provides resources and lessons for people to become familiar with technology, fueling innovation and entrepreneurship. While efforts to make free and accessible tools for community use is foundational to furthering engineering development as a whole [159], it is also critical to make research as accessible as possible within the academic community and the general public it serves. This is echoed in efforts by the US government to make tax funded research accessible to taxpayers, with accessibility being only one aspect of the initiative. The push for free access extends to the underlying data behind publications as well as the articles themselves [160]. It can be argued that associated code falls within this scheme as well.

Within the realm of open-source software, there are several packages that perform many of the tasks typically used within the engineering community. This includes CAD modeling [161], [162], finite element analysis [163], [164], and numerical computing [165]–[167]. These tools can be incorporated into the framework presented in this work to be made available to the engineering community for further development and implementation.

Future plans for the framework presented here also include reducing computational cost and improving efficiency. Improvements include changing the particle dynamics simulations to be object oriented for particle shape and size distributions, using less memory- intensive and more efficient computations per iteration, and utilizing computer hardware optimally such as offloading matrix operations to GPU processing. These changes were not feasible on the server used for this work.

### Planned Publications

A Computational Homogenization Framework for Predicting Properties from Multifield Processing Conditions in Polymer Matrix Particle Composites (May 2023, *Smart Materials and Structures*)

Multi-Objective Optimization of Predicted Properties from Multifield Processing Conditions in Polymer Matrix Particle Composites (*Journal of Intelligent Material Systems and Structures*, May 2023)

Predicting Conductivity of Particle Filled Polymer Matrix Composites Using Artificial Intelligence Search Techniques (*Journal of Intelligent Material Systems and Structures*, August 2023)

A Machine Learning Approach for Predicting Processing Parameters for Desired Polymer Particle Composite Properties (*Journal of Intelligent Material Systems and Structures*, August 2023)

## REFERENCES

- [1] ASTM Standard 25900, “Standard Terminology for Additive Manufacturing – General Principles – Terminology.” ASTM International, 2015.
- [2] M. Rafiee, R. D. Farahani, and D. Therriault, “Multi-Material 3D and 4D Printing: A Survey,” *Adv. Sci.*, vol. 7, no. 12, p. 1902307, Jun. 2020, doi: 10.1002/advs.201902307.
- [3] K. Gandha *et al.*, “Additive manufacturing of anisotropic hybrid NdFeB-SmFeN nylon composite bonded magnets,” *J. Magn. Magn. Mater.*, vol. 467, pp. 8–13, Dec. 2018, doi: 10.1016/j.jmmm.2018.07.021.
- [4] Y. Sano, R. Matsuzaki, M. Ueda, A. Todoroki, and Y. Hirano, “3D printing of discontinuous and continuous fibre composites using stereolithography,” *Addit. Manuf.*, vol. 24, pp. 521–527, Dec. 2018, doi: 10.1016/j.addma.2018.10.033.
- [5] J. J. Restrepo and H. A. Colorado, “Additive manufacturing of composites made of epoxy resin with magnetite particles fabricated with the direct ink writing technique,” *J. Compos. Mater.*, vol. 54, no. 5, pp. 647–657, Mar. 2020, doi: 10.1177/0021998319865019.
- [6] Y. Wu, D. Isakov, and P. Grant, “Fabrication of Composite Filaments with High Dielectric Permittivity for Fused Deposition 3D Printing,” *Materials*, vol. 10, no. 10, p. 1218, Oct. 2017, doi: 10.3390/ma10101218.
- [7] S. Park, W. Shou, L. Makatura, W. Matusik, and K. (Kelvin) Fu, “3D printing of polymer composites: Materials, processes, and applications,” *Matter*, vol. 5, no. 1, pp. 43–76, Jan. 2022, doi: 10.1016/j.matt.2021.10.018.
- [8] “Ultimaker S5,” *ultimaker.com*. [Online]. Available: <https://ultimaker.com/3d-printers/ultimaker-s5>
- [9] “Stratasys F123 Composite-Ready Printer,” *stratasys.com*. [Online]. Available: <https://www.stratasys.com/en/3d-printers/prINTER-catalog/fdm-printers/F123-composite-ready/>
- [10] “Flashforge Creator Pro,” *flashforge.com*. [Online]. Available: <https://www.flashforge.com/product-detail/flashforge-creator-pro-3d-printer>
- [11] M. A. H. Khondoker, A. Asad, and D. Sameoto, “Printing with mechanically interlocked extrudates using a custom bi-extruder for fused deposition modelling,” *Rapid Prototyp. J.*, vol. 24, no. 6, pp. 921–934, 2018.
- [12] “PolyJet,” *stratasys.com*. [Online]. Available: <https://www.stratasys.com/en/3d-printers/prINTER-catalog/polyjet/?filter=PolyJet>
- [13] A. Elkaseer, K. J. Chen, J. C. Janhsen, O. Refle, V. Hagenmeyer, and S. G. Scholz, “Material jetting for advanced applications: A state-of-the-art review, gaps and future directions,” *Addit. Manuf.*, vol. 60, p. 103270, Dec. 2022, doi: 10.1016/j.addma.2022.103270.
- [14] T. Stichel, T. Brandl, T. Hauser, B. Geißler, and S. Roth, “Electrophotographic multi-material powder deposition for additive manufacturing,” *Procedia CIRP*, vol. 74, pp. 249–253, 2018, doi: 10.1016/j.procir.2018.08.104.
- [15] T. Laumer, T. Stichel, P. Amend, and M. Schmidt, “Simultaneous laser beam melting of multimaterial polymer parts,” *J. Laser Appl.*, vol. 27, no. S2, p. S29204, Feb. 2015, doi: 10.2351/1.4906303.
- [16] F. Brueckner *et al.*, “Enhanced manufacturing possibilities using multi-materials in laser metal deposition,” *J. Laser Appl.*, vol. 30, no. 3, p. 032308, Aug. 2018, doi: 10.2351/1.5040639.

- [17] U. Shaukat, E. Rossegger, and S. Schlögl, “A Review of Multi-Material 3D Printing of Functional Materials via Vat Photopolymerization,” *Polymers*, vol. 14, no. 12, p. 2449, Jun. 2022, doi: 10.3390/polym14122449.
- [18] D. Kokkinis, M. Schaffner, and A. R. Studart, “Multimaterial magnetically assisted 3D printing of composite materials,” *Nat. Commun.*, vol. 6, 2015, doi: 10.1038/ncomms9643.
- [19] B. Bharti, F. Kogler, C. K. Hall, S. H. L. Klapp, and O. D. Velev, “Multidirectional colloidal assembly in concurrent electric and magnetic fields,” *Soft Matter*, vol. 12, no. 37, pp. 7747–7758, 2016, doi: 10.1039/c6sm01475e.
- [20] M. A. Masud, “On the Electric and Magnetic Field Assembly of Particulate Polymer Composites,” PhD Dissertation, Department of Mechanical and Nuclear Engineering The Pennsylvania State University, State College, Pennsylvania, 2019.
- [21] Z. Varga, G. Filipcsei, A. Szilágyi, and M. Zrínyi, “Electric and Magnetic Field-Structured Smart Composites,” *Macromol. Symp.*, vol. 227, no. 1, pp. 123–134, Jul. 2005, doi: 10.1002/masy.200550912.
- [22] O. D. Velev, S. Gangwal, and D. N. Petsev, “Particle-localized AC and DC manipulation and electrokinetics,” *Annu. Rep. Prog. Chem. - Sect. C*, vol. 105, pp. 213–246, 2009, doi: 10.1039/b803015b.
- [23] G. K. Hu and G. J. Weng, “The connections between the double-inclusion model and the Ponte Castaneda–Willis, Mori–Tanaka, and Kuster–Toksoz models,” *Mech. Mater.*, vol. 32, no. 8, pp. 495–503, Aug. 2000, doi: 10.1016/S0167-6636(00)00015-6.
- [24] T. Kanit, S. Forest, I. Galliet, V. Mounoury, and D. Jeulin, “Determination of the size of the representative volume element for random composites: statistical and numerical approach,” *Int. J. Solids Struct.*, vol. 40, no. 13–14, pp. 3647–3679, Jun. 2003, doi: 10.1016/S0020-7683(03)00143-4.
- [25] D. Widdowson, P. von Lockette, A. Erol, and M. A. Rodriguez, “A Computational Framework for Predicting Properties from Multifield Processing Conditions in Polymer Matrix Composites,” presented at the SMASIS 2020, Oct. 2020.
- [26] A. Erol, “Studies of Dipole-Driven Active Materials: Multifield Processing Simulations, Microstructure-Based Constitutive Modeling, and Macroscopic Active Composite Optimization,” PhD Dissertation, College of Engineering, The Pennsylvania State University, State College, Pennsylvania, 2019.
- [27] M. Štaffová, F. Ondreáš, J. Svatík, M. Zbončák, J. Jančář, and P. Lepcio, “3D printing and post-curing optimization of photopolymerized structures: Basic concepts and effective tools for improved thermomechanical properties,” *Polym. Test.*, vol. 108, p. 107499, Apr. 2022, doi: 10.1016/j.polymertesting.2022.107499.
- [28] P. Ch. Papaphilippou *et al.*, “Ofloxacin Removal from Aqueous Media by Means of Magnetoactive Electrospun Fibrous Adsorbents,” *Nanomaterials*, vol. 12, no. 20, p. 3648, Oct. 2022, doi: 10.3390/nano12203648.
- [29] D. J. Roach, C. M. Hamel, C. K. Dunn, M. V. Johnson, X. Kuang, and H. J. Qi, “The m4 3D printer: A multi-material multi-method additive manufacturing platform for future 3D printed structures,” *Addit. Manuf.*, vol. 29, p. 100819, Oct. 2019, doi: 10.1016/j.addma.2019.100819.
- [30] C. Zhou, Y. Chen, Z. Yang, and B. Khoshnevis, “Digital material fabrication using mask-image-projection-based stereolithography,” *Rapid Prototyp. J.*, vol. 19, no. 3, pp. 153–165, Apr. 2013, doi: 10.1108/13552541311312148.

- [31] H. Nassar, M. Ntagios, W. T. Navaraj, and R. Dahiva, "Multi-Material 3D Printed Bendable Smart Sensing Structures," in *2018 IEEE SENSORS*, New Delhi, Oct. 2018, pp. 1–4. doi: 10.1109/ICSENS.2018.8589625.
- [32] L. Li *et al.*, "Big Area Additive Manufacturing of High Performance Bonded NdFeB Magnets," *Sci. Rep.*, vol. 6, no. 1, p. 36212, Dec. 2016, doi: 10.1038/srep36212.
- [33] C. Zhang *et al.*, "Additive manufacturing of functionally graded materials: A review," *Mater. Sci. Eng. A*, vol. 764, p. 138209, Sep. 2019, doi: 10.1016/j.msea.2019.138209.
- [34] M. A. Skylar-Scott, J. Mueller, C. W. Visser, and J. A. Lewis, "Voxelated soft matter via multimaterial multinozzle 3D printing," *Nature*, vol. 575, no. 7782, pp. 330–335, Nov. 2019, doi: 10.1038/s41586-019-1736-8.
- [35] B. Stoner, J. Bartolai, D. V. Kaweesa, N. A. Meisel, and T. W. Simpson, "Achieving Functionally Graded Material Composition Through Bicontinuous Mesosstructural Geometry in Material Extrusion Additive Manufacturing," *JOM*, vol. 70, no. 3, pp. 413–418, Mar. 2018, doi: 10.1007/s11837-017-2669-z.
- [36] O. D. Velev and K. H. Bhatt, "On-chip micromanipulation and assembly of colloidal particles by electric fields," *Soft Matter*, vol. 2, no. 9, pp. 738–750, 2006, doi: 10.1039/b605052b.
- [37] M. Al Masud, A. Erol, C. Edson, Z. Ounaies, and P. von Lockette, "Towards complex microarchitectural composites using multi-field processing," in *Behavior and Mechanics of Multifunctional Materials XIII*, Denver, United States, Mar. 2019. doi: 10.1117/12.2515259.
- [38] B. Bharti and O. D. Velev, "Multidirectional, Multicomponent Electric Field Driven Assembly of Complex Colloidal Chains," *Z. Phys. Chem.*, vol. 229, no. 7–8, pp. 1075–1088, 2015, doi: 10.1515/zpch-2014-0543.
- [39] J. L. Mietta *et al.*, "Anisotropic Magnetoresistance and Piezoresistivity in Structured Fe<sub>3</sub>O<sub>4</sub>-Silver Particles in PDMS Elastomers at Room Temperature," *Langmuir*, vol. 28, no. 17, pp. 6985–6996, May 2012, doi: 10.1021/la204823k.
- [40] S. E. Jacobo, J. C. Apesteguy, R. Lopez Anton, N. N. Schegoleva, and G. V. Kurlyandskaya, "Influence of the preparation procedure on the properties of polyaniline based magnetic composites," *Eur. Polym. J.*, vol. 43, no. 4, pp. 1333–1346, Apr. 2007, doi: 10.1016/j.eurpolymj.2007.01.024.
- [41] Z. Varga, G. Filipcsei, and M. Zrínyi, "Smart composites with controlled anisotropy," *Polymer*, vol. 46, no. 18, pp. 7779–7787, Aug. 2005, doi: 10.1016/j.polymer.2005.03.102.
- [42] S. Pané, E. Gómez, and E. Vallés, "Influence of a magnetic field during the CoNi electrodeposition in the presence of magnetic nanoparticles," *J. Electroanal. Chem.*, vol. 615, no. 2, pp. 117–123, Apr. 2008, doi: 10.1016/j.jelechem.2007.12.002.
- [43] B. Nagarajan, A. F. Eufrazio Aguilera, M. Wiechmann, A. J. Qureshi, and P. Mertiny, "Characterization of magnetic particle alignment in photosensitive polymer resin: A preliminary study for additive manufacturing processes," *Addit. Manuf.*, vol. 22, pp. 528–536, Aug. 2018, doi: 10.1016/j.addma.2018.05.046.
- [44] S. Jin, T. N. Tiefel, and R. Wolfe, "Directionally-conductive, optically-transparent composites by magnetic alignment," *IEEE Trans. Magn.*, vol. 28, no. 5, pp. 2211–2213, Sep. 1992, doi: 10.1109/20.179446.
- [45] Z. Varga, G. Filipcsei, and M. Zrínyi, "Magnetic field sensitive functional elastomers with tuneable elastic modulus," *Polymer*, vol. 47, no. 1, pp. 227–233, Jan. 2006, doi: 10.1016/j.polymer.2005.10.139.

- [46] J. S. Leng, W. M. Huang, X. Lan, Y. J. Liu, and S. Y. Du, "Significantly reducing electrical resistivity by forming conductive Ni chains in a polyurethane shape-memory polymer/carbon-black composite," *Appl. Phys. Lett.*, vol. 92, no. 20, p. 204101, May 2008, doi: 10.1063/1.2931049.
- [47] C. Breznak and P. von Lockette, "Particle orientation and bulk properties of magnetoactive elastomers fabricated with aligned barium hexaferrite," *J. Mater. Res.*, vol. 34, no. 6, pp. 972–981, Mar. 2019, doi: 10.1557/jmr.2018.496.
- [48] S. Odenbach, Ed., *Colloidal Magnetic Fluids: Basics, Development and Application of Ferrofluids*, vol. 763. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009. doi: 10.1007/978-3-540-85387-9.
- [49] J. M. M. Linke and S. Odenbach, "Anisotropy of the magnetoviscous effect in a ferrofluid with weakly interacting magnetite nanoparticles," *J. Phys. Condens. Matter J Phys Condens Matter*, vol. 27, p. 7, 2015, doi: 10.1088/0953-8984/27/17/176001.
- [50] L. M. Pop and S. Odenbach, "Capillary viscosimetry on ferrofluids," 2008, doi: 10.1088/0953-8984/20/20/204139.
- [51] H. Le Ferrand, "External fields for the fabrication of highly mineralized hierarchical architectures," *J. Mater. Res.*, vol. 34, no. 1, pp. 169–193, Jan. 2019, doi: 10.1557/jmr.2018.304.
- [52] M. A. Rodriguez and P. VonLockette, "EVOLUTION OF TEXTURE IN THE FABRICATION OF MAGNETO-ACTIVE ELASTOMERS," presented at the Conference on Smart Materials, Adaptive Structures and Intelligent Systems, Snowbird UT, 2017.
- [53] A. F. Demirörs, D. Courty, R. Libanori, and A. R. Studart, "Periodically microstructured composite films made by electric- and magnetic-directed colloidal assembly," *Proc. Natl. Acad. Sci.*, vol. 113, no. 17, pp. 4623–4628, Apr. 2016, doi: 10.1073/pnas.1524736113.
- [54] Y. Benveniste, "A NEW APPROACH TO THE APPLICATION OF MORI-TANAKA'S THEORY IN COMPOSITE MATERIALS," *Mech. Mater.*, vol. 6, pp. 147–157, 1987.
- [55] J. C. Halpin, S. T. Louis, and J. L. Kardos, "The Halpin-Tsai Equations: A Review," *Polym. Eng. Sci.*, vol. 16, no. 5, 1976.
- [56] J. C. Halpin, *Effects of Environmental Factors on Composite Materials*. WRIGHT-PATTERSON AIR FORCE BASE, OHIO: Air Force Materials Laboratory, Air Force Systems Command, 1969.
- [57] J. C. Halpin, *Primer on Composite Materials: Analysis*. Lancaster, Pennsylvania: Technomic Pub. Co, 1984.
- [58] J. D. Eshelby, "The determination of the elastic field of an ellipsoidal inclusion, and related problems," *RE Peierls FRS*, pp. 376–396, 1957.
- [59] J. Qu and M. Cherkaoui, *Fundamentals of Micromechanics of Solids*. Hoboken, New Jersey: John Wiley & Sons, 2006.
- [60] E. Giner, A. Vercher, M. Marco, and C. Arango, "Estimation of the reinforcement factor  $\xi$  for calculating the transverse stiffness E2 with the Halpin–Tsai equations using the finite element method," *Compos. Struct.*, vol. 124, pp. 402–408, Jun. 2015, doi: 10.1016/j.compstruct.2015.01.008.
- [61] J. C. M. Garnett, "XII. Colours in metal glasses and in metallic films," *Philos. Trans. R. Soc. Lond. Ser. Contain. Pap. Math. Phys. Character*, vol. 203, no. 359–371, pp. 385–420, Jan. 1904, doi: 10.1098/rsta.1904.0024.
- [62] K. W. Wagner, "Erklärung der Dielektrischen Nachwirkungsvorgänge auf Grund Maxwellscher Vorstellungen," *Phys.-Tech. Reichsanst.*, pp. 371–387.

- [63] R. W. Sillars, "The Properties of a Dielectric Containing Semi-Conducting Particles of Various Shapes," *Inst. Electr. Eng.*, vol. 12, no. 35, pp. 139–155, 1936.
- [64] H. Fricke, "The Maxwell-Wagner Dispersion in a Suspension of Ellipsoids," *J. Phys. Chem.*, vol. 57, no. 9, pp. 934–937, Sep. 1953, doi: 10.1021/j150510a018.
- [65] H. Fricke, "A Mathematical Treatment of the Electric Conductivity and Capacity of Disperse Systems I. The Electric Conductivity of a Suspension of Homogeneous Spheroids," *Phys. Rev.*, vol. 24, no. 5, pp. 575–587, Nov. 1924, doi: 10.1103/PhysRev.24.575.
- [66] D. A. G. Bruggeman, "Berechnung verschiedener physikalischer Konstanten von heterogenen Substanzen. I. Dielektrizitätskonstanten und Leitfähigkeiten der Mischkörper aus isotropen Substanzen," *Ann. Phys.*, vol. 416, no. 7, pp. 636–664, 1935, doi: 10.1002/andp.19354160705.
- [67] G. Banhegyi, "Comparison of electrical mixture rules for composites," *Colloid Polym. Sci.*, vol. 264, no. 12, pp. 1030–1050, Dec. 1986, doi: 10.1007/BF01410321.
- [68] C. J. F. Böttcher, "The dielectric constant of crystalline powders," *Recl. Trav. Chim. Pays-Bas*, vol. 64, no. 2, pp. 47–51, 1945, doi: 10.1002/recl.19450640205.
- [69] D. POLDER and J. H. VAN SANTEN, "THE EFFECTIVE PERMEABILITY OF MIXTURES OF SOLIDS," *Phys. XII*, vol. 9, 1946.
- [70] W. Y. Hsu, T. D. Gierke, and C. J. Molnar, "Morphological effects on the physical properties of polymer composites," *Macromolecules*, vol. 16, 1983.
- [71] P. Castañeda, "The effect of spatial distribution on the effective behavior of composite materials and cracked media," *J. Mech. Phys. Solids*, vol. 43, no. 12, pp. 1919–1951, Dec. 1995, doi: 10.1016/0022-5096(95)00058-Q.
- [72] M. Pahlavanpour, H. Moussaddy, E. Ghossein, P. Hubert, and M. Lévesque, "Prediction of elastic properties in polymer–clay nanocomposites: Analytical homogenization methods and 3D finite element modeling," *Comput. Mater. Sci.*, vol. 79, pp. 206–215, Nov. 2013, doi: 10.1016/j.commatsci.2013.06.029.
- [73] J. Segurado and J. Llorca, "A numerical approximation to the elastic properties of sphere-reinforced composites," *J. Mech. Phys. Solids*, vol. 50, pp. 2107–2121, 2002.
- [74] A. Trofimov, B. Drach, and I. Sevostianov, "Effective elastic properties of composites with particles of polyhedral shapes," *Int. J. Solids Struct.*, 2017, doi: 10.1016/j.ijsolstr.2017.04.037.
- [75] L. Pérez *et al.*, "DEM–FEA estimation of pores arrangement effect on the compressive Young's modulus for Mg foams," *Comput. Mater. Sci.*, vol. 110, pp. 281–286, Dec. 2015, doi: 10.1016/j.commatsci.2015.08.042.
- [76] D. Ivaneyko, V. Toshchevnikov, M. Saphiannikova, and G. Heinrich, "Effects of particle distribution on mechanical properties of magneto-sensitive elastomers in a homogeneous magnetic field," *Condens. Matter Phys.*, vol. 15, no. 3, p. 33601, Sep. 2012, doi: 10.5488/CMP.15.33601.
- [77] H. V. Ly, F. Reitich, M. R. Jolly, H. T. Banks, and K. Ito, "Simulations of Particle Dynamics in Magnetorheological Fluids | Elsevier Enhanced Reader," *Journal of Computational Physics*, vol. 155, no. 1, pp. 160–1777, Oct. 1999, doi: <https://doi.org/10.1006/jcph.1999.6335>.
- [78] S. Bargmann *et al.*, "Generation of 3D representative volume elements for heterogeneous materials: A review," *Prog. Mater. Sci.*, vol. 96, pp. 322–384, Jul. 2018, doi: 10.1016/j.pmatsci.2018.02.003.



- [79] F. Nilsson *et al.*, “Simulating the effective electric conductivity of polymer composites with high aspect ratio fillers,” *Compos. Sci. Technol.*, vol. 132, pp. 16–23, Aug. 2016, doi: 10.1016/j.compscitech.2016.06.008.
- [80] A. El Moumen, T. Kanit, A. Imad, and H. El Minor, “Effect of reinforcement shape on physical properties and representative volume element of particles-reinforced composites: Statistical and numerical approaches,” *Mech. Mater.*, vol. 83, pp. 1–16, Apr. 2015, doi: 10.1016/j.mechmat.2014.12.008.
- [81] T. B. Jones, *Electromechanics of Particles*. Cambridge University Press, 1995.
- [82] S. P. Gurrum, J. H. Zhao, and D. R. Edwards, “Inclusion interaction and effective material properties in a particle-filled composite material system,” *J. Mater. Sci.*, vol. 46, no. 1, pp. 101–107, 2011, doi: 10.1007/s10853-010-4844-2.
- [83] Y. Chen *et al.*, “Transparent and through thickness conductive polystyrene films using external magnetic fields for ‘Z’ alignment of nickel nanoparticles,” *Nanoscale*, vol. 7, no. 35, pp. 14636–14642, 2015, doi: 10.1039/C5NR03328D.
- [84] S. Dong, S. Chen, B. Li, and X. Wang, “Electrical properties of anisotropic graphene/PDMS composites induced by aligned ferromagnetic particles,” *J. Intell. Mater. Syst. Struct.*, p. 1045389X2096986, Nov. 2020, doi: 10.1177/1045389X20969864.
- [85] S. Dong and X. Wang, “Alignment of carbon iron into polydimethylsiloxane to create conductive composite with low percolation threshold and high piezoresistivity: experiment and simulation,” *Smart Mater. Struct.*, vol. 26, no. 4, p. 045027, Apr. 2017, doi: 10.1088/1361-665X/aa62d2.
- [86] S. Kalnaus, A. S. Sabau, S. Newman, W. E. Tenhaeff, C. Daniel, and N. J. Dudney, “Effective conductivity of particulate polymer composite electrolytes using random resistor network method,” *Solid State Ion.*, vol. 199–200, pp. 44–53, Sep. 2011, doi: 10.1016/j.ssi.2011.07.016.
- [87] F. Dalmas, R. Dendievel, L. Chazeau, J.-Y. Cavaillé, and C. Gauthier, “Carbon nanotube-filled polymer composites. Numerical simulation of electrical conductivity in three-dimensional entangled fibrous networks,” *Acta Mater.*, vol. 54, no. 11, pp. 2923–2931, Jun. 2006, doi: 10.1016/j.actamat.2006.02.028.
- [88] C. Feng and L. Jiang, “Micromechanics modeling of the electrical conductivity of carbon nanotube (CNT)–polymer nanocomposites,” *Compos. Part Appl. Sci. Manuf.*, vol. 47, pp. 143–149, Apr. 2013, doi: 10.1016/j.compositesa.2012.12.008.
- [89] S. Y. Kim, Y. J. Noh, and J. Yu, “Prediction and experimental validation of electrical percolation by applying a modified micromechanics model considering multiple heterogeneous inclusions,” *Compos. Sci. Technol.*, vol. 106, pp. 156–162, Jan. 2015, doi: 10.1016/j.compscitech.2014.11.015.
- [90] R. Hashemi and G. J. Weng, “A theoretical treatment of graphene nanocomposites with percolation threshold, tunneling-assisted conductivity and microcapacitor effect in AC and DC electrical settings,” *Carbon*, vol. 96, pp. 474–490, Jan. 2016, doi: 10.1016/j.carbon.2015.09.103.
- [91] S. S. Rahatekar, M. S. P. Shaffer, and J. A. Elliott, “Modelling percolation in fibre and sphere mixtures: Routes to more efficient network formation,” *Compos. Sci. Technol.*, vol. 70, no. 2, pp. 356–362, Feb. 2010, doi: 10.1016/j.compscitech.2009.11.007.
- [92] C. Dussert, G. Rasigni, and M. Rasigni, “Minimal spanning tree approach to percolation and conductivity threshold,” *Phys. Lett. A*, vol. 139, no. 1–2, pp. 35–38, Jul. 1989, doi: 10.1016/0375-9601(89)90603-8.

- [93] S. J. Russell, P. Norvig, and E. Davis, *Artificial intelligence: a modern approach*, 3rd ed. Upper Saddle River: Prentice Hall, 2010.
- [94] A. Michaels, M. C. Wu, and E. Yablonovitch, "Hierarchical Design and Optimization of Silicon Photonics," *IEEE J. Sel. Top. Quantum Electron.*, vol. 26, no. 2, pp. 1–12, Mar. 2020, doi: 10.1109/JSTQE.2019.2935299.
- [95] T. W. Hughes, M. Minkov, I. A. D. Williamson, and S. Fan, "Adjoint Method and Inverse Design for Nonlinear Nanophotonic Devices," *ACS Photonics*, vol. 5, no. 12, pp. 4781–4787, Dec. 2018, doi: 10.1021/acsphotonics.8b01522.
- [96] E. Asadi, M. G. da Silva, C. H. Antunes, L. Dias, and L. Glicksman, "Multi-objective optimization for building retrofit: A model using genetic algorithm and artificial neural network and an application," *Energy Build.*, vol. 81, pp. 444–456, Oct. 2014, doi: 10.1016/j.enbuild.2014.06.009.
- [97] D. Yadav, D. Chhabra, R. Kumar Garg, A. Ahlawat, and A. Phogat, "Optimization of FDM 3D printing process parameters for multi-material using artificial neural network," *Mater. Today Proc.*, vol. 21, pp. 1583–1591, 2020, doi: 10.1016/j.matpr.2019.11.225.
- [98] Y. Zhang, S. Wang, and G. Ji, "A Comprehensive Survey on Particle Swarm Optimization Algorithm and Its Applications," *Math. Probl. Eng.*, vol. 2015, pp. 1–38, 2015, doi: 10.1155/2015/931256.
- [99] F. Kheiri, "A review on optimization methods applied in energy-efficient building geometry and envelope design," *Renew. Sustain. Energy Rev.*, vol. 92, pp. 897–920, Sep. 2018, doi: 10.1016/j.rser.2018.04.080.
- [100] C.-K. Looi, "Neural network methods in combinatorial optimization," *Comput. Oper. Res.*, vol. 19, no. 3–4, pp. 191–208, Apr. 1992, doi: 10.1016/0305-0548(92)90044-6.
- [101] D. McNeish and E. L. Hamaker, "A primer on two-level dynamic structural equation models for intensive longitudinal data in Mplus," *Psychol. Methods*, vol. 25, no. 5, pp. 610–635, Oct. 2020, doi: 10.1037/met0000250.
- [102] K. Wang and W. Sun, "Meta-modeling game for deriving theory-consistent, microstructure-based traction–separation laws via deep reinforcement learning," *Comput. Methods Appl. Mech. Eng.*, vol. 346, pp. 216–241, Apr. 2019, doi: 10.1016/j.cma.2018.11.026.
- [103] C. Yang, Y. Kim, S. Ryu, and G. X. Gu, "Prediction of composite microstructure stress-strain curves using convolutional neural networks," *Mater. Des.*, vol. 189, p. 108509, Apr. 2020, doi: 10.1016/j.matdes.2020.108509.
- [104] G. Srinivasu, R. N. Rao, T. K. Nandy, and A. Bhattacharjee, "Artificial Neural Network Approach for Prediction of Titanium Alloy Stress-Strain Curve," *Procedia Eng.*, vol. 38, pp. 3709–3714, 2012, doi: 10.1016/j.proeng.2012.06.426.
- [105] G. R. Liu, X. Han, Y. G. Xu, and K. Y. Lam, "Material characterization of functionally graded material by means of elastic waves and a progressive-learning neural network," *Compos. Sci. Technol.*, vol. 61, no. 10, pp. 1401–1411, Aug. 2001, doi: 10.1016/S0266-3538(01)00033-1.
- [106] J. S. Huang, J. X. Liew, A. S. Ademiloye, and K. M. Liew, "Artificial Intelligence in Materials Modeling and Design," *Arch. Comput. Methods Eng.*, vol. 28, no. 5, pp. 3399–3413, Aug. 2021, doi: 10.1007/s11831-020-09506-1.
- [107] C.-T. Chen and G. X. Gu, "Machine learning for composite materials," *MRS Commun.*, vol. 9, no. 2, pp. 556–566, Jun. 2019, doi: 10.1557/mrc.2019.32.

- [108] J. Chapman and R. Ramprasad, "Predicting the dynamic behavior of the mechanical properties of platinum with machine learning," *J. Chem. Phys.*, vol. 152, no. 22, p. 224709, Jun. 2020, doi: 10.1063/5.0008955.
- [109] D. Mohammadzadeh S., S.-F. Kazemi, A. Mosavi, E. Nasseralshariati, and J. H. M. Tah, "Prediction of Compression Index of Fine-Grained Soils Using a Gene Expression Programming Model," *Infrastructures*, vol. 4, no. 2, p. 26, May 2019, doi: 10.3390/infrastructures4020026.
- [110] M. G. M. Abdolrasol *et al.*, "Artificial Neural Networks Based Optimization Techniques: A Review," *Electronics*, vol. 10, no. 21, p. 2689, Nov. 2021, doi: 10.3390/electronics10212689.
- [111] D. Z. Huang, K. Xu, C. Farhat, and E. Darve, "Learning constitutive relations from indirect observations using deep neural networks," *J. Comput. Phys.*, vol. 416, p. 109491, Sep. 2020, doi: 10.1016/j.jcp.2020.109491.
- [112] K. Xu, D. Z. Huang, and E. Darve, "Learning constitutive relations using symmetric positive definite neural networks," *J. Comput. Phys.*, vol. 428, p. 110072, Mar. 2021, doi: 10.1016/j.jcp.2020.110072.
- [113] B. Jena, S. Saxena, G. K. Nayak, L. Saba, N. Sharma, and J. S. Suri, "Artificial intelligence-based hybrid deep learning models for image classification: The first narrative review," *Comput. Biol. Med.*, vol. 137, p. 104803, Oct. 2021, doi: 10.1016/j.compbiomed.2021.104803.
- [114] K. Roose, "AI-Generated Art Won a Prize. Artists Aren't Happy," *The New York Times*, Sep. 02, 2022. [Online]. Available: <https://www.nytimes.com/2022/09/02/technology/ai-artificial-intelligence-artists.html>
- [115] H. Guo and W. Wang, "Granular support vector machine: a review," *Artif. Intell. Rev.*, vol. 51, no. 1, pp. 19–32, Jan. 2019, doi: 10.1007/s10462-017-9555-5.
- [116] Z. Zhang, "Introduction to machine learning: k-nearest neighbors," *Ann. Transl. Med.*, vol. 4, no. 11, pp. 218–218, Jun. 2016, doi: 10.21037/atm.2016.03.37.
- [117] Y. Song and Y. Lu, "Decision tree methods: applications for classification and prediction," vol. 27, no. 2, p. 6, 2015.
- [118] G. D. Luca, "SVM Vs Neural Network | Baeldung on Computer Science," Aug. 06, 2020. <https://www.baeldung.com/cs/svm-vs-neural-network> (accessed Nov. 27, 2021).
- [119] F. E. Bock, R. C. Aydin, C. J. Cyron, N. Huber, S. R. Kalidindi, and B. Klusemann, "A Review of the Application of Machine Learning and Data Mining Approaches in Continuum Materials Mechanics," *Front. Mater.*, vol. 6, p. 110, May 2019, doi: 10.3389/fmats.2019.00110.
- [120] "Layer activation functions," *Keras*. <https://keras.io/api/layers/activations/> (accessed Nov. 06, 2022).
- [121] J. Xiong, G. Zhang, J. Hu, and L. Wu, "Bead geometry prediction for robotic GMAW-based rapid manufacturing through a neural network and a second-order regression analysis," *J. Intell. Manuf.*, vol. 25, no. 1, pp. 157–163, Feb. 2014, doi: 10.1007/s10845-012-0682-1.
- [122] A. E. Kurtoglu and D. Bakbak, "Modeling the Shear Strength of Reinforced Aerated Concrete Slabs via Support Vector Regression," p. 9, 2019.
- [123] Y. Bengio, A. Lodi, and A. Prouvost, "Machine learning for combinatorial optimization: A methodological tour d'horizon," *Eur. J. Oper. Res.*, vol. 290, no. 2, pp. 405–421, Apr. 2021, doi: 10.1016/j.ejor.2020.07.063.
- [124] O. Wichrowska *et al.*, "Learned Optimizers that Scale and Generalize," p. 10.

- [125] M. Andrychowicz, M. Denil, S. Gómez, M. W. Hoffman, D. Pfau, and T. Schaul, "Learning to learn by gradient descent by gradient descent," p. 9.
- [126] K. Li and J. Malik, "Learning to Optimize Neural Nets." arXiv, Nov. 30, 2017. Accessed: Nov. 06, 2022. [Online]. Available: <http://arxiv.org/abs/1703.00441>
- [127] P. Perdikaris, D. Venturi, and G. E. Karniadakis, "Multifidelity Information Fusion Algorithms for High-Dimensional Systems and Massive Data sets," *SIAM J. Sci. Comput.*, vol. 38, no. 4, pp. B521–B538, Jan. 2016, doi: 10.1137/15M1055164.
- [128] H. Wang, E. Li, and G. Y. Li, "The least square support vector regression coupled with parallel sampling scheme metamodeling technique and application in sheet forming optimization," *Mater. Des.*, vol. 30, no. 5, pp. 1468–1479, May 2009, doi: 10.1016/j.matdes.2008.08.014.
- [129] M. Kallio, T. Lindroos, S. Aalto, E. Järvinen, T. Kärnä, and T. Meinander, "Dynamic compression testing of a tunable spring element consisting of a magnetorheological elastomer," *Smart Mater. Struct.*, vol. 16, no. 2, pp. 506–514, Apr. 2007, doi: 10.1088/0964-1726/16/2/032.
- [130] H. Deng, X. Gong, and L. Wang, "Development of an adaptive tuned vibration absorber with magnetorheological elastomer," *Smart Mater. Struct.*, vol. 15, no. 5, pp. N111–N116, Oct. 2006, doi: 10.1088/0964-1726/15/5/N02.
- [131] G. Y. Zhou and Q. Wang, "Magnetorheological elastomer-based smart sandwich beams with nonconductive skins," *Smart Mater. Struct.*, vol. 14, no. 5, pp. 1001–1009, Oct. 2005, doi: 10.1088/0964-1726/14/5/038.
- [132] M. Mohebi, N. Jamasbi, and J. Liu, "Simulation of the formation of nonequilibrium structures in magnetorheological fluids subject to an external magnetic field," *Phys. Rev. E*, vol. 54, no. 5, pp. 5407–5413, Nov. 1996, doi: 10.1103/PhysRevE.54.5407.
- [133] R. E. Rosenzweig, *Ferrohydrodynamics*. Mineola NY: Dover, 1997.
- [134] L. Paramonov and S. N. Yaliraki, "The directional contact distance of two ellipsoids: Coarse-grained potentials for anisotropic interactions," *J. Chem. Phys.*, vol. 123, no. 19, p. 194111, Nov. 2005, doi: 10.1063/1.2102897.
- [135] D. J. Cleaver, C. M. Care, M. P. Allen, and M. P. Neal, "Extension and generalization of the Gay-Berne potential," *Phys. Rev. E*, vol. 54, no. 1, pp. 559–567, Jul. 1996, doi: 10.1103/PhysRevE.54.559.
- [136] G. A. Lyngdoh and S. Das, "Integrating multiscale numerical simulations with machine learning to predict the strain sensing efficiency of nano-engineered smart cementitious composites," *Mater. Des.*, vol. 209, p. 109995, Jul. 2021, doi: 10.1016/j.matdes.2021.109995.
- [137] J. G. Simmons, "Generalized Formula for the Electric Tunnel Effect between Similar Electrodes Separated by a Thin Insulating Film," *J. Appl. Phys.*, vol. 34, no. 1793, p. 12, 1963.
- [138] A. Oskouyi, U. Sundararaj, and P. Mertiny, "Tunneling Conductivity and Piezoresistivity of Composites Containing Randomly Dispersed Conductive Nano-Platelets," *Materials*, vol. 7, no. 4, pp. 2501–2521, Mar. 2014, doi: 10.3390/ma7042501.
- [139] J. Payandehpeyman, M. Mazaheri, and M. Khamsehchi, "Prediction of electrical conductivity of polymer-graphene nanocomposites by developing an analytical model considering interphase, tunneling and geometry effects," *Compos. Commun.*, vol. 21, p. 100364, Oct. 2020, doi: 10.1016/j.coco.2020.100364.
- [140] M. Hori, "Double-inclusion model and overall moduli of multi-phase composites," p. 18.

- [141] C. Chiew and M. H. Malakooti, "A double inclusion model for liquid metal polymer composites," *Compos. Sci. Technol.*, vol. 208, p. 108752, May 2021, doi: 10.1016/j.compscitech.2021.108752.
- [142] P. Lu, Y. W. Leong, P. K. Pallathadka, and C. B. He, "Effective moduli of nanoparticle reinforced composites considering interphase effect by extended double-inclusion model – Theory and explicit expressions," *Int. J. Eng. Sci.*, vol. 73, pp. 33–55, Dec. 2013, doi: 10.1016/j.ijengsci.2013.08.003.
- [143] J. A. Stratton, *Electromagnetic Theory*. New York and London: McGraw-Hill Book Company, 1941.
- [144] D. Rousselle, A. Berthault, O. Acher, J. P. Bouchaud, and P. G. Zerah, "Effective medium at finite frequency: Theory and experiment," *J. Appl. Phys.*, vol. 74, no. 1, pp. 475–479, Jul. 1993, doi: 10.1063/1.355256.
- [145] R. T. Marler and J. S. Arora, "Survey of multi-objective optimization methods for engineering," *Struct. Multidiscip. Optim.*, vol. 26, no. 6, pp. 369–395, Apr. 2004, doi: 10.1007/s00158-003-0368-6.
- [146] A. Candelieri, R. Perego, and F. Archetti, "Green machine learning via augmented Gaussian processes and multi-information source optimization," *Soft Comput.*, vol. 25, no. 19, pp. 12591–12603, Oct. 2021, doi: 10.1007/s00500-021-05684-7.
- [147] P. Shepherd, K. K. Mallick, and R. J. Green, "Magnetic and structural properties of M-type barium hexaferrite prepared by co-precipitation," *J. Magn. Magn. Mater.*, vol. 311, no. 2, pp. 683–692, Apr. 2007, doi: 10.1016/j.jmmm.2006.08.046.
- [148] D. S. Klygach *et al.*, "Measurement of permittivity and permeability of barium hexaferrite," *J. Magn. Magn. Mater.*, vol. 465, pp. 290–294, Nov. 2018, doi: 10.1016/j.jmmm.2018.05.054.
- [149] Z. Wang, A. A. Volinsky, and N. D. Gallant, "Crosslinking effect on polydimethylsiloxane elastic modulus measured by custom-built compression instrument," *J. Appl. Polym. Sci.*, vol. 131, no. 22, p. n/a-n/a, Nov. 2014, doi: 10.1002/app.41050.
- [150] F. Chen, I. Sevostianov, A. Giraud, and D. Grgic, "Evaluation of the effective elastic and conductive properties of a material containing concave pores," *Int. J. Eng. Sci.*, vol. 97, pp. 60–68, 2015, doi: 10.1016/j.ijengsci.2015.08.012.
- [151] D. Widdowson, D. Papula, Z. Ounaies, and A. Erol, "Multi-Objective Optimization of Predicted Magnetic Properties from Multifield Processing Conditions in Polymer Matrix Particle Composites," p. 8.
- [152] D.-H. Xiao, L.-X. Guo, W. Liu, W.-Q. Qin, and M.-Y. Hou, "Latin Hypercube Sampling and Support Vector Regression-Assisted Analysis of Electromagnetic Scattering Characteristics of Targets," p. 3.
- [153] "TensorFlow v2.11.0." [Online]. Available: [https://www.tensorflow.org/api\\_docs/python/tf/nm/](https://www.tensorflow.org/api_docs/python/tf/nm/)
- [154] A. A. Wao and B. K. Soni, "Performance Analysis of Sigmoid and Relu Activation Functions in Deep Neural Network," in *Intelligent Systems*, A. Sheth, A. Sinhal, A. Shrivastava, and A. K. Pandey, Eds. Singapore: Springer Singapore, 2021, pp. 39–52. doi: 10.1007/978-981-16-2248-9\_5.
- [155] J. Horvath, "A Brief History of 3D Printing," in *Mastering 3D Printing*, Berkeley, CA: Apress, 2014, pp. 3–10. doi: 10.1007/978-1-4842-0025-4\_1.
- [156] "Wikipedia." [Online]. Available: <https://en.wikipedia.org/wiki/Wikipedia>
- [157] "VLC." [Online]. Available: <https://www.videolan.org/vlc/>

- [158] “RaspberryPi.” [Online]. Available: <https://www.raspberrypi.com/about/>
- [159] R. Tabarés and H. Kuittinen, “A tale of two innovation cultures: Bridging the gap between makers and manufacturers,” *Technol. Soc.*, vol. 63, p. 101352, Nov. 2020, doi: 10.1016/j.techsoc.2020.101352.
- [160] J. Brainard and J. Kaiser, “U.S. to require free access to papers on all research it funds,” *Science*, vol. 377, no. 6610, pp. 1026–1027, Sep. 2022, doi: 10.1126/science.ade6577.
- [161] “LibreCAD.” [Online]. Available: <https://librecad.org/>
- [162] “FreeCAD.” [Online]. Available: <https://www.freecadweb.org/>
- [163] “FEniCSx.” [Online]. Available: <https://fenicsproject.org/>
- [164] “FreeFEM.” [Online]. Available: <https://freefem.org/>
- [165] “SciLab.” [Online]. Available: <https://www.scilab.org/>
- [166] “GNU Octave.” [Online]. Available: <https://octave.org/>
- [167] “Sage.” [Online]. Available: <https://www.sagemath.org/index.html>

## APPENDIX A

The magnetic dipole moment,  $\mathbf{j}$ , can be converted into emu by dimensional analysis as shown below, where  $\mathbf{j}$  has units of  $[Tm^3]$

$$[Tm^3] = [Wb\ m] \quad (A-1)$$

$$[Wb\ m] \frac{[emu]}{4\pi 10^{-10} [Wb\ m]} = [emu]$$

This dipole moment can then be mass normalized by either the particle mass or the composite mass.

$$\frac{j\ [emu]}{m_{composite}[g]} = \sigma \left[ \frac{emu}{g} \right] \quad (A-2)$$

$$\frac{j\ [emu]}{m_{particles}[g^*]} = \sigma \left[ \frac{emu}{g^*} \right] \quad (A-3)$$

## APPENDIX B

With an AC voltage applied to the system the response will be a complex current. The circuit in Figure 2-7 can be analyzed via the state-space equations ( B-1 ) through ( B-3 ).

$$e_C = e_R \quad (B-1)$$

$$i_C = C \frac{de_C}{dt} \quad (B-2)$$

$$Ri_R = e_R \quad (B-3)$$

where  $e_C$  and  $e_R$  are the capacitor and resistor voltage respectively,  $i_C$ ,  $i_R$ , and  $i_o$  are the capacitor, resistor, and total output current respectively,  $C$  is the capacitance, and  $R$  is the resistance. These equations can be used to substituted into Kirchhoff's first law, shown in equation ( B-6 ), to obtain the differential equation that describes the system's response, shown in equation ( B-7 ).

$$i_C + i_R = i_o \quad (B-4)$$

$$C \frac{de_C}{dt} + \frac{e_r}{R} = i_o \quad (B-5)$$

$$(B-6)$$

The transfer function between the input voltage and output current can be found by using a Fourier transform and solving for the ratio of the input to output, shown in equation ( B-6 ). This result of this function is a complex number can be used to determine the phase angle of the response as shown in equation ( B-7 ).

$$\frac{I_{out}}{V_{in}} = T(j\omega) = j\omega C + \frac{1}{R} \quad (B-6)$$

$$\phi = \tan^{-1} \left( \frac{\text{Im}(T(j\omega))}{\text{Re}(T(j\omega))} \right) \quad (B-7)$$



The steady state sinusoidal response of a system is defined as

$$y(t) = |T(j\omega)| \sin(\omega t + \phi) \quad (\text{B-8})$$

$$i_o(t) = |T(j\omega)| \sin(\omega t + \phi) \quad (\text{B-9})$$

The output current is the sum of the capacitive and conductive components of the current. These are defined above in equations ( B-10) and ( B-11) and yield the following relationships for a sinusoidal input.

$$i_{cap}(t) = C \frac{d \sin(\omega t)}{dt} = \omega C \cos(\omega t) \quad (\text{B-10})$$

$$i_{cond}(t) = \frac{1}{R} \sin(\omega t) \quad (\text{B-11})$$

$$i_o(t) = i_{cond}(t) + i_{cap}(t) = \frac{1}{R} \sin(\omega t) + \omega C \cos(\omega t) \quad (\text{B-12})$$

Relating equation ( B-12) to the solution presented in equation ( B-9) the material constants  $R$  and  $C$  can be determined.

$$i_o(t) = \frac{1}{R} \sin(\omega t) + \omega C \cos(\omega t) = |T(j\omega)| \sin(\omega t + \phi) \quad (\text{B-13})$$

$$\frac{1}{R} \sin(\omega t) + \omega C \cos(\omega t) = |T(j\omega)| \sin(\omega t) \cos(\phi) + |T(j\omega)| \cos(\omega t) \sin(\phi) \quad (\text{B-14})$$

$$\frac{1}{R} = |T(j\omega)| \sin(\phi) \text{ and } \omega C = |T(j\omega)| \cos(\phi) \quad (\text{B-15})$$

In the case where the material properties are not known but the current response can be measured, and the voltage input is known the transfer function is defined by the measured value of the current and the voltage.

$$|T(j\omega)| = \frac{|I_{out}(t)|}{|V_{in}(t)|} \quad (\text{B-16})$$

$$\frac{1}{R} = \frac{I_{out} \sin(\phi)}{V_{in}} \quad (\text{B-17})$$

$$C = \frac{I_{out} \cos(\phi)}{\omega V_{in}} \quad (\text{B-18})$$

## APPENDIX C

The search algorithm used to find all conductive pathways is presented here both in logic flow and the code used to execute the search.

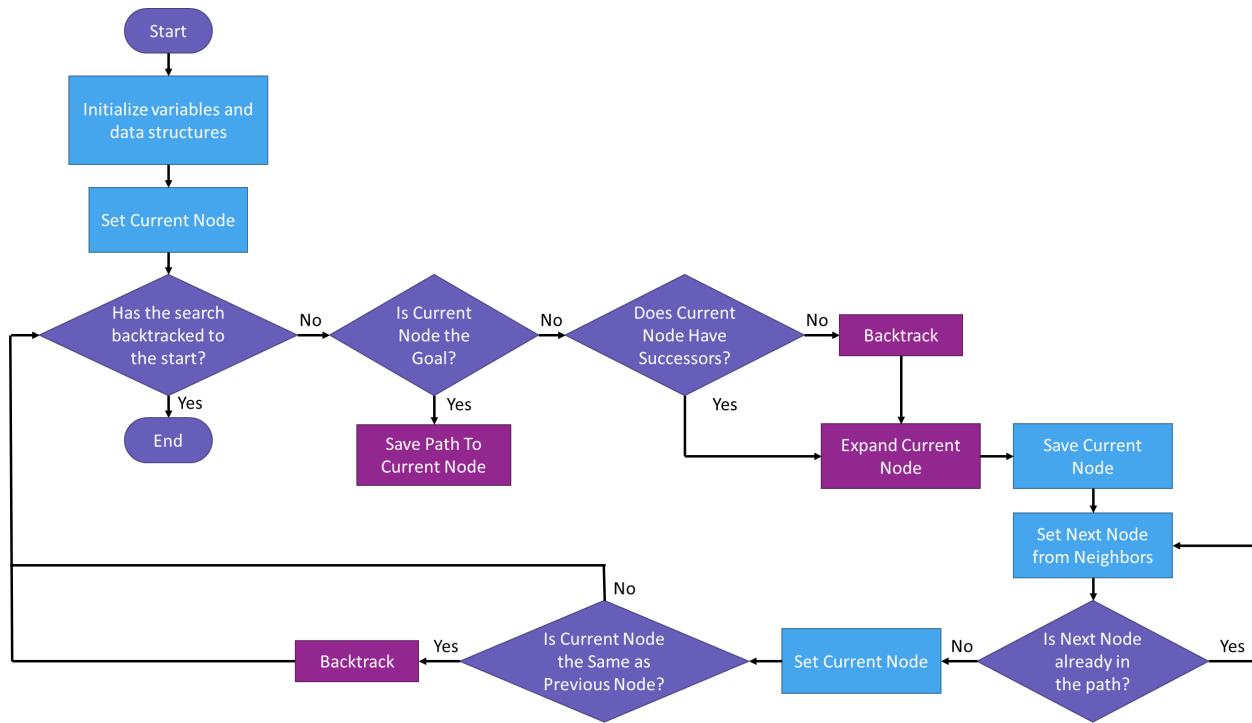


Figure 0-1: Flow Chart of All Paths Search Algorithm

```

def depthFirstSearch(problem):
    currentPath={}
    Nodes={}
    paths=[]
    points=[]

    ###Init problem with first node
    topVertex=problem.start
    currentNode=Node(topVertex,None)
    Nodes[topVertex]=currentNode
    movesStack=util.Stack()

    ###Get my neighbors
    currentPath[currentNode.state()]=currentNode
    neighbors=expand(currentNode,problem,Nodes)
  
```

```

while True:
    ###Get my neighbors
    neighbors=expand(currentNode,problem,Nodes)

    if problem.isGoalState(currentNode.state()):
        successfulPath=[]
        for point in currentPath:
            successfulPath.append(point)
        points.append(successfulPath)

        ###remove the current node (Goal) and backtrack to where there is a new path possible
        currentPath.pop(currentNode.state(),None)
        tempList=list(currentPath.items())
        if len(tempList)==0:
            break
        currentNode=tempList[len(tempList)-1][1]
        [currentNode,currentPath]=backtrack(currentNode,currentPath,problem,Nodes)
        neighbors=expand(currentNode,problem,Nodes)

    elif currentNode.successors==None:
        [currentNode,currentPath]=backtrack(currentNode,currentPath,problem,Nodes)
        neighbors=expand(currentNode,problem,Nodes)

    ###If we've backtracked to the top, end the search
    if currentNode==None:
        break

    ###Save the current node and look for next step in path
    previousNode=currentNode
    ###Add the next neighbor to try
    for vertex in neighbors: #for each neighboring node
        printBool=(previousNode.state() in vertex.seenBy)
        if not vertex.state() in currentPath and not (previousNode.state() in vertex.seenBy): #not
already in the current path and not previously tried
            vertex.seenBy.append(previousNode.state())
            currentPath[vertex.state()]=vertex
            currentNode=vertex
            break

    ###If there were no new neighbors to add, backtrack
    if currentNode.state()==previousNode.state():
        currentPath.pop(currentNode.state(),None)
        resetChildren(currentNode,problem,Nodes,currentPath)
        tempList=list(currentPath.items())
        if len(tempList)==0:

```

```
        break
    currentNode=tempList[len(tempList)-1][1]
    [currentNode,currentPath]=backtrack(currentNode,currentPath,problem,Nodes)
    neighbors=expand(currentNode,problem,Nodes)
    ###If we've backtracked to the top, end the search
    if currentNode==None:
        break

return Nodes,points;
```

***APPENDIX D***

Code for running optimization and homogenization including MATLAB, python, and bash files for custom functions called in the optimization code

```

from datetime import datetime as dt
import sqlite3 as sq1
import os
import scipy.io as sio
import numpy as np
from sklearn.utils import shuffle
from sklearn.preprocessing import MinMaxScaler
from sklearn.svm import SVR
import random
import matplotlib.pyplot as plt
import imageio
import copy
import math
from scipy.optimize import minimize
from scipy.optimize import Bounds
import pickle
from numpy.linalg import norm
from scipy.stats import qmc
import subprocess
from os.path import exists
from sklearn.multioutput import MultiOutputRegressor

global coeffs
coeffs=[]
global con
global R2s
R2s=[]
global svr
global x1
global x2a
global x2b
global prop2a
global prop2b
global mor

'''
def readInData(dbName, sql):

    #bookkeeping
    date_index=0
    particle_index=1
    volFrac_index=2
    Ez_index=5
    Hy_index=7
    Hz_index=8
    mag_x_index=9
    mag_y_index=10
    mag_z_index=11
    mod_x_index=12
    mod_y_index=13
    mod_z_index=14
    e_rel_x_index=15
    e_rel_y_index=16
    e_rel_z_index=17

    con=sql.connect(dbName)
    cursor=con.cursor()
    with con:
        dataPoints=con.execute(sql).fetchall()

    modelInputs=np.zeros((len(dataPoints),4))
    modelOutputs=np.zeros((len(dataPoints),3))
    dateStamps=[]

    magCounter=0
    for i in range(0,len(dataPoints)):
        data=dataPoints[i]
        if not data[mag_x_index]==None:
            dateStamps.append([data[date_index]]*3)
            modelInputs[magCounter,:]=np.array([data[volFrac_index]/100.0,data[Ez_index]/100000000.0,data[Hy_index]/250000.0,data[Hz_index]/250000.0])
            modelOutputs[magCounter,:]=np.array([data[mag_x_index],data[mag_y_index],data[mag_z_index],data[mod_x_index],data[mod_y_index],data[mod_z_index],data[e_rel_x_index],data[e_rel_y_index],data[e_rel_z_index]])
            magCounter=magCounter+1

    scaledModelInputs=np.stack(modelInputs, axis=0)
    modelOutputs=np.stack(modelOutputs, axis=0)

    #sort data
    # modelOutputs=sorting_of_element(modelOutputs.tolist(),dateStamps)
    # modelInputs=sorting_of_element(modelInputs.tolist(),dateStamps)
    zipped_lists = zip(dateStamps,scaledModelInputs[:,0],scaledModelInputs[:,1],scaledModelInputs[:,2],scaledModelInputs[:,3],modelOutputs[:,0],modelOutputs[:,1],modelOutputs[:,2],modelOutputs[:,3],modelOutputs[:,4],modelOutputs[:,5],modelOutputs[:,6],modelOutputs[:,7],modelOutputs[:,8])
    sorted_pairs = sorted(zipped_lists)
    tuples = zip(*sorted_pairs)
    dateStamps, modelInputs[:,0], modelInputs[:,1],modelInputs[:,2],modelInputs[:,3], modelOutputs[:,0], modelOutputs[:,1],modelOutputs[:,2],modelOutputs[:,3],modelOutputs[:,4],modelOutputs[:,5],modelOutputs[:,6],modelOutputs[:,7],modelOutputs[:,8] = [ list(tuple) for tuple in tuples]

    #scale transform data
    # scaler = MinMaxScaler(feature_range=(0,1))
    # scaledModelInputs = scaler.fit_transform(modelInputs)
    # scaledModelOutputs = scaler.fit_transform(modelOutputs)

    scaledModelInputs = modelInputs
    scaledModelOutputs = modelOutputs

    for i in range(len(scaledModelOutputs)):
        for j in range(len(scaledModelOutputs[i,:])):
            scaledModelOutputs[i,j]=round(scaledModelOutputs[i,j],4)
    return scaledModelInputs, scaledModelOutputs

'''
def makeSVRplot(X,y):
    global pics
    global coeffs
    global mor
    svr_rbf = SVR(kernel="rbf", C=1000, gamma='auto', epsilon=0.0001)
    svr = svr_rbf
    mor=MultiOutputRegressor(svr)
    #print(X)
    #print(y)
    model=mor.fit(X, y)
    R2=mor.score(X,y)

    #convergence criteria
    # print("svr_dual_coef_")
    # print(svr_dual_coef_)
    #temp=list(svr_dual_coef_[0])
    #if len(temp)==0:
    #    temp=0
    #    print(temp)
    #coeffs.append(temp)

    print("R2: ", R2)
    return R2

def checkConvergence(tolerance,convergeCount):
    global R2s
    flatCount=0
    if len(R2s)>0 and R2s[len(R2s)-1]==0.9:
        for i in range(1,len(R2s)):
            #if the length of coeffs increased reset count
            slope=R2s[i]-R2s[i-1] #denominator=1
            if slope<tolerance:
                flatCount=flatCount+1
                if flatCount==convergeCount:
                    return True
            else:
                flatCount=0

    return False

def plotCoeffs():
    if len(coeffs)>0:
        currentC=coeffs[len(coeffs)-1]
        if isinstance(currentC, list):
            print(len(currentC))
            # currentC=currentC[0]
            axes.scatter(np.ones(len(currentC))*1,currentC,facecolor="c")
        else:
            axes.scatter(1,currentC,facecolor="m")

#OPTIMIZATION
def fun1(x):
    global x1
    global x2a
    global x2b
    global prop2a
    global prop2b
    weights=[1/3,1/3,1/3]
    x1=x
    f2Bounds = Bounds([0, 0, 0], [1, 1, 1])
    x0a = x2a
    x0b = x2b
    res_max_mag = minimize(mag_max, x0a[0], jac='3point', method='SLSQP', bounds=f2Bounds, options={'eps':[1000/100000000,1000/250000,1000/250000],'ftol':.1})
    res_max_mod = minimize(mod_max, x0a[1], jac='3point', method='SLSQP', bounds=f2Bounds, options={'eps':[1000/100000000,1000/250000,1000/250000],'ftol':.1})
    res_max_ere1 = minimize(ere1_max, x0a[2], jac='3point', method='SLSQP', bounds=f2Bounds, options={'eps':[1000/100000000,1000/250000,1000/250000],'ftol':.1})
    res_min_mag = minimize(mag_min, x0b[0], jac='3point', method='SLSQP', bounds=f2Bounds, options={'eps':[1000/100000000,1000/250000,1000/250000],'ftol':.1})
    res_min_mod = minimize(mod_min, x0b[1], jac='3point', method='SLSQP', bounds=f2Bounds, options={'eps':[1000/100000000,1000/250000,1000/250000],'ftol':.1})
    res_min_ere1 = minimize(ere1_min, x0b[2], jac='3point', method='SLSQP', bounds=f2Bounds, options={'eps':[1000/100000000,1000/250000,1000/250000],'ftol':.1})
    x2a=[res_max_mag.x,res_max_mod.x,res_max_ere1.x]
    x2b=[res_min_mag.x,res_min_mod.x,res_min_ere1.x]

    mag_range_norm=(res_max_mag.fun-res_min_mag.fun)/res_max_mag.fun
    mod_range_norm=(res_max_mod.fun-res_min_mod.fun)/res_max_mod.fun
    ere1_range_norm=(res_max_ere1.fun-res_min_ere1.fun)/res_max_ere1.fun

    prop2a=[res_max_mag,res_max_mod,res_max_ere1]
    prop2b=[res_min_mag,res_min_mod,res_min_ere1]

    #weighted multi-objective function
    objectiveFunctionWeights[0]*mag_range_norm+weights[1]*mod_range_norm+weights[2]*ere1_range_norm
    #print(objectiveFunction)
    return -objectiveFunction

def mag_max(x2):
    global x1
    global mor
    inputs=[[x1[0],x2[0],x2[1],x2[2]]]

```

```

# print(inputs)
properties-mor.predict(inputs)
#indexing below doesn't return upper bound, exclusive bounds
mags-properties[0,0:3]
mods-properties[0,3:6]
erels-properties[0,6:9]
max_val=(max(mags)/25)+(max(mods)/2.1e6)+(max(erels)/2.67)
return -max(mags)

def mag_min(x2):
    global x1
    global mor
    inputs=[[x1[0],x2[0],x2[1],x2[2]]]
    # print(inputs)
    properties-mor.predict(inputs)
    #indexing below doesn't return upper bound, exclusive bounds
    mags-properties[0,0:3]
    mods-properties[0,3:6]
    erels-properties[0,6:9]
    min_val=(min(mags)/25)+(min(mods)/2.1e6)+(min(erels)/2.67)
    return min(mags)

def mod_max(x2):
    global x1
    global mor
    inputs=[[x1[0],x2[0],x2[1],x2[2]]]
    # print(inputs)
    properties-mor.predict(inputs)
    #indexing below doesn't return upper bound, exclusive bounds
    mags-properties[0,0:3]
    mods-properties[0,3:6]
    erels-properties[0,6:9]
    min_val=(min(mags)/25)+(min(mods)/2.1e6)+(min(erels)/2.67)
    return -max(mods)

def mod_min(x2):
    global x1
    global mor
    inputs=[[x1[0],x2[0],x2[1],x2[2]]]
    # print(inputs)
    properties-mor.predict(inputs)
    #indexing below doesn't return upper bound, exclusive bounds
    mags-properties[0,0:3]
    mods-properties[0,3:6]
    erels-properties[0,6:9]
    min_val=(min(mags)/25)+(min(mods)/2.1e6)+(min(erels)/2.67)
    return min(mods)

def erel_max(x2):
    global x1
    global mor
    inputs=[[x1[0],x2[0],x2[1],x2[2]]]
    # print(inputs)
    properties-mor.predict(inputs)
    #indexing below doesn't return upper bound, exclusive bounds
    mags-properties[0,0:3]
    mods-properties[0,3:6]
    erels-properties[0,6:9]
    min_val=(min(mags)/25)+(min(mods)/2.1e6)+(min(erels)/2.67)
    return -max(erels)

def erel_min(x2):
    global x1
    global mor
    inputs=[[x1[0],x2[0],x2[1],x2[2]]]
    # print(inputs)
    properties-mor.predict(inputs)
    #indexing below doesn't return upper bound, exclusive bounds
    mags-properties[0,0:3]
    mods-properties[0,3:6]
    erels-properties[0,6:9]
    min_val=(min(mags)/25)+(min(mods)/2.1e6)+(min(erels)/2.67)
    return min(erels)

def tryAgain(x,resultsFile,propertyToRun):
    print("TRY AGAIN")
    #For bookkeeping and variable clarity the vector x is reassigned to more descriptive variables
    numOParticles=x[0]
    volFrac=round(x[1],2)
    Ex=0
    Ey=0
    Ez=round(x[2],0)
    Hx=0
    Hy=round(x[3],0)
    Hz=round(x[4],0)
    print(propertyToRun)
    if propertyToRun=="magnetic":
        qsub_command="qsub -N job11-{0} -v numOParticles={0},volFrac={1},Ex={2},Ey={3},Ex={4},Hx={5},Hy={6},Hz={7} SWHtryMagAgainJobscript.pbs""".format(numOParticles,volFrac,Ex,Ey,Ez,Hx,Hy,Hz)
    if propertyToRun=="elastic":
        qsub_command="qsub -N job11-{0} -v numOParticles={0},volFrac={1},Ex={2},Ey={3},Ex={4},Hx={5},Hy={6},Hz={7} SWHtryModAgainJobscript.pbs""".format(numOParticles,volFrac,Ex,Ey,Ez,Hx,Hy,Hz)
    if propertyToRun=="dielectric":
        qsub_command="qsub -N job11-{0} -v numOParticles={0},volFrac={1},Ex={2},Ey={3},Ex={4},Hx={5},Hy={6},Hz={7} SWHtryElecAgainJobscript.pbs""".format(numOParticles,volFrac,Ex,Ey,Ez,Hx,Hy,Hz)

    if not qsub_command == None:
        out=subprocess.Popen(qsub_command, shell=True, stdout=subprocess.PIPE)
        jobId=out.communicate()[0]
        jobIdNum=jobID.decode().split(".")
        print(jobIDnum[0])

        #This script checks to see if the consol job is complete
        bashCommand="bash checkJobStatus.sh {0}""".format(jobIDnum[0])
        out=subprocess.Popen(bashCommand, shell=True, stdout=subprocess.PIPE)
        out.communicate()
        return

    print("Something Went Wrong")

def submitAndWait(x):
    #For bookkeeping and variable clarity the vector x is reassigned to more descriptive variables
    numOParticles=x[0]
    volFrac=round(x[1],0)
    Ex=0
    Ey=0
    Ez=round(x[2],0)
    Hx=0
    Hy=round(x[3],0)
    Hz=round(x[4],0)

    #Check Database for prexisting results
    sql="SELECT * FROM SVM_COMBINE_OPTIMIZATION WHERE numOParticles={0} AND ROUND(volFrac,{0}-1) AND ROUND(Ex,{0}-2) AND ROUND(Ey,{0}-3) AND ROUND(Ez,{0}-4) AND ROUND(Hx,{0}-5) AND ROUND(Hy,{0}-6) AND ROUND(Hz,{0}-7)""".format(numOParticles,volFrac,Ex,Ey,Ez,Hx,Hy,Hz)
    #print(sql)
    with con:
        data=con.execute(sql).fetchall()
    if not len(data)==0:
        dateStamp=data[0][0]
        modulus=[data[0][12],1e6,data[0][13],1e6,data[0][14],1e6]
        e_rel=[data[0][15],data[0][16],data[0][17]]
        magnetization=[data[0][9],data[0][10],data[0][11]]
    else:
        #This function submits the server job to run the consol script and returns the effective material properties
        qsub_command="qsub -N job11-{0} -v numOParticles={0},volFrac={1},Ex={2},Ey={3},Ex={4},Hx={5},Hy={6},Hz={7} SWHoptimizeJobscript.pbs""".format(numOParticles,volFrac,Ex,Ey,Ez,Hx,Hy,Hz)
        out=subprocess.Popen(qsub_command, shell=True, stdout=subprocess.PIPE)
        jobId=out.communicate()[0]
        jobIdNum=jobID.decode().split(".")
        print(jobIDnum[0])

        #This script checks to see if the consol job is complete
        bashCommand="bash checkJobStatus.sh {0}""".format(jobIDnum[0])
        out=subprocess.Popen(bashCommand, shell=True, stdout=subprocess.PIPE)
        out.communicate()

        #open results file and read in processed data
        f=open(path+"/svmOptimizationRuns/results.txt","r")
        resultsFile=f.readline().strip()
        dateStamp=resultsFile.split("_")[0]

        #check to make sure the simulation actually ran
        while not exists(path+"/svmOptimizationRuns/"+resultsFile+"/ElasticityMatrix.txt"):
            #submit another of the same job and try again
            magnetization=tryAgain(x,resultsFile,"elastic")

        while not exists(path+"/svmOptimizationRuns/"+resultsFile+"/MagneticBResponse.txt"):
            #submit another of the same job and try again
            magnetization=tryAgain(x,resultsFile,"magnetic")

        while not exists(path+"/svmOptimizationRuns/"+resultsFile+"/ACResponse.txt"):
            #submit another of the same job and try again
            magnetization=tryAgain(x,resultsFile,"dielectric")

        #create .mat file
        qsub_command="qsub MakeMatFileScript.pbs""".format(numOParticles,volFrac,Ex,Ey,Ez,Hx,Hy,Hz)
        out=subprocess.Popen(qsub_command, shell=True, stdout=subprocess.PIPE)
        jobId=out.communicate()[0]
        jobIdNum=jobID.decode().split(".")
        print(jobIDnum[0])

        #This script checks to see if the consol job is complete
        bashCommand="bash checkJobStatus.sh {0}""".format(jobIDnum[0])
        out=subprocess.Popen(bashCommand, shell=True, stdout=subprocess.PIPE)
        out.communicate()

        #read .mat file
        mat=sio.loadmat(path+"/svmOptimizationRuns/"+resultsFile+"/"+"resultsFile"+"-mat")
        #populate variables to use in objective functions
        e_rel=[mat['e_rel'][0,0],mat['e_rel'][0,1],mat['e_rel'][0,2]]
        print(mat['Magnetization'])
        magnetization=[mat['Magnetization'][0,0],mat['Magnetization'][0,1],mat['Magnetization'][0,2]]
        modulus=[mat['Ex'][0,0],1e6],mat['Ey'][0,0],1e6],mat['Ez'][0,0],1e6]
        #Saving run to database
        saveRun(dateStamp,numOParticles,volFrac,0,0,Ez,0,Hx,Hx,magnetization,modulus,e_rel)

    print(dateStamp)

```



```

print("simulation results: ",max(magnetization),", ",max(modulus),", ",max(e_rel))
return dateStamp,magnetization,modulus,e_rel

def saveRun(dateStamp,numberOfParticles,volFrac,Ex,Ey,Ez,Hx,Hy,Hz,magnetization,modulus,e_rel):
    print("saving data")
    sql="INSERT INTO SVM_COMBINED_OPTIMIZATION(date_stamp,numberOfParticles,volFrac,Ex,Ey,Ez,Hx,Hy,Hz,mag_x,mag_y,mag_z,modulus_x,modulus_y,modulus_z,e_rel_x,e_rel_y,e_rel_z)values(?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)"
    data=(dateStamp,numberOfParticles,volFrac,Ex,Ey,Ez,Hx,Hy,Hz,magnetization[0],magnetization[1],magnetization[2],modulus[0]*1e6,modulus[1]*1e6,modulus[2]*1e6,e_rel[0],e_rel[1],e_rel[2])
    with con:
        con.executemany(sql,data)

#sql=""SELECT * FROM SVM_MAG_OPTIMIZATION""
#scaledInputs, scaledOutputs=readInData("COMSQL_RUNS_OPT.db",sql)

fig = plt.figure()
axes = fig.add_subplot()
lw = 2

i=1

#INIT MODEL SOMEHOW
def initModel():
    #Latin Hypercube
    sampler = qmc.LatinHypercube(d=4,seed=42)
    sample = sampler.random(n=10)
    numOfParticles=400
    results=[]

    #reshape for vol frac bounds
    sample[:,0] = sample[:,0]*0.3+0.01
    print("starting sampling")
    for i in range(100):
        print((numOfParticles,sample[i,0]*100,sample[i,1]*100000000,sample[i,2]*250000,sample[i,3]*250000))
        [dateStamp,magnetization,modulus,e_rel]=submitAndWait((numOfParticles,sample[i,0]*100,sample[i,1]*100000000,sample[i,2]*250000,sample[i,3]*250000))
    #Saving run to database
    #saveRun(dateStamp,numOfParticles,sample[i,0]*100,0,0,sample[i,1]*100000000,0,sample[i,2]*250000,sample[i,3]*250000,magnetization,modulus,e_rel)
    results.append(magnetization[0],magnetization[1],magnetization[2],modulus[0],modulus[1],modulus[2],e_rel[0],e_rel[1],e_rel[2])

    #Print(mags)
    results=np.array(results)

    R2=makeSVMPLOT(sample,results)

    return sample, results

def optimizeSet():
    global x1
    global x2a
    global x2b
    global prop2a
    global prop2b
    f1Bounds = Bounds([0.01], [0.3])
    x1=[-1]
    x2a = [[0, 0, 0],[0, 0, 0],[0, 0, 0]]
    x2b = [[0, 0, 1],[0, 0, 1],[0, 0, 1]]
    res = minimize(fun1, x0, jac='3point', method='SLSQP', bounds=f1Bounds, tol=0.01, options={'eps':.2})

    x1=res.x
    print("Volume fraction with largest range: ",x1)
    print("max mag: ", round(x2a[0][0]*100000000,1),round(x2a[0][1]*250000,1),round(x2a[0][2]*250000,1))
    print("max mod: ", round(x2a[1][0]*100000000,1),round(x2a[1][1]*250000,1),round(x2a[1][2]*250000,1))
    print("max ere1: ", round(x2a[2][0]*100000000,1),round(x2a[2][1]*250000,1),round(x2a[2][2]*250000,1))
    print("min mag: ", round(x2b[0][0]*100000000,1),round(x2b[0][1]*250000,1),round(x2b[0][2]*250000,1))
    print("min mod: ", round(x2b[1][0]*100000000,1),round(x2b[1][1]*250000,1),round(x2b[1][2]*250000,1))
    print("min ere1: ", round(x2b[2][0]*100000000,1),round(x2b[2][1]*250000,1),round(x2b[2][2]*250000,1))

def smartSearch(X,y):
    global mor
    line=mor.predict(X)
    spacing=[]
    errors=[]
    searchTerm=np.zeros((len(y),len(y)))
    searchX=[]

    #Checking the error from the regression to the data
    #for each point
    for i in range(len(X)-1):
        error.append(abs(np.subtract(line[i],y[i])))

    #for each point find distance to the neighbor
    for i in range(len(X)-1):
        for j in range(i,len(X)-4,1):
            xDist=np.abs(np.subtract(X[i,:],X[j,:]))
            temp=xDist*(norm(error[i])+norm(error[j]))
            searchTerm[i,j]=temp

    #find the max "error area"
    maxArea=np.amax(searchTerm)
    rowIndex,columnIndex=np.where(searchTerm==maxArea)
    i=rowIndex[0]
    j=columnIndex[0]
    diff=np.subtract(X[i],X[j])/2
    nextX=np.add(X[j],diff)

    #Get the next biggest "error area" if this point has already been added to the SVR
    while nextX in X:
        searchTerm[i,j]=0
        maxArea=np.amax(searchTerm)
        rowIndex,columnIndex=np.where(searchTerm==maxArea)
        i=rowIndex[0]
        j=columnIndex[0]
        diff=np.subtract(X[i],X[j])/2
        nextX=np.add(X[j],diff)

    print("my two points are:")
    print(X[i],X[j])

    print("my next point is:")
    print(nextX)

    #Run simulation
    simSize=100
    simArray=[simSize,nextX[0]*100,nextX[1]*100000000,nextX[2]*250000,nextX[3]*250000]
    print(simArray)
    [dateStamp,magnetization,modulus,e_rel]=submitAndWait(simArray)

    results=np.zeros(9)
    results[0:3]=magnetization
    results[3:6]=modulus
    results[6:9]=e_rel

    #add new point to set
    X=np.append(X,[nextX],axis=0)
    y=np.append(y,[results],axis=0)

    return X, y

def checkOptimalPoint(acceptableError,scaledInputs,scaledOutputs,x1,x2,propString):
    global mor
    print("acceptableError: ",acceptableError)
    #Run simulation
    simSize=100
    simArray=[simSize,x1[0]*100,x2[0]*100000000,x2[1]*250000,x2[2]*250000]
    print(simArray)
    [dateStamp,magnetization,modulus,e_rel]=submitAndWait(simArray)
    simulationResult=np.zeros(9)
    simulationResult[0:3]=magnetization
    simulationResult[3:6]=modulus
    simulationResult[6:9]=e_rel

    #predict values from SVR
    inputs=[round(x1[0],2),round(x2[0]*100000000,0)/100000000,round(x2[1]*250000,0)/250000,round(x2[2]*250000,0)/250000]
    modelResult=mor.predict(inputs)

    #calculate error between simulation and SVR
    tempError=abs(simulationResult-modelResult)
    if propString=="mag":
        error=tempError[0][0:3]/30
    elif propString=="mod":
        error=tempError[0][3:6]/3
    else:
        error=tempError[0][6:9]/32

    print("sim results: ", simulationResult)
    print("svr results: ", modelResult)
    print("error: ",error)

    #determine if this error is acceptable
    if error[0]<acceptableError and error[5]<acceptableError and error[2]<acceptableError:
        print("matches, done!")
        return True, scaledInputs, scaledOutputs

    #error wasn't ok
    #add this point to the model and check error again
    results=[simulationResult]
    scaledInputs=np.append(scaledInputs,inputs,axis=0)
    scaledOutputs=np.append(scaledOutputs,results,axis=0)
    model=mor.fit(scaledInputs, scaledOutputs)
    modelResult=mor.predict(inputs)

    tempError=abs(simulationResult-modelResult)
    if propString=="mag":
        error=tempError[0][0:3]/30
    elif propString=="mod":
        error=tempError[0][3:6]/3
    else:
        error=tempError[0][6:9]/32

    print("rechecked sim results: ", simulationResult)
    print("rechecked svr results: ", modelResult)
    print("rechecked error: ",error)

```

```

if error[0]<acceptableError and error[1]<acceptableError and error[2]<acceptableError:
    print("Improved enough to optimize again")
    return False, scaledInputs, scaledOutputs
else:
    print("need to train more")
    scaledInputs, scaledOutputs=smartSearch(scaledInputs, scaledOutputs)
    return False, scaledInputs, scaledOutputs

path = os.getcwd()
print(path)
conn=sql.connect("CONSOLE_RUNS_OPT.db")
optimumFound=False
scaledInputs,scaledOutputs=initModel()
#axes.scatter(scaledInputs[:,0],scaledOutputs)
# plt.show()

acceptableError=0.1
while not optimumFound:
    optimize()
    print("Checking max vals found")
    max_found_mag, scaledInputs, scaledOutputs=checkOptimalPoint(acceptableError,scaledInputs,scaledOutputs,x1,x2a[0], 'mag')
    max_found_mod, scaledInputs, scaledOutputs=checkOptimalPoint(acceptableError,scaledInputs,scaledOutputs,x1,x2a[1], 'mod')
    max_found_ere1, scaledInputs, scaledOutputs=checkOptimalPoint(acceptableError,scaledInputs,scaledOutputs,x1,x2a[2], 'ere1')
    print("Checking min vals found")
    min_found_mag, scaledInputs, scaledOutputs=checkOptimalPoint(acceptableError,scaledInputs,scaledOutputs,x1,x2b[0], 'mag')
    min_found_mod, scaledInputs, scaledOutputs=checkOptimalPoint(acceptableError,scaledInputs,scaledOutputs,x1,x2b[1], 'mod')
    min_found_ere1, scaledInputs, scaledOutputs=checkOptimalPoint(acceptableError,scaledInputs,scaledOutputs,x1,x2b[2], 'ere1')
    if max_found_mag and max_found_mod and max_found_ere1 and min_found_mag and min_found_mod and min_found_ere1:
        optimumFound=True
    print("Doing another round of optimization")

print(scaledInputs[:,0])
axes.scatter(scaledInputs[:,0],scaledOutputs)

print("solution found in: ", len(scaledOutputs), "points")
# plt.show()

```



```

public static Model generateEllipsoid(Model model, String[] posArray, String[] dirArray){
    System.out.println("generating ellipsoid...");
    numOfParticle = posArray[1].length/2;
    String[] selectDirArray = new String[numOfParticle];
    for (int particle = 0; particle<numOfParticle; particle++){
        String particleString = "c1p" + particle;
        String refString = "Rytic" + particle;
        int particleIndex = particle*2;
        String positionString = posArray[0][particleIndex] + ", " + posArray[1][particleIndex] + ", " + posArray[2][particleIndex];
        String orientationString = dirArray[0][particleIndex] + ", " + dirArray[1][particleIndex] + ", " + dirArray[2][particleIndex];
        //creates ellipsoid at each coordinate
        model.component("comp1").geom("geom1").create(particleString, "Ellipsoid");
        model.component("comp1").geom("geom1").feature(particleString).set("pos", positionString);
        model.component("comp1").geom("geom1").feature(particleString).set("semiaxes", new double[]{0.5e-6, 0.5e-6, 0.15e-6});
        model.component("comp1").geom("geom1").feature(particleString).set("axisType", "cartesian");
        model.component("comp1").geom("geom1").feature(particleString).set("rot", orientationString);
    }
    selectionArray[particle] = particleString;

    model.component("comp1").geom("geom1").create("sel1", "ExplicitSelection");
    model.component("comp1").geom("geom1").feature("sel1").selection("selection").init();
    model.component("comp1").geom("geom1").feature("sel1").selection("selection").set(selectionArray);
    model.component("comp1").geom("geom1").feature("sel1").set("selshow", "all");

    model.component("comp1").material("mat1").selection().named("geom1_sel1_dom");

    //generate geometry union and run
    model.component("comp1").geom("geom1").create("un1", "Union");
    //model.component("comp1").geom("geom1").feature("un1").set("intend", false);
    //model.component("comp1").geom("geom1").feature("un1").selection("input").named("sel1");

    //model.component("comp1").geom("geom1").create("un1", "Union");
    //model.component("comp1").geom("geom1").feature("un1").set("intend", false);
    //model.component("comp1").geom("geom1").feature("un1").set("repairtoltype", "auto");
    //model.component("comp1").geom("geom1").feature("un1").set("absrepairtol", 1.0E-8);
    //model.component("comp1").geom("geom1").feature("un1").selection("input").named("sel1");
    model.component("comp1").geom("geom1").run();

    saveModel(model, saveFileName);
    return model;
}

public static Model meshModel(Model model){
    System.out.println("meshing model...");
    model.component("comp1").mesh().create("mesh1");
    model.component("comp1").mesh("mesh1").create("fret1", "Freelet");
    model.component("comp1").mesh("mesh1").create("fret2", "Freelet");
    model.component("comp1").mesh("mesh1").feature("fret1").create("s1el", "Size");
    model.component("comp1").mesh("mesh1").feature("fret2").create("s1el", "Size");

    model.component("comp1").mesh("mesh1").feature("fret1").selection(1).named("geom1_sel1_dom");
    model.component("comp1").mesh("mesh1").feature("fret1").feature("s1el").set("custom", "on");
    model.component("comp1").mesh("mesh1").feature("fret1").feature("s1el").set("min", 1.22E-9);
    model.component("comp1").mesh("mesh1").feature("fret1").feature("s1el").set("inactive", true);
    model.component("comp1").mesh("mesh1").feature("fret2").feature("s1el").set("custom", "on");
    model.component("comp1").mesh("mesh1").feature("fret2").feature("s1el").set("min", 1.22E-9);
    model.component("comp1").mesh("mesh1").feature("fret2").feature("s1el").set("inactive", true);
    model.component("comp1").mesh("mesh1").feature("fret2").feature("s1el").selection().geom("geom1", 3);
    model.component("comp1").mesh("mesh1").feature("fret2").feature("s1el").selection().set();

    System.out.println("periodic save...");
    saveModel(model, saveFileName);
    return model;
}

public static Model solveModel(Model model){
    System.out.println("solving model...");
    //model.study().create("cpstd1");
    model.study("cpstd1").create("cpstat1", "Stationary");
    model.sol().create("cpso1");
    model.sol("cpso1").study("cpstd1");
    model.sol("cpso1").attach("cpstd1");
    model.sol("cpso1").create("st1", "StudyStep");
    model.sol("cpso1").create("vt", "Variables");
    model.sol("cpso1").create("st1", "Stationary");
    model.sol("cpso1").feature("st1").create("pt", "Parametric");
    model.sol("cpso1").feature("st1").create("fc1", "FullyCoupled");
    model.sol("cpso1").feature("st1").feature().remove("fdef");
    //
    model.study().create("cpstd1");
    model.study("cpstd1").create("cpstat1", "Stationary");
    model.study("cpstd1").feature("cpstat1").set("useLoadcase", true);
    model.study("cpstd1").feature("cpstat1").setIndex("loadcase", "load case 1", 0);
    model.study("cpstd1").feature("cpstat1").setIndex("loadgroupweight", "1.0", 0, 0);
    model.study("cpstd1").feature("cpstat1").setIndex("loadgroup", false, 0, 3);
    model.study("cpstd1").feature("cpstat1").setIndex("loadgroupweight", "1.0", 0, 3);
    model.study("cpstd1").feature("cpstat1").setIndex("loadgroup", false, 0, 4);
    model.study("cpstd1").feature("cpstat1").setIndex("loadgroupweight", "1.0", 0, 2);
    model.study("cpstd1").feature("cpstat1").setIndex("loadgroup", false, 0, 3);
    model.study("cpstd1").feature("cpstat1").setIndex("loadgroupweight", "1.0", 0, 3);
    model.study("cpstd1").feature("cpstat1").setIndex("loadgroup", false, 0, 3);
    model.study("cpstd1").feature("cpstat1").setIndex("loadgroupweight", "1.0", 0, 5);
    model.study("cpstd1").feature("cpstat1").setIndex("loadgroup", true, 0, 0);
    model.study("cpstd1").feature("cpstat1").setIndex("loadcase", "load case 2", 3);
    model.study("cpstd1").feature("cpstat1").setIndex("loadgroup", false, 1, 0);
    model.study("cpstd1").feature("cpstat1").setIndex("loadgroupweight", "1.0", 1, 1);
    model.study("cpstd1").feature("cpstat1").setIndex("loadgroup", false, 1, 3);
    model.study("cpstd1").feature("cpstat1").setIndex("loadgroupweight", "1.0", 1, 2);
    model.study("cpstd1").feature("cpstat1").setIndex("loadgroup", false, 1, 3);
    model.study("cpstd1").feature("cpstat1").setIndex("loadgroupweight", "1.0", 1, 3);
    model.study("cpstd1").feature("cpstat1").setIndex("loadgroup", false, 1, 4);
    model.study("cpstd1").feature("cpstat1").setIndex("loadgroupweight", "1.0", 1, 5);
    model.study("cpstd1").feature("cpstat1").setIndex("loadgroup", true, 1, 3);
    model.study("cpstd1").feature("cpstat1").setIndex("loadgroup", false, 2, 0);
    model.study("cpstd1").feature("cpstat1").setIndex("loadgroupweight", "1.0", 2, 0);
    model.study("cpstd1").feature("cpstat1").setIndex("loadgroup", false, 2, 3);
    model.study("cpstd1").feature("cpstat1").setIndex("loadgroupweight", "1.0", 2, 3);
    model.study("cpstd1").feature("cpstat1").setIndex("loadgroup", false, 2, 3);
    model.study("cpstd1").feature("cpstat1").setIndex("loadgroupweight", "1.0", 2, 2);
    model.study("cpstd1").feature("cpstat1").setIndex("loadgroup", false, 2, 4);
    model.study("cpstd1").feature("cpstat1").setIndex("loadgroupweight", "1.0", 2, 5);
    model.study("cpstd1").feature("cpstat1").setIndex("loadgroup", true, 2, 2);
    model.study("cpstd1").feature("cpstat1").setIndex("loadcase", "load case 4", 3);
    model.study("cpstd1").feature("cpstat1").setIndex("loadgroup", false, 3, 0);
    model.study("cpstd1").feature("cpstat1").setIndex("loadgroupweight", "1.0", 3, 0);
    model.study("cpstd1").feature("cpstat1").setIndex("loadgroup", false, 3, 3);
    model.study("cpstd1").feature("cpstat1").setIndex("loadgroupweight", "1.0", 3, 3);
    model.study("cpstd1").feature("cpstat1").setIndex("loadgroup", false, 3, 4);
    model.study("cpstd1").feature("cpstat1").setIndex("loadgroupweight", "1.0", 3, 5);
    model.study("cpstd1").feature("cpstat1").setIndex("loadgroup", true, 3, 3);
    model.study("cpstd1").feature("cpstat1").setIndex("loadcase", "load case 5", 4);
    model.study("cpstd1").feature("cpstat1").setIndex("loadgroup", false, 4, 0);
    model.study("cpstd1").feature("cpstat1").setIndex("loadgroupweight", "1.0", 4, 0);
    model.study("cpstd1").feature("cpstat1").setIndex("loadgroup", false, 4, 3);
    model.study("cpstd1").feature("cpstat1").setIndex("loadgroupweight", "1.0", 4, 1);
    model.study("cpstd1").feature("cpstat1").setIndex("loadgroup", false, 4, 2);
    model.study("cpstd1").feature("cpstat1").setIndex("loadgroupweight", "1.0", 4, 2);
    model.study("cpstd1").feature("cpstat1").setIndex("loadgroup", false, 4, 3);
    model.study("cpstd1").feature("cpstat1").setIndex("loadgroupweight", "1.0", 4, 4);
    model.study("cpstd1").feature("cpstat1").setIndex("loadgroup", false, 4, 5);
    model.study("cpstd1").feature("cpstat1").setIndex("loadgroupweight", "1.0", 4, 5);
    model.study("cpstd1").feature("cpstat1").setIndex("loadgroup", true, 4, 0);
    model.study("cpstd1").feature("cpstat1").setIndex("loadcase", "load case 6", 3);
    model.study("cpstd1").feature("cpstat1").setIndex("loadgroup", false, 5, 0);
    model.study("cpstd1").feature("cpstat1").setIndex("loadgroupweight", "1.0", 5, 0);
    model.study("cpstd1").feature("cpstat1").setIndex("loadgroup", false, 5, 3);
    model.study("cpstd1").feature("cpstat1").setIndex("loadgroupweight", "1.0", 5, 3);
    model.study("cpstd1").feature("cpstat1").setIndex("loadgroup", false, 5, 4);
    model.study("cpstd1").feature("cpstat1").setIndex("loadgroupweight", "1.0", 5, 5);
    model.study("cpstd1").feature("cpstat1").setIndex("loadgroup", true, 5, 5);

    model.sol().create("cpso1");
    model.sol("cpso1").study("cpstd1");
    model.sol("cpso1").attach("cpstd1");
    model.sol("cpso1").create("st1", "StudyStep");
    model.sol("cpso1").create("vt", "Variables");
    model.sol("cpso1").create("st1", "Stationary");
    model.sol("cpso1").feature("st1").create("pt", "Parametric");
    model.sol("cpso1").feature("st1").create("fc1", "FullyCoupled");
    model.sol("cpso1").feature("st1").feature().remove("fdef");

    model.sol("cpso1").feature("st1").feature("ddef").set("linewave", "mmps");
    model.sol("cpso1").feature("st1").feature("ddef").set("occ", true);
    model.study("cpstd1").feature("cpstat1").set("pdistrib", true);
    model.sol("cpso1").feature("st1").feature("st1").set("pdistrib", true);

    model.sol("cpso1").feature("st1").feature("ddef").set("convinfo", "detailed");

    System.out.println("periodic save...");
    saveModel(model, saveFileName);
    model.component("comp1").mesh("mesh1").run();
    System.out.println("periodic save post meshing...");

    //Reset kernel to decrease memory usage
    //ModelUtil.setDefaultGeometryKernel("cnsol");
    model.study("cpstd1").run();
    saveModel(model, saveFileName);
    return model;
}

public static void exportResults(Model model){
    model.result().table().create("tbl1", "table");
    model.result().numerical().create("gmw1", "EzGlobalMatrix");
    model.result().numerical("gmw1").set("probtag", "none");
    model.result().export().create("tbl1", "table");
    model.result().numerical("gmw1").set("table", "tbl1");
    model.result().numerical("gmw1").set("exp", "solid.q1.D");
    model.result().numerical("gmw1").set("unit", "Pa");
}

```

```

model.result().numerical("gmev1").set("descr", "Elasticity matrix");
model.result().numerical("gmev1").set("cpnt", new String[]{"solid.refpntx", "0", "Reference point for moment computation, x coordinate"}, {"solid.refpnty", "0", "Reference point for moment computation, y coordinate"}, {"solid.refpntz", "0", "Reference point for moment computation, z coordinate"}));
model.result().numerical("gmev1").set("cpnt", new String[]{"solid.refpntx", "0", "Reference point for moment computation, x coordinate"}, {"solid.refpnty", "0", "Reference point for moment computation, y coordinate"}, {"solid.refpntz", "0", "Reference point for moment computation, z coordinate"}));
model.result().numerical("gmev1").set("dataseries", "average");
model.result().numerical("gmev1").run();
model.result().numerical("gmev1").setResult();
model.result().export("tbl1").set("table", "tbl1");
model.result().export("tbl1").set("filename", "ElasticityMatrix.txt");

saveMyModel(model, saveFileName);

model.result().export("tbl1").run();
}

public static void saveMyModel(Model model, String saveFileName){
    try{
        model.save(saveFileName);
    }
    catch(Exception e){}
}

public static String readInputArgs(){
    String fileName=null;
    try{
        BufferedReader argsFile = new BufferedReader(new FileReader("inputArgs.txt")); //opens file containing what should be commandline args
        fileName = argsFile.readLine();
    }catch(Exception e){}
    return fileName;
}
}

```

```

/* This script generates an FEA model for calculating effective medium properties for a particle composite
 * It imports results from a particle dynamics simulation from text files and generates the resulting geometry
 * Boundary conditions as appropriate load cases are generated
 * Results are exported to text files in the model folder for selected properties
 *
 * Denise Widdowson
 * MACS Lab
 * The Pennsylvania State University
 */

import com.comsol.model.*;
import com.comsol.model.util.*;
import java.io.*;
import java.io.BufferedReader;
import java.util.Arrays;
import java.math.BigDecimal;

public class dielectricACEffectiveMediumModel{
    public static String saveFileName;
    public static int numOfParticles=10;
    //public static String additionalFolder = "Ex=0,Ey=0,Ez=100000000,Hx=50000,Hy=50000,Hz=50000";

    public static void main(String[] args)throws IOException{
        //ModelUtil.initStandalone(true);
        saveFileName=readInputArgs(); //first argument is the filename
        //saveFileName = additionalFolder + "/" + saveFileName;
        System.out.println(saveFileName);
        Model model = initModel(saveFileName); //initialize model
        String[][] dirArray = new String[3][];
        String[][] posArray = new String[3][];
        String[] initArray = new String[3];
        System.out.println("setup materials...");
        model = setupMaterials(model); //define material properties for particles and medium

        //read in data from particle simulation results
        try{
            System.out.printf("importing results...\n");
            System.out.println(saveFileName);

            //positions
            BufferedReader respos = new BufferedReader(new FileReader(saveFileName + "_resPos.txt")); //opens result position file to read in particle positions
            String tempLine = null;
            for(int i=0; i<3; i++){ //only need 3 loops, file has 3 line x, y, and z positions
                if((tempLine= respos.readLine()) != null){
                    posArray[i] = tempLine.trim().split("\\s+"); //splits line from file using spaces as delimiters and creates an array of strings of numbers in scientific notation
                    //System.out.println(Arrays.toString(posArray[i])); //use for debugging
                }
            }

            //directions
            BufferedReader resdir = new BufferedReader(new FileReader(saveFileName + "_resDir.txt")); //opens result direction file to read in particle orientations
            for(int i=0; i<3; i++){ //only need 3 loops, file has 3 line x, y, and z angles
                if((tempLine= resdir.readLine()) != null){
                    dirArray[i] = tempLine.trim().split("\\s+"); //splits line from file using spaces as delimiters and creates an array of strings of numbers in scientific notation
                    //System.out.println(Arrays.toString(dirArray[i])); //use for debugging
                }
            }

            //initilazion data
            BufferedReader initData = new BufferedReader(new FileReader(saveFileName + "_init_data.txt")); //opens result direction file to read in particle orientations
            if((tempLine= initData.readLine()) != null){
                initArray = tempLine.trim().split("\\s+"); //splits line from file using spaces as delimiters and creates an array of strings of numbers in scientific notation
                System.out.println("init array");
                System.out.println(Arrays.toString(initArray)); //use for debugging
            }
        }
        catch(Exception e){ //necessary for java io operations

            //get in medium dimensions
            System.out.println("transforming data...");
            double tempVal = new BigDecimal(initArray[2]).doubleValue();
            System.out.println("transformed big decimal");
            double boundaryLength = (tempVal+0.7e-6)*2;
            double mediumCenter = -boundaryLength/2;
            double[] posCoord = {mediumCenter, mediumCenter, mediumCenter}; //spatial coordinate for generating RVE bounds
            double[] blockSize = {boundaryLength, boundaryLength, boundaryLength}; //size of RVE in 3 dimensions
            model = generateMedium(model, blockSize, posCoord); //generate the medium and assign material
            System.out.println("boundary conditions...");
            model = boundaryConditionsDielectric(model); //assign boundary conditions (do before generating particles because face numbers change)
            model = generateEllipsoids(model, posArray, dirArray); //generate the particles and assign material
            System.out.println("meshing...");
            model = meshModel(model); //generate the mesh
            System.out.println("solving...");
            model = solveModel(model); //solve the model
            System.out.println("exporting results...");
            exportResults(model); //export effective material properties

            System.exit(0);
        }

        public static Model initModel(String saveFileName){
            Model model = ModelUtil.create("Model");
            //model.modelPath("/storage/work/d/dqw5479/java_scripts/testModel");
            //model.label("testModel.mph");
            model.component().create("comp1", false);
            model.component("comp1").geom().create("geom1", 3);
            //model.component("comp1").geom("geom1").run();
            saveMyModel(model, saveFileName);
            return model;
        }

        public static Model setupMaterials(Model model){
            model.component("comp1").material().create("mat1", "Common");
            model.component("comp1").material().create("mat2", "Common");

            model.component("comp1").material("mat1").label("Barium hexaferrite");
            model.component("comp1").material("mat1").propertyGroup("def")

```

```

        .set("relpermittivity", new String[]{"32", "0", "0", "0", "32", "0", "0", "0", "32"});
model.component("comp1").material("mat1").propertyGroup("def")
    .set("electricconductivity", new String[]{"6e-7", "0", "0", "0", "6e-7", "0", "0", "0", "6e-7"});

model.component("comp1").material("mat2").label("PDMS");
model.component("comp1").material("mat2").propertyGroup("def")
    .set("electricconductivity", new String[]{"1e-14", "0", "0", "0", "1e-14", "0", "0", "0", "1e-14"});
model.component("comp1").material("mat2").propertyGroup("def")
    .set("relpermittivity", new String[]{"2.67", "0", "0", "0", "2.67", "0", "0", "0", "2.67"});

System.out.println("periodic save...");
saveMyModel(model, saveFileName);

return model;
}

public static Model generateMedium(Model model, double[] blockSize, double[] posCoord){
    System.out.println("generating medium...");
    model.component("comp1").geom("geom1").create("blk1", "Block");
    model.component("comp1").geom("geom1").feature("blk1").set("size", blockSize);
    model.component("comp1").geom("geom1").feature("blk1").set("pos", posCoord);
    model.component("comp1").geom("geom1").run();
    model.component("comp1").material("mat2").selection().set(1);
    System.out.println("periodic save...");
    saveMyModel(model, saveFileName);

    return model;
}

public static Model boundaryConditionsDielectric(Model model){
    model.component("comp1").physics().create("ec", "ConductiveMedia", "geom1");

    model.component("comp1").physics("ec").create("pc1", "PeriodicCondition", 2);
    model.component("comp1").physics("ec").feature("pc1").selection().set(1, 6);
    model.component("comp1").physics("ec").feature("pc1").active(false);
    model.component("comp1").physics("ec").create("pc2", "PeriodicCondition", 2);
    model.component("comp1").physics("ec").feature("pc2").selection().set(2, 5);
    model.component("comp1").physics("ec").feature("pc2").active(false);
    model.component("comp1").physics("ec").create("pc3", "PeriodicCondition", 2);
    model.component("comp1").physics("ec").feature("pc3").selection().set(3, 4);
    model.component("comp1").physics("ec").feature("pc3").active(false);

    model.component("comp1").physics("ec").create("term1", "Terminal", 2);
    model.component("comp1").physics("ec").feature("term1").selection().set(6);
    model.component("comp1").physics("ec").feature("term1").set("TerminalType", "Voltage");
    model.component("comp1").physics("ec").feature("term1").set("V0", 1);
    model.component("comp1").physics("ec").feature("term1").active(false);
    model.component("comp1").physics("ec").create("term2", "Terminal", 2);
    model.component("comp1").physics("ec").feature("term2").selection().set(5);
    model.component("comp1").physics("ec").feature("term2").set("TerminalType", "Voltage");
    model.component("comp1").physics("ec").feature("term2").set("V0", 1);
    model.component("comp1").physics("ec").feature("term2").active(false);
    model.component("comp1").physics("ec").create("term3", "Terminal", 2);
    model.component("comp1").physics("ec").feature("term3").selection().set(4);
    model.component("comp1").physics("ec").feature("term3").set("TerminalType", "Voltage");
    model.component("comp1").physics("ec").feature("term3").set("V0", 1);
    model.component("comp1").physics("ec").feature("term3").active(false);

    model.component("comp1").physics("ec").create("gnd1", "Ground", 2);
    model.component("comp1").physics("ec").feature("gnd1").selection().set(1);
    model.component("comp1").physics("ec").feature("gnd1").active(false);
    model.component("comp1").physics("ec").create("gnd2", "Ground", 2);
    model.component("comp1").physics("ec").feature("gnd2").selection().set(2);
    model.component("comp1").physics("ec").feature("gnd2").active(false);
    model.component("comp1").physics("ec").create("gnd3", "Ground", 2);
    model.component("comp1").physics("ec").feature("gnd3").selection().set(3);
    model.component("comp1").physics("ec").feature("gnd3").active(false);

    System.out.println("periodic save...");
    saveMyModel(model, saveFileName);
    return model;
}

public static Model generateEllipsoids(Model model, String[][] posArray, String[][] dirArray){
    System.out.println("generating ellipsoids...");
    numOfParticles = posArray[1].length/2;
    String[] selectionArray = new String[numOfParticles];
    for (int particle = 0; particle < numOfParticles; particle++){
        String particleString = "elp" + particle;
        String mfcString = "Mymfc" + particle;
        int particleIndex = particle + numOfParticles; //first half of indices are for starting position before particle dynamics simulation
        String positionString = posArray[0][particleIndex] + ", " + posArray[1][particleIndex] + ", " + posArray[2][particleIndex];
        String orientationString = dirArray[0][particleIndex] + ", " + dirArray[1][particleIndex] + ", " + dirArray[2][particleIndex];
        //creates ellipsoid at each coordinate
        model.component("comp1").geom("geom1").create(particleString, "Ellipsoid");
        model.component("comp1").geom("geom1").feature(particleString).set("pos", positionString);
        model.component("comp1").geom("geom1").feature(particleString).set("semiaxes", new double[]{0.5e-6, 0.5e-6, 0.15e-6});
        model.component("comp1").geom("geom1").feature(particleString).set("axistype", "cartesian");
        model.component("comp1").geom("geom1").feature(particleString).set("ax3", orientationString);

        selectionArray[particle] = particleString;
    }

    model.component("comp1").geom("geom1").create("sel1", "ExplicitSelection");
    model.component("comp1").geom("geom1").feature("sel1").selection("selection").init();
    model.component("comp1").geom("geom1").feature("sel1").selection("selection").set(selectionArray);
    model.component("comp1").geom("geom1").feature("sel1").set("selshow", "all");

    model.component("comp1").material("mat1").selection().named("geom1_sel1_dom");

    //generate geometry union and run
    //model.component("comp1").geom("geom1").create("un1", "Union");
    //model.component("comp1").geom("geom1").feature("un1").set("intbnd", false);
    //model.component("comp1").geom("geom1").feature("un1").selection("input").named("sel1");

    //model.component("comp1").geom("geom1").create("un1", "Union");
    //model.component("comp1").geom("geom1").feature("un1").set("intbnd", false);
    //model.component("comp1").geom("geom1").feature("un1").set("repairtoltype", "auto");
    //model.component("comp1").geom("geom1").feature("un1").set("absrepairtol", 1.0E-8);

```

```

//model.component("comp1").geom("geom1").feature("uni1").selection("input").named("sel1");
model.component("comp1").geom("geom1").run();

saveMyModel(model, saveFileName);
return model;
}

public static Model meshModel(Model model){
    System.out.println("meshing model...");
    model.component("comp1").mesh().create("mesh1");
    model.component("comp1").mesh("mesh1").create("ftet1", "FreeTet");
    model.component("comp1").mesh("mesh1").create("ftet2", "FreeTet");
    model.component("comp1").mesh("mesh1").feature("ftet1").create("size1", "Size");
    model.component("comp1").mesh("mesh1").feature("ftet2").create("size1", "Size");

    model.component("comp1").mesh("mesh1").feature("ftet1").selection().named("geom1_sel1_dom");
    model.component("comp1").mesh("mesh1").feature("ftet1").feature("size1").set("custom", "on");
    model.component("comp1").mesh("mesh1").feature("ftet1").feature("size1").set("hmin", 1.22E-9);
    model.component("comp1").mesh("mesh1").feature("ftet1").feature("size1").set("hminactive", true);
    model.component("comp1").mesh("mesh1").feature("ftet2").feature("size1").set("custom", "on");
    model.component("comp1").mesh("mesh1").feature("ftet2").feature("size1").set("hmin", 1.22E-9);
    model.component("comp1").mesh("mesh1").feature("ftet2").feature("size1").set("hminactive", true);
    model.component("comp1").mesh("mesh1").feature("ftet2").feature("size1").selection().geom("geom1", 3);
    model.component("comp1").mesh("mesh1").feature("ftet2").feature("size1").selection().set(1);

    System.out.println("periodic save...");
    saveMyModel(model, saveFileName);
    return model;
}

public static Model solveModel(Model model){
    System.out.println("solving model...");

    model.study().create("std1");
    model.study("std1").create("freq", "Frequency");

    model.sol().create("sol1");
    model.sol("sol1").study("std1");
    model.sol("sol1").attach("std1");
    model.sol("sol1").create("st1", "StudyStep");
    model.sol("sol1").create("v1", "Variables");
    model.sol("sol1").create("s1", "Stationary");
    model.sol("sol1").feature("s1").create("p1", "Parametric");
    model.sol("sol1").feature("s1").create("fc1", "FullyCoupled");
    model.sol("sol1").feature("s1").create("i1", "Iterative");
    model.sol("sol1").feature("s1").feature("i1").create("mg1", "Multigrid");
    model.sol("sol1").feature("s1").feature().remove("fcDef");
    model.study("std1").feature("freq").setIndex("plist", "100000", 0);

    model.component("comp1").mesh("mesh1").run();

    for (int i=1; i<=3; i++){
        //reset

        System.out.println("periodic save...");
        System.out.println(i);
        saveMyModel(model, saveFileName);
        switch (i) {
            case 1:
                System.out.println("case 1...");
                model.component("comp1").physics("ec").feature("pc1").active(false);
                model.component("comp1").physics("ec").feature("pc2").active(true);
                model.component("comp1").physics("ec").feature("pc3").active(true);

                model.component("comp1").physics("ec").feature("term3").active(false);
                model.component("comp1").physics("ec").feature("term2").active(false);
                model.component("comp1").physics("ec").feature("term1").active(true);

                model.component("comp1").physics("ec").feature("gnd3").active(false);
                model.component("comp1").physics("ec").feature("gnd2").active(false);
                model.component("comp1").physics("ec").feature("gnd1").active(true);

                model.sol("sol1").runAll();
                model.result().numerical().create("gev1", "EvalGlobal");
                model.result().numerical("gev1").set("probetag", "none");
                model.result().table().create("tbl1", "Table");
                model.result().numerical("gev1").set("table", "tbl1");
                model.result().numerical("gev1")
                    .set("exprn", new String[]{"ec.I0_1"});
                model.result().numerical("gev1").set("unit", new String[]{"A"});
                model.result().numerical("gev1").set("descr", new String[]{"I0"});
                model.result().numerical("gev1").setResult();

                break;

            case 2:
                System.out.println("case 2...");
                model.component("comp1").physics("ec").feature("pc2").active(false);
                model.component("comp1").physics("ec").feature("pc1").active(true);
                model.component("comp1").physics("ec").feature("pc3").active(true);

                model.component("comp1").physics("ec").feature("term1").active(false);
                model.component("comp1").physics("ec").feature("term3").active(false);
                model.component("comp1").physics("ec").feature("term2").active(true);

                model.component("comp1").physics("ec").feature("gnd1").active(false);
                model.component("comp1").physics("ec").feature("gnd3").active(false);
                model.component("comp1").physics("ec").feature("gnd2").active(true);

                model.sol("sol1").runAll();
                model.result().numerical().create("gev2", "EvalGlobal");
                model.result().numerical("gev2").set("probetag", "none");
                model.result().table().create("tbl2", "Table");
                model.result().numerical("gev2").set("table", "tbl2");
                model.result().numerical("gev2")
                    .set("exprn", new String[]{"ec.I0_2"});
                model.result().numerical("gev2").set("unit", new String[]{"A"});
                model.result().numerical("gev2").set("descr", new String[]{"I0"});
                model.result().numerical("gev2").setResult();

```



```

        break;

    case 3:
        System.out.println("case 3...");
        model.component("comp1").physics("ec").feature("pc3").active(false);
        model.component("comp1").physics("ec").feature("pc1").active(true);
        model.component("comp1").physics("ec").feature("pc2").active(true);

        model.component("comp1").physics("ec").feature("term1").active(false);
        model.component("comp1").physics("ec").feature("term2").active(false);
        model.component("comp1").physics("ec").feature("term3").active(true);

        model.component("comp1").physics("ec").feature("gnd1").active(false);
        model.component("comp1").physics("ec").feature("gnd2").active(false);
        model.component("comp1").physics("ec").feature("gnd3").active(true);

        model.sol("sol1").runAll();
        model.result().numerical().create("gev3", "EvalGlobal");
        model.result().numerical("gev3").set("probetag", "none");
        model.result().table().create("tbl3", "Table");
        model.result().numerical("gev3").set("table", "tbl3");
        model.result().numerical("gev3")
            .set("expr", new String[]{"ec.I0_3"});
        model.result().numerical("gev3").set("unit", new String[]{"A"});
        model.result().numerical("gev3").set("descr", new String[]{"I0"});
        model.result().numerical("gev3").setResult();
        break;

    default: break;
}

}

saveMyModel(model, saveFileName);
return model;
}

public static void exportResults(Model model){

    model.result().export().create("tbl1", "Table");
    model.result().export("tbl1").set("table", "tbl1");
    model.result().export("tbl1").set("filename", "ACResponse1.txt");
    model.result().export("tbl1").run();
    model.result().export().create("tbl2", "Table");
    model.result().export("tbl2").set("table", "tbl2");
    model.result().export("tbl2").set("filename", "ACResponse2.txt");
    model.result().export("tbl2").run();
    model.result().export().create("tbl3", "Table");
    model.result().export("tbl3").set("table", "tbl3");
    model.result().export("tbl3").set("filename", "ACResponse3.txt");
    model.result().export("tbl3").run();

    saveMyModel(model, saveFileName);
}

public static void saveMyModel(Model model, String saveFileName){
    try{
        model.save(saveFileName);
    }
    catch(Exception e){}
}

public static String readInputArgs(){
    String fileName=null;
    try{
        BufferedReader argsFile = new BufferedReader(new FileReader("dielectricInputArgs.txt")); //opens file containing what should be commandline args
        fileName = argsFile.readLine();
    }catch(Exception e){}
    return fileName;
}

}

```

```

/* This script generates an FEA model for calculating effective medium properties for a particle composite
 * It imports results from a particle dynamics simulation from text files and generates the resulting geometry
 * Boundary conditions as appropriate load cases are generated
 * Results are exported to text files in the model folder for selected properties
 *
 * Denise Widdowson
 * MACS Lab
 * The Pennsylvania State University
 */

import com.comsol.model.*;
import com.comsol.model.util.*;
import java.io.*;
import java.io.BufferedReader;
import java.util.Arrays;
import java.math.BigDecimal;

public class magneticEffectiveMediumModel{
    public static String saveFileName;
    public static int numOfParticles=10;
    //public static String additionalFolder = "Ex=0,Ey=0,Ez=100000000,Hx=50000,Hy=50000,Hz=50000";

    public static void main(String[] args)throws IOException{
        //ModelUtil.initStandalone(true);
        ModelUtil.setDefaultGeometryKernel("cadps");
        saveFileName=readInputArgs(); //first argument is the filename
        //saveFileName = additionalFolder + "/" + saveFileName;
        System.out.println(saveFileName);
        Model model = initModel(saveFileName); //initialize model
        String[][] dirArray = new String[3][];
        String[][] posArray = new String[3][];
        String[] initArray = new String[3];
        System.out.println("setup materials...");
        model = setupMaterials(model); //define material properties for particles and medium

        //read in data from particle simulation results
        try{
            System.out.printf("importing results...\n");

            //positions
            BufferedReader respos = new BufferedReader(new FileReader(saveFileName + "_resPos.txt")); //opens result position file to read in particle positions
            String tempLine = null;
            for(int i=0; i<3; i++){ //only need 3 loops, file has 3 line x, y, and z positions
                if((tempLine= respos.readLine()) != null){
                    posArray[i] = tempLine.trim().split("\\s+"); //splits line from file using spaces as delimiters and creates an array of strings of numbers in scientific notation
                    //System.out.println(Arrays.toString(posArray[i])); //use for debugging
                }
            }

            //directions
            BufferedReader resdir = new BufferedReader(new FileReader(saveFileName + "_resDir.txt")); //opens result direction file to read in particle orientations
            for(int i=0; i<3; i++){ //only need 3 loops, file has 3 line x, y, and z angles
                if((tempLine= resdir.readLine()) != null){
                    dirArray[i] = tempLine.trim().split("\\s+"); //splits line from file using spaces as delimiters and creates an array of strings of numbers in scientific notation
                    //System.out.println(Arrays.toString(dirArray[i])); //use for debugging
                }
            }

            //initilazion data
            BufferedReader initData = new BufferedReader(new FileReader(saveFileName + "_init_data.txt")); //opens result direction file to read in particle orientations
            if((tempLine= initData.readLine()) != null){
                initArray = tempLine.trim().split("\\s+"); //splits line from file using spaces as delimiters and creates an array of strings of numbers in scientific notation
                System.out.println("init array");
                System.out.println(Arrays.toString(initArray)); //use for debugging
            }
        }
        catch(Exception e){ //necessary for java io operations

            //get in medium dimensions
            System.out.println("transforming data...");
            System.out.println(initArray[2].toString());
            double tempVal = new BigDecimal(initArray[2]).doubleValue();
            System.out.println("transformed big decimal");
            double boundaryLength = (tempVal+0.7e-6)*2;
            double mediumCenter = -boundaryLength/2;
            double[] posCoord = {mediumCenter, mediumCenter, mediumCenter}; //spatial coordinate for generating RVE bounds
            double[] blockSize = {boundaryLength, boundaryLength, boundaryLength}; //size of RVE in 3 dimensions
            double[] airSize = {2*boundaryLength, 2*boundaryLength, 2*boundaryLength}; //size of RVE in 3 dimensions
            model = generateMedium(model, blockSize, posCoord, airSize); //generate the medium and assign material
            System.out.println("boundary conditions...");
            model = boundaryConditionsMech(model); //assign boundary conditions (do before generating particles because face numbers change)
            model = generateEllipsoids(model, posArray, dirArray); //generate the particles and assign material
            System.out.println("meshing...");
            model = meshModel(model); //generate the mesh
            System.out.println("solving...");
            model = solveModel(model); //solve the model
            System.out.println("exporting results...");
            exportResults(model); //export effective material properties

            System.exit(0);
        }

        public static Model initModel(String saveFileName){
            Model model = ModelUtil.create("Model");
            //model.modelPath("/storage/work/d/dqw5479/java_scripts/testModel");
            //model.label("testModel.mph");
            model.component().create("comp1", false);
            model.component("comp1").geom().create("geom1", 3);
            //model.component("comp1").geom("geom1").run();
            saveMyModel(model, saveFileName);
            return model;
        }

        public static Model setupMaterials(Model model){
            model.component("comp1").material().create("mat1", "Common");
            model.component("comp1").material().create("mat2", "Common");
            model.component("comp1").material().create("mat3", "Common");
        }
    }
}

```

```

model.component("comp1").material("mat2").propertyGroup().create("Enu", "Young's modulus and Poisson's ratio");
model.component("comp1").material("mat2").propertyGroup().create("BHCurve", "B-H curve");

model.component("comp1").material("mat1").label("Soft Iron (Without Losses)");
model.component("comp1").material("mat1").set("family", "iron");
model.component("comp1").material("mat1").propertyGroup("def")
    .set("electricconductivity", new String[]{"0[S/m]", "0", "0", "0", "0[S/m]", "0", "0", "0", "0[S/m]"});
model.component("comp1").material("mat1").propertyGroup("def")
    .set("relpermittivity", new String[]{"1", "0", "0", "0", "1", "0", "0", "0", "1"});
model.component("comp1").material("mat1").propertyGroup("def")
    .set("relpermeability", new String[]{"2", "0", "0", "0", "2", "0", "0", "0", "2"});

model.component("comp1").material("mat2").label("PDMS - Polydimethylsiloxane");
model.component("comp1").material("mat2").set("groups", new String[][]{{"polymers", "Polymers"}});
model.component("comp1").material("mat2").propertyGroup("def")
    .set("thermalexpansioncoefficient", new String[]{"9e-4[1/K]", "0", "0", "0", "9e-4[1/K]", "0", "0", "0", "9e-4[1/K]"});
model.component("comp1").material("mat2").propertyGroup("def").set("heatcapacity", "1460[J/(kg*K)");
model.component("comp1").material("mat2").propertyGroup("def")
    .set("relpermittivity", new String[]{"2.75", "0", "0", "0", "2.75", "0", "0", "0", "2.75"});
model.component("comp1").material("mat2").propertyGroup("def").set("density", "970[kg/m^3]");
model.component("comp1").material("mat2").propertyGroup("def")
    .set("thermalconductivity", new String[]{"0.16[W/(m*K)]", "0", "0", "0", "0.16[W/(m*K)]", "0", "0", "0", "0.16[W/(m*K)]"});
model.component("comp1").material("mat2").propertyGroup("def")
    .set("relpermeability", new String[]{"1", "0", "0", "0", "1", "0", "0", "0", "1"});
model.component("comp1").material("mat2").propertyGroup("Enu").set("youngsmodulus", "25[MPa]");
model.component("comp1").material("mat2").propertyGroup("Enu").set("poissonsratio", "0.49");
model.component("comp1").material("mat2").propertyGroup("BHCurve").set("normB", "");
model.component("comp1").material("mat2").propertyGroup("BHCurve").set("normH", "");
model.component("comp1").material("mat2").propertyGroup("BHCurve").set("Wpm", "");
model.component("comp1").material("mat2").propertyGroup("BHCurve").set("normB", "0");
model.component("comp1").material("mat2").propertyGroup("BHCurve").set("normH", "0");
model.component("comp1").material("mat2").propertyGroup("BHCurve").set("Wpm", "0");
model.component("comp1").material("mat2").propertyGroup("BHCurve").addInput("magneticfield");
model.component("comp1").material("mat2").propertyGroup("BHCurve").addInput("magneticfluxdensity");

model.component("comp1").material("mat3").set("family", "air");
model.component("comp1").material("mat3").propertyGroup("def")
    .set("relpermeability", new String[]{"1", "0", "0", "0", "1", "0", "0", "0", "1"});

System.out.println("periodic save...");
saveMyModel(model, saveFileName);

return model;
}

public static Model generateMedium(Model model, double[] blockSize, double[] posCoord, double[] airSize){
    System.out.println("generating medium...");
    model.component("comp1").geom("geom1").create("blk1", "Block");
    model.component("comp1").geom("geom1").feature("blk1").set("size", blockSize);
    model.component("comp1").geom("geom1").feature("blk1").set("pos", posCoord);

    model.component("comp1").geom("geom1").create("blk2", "Block");
    model.component("comp1").geom("geom1").feature("blk2").set("size", airSize);
    model.component("comp1").geom("geom1").feature("blk2").set("base", "center");
    model.component("comp1").geom("geom1").feature("blk2").set("pos", new int[]{0, 0, 0});

    model.component("comp1").geom("geom1").run();
    model.component("comp1").material("mat2").selection().set(1);
    model.component("comp1").material("mat3").selection().set(2);

    System.out.println("periodic save...");
    saveMyModel(model, saveFileName);

    return model;
}

public static Model boundaryConditionsMech(Model model){
    model.param().set("Hx", "0[A/m]", "background applied field");
    model.param().set("Hy", "0[A/m]", "background applied field");
    model.param().set("Hz", "0[A/m]", "background applied field");
    model.param().set("Mc", "-300[A/m]", "background applied field");

    model.component("comp1").physics().create("mfnc", "MagnetostaticsNoCurrents", "geom1");
    /*model.component("comp1").physics("mfnc").create("pc1", "PeriodicCondition", 2);
    model.component("comp1").physics("mfnc").feature("pc1").selection().set(1, 6);
    model.component("comp1").physics("mfnc").create("pc2", "PeriodicCondition", 2);
    model.component("comp1").physics("mfnc").feature("pc2").selection().set(3, 4);
    model.component("comp1").physics("mfnc").create("pc3", "PeriodicCondition", 2);
    model.component("comp1").physics("mfnc").feature("pc3").selection().set(2, 5);
    */

    model.component("comp1").physics("mfnc").create("mflx1", "MagneticFluxDensity", 2);
    model.component("comp1").physics("mfnc").feature("mflx1").selection().set(4);
    model.component("comp1").physics("mfnc").prop("BackgroundField").set("SolveFor", "ReducedField");
    model.component("comp1").physics("mfnc").prop("BackgroundField").set("Hb", new String[][]{{"Hx"}, {"Hy"}, {"Hz"}});
    model.component("comp1").physics("mfnc").feature("mfcl").set("materialType", "solid");
    model.component("comp1").physics("mfnc").feature("mfcl").set("ConstitutiveRelationBH", "RelativePermeability");
    model.component("comp1").physics("mfnc").feature("mflx1").set("Bn", 100);
    model.component("comp1").physics("mfnc").feature("mflx1").active(false);

    System.out.println("periodic save...");
    saveMyModel(model, saveFileName);

    return model;
}

public static Model generateEllipsoids(Model model, String[][] posArray, String[][] dirArray){
    double magnetization=370000;//A/m
    double mu_o=1.25663706e-6;//permeability of free space
    System.out.println("generating ellipsoids...");
    numOffParticles = posArray[1].length/2;
    String[] selectionArray = new String[numOffParticles];

```

```

String[] domainArray = new String[numOfParticles+1];
for (int particle = 0; particle<numOfParticles; particle++){
    String particleString = "elp" + particle;
    int particleIndex = particle+numOfParticles; //first half of indices are for starting position before particle dynamics simulation
    String positionString = posArray[0][particleIndex] + ", " + posArray[1][particleIndex] + ", " + posArray[2][particleIndex];
    String orientationString = dirArray[0][particleIndex] + ", " + dirArray[1][particleIndex] + ", " + dirArray[2][particleIndex];
    //creates ellipsoid at each coordinate
    model.component("comp1").geom("geom1").create(particleString, "Ellipsoid");
    model.component("comp1").geom("geom1").feature(particleString).set("pos", positionString);
    model.component("comp1").geom("geom1").feature(particleString).set("semiaxes", new double[] {0.5e-6, 0.5e-6, 0.15e-6});
    model.component("comp1").geom("geom1").feature(particleString).set("axistype", "cartesian");
    model.component("comp1").geom("geom1").feature(particleString).set("ax3", orientationString);
    model.component("comp1").geom("geom1").feature(particleString).set("selresult", true);

    selectionArray[particle] = particleString;
    domainArray[particle] = particleString;
}
model.component("comp1").geom("geom1").run();

for (int particle = 0; particle<numOfParticles; particle++){
    String particleString = "elp" + particle;
    int particleIndex = particle+numOfParticles; //first half of indices are for starting position before particle dynamics simulation

    model.component("comp1").physics("mfnc").create(particleString, "MagneticFluxConservation", 3);
    model.component("comp1").physics("mfnc").feature(particleString).selection().named("geom1_"+particleString+"_dom");

    model.component("comp1").physics("mfnc").feature(particleString).setIndex("materialType", "solid", 0);
    model.component("comp1").physics("mfnc").feature(particleString).set("ConstitutiveRelationBH", "RemanentFluxDensity");
    String xComp=Double.toString(magnetization*Double.parseDouble(dirArray[0][particleIndex]));
    String xMag="("+xComp+"*Hz"+dirArray[2][particleIndex]+>Mc)-"+xComp+"*(Hz"+dirArray[2][particleIndex]+<Mc))*"+Double.toString(mu_o);
    String yComp=Double.toString(magnetization*Double.parseDouble(dirArray[1][particleIndex]));
    String yMag="("+yComp+"*Hz"+dirArray[2][particleIndex]+>Mc)-"+yComp+"*(Hz"+dirArray[2][particleIndex]+<Mc))*"+Double.toString(mu_o);
    String zComp=Double.toString(magnetization*Double.parseDouble(dirArray[2][particleIndex]));
    String zMag="("+zComp+"*Hz"+dirArray[2][particleIndex]+>Mc)-"+zComp+"*(Hz"+dirArray[2][particleIndex]+<Mc))*"+Double.toString(mu_o);
    model.component("comp1").physics("mfnc").feature(particleString).set("Br", new String[] {xMag,yMag,zMag});

}

domainArray[numOfParticles]="blk1";

model.component("comp1").geom("geom1").create("sel1", "ExplicitSelection");
model.component("comp1").geom("geom1").create("sel2", "ExplicitSelection");

model.component("comp1").geom("geom1").feature("sel2").selection("selection").init();
model.component("comp1").geom("geom1").feature("sel2").selection("selection").set(domainArray);
model.component("comp1").geom("geom1").feature("sel2").set("selshow", "all");

model.component("comp1").geom("geom1").feature("sel1").selection("selection").init();
model.component("comp1").geom("geom1").feature("sel1").selection("selection").set(selectionArray);
model.component("comp1").geom("geom1").feature("sel1").set("selshow", "all");

model.component("comp1").material("mat1").selection().named("geom1_sel1_dom");

model.component("comp1").geom("geom1").run();

saveMyModel(model, saveFileName);
return model;
}

public static Model meshModel(Model model){
    System.out.println("meshing model...");
    model.component("comp1").mesh().create("mesh1");
    model.component("comp1").mesh("mesh1").create("ftet1", "FreeTet");
    model.component("comp1").mesh("mesh1").create("ftet2", "FreeTet");
    model.component("comp1").mesh("mesh1").feature("ftet1").create("size1", "Size");
    model.component("comp1").mesh("mesh1").feature("ftet2").create("size1", "Size");

    model.component("comp1").mesh("mesh1").feature("ftet1").selection().named("geom1_sel1_dom");
    model.component("comp1").mesh("mesh1").feature("ftet1").feature("size1").set("custom", "on");
    model.component("comp1").mesh("mesh1").feature("ftet1").feature("size1").set("hmin", 1.22E-9);
    model.component("comp1").mesh("mesh1").feature("ftet1").feature("size1").set("hmininactive", true);
    model.component("comp1").mesh("mesh1").feature("ftet2").feature("size1").set("custom", "on");
    model.component("comp1").mesh("mesh1").feature("ftet2").feature("size1").set("hmin", 1.22E-9);
    model.component("comp1").mesh("mesh1").feature("ftet2").feature("size1").set("hmininactive", true);
    model.component("comp1").mesh("mesh1").feature("ftet2").feature("size1").selection().geom("geom1", 3);
    model.component("comp1").mesh("mesh1").feature("ftet2").feature("size1").selection().set(1);

    System.out.println("periodic save...");
    saveMyModel(model, saveFileName);
    return model;
}

public static Model solveModel(Model model){
    System.out.println("solving model...");

    model.study().create("std1");
    model.study("std1").create("stat", "Stationary");

    model.sol().create("sol1");
    model.sol("sol1").study("std1");
    model.sol("sol1").attach("std1");
    model.sol("sol1").create("st1", "StudyStep");
    model.sol("sol1").create("v1", "Variables");
    model.sol("sol1").create("s1", "Stationary");
    model.sol("sol1").feature("s1").create("fc1", "FullyCoupled");
    model.sol("sol1").feature("s1").create("i1", "Iterative");
    model.sol("sol1").feature("s1").feature("i1").create("mg1", "Multigrid");
    model.sol("sol1").feature("s1").feature().remove("fcDef");

    model.sol("sol1").attach("std1");
    model.sol("sol1").feature("s1").feature("i1").set("linsolver", "cg");
    model.sol("sol1").feature("s1").feature("i1").feature("mg1").set("prefun", "amg");

    System.out.println("periodic save...");
    saveMyModel(model, saveFileName);
    model.component("comp1").mesh("mesh1").run();
}

```

```

    model.study("std1").run();

    saveMyModel(model, saveFileName);
    return model;
}

public static void exportResults(Model model){
    model.result().table().create("tbl1", "Table");
    model.result().numerical().create("int1", "IntVolume");
    model.result().numerical("int1").selection().named("geom1_sel2_dom");
    model.result().numerical("int1").set("expr", new String[]{"mfnc.Bx"});
    model.result().numerical("int1").set("descr", new String[]{"Magnetic flux density, x component"});
    model.result().numerical("int1").set("unit", new String[]{"kg*m^3/(s^2*A)"});
    model.result().numerical("int1").setIndex("expr", "mfnc.By", 1);
    model.result().numerical("int1").setIndex("expr", "mfnc.Bz", 2);
    model.result().numerical("int1").set("table", "tbl1");
    model.result().numerical("int1").setResult();

    model.result().export().create("tbl1", "Table");
    model.result().export("tbl1").set("table", "tbl1");
    model.result().export("tbl1").set("filename", "MagneticBHRresponse.txt");

    saveMyModel(model, saveFileName);

    model.result().export("tbl1").run();
}

public static void saveMyModel(Model model, String saveFileName){
    try{
        model.save(saveFileName);
    }
    catch(Exception e){}
}

public static String readInputArgs(){
    String fileName=null;
    try{
        BufferedReader argsFile = new BufferedReader(new FileReader("magInputArgs.txt")); //opens file containing what should be commandline args
        fileName = argsFile.readLine();
    }catch(Exception e){}
    return fileName;
}
}

```

```

function makeMatFiles(saveFile,folder)
% clear all %because matlab freaks out if something is saved from another run

% %%Read File
disp(saveFile)
info=split(saveFile,' ');
numOfParticles=str2num(info{3});
volFrac=str2num(info{5});

fields{1}=strcat('Ex=',num2str(0));
fields{2}=strcat('Ey=',num2str(0));
fields{3}=strcat('Ez=',info{7});
fields{4}=strcat('Hx=',num2str(0));
fields{6}=strcat('Hz=',info{9});

if length(info)>9
    fields{5}=strcat('Hy=',info{11});
else
    fields{5}=strcat('Hy=',num2str(0.0));
end
disp(fields)
%%
%%*****Elastic Response*****
elasticityMatrixFile=(strcat('/storage/work/d/dqw5479/java_scripts/',folder,'/',saveFile,'/ElasticityMatrix.txt'));%gets the full file string
fileID=fopen(elasticityMatrixFile);%opens the file for reading
if fileID~= -1
    for n=1:11 %reads the file line by line
        line=fgetl(fileID);
        if(n>5)%after header of 5 lines
            tempArr=split(line,' ');%splits around whitespace that is exactly 5 spaces into vertical array
            tempArr = tempArr(~any(cellfun('isempty', tempArr), 2), :);
            ElasticityString(n-5,:)=tempArr. ';%transposes the array back to original format in file
        end
    end
    %Need to convert each field to a number
    for i=1:6
        for j=1:6
            ElasticityData(i,j)=str2num(ElasticityString{i,j});
        end
    end
    ComplianceMatrix=inv(ElasticityData);
    Ex=1/ComplianceMatrix(1,1);
    Ey=1/ComplianceMatrix(2,2);
    Ez=1/ComplianceMatrix(3,3);
    %
    vxy=abs(ComplianceMatrix(1,2)/ComplianceMatrix(2,2));
    %
    vyz=abs(ComplianceMatrix(2,3)/ComplianceMatrix(3,3));
    %
    vxz=abs(ComplianceMatrix(1,3)/ComplianceMatrix(3,3));
else
    ElasticityData=0;
    Ex=0;
    Ey=0;
    Ez=0;
end
%%
clear collection
%%*****Magnetic Repsonse
magneticResponseFile=(strcat('/storage/work/d/dqw5479/java_scripts/',folder,'/',saveFile,'/MagneticBHRResponse.txt'));%gets the full file string
fileID=fopen(magneticResponseFile);%opens the file for reading
if fileID~= -1
    for n=1:6 %reads the file line by line
        line=fgetl(fileID);
        stringLength=size(line);
        if(n>5 && stringLength(2)>1)%after header of 5 lines
            tempArr=split(line,' ');
            items=size(tempArr);
            count=1;
            for k=1:items(1)
                content=char(tempArr(k));
                stringLength=size(content);
                if stringLength(2)>0
                    %collection(count,:)=str2num(tempArr{k})/(4*pi*1e-10);
                    collection(count,:)=str2num(tempArr{k})/(4*pi*1e-10*8.29e-11);%emu/g* for 100 particles
                    count=count+1;
                end
            end
            MagneticResponse(n-5,:)=collection. ';
        end
    end
end
fclose(fileID);
else
    MagneticResponse=0;
end
%%
clear collection
%%*****Dielectric Response*****
%for loadCase=1:3
%dielectricResponseFile=(strcat('/storage/work/d/dqw5479/java_scripts/',folder,'/',saveFile,'/DielectricResponse',num2str(loadCase),'.txt'));%gets the full file string
%fileID=fopen(dielectricResponseFile);%opens the file for reading
%if fileID~= -1
%for n=1:6 %reads the file line by line
%line=fgetl(fileID);
%if(n>5)%after header of 5 lines
%tempArr=split(line,' ');
%items=size(tempArr);
%count=1;
%for k=1:items(1)
%content=char(tempArr(k));
%stringLength=size(content);
%if stringLength(2)>0
%collection(count,:)=str2num(tempArr{k});
%count=count+1;
%end
%end
%dielectricResponse(loadCase,:)=collection. ';
%end
%end

```

```

        %fclose(fileID);
    %else
        %DielectricResponse=0;
    %end
%end
%if length(DielectricResponse)>1
    %D=DielectricResponse(:,1:3).';
    %E=DielectricResponse(:,4:6).';
    %x=linsolve(E,D);
    %eps=8.85418782e-12;
    %e_rel=x/eps;
%else
    %e_rel=0;
%end
%%
%%*****AC Response*****
initData=(strcat('/storage/work/d/dqw5479/java_scripts/',folder,'/',saveFile,'/',saveFile,'_init_data.txt'));%gets the full file string
fileID=fopen(initData);%opens the file for reading
if fileID~= -1
    line=fgetl(fileID);
    tempArr=split(line, ' ');
    cap_dist=str2num(tempArr{3})
end

for loadCase=1:3
    ACResponseFile=(strcat('/storage/work/d/dqw5479/java_scripts/',folder,'/',saveFile,'/ACResponse',num2str(loadCase),'.txt'));%gets the full file string
    fileID=fopen(ACResponseFile);%opens the file for reading
    if fileID~= -1
        for n=1:6 %reads the file line by line
            line=fgetl(fileID);
            if(n>5)%after header of 5 lines
                tempArr=split(line, ' ');
                items=size(tempArr);
                count=1;
                for k=1:items(1)
                    content=char(tempArr(k));
                    stringLength=size(content);
                    if stringLength(2)>0
                        collection(count,:)=str2num(tempArr{k});
                        count=count+1;
                    end
                end
            end
            ACResponse(loadCase,:)=collection.';
            ACResponse(loadCase,2);
            Real=real(ACResponse(loadCase,2))
            Imag=imag(ACResponse(loadCase,2))
            I0=sqrt(Real^2+Imag^2);
            Phase=atan(Imag/Real);
            e_rel(loadCase)=I0*sin(Phase)/(cap_dist*8.85e-12*6.28e5);
            sigma(loadCase)=I0*cos(Phase)/cap_dist;
        end
    end

    fclose(fileID);
    else
        e_rel=0;
        sigma=0;
    end
end

%%
%%*****Magnetization*****

%%CURRENTLY SET TO 100 PARTICLES MASS MAG!!!!!!
lengths=size(MagneticResponse);
if lengths(2)==1
    Magnetization=0;
else
    Magnetization=MagneticResponse;
end

%%
%%Saves .mat files with all relevant data
matName=strcat('/storage/work/d/dqw5479/java_scripts/',folder,'/',saveFile,'/',saveFile,'.mat');
disp(matName)
save(matName,'fields','volFrac','Magnetization','e_rel','sigma','ElasticityData','Ex','Ey','Ez');

% clear MagneticResponse DielectricResponse ElasticityData;

exit

```

```

#!/bin/bash

#PBS -l nodes=1:ppn=6
#PBS -A open
#PBS -l pmem=100gb
#PBS -l walltime=12:00:00
#PBS -m n
#PBS -l feature=rhel7

#48 hrs 100gb nodes1 ppn6
#Move to submission directory
cd $PBS_O_WORKDIR

#Set environment
module use /gpfs/group/RISE/sw7/modules
module load comsol/5.4
module load matlab
echo $numOfParticles
#Run matlab

d=$(date +"%Y-%m-%d-%H-%M")
folderName=${d}_numOfParticles_${numOfParticles}_volFrac_${volFrac}_Ez_${Ez}_Hz_${Hz}_Hy_${Hy}
mkdir "$folderName"
saveFile="$folderName/${folderName}"
echo $saveFile
#echo $saveFile>inputArgs.txt
#echo $saveFile>magInputArgs.txt
#echo $saveFile>dielectricInputArgs.txt

#Run matlab particle simulation
#matlab -nodesktop -nosplash -r "DPM3Dconv_run_file_New(${numOfParticles},${volFrac},${Ex},${Ey},${Ez},${Hx},${Hy},${Hz},'${saveFile}')"

#ellipsoidal distance
#matlab -nodesktop -nosplash -r "DPM3Dconv_run_file_NewForce(${numOfParticles},${volFrac},${Ex},${Ey},${Ez},${Hx},${Hy},${Hz},'${saveFile}')"

#Run java script to generate model
#comsol batch -mpirnk pbs -nnhost 3 -np 4 -mpidebug 10 -tmpdir ~/scratch/comsol_tmp -recoverydir ~/scratch/comsol_recovery -inputfile effectiveMediumModel.class -comsolinfile comsolclusterbatch.ini
comsol batch -mpirnk pbs -np 3 -np 2 $PBS_NUM_NODES -tmpdir ~/scratch/comsol_tmp -recoverydir ~/scratch/comsol_recovery -inputfile effectiveMediumModel.class -comsolinfile comsolclusterbatch.ini

#Run java script for magnetic model
comsol batch -mpirnk pbs -nnhost 3 -np 2 -tmpdir ~/scratch/comsol_tmp -recoverydir ~/scratch/comsol_recovery -inputfile magneticEffectiveMediumModel.class -comsolinfile comsolclusterbatch.ini

#Run java script for dielectric model
comsol batch -mpirnk pbs -nnhost 3 -np 2 -tmpdir ~/scratch/comsol_tmp -recoverydir ~/scratch/comsol_recovery -inputfile dielectricACEffectiveMediumModel.class -comsolinfile comsolclusterbatch.ini

#Run matlab script to make .mat file
matlab -nodesktop -nosplash -r "makeMatFile('${folderName}')"

#Add to Database
databaseName="EffectiveMedium"
tableName="Runs"
python addToDb.py ${folderName} ${tableName} ${databaseName}

```



```

from datetime import datetime as dt
import sqlite3 as sl
import os
import scipy.io as sio
import numpy as np
from sklearn.utils import shuffle
from sklearn.preprocessing import MinMaxScaler
from sklearn.svm import SVR
import random
import matplotlib.pyplot as plt
import imageio
import copy
import math
from scipy.optimize import minimize
from scipy.optimize import Bounds
import pickle
from numpy.linalg import norm
from scipy.stats import qmc
import subprocess
from os.path import exists
from sklearn.multioutput import MultiOutputRegressor
import pandas as pd

con=sl.connect('COMSOL_RUNS_OPT.db')
cursor=con.cursor()

def collectData():
    #bookkeeping
    date_index=0
    particle_index=1
    volFrac_index=2
    Ez_index=5
    Hy_index=7
    Hz_index=8
    mag_x_index=9
    mag_y_index=10
    mag_z_index=11
    mod_x_index=12
    mod_y_index=13
    mod_z_index=14
    e_rel_x_index=15
    e_rel_y_index=16
    e_rel_z_index=17

    inputs=[]
    results=[]
    justPerm=[]
    justMod=[]
    justMag=[]

    sql="""SELECT * FROM SVM_COMBINED_OPTIMIZATION"""
    with con:
        mySet=con.execute(sql).fetchall()

    for data in mySet:
        dateStamp=data[date_index]
        modulus=[data[mod_x_index],data[mod_y_index],data[mod_z_index]]
        e_rel=[data[e_rel_x_index],data[e_rel_y_index],data[e_rel_z_index]]
        magnetization=[data[mag_x_index],data[mag_y_index],data[mag_z_index]]
        #print(e_rel)

        volFrac=data[volFrac_index]
        Ez=data[Ez_index]
        Hy=data[Hy_index]
        Hz=data[Hz_index]
        simulationResult=[abs(magnetization[0])/30,abs(magnetization[1])/30,abs(magnetization[2])/30,modulus[0]/3e6,modulus[1]/3e6,modulus[2]/3e6,e_rel[0]/32,e_rel[1]/32,e_rel[2]/32]
        sims=[]
        prediction=[]
        for i in range(len(simulationResult)):
            sims.append(round(simulationResult[i],2))

        inputs.append([volFrac/30,Ez/100000000,Hy/250000,Hz/250000])
        print([volFrac/30,Ez/100000000,Hy/250000,Hz/250000])
        print([volFrac,Ez,Hy,Hz])
        results.append(sims)
        justPerm.append([e_rel[0]/15,e_rel[1]/15,e_rel[2]/15])
        #print([e_rel[0]/15,e_rel[1]/15,e_rel[2]/15])
        justMod.append([modulus[0]/3e6,modulus[1]/3e6,modulus[2]/3e6])
        justMag.append([abs(magnetization[0])/30,abs(magnetization[1])/30,abs(magnetization[2])/30])

    return inputs,results,justMod,justMag,justPerm

def mapDesignSpace():
    global model_mod
    global model_mag
    global model_perm

    discretization=21
    volFrac=((np.linspace(0,1,discretization)*.29)+.01)/.3

    Ez=np.linspace(0,1,discretization)
    print(Ez)
    Hy=np.linspace(0,1,discretization)
    Hz=np.linspace(0,1,discretization)
    svr_results=np.zeros((9))
    myInputs=[]
    x=[]
    y=[]
    z=[]
    for i in range(len(Ez)):
        for j in range(len(Hy)):
            for k in range(len(Hz)):
                for l in range(len(volFrac)):
                    inputs=[volFrac[l],Ez[i],Hy[j],Hz[k]]
                    mods=model_mod.predict(inputs)[0]
                    mags=model_mag.predict(inputs)[0]
                    erels=model_perm.predict(inputs)[0]
                    #mags,mods,erels -> [0][2],[0][5],[0][8]
                    x.append(mags[2]*30)
                    y.append(mods[2]*3)
                    z.append(erels[2]*15)
                    myInputs.append(inputs)

    #barPlotBins(x)
    mags, mods, erels = HashMapResults(x,y,z,myInputs)

    #ax.scatter(x,y,z,label=str(volFrac))
    #ax.set_xlabel("magnetization")
    #ax.set_ylabel("modulus")
    #ax.set_zlabel("erel")
    return mags, mods, erels

def makePlot(plt,myMor):
    fig = plt.figure()
    ax = fig.subplots(1,3)
    make2DSVMPLOT(myMor,ax,0.3,"#a80a5c")
    make2DSVMPLOT(myMor,ax,0.1,"#078c28")
    make2DSVMPLOT(myMor,ax,0.05,"#6229c2")
    make2DSVMPLOT(myMor,ax,0.01,"#fa9200")

```

```

plt.show()

def barPlotBins(results):
    bins=np.linspace(0,30,31)
    x=[]
    y=[]
    for result in results:
        x.append(result[2])
        y.append(round(result[2]*30,0))

    print(y)
    df = pd.DataFrame({"x":x})

    #bins= [0,0.35,0.7,1]
    plt.hist(y, bins=bins, edgecolor="k")
    #plt.hist(df.values*30, bins=bins, edgecolor="k")
    #plt.hist(df.values, edgecolor="k")
    plt.xticks(bins)

    plt.show()

def HashMapResults(x,y,z,inputs):
    mags={}
    mods={}
    erels={}
    for i in range(len(inputs)):
        magnetization=round(x[i],1)
        #print(inputs[i][0],results[i][2]*30,magnetization)
        if magnetization in mags:
            mags[magnetization].append(inputs[i][0])
        else:
            mags[magnetization]=inputs[i]

        modulus=round(y[i],1)
        if modulus in mods:
            mods[modulus].append(inputs[i][0])
        else:
            mods[modulus]=inputs[i]

        permittivity=round(z[i],1)
        if permittivity in erels:
            erels[permittivity].append(inputs[i][0])
        else:
            erels[permittivity]=inputs[i]

    #print(mags[4])
    #print([0.01, 0.0, 0.5, 0.0] in mags[4])

    return mags, mods, erels

inputs,results,justMod,justMag,justPerm = collectData()

svr_rbf_mod = SVR(kernel="rbf", C=1, gamma='auto', epsilon=0.1)
svr_rbf_mag = SVR(kernel="rbf", C=1, gamma='auto', epsilon=0.1)
svr_rbf_perm = SVR(kernel="rbf", C=1, gamma='auto', epsilon=0.1)

mor_mod=MultiOutputRegressor(svr_rbf_mod)
mor_mag=MultiOutputRegressor(svr_rbf_mag)
mor_perm=MultiOutputRegressor(svr_rbf_perm)

global model_mod
global model_mag
global model_perm
model_mod=mor_mod.fit(inputs, justMod)
model_mag=mor_mag.fit(inputs, justMag)
model_perm=mor_perm.fit(inputs, justPerm)

for estimator in model_mod.estimators_:
    print(len(estimator.dual_coef_[0]))
for estimator in model_mag.estimators_:
    print(len(estimator.dual_coef_[0]))
for estimator in model_perm.estimators_:
    print(len(estimator.dual_coef_[0]))

#print(mor_new.score(inputs, results))
#makePlot(plt,mor)

mags, mods, erels = mapDesignSpace()
#testPoint=[0.01/.3,1,0,0]
testPoint=[0.96, 0.76, 0.41, 0.03]
print(testPoint)
testVal=[model_mag.predict([testPoint])[0][2],model_mod.predict([testPoint])[0][2],model_perm.predict([testPoint])[0][2]]
#print("mags: ",mags)
#print("mods: ",mods.keys())
#print("erels: ",erels.keys())
print("Target Properties: ")
print(round(testVal[0]*30,1),"[emu/g]", ",round(testVal[1]*3,1),"[MPa]", ",round(testVal[2]*15,1))
#print(mags[9])
#print(mods[3])
#print(erels[8])

listOffields=[]
#mags,mods,erels
desiredProp=[0.8,4.2,10.9]
while len(mags[desiredProp[0]])>0 and len(mods[desiredProp[1]])>0 and len(erels[desiredProp[2]])>0:
    currentField=mags[desiredProp[0]].pop(0)

    if (currentField in mods[desiredProp[1]]) and (currentField in erels[desiredProp[2]]):
        listOffields.append(currentField)
        #because the item is an array and shared between the three props dictionaries, the remove function for this field on one, removes it for all.... like it destroyed the array or something, but cool ok
        mods[desiredProp[1]].remove(currentField)

#print("Solution found? ->",[0.01/.3,1,0,0] in listOffields)
print("Solution found? -> ",testPoint in listOffields)
print("Possible Volume Fractions and Processing Fields: ")
for ins in listOffields:
    print(round(ins[0]*30,0),"%", ",round(ins[1]*100000000,0),"[V/m]", ",round(ins[2]*250000,0),"[A/m]", ",round(ins[3]*250000,0),"[A/m]")
print("Properties Found:")
for ins in listOffields:
    testVal=[model_mag.predict([ins])[0][2],model_mod.predict([ins])[0][2],model_perm.predict([ins])[0][2]]
    print(round(testVal[0]*30,1),"[emu/g]", ",round(testVal[1]*3,1),"[MPa]", ",round(testVal[2]*15,1))

```

```

#!/usr/bin/env python
import subprocess
import scipy.io as sio
import os

path = os.getcwd()
print(path)

#submits the job
qsub_command="""qsub test.pbs"""
#out=subprocess.Popen(qsub_command, shell=True, stdout=subprocess.PIPE)
#jobID=out.communicate()[0]
#jobIDnum=jobID.decode().split(".")
#print(jobIDnum[0])

#waits for the job to finish and return values
bashCommand="""bash checkJobStatus.sh {0}""".format(jobIDnum[0])
#out=subprocess.Popen(bashCommand, shell=True, stdout=subprocess.PIPE)
#out.communicate()
#print("DONE")

#open results file and read in processed data
f=open(path+"/optimizationRuns/results.txt","r")
resultsFile=f.readline().strip()
print(path+"/"+resultsFile)
#read mat file in
mat=sio.loadmat(path+"/"+resultsFile)
#populate variables to use in objective functions
e_rel=mat['e_rel']
magnetization=mat['Magnetization'][0]
print(e_rel)

```

```
#!/bin/bash

module load matlab

ID=${1}
echo $ID
#initialize the exit code
runCode=true

#while the run code is true check job status
while $runCode
do
    #gets the job status full line
    status=$(qstat -r $ID)
    #echo $status
    #splits the status string around whitespace
    array=($status)
    #gets the length of the array
    length="${#array[@]}"
    echo $length

    #returns the second to last item which is the job status
    myStatus=${array[$length-2]}
    #remove whitespace
    myStatus=${myStatus// /}
    echo $myStatus
    if [[ $myStatus == *"C"* ]] #if the job is complete, exit this script
    then
        #matlab -nodesktop -nosplash -r "optimizationExportData('${saveFile}')"
        runCode=false
    fi
done
```

# Denise Widdowson

## EDUCATION

---

### The Pennsylvania State University

Ph.D. in Mechanical Engineering, Advisor: Dr. Paris von Lockette

State College, PA

2021–2023

### The Pennsylvania State University

M.S. in Mechanical Engineering

GPA: 4.0

State College, PA

2019–2022

### University of Pittsburgh

B.S. in Mechanical Engineering, Cum Laude

Minor: Computer Science

Pittsburgh, PA

2010–2015

## PUBLICATIONS

---

1. D. Widdowson, A. Erol, and P. von Lockette, “A Computational Framework for Predicting Properties from Multifield Processing Conditions in Polymer Matrix Composites”, The ASME 2020 Conference on *Smart Materials, Adaptive Structures, and Intelligent Systems*, October 2020
2. D. Papula, Z. Ounaies, P. von Lockette, D. Widdowson, A. Erol, A. Masud, “Characterization and Quantification of Hierarchical Particle Microstructures in External Field-Processed Composites”, The ASME 2021 Conference on *Smart Materials, Adaptive Structures, and Intelligent Systems*, September 2021
3. D. Widdowson, A. Erol, D. Papula, Z. Ounaies, P. von Lockette, “Multi-Objective Optimization of Predicted Magnetic Properties From Multifield Processing Conditions in Polymer Matrix Particle Composites”, The ASME 2022 Conference on *Smart Materials, Adaptive Structures, and Intelligent Systems*, September 2022
4. A. Masud, D. Papula, A. Erol, C. Edson, D. Widdowson, P. von Lockette, and Z. Ounaies, (under review) “Towards Magneto-Active Polymer Composites: Experiment and Simulation”, *Smart Materials and Structures*
5. D. Papula, A. Erol, A. Masud, D. Widdowson, P. von Lockette, and Z. Ounaies, (planned submission) “Image Processing Letter”, *IEEE Transaction of Image Processing*

## EXPERIENCE

---

### The Pennsylvania State University

Graduate Research Assistant - MACS Lab

State College, PA

August 2019–Present

- Conduct finite element analysis for computational homogenization of composite properties
- Create a multi-object optimization scheme to find ideal constituent set(s) for multi-field processed composites
- Develop data science model including machine learning models for materials design research
- Collaborate with the Material Science department to experimentally validate simulation results

## SCHOLARSHIPS AND AWARDS

---

### NSF Intern Program

Awarded 2020

The project aims to use a parametric study to simulate the novel magnetic separation device developed by the sponsor, Biomagnetics Solutions, which can be used to separate biological cells of all types. This offers a means of continuous processing as opposed those commonly used today such as centrifuges and dielectrophoretic methods.