A Stochastic Finite Element Analysis Framework for the Multiple Physical Modeling of Filler Modified Polymers

by

Hamidreza Ahmadimoghaddamseighalani

A thesis submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

in

Engineering Management

Department of Mechanical Engineering

University of Alberta

Abstract

Polymers have various attractive properties (e.g., low volume mass density, low cost and excellent degradation resistance), and hence, they have become essential for a wide range of applications (e.g., aerospace, oil and gas and renewable energy industries). However, due to their limited mechanical, thermal and electrical properties in comparison with typical metallic materials (e.g., steel and aluminum alloys), polymers often need to be modified with different types of fillers to meet the requirements of specific applications. A multitude of filler materials (e.g., nano silver particles, carbon nano tubes, graphene) are available with a wide range of geometries, such as spherical, fiber and platelet shapes. Therefore, developing methods for efficiently characterizing and identifying mechanical and physical properties of filler modified polymers is an important engineering task. The objective of this thesis is to create and validate a numerical modeling framework for predicting mechanical, thermal and electrical properties of particulate polymer composites. In contrast to the high cost and time consuming nature of experiments, this framework is to provide a relatively time- and cost-effective means to guide material design and elucidate experimental observations and analysis results. The first phase of this research project was focused on developing a numerical multiphysics modeling framework to predict the mechanical and thermal properties (i.e., effective modulus of elasticity, Poisson's ratio, coefficient of thermal expansion, and thermal conductivity), of a single-phase particulate polymer composite with randomly distributed spherical particles. The second phase focus was on expanding the modeling framework to enable property predictions, specifically thermal properties (i.e., effective thermal conductivity), of randomly distributed but aligned cylindrical particulate composites fillers. In the final phase of the research, the numerical framework was extended to predict electrical properties (i.e., effective electrical conductivity, percolation threshold and piezoresistivity), of spherical shape particulate composites. Predicted data were compared against experimental values and analytical models in the different contexts, which indicated overall good agreement between the predictions from the developed modeling framework and experimental works. The developed modeling framework is a valuable contribution facilitating the study of a broad range of material properties, thus improving the material design of filler modified polymers, by comprehensively capturing random aspects of particle and composite morphology.

Preface

Parts of this thesis document were published in the form of peer reviewed journal papers, a book chapter, and conference proceedings as follows:

- Some sections of the content presented in Chapter 1 of this thesis document were published in Publications No.1 to No.5 as shown below.
- Some sections of the work presented in Chapter 2 of this thesis document were published in Publication No.1 as part of conference proceedings and Publication No.2 as a peer reviewed journal paper.
- Some sections of the work presented in Chapter 3 of this thesis document were published in Publication No.3 as a peer reviewed journal paper.
- Some sections of the work presented in Chapter 4 of this thesis document were published in Publication No.4 as a peer reviewed book chapter.
- Some sections of the work presented in Chapter 5 of this thesis document were published in Publication No.5 as a peer reviewed journal paper.

Publication No.1: Ahmadi Moghaddam, H.; Mertiny, P. Composite Polymer with Random Distribution of Spherical Particles. Canadian International Conference on Composite Materials. 2015.

Publication No.2: Ahmadi Moghaddam, H.; Mertiny, P. Stochastic Finite Element Analysis Framework for Modelling Mechanical Properties of Particulate Modified Polymer Composites. Materials. 2019, 12, 2777, DOI: 10.3390/ma12172777.

Publication No.3: Ahmadi Moghaddam, H.; Mertiny, P. Stochastic finite element analysis framework for modelling thermal conductivity of particulate modified polymer composites. Results Phys. 2018, 11, 905-914, DOI: 10.1016/j.rinp.2018.10.045.

Publication No.4: Ahmadi Moghaddam, H.; Mertiny, P. Stochastic finite element modeling of char-forming filler addition and alignment – Effects on heat conduction in polymer condensed phase. In Flame Retardants; Zafar, F.; Sharmin, E.; Intech Open: London, United Kingdom, 2018, DOI: 10.5772/intechopen.82878.

Publication No.5: Ahmadi Moghaddam, H.; Mertiny, P. Stochastic Finite Element Analysis Framework for Modelling Electrical Properties of Particle-Modified Polymer Composites. Nanomaterials. 2020, 10, 1754, DOI: 10.3390/nano10091754.

I performed all the work presented in this thesis document with the assistance and guidance of my PhD supervisor Dr. Pierre Mertiny.

Acknowledgments

First of all, I would like to express my sincerest gratitude to my supervisor, Dr. Pierre Mertiny for his guidance, support and excellent supervision throughout my doctoral program. Completing of this research program would have been impossible without his support and encouragement. Also, I would like to thank the members of my doctoral supervisory committee, Drs. Walied Moussa and James Hogan for their time and valuable advice. In addition, I would like to thank the member of the examining committee, Dr. Abbas Milani, Dr. Mustafa Gul and the examining committee chair Dr. Benjamin Cheung, for their time evaluating this thesis.

Furthermore, I would like to thank Hikalibre Equipment Ltd., especially Patrick Rabby, Tod Hohl and Tony Standbridge for their support and assistance throughout my program.

Last but certainly not the least, I would like to thank my parents, Saied Ahmadi Moghaddam, Shayesteh Oshagh al Hosseini, for their support and encouragement for pursuing my studies towards my graduate degree and all accomplishments that I have made so far. Finally, I am unmeasurably grateful to my wife, Sharareh Mohammad Taheri. I love you so much and appreciate all your patience and support during my studies. I also dedicate this Ph.D. thesis to my lovely daughters Adrina Ahmadi Moghaddam and Avina Ahmadi Moghaddam who are the pride, joy and happiness of my life.

Table of Contents

Abstract .... **ii**

Preface .... **iii**

Acknowledgments .... **v**

Table of Contents .... **vi**

List of Tables .... **xii**

List of Figures .... **xiv**

Chapter 1 Introduction **1**

Nomenclature 1

1.1 Background 4

1.2 Mechanical Properties of Particulate Polymer Composites 5

1.3 Thermal Properties of Particulate Polymer Composites 6

1.4 Electrical Properties of Particulate Polymer Composites 8

1.5 Stochastic Analysis 9

1.5.1 Monte Carlo Simulation 10

1.5.2 Stochastic Analysis 13

1.5.3 Probability Distributions 18

1.5.4 Random Number Generation 20

1.6 Finite Element Analysis in Stochastic Simulation 22

1.6.1 Fundamental of Finite Element Analysis 22

1.6.2 Structural Numerical Analysis 24

1.6.3 Thermal Numerical Analysis 27

1.6.4 Structural-Thermal and Structural-Electrical FEA 31

1.6.5 Element Place in Finite Element Analysis 32

1.6.6 Contact Overview in Finite Element Analysis 34

1.6.7 Parametric Finite Element Analysis 36

1.6.8 Stochastic Finite element analysis 37

1.7 Knowledge Gap 41

1.8 Research Hypothesis and Objectives 41

1.8.1 Mechanical Characterization of Filler Modified Polymers 42

1.8.2 Thermal Characterization of Filler Modified Polymers 43

1.8.3 Electrical Characterization of Filler Modified Polymers 45

1.9 Thesis Organization 45

1.10 References 46

Chapter 2 Mechanical Characterization of Particulate Polymer Composites........................ 59

Nomenclature.. 60

2.1 Mechanical Characterization – Preliminary Study 61

2.1.1 Introduction 61

2.1.2 Modelling Approach 63

2.1.3 Results and Discussion 70

2.1.4 Conclusions 73

2.2 Mechanical Characterization - Comprehensive Investigation 74

2.2.1 Introduction 74

2.2.2 Stochastic FEA Framework 79

2.2.3 Model Details 84

2.2.4 Results and Discussion 89

2.2.5 Conclusions 99

2.3 References 100

Chapter 3 Thermal Characterization of Randomly Distributed Particulate Polymer Composites 108

Nomenclature.. 108

3.1 Introduction 110

3.2 Stochastic FEA Framework 117

3.3 Steady-State Thermal Numerical Model 122

3.4 Result and Discussions 127

3.5 Conclusions 137

3.6 References 137

Chapter 4 Thermal Characterization of Aligned Particulate Polymer Composites 145

Nomenclature 145

4.1 Introduction 146

4.2 Stochastic Finite Element Analysis Framework 148

4.3 Steady-state Numerical Modeling 153

4.4 Results and Discussion 158

4.5 Conclusions 168

4.6 References 168

Chapter 5 Electrical Characterization of Particulate Polymer Composites 174

Nomenclature 174

5.1 Introduction 176

5.2 Overview of SFEA Framework 180

5.3 Steady-State Electric Conduction Numerical Model 184

5.3.1 Electrical Conduction and Percolation Threshold Modeling 184

5.3.2 Electrical Piezoresistivity Modeling 190

5.3.3 Thermal-Electrical Numerical Model 191

5.4 Results and Discussion 193

5.4.1 Effective Electrical Conductivity and Percolation Behavior 193

5.4.2 Piezoresistivity Behavior 201

5.4.3 Temperature Effect on Effective Electrical Conductivity 204

5.5 Conclusions 207

5.6 References 208

Chapter 6 Conclusions 217

6.1 Conclusions and Discussions 217

6.2 Recommendations for Future Work 221

6.3 References 223

Bibliography 228

Appendix I 252

Appendix II 254

Appendix III 288

Appendix IV 296

Appendix V 305

Appendix VI 310

Appendix VII 324

List of Tables

Table 2-1 Mechanical properties of the constituent materials [16] [19]. 66

Table 2-2 Properties of filler particles and polymer matrix. 84

Table 2-3 Properties calculated along mutually perpendicular axes defining the RVE for 0.275 FVF. 90

Table 2-4 Statistical analysis of modeling results for the modulus of elasticity. 94

Table 2-5 Statistical analysis of modeling results for the Poisson's ratio. 94

Table 2-6 Statistical analysis of modeling results for the coefficient of thermal expansion. 95

Table 3-1 Thermal properties of filler particles and polymer matrix [54]. 122

Table 3-2 Thermal conductivity in mutually perpendicular directions in RVE with 8.00% filler volume fraction. 129

Table 3-3 Results for thermal conductivity simulations and statistical analyses for PEEK-matrix/Ag-particle composites. 131

Table 3-4 Filtered thermal conductivity simulation data and statistical analyses for PEEK-matrix/Ag-particle composites. 134

Table 4-1 Volumetric mass density and thermal conductivity of polymer and CNT [9, 28, 29]. ..155

Table 4-2 Average and directional thermal conductivities of composites with randomly distributed and randomly oriented filler particles. 160

Table 4-3 Results for longitudinal effective thermal conductivity and statistical analyses. 162

Table 4-4 Results for lateral effective thermal conductivity and statistical analyses. 163

Table 5-1 Electrical properties of polymer matrix and particles [10]. 185

Table 5-2 Mechanical properties of nano-silver particles and the polymer matrix. 190

Table 5-3 Coefficient of thermal expansion of fillers and matrix. 192

Table 5-4 Effective electrical conductivity results and statistical analyses for a silver/epoxy nanocomposite with particle size of 7 nm, tunneling distance of 1.5 nm and 30 vol% filler loading. 195

List of Figures

Figure 1-1 Monotonic increasing function for creating random numbers...................................... 18

Figure 1-2 Illustration of spring elements. 23

Figure 1-3 Structural analysis sequence schematic. 25

Figure 1-4 Linear heat conduction element schematic. 28

Figure 1-5 schematic of a discretized linear heat conduction element. 29

Figure 1-6 Near field structural- thermal contact schematic. 35

Figure 1-7 SFEA framework schematic for predicting thermal property. Step 1 represents the numerical model geometry and mesh. Steps 2 to 4 illustrate different types of contacts i.e. particle-to-matrix, particle-to-particle and particle-to-RVE surface. Step 5 and 6 show applying stochastic principals and final thermal property results, respectively. 40

Figure 2-1 Size distribution histogram for glass beads, adopted from [17]. 65

Figure 2-2 RVE configuration and coordinate system. 65

Figure 2-3 Randomly dispersed particles inside the RVE with filler volume fraction of 19.4% (A), and reaction force versus displacement data computed for RVE with 5% filler volume fraction (B). 67

Figure 2-4 Meshing of the RVE, showing the polymer matrix (A) and the particles (B). 69

Figure 2-5 Convergence study for different mesh refinement levels. The average von Mises stress relates to RVE face on which the uniform displacement was imposed. Date was normalized with respect to ~1,785,000 degrees of freedom. 70

Figure 2-6 Modulus of elasticity for epoxy composite with spherical graphite filler. 72

Figure 2-7 Poisson’s ratio for epoxy composite with spherical graphite filler. 72

Figure 2-8 Modulus of elasticity for epoxy composite with glass beads. 73

Figure 2-9 Poisson's ratio for epoxy composite with spherical graphite filler. 73

Figure 2-10 High-level architecture of developed stochastic FEA framework which shows how individual modules are connected together to form the framework [48]. 80

Figure 2-11 Schematic representing algorithm of Monte Carlo Simulation (MCS) module [48]. .. 81

Figure 2-12 Schematic illustrating the Random Number Generator (RNG) module [48]. 82

Figure 2-13 Schematic of FEA platform for model generation and analysis sub-processes [48]. ... 83

Figure 2-14 Glass bead size distribution, adapted from [17]. 85

Figure 2-15 Example meshing of (a) RVE and (b) particles. 86

Figure 2-16 FE model depicting normal stress on RVE face being subjected to uniform nodal displacement. 87

Figure 2-17 Deformations due to thermal expansion normal to the RVE plane (deformations scaled). 89

Figure 2-18 Distribution of particles in RVE for 0.45 FVF. Symbol size is indicative of particle size. 90

Figure 2-19 Normalized probability density (a) and cumulative distribution function (b) of modulus of elasticity data for simulated glass bead/epoxy composites. 92

Figure 2-20 Normalized probability density (a) and cumulative distribution function (b) of Poisson's ratio data for simulated glass bead/epoxy composites... 93

Figure 2-21 Normalized probability density (a) and cumulative distribution function (b) of coefficient of thermal expansion data for simulated glass bead/epoxy composites. 93

Figure 2-22 Comparison of modeling results for the modulus of elasticity with analytical model predictions and experimental data [19,20]. Error bars express experimental data scatter. 96

Figure 2-23 Comparison of modeling results for the Poisson's ratio with experimental data [20]. Error bars express experimental data scatter. 97

Figure 2-24 Comparison of modeling results for the coefficient of thermal expansion with Voigt model predictions. 98

Figure 3-1 Schematic of the SFEA framework showing connections between individual modules. 117

Figure 3-2 Schematic illustrating the algorithm for the Monte Carlo Simulation (MCS) module. 118

Figure 3-3 Schematic illustrating the Random Number Generator (RNG) module. 119

Figure 3-4 Schematic illustrating the algorithm for the FEA module. 121

Figure 3-5 RVE finite element mesh (A) and filler particles included in RVE devoid matrix (B).123

Figure 3-6 Illustration of modeling approach for capturing ITR and direct particle-to-particle heat transfer (A), temperature distribution at the contact interface (B). 125

Figure 3-7 Illustration of heat flux across a parallel plane between the warm and cold RVE faces. 126

Figure 3-8 The effective thermal conductivity relates to RVE front surface for different level of mesh refinement. 128

Figure 3-9 Sample spacial distribution of particle centers in RVE for 13.2% filler volume fraction. 128

Figure 3-10 Normalized probability density of effective thermal conductivity data for 13.2% filler volume fraction. 132

Figure 3-11 Normalized probability density of effective thermal conductivity data for different filler volume fractions (FVF). 134

Figure 3-12 Cumulative density function of effective thermal conductivity data for different filler volume fractions (FVF). 135

Figure 3-13 Thermal conductivity comparison for Silver Nano particles embedded in PEEK [54]. 136

Figure 4-1 Schematic of the SFEA framework showing connections between individual modules [27].. 149

Figure 4-2 Monte Carlo simulation (MCS) module algorithm [27]. 151

Figure 4-3 Schematic of series of spheres representing rod-shaped particles. 151

Figure 4-4 Random number generator algorithm [27]. 153

Figure 4-5 FEA module algorithm [27]. Schematic illustrating the algorithm for the FEA module. 153

Figure 4-6 Illustration of local coordinate systems for rod-shaped particles. 155

Figure 4-7 Finite element mesh of matrix (left) and aligned rod-shaped particles occupying the RVE devoid matrix (right). 157

Figure 4-8 Illustration of longitudinal (left) and lateral (right) heat flux between opposing warm and cold RVE surfaces. 157

Figure 4-9 Convergence study for composites with 4% filler volume fraction. 159

Figure 4-10 Example of randomly distributed and randomly oriented filler particles. 160

Figure 4-11 Normalized probability density of effective thermal conductivity data for the direction lateral to filler alignment and 4.0% filler volume fraction. 162

Figure 4-12 Normalized probability density of effective thermal conductivity data for the direction of filler alignment and different filler volume fractions (VF). 164

Figure 4-13 Normalized probability density of effective thermal conductivity data for the direction lateral to filler alignment and different filler volume fractions (VF). 165

Figure 4-14 Cumulative density function of effective thermal conductivity data for the direction of filler alignment and different filler volume fractions (VF). 165

Figure 4-15 Cumulative density function of effective thermal conductivity data for the direction lateral to filler alignment and different filler volume fractions (VF). 166

Figure 4-16 Effective thermal conductivity of randomly oriented and aligned rod-shaped particles embedded in epoxy polymer matrix. 166

Figure 4-17 Relative increase in effective thermal conductivity of randomly oriented and aligned rod-shaped particles embedded in epoxy polymer matrix. 167

Figure 5-1 Schematic of the SFEA framework showing connections between individual modules [58]. 180

Figure 5-2 Schematic illustrating the algorithm for the Monte Carlo Simulation (MCS) module [58]. 182

Figure 5-3 Schematic illustrating the Random Number Generator (RNG) module [58]. 183

Figure 5-4 Schematic illustrating the algorithm for the FEA module [58]. 184

Figure 5-5 A cut-away view of the finite element mesh for RVE and electrical charge plates (left) and an isolated view of electrical charge plates and particles (right) for the 0.3 filler volume fraction. 186

Figure 5-6 A schematic depicting the modeling approach for direct particle-to-particle electrical contact. 187

Figure 5-7 Electrical conductivity versus tunneling distance for different polymer barrier heights λ, based on Eq. (6-7). 189

Figure 5-8 Example of simulated electrical current density on an electrical charge plate after achieving percolation conditions. 190

Figure 5-9 Effective electrical conductivity for different model mesh refinements for 21 vol% filler loading. 194

Figure 5-10 A normalized probability density graph for effective electrical conductivity data of a silver/epoxy nanocomposite system with a particle size of 3 nm, tunneling distance of 1.5 nm, and 30 vol% filler loading. 196

Figure 5-11 The cumulative distribution function for effective electrical conductivity data of a silver/epoxy nanocomposite system with a particle size of 3 nm, tunneling distance of 1.5 nm, and 30 vol% filler loading. 197

Figure 5-12 Predicted effective electrical conductivity versus filler volume fraction for silver/epoxy nanocomposites with different particle sizes (D) and tunneling distances (d). Experimental (Exp.) data (square symbols) from [14], with (a) and (b) indicating tests before and after thermal treatment, respectively. 198

Figure 5-13 The influence of filler loading, tunneling distance, and particle size on the effective electrical conductivity data for silver/epoxy nanocomposites above percolation. 200

Figure 5-14 The probability of reaching electrical percolation, for an effective electrical conductivity threshold of 1.0 S/m, versus filler volume fraction for silver/epoxy nanocomposites with different particle sizes (D) and tunneling distances (d). 200

Figure 5-15 The influence of filler loading, tunneling distance and particle size on the probability of for a silver/epoxy nanocomposite reaching electrical percolation, for an effective electrical conductivity threshold of 1.0 S/m. 201

Figure 5-16 An illustration of the applied tensile strain and Poisson's effects on filler particle proximity: particles in undeformed RVE (left), and particles in deformed RVE (right), for 90,000 microstrain applied to the RVE (horizontal), RVE and particle size of 30 nm and 5 nm, respectively, and filler loading of 30 vol%. For clarity, the matrix material is not shown, and deformations and displacements are depicted with a 150× scaling factor. 202

Figure 5-17 Normalized effective electrical resistivity versus applied mechanical strain for silver/epoxy nanocomposites with a particle size of 3 nm, tunneling distance of 1.5 nm, filler loading of 30 vol%, and polymer matrix Poisson's ratio (PR) of 0.35, 0.40, and 0.45.............................. 204

Figure 5-18 Normalized electrical conductivity, considering only thermal expansion effects, versus temperature for silver/epoxy nanocomposites with a particle size of 3 nm, tunneling distance of 1.5 nm, and filler loading of 30 vol%.. 206

Figure 5-19 Normalized electrical conductivity, considering thermal expansion and electron activity effects, versus temperature for silver/epoxy nanocomposites with a particle size of 3 nm, tunneling distance of 1.5 nm, and filler loading of 30 vol%. 206

Chapter 1

Introduction

Nomenclature

ϕ	filler volume fraction
X_θ	family of random patterns
Θ	index set
σ	standard deviation
$X_{(u,v,w)}$	three-dimensional vectors
ζ	shape factor
η	random variable
μ	mean
t_k	conditional distribution
$r(i)$	random number
E_{L}	lower bound modulus of elasticity
$Fs(s)$	stress distribution function
E_{T}	upper bound modulus of elasticity
$Fn(s)$	experimental distribution
E_{f}	filler modulus of elasticity function
E_{m}	matrix modulus of elasticity
E_{c}	effective modulus of elasticity
x	random variable
$P(x)$	probability density

N	set of samples
$E(f)$	quantity of interest
$f(x)$	scalar quantity
$R(t)$	structural reliability
$f_r(r)$	material resistance
$f_s(s)$	stress probability density function
$F(\eta)$	uniform distribution
F	force
K	spring constant
δ	spring deformation
u	deformation
σ	stress vector
D^T	differential matrix operator
$\bar{P}_V$	body forces
E	material matrix
ϵ	strain vector
D	differential matrix operator
A^T	transformation matrix
$\bar{P}_S$	surface tractions
V^e	element displacement vector
P^e	internal nodal force
W_{int}	virtual work of internal forces
W_{ext}	virtual work of external forces

q	heat flux
A	cross section area
k	thermal conductivity coefficient
T	temperature vector
F	load vector
ρ	density
C	specific heat
$T(t)$	time dependent temperature
$\{F(x,t)\}$	heat flux vectors
$[C(x,t)]$	capacitance matrix
n	number of time steps
α	coefficient of thermal expansion vector
T_0	absolute temperature
C_{P}	specific heat
K_{UU}	structural stiffness
K_{TT}	thermal conductivity coupling
K_{UT}	thermo-elastic coupling
C_{TT}	specific heat damping
C_{UU}	structural damping
M_{UU}	mass
F	force
Q	pressure
U	displacement

T	temperature
TCC	thermal contact conductance coefficient
ECC	electrical contact conductance coefficient

1.1 Background

Polymers have various attractive characteristics, such as low volumetric mass density [1], excellent degradation resistance and low cost [2-3]. However, material properties (such as mechanical, thermal and/or electrical) can be enhanced by adding one or several types of suitable fillers to a polymer [4], creating single, binary or even ternary modified polymer composites [5]. Properties enhancements should occur synergistically, that is, the enhancement of a certain material property should not occur at the detriment of another. In terms of advance material manufacturing processes and technologies, researchers have made a substantial progress in achieving filler modified polymers with fully dispersed, well distributed, and even aligned particles [6-7]. However, the wide variety of possible micro- and nano-filler materials [8], their filler loading and combinations, make the material design a challenging task. Moreover, the number of identified applications that would benefit from suitable modified polymers are increasing drastically, and it is hence critical to find alternative means beside experimentation for performing material characterization and identifying mechanical, thermal and electrical properties. Relying on experimental methods alone is insufficient for exploring the plethora of possible polymer filler modifications due to the costly and time-consuming nature of experimental work [9]. As such, accurate modeling techniques and expedient property prediction approaches are needed to help material engineers design advanced polymer products [10-12].

1.2 Mechanical Properties of Particulate Polymer Composites

Numerous studies have been performed and documented in the technical literature for the prediction of material mechanical properties of binary composites [13-16]. The stress field analysis for an isolated ellipsoidal inclusion developed by Eshelby is pioneering work in this field [17]. In other analytical works, models were proposed by Voight [18] and Reuss [18] to identify the lower and upper bounds for the modulus of elasticity as a function of Filler Volume Fraction (FVF) as shown in Eqs. (1-1) and (1-2), respectively.

$$E_{\mathrm{L}} = \phi E_{\mathrm{f}} + (1-\phi)E_{\mathrm{m}} \tag{1-1}$$

$$\frac{1}{E_{\mathrm{T}}} = \frac{\phi}{E_{\mathrm{f}}} + \frac{(1-\phi)}{E_{\mathrm{m}}} \tag{1-2}$$

where E_{L} and E_{T} are lower and upper bound modulus of elasticity; E_{f}, E_{m} and ϕ are the moduli of elasticity for the filler and matrix materials, and the FVF, respectively.

The Halpin-Tsai model, described by Eqs. (1-3) and (1-4), is another method of predicting the effective modulus of elasticity, which can be used for different particle geometry [19].

$$E_{\mathrm{c}} = E_{\mathrm{m}} \frac{1+\zeta\eta\phi}{1-\eta\phi} \tag{1-3}$$

$$\eta = (\frac{E_{\mathrm{F}}}{E_{\mathrm{m}}} - 1)(\frac{E_{\mathrm{f}}}{E_{\mathrm{m}}} + \zeta)^{-1} \tag{1-4}$$

where E_{c} and ζ are correspondingly the effective modulus of elasticity and a shape factor. The latter is calculated based on filler geometry, aspect ratio and loading direction.

Another technique for predicting the effective modulus of elasticity is the variational approach developed by Hashin and Shtrikman [20]. Other analytical techniques include models by Mori and Tanaka [21] as well as enhanced modeling approaches performed by Benveniste [22], Christensen and Lo [23], Hill [24] and Torquato [25].

Besides randomly oriented particle composites, aligned filler modified polymers have created significant interest among researchers for enhancing mechanical properties. High strength composite materials [26-28], sensors [29-30], and field emission displays [31] are examples of different applications that require filler alignment to achieve specific anisotropic mechanical properties. For example, Carbon Nano-Tubes (CNT) constitute a type of filler that can create anisotropic mechanical properties for many applications [32-33].

1.3 Thermal Properties of Particulate Polymer Composites

The improvement of polymer thermal properties has received considerable attention in recent years, and therefore, several empirical, theoretical models, and experimental methods were developed to effectively assess thermal properties such as the thermal conductivity of modified polymers [34-36]. Generally, adding highly thermally conductive fillers enhances polymer thermal conductivity. The effective thermal conductivity of filler modified polymers depends on different factors such as the filler thermal conductivity [37], the state of filler dispersion in the matrix [37-39], the composite morphology [40], and the filler to polymer matrix contact and interaction [41]. Adding well distributed and dispersed fillers with high thermal conductivity increases the effective thermal conductivity of modified polymers. The composite morphology is defined by the filler size and shape, its orientation and spatial distribution. Metallic and dielectric fillers are the two main categories that can be used to improve thermal properties [42-43]. Thermal property enhancements using fillers expanded the applications of polymers to include high conductive materials [44-48], semiconductor devices [49-50], and removing heat from heat generating environments such as fuel cells [51]. The latter have attracted significant attention from engineers for their high energy conversion efficiency and environmental friendliness [52]. The proton conductive membrane is a key component of a fuel cell that needs to provide effective heat transfer [53]. A Polymer Electrolyte Membrane (PEM) (i.e., a silica colloidal filler modified polymer) may be employed in this context [51-54]. These particles sufficiently increase thermal conductance to transfer heat that is generated in the process [53].

Another example for a thermal property enhanced polymer is Graphene Nano-Platelet (GNP) modified polyurethane elastomer. Such a material system may not only provide excellent wear and corrosion properties but can also provide excellent thermal conductivity to dissipate frictional heat. Given the fact that GNP particle have a high in-plane dimension to thickness aspect ratio, aligning GNP particles results in anisotropic thermal conductivity, which can maximize directional heat transfer required for certain applications. For example, the performance and life span of heavy-duty vehicle tires may be enhanced using an elastomer with aligned GNP filler in order to conduct heat generated within the material more effectively to the tire surface.

Besides increased thermal conductivity, filler modified polymers are also sought for applications requiring thermal insulation and/or thermal stability, in which case filler materials with low thermal conductivity are typically used. Adding micro glass beads to a polymer such as epoxy is a good example of this kind of material system. Industrial applications for thermally durable composites include components for heaters [55-56].

Finally, there are several types of analytical methods for predicting the thermal conductivity of filler modified polymers. However, no analytical solution is available for determining the thermal conductivity of anisotropic conductive materials. Therefore, alternative numerical methods, (e.g., Finite Element Analysis (FEA)), can be employed instead of analytical solutions to calculate the thermal conductivity of materials of this kind. The latter is pursued in this thesis.

1.4 Electrical Properties of Particulate Polymer Composites

Electrical properties of polymers are critical in a variety of industrial applications, such as in aerospace and automotive products. An important electrical property is the resistivity (or conductivity) of the material, with different experimental methods available for measuring this property (e.g., volume and surface resistivity using ASTM D257 [95]).

The electrical properties of polymer composites with electrically conductive fillers are influenced by a phenomenon called 'electron tunneling' in which electrical charge can be transferred from one particle to another if the distance between particles is less than an explicit value [57-61]. As a result, a nonlinear current-voltage relation is established between two particles [62]. Hence, for a composite material containing suitably dispersed and distributed electrically conductive particles a specific volume fraction exists at which one or more pathway for electrical charge transfer are established that render the material electrically conductive. This specific volume fraction is known as the percolation threshold [63]. Electrical properties of conductive polymer composites also depend on temperature affecting two physical phenomena: particle proximity (tunneling distance) as the result of composite thermal expansion, and increased electron activity [64-65]. The combination of these physical phenomena tends to increase electrical conductivity. Given that the electrical conductivity and percolation threshold is appreciably affected by the composite morphology and experimental conditions, using experimental methods for exploring and understanding the percolation behavior can be a time consuming and expensive process.

Besides experimental methods, various modeling approaches have been developed for predicting the electrical properties of particulate polymer composites [66-70]. While these methods have the advantage of being inexpensive, they frequently do not reflect the continuum nature of the composite material (i.e., considering the polymer matrix and filler particles as a resistor network), and hence, changes in proximity between filler particles due to thermal expansion or mechanical strain are typically not considered. Considering these challenges in material characterization and design, developing novel methods for predicting the electrical properties of composites with conductive filler is an attractive proposition.

1.5 Stochastic Analysis

Engineering analysis is an important step of the design process for creating safe products. More advance products with sophisticated material properties are conceived in the industry, leading to more complex engineering design processes and requiring new analysis methods to facilitate such processes. Establishing an analytical approach providing a closed form solution to a modeling problem is certainly an attractive means for predicting certain properties or conditions. However, such methods are not always available for solving problems, which motivates the use of numerical modeling (i.e., FEA). Moreover, in order to realistically predict material behavior, uncertainty in any parameter that affects the material properties should be taken into consideration. In the context of this thesis research, filler particle size and orientation are parameters that affect material properties. Therefore, FEA should be conducted in conjunction with stochastic analysis to enable predicting material properties in a thorough and reliable manner. Such an approach is often termed stochastic finite element analysis (SFEA) [71-72]. Simulations following this approach are typically performed as part of a Monte Carlo method to systematically compute material properties.

Several options are available to combine stochastic analysis with FEA for considering uncertainty to the simulations. One approach is using FEA packages with built-in stochastic capabilities [73-75]. This option makes the process of applying uncertainty to FEA straightforward compared to other approaches since such software packages provide the required functionality. The other basic approach is the use of advance programming language along with scripting techniques to connect stochastic analysis with FEA. This approach has several advantages as it provides the user with more control over the simulation processes (i.e., defining input parameters) and added flexibility in terms of applying uncertainty to the analysis [76] by generating random numbers for desired input parameter associated with the numerical analysis.

In addition to combining FEA and processes to incorporate uncertainty, Monte Carlo simulation, as the preferred modeling approach, requires scripting capabilities to develop procedures that calls the FEA process repeatedly and records results in a database. Part of the Monte Carlo simulation are principles in statistics, namely mean and variance, that are calculated at the end of each iteration in order to implement an iteration termination criterion based on an acceptable change in variance from the mean between iterations. Such an approach is completely feasible and sensible as the required time to do so is short compared to executing the FEA and connected uncertainty program processes. The following subsections discuss some of the fundamentals relating to stochastic analysis for the key aspects of the developed SFEA modeling approach.

1.5.1 Monte Carlo Simulation

The terms ‘simulation’ and ‘sampling’ are frequently used in the context of Monte Carlo methods. In fact, it is difficult to distinguish Monte Carlo sampling from Monte Carlo simulation. Monte Carlo sampling is used for deterministic mathematical or statistical problems in which a limited number of samples are needed to be drawn and there are no dynamic parameters to be simulated over time. ‘Simulation’ is an imaginary representation of reality which means drawing a random variable in the context of Monte Carlo methods, therefore, in Monte Carlo simulation a large number of random numbers are needed to be drawn from a probability distribution (e.g., uniform, normal and log-normal probability distributions), that enables obtaining the statistical properties or behavior of some phenomenon. These numbers can be drawn at the same time or once at several different times.

Monte Carlo simulation enables predicting outcomes and possibilities by forming a ‘scientific guess’ and using principles of interferential statistics [77]. Inferential statistics make reference to “a random sample that tends to exhibit the same properties as the population from which it is drawn” [77]. In other words, inferential statistics can be used to take data from samples and perform generalizations about a population. Sample size and sample variance are two important parameters which need to be carefully considered for achieving an acceptable level of confidence in performing Monte Carlo simulations. Generally, as the variance expands, a larger sample size is required for maintaining an acceptable level of confidence. This phenomenon is known as the Law of Large Number (LLN) [78-79]. LLN increases the chance of achieving an average expected result by repeating the same experiment for a large number of times in the simulation. An expression for approximating a distribution is shown in Eq. (1-5) by using a large number of samples. Eq. (1-6) shows how these samples can be used to compute expectations.

$$P_N(x) = \frac{1}{N}\sum_{i=1}^{N} 1\left(x^{(i)} = x\right) \qquad \overrightarrow{N \to \infty} \qquad P(x) \tag{1-5}$$

$$E_N(f) = \frac{1}{N}\sum_{i=1}^{N} f\left(x^{(i)}\right) \qquad \overrightarrow{N \to \infty} \qquad E(f) = \sum_{x} f(x)p(x) \tag{1-6}$$

where x, $P(x)$ and N are a collection of random variables, a probability density or distribution and a set of samples, respectively. $E(f)$ and $f(x)$ are correspondingly the quantity of interest and scalar quantity determined by x.

Monte Carlo simulation can be an expedient design tool for solving problems that are impossible or challenging to solved with analytical models. For example, Monte Carlo simulation enabled designing a staged fusion bomb by Teller-Ulam for the first time [80]. Nowadays, Monte Carlo simulation is a well-known and widely used research strategy that enables implementing randomness in a modelling approach.

In order to be able to perform Monte Carlo simulations, it is critical to decide on the type of probability distribution that best applies uncertainty to parameters involved in the model. Normal, log-normal and uniform distributions are examples of probability distributions for use in Monte Carlo simulations. In the present research study (i.e., the simulation of particulate modified polymers), a uniform probability distribution is a suitable probability distribution for generating sets of random numbers constituting Cartesian coordinates that enable identifying particle locations in a Representative Volume Element (RVE). On the other side, a normal probability distribution is an appropriate probability distribution for generating random numbers that represent particle angles in a particle alignment simulation. Applying randomness in particle locations, sizes, angles and other important parameters associated with the model are key components for developing a numerical framework that enables predicting mechanical, thermal and electrical properties of particulate polymer composites.

Another important aspect of the Monte Carlo simulation is identifying the number of iterations (i.e., sample size) required that enables generating accurate outcomes. Generally, increasing the sample size reduces the standard deviation and leads to generating more accurate results. However, considering the time and costs associated with the simulation, it is required to calculate a suitable sample size that provides an acceptable level of accuracy in a practicable period of time. Hahn calculated the sample size required for performing a Monte Carlo simulation, which is a function of an initial estimation of the population standard deviation, desired probability or confidence level, and a maximum allowable error in predicting the true average [101].

1.5.2 Stochastic Analysis

The term ‘stochastic’ by itself entails a parameter or variable randomly determined in a model based on a probability distribution or a random pattern. Stochastic analysis or stochastic simulation are other terms that are often used instead of Monte Carlo simulation. As mentioned earlier, in order to appreciate the value of stochastic modelling, this method shall be compared with deterministic modelling. It is very important to appreciate that, strictly speaking, deterministic mathematical models are unsuitable for solving complex problems since these methods consider limited distinct parameters only and are not capable of considering all the input parameters. Consequently, these model may neglect important parameters that may affect the outcomes. The outputs of deterministic mathematical models depend solely on input parameters and the boundary conditions. These models are inappropriate for predicting outcomes that are dependent on time and involve randomness in input parameters, or fail to determine these explicitly [81]. Therefore, in a deterministic modeling approach, by its very nature, there is only a single solution generated for a problem. However, stochastic models are capable of imposing randomness in input parameters and initial conditions that ultimately will lead to generating different outputs in every analysis.

Engineers often seek an explicit modeling approach by prescribing explicit values for input parameters or boundary conditions and calculating a respective outcome through the analysis. However, as the matter of fact, an explicit value does not exist for any parameters or event. Therefore, considering uncertainty for each parameter involved in the design is important for solving problems adequately. Different methods are available for applying uncertainty to a model e.g. fuzzy logic [82]. Being able to apply uncertainty to an FEA is a notable advantage of SFEA because it enables considering uncertainty for parameters involved in the design, and as a result, sophisticated problems can be solved.

In stochastic analysis, variation and fluctuation can be associated with a complex model including a large number of variables for producing different sets of answers in order to solve a specific problem. In some applications, it is important that stochastic models are able to consider a variety of outcomes since new variables can occur at any time in the future. As such, stochastic models can take uncertainty in analyses into account. In many applications, uncertainty in parameters is not occurring independent of time, and hence, model prediction may be rather complicated. Nevertheless, the strength of stochastic modeling is the ability to solve such problems accurately by running a large number of simulations known as stochastic projections [83].

Stochastic analysis has been applied widely to sophisticated engineering problems, e.g. structural reliability problems. For example, in structural analysis the reliability term is referring to the probability of stress calculated for the structure to be less than the yield strength. Structural reliability can thus be expressed with Eq. (1-7) [102].

$$R(t) = \int_{-\infty}^{\infty} f_s(s) \left[\int_{s}^{\infty} f_r(r)\, dr \right] ds \tag{1-7}$$

where $R(t)$ represent structural reliability, $f_r(r)$ and $f_s(s)$ are material resistance and structural stress probability density functions, respectively.

The probability distribution functions described in Eq. (1-7) have to be determined in advance to enable performing reliability analysis for a structure. However, in a realistic, complicated engineering problem, it is often challenging to determine a suitable probability distribution function that is most appropriate for predicting outcomes. Therefore, other methods can be used that avoid calculating required probability distribution functions and taking the integral of these functions in order to perform the reliability analysis.

First-order second-moment stochastic analysis approach, also known as mean value first-order second-moment analysis, is another statistical method that can be used when random variables are following normal or log-normal distributions and the equation stating the problem is straightforward to setup. However, if the probability distributions are not following normal or log-normal distributions and the equations stating the problem are complicated, it would be challenging to setup and solve the problem by using a first-order second-moment stochastic analysis approach. Thus, other statistical methods (e.g., Monte Carlo simulation) can be used that enable performing stochastic analysis and solving complicated problems where random variables related to the problem fall under a custom probability distribution [84].

In order to have a better understanding of stochastic analysis, the mathematical form of a family of random patterns or random variables can be presented with $\{X_\theta\}$, where θ is part of the index set Θ which often represents continuous time in stochastic analysis. However, if the index set Θ represents specific points in time the stochastic process is considered discrete time and the parameter θ can be changed to *n*. As the result, the stochastic process can be presented with $\{X_n\}$. The parameter Θ may also represent location in space instead of time. In this case, Θ is a vector that varies within three-dimensional space and may be a discrete process of $\{X_{(u,v,w)}\}$. Both time and space can be involved in stochastic analysis known as a spatio-temporal process.

In stochastic models, random variables generally depend on earlier values in both a discrete and continuous process as shown in Eq. (1-8).

$$Pr\ (X_{t_k}|X_{t_{k-1}}, X_{t_{k-2}}, \dots, X_{t_1}) \tag{1-8}$$

where t_k is greater than t_{k-1} representing a conditional distribution in stochastic models. This process also known as Markov chain [85] that describes a sequence of possible events in which the probability of each event depends on the previous event and it is a function of time. The other form of stochastic process refers to models in which random variables are independent of earlier values. This type of stochastic process is less complicated and most of the engineering problems can be solved with this stochastic method.

Calculating a structure's response to an applied load is an important step in structural reliability analysis. Nowadays, engineers frequently use numerical analyses that enable calculating the load response in engineering problems. FEA has increasingly been used for solving engineering problems supported by the rapid development in computer science and electronic devices (e.g., computers and supercomputers). Engineering problems are simulated considering their geometries, and defining material properties and boundary conditions applied to the geometries. However, as mentioned earlier, associated parameters, strictly speaking, do not have explicit values, and for that reason, stochastic models can be used to fill the gap and enable generating random variables for parameters involved in the analysis. Since a stochastic process requires performing a large number of FEA runs, engineers often use FEA software with built in scripting capabilities or develop customized codes that enable manipulating parameters involved in the model and automating the FEA process. Scripting capabilities are an important step for developing an SFEA framework. Note that SFEA typically entails handling large volumes of data, and hence, high performance computers are needed for efficient data processing and thus achieving acceptable run times.

One of the first considerations when performing FEA in conjunction with Monte Carlo simulation is creating proper probability distributions (i.e., a uniform, normal, log-normal or custom probability distribution) for the input parameters, such as for the geometry, boundary conditions and material properties. Since in manufacturing processes, components are created with certain manufacturing tolerances (e.g., particle sizes in the present context of filler modified polymer), associated variables often fall under a normal probability distribution because of the tendency of sizes of individual entities to cluster around a mean value with diminishing probability to meet upper and lower bound tolerances. In terms of boundary conditions, e.g. mechanical, thermal and electrical loads and constraints, suitable probability distributions can be defined from sufficiently large datasets collected from actual experiments. Such data is to be analyzed using statistical analysis methods to enable determining appropriate custom probability distributions. Similarly, probability distributions that represent material properties can be obtained from a multiplicity test data. In many cases, probability distributions for boundary conditions and material properties fall under a uniform and normal distribution, respectively.

A second important consideration for setting up an SFEA is random number generation to ensure proper creation of a desired probability distribution. In other words, an appropriate random number generator must be selected. For example, Eq. (1-9) describes a uniform distribution function $F(\eta)$ that can be used for creating a random number between zero and unity.

$$r = F(\eta) = \int_{-\infty}^{\eta} f(\eta) d\eta \tag{1-9}$$

where η is random variable and $F(\eta)$ is uniform distribution function that creates random numbers $r(i)$. The graphical representation of Eq. (1-9) is shown in Fig. 1-1. The curve in this figure increases monotonically yielding a subset of real numbers between zero and unity.

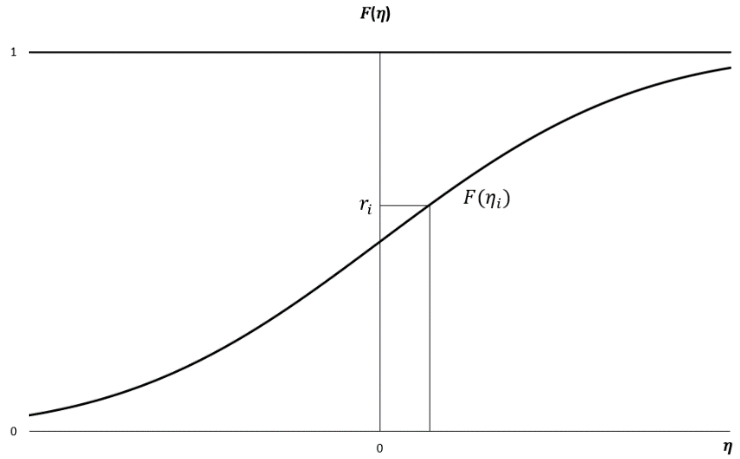


Figure 1-1 Monotonic increasing function for creating random numbers.

In many cases of SFEA, generate random numbers are needed for intervals other than zero and unity (e.g. dimensions). In such cases, Eq. (1-10) can be used to shift the upper and lower limit to numbers as defined by the problem.

$$R(x, y) = F(\eta) \times (y - x) + x \tag{1-10}$$

where R is a random number, η is a random variable, F is the uniform distribution function, x and y are required upper and lower bound limits, respectively. The limit values can be collected from experimental data.

1.5.3 Probability Distributions

As mentioned in the previous section, choosing an appropriate probability distribution is an important step in developing an SFEA framework, with uniform, normal and log-normal probability distributions being the most commonly used probability distributions for treating engineering problems. For a uniform probability distribution, the probability of having random variables for any value between designated upper and lower boundaries is equal. This probability distribution falls under the group of symmetrical probability distributions. The uniform probability density function and cumulative distribution function are presented in Eq. (1-11) and (1-12), respectively.

$$f(x) = \begin{cases} \dfrac{1}{b-a} & a \leq x \geq b \\ \\ 0 & x < a \; or \; x > b \end{cases} \tag{1-11}$$

$$F(x) = \begin{cases} 0 & x < a \\ \dfrac{x-a}{b-a} & a \leq x \geq b \\ 1 & x > b \end{cases} \tag{1-12}$$

where a and b are upper and lower boundaries.

In the context of the present research study (i.e., modeling of randomly distributed and dispersed particles embedded in polymer matrix), a uniform probability distribution is considered most appropriate for generating particle locations and orientations inside the RVE since the probability of having particles anywhere in the RVE space is the same. Nevertheless, other probability distributions can be used or even a custom probability distribution can be developed too, for example, to incorporate a bias in terms of particle alignment or clustering is desired.

A large set of engineering design parameters and manufacturing events fall under normal or log-normal probability distributions which is also known as a function of Gaussian probability distribution. These probability distributions are a subset of symmetric distributions for which the chance of having a random variable towards the middle of the distribution is higher than toward to outer limits of the distribution. Normal or log-normal probability distributions enable providing random variables even if the actual distribution is unknown. Quality control such as the Six Sigma concept is a good example in which normal probability distributions are used for improving manufacturing processes by utilizing probability theories in order to reduce the number of defects.

Normal probability distributions are practical in terms of the Central Limit Theorem (CLT), which expresses that if independent random variables are added the normalized sum of these variables falls under normal distributions and indicates that a normal distribution can be applied to the vast majority of problems that originally involved with other types of probability distributions. This theorem is also used for reducing the number of iterations enquired for performing stochastic finite element analysis.

1.5.4 Random Number Generation

In this doctoral research, the developed SFEA framework necessarily employs a random number generation method, and hence, it is important to consider best practice guidelines and different available methods for generating random numbers. The goal is to select the most effective and efficient method to meet the requirements of the developed SFEA framework.

From the early 1950's, researchers have been studying methods for generating random numbers, seeking to understand the impact and uncertainty of each method for different applications (e.g., in design, cost analysis, project management, and forecasting models [86-88]). Hence, various algorithms have been developed for generating random numbers, however, studies showed that not all of these algorithms are able to create true randomness and often fail by falling into a predictable pattern for a given problem [89-90]. Therefore, utilizing an appropriate method is critical for generating random numbers and creating true randomness in advanced modelling applications, especially to avoid artificial effects that are not part of the actual modeling problem.

MATLAB (MathWorks, Natick, Massachusetts, USA) is one of various numerical computing environment that uses a sophisticated algorithm for generating random numbers without aforementioned flaws. A Pseudorandom Number Generator (PRNG) is another complex algorithm that was developed for generating a set of numbers in order to estimate the properties of sequences of random numbers based on number theory [91]. PRNG is also known as Deterministic Random

Bit Generator (DRBG) since generated random numbers depend on an initial value known as 'seed'. It means that using same seed every time generates identical sets of random numbers, and for that reason using PRNGs only may not necessarily lead to true randomness. Therefore, the initial seed is typically changed during an analysis in order to generate a new set of random numbers and create true randomness. As a best practice, it is recommended to associate the initial seed with a clock function as this causes each set of random numbers being independent from previous sets as time progresses. The random number generation algorithm is often expressed in terms of a recursive formula as shown in Eq. (1-13):

$$x_{i+1} = mod(ax_i + c, m) \tag{1-13}$$

where a, c and m are given integers, and x_i and x_{i+1} are sequenced random numbers.

In addition to the random number generators mentioned above, quasi-random Number Generator (QRNG) is another type of frequently used method, also known as a low discrepancy sequence method, that generates random numbers that are far away from numbers existing in a data set. While this method can avoid random number clustering and in some applications decrease time required for convergence, it may generate random numbers that are too uniform, therefore, results may not pass randomness test. Pseudorandom numbers, on the other hand, are less uniform compared to quasi-random numbers, and they create better outcomes in many applications.

Stochastic analyses or Monte Carlo simulations are usually relying on PRNG for generating true randomness in models. Employing good-quality PRNGs to avoid having inferior or false results is therefore critical [92]. For example, Java is a widely used programing language in web-based applications that uses a Linear Congruential Generator (LCG) for generating random numbers. LCGs are PRNGs that are generating low quality random numbers compare to other available PRNGs [93], and therefore, a developer needs to make appropriate choices in terms of software tools when developing advance numerical frameworks.

1.6 Finite Element Analysis in Stochastic Simulation

1.6.1 Fundamental of Finite Element Analysis

Generally, the laws of physics can be expressed mathematically by using partial differential equations (PDEs) for both time-independent and time-dependent problems. Due to geometry features, engineering problems are typically complex in nature, as the result, these problems can often not be solved with enough detail using PDEs directly. Therefore, an approximation approach based on PDEs can be utilized for predicting final results where the complex engineering problem (i.e., geometry) is discretized. This approximation can be accomplished using a system of equations constituting a numerical model also known as finite element analysis (FEA). Having a high degree of freedom for choosing the model discretization is the most important advantage of the finite element method (FEM). The model discretization refers to the elements that compose the original structure. In general, smaller spatial and temporal discretization results in a better prediction of final results. Uniform discretization and diminished discretization in the areas of interest are the most common methods utilized in FEM.

FEM uses the element stiffness matrix which is a fundamental characteristic of FEA in order to identify the response of an element to change. The change can be deformation in the context of structural analysis, temperature change in case of a thermal analysis, or voltage change in an electrical analysis. The stiffness matrix carries the information regarding material behavior and geometry of the problem. Considering spring elements provides a basic example for a better understanding how a stiffness matrix can be created and utilized for solving engineering problems with FEM. In case of connecting two springs together and forming a boundary conditions as illustrated in Fig. 1-2, the constitutive equation that defines deformation of springs can be written as Eq. (1-14).

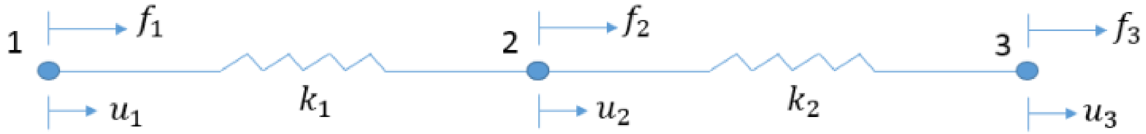


Figure 1-2 Illustration of spring elements.

$$F = K\delta \tag{1-14}$$

where k (or K) is the spring constant, and f (or F) and u are the force and displacement located at nodes 1 to 3, respectively. Eqs. (1-15) to (1-17) show the equilibrium equations.

$$F_1 + F_2 + F_3 = 0 \tag{1-15}$$

$$\delta_1 = u_2 - u_1 \tag{1-16}$$

$$\delta_2 = u_3 - u_2 \tag{1-17}$$

where δ denotes spring deformation. By assembling the equations for each element as well as considering that the sum of these equations is zero, Eq. (1-14) can be written in a form of matrix as Eq. (1-18).

$$\begin{bmatrix} F_1 \\ F_2^{(1)} + F_2^{(2)} \\ F_3 \end{bmatrix} = \begin{bmatrix} K_1 & -K_1 & 0 \\ -K_1 & K_{1+2} & -K_2 \\ 0 & -K_2 & K_2 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} \tag{1-18}$$

where $F_2^{(1)}$ and $F_2^{(2)}$ are the forces applied on node 2 shared between element 1 and 2, which enabled transferring boundary conditions from one element to another.

1.6.2 Structural Numerical Analysis

Since structural numerical analysis was extensively used in this research study, it is important to know the principles of this numerical method. In case of solving any type of structural analysis, three basic ‘principles’ are involved regardless of the type of structure applied boundary conditions and the nature of structure’s material. These are individual principles that can be distinguished from each other and having a good understanding of each of their significance is very important for performing a FEA.

“Equilibrium” is the first principle that enables relating stress to applied forces. The equation of equilibrium can be considered ‘linear’, if the displacement as the result of boundary conditions applied are sufficiently small. The second principle is “compatibility” that relates strain to displacement which depends on geometrical arguments and the type of deformation. “Stress-Strain Law” is the third principle which is also known as the empirical constitutive equation.

The simplest approach for solving a structural analysis is the “displacement method” where displacements shall be solved prior to solving other variables. Referring to Fig. 1-3, the route followed for solving a structural analysis problem with the displacement method is (1)-(3)-(4)-(2). However, if the structure is determined statically, the route shall be (1)-(2)-(3)-(4).

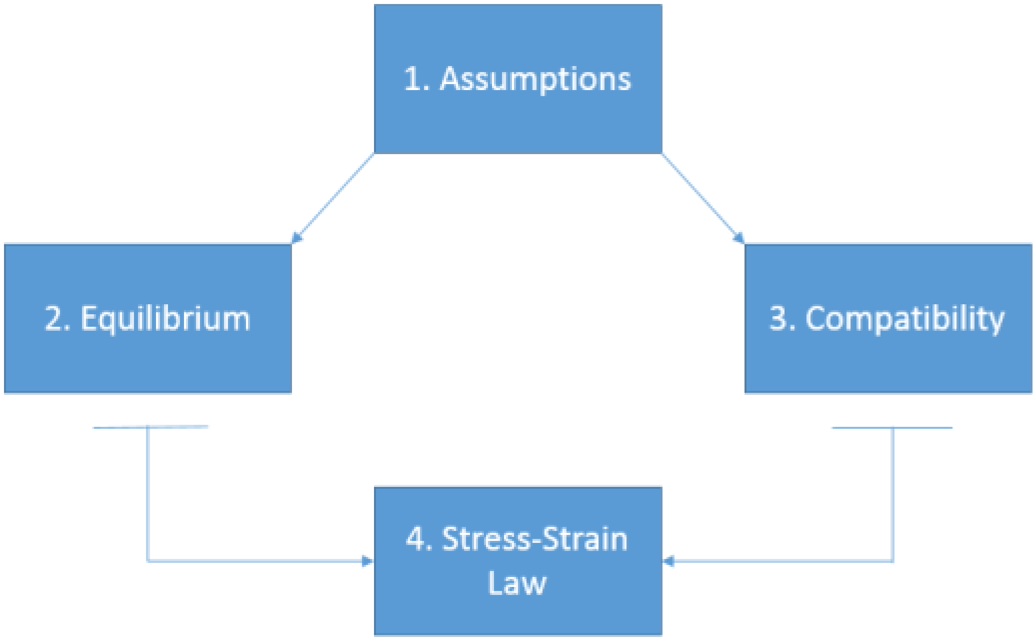


Figure 1-3 Structural analysis sequence schematic.

Particularly, FEM enables solving engineering problems that is based on the principles of virtual work that identifies the equilibrium equations, compatibility and constitutive equations. Considering an elastic body is subjected to a body force the equilibrium equation can be written as Eq. (1-19).

$$D^T\sigma + \bar{P}_V = 0 \tag{1-19}$$

where σ and $\bar{P}_V$ are stress vectors and body forces, respectively. D^T is differential matrix operator. The constitutive equation can be written as Eq. (1-20) that is known as Hook's law.

$$\sigma = E\epsilon \tag{1-20}$$

where E and ϵ are correspondingly the material matrix and strain vector that includes Cauchy's strain tensor. Eq. (1-21) denotes the strain-displacement relationship.

$$\epsilon = Du \tag{1-21}$$

where D and u are differential matrix operator and displacements, respectively. Hence, the system of differential equations can be defined according to Eq. (1-22) which is known as Lame's displacement equation.

$$D^T EDu + \overline{P_V} = 0 \tag{1-22}$$

The boundary conditions can be defined by the equilibrium condition on the boundary of the specified body as expressed in Eq. (1-23) and Eq. (1-24)

$$A^T \sigma = \overline{P_S} \tag{1-23}$$

$$u = \bar{u} \tag{1-24}$$

where A^T and $\overline{P_S}$ are correspondingly a transformation matrix and surface tractions. Assuming stresses statically satisfy the equilibrium relation specified in Eq. (1-19) and boundary conditions in Eq. (1-23), the result, in integral form of these sets of equations, can be expressed by Eq. (1-25) [103].

$$\int_V \delta u^T (D^T \sigma + \overline{P_V}) dv + \int_{S_p} \delta u^T (A^T \sigma - \overline{P_S}) dS = 0 \tag{1-25}$$

Performing an integration based on Gauss' integral theorem yields Eq. (1-26).

$$\int_V \delta \varepsilon^T \sigma \, dv - \int_V \delta u^T \overline{P_V} \, dv - \int_{S_p} \delta u^T \overline{P_S} \, dS = 0 \tag{1-26}$$

Eq. (1-26) signifies that the sum of virtual work of internal forces (W_{int}) and external forces (W_{ext}) is equal to zero. This equation represents the principal of virtual work that is referred to as the weak form of the differential equation. By substituting Eqs. (1-20) and (1-21), the work of the internal nodal force can be written as Eq. (1-27). This equation denotes the internal nodal force for the structural elements in FEA.

$$\int_{V^e} (D\delta u)^T EDu \, dv - \int_{V^e} \delta u^T \overline{P_V} \, dv - \int_{S_p^e} \delta u^T \overline{P_S} \, dS \; - \; \delta V^{eT} P^e = 0 \tag{1-27}$$

where V^e and P^e are the element displacement vector and internal nodal force, respectively, where the superscript e indicates an individual element. A contracted form of Eq. (1-27) is given by Eq. (1-28). Therefore, displacements can be approximated using this equation.

$$\delta W_{int}^{e} - \delta W_{ext}^{e} = 0 \tag{1-28}$$

For solving an elasticity problem, the element equations expressed in Eq. (1-27) are assembled for all the structural elements existing in the model in the matrix form in such a way that the sum of the internal and external work is zero (i.e., the total work done in the structure is the sum of the work of each individual element exist in the problem), which can be written as Eq. (1-29) and Eq. (1-30).

$$\delta W_{int} = \sum_{e} \delta W_{int}^{e} \tag{1-29}$$

$$\delta W_{ext} = \sum_{e} \delta W_{ext}^{e} \tag{1-30}$$

1.6.3 Thermal Numerical Analysis

The fundamentals for thermal numerical analysis are based on heat conductions (i.e., conduction of heat through solids). The mathematical expression of this phenomena was provided by "Fourier's Law". This expression indicates that the rate of heat transfer per unit area is proportion to the temperature gradient in the normal direction as described by Eq. (1-31).

$$q = -kA\frac{dT}{dx} \tag{1-31}$$

where q, A and k are rate of heat transfer or heat flux, cross section area and thermal conductivity coefficient, respectively. dT and dx are correspondingly a temperature difference and the increment of length that heat is transferred along. A schematic for a one-dimensional steady-state heat conduction element is shown in Fig. 1-4.

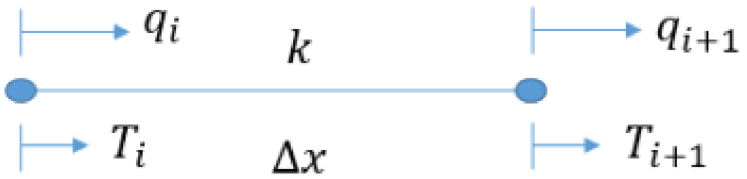


Figure 1-4 Linear heat conduction element schematic.

Using Eq. (1-31) the heat transfer in node i and $i+1$ can be written as Eqs. (1-32) and (1-33), respectively.

$$q_i = -\frac{kA}{\Delta x}[T_{i+1} - T_i] \tag{1-32}$$

$$-q_{i+1} = \frac{kA}{\Delta x}[T_{i+1} - T_i] \tag{1-33}$$

These equations can be written in matrix form as Eqs. (1-34) and (1-35).

$$\frac{kA}{\Delta x}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}\begin{bmatrix} T_i \\ T_{i+1} \end{bmatrix} = \begin{Bmatrix} q_i \\ q_{i+1} \end{Bmatrix} \tag{1-34}$$

$$[K][T] = \{F\} \tag{1-35}$$

where K denotes the conductivity matrix (or stiffness matrix in analogy to structural analysis); T and F are correspondingly temperature and load vector. By assuming perfect thermal isolation as the boundary conditions i.e. adiabatic condition and conservation of energy, the equilibrium equation can be written in the form of Eq. (1-36). Eventually, the temperature gradient can be calculated by solving two equations with two unknowns.

$$q_i = -q_{i+1} \tag{1-36}$$

Having presented the concept of a one-dimensional heat conduction element, the same concept can be utilized in any discretized system to form the global matrices and linking the heat flow throughout the entire structure. Fig. 1-5 illustrates a discretization of three one-dimensional elements in series. The global matrix can be written as Eq. (1-37) for this discretization.

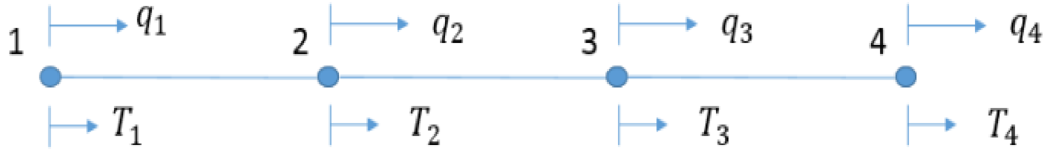


Figure 1-5 schematic of a discretized linear heat conduction element.

$$\frac{kA}{\Delta x}\begin{bmatrix} 1 & -1 & 0 & 0 \\ -1 & 1+1 & -1 & 0 \\ 0 & -1 & 1+1 & -1 \\ 0 & 0 & -1 & 1 \end{bmatrix}\begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \end{bmatrix} = \begin{Bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{Bmatrix} \tag{1-37}$$

The load vector reflects the internal conservation, and therefore, q_2 and q_3 are zero. Consequently, by eliminating q_1 and q_4, Eq. (1-37) can be reduced to Eq. (1-38). Temperatures T_1 and T_2 are known from the boundary conditions. Therefore, the internal temperatures T_2 and T_3 can be defined by solving this set of equation, respectively.

$$\begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix}\begin{bmatrix} T_2 \\ T_3 \end{bmatrix} = \begin{Bmatrix} T_1 \\ T_4 \end{Bmatrix} \tag{1-38}$$

If the boundary conditions surrounding a solid body are in a steady-state condition, the solid body is in thermal equilibrium. However, if the surrounding boundary conditions are subject to a time-dependent change, the solid body temperature either increases or decreases until an equilibrium is achieved. This process is known as transient or time-dependent analysis. In this type of numerical analysis, an equilibrium condition is reached at each time step defined in the analysis and a temperature distribution is calculated for each instance. Therefore, Eq. (1-39) is the governing equation for the transient thermal analysis.

$$\rho C \frac{dT}{dt} - \frac{d}{dx}\left[k(x)\frac{dT}{dx}\right] = Q(x) \tag{1-39}$$

where, in contrast to steady-state thermal analysis, two additional parameters are presented (i.e., ρ and C are density and specific heat, respectively). Parameter t denotes the time step defined for the analysis. This analysis type indicates how boundary conditions can be time-depended during the analysis. The first and second term on the left-hand side of Eq. (1-39) requires an initial boundary and spatial condition specification to be defined, respectively, which enables performing a transient thermal analysis. Eq. (1-39) can be written as Eq. (1-40) after discretizing the finite element analysis.

$$[C(x,t)]\left\{\frac{dT(t)}{dt}\right\} + [k(x)]\{T(t)\} = \{F(x,t)\} \tag{1-40}$$

where $T(t)$ and $\{F(x,t)\}$ are correspondingly time dependent temperature and heat flux vectors. $[C(x,t)]$ is the capacitance matrix that contains density and specific heat parameters. Considering n as the number of time steps in a transient thermal analysis, Eq. (1-40) can be written as Eq. (1-41).

$$\left(\frac{1}{\Delta t_n}[C(x,t)] + [k(x)]\right)\{T(t)\}_n = \{F(x,t)\}_n + \left(\frac{1}{\Delta t_n}[C(x,t)]\right)\{T(t)\}_{n-1} \tag{1-41}$$

Temperature at the $(n-1)^{th}$ time step is always a known parameter for identifying the temperature at the n^{th} time step, and Eq. (1-39) shall be solved at each time step calculating the temperature distribution in the numerical model. This method is a powerful solution that enables solving linear and non-linear problems. The only difference between the linear and non-linear problems is that the capacitance matrix and conductivity matrix are also temperature dependent in non-linear analysis. Consequently, both temperature and time signify the non-linear transient thermal analysis which can be written in Eq. (1-42).

$$[C(x,t,T)]\left\{\frac{dT(t)}{dt}\right\} + [k(x,T)]\{T(t)\} = \{F(x,t,T)\} \tag{1-42}$$

1.6.4 Structural-Thermal and Structural-Electrical FEA

Modeling interfacial effects and thermal contact problems is typically a challenging task. In general, thermal and electrical contact problems are considered structural-thermal and structural-electrical problems, respectively, since interfacial effects firmly depend on structural contact and deformations, i.e., deformations change the interfacial contact in terms of size and distribution, and hence, affect the heat transfer and electrical conductivity across the contact surface. Simple contact problems can be solved using a load transfer method (i.e., weak coupling method), that is, a structural analysis is performed first, and then the deformed structure is used for performing a thermal or electrical analysis. In this method two iterations are needed to be performed for solving each stage of the analysis. However, for complicated problems involving factors such as temperature dependent materials and thermal expansion, it is recommended to use a strong coupled field numerical analysis to solve the problem where one iteration is required to be performed. Compared to an uncoupled ‘single physics’ approach, a strong coupled structural-thermal and structural-electrical numerical analysis creates a more comprehensive solution, combining structural, thermal and electrical effects simultaneously, leading to results that more closely resemble reality.

The structural and thermal field interaction is captured by means of coefficient of thermal expansion. Eqs. (1-43) and (1-44) show the coupled structural and thermal field.

$$\{\varepsilon\} = [\mathrm{D}]^{-1}\{\sigma\} + \{\alpha\}\Delta T \tag{1-43}$$

$$S = \{\alpha\}^{\mathrm{T}}\{\sigma\} + \frac{\rho C_{\mathrm{P}}}{T_0}\Delta T \tag{1-44}$$

where σ, ε, α and D are the total stress vector, strain vector, coefficient of thermal expansion vector and elastic stiffness matrix, respectively. ΔT and S are correspondingly the temperature difference and entropy density. T_0, C_P and ρ are the absolute temperature, specific heat and material density, respectively. Substituting Eq. (1-44) into Eq. (1-43) the following constitutive equation can be derived which is the foundation of the structural-thermal element.

$$\begin{bmatrix} \mathrm{K_{UU}} & \mathrm{K_{UT}} \\ 0 & \mathrm{K_{TT}} \end{bmatrix} \begin{Bmatrix} \mathrm{U} \\ \mathrm{T} \end{Bmatrix} + \begin{bmatrix} \mathrm{C_{UU}} & 0 \\ -T_0\mathrm{K_{UT}^T} & \mathrm{C_{TT}} \end{bmatrix} \begin{Bmatrix} \dot{\mathrm{U}} \\ \dot{\mathrm{T}} \end{Bmatrix} + \begin{bmatrix} \mathrm{M_{UU}} & 0 \\ 0 & 0 \end{bmatrix} \begin{Bmatrix} \ddot{\mathrm{U}} \\ \ddot{\mathrm{T}} \end{Bmatrix} = \begin{Bmatrix} \mathrm{F} \\ \mathrm{Q} \end{Bmatrix} \tag{1-45}$$

where $\mathrm{K_{UU}}$, $\mathrm{K_{TT}}$, and $\mathrm{K_{UT}}$ are the structural stiffness, thermal conductivity and thermo-elastic coupling, respectively. $\mathrm{C_{TT}}$, $\mathrm{C_{UU}}$ and $\mathrm{M_{UU}}$ are correspondingly the specific heat damping, structural damping, and mass. Also, F, Q, U and T denote for force or pressure i.e. structural boundary conditions, heat generation displacement and temperature, respectively.

1.6.5 Element Place in Finite Element Analysis

As mentioned earlier, the engineering problem shall be discretized for performing FEA. Elements provide means for discretizing the geometry of engineering problems which carry the information required for solving the problem (i.e., the constitutive equation). Depends on the shape of the element, each element is constructed from at least one or higher number of nodes. In general four shapes are possible that can be considered for an element (e.g., point, line, area or volume). A "point" element is typically defined by only one node (e.g., mass elements). A "line" and "area" element are typically defined with two and a minimum of three nodes, respectively. Also, a "volume" element can be defined with a minimum of four nodes (i.e., tetrahedral element). Depending on the type of numerical analysis (i.e., linear or non-linear), element nodes can be connected together by a line or arc. Generally, using a nonlinear element increases the accuracy of the result, however, increases the time required for solving the problem, consequently, increases the cost associated with the analysis. Element degrees of freedom determine the discipline or the physics that the element is applicable i.e. structural, thermal or electrical. The numerical model response can be characterized by selecting an element type along with necessary degrees of freedom (e.g., displacement, rotations, temperatures and pressures). Also, the element types utilize various material properties (i.e., Young's modulus, density, coefficient of thermal expansion, thermal and electrical conductivity), which can also be a function of temperature. The boundary conditions can be applied to the numerical model through elements e.g. surface, body inertial loads are element loads which are always associated with a specific element even if the input is located at the nodes. In contrast, nodal loads are defined at the nodes which are not related to the element directly and are linked with the degrees of freedom.

Another important aspect of the element and nodes in numerical analysis is that these numerical components enable extracting the solution from numerical analysis. Element solution and nodal solution are two different methods for extracting the solution from the numerical model. Generally, the element solutions are displayed within individual elements which are not averaged solution i.e. the distribution of solution is unique for an individual element. Having a good understanding of local nodes is very important for understanding element solution. In contrast, nodal solutions are calculated at each global node and it is in averaged form i.e. the calculated result at a global node is the average of all the local nodes values sharing that global node. The nodal solution from the analysis can be consist of the degrees of freedom solutions e.g. nodal displacement, temperature and pressure, and reaction solution which can be calculated at each constrained node e.g. forces and heat flows in the context of structural and thermal analysis, respectively.

1.6.6 Contact Overview in Finite Element Analysis

Generally, contact problems are highly nonlinear in nature, therefore, require significant computing resources. Conventionally, one side of the contact is established as the “Target” surface and the other side as “Contact”. Defining target and contact elements enable tracking the kinematics of the deformation process in the context of structural analysis. These elements share a real constant set while making up a contact pair. It is very important to mention that the contact element is constraint against penetrating the target surface, however, target elements can penetrate into the contact surface.

Rigid-to-flexible and flexible-to-flexible are two types of contact problems that can be defined in FEA. If one or two contacting surfaces are rigid, the contact problem shall be treated as rigid-to-flexible (i.e., if a soft material got in contact with a hard material), it is appropriate that contact be treated as rigid-to-flexible. On the other hand, flexible-to-flexible contact shall be defined in the case when two hard materials come into contact with each other. In contact problems, the target surface shall be the rigid surface and the contact surface is always the deformable surface. In the case of flexible-to-flexible contacts, the choice of designating contact and target surface may affect the amount of penetration and consequently affect the accuracy of the results. Therefore, it is recommended to follow a practical guideline for choosing the contact and the target surfaces. For example, if a convex surface got into contact with a flat or a concave surface, it is recommended that the flat or concave surface be selected as the target surface. Also, if a surface is stiffer than the other mating surface, the stiffer surface shall be chosen as the target surface. In the case of having fine mesh or coarse mesh at contact surfaces, it is recommended that the surface with the coarse mesh should be selected as the target surface. In addition, if two surfaces with a different order of elements (i.e., low and high order of elements) got into contact with each other, the surface with the low-order elements shall be selected for the target surface.

Besides these considerations, contacts are important since they should address different physics effects simultaneously (i.e., pressure, thermal and electrical conductance and electrical current at the contact area). Generally, multiphysics numerical analysis needs interacting with two sides of the contact or interacting between a surface and its surrounding. This can be achieved by utilizing a surface-to-surface or a node-to-surface contact element in conjunction with a structural-thermal solid element or a pure thermal element that enables modeling structural deformations and heat transfer which is occurring at the contact surface. A schematic of a thermal contact is shown in Fig 1-6.

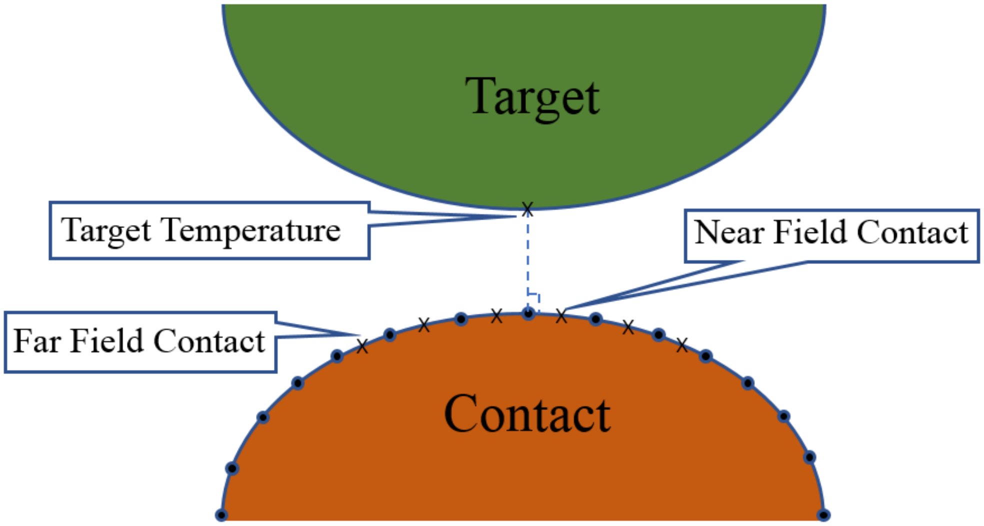


Figure 1-6 Near field structural- thermal contact schematic.

Modeling heat conduction at the contact location can be achieved by using a thermal contact conductance (TCC) coefficient that is a constant of the contact element as shown in Eq. (1-46).

$$q = TCC \times (T_t - T_c) \tag{1-46}$$

where q and TCC denote for the heat flux and thermal contact conductance coefficient, respectively. T_t and T_c are correspondingly temperature of the contact at target and contact surfaces. It is important to mention that the TCC can be defined as a function of temperature. This type of contact provides a means for simulating the interfacial thermal resistance (i.e., Kapitza resistance), which is a critical aspect when calculating the effective thermal conductivity of filler modified polymers.

Also, modelling electrical conduction at contact can be done by utilizing a electrical contact conductance as shown in Eq. (1-47).

$$J = ECC \times (V_t - V_c) \tag{1-47}$$

where J and ECC are current density for a potential degree of freedom, i.e. Voltage, and electrical contact conductance coefficient, respectively. V_t and V_c denote for voltages at the target and contact surfaces. The ECC also can be a function of temperature.

1.6.7 Parametric Finite Element Analysis

Generally, the term ‘parametric’ refers to tasks or processes that permit changing input conditions. In recent years, parametric features are increasingly receiving attention and becoming more common in FEA. A key benefit of performing a parametric study in FEA is the ability to evaluate quickly the effect of design changes in the model. Parameters, e.g. geometry dimensions, mesh criteria and boundary conditions, can be defined as variables in a numerical model. This enables applying randomness to each parameter by connecting the variable to an appropriate random number generator. FEA scripting (i.e., FEA coding) is a powerful tool that can be used for connecting random variables in the FEA with random number generators. This method provides more flexibility and reduces time required for performing SFEA significantly and eliminates modifying parameter in the model manually. Also, this method can be used in Design of Experiment (DOE) studies to calculate response surfaces associated with input and output parameters. A parametric modeling approach was employed in this thesis research in order to perform SFEA and predict the properties of the filler modified polymers.

1.6.8 Stochastic Finite element analysis

In light of all the information provided for the stochastic simulation, finite element analysis and concept of parametric FEA in earlier sections, this section introduces a novel concept of SFEA framework which is the main engine and robust solution that enables performing material characterization of filler modified composites. This novel framework is a combination of all the aforementioned concept. It is very important to mention that developing a fully customized parametric FEA framework requires a significant scripting contribution that enables integrating stochastic simulation in FEA successfully. In other words, all steps involved in the FEA shall be automated, which is explained further in detail.

It is important to mention that the accuracy of SFEA framework is a function of randomness (i.e., how true randomness is applied to input parameters) [104]. One of the important input parameters involved in the framework is the RVE size, which needs to be large enough to resemble true randomness in terms of particle distribution and dispersion that will yield accurate effective properties. Therefore, identifying a minimum RVE size that meets this condition is critical. Researchers have developed different methods for calculating a minimum RVE size (e.g., in the context of mechanical properties) [105]. A well-known criterion is Hill's condition [106]. Also, Karimi et al. defined the minimum required area of a statistical RVE (SRVE) and evaluated its randomness. These and similar methods can be employed to ensure an appropriate RVE size.

In performing an FEA, multiple steps shall be taken into consideration which starts with generating a geometry. Since in this doctoral research it was assumed that the fillers are very well distributed and dispersed within a polymer, a proper pseudorandom number generator (i.e., the mathematical programming environment MATLAB (MathWorks, Natick, MA, USA)) was utilized that enabled generating random number based on a uniform probability distribution which indicates the X, Y and Z coordinates of fillers within the matrix. However, other probability distributions i.e. normal or custom probability distributions can be used for representing any type of clustering or different morphologies of the particulate composite which was not part of this research study. This method of generating geometry enabled incorporating filler alignment for particle shape that alignment matters e.g. cylindrical or disk shape particles.

The second step for performing an FEA is applying material properties. As mentioned in earlier sections, there is no absolute event in the reality and considering this concept in material properties, it is inappropriate to consider a value for representing a material property (i.e., modulus of elasticity and thermal or electrical conductivity). Therefore, it is recommended to utilize a normal or log-normal probability distribution for generating random numbers that represent the material properties. This method enables applying true randomness in material properties.

The third step of the FEA is generating coordinate systems that define how material properties are applied. Generally, in the case of defining isotropic material, one coordinate system is satisfactory for generating the numerical model, however, for defining orthotropic materials each filler located within the matrix requires its own coordinate system (e.g., creating a material property for carbon nanotubes). Coordinate systems utilize filler coordinates for locating itself properly within the matrix.

The fourth step of the FEA is defining contact which is very important in developing a framework that predicts the material properties of filler modified polymers. In this research study, it was assumed that there is no collision between particles located within the matrix. Therefore, besides generating random numbers, the pseudorandom number generator performs collision detection and doesn't allow particles to collide with each other. A script was developed that enabled reading particle coordinates and applying contact between particles based on their distance from each other. A flexible-to-flexible type of contact was used for creating contact between particle-to-particles. In the case of creating particle-to-particle contact, since particles are considered as stiff material, it is not very important that which particle surface shall be selected as contact and target surface. However, in the case of particle-to-matrix contact, it was decided to choose particles as the contact surface and the matrix as target surface because of their convex and concave shape, respectively.

The fifth step of the FEA is mesh generation which is one of the important steps in the process that enables increasing the accuracy of material properties prediction. Mesh generation was done based on the analysis type and the element type required for performing the numerical analysis. A convergence study was performed to ensure that the mesh generated for the numerical model is dense enough that provides an acceptable accuracy in final results. After performing the convergence study, all the mesh criteria were stored in the numerical model as parameters so the same mesh settings can be used in all the iterations in the stochastic simulation.

The sixth step of the FEA involves applying the boundary conditions to the numerical model. The parametric FEA concept was used for defining boundary conditions values which facilitates applying the stochastic principles to the framework in terms of recreating the numerical model. This method enabled reading values of boundary conditions in each iteration and applying them to the numerical model. "Face load" and "Nodal load" methods were used for applying boundary conditions (e.g., applying face constraint, uniform face displacement and nodal constraint to the numerical model). A scripting method was used that enabled applying these boundary conditions by using a unique topology ID and locating them properly in the model in each iteration.

Finally, the last step of the FEA is extracting the solution from the numerical model. It was decided to use a "nodal solution" method for extracting the results from the numerical model where the extracted value for each global node is the average value of all local node values that share the global node.

All the information provided in this section which explained how to perform one iteration only, however, in order to combine the stochastic simulation with FEA and consequently applying stochastic principles, the FEA process shall be repeated many times until the standard deviation of the extracted result from FEA is below the specific threshold that is already defined for the framework. After achieving the threshold, this value represents the material property that the framework was aimed for it. Fig. 1-7 illustrates a schematic of the SFEA framework in the context of a thermal property i.e. effective thermal conductivity.

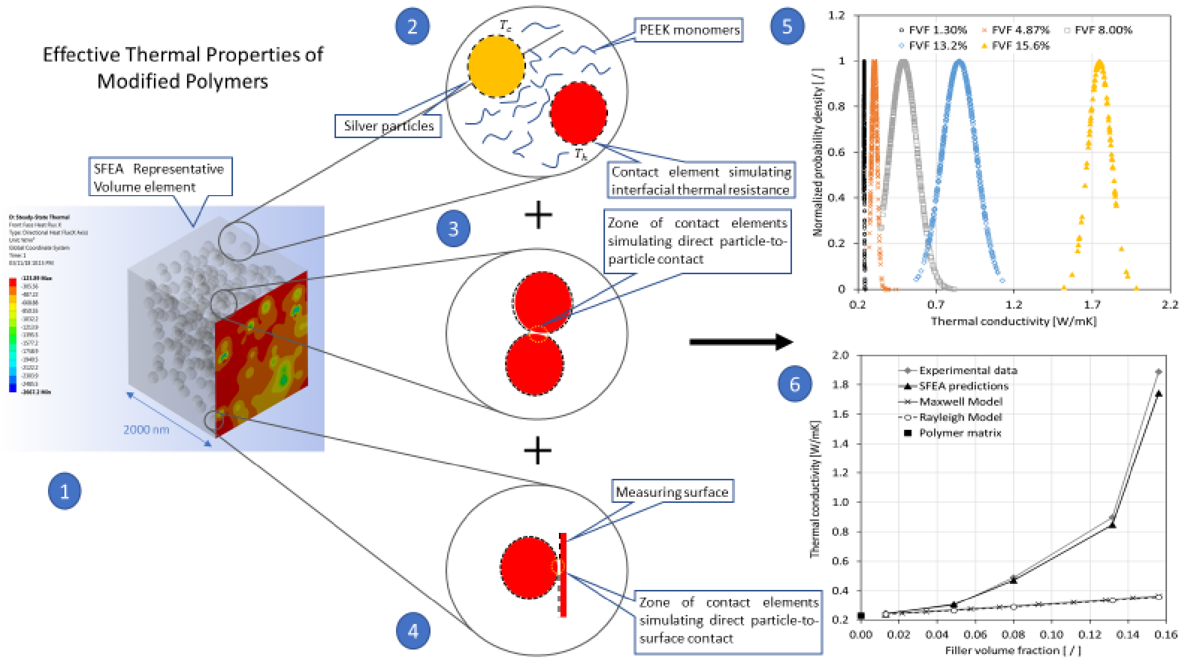

Effective Thermal Properties of Modified Polymers
SFEA Representative Volume element
2000 nm
1
2
T_c
T_h
PEEK monomers
Silver particles
Contact element simulating interfacial thermal resistance
3
Zone of contact elements simulating direct particle-to-particle contact
4
Measuring surface
Zone of contact elements simulating direct particle-to-surface contact
+
+
5
FVF 1.30%
FVF 4.87%
FVF 8.00%
FVF 13.2%
FVF 15.6%
Normalized probability density [/]
Thermal conductivity [W/mK]
0
0.2
0.4
0.6
0.8
1
1.2
0.2
0.7
1.2
1.7
2.2
6
Experimental data
SFEA predictions
Maxwell Model
Rayleigh Model
Polymer matrix
Thermal conductivity [W/mK]
Filler volume fraction [/]
0.2
0.4
0.6
0.8
1.0
1.2
1.4
1.6
1.8
2.0
0.00
0.02
0.04
0.06
0.08
0.10
0.12
0.14
0.16

Figure 1-7 SFEA framework schematic for predicting thermal property. Step 1 represents the numerical model geometry and mesh. Steps 2 to 4 illustrate different types of contacts i.e. particle-to-matrix, particle-to-particle and particle-to-RVE surface. Step 5 and 6 show applying stochastic principals and final thermal property results, respectively.

1.7 Knowledge Gap

Besides the aforementioned shortcomings of analytical models (limited ability and versatility to predict effective properties of filler modified polymers), conventional numerical approaches lack the ability to impart true randomness in the model generation by implementing stochastic principles. These shortcomings give rise to novel modeling methods, that is, the concept of a stochastic FEA framework, which the subject matter of this research.

1.8 Research Hypothesis and Objectives

This doctoral research work is founded on the following hypothesis: A numerical modeling framework is a feasible and accurate means for material designers to effectively explore the design space for a wide variety of particulate polymer composites. The study therefore investigates methods for creating a time and cost-effective numerical framework, with the scope of predicting the mechanical, thermal and electrical properties of particulate polymer composite systems using a stochastic simulation. This doctoral research focused on solid filler particles with spherical and cylindrical geometry since many filler particles either have such shapes or can reasonably be approximated by these shapes. Specifics to modeling mechanical, thermal and electrical properties are addressed in the following subsections.

The main contribution of this doctoral work is the development and validation of a numerical multiple physical modeling framework that predicts the mechanical, thermal and electrical properties of composites with randomly distributed and aligned filler particles. Moreover, the modeling framework was employed to investigate the effect of particle alignment on material properties. The accuracy of results obtained by the modeling framework was tested within the context of this research.

Based on the research hypothesis the following objectives for this research study were derived:

i) Develop a stochastic FEA framework for predicting mechanical properties, specifically the effective modulus of elasticity, Poisson's ratio and coefficient of thermal expansion of particulate polymer composites.

ii) Expand the SFEA framework to predict thermal properties of particulate polymer composites, i.e., their effective thermal conductivity.

iii) Adapt the SFEA framework to assess the electrical properties of polymer composites with electrically conductive fillers, (i.e., predict their effective electrical conductivity, percolation threshold, piezoresistivity as the result of mechanical strain). In addition, explore the effective electrical conductivity change as the result of a change in temperature.

In the following, short summaries are given on the modeling activities related to the above research objectives. Note that further context in terms technical literature and greater detail on the modeling work is included in subsequent chapters, which are based on published manuscripts.

1.8.1 Mechanical Characterization of Filler Modified Polymers

Mechanical properties typically need to meet the requirements for a specific design, and knowledge of associated parameters is therefore of great importance for the design process. A variety of manufacturing routes is available for filler modified polymers that may achieve required properties, however, it is currently challenging to identify and select appropriate processes that deliver the desired properties for specific components [4]. The reason for this shortcoming is the complex interaction between manufacturing parameters and resulting mechanical properties [96]. This difficulty is compounded by uncertainties related to the effects that the composite morphology has on mechanical properties. This research aims at elucidating the latter. For example, many applications require mechanical properties enhancements only in a specific direction while in other directions, lower mechanical properties are acceptable or even desirable (e.g., to minimize the amount and thus cost of fillers [5-6]). Therefore, the production of anisotropic mechanical properties is important. Randomly distributed but aligned particles facilitate the creation of such conditions, yet, it is unclear how a specific filler and probable deviations from perfect alignment affect desired properties [26-28].

The first part of the doctoral research focused on developing a numerical framework based on Monte Carlo simulations to study composites that primarily incorporate randomly distributed spherical particles. The modeling work allowed for predicting the elastic properties, employing a linear static structural analysis approach for different filler volume fractions. The technical literature was consulted to select practicable material compositions [97-98]. It is assumed that particles are perfectly bonded to the polymer matrix. Given the nature of Monte Carlo simulation, multiple numerical samples needed to be analyzed for a specific material system. The study did not only compute average values but also assessed uncertainties and scatter that may result in terms of material properties. To ensure that the modeling approach yields realistic results, the numerical results were contrasted and discussed in light of analytical and experimental data from the technical literature.

1.8.2 Thermal Characterization of Filler Modified Polymers

The second part of this doctoral research focused on expanding the numerical framework to enable the thermal characterization on isotropic and anisotropic polymer composites modified with randomly distributed spherical and cylindrical particles. Composites with anisotropic thermal properties offer great potential for applications that require a high thermal conductivity in one direction while permitting medium or low thermal conductivity in other directions [99-100]. Since making such composite materials requires significant experimental effort, developing material design simulation tools is considered highly valuable in order to save development cost and time [9]. The basis for the developed numerical framework is steady-state thermal analysis (i.e., material thermal properties were considered time-independent). The modeling approach calculates the effective thermal conductivity by performing multiple numerical analyses in the context of the Monte Carlo simulation.

Kapitza resistance, also known as interfacial thermal resistance, is an important effect that strongly influences interfacial layer resistance to thermal flow [94]. Heat conduction is associated with two physical phenomena: (i) transferring kinetic energy between neighboring atoms in the form of movement or vibration, and (ii) motion of electrons causing a transfer energy from a hot zone to a cold zone. Since these two phenomena are impeded at the contact of fillers and matrix, it causes a significant temperature drop and reduction in heat transfer. Kapitza effects are captured in this modeling approach. Within the FEA analysis portion of the model, surface-to-surface elements were employed to simulate thermal resistance between particles and polymer. These elements capture properties such as thermal conductance, temperature and heat flux due to conduction, allowing Kapitza effects to be included in the numerical model. Measuring actual thermal resistance effect at material interfaces is a challenging experimental endeavor, and therefore, information needed for the modeling work was adopted from the technical literature. A comparison was performed between numerical framework results and analytical as well as experimental results obtained from literature in order to validate the developed numerical framework.

1.8.3 Electrical Characterization of Filler Modified Polymers

The final part of this research focused on adapting the SFEA framework for predicting electrical properties (i.e., the effective electrical conductivity and the percolation behavior of polymer composites with randomly distributed and dispersed spherical electrically conductive particles). Also, the effects of changing temperature and applying mechanical strain on effective electrical conductivities were explored. An analytical description was adopted that enables defining the onset and transfer of electrical charge for particle to polymer, particle to particle, and particle to RVE surface based on proximity. Below a specific threshold distance between particles, i.e. the tunneling distance, which was adopted from literature, transfer of electrical charge occurred. Finally, performing the numerical analysis enabled calculating effective electrical conductivity, and by checking whether electricity is transferred from one side of the RVE to other the percolation threshold was determined.

1.9 Thesis Organization

This doctoral thesis is composed of the following chapters beyond the present introductory chapter:

- Chapter 2 focuses on developing the SFEA framework for predicting mechanical properties of filler modified polymer composites. In the first section of this chapter, a semi-automated approach was used for establishing and evaluating the SFEA framework also exploring the validity of the research hypothesis. In the second section of this chapter, an advanced, automated version of the SFEA framework was developed for predicting mechanical properties comprehensively (i.e., the effective modulus of elasticity, Poisson's ratio and coefficient of thermal expansion were simulated).
- In Chapter 3, the SFEA framework was developed further to enable the automated

computation of thermal properties of particulate polymer composites by considering Kapitza resistance effects as well as direct particle-to-particle heat transfer.

- In Chapter 4, the study from Chapter 3 expanded to explore the effect of filler alignment on effective thermal conductivity, thus extending the capabilities of the modeling framework.
- In Chapter 5, the development of the SFEA framework culminates in the prediction of the electrical properties of polymer composites with electrically conductive fillers. The effective electrical conductivity, electrical percolation threshold, piezoresistivity effects, and electrical conductivity changes due to changing temperature were studied.
- In Chapter 6 concludes this thesis by summarizing the completed work while recognizing research accomplishments and contributions to the current field of inquiry.

1.10 References

[1] Beret, S.; Prausnitz, J. Densities of liquid polymers at high pressure. Pressure-volume-temperature measurements for polythylene, polyisobutylene, poly(vinyl acetate), and poly(dimethylsiloxane) to 1 kbar. Macromolecules. 1975, 8, 536–538, DOI: 10.1021/ma60046a028.

[2] Manfredi, L.; Rodriguez, E.; Wladyka-Przybylak, M.; Vazquez, A. Thermal degradation and fire resistance of unsaturated polyester modified acrylic resins and their composites with natural fibres. Polym. Degrad. Stab. 2006, 91, 255-261, DOI: 10.1016/j.polymdegradstab.2005.05.003.

[3] Drury, J.; Mutsaers, C.; Hart, C.; Matters, M.; de Leeuw D. Low-cost all-polymer integrated circuits. Appl. Phys. Lett. 1998, 73, 108-110, DOI: 10.1063/1.121783.

[4] Sheng, N.; Boyce, M.; Parks, D.; Rutledge, G.; Abes, J. Multiscale micromechanical modeling of polymer/clay nanocomposites and the effective clay particle. Polymer. 2004, 45, 487-506, DOI: 10.1016/j.polymer.2003.10.100.

[5] Chen, H.; Wang, M.; Lin, Y.; Chan, C.; Wu, J. Morphology and mechanical property of binary and ternary polypropylene nanocomposites with nanoclay and CaCo3 particles. J. Appl. Polym. Sci. 2007, 106, 3409–3416, DOI: 10.1002/app.27017.

[6] Ma, P.; Siddiqui, N.; Marom, G.; Kim, J. Dispersion and functionalization of carbon nanotubes for polymer-based nanocomposites: A review. Composites, Part A Composites. 2010, 41, 1345-1367, DOI: 10.1016/j.compositesa.2010.07.003.

[7] Akcora, P.; Liu, H.; Kumar, S.; Moll, J.; Brian Y.; Benicewicz, C.; Schadler, L.; Acehan, D.; Panagiotopoulos, A.; Pryamitsyn, V.; Ganesan, V.; Ilavsky, J.; Thiyagarajan, P.; Colby, R.; Douglas, J. Anisotropic self-assembly of spherical polymer-grafted nanoparticles, Nat. Mater. 2009, 8, 354–359, DOI: 10.1038/nmat2404.

[8] Li, Z.; Okamoto, K.; Ohki, Y.; Tanaka, T. Effects of nano-filler addition on partial discharge resistance and dielectric breakdown strength of Micro-Al2O3Epoxy composite. IEEE Trans. Dielectr. Electr. Insul. 2010, 17, 653-661, DOI: 10.1109/TDEI.2010.5492235.

[9] Meincke, O.; Kaempfer, D.; Weickmann, H.; Friedrich, C.; Vathauer, M.; Warth, H. Mechanical properties and electrical conductivity of carbon-nanotube filled polyamide-6 and its blends with acrylonitrile/butadiene/styrene. Polymer, 2004, 45, 739-748, DOI: 10.1016/j.polymer.2003.12.013.

[10] Valavala, P.; Odegard, G.; Modeling techniques for determination of mechanical properties of polymer Nanocomposites. Department of Mechanical Engineering, Michigan Technological University, 1400 Townsend Drive, Houghton, MI 49931, USA, 2005

[11] Odegard, G.; Clancy, T.; Gates, T. Modeling of the mechanical properties of nanoparticle/polymer composites. Polymer, 2005, 46, 553-562, DOI: 10.1016/j.polymer.2004.11.022.

[12] Liu, Y.; Chen, X. Evaluations of the effective material properties of carbon nanotube-based composites using a nanoscale representative volume element. Mech. Mater. 2003, 35, 69–81, DOI: 10.1016/S0167-6636(02)00200-4.

[13] Prasad, K.; Das, B.; Maitra, U.; Ramamurty, U.; Rao, C. Extraordinary synergy in the mechanical properties of polymer matrix composites reinforced with 2 nanocarbons. PNAS. 2009, 106, 13186-13189, DOI: 10.1073/pnas.0905844106.

[14] Bewlay, B.; Jackson, M.; Lipsitt, H. The balance of mechanical and environmental properties of a multielement niobium-niobium silicide-basedIn Situ composite. Metall. Mater. Trans. A. 1996, 27, 3801–3808, DOI: 10.1007/BF02595629.

[15] Rana, D.; Lee, C.; Cho, K.; Lee, B.; Choe, S. Thermal and mechanical properties for binary blends of metallocene polyethylene with conventional polyolefins. J. Appl. Polym. Sci. 1998, 69, 2441–2450, DOI: 10.1002/(SICI)1097-4628(19980919)69:12<2441::AID-APP15>3.0.CO;2-%23.

[16] Sathe, S.; Devi, S.; Srinivasa, G.; Rao, K. Relationship between morphology and mechanical properties of binary and compatibilized ternary blends of polypropylene and nylon 6. J. Appl. Polym. Sci. 1996, 61, 97–107, DOI: 10.1002/(SICI)1097-4628(19960705)61:1<97::AID-APP11>3.0.CO;2-X.

[17] Eshelby, J. The determination of the elastic field of an ellipsoidal inclusion and related problems. Proc. R. Soc. London, Ser. A. 1957, 241, 561–569, DOI: 10.1098/rspa.1957.0133.

[18] Voigt, W. Ueber die Beziehung zwischen den beiden Elasticitätsconstanten isotroper Körper, Annalen der Physik, 1889, 274, 573–587, DOI: 10.1002/andp.18892741206.

[19] Halpin, J.; Kardos, J. The Halpin-Tsai equations: A review. Polym. Eng. Sci. 1976, 16, 344–352, DOI: 10.1002/pen.760160512.

[20] Hashin, Z.; Shtrikman, S. A variational approach to the theory of elastic behavior of multiphase materials. J. Mech. Phys. Solids, 1963, 11, 127–140, DOI: 10.1016/0022-5096(63)90060-7.

[21] Mori, T.; Tanaka, K. Average stress in the matrix and average elastic energy of materials with misfitting inclusions. Acta Metall. 1973, 21, 571–574, DOI: 10.1016/0001-6160(73)90064-3.

[22] Benveniste, Y. A new approach to the application of Mori–Tanaka's theory in composite materials. Mech. Mater. 1987, 6, 147–157, DOI: 10.1016/0167-6636(87)90005-6.

[23] Christensen, R.; Lo, K. Solutions for effective shear properties of three phase sphere and cylinder models. J. Mech. Phys. Solids, 1987, 27, 315–330, DOI: 10.1016/0022-5096(79)90032-2.

[24] Hill, R. A self-consistent mechanics of composite materials. J. Mech. Phys. Solids. 1965, 13, 213–222, DOI: 10.1016/0022-5096(65)90010-4.

[25] Torquato, S. Effective stiffness tensor of composite media: II. Applications to isotropic dispersions. J. Mech. Phys. Solids. 1998, 46, 1411–1440, DOI: 10.1016/S0022-5096(97)00083-5.

[26] Coleman, J.; Khan, U.; Gun'ko, Y. Mechanical reinforcement of polymers using carbon nanotubes. Adv. Mater. 2006, 18, 689-706, DOI: 10.1002/adma.200501851.

[27] Moniruzzaman, M.; Winey, K. Polymer nanocomposites containing carbon nanotubes. Macromolecules. 2006, 39, 5194–5205, DOI: 10.1021/ma060733p.

[28] Green, M.J.; Behabtu, N.; Pasquali, M.; Adams, W. Nano tubes as polymer. Polymer. 2009, 50, 4979-4997, DOI: 10.1016/j.polymer.2009.07.044.

[29] Liu, J.; Liu, J.; Yang, L.; Chen, X.; Zhang, M.; Meng, F.; Luo, T.; Li, M. Nanomaterial-assisted signal enhancement of hybridization for DNA biosensors: a review. Sensors. 2009, 9, 7343-7364, DOI: 10.3390/s90907343.

[30] Balasubramanian, K.; Burghard, M. Biosensors based on carbon nanotubes, Anal. Bioanal. Chem. 2006, 385, 452, DOI: 10.1007/s00216-006-0314-8.

[31] Fan, S.S.; Chapline, M.; Franklin, N.; Tombler, T.; Cassell, A.; Dai, H. Self-oriented regular arrays of carbon nanotubes and their field emission properties. Science. 1999, 283, 512-514, DOI: 10.1126/science.283.5401.512.

[32] Harris P. Carbon nanotubes and related structures: New materials for the twenty-first century. American Journal of Physics. 2004;72(3):415. DOI: 10.1119/1.1645289.

[33] Dai L. Carbon Nanotechnology: Recent Developments in Chemistry, Physics, Materials Science and Device Applications. 1st ed. Amsterdam: Elsevier; 2006. ISBN-13: 978-0-444-51855-2.

[34] Lin, F.; Bhatia, G.; Ford, J. Thermal conductivities of powder-filled epoxy resins. J. Appl. Polym. Sci. 1993, 49, 1901–1908, DOI: 10.1002/app.1993.070491105.

[35] Agari, Y.; Uno, T. Estimation on thermal conductivities of filled polymers. J. Appl. Polym. Sci. 1986, 32, 5705–5712, DOI: 10.1002/app.1986.070320702.

[36] Cheng, S.; Vachon, R. The prediction of the thermal conductivity of two and three phase solid heterogeneous mixtures. Int. J. Heat Mass Transfer. 1969, 12, 249-264, DOI: 10.1016/0017-9310(69)90009-X.

[37] Hasselman, D.; Johnson, L. Effective Thermal Conductivity of Composites with Interfacial Thermal Barrier Resistance. J. Compos. Mater. 1987, 21, 508-515, DOI: 10.1177/002199838702100602.

[38] Song, Y.; Youn, J. Influence of dispersion states of carbon nanotubes on physical properties of epoxy nanocomposites. Carbon. 2005, 43, 1378-1385, DOI: 10.1016/j.carbon.2005.01.007.

[39] Karasek, L.; Sumita, M. Characterization of dispersion state of filler and polymer-filler interactions in rubber-carbon black composites. J. Mater. Sci. 1996, 31, 281–289, DOI: 10.1007/BF01139141.

[40] Paul, M.; Alexandre, M.; Degee, P.; Henrist, C.; Rulmont, A.; Dubois, P. New nanocomposite materials based on plasticized poly(l-lactide) and organo-modified montmorillonites: thermal and morphological study. Polymer. 2003, 44, 443-450, DOI: 10.1016/S0032-3861(02)00778-4.

[41] Yung, K.; Liem, H. Enhanced thermal conductivity of boron nitride epoxy-matrix composite through multi-modal particle size mixing. J. Appl. Polym. Sci. 2007, 106, 3587–3591, DOI: 10.1002/app.27027.

[42] Miranda, O.; Yang, R.; Alvarado-Gil, J. A crowding factor model for the thermal conductivity of particulate composites at non-dilute limit. J. Appl. Phys. 2013, 114, 064306, DOI: 10.1063/1.4818409.

[43] Lin, Z.; Zhigilei, L. V.; Celli, V. Electron-phonon coupling and electron heat capacity of metals under conditions of strong electron-phonon non equilibrium. Phys. Rev. B: Condens. Matter. 2008, 77, 075133, DOI: 10.1103/PhysRevB.77.075133.

[44] Fan, Z.; Wei, T.; Luo, G.; Wei, F. Fabrication and characterization of multi-walled carbon nanotubes-based ink. J. Mater. Sci. 2005, 40, 5075–5077, DOI: 10.1007/s10853-005-1622-7.

[45] Kordas, K.; Mustonen, T.; Toth, G.; Jantunen, H.; Lajunen, M.; Soldano, C.; Talapatra, S.; Kar, S.; Vajtai, R.; Ajayan, P. Inkjet printing of electrically conductive patterns of carbon nanotubes. Small. 2006, 2, 1021-1025, DOI: 10.1002/smll.200600061.

[46] Small, W.; Panhuis, M. Inkjet printing of transparent electrically conducting single-walled carbon-nanotube composites. Small. 2007, 3, 1500-1503, DOI: 10.1002/smll.200700110.

[47] Wei, T.; Ruan, J.; Fan, Z.; Luo, G.; Wei, F. Preparation of graphene Nano sheet/polymer composites using in situ reduction–extractive dispersion. Carbon. 2007, 45, 2296-2299, DOI: 10.1016/j.carbon.2009.04.030.

[48] Dragoman, M.; Flahaut, E.; Dragoman, D.; Al Ahmad, M.; Plana, R. Writing simple RF electronic devices on paper with carbon nanotube ink. Nanotechnology. 2009, 20, 5203, DOI: 10.1088/0957-4484/20/37/375203/meta.

[49] Cao, Q.; Rogers, J. Ultrathin films of single-walled carbon nanotubes for electronics and sensors: a review of fundamental and applied aspects. Adv. Mater. 2009, 21, 29-53, DOI: 10.1002/adma.200801995.

[50] Avouris, P.; Chen, Z.H.; Perebeinos, V. Carbon-based electronics. Nanoscience and Technology. 2009, 2, 174-184, DOI: 10.1142/9789814287005_0018.

[51] Khabibullin, A.; Zharov, I. Fuel cell membranes based on polymer-modified silica colloidal crystals and glasses: Proton conductivity and fuel cell performance. Mater. Res. Soc. Symp. Proc. 2013, 1502, DOI: 10.1557/opl.2013.568.

[52] Tripathi, B.; Shahi, K. Prog. Organic–inorganic nanocomposite polymer electrolyte membranes for fuel cell applications. Prog. Polym. Sci. 2011, 36, 945-979, DOI: 10.1016/j.progpolymsci.2010.12.005.

[53] Winter, M.; Brodd, R. What are batteries, fuel cells, and supercapacitors. Chem. Rev. 2004, 104, 4245–4270, DOI: 10.1021/cr020730k.

[54] Gary, F. Book review: Polymer electrolytes. Royal Society of Chemistry, Cambridge, 1997, ISBN 0-85404-557-0, DOI: 10.1002/(SICI)1097-0126(199805)46:1<78::AID-PI16>3.0.CO;2-I.

[55] Kim, J.; Kang, P.; Chang Nho Y. Positive temperature coefficient behavior of polymer composites having a high melting temperature. J. Appl. Polym. Sci. 2004, 92, 394–401, DOI: 10.1002/app.20064.

[56] Nikkeshi, S.; Kudo, M.; Masuko, T. Dynamic viscoelastic properties and thermal properties of powder-epoxy resin composite. J. Appl. Polym. Sci. 1998, 69, 2593–2598, DOI: 10.1002/(SICI)1097-4628(19980926)69:13<2593::AID-APP9>3.0.CO;2-5.

[57] Lia, C.; Thostenson, E.; Chou, T. Dominant role of tunneling resistance in the electrical conductivity of carbon nanotube-based composites. Appl. Phys. Lett. 2007, 91, 223114, DOI: 10.1063/1.2819690.

[58] Seidel G.; Lagoudas D. A micromechanics model for the electrical conductivity of nanotube–polymer nanocomposites. J. Compos. Mater. 2009, 43, 917–941. DOI: 10.1177/ 0021998308105124.

[59] Postma, H.; Jonge, M.; Yao, Z.; Dekker, C. Electrical transport through carbon nanotube junctions created by mechanical manipulation. Phys. Rev. B. 2000, 62, R10 653–R10 656, DOI: 10.1103/PhysRevB.62.R10653.

[60] Choi, E.; Brooks, J.; Eaton, D. Enhancement of thermal and electrical properties of carbon nanotube polymer composites by magnetic field processing. J. Appl. Phys. 2003, 94, 6034, DOI: 10.1063/ 1.1616638.

[61] Sandler, J.; Kirk, J.; Kinloch, I.; Shaffer, M.; Windle, A. Ultra-low electrical percolation threshold in carbon-nanotube epoxy composites. Polymer. 2003, 44, 5893–5899, DOI: 10.1016/S0032-3861(03)00539-1.

[62] Zhang, W.; Dehghani-Sanij, A.; Blackburn, R. Carbon based conductive polymer composites. J. Mater. Sci. 2007, 42, 3408–3418, DOI: 10.1007%2Fs10853-007-1688-5.

[63] Cai, W.; Tu, S.; Gong, J. A physically based percolation model of the effective electrical conductivity of particle filled composites. J. Compos. Mater. 2006, 40, 2131–2142, DOI: 10.1177/0021998306062312.

[64] Hindermann-Bischoff, M.; Ehrburger-Dolle, F. Electrical conductivity of carbon black–polyethylene composites: Experimental evidence of the change of cluster connectivity in the PTC effect. Carbon. 2001, 39, 375-382, DOI: 10.1016/S0008-6223(00)00130-5.

[65] Tang, H.; Chen, X.; Luo, Y. Studies on the PTC/NTC effect of carbon black filled low density polyethylene composites. Eur. Polym. J. 1997, 33, 1383-1386, DOI: 10.1016/S0014-3057(96)00221-2.

[66] Seidel, G.; Puydupin-Jamin, A. Analysis of clustering, interphase region, and orientation effects on the electrical conductivity of carbon nanotube–polymer nanocomposites via computational micromechanics. Mech. Mater. 2011, 43, 755–774, DOI: 10.1016/j.mechmat.2011.08.010.

[67] Natsuki, T.; Endo, M.; Takahashi, T. Percolation study of orientated short-fiber composites by a continuum model. Phys. Stat. Mech. Appl. 2005, 352, 498–508 DOI: 10.1016/j.physa.2004.12.059.

[68] Ma, H.; Gao, X. A three-dimensional Monte Carlo model for electrically conductive polymer matrix composites filled with curved fibers. Polymer. 2008, 49, 4230–4238, DOI: 10.1016/j.polymer.2008.07.034.

[69] Bao, W.; Meguid, S.; Zhu, Z.; Pan, Y.; Weng, G. A novel approach to predict the electrical conductivity of multifunctional nanocomposites. Mech. Mater. 2012, 46, 29–138, DOI: 10.1016/j.mechmat.2011.12.006.

[70] Wang, W.; Jayatissa, A. Computational and experimental study of electrical conductivity of graphene/poly(methyl methacrylate) nanocomposite using Monte Carlo method and percolation theory. Synth. Met. 2015, 204, 141–147, DOI: 10.1016/j.synthmet.2015.03.022.

[71] Haldar, A.; Mahadevan, S. Reliability assessment using stochastic finite element analysis. New York, NY, Wiley, 2000, ISBN: 978-0-471-36961-5.

[72] Koduru, S.; Haukaas, T. Feasibility of FORM in finite element reliability analysis. Struct. Saf. 2010, 32, 145–153, DOI: 10.1016/j. strusafe.2009.10.001.

[73] Thacker, B.; Riha, D.; Fitch, S.; Huyse, L.; Pleming, J. Probabilistic engineering analysis using the NESSUS software. Struct. Saf. 2006, 28, 83–107, DOI: 10.1016/j.strusafe.2004.11.003.

[74] Patelli, E.; Panayirci, M.; Broggi, M.; Goller, B.; Beaurepaire, P.; Pradlwarter, H.; Schuëller, G. General purpose software for efficient uncertainty management of large finite element models. Finite Elem. Anal. Des. 2012, 51, 31–48, DOI: 10.1016/j. finel.2011.11.003.

[75] Akula, V. Multiscale reliability analysis of a composite stiffened panel. Compos. Struct. 2014, 116, 432–440, DOI: 10.1016/j. compstruct.2014.06.001.

[76] McKenna, F.; Fenves, G.; Scott, M. Open system for earthquake engineering simulation. Berkeley, CA: University of California. Retrieved from http://opensees.berkeley.edu, 2000.

[77] Wang, J. Industrial design engineering: Inventive Problem Solving. Taylor and Francis group. 2017, ISBN: 9781498709606.

[78] Grimmett, G.; Stirzaker R. Probability and random processes. 2nd Edition. Clarendon Press, Oxford. ISBN: 0-19-853665-8.

[79] Sen, P.; Singer, J. Large sample methods in statistics. Singer. Chapman & Hall, Inc. 1994, ISBN: 0412042215.

[80] Teller, E.; Ulam, S. Heterocatalytic detonations, LAMS-1225, 2011.

[81] Brzenziak, Z. Basic Stochastic Processes: A Course Through Exercises (Springer Undergraduate Mathematics Series), 2000, ISBN: 3540761756,

[82] Zadeh, L. Fuzzy sets. Information and Control. 1965, 8, 338–353, DOI:10.1016/s0019-9958(65)90241-x.

[83] Maybeck, P. Stochastic models, Estimation and control, Mathematic in Science and Engineering, 141-3, ISBN: 0-12-480701-1.

[84] Guoliang, J.; Lin, C.; Changhai, L. The study of probability distribution of mechanical properties of steel used in pressure vessels for petrochemical industries, Journal of Daqing Petroleum Institute, 1991.

[85] Gagniuc, P. Markov Chains: From Theory to Implementation and Experimentation. USA, NJ: John Wiley & Sons. 2017, ISBN: 978-1-119-38755-8.

[86] Kwak, Y.; Ingall, L. Exploring Monte Carlo Simulation Applications for Project Management. Risk Manag. 2007, 9, 44–57, DOI: 10.1057/palgrave.rm.8250017.

[87] Power, D.; Sharda, R. Model-driven decision support systems: Concepts and research directions. Decision Support Systems. 2007, 43, 1044-1061, DOI: 10.1016/j.dss.2005.05.030.

[88] Edjer, T.; Baltussen, R.; Adam, T.; Hutubessy, R.; Acharya, A.; Evans, D.; Murray, C. Making Choices in Health: WHO Guide to Cost-effectiveness Analysis, ISBN 9241546018.

[89] L'Ecuyer, P. Random number generators, Encyclopedia of operations research and management science, Second Edition, 2001, 695-702, DOI: 10.1007/1-4020-0611-X_852.

[90] P. L'Ecuyer, History of uniform random number generation. Proceedings of the 2017 winter simulation conference, 2017, 13, 1–29, DOI: 10.5555/3242181.3242195.

[91] Barker, E.; Recommendation for Key Management, NIST Special Publication 800-57. NIST, 2013, DOI: 10.6028/NIST.SP.800-57pt1r4.

[92] Ecuyer, L. Uniform random number generators, International Encyclopedia of Statistical Science, Springer, 2010.

[93] Random (Java Platform SE 8), Java Platform Standard Edition 8, Documentation.

[94] Pollack, G. Kapitza Resistance. Rev. Mod. Phys. 1996, 41, 48, DOI: 10.1103/RevModPhys.41.48.

[95] ASTM D257-14, Standard Test Methods for DC Resistance or Conductance of Insulating Materials, ASTM International, West Conshohocken, PA, 2014, www.astm.org, DOI: 10.1520/D0257-14.

[96] Bigg, D. Mechanical properties of particulate filled polymers. Polym. Compos. 1987, 8, 115-122, DOI: 10.1002/pc.750080208.

[97] Ju, J.; Yanase, K. Micromechanics and effective elastic moduli of particle-reinforced composites with near-field particle interactions. Acta Mech. , 2010, 215, 135-153, DOI: 10.1007/s00707-010-0337-2.

[98] Smith, J. Experimental values for the elastic constants of a particulate-filled glassy polymer. J. Res. Nat. Bur. Stand. 1976, 80, 45–49, DOI: 10.6028/jres.080A.008.

[99] Kashiwagi, T.; Grulke, E.; Hilding, J.; Groth, K.; Harris, R.; Butler, K. Thermal and flammability properties of polypropylene/carbon nanotube nanocomposites. Polymer. 2004, 45, 4227-4239, DOI: 10.1016/j.polymer.2004.03.088.

[100] Liu, S.; Yan, H.; Fang, Z.; Wang, H. Effect of graphene nanosheets on morphology, thermal stability and flame retardancy of epoxy resin. Compos. Sci. Technol. 2014, 90, 40-47, DOI: 10.1016/j.compscitech.2013.10.012.

[101] Hahn, G. Sample Sizes for Monte Carlo Simulation. IEEE Transactions on Systems, Man and Cybernetics, 1972.

[102] Guoliang, J.; Lin, C.; Jiamei, D. Monte Carlo finite element method of structure reliability analysis. Reliab. Eng. Syst. Saf. 1993, 40, 77-83, DOI: 10.1016/0951-8320(93)90121-E.

[103] Pilkey, W.; Pilkey, D. Stress concentration factors. Wiley, Third edition, DOI: 10.1002/9780470211106.

[104] Ostoja-Starzewski, M. Material spatial randomness: from statistical to representative volume element. Probab. Eng. Mech. 2006, 21, 112–32, DOI: 10.1016/j.probengmech.2005.07.007.

[105] Drugan, W.; Willis, J. A micromechanics-based nonlocal constitutive equation and estimates of the representative volume element size for elastic composites. J. Mech. Phys. Solids. 1996, 44, 497–524, DOI: 10.1016/0022-5096(96)00007-5.

[106] Hill, R. Elastic properties of reinforced solids: some theoretical principles. J. Mech. Phys. Solids. 1963, 11, 357–72, DOI: 10.1016/0022-5096(63)90036-X.

[107] Karimi, D.; Milani, A. SRVE modeling of particulate polymer matrix composites with irregularly shaped inclusions: Application to a green stone composite. Compos. Struct. 2019, 228, 111331, DOI: 10.1016/j.compstruct.2019.111331.

Chapter 2
Mechanical Characterization of Particulate Polymer Composites

In the first part of this research study, a stochastic FEA framework was developed for calculating effective mechanical properties (i.e., the effective modulus of elasticity, Poisson's ratio and coefficient of thermal expansion) of spherical particulate polymer composites. Uncertainty was assumed for particle size and distribution only and constant values were used on the material properties of the constituents. The first part of this chapter describes an initial study in which a semi-automated process was employed to facilitate preliminary testing of the thesis hypothesis. The modeling involved finite element analyses with random input variables to calculate the effective modulus of elasticity and Poisson's ratio and compare the results with experimental result. This work was published as publication no. 1 listed the Preface of this document. The second part of this chapter describes the substantially expanded stochastic FEA framework that enabled calculating, in an automated fashion, the effective mechanical properties of randomly distributed and dispersed spherical particulate polymer composites. Since an automated modeling process was established, greater number of iterations were performed for predicting the effective properties. The second part of this chapter was published as publication no. 2 listed in the Preface of this thesis document.

Nomenclature

σ	stress
ε	strain
E	modulus of elasticity
l	uniform displacement
L	initial RVE dimension
ε_L	displacement direction strain
ε_T	transverse strain
υ	Poisson’s ratio
E_{c}	effective modulus of elasticity
$\alpha_{\zeta i}$	node coefficient of thermal
E_{f}	filler modulus of elasticity expansion
E_{m}	matrix modulus of elasticity
α_{eff}	effective coefficient of thermal
ϕ	volume fraction
i	node number
ζ	shape factor
P	probability
E_{U}	upper bound effective modulus of elasticity
χ	continues random variable
E_{L}	lower bound effective modulus of elasticity
$f(\chi)$	probability distribution function

$\sigma_{\zeta i}$	node normal stress
E_i	node modulus of elasticity
ε_{ζ}	global strain (Hencky strain)
l_i	deformed RVE length
L	initial RVE length
E_{eff}	effective modulus of elasticity
E_i	node modulus of elasticity
n	number of nodes
$\varepsilon_{\psi i}$	transverse strain
$\varepsilon_{\zeta i}$	normal strain
ν_i	node Poisson's ratio
ν_{eff}	effective Poisson's ratio
$\alpha_{\zeta i}$	normal deformation
$\Delta l_{\zeta i}$	axial displacement
ΔT	differential temperature

2.1 Mechanical Characterization – Preliminary Study

2.1.1 Introduction

Polymer materials are widely used due to their low cost, low volumetric mass density [1], and excellent degradation resistance, such as against corrosion [2-3]. However, in absolute terms, polymers have low mechanical properties and are poor heat and electricity conductors. The latter lesser qualities may be enhanced by the addition of suitable fillers into the polymer matrix [4]. With the advent of nanotechnology, researchers have developed a greater understanding pertaining to filler synthesis and the dispersion and distribution of particles to achieve desired property enhancements.

Of critical importance is the prediction of material properties prior producing filler-modified polymers. The reliance on purely experimental studies to characterize the material properties is generally time-consuming and expensive due to the multitude of possible filler options and process parameters. One of the most important subjects is determining the mechanical properties using analytical methods. A number of studies has been conducted thus far regarding this topic. For example, Eshelby pioneered the analysis of the stress field in an isolated ellipsoidal inclusion in an infinite elastic matrix [5]. Popular models for the lower and upper bounds for the effective modulus of elasticity were derived by Voigt [6] and Reuss [7]. Hashin and Shtrikman proposed an enhanced model for bounding the effective modulus of elasticity using variational principles [8]. Other researcher who provided expanded and enhanced modeling approaches in the context of micromechanics of materials include Mori and Tanaka [9], Benveniste [10], Hill [11], Christensen and Lo [12] and Torquato [13].

Numerical modeling approaches using statistical methods and finite element analysis (FEA) emerged and have become increasingly popular with the advance of computational facilities. Innovative work in this field was reported by in Drugan and Willis [14-15], who used a small-size representative volume element (RVE) with a small number of random spherical particles and numerical calculation to determine the effective modulus of elasticity of a filler-modified material. However, due to the small number of fillers that could be included, their model representation was limited in fully capturing a material morphology with random particle distribution. Segurado and Llorca continued this modeling approach to create a model with randomly distributed particles using a random sequential adsorption algorithm [16]. They generated 30 particles in a RVE and calculated the modulus of elasticity employing FEA and periodic boundary conditions.

The present article describes a methodology for calculating the effective modulus of elasticity and the Poisson's ratio for a composite containing randomly distributed spherical particles. Inspired by aforementioned numerical modeling work, the model uses a three-dimensional RVE in which filler particles are generated using a Monte Carlo approach and FEA to solve for the mechanical properties. The capabilities of the commercial FEA software ANSYS (Version 15, ANSYS Inc., Canonsburg, PA, USA) were used to a substantial degree for the modeling. Spherical graphite and glass beads were the two filler materials considered in this study along with epoxy polymer as the matrix. Results from the numerical modeling work were contrasted with data available in the technical literature.

2.1.2 Modelling Approach

The initial step in creating the three-dimensional geometry representing the material with randomly distributed particles was generating a set of random numbers. Specifically, through a Monte Carlo approach, random numbers were generated that are normal distributed. The mathematical programming environment MATLAB (MathWorks, Natick, MA, USA) was used for executing a code for random number generation and the Monte Carlo algorithm. The set of random numbers was used to define the size of spherical particles and their positions within the RVE. For each particle, three random numbers were generated representing Cartesian coordinates for the particle location and a fourth number for its size. Particles were thus successively created for inclusion in the RVE.

In the present modeling approach, particles are not allowed to intersect with each other or the RVE boundaries. These restrictions were imposed to ease the determination of the filler fraction and, more importantly, avoid mesh distortions in subsequent FEA modeling. With respect to the latter, a diminishing gap between particles, or a particle and the RVE boundary, yields high mesh aspect ratios, that is, mesh distortion, which is known to create challenges in terms of convergence of the numerical model and/or poor numerical results. Note that due to the limited RVE size the chance of particle intersection is comparatively high. The developed MATLAB code performs an intersection detection check for each added particle with respect to RVE boundaries and the particles already existing in the RVE. Once intersection is detected, the particle is discarded and a new particle is created instead. The feature of particle intersection detection makes the chosen modeling approach computationally expensive for increasing filler volume fractions. Hence, modeling was limited to filler volume fraction up to 20%.

Histogram data from the technical literature was used to incorporate realistic particle size distributions [17-18]. As an example, the experimentally determined size distribution for the considered glass bead filler is shown in Fig. 2-1. Data from these charts was employed by the numerical code to generate particle sizes. The procedure of generating particles for the RVE was performed until the desired filler volume fraction was reached, and particle location and size data was then stored in a text file (CSV format) for import into the FEA software.

The FEA model geometry was created using the ANSYS Parametric Design Language (APDL), employing the data from the previously described MATLAB code. After creating the geometry, the model was saved as a step file and then imported into ANSYS Workbench for further study. The boundary conditions for the cubic shaped RVE were defined as follows: a displacement boundary condition in the x-direction, which is the direction of the applied displacement, was applied on one side of the RVE, and a uniform displacement was applied on the opposite side, indicated by face B in Fig. 2-2.

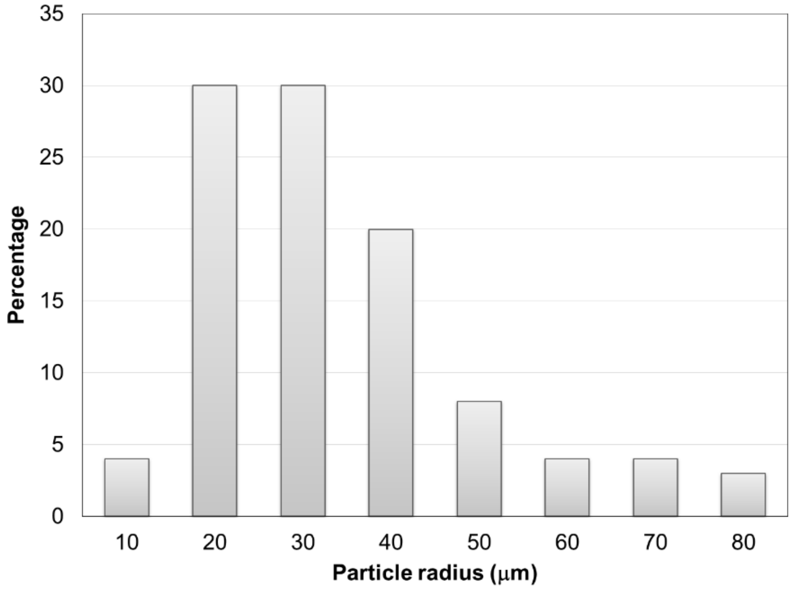


Figure 2-1 Size distribution histogram for glass beads, adopted from [17].

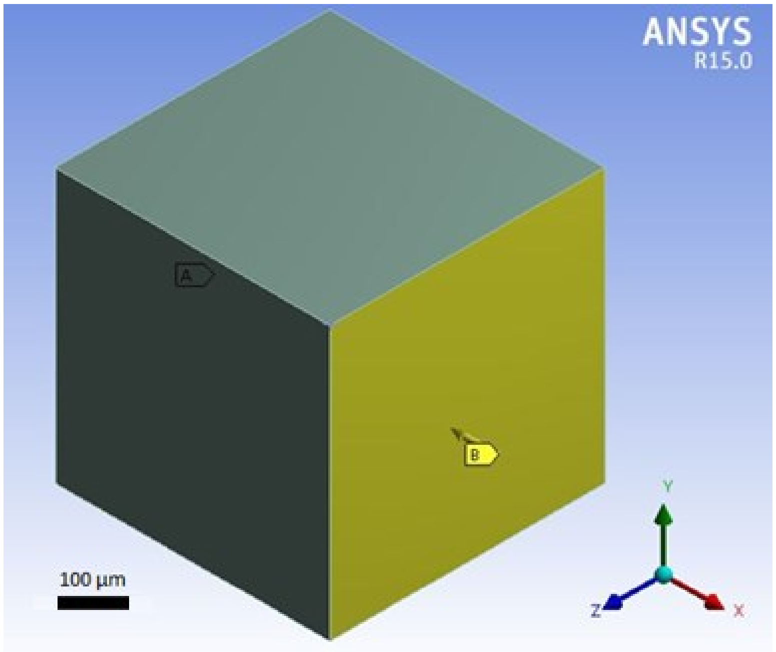


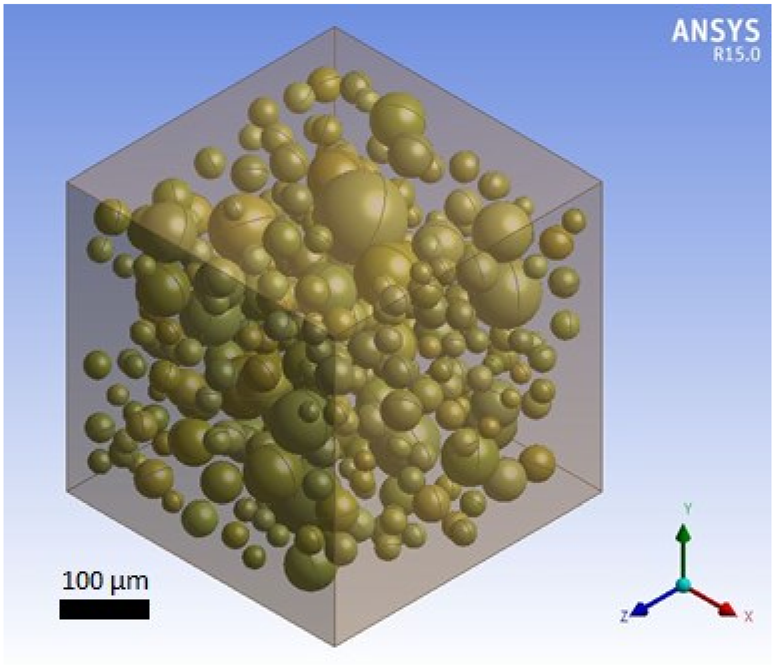
Figure 2-2 RVE configuration and coordinate system.

All materials were treated as linear elastic and respective properties for the spherical graphite, the glass beads and the epoxy polymer matrix are presented in Table 2-1. A sample RVE with dispersed particles is shown in Fig. 2-3A. The particles and matrix were assumed to be perfectly bonded, and no sliding or separation at interfaces was allowed.

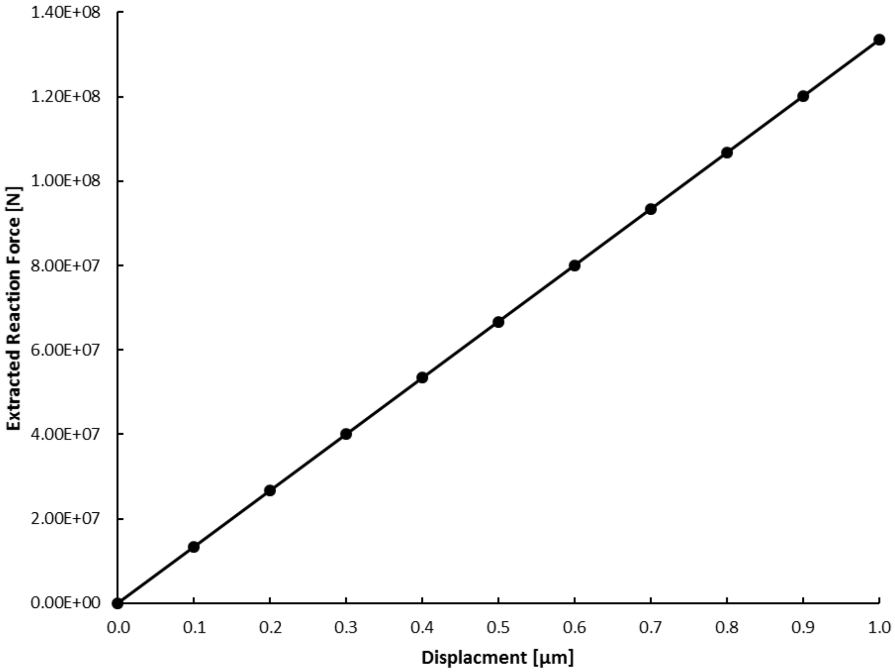
An important aspect in the context of this analysis is to assess the relation between incremental displacements applied on the RVE and the reaction force. This assessment was performed by subjecting a model with 5% filler volume fraction to 10 displacement substeps. Force-displacement data were extracted and are plotted in the graph in Fig. 2-3B. This graph clearly indicates a linear force-displacement relationship.

Table 2-1 Mechanical properties of the constituent materials [16] [19].

Property	Epoxy polymer	Spherical graphite	Glass beads
Young’s Modulus (GPa)	3.0	13.3	76.0
Poisson’s ratio (/)	0.4	0.23	0.23



(A)



(B)

Figure 2-3 Randomly dispersed particles inside the RVE with filler volume fraction of 19.4% (A), and reaction force versus displacement data computed for RVE with 5% filler volume fraction (B).

A tetrahedral type of mesh was used to mesh the model, see Fig. 2-4. A study was performed to validate the convergence of the FEA results for increasing mesh refinement. For this purpose, the average von Mises stress was computed within ANSYS for the RVE face on which the uniform displacement was imposed for different mesh refinement levels (see Section 2.3 for further details on the procedure). As shown in Fig. 2-5, the change in von Mises stress was less than 1% for an increase in degrees of freedom from approximately 1,065,000 to 1,785,000. Due to restrictions in computational facilities, the level of mesh refinement in this study was thus limited to a maximum number of approximately 1,785,000 degrees of freedom. Hence, the global mesh size was set to 1.6% of the RVE length for these analyses.

As in the convergence study, von Misses stress was calculated on the surface on which the uniform displacement was applied. The stress data from the multitude of nodal points was exported to the spreadsheet application Excel (Microsoft, Redmond, WA, USA). From these data the modulus of elasticity, E, was computed using Hook's law (i.e., Eq. (2-1)).

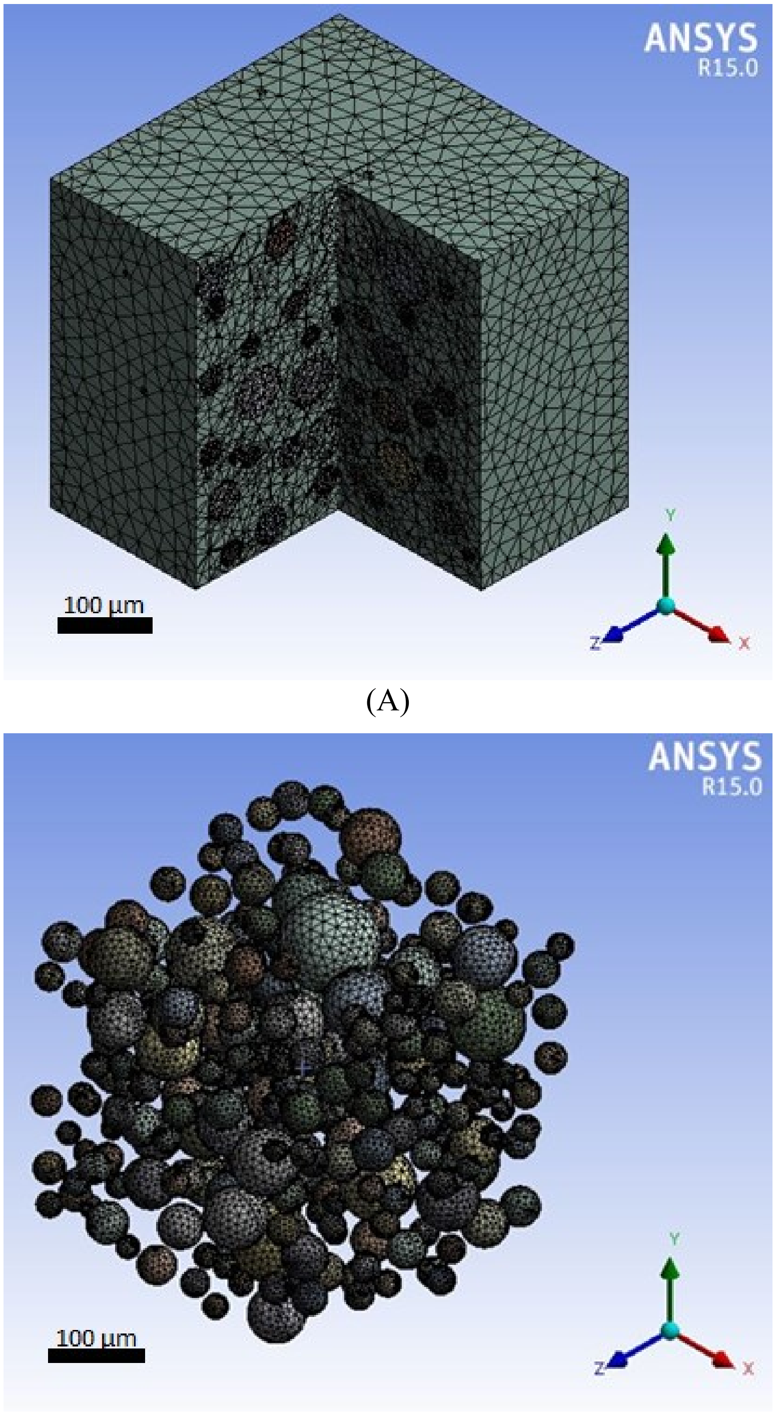
$$\sigma = E\varepsilon \tag{2-1}$$

where σ and ε are stress and strain, respectively. The latter was determined as Hencky strain according to Eq. (2-2).

$$\varepsilon = ln(l/L) \tag{2-2}$$

where l is measured length after applying the uniform displacement and L is the initial RVE dimension in the displacement direction. Note that strain in the displacement direction, ε_L, is identical for all points over the RVE face since a uniform displacement was imposed. The modulus of elasticity was calculated for each point and the average of the results yielded the modulus of elasticity for the RVE. Strain transverse to the displacement direction, ε_T, was also computed and used according to Eq. (2-3) to determine the RVE Poisson's ratio.

$$\upsilon = \frac{\varepsilon_T}{\varepsilon_L} \tag{2-3}$$



(A)

(B)

Figure 2-4 Meshing of the RVE, showing the polymer matrix (A) and the particles (B).

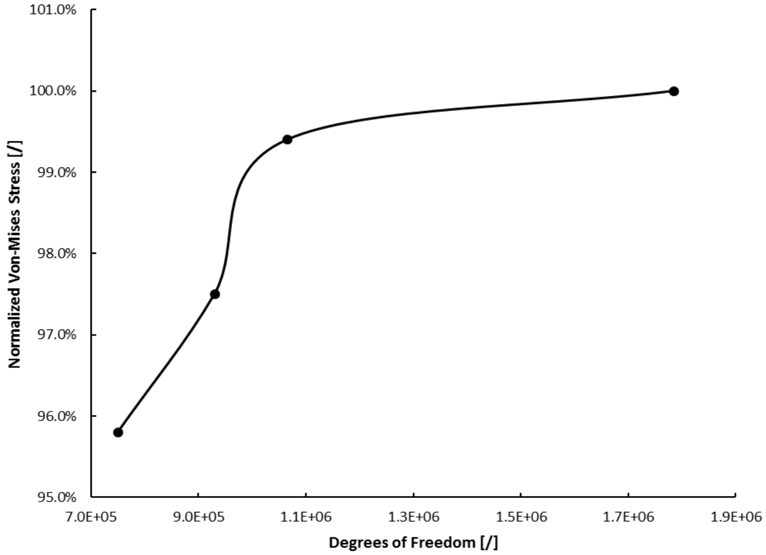


Figure 2-5 Convergence study for different mesh refinement levels. The average von Mises stress relates to RVE face on which the uniform displacement was imposed. Date was normalized with respect to ~1,785,000 degrees of freedom.

2.1.3 Results and Discussion

Moduli of elasticity and Poisson's ratios for the polymer composites filled with either spherical graphite or glass beads are shown in Figs. 2-6 to 2-9 for filler volume fractions ranging up to 20%. For every level of filler volume fraction, values were computed for five generated RVEs. While initially data scatter is modest, it can be observed that the spread of data points increases with increasing level of filler volume fraction. This important observation reveals the potentially significant effects that spatial particle distribution can have on the RVE material properties variations, which further depends on the particle size distribution. Such insight cannot be gained from analytical models and conventional numerical methods that lack the stochastic nature on the modeling approach taken in the present study. Note that for high filler volume fractions approaching 20% it was challenging and time consuming for the Monte Carlo algorithm to achieve the desired level of filler loading, especially since the model adhered to the filler size distribution histograms in order to achieve a realistic material representation. In general, it was not possible to exactly reach a filler loading of 20%, and the algorithm was terminated when a filler volume fraction was near the target value, resulting in some scatter with respect to the ordinate direction.

After calculating the mechanical properties for the prescribed displacement direction for each RVE (i.e., the x-direction), the process was repeated for a displacement transverse to the initial direction, that is, the y- and z-directions. Comparing the results for all directions revealed a negligible difference between them, which indicates that the RVE can be considered isotropic.

The numerical results for the glass bead modified epoxy were also compared with experimental data from the technical literature [20], where a modulus of 3.3 GPa and 4.19 GPa were reported correspondingly for filler loadings of 5% and 15%. According to Fig. 2-8, these values compare to 3.3 GPa and 3.6 GPa for respective filler volume fractions. It can thus be concluded that the model predictions describe same trend as the experimental data, yet, predicted values are lower than the reported measurements for higher filler loadings. In terms of Poisson's ratio (see Fig. 2-9), experimental values are 0.39 and 0.37 compare to predicted data of 0.39 and 0.36 for 5% and 15% filler loading, respectively. Consequently, good agreement in terms of Poisson's ratio can be ascertained between predicted and experimental data.

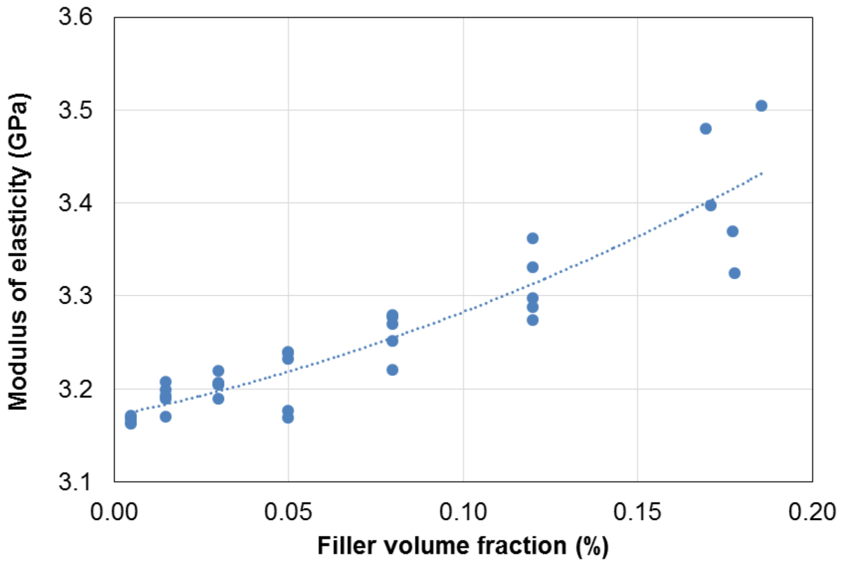


Figure 2-6 Modulus of elasticity for epoxy composite with spherical graphite filler.

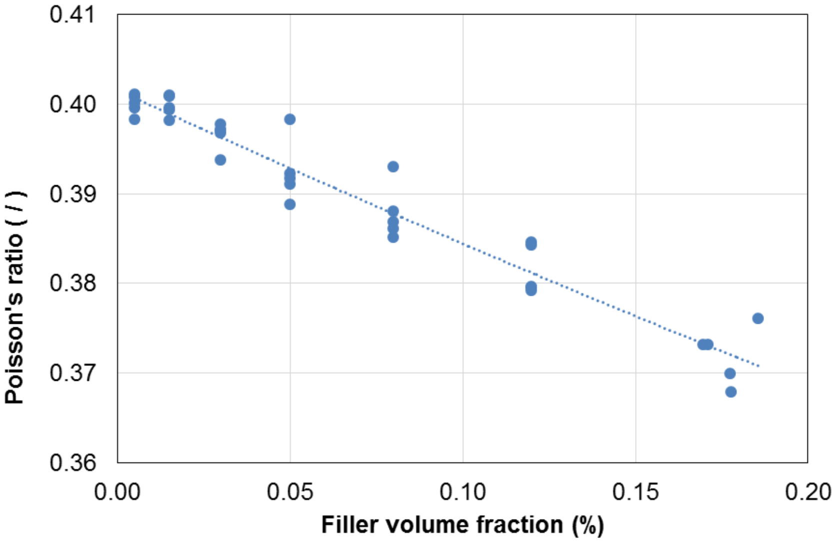


Figure 2-7 Poisson’s ratio for epoxy composite with spherical graphite filler.

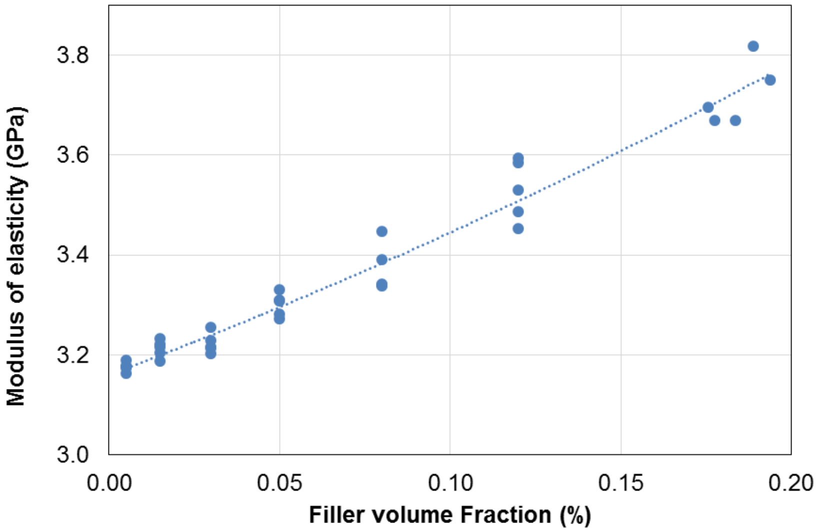


Figure 2-8 Modulus of elasticity for epoxy composite with glass beads.

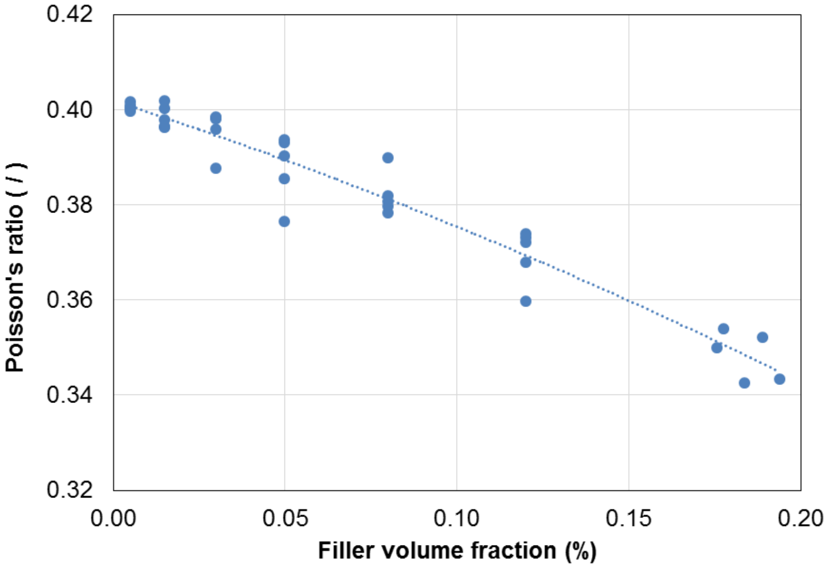


Figure 2-9 Poisson’s ratio for epoxy composite with spherical graphite filler.

2.1.4 Conclusions

A Monte Carlo approach combined with finite element modeling was employed to predict the mechanical properties of a polymer matrix modified with spherical filler particles. A random distribution of particles inside a representing volume element was thus achieved. The chosen modeling approach did not permit particles to intersect other particles or the RVE boundaries, and good convergence and mesh quality were ascertained. Uncertainty was imposed on particle size and distribution only, and constant values were used for the material properties of the constituents. Glass beads and spherical graphite were considered as filler materials in this study. The modulus of elasticity and the Poisson's ratio for different filler loadings were determined applying uniaxial traction to the RVE as part of the finite element analysis. When subsequently subjecting the RVEs to traction loading in mutually perpendicular directions, it was found that the RVEs exhibited isotropic behavior. A comparison between predicted values and experimental data from the technical literature for the glass bead filler revealed good agreement in terms of Poisson's ratio, whereas predictions of the modulus of elasticity remained below measurements for high filler loadings.

The presented modeling technique is seen as a promising approach to also predict other material properties. For example, the finite element environment allows for multiphysics analyses, and as such, thermal properties could be investigated in conjunction with the mechanical behavior. Moreover, non-linear effects such as viscoelasticity may be incorporated using suitable material models for the matrix and filler. Consequently, the given modeling approach is versatile and provides a powerful means for advanced material properties predictions.

2.2 Mechanical Characterization - Comprehensive Investigation

2.2.1 Introduction

Having low volumetric mass density and high environmental degradation resistance (e.g., corrosion) while being low-cost, polymers are attractive for a variety of engineering applications [1-3]. Yet, limited mechanical, thermal and electrical properties often restrict the use of polymers. This problem has motivated material designers to enhance polymer properties by adding appropriate types of fillers [4]. A variety of filler types and morphologies, in addition to the chosen filler volume fractions (FVF), are available for modifying polymers. The multitude of micro and nano-scale particles, and combinations thereof creating not only single but binary, ternary and even higher order modified polymers, create a need for modeling and experimental processes to guide the design of filler-modified polymers with well dispersed and distributed particles [21-22]. As such, the development of filler modified polymer these composites is a costly and time-consuming process that is challenging to perform [23] by material designers.

Various studies have been reported for predicting the properties of advanced materials [24-27]. Eshelby pioneered the characterization the particulate composites by performing stress field analysis with an ellipsoidal inclusion [5]. The Halpin-Tsai empirical model, as shown in Eqs. (2-4) and (2-5), was proposed for predicting the composite modulus of elasticity for a variety of particle geometries [28].

$$E_{\mathrm{c}} = E_{\mathrm{m}} \frac{1 + \zeta\eta\phi}{1 - \eta\phi} \tag{2-4}$$

$$\eta = (\frac{E_{\mathrm{f}}}{E_{\mathrm{m}}} - 1)(\frac{E_{\mathrm{f}}}{E_{\mathrm{m}}} + \zeta)^{-1} \tag{2-5}$$

where E_{f} and E_{m} are modulus of elasticity of filler and matrix respectively, ϕ is the volume fraction of filler, and ζ and E_{c} are correspondingly a shape factor and the effective modulus of elasticity.

The approximation by Mori and Tanaka [9] is an extension of the Eshelby solution. Other analytical methods include modelling approaches by Benveniste [10], Christensen and Lo [12], and Torquato [13].

Some analytical approaches provide estimates for the upper and lower bounds of the modulus of elasticity with respect to the filler volume fraction. The Voight [6] and Reuss [7] models, as indicated by Eqs. (2-6) and (2-7), are the most basic upper and lower bounding techniques, respectively. The variational approach developed by Hashin and Shtrikman [8] is another popular analytical technique for predicting the bounds for the effective modulus of elasticity.

$$E_{\mathrm{U}} = \phi E_{\mathrm{f}} + (1-\phi)E_{\mathrm{m}} \tag{2-6}$$

$$\frac{1}{E_{\mathrm{L}}} = \frac{\phi}{E_{\mathrm{f}}} + \frac{(1-\phi)}{E_{\mathrm{m}}} \tag{2-7}$$

where E_{U} are E_{L} are upper and lower bound effective modulus of elasticity, respectively.

In addition to randomly oriented particles, modified polymers with aligned particles have gained interest among material designers for improving mechanical and thermal properties [29]. High strength filler modified polymers [30-32] and sensors [33-34] are example applications that require filler alignment to create explicit anisotropic mechanical properties. Carbon nanotubes, for example, are a popular filler type being considered for creating anisotropic materials for a variety of applications [35-36].

Analytical approaches are widely recognized as expedient for predicting mechanical properties of filler modified polymers. However, these methods often lack in accuracy, especially for high FVF. Experimental studies, on the other hand, provide direct information on material properties. But being time-consuming and costly, experiments by themselves are an unsatisfactory solution for researchers designing advance materials [37]. In light of the aforementioned complexities associated with filler modified polymers, and a rapidly growing number of opportunities of these composites in industry, it is imperative to identify alternative approaches that allow for the efficient and effective prediction of material properties [38-40].

Stochastic methods are prominently used in reliability analysis which requires analyzing multiple input variables and predicting outcomes with a suitable level of accuracy. Stochastic methods have been powerful in other areas as well, such as complicated financial and forecasting models that involve several random and difficult to assess input variables. There are different methods of stochastic analyses, some relying on analytical approaches for predicting outcomes, e.g., first and second-order reliability methods. Using statistical principles, these methods enable predicting outputs from randomly determined input variables, which often follow certain types of random patterns or probability distributions. Stochastic methods thus consider input data variation or fluctuation and enable precise outcome estimation, even for complex systems. It is important to recognize that in some applications, uncertainty in input data may be dependent on each other. Such problems would be very difficult to predict, however stochastic analysis enables simulating the outcome by performing a large number of simulations known as stochastic projections [41].

It is convenient to consider explicit input parameters when solving engineering problems. As such, also explicit outputs are simulated. However, strictly speaking, explicit input values do not exist for e.g. material properties, boundary conditions and structural geometries, since such data may be subject to variation or fluctuation. Therefore, considering input parameters that are explicit can lead to generating hidden errors in final solutions and outcomes that are far from reality. Alternatively, by considering uncertainty for input parameters, robust results can be obtained. Moreover, the number of input parameters that influence output values increases as engineering models become more comprehensive. Consequently, material designers are faced with multiple challenges, which highlights the importance of developing innovative simulation methods.

Stochastic finite element analysis (SFEA) is a statistical method developed recently for solving sophisticated engineering problems that are impossible to solve using analytical methods [42-43]. As the name implies, SFEA is capable of considering uncertainty for model input parameters. In the present context of filler-modified polymers, several random variables need to be considered for the material design, e.g., particle locations, orientations, dimensions and properties. All of these parameters are known to affect outcomes appreciably. Given the complex nature of this problem, SFEA is considered a viable approach for material properties prediction.

Two different options are pursued for connecting stochastic analysis with finite element analysis (FEA) in order to perform SFEA on engineering models. The first option entails the use of dedicated commercial FEA software with stochastic analysis and scripting capability [44-46]. Stochastic analysis and scripting capabilities are hereby integrated into the software, making it convenient for a user to perform SFEA since the software applies uncertainty to input variables. However, this first option is limited in terms of flexibility to incorporate specific modeling requirements. The second and arguably more powerful option involves advance scripting for connecting stochastic analysis with FEA, providing maximum modeling flexibility, e.g., in terms of applying uncertainty to input variables, choosing a suitable pseudorandom number generator and performing customized FEA processes, an examples of which is demonstrated by [47]. However, the second option is typically more challenging for users to perform. Also, with greater opportunities for capturing complexities in terms of model features and input data variation, the second option is prone to become computationally expensive, necessitating the use of high-performance computational resources. Recently, the present authors developed an SFEA framework for predicting the effective thermal conductivity of particulate polymer composites [48]. The interested reader is referred to this article for detailed information about the developed SFEA framework, which is briefly summarized in the subsequent section. In general, this approach is capable and intended of modeling filler particle sizes ranging from nano- to micro-scales. However, especially for nano-scale fillers, great care must be taken to properly identify and implement the particle-to-particle and particle-to-matrix interactions in order to correctly capture the effective material properties. For the present study, the SFEA framework was employed to explore mechanical properties, i.e., elastic modulus, Poisson's ratio and coefficient of thermal expansion (CTE), for a composite with size-distributed spherical particles. Specifically, a composite consisting of an epoxy matrix with micro-sized spherical glass particles was studied and numerical results were contrasted with analytical predictions and experimental data from the technical literature.

2.2.2 Stochastic FEA Framework

The developed modeling framework employs Monte Carlo simulation techniques and computes outcomes using statistical analysis, which necessitates large numbers of model iterations. A customized stochastic analysis process in conjunction with parametric FEA was thus developed that automates the process of applying uncertainties to input variables by connecting several program modules using multiple programming languages. The illustration in Fig. 2-10 is the high-level architecture of the algorithm used for the modeling framework. In the following, the various modules depicted in Fig. 2-10, including associated acronyms, will be explained.

Visual Basic for Applications (VBA) programming language was employed for connecting and routing information between the various modules used in the framework. A customized “Front End” form was created that enables the capturing of all input parameters required for the analysis and storing them in the database (‘DBMS’). Input parameters captured by the “Front End” form include: Size information for the modeled representative volume element (RVE), the FVF that are to be analyzed, filler and matrix properties, filler particle size distribution, boundary conditions, settings that control the contact behavior for filler particles and matrix, and parameters needed for finite element (FE) mesh generation.

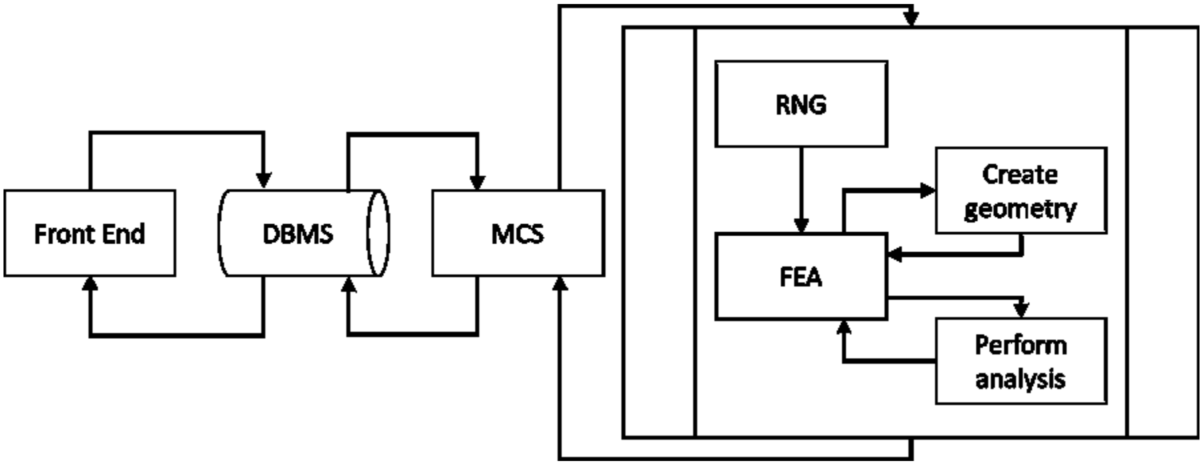


Figure 2-10 High-level architecture of developed stochastic FEA framework which shows how individual modules are connected together to form the framework [48].

An Open Database Connectivity (ODBC) method was used for developing the Database Management System (DBMS). The ODBC approach enabled creating an operating routine that is independent of the database and provides a high degree of flexibility for accessing the database at any time in the analysis. One task of the DBMS is the transfers of input parameters to the Monte Carlo Simulation (MCS) module in order to perform the SFEA.

Using VBA programming language the MCS module was developed having a tabulated structure. The module enables storing input parameters and results created by the model generation and analysis sub-processes. The algorithm as shown in Fig. 2-11 is repeated for each specified FVF until statistical objectives are satisfied. MCS module computes mean values, standard deviations and variances, which are used as objective values for terminating the framework iteration procedure. Once acceptance criteria defined in the MCS module are satisfied, e.g., an explicit standard deviation or variance, the MCS stops iterating and calculates the final effective mechanical properties. Results calculated in each iteration and mean values are saved to the database, which can be accessed through the “Front End”.

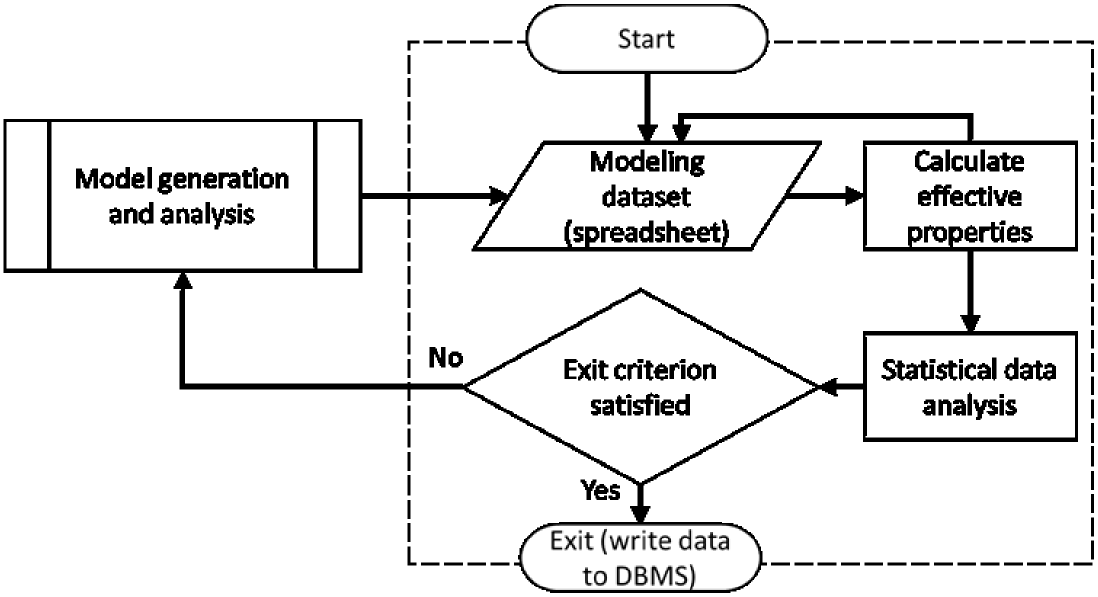

Start
Model generation and analysis
Modeling dataset (spreadsheet)
Calculate effective properties
Statistical data analysis
Exit criterion satisfied
No
Yes
Exit (write data to DBMS)

Figure 2-11 Schematic representing algorithm of Monte Carlo Simulation (MCS) module [48].

The Random Number Generator module (RNG, see Fig. 2-10) was developed using the general mathematical programming environment MATLAB, enabling the generation of random numbers required for input variables. In the present study, these variables are Cartesian coordinates for particle locations (X, Y, Z) and the particle diameter. Particles are thus randomly distributed and dispersed within the RVE. The RNG module, illustrated in Fig. 2-12, also performs collision detection between particles inserted into the RVE. A routine was defined that determines the distance for each particle newly inserted into the RVE with respect to all other particles already contained in the RVE. If no particle collision is detected, the new particle is accepted into the RVE; otherwise, the particle is rejected. This process continues until the specified FVF is satisfied, at which point the RNG module stops providing randomized particle data.

The composition of particulate filler materials may follow a certain particle size distribution. Simply using an average particle size may not reflect the behavior of the composite material. Hence, the SFEA framework was developed to account for particle size data that conforms to a prescribed size distribution using a data binning approach. The interested reader is referred to [48] for details on the algorithm that was included to conform particle dimensions to an explicit size distribution, and for the influence on computational performance when adding size-ordered particles to the RVE. The latter was implemented in the SFEA framework.

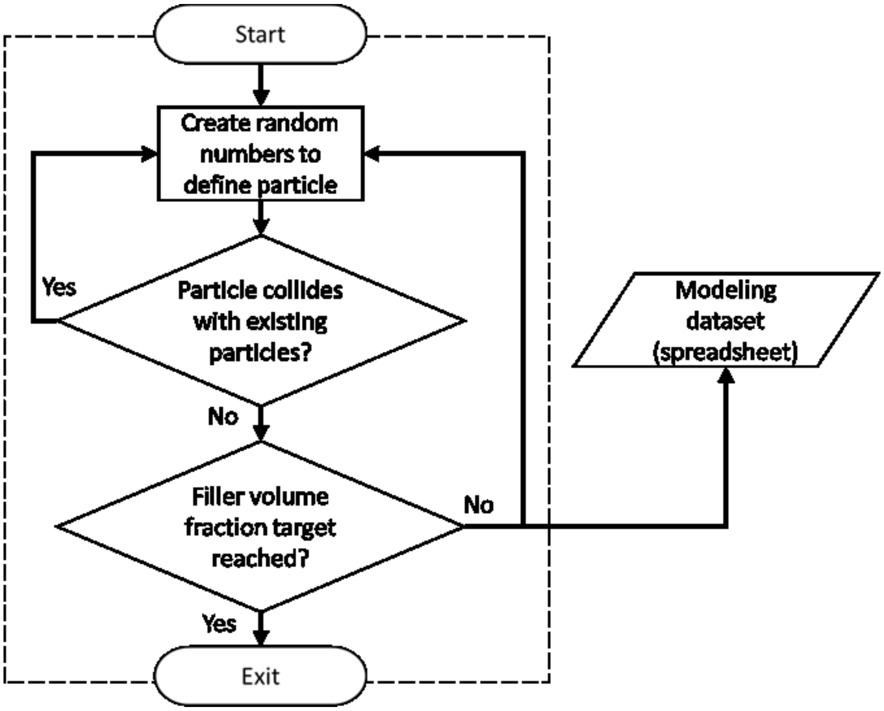

Start
Create random numbers to define particle
Particle collides with existing particles?
Yes
No
Filler volume fraction target reached?
No
Yes
Exit
Modeling dataset (spreadsheet)

Figure 2-12 Schematic illustrating the Random Number Generator (RNG) module [48].

For the model generation and analysis sub-processes a customized parametric FEA platform was developed using the commercial FEA software ANSYS Workbench (Version 19, ANSYS Inc., Canonsburg, PA, USA). This FEA platform as depicted in Fig. 2-13 was realized in conjunction with scripting in IronPython programming language. Two distinct submodules were developed in order to create the parametric three-dimensional model geometry and subsequently the full FEA model in a two-step process. In the first step, ANSYS DesignModeler in conjunction with scripting in JAVA programming language was used to read the data generated by the RNG module and create geometries for particles and the RVE. The resulting geometric representation was then transferred to the FEA modeling environment. For the second step, the FEA environment was developed using ANSYS Mechanical, again in combination with JAVA scripting, which facilitates an automated process of reading data from the database and creating the FEA model, including the application of material properties, mesh generation, and extraction of FEA results. Subsequent to mesh generation, the FEA environment submodule performs a convergence study by refining the mesh and extracting results to check whether results converge satisfactorily. Finally, results obtained with appropriately refined meshing are transferred to the MCS module and saved for further statistical analysis.

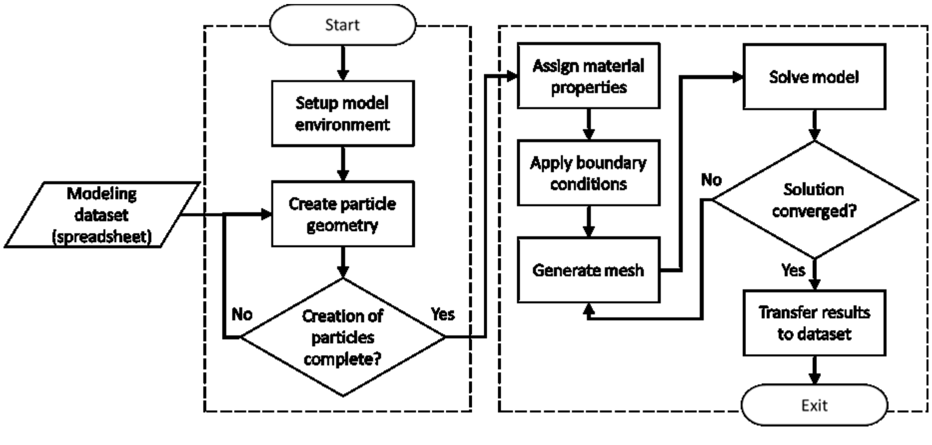

Start
Setup model environment
Modeling dataset (spreadsheet)
Create particle geometry
No
Creation of particles complete?
Yes
Assign material properties
Apply boundary conditions
Generate mesh
Solve model
No
Solution converged?
Yes
Transfer results to dataset
Exit

Figure 2-13 Schematic of FEA platform for model generation and analysis sub-processes [48].

2.2.3 Model Details

The developed SFEA framework was herein employed to predict the mechanical properties of particulate polymer composites with randomly distributed spherical particles under static-structural condition. Specifically, spherical glass bead particles embedded in epoxy were considered. Properties were adopted from the technical literature accordingly [19-20], see Table 2-2. Materials were treated as linear elastic. The particle size distribution illustrated in Fig. 2-14 was adapted from [17] assuming that all particles fall within a band ranging from 0 to 50 μm, and a data binning approach with 5 bins was used to have particle sizes emulate the given distribution. The size of the cubical RVE was set at 400 μm based on preliminary size effect studies.

Table 2-2 Properties of filler particles and polymer matrix.

Property	Glass Bead	Epoxy
Modulus of Elasticity (Gpa) [19-20]	76.0	3.0
Poisson's Ratio (/) [19-20]	0.23	0.40
Coefficient of thermal expansion (K^{-1}) [49,50]	9.0×10^{-6}	55.0×10^{-6}

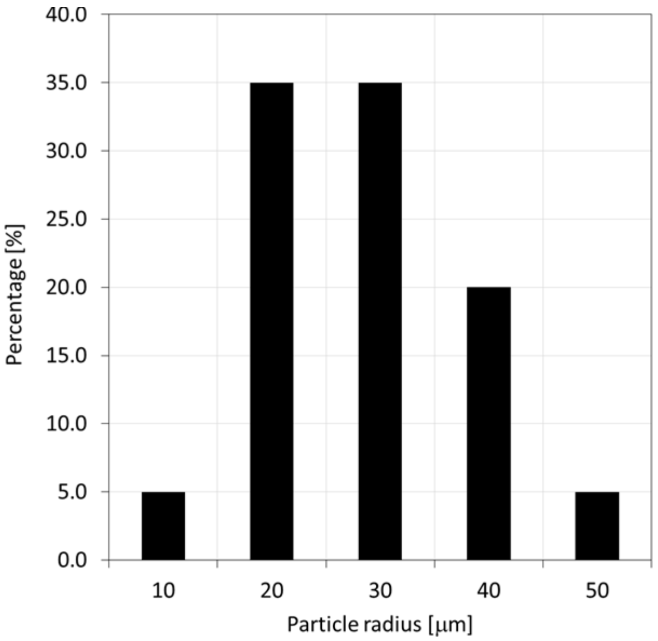


Figure 2-14 Glass bead size distribution, adapted from [17].

Three-dimensional 10-node quadratic tetrahedral structural solid elements (SOLID187) were used for meshing both particles and matrix. Fig. 2-15 illustrates an example mesh generated for the RVE and the particles included within it. The interface between particles and matrix was defined employing three-dimensional 8-node surface-to-surface contact elements (CONTA174 and TARGE170). For the study presented herein it was assumed that interfacial contact between particles and polymer matrix constitutes perfect bonding, and hence, no relative displacement at the interface was allowed.

Boundary conditions were applied to the RVE for generating strain and stress in the FE model. Nodes in one of two opposing RVE faces were restrained from out-of-plane displacement and rotation, while nodes remained free to move with the plane. This constraint was accomplished defining a cylindrical coordinate system at each respective node with the axial coordinate being aligned in the out-of-plane direction ζ and constraining the appropriate coordinate directions. Nodes in the other opposing RVE face were subjected to a uniform displacement of 1 μm perpendicular opposite to the constrained RVE face. As such, a global normal strain $\varepsilon\zeta$ of 2500 $\mu\varepsilon$ was applied to the RVE with size of 400 μm. Also, it was decided to choose a global mesh size of 2% of the RVE length for these analyses.

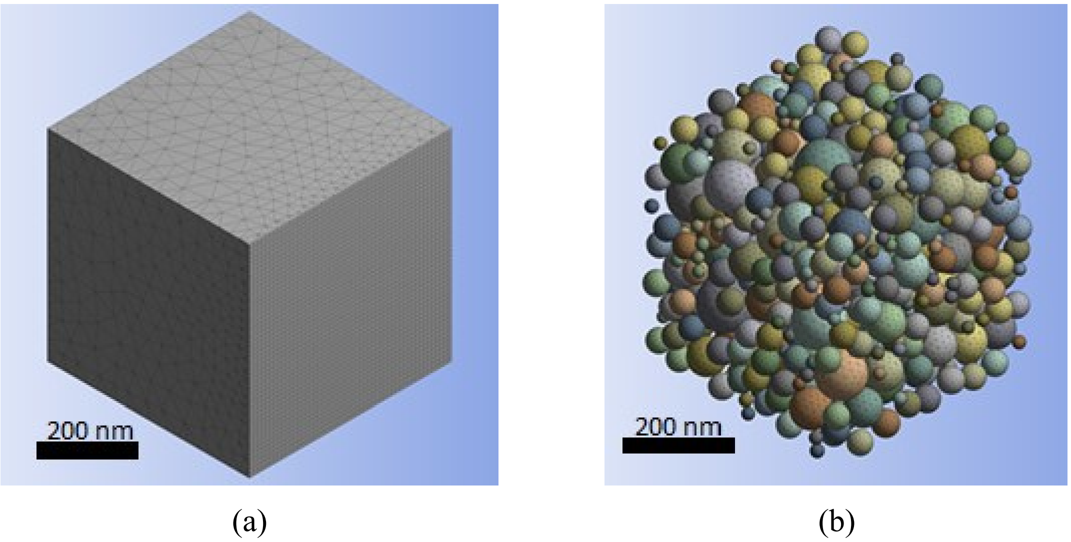


Figure 2-15 Example meshing of (a) RVE and (b) particles.

Using Eq. (2-8) the modulus of elasticity was computed for each node i located in the RVE face subjected to the prescribed displacement.

$$\sigma_{\zeta i} = E_i \varepsilon_\zeta \tag{2-8}$$

where $\sigma_{\zeta i}$ and E_i are the normal stress extracted from the FE results and modulus of elasticity on the ith node, respectively. As an example, Fig. 2-16 displays normal stress on the model for a FVF of 0.45.

Invoking the Hencky strain method as given by Eq. (2-9) for all nodes subjected to the prescribed displacement yields nodal strains being equivalent to the global strain.

$$\varepsilon_{\zeta i} = \ln \left({}^{l_i}\!/_{L} \right) \tag{2-9}$$

where l_i is the deformed RVE length at a specific node position and L the initial RVE length.

Finally, the effective modulus of elasticity E_{eff} for the polymer composite embodied by the RVE was calculated using Eq. (2-10).

$$E_{\text{eff}} = \frac{\sum_{i=1}^{n} E_i}{n} \tag{2-10}$$

where n is the total number of nodes.

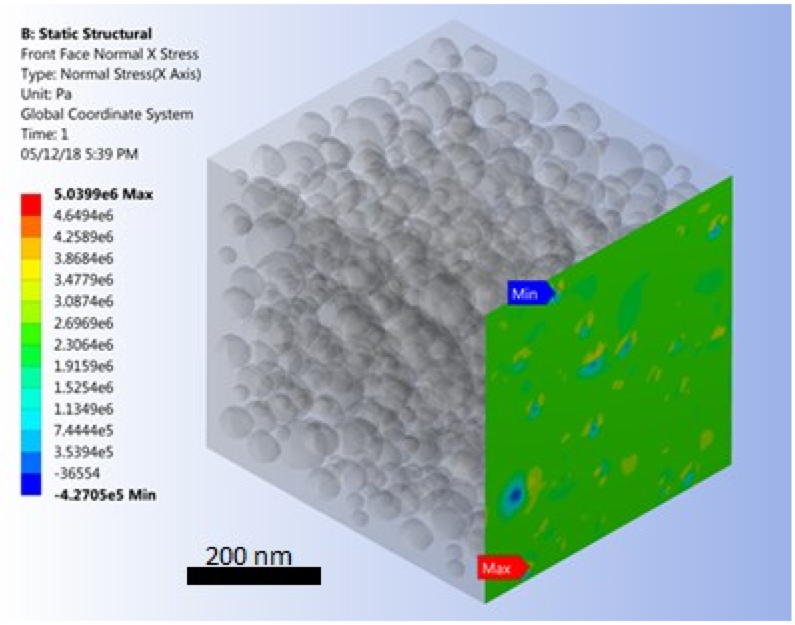


Figure 2-16 FE model depicting normal stress on RVE face being subjected to uniform nodal displacement.

In order to facilitate evaluating the effect of filler loading on Poisson's ratio, boundary conditions were modified as compared to the modulus of elasticity analysis - that is, Cartesian boundary conditions were applied restraining the out-of-plane displacement of nodes located in one of two opposing RVE faces, with nodes being free to move in the mutually perpendicular directions. To restrain rigid body motion of the RVE, one corner node of the restrained RVE face was held fixed in all three Cartesian directions. As in the previous case, nodes associated with the RVE face opposite to the constrained plane were again subjected to a 1 μm displacement normal to the plane.

Using Eq. (2-11), Poisson's ratio values ν_i were calculated for each node in the displaced RVE face. The effective Poisson's ratio ν_{eff} for the RVE was determined by Eq. (2-12).

$$\nu_i = \frac{\varepsilon_{\psi i}}{\varepsilon_{\zeta i}} \tag{2-11}$$

$$\nu_{\mathrm{eff}} = \frac{\sum_{i=1}^{n} \nu_i}{n} \tag{2-12}$$

where $\varepsilon_{\psi i}$ and $\varepsilon_{\zeta i}$ represent transverse and normal strain, respectively.

The effect of filler loading on the CTE was assessed by applying the same constraints to a single RVE face as for the modulus of elasticity analysis. In addition, a temperature change ΔT of 10 K was applied to the entire RVE. The CTE associated with a node i, $\alpha_{\zeta i}$, and the effective CTE for the RVE, α_{eff}, were calculated using Eqs. (2-13) and (2-14), respectively. As an example, deformations normal to the RVE plane due to thermal expansion are depicted in Fig. 2-17.

$$\alpha_{\zeta i} = \frac{\Delta l_{\zeta i}}{L\, \Delta T} \tag{2-13}$$

$$\alpha_{\mathrm{eff}} = \frac{\sum_{i=1}^{n} \alpha_{\zeta i}}{n} \tag{2-14}$$

where $\Delta l_{\zeta i}$ is axial displacement extracted from the FE results at the i^{th} node.

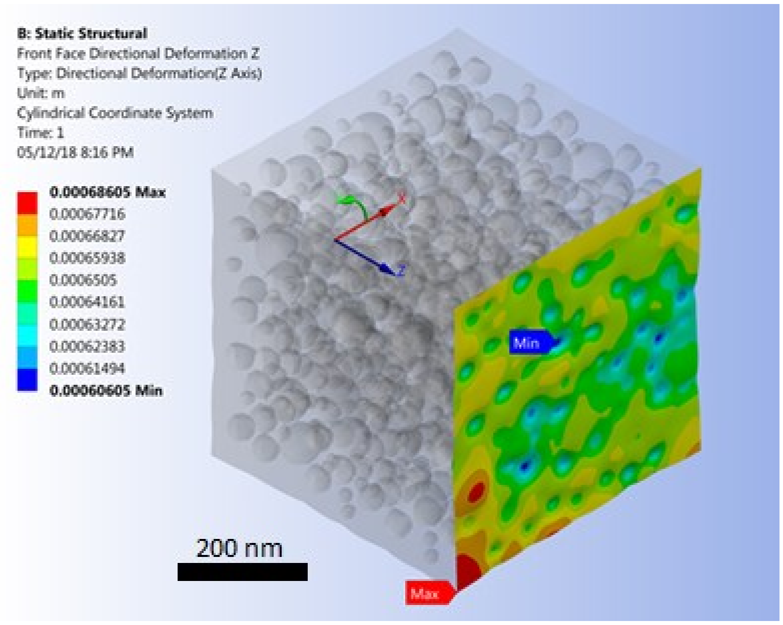
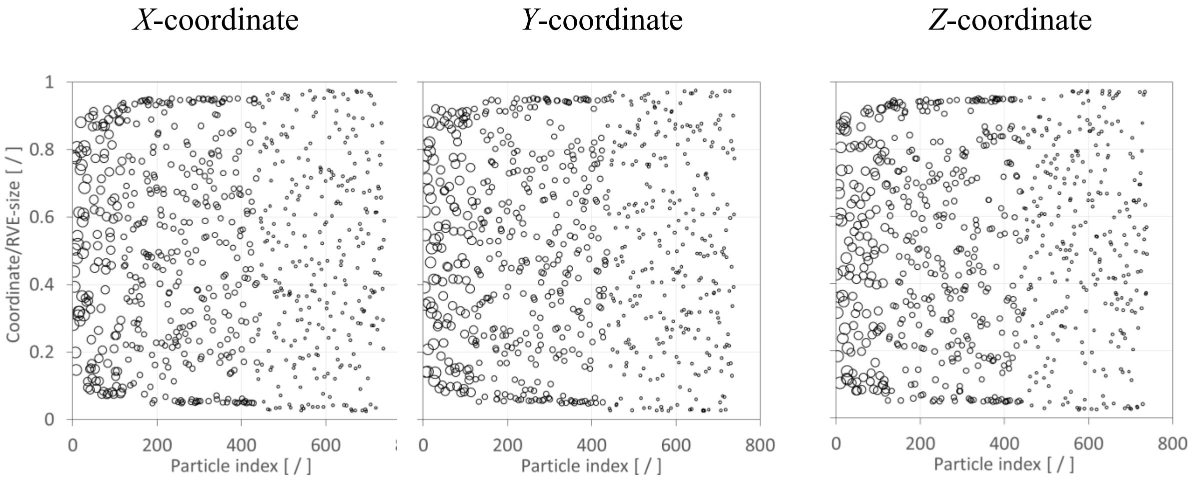


Figure 2-17 Deformations due to thermal expansion normal to the RVE plane (deformations scaled).

2.2.4 Results and Discussion

Prior to studying the effect of filler addition on mechanical properties, test cases were performed to assess the distribution of filler particles within RVE. The particle coordinates for the three Cartesian directions were plotted as depicted in the example shown in Fig. 2-18. It can be observed that particles were randomly distributed and dispersed within RVE. Fig. 2-18 also illustrates that, as mentioned above, the largest particles were added to the RVE first (smallest particle index), followed by particles in decreasing size order. Moreover, three sets of numerical analyses were performed, and mechanical properties were calculated for each of the mutually perpendicular axes defining the RVE (X, Y, Z). The example data in Table 2-3 indicates that differences between the results for different directions are negligible. Therefore, it was concluded that the SFEA framework, and especially the random number generator integrated in the RNG module, performed satisfactorily for creating true randomness in the analysis.



X-coordinate *Y*-coordinate *Z*-coordinate

Figure 2-18 Distribution of particles in RVE for 0.45 FVF. Symbol size is indicative of particle size.

Table 2-3 Properties calculated along mutually perpendicular axes defining the RVE for 0.275 FVF.

Property	X Coordinate	Y Coordinate	Z Coordinate
Moduls of Elasticity (Gpa)	5.642	5.639	5.642
Poisson's Ratio (/)	0.3055	0.3035	0.3170
Thermal Coefficient of Expansion (m)	40.13	39.94	39.91

After confirming its proper function the SFEA framework was employed to predict the effective modulus of elasticity, Poisson's ratio and CTE for the glass bead/epoxy composite for FVF of 0.05, 0.10, 0.15, 0.20, 0.275, 0.35 and 0.45. As mentioned above, the effective properties for each volume fraction are computed via the MCS module and then transferred to the database for statistical analysis. The unbiased standard deviation, variance and mean are calculated for the stored effective properties data. The MCS module iterates the FEA model until the acceptance criteria defined in MCS module are satisfied, i.e., the standard deviation reaches an explicit value. Then, the final effective properties for a given FVF are computed as the average from all iterations.

After completing a full Monte Carlo simulation for a specific FVF, a study was performed to assess the quality of the data that was acquired. It was hypothesized that the SFEA framework creates true randomness with data falling under a normal distribution with close congruence between the mean and median values. Ideally the determination of properties for each FVF would be done with an infinite number of iterations, and hence, predictions are considered continuous random variables for statistical analysis purposes. A probability density function (PDF) was thus computed for each dataset related to a given FVF. The normalized PDF for modulus of elasticity, Poisson's ratio and CTE data is depicted in Figs. 2-19(a) to 2-21(a).

The data presented in Figs. 2-19 to 2-21 suggests that the effective properties are normally distributed. To test this assumption, statistical data analyses were performed based on mean, median, skewness and kurtosis values, with skewness and kurtosis providing a measure for asymmetry and peakedness of the distributions, respectively. In other words, these measures were employed to assess how well the data adheres to a normal distribution (e.g., skewness and kurtosis of zero indicate a perfect normal distribution). A comprehensive study was performed by West et al. [51] regarding normal distribution quality indicators. Based on their study, skewness greater than 2 and kurtosis greater than 7 are considered indicators for a significant deviation from normal distribution. Also, the number of iterations was emphasized as a parameter that has the greatest influence on skewness and kurtosis. Tables 2-4 to 2-6 are showing correspondingly the statistical analyses performed for modeling data relating to the modulus of elasticity, Poisson's ratio and CTE. The statistical analysis results show that mean and median values are closely congruent, and skewness and kurtosis values are close to zero. Therefore, all datasets are considered to have passed normality test requirements, which is considered also a confirmation that sufficient numbers of iterations were performed.

As mentioned above, calculated final effective properties can be considered continuous random variables, and hence, the probability of occurrence of a specific value can be calculated within an interval as expressed by Eq. (2-15). Data from the model simulations can thus be presented in the form of cumulative distribution functions (CDFs). CDFs for the effective properties, i.e., the modulus of elasticity, Poisson's ratio and CTE, are shown in Figs. 2-19(b) to 2-21(b), respectively.

$$P(a \leq X \ \leq b) = \int_a^b f(\chi)d\chi \tag{2-15}$$

where P is the probability of an effective property occurring within an interval a and b; χ and $f(\chi)$ are a continuous random variable and the PDF, respectively.

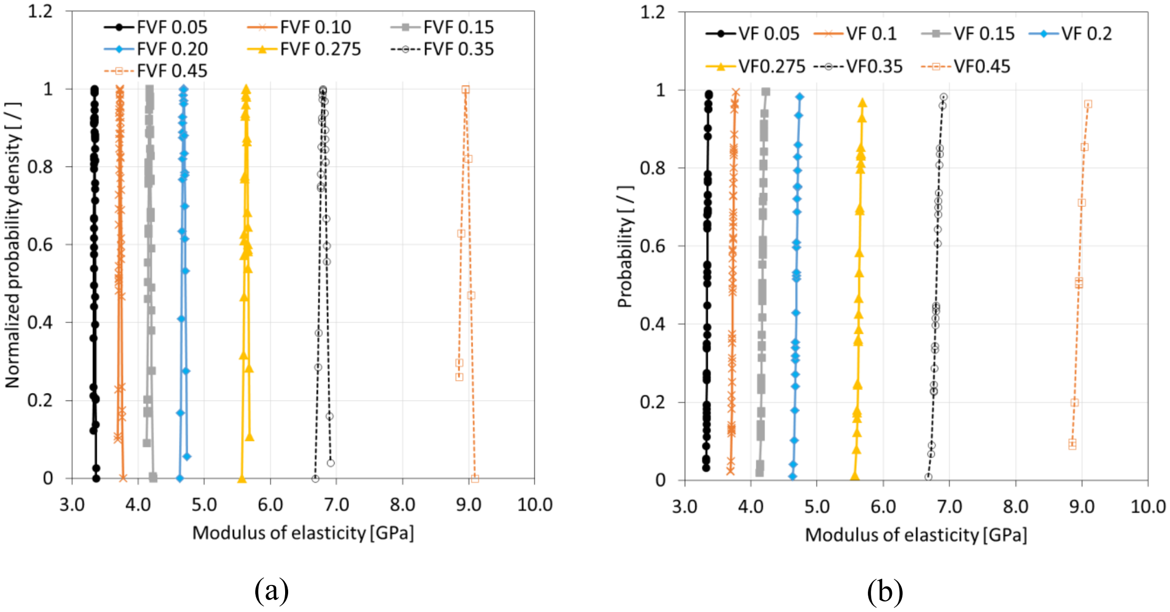
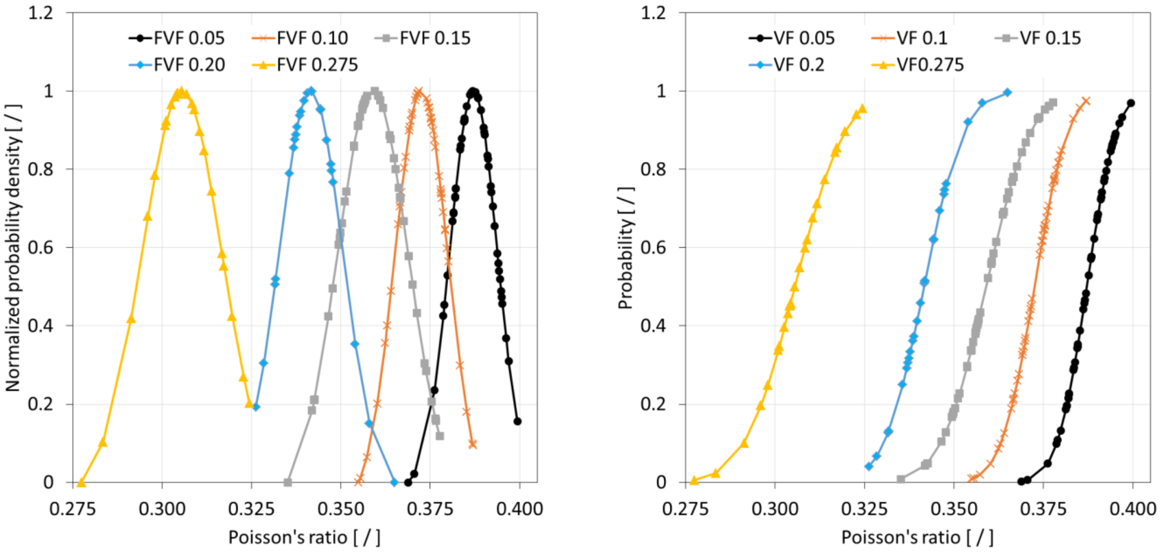
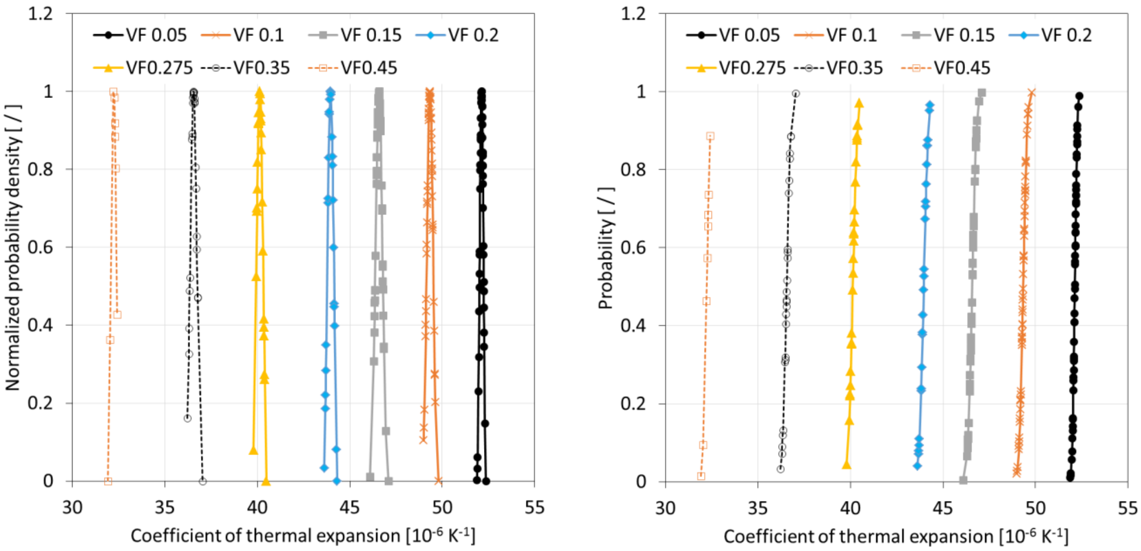


Figure 2-19 Normalized probability density (a) and cumulative distribution function (b) of modulus of elasticity data for simulated glass bead/epoxy composites.



(a) (b)

Figure 2-20 Normalized probability density (a) and cumulative distribution function (b) of Poisson’s ratio data for simulated glass bead/epoxy composites.



(a) (b)

Figure 2-21 Normalized probability density (a) and cumulative distribution function (b) of coefficient of thermal expansion data for simulated glass bead/epoxy composites.

Table 2-4 Statistical analysis of modeling results for the modulus of elasticity.

Volume Fraction	0.050	0.100	0.150	0.200	0.275	0.350	0.450
Number of iterations (/)	50	50	50	25	25	25	10
Mean (Pa)	3.34	3.728	4.173	4.688	5.635	6.798	8.951
Median (Pa)	3.34	3.728	4.173	4.69	5.633	6.791	8.952
Standard deviation (Pa)	0.01	0.017	0.02	0.024	0.028	0.053	0.075
Skewness (/)	0.381	0.186	0.361	-0.288	-0.179	-0.065	0.302
Kurtosis (/)	-0.234	-0.017	0.807	0.516	-0.452	0.57	-0.157

Table 2-5 Statistical analysis of modeling results for the Poisson's ratio.

Volume Fraction	0.050	0.100	0.150	0.200	0.275
Number of iterations (/)	50	50	50	25	25
Mean (/)	0.3871	0.3725	0.3589	0.3416	0.3055
Median (/)	0.3872	0.3742	0.3568	0.3406	0.3044
Standard deviation (/)	0.0065	0.0073	0.0098	0.0087	0.011
Skewness (/)	-0.561	-0.394	-0.001	0.762	-0.617
Kurtosis (/)	0.387	0.149	-0.347	1.169	0.865

Table 2-6 Statistical analysis of modeling results for the coefficient of thermal expansion.

Volume Fraction	0.050	0.100	0.150	0.200	0.275	0.350	0.450
Number of iterations (/)	50	50	50	25	25	25	10
Mean (C^-1)	52.12	49.34	46.58	43.96	40.17	36.55	32.24
Median (C^-1)	52.14	49.33	46.59	43.96	40.19	36.54	32.28
Standard deviation (C^-1)	0.015	0.024	0.026	0.038	0.034	0.035	0.046
Kurtosis (/)	-0.283	0.164	0.179	-0.01	-0.114	0.479	-1.41
Skewness (/)	-0.044	0.17	0.665	-0.822	-0.49	1	1.681

In a final step of the modeling data analysis, results for the effective modulus of elasticity and Poisson’s ratio were compared with experimental results published in the technical literature [19,20], as shown in Figs. 2-22 and 2-23, respectively. For the modulus of elasticity, the experiments indicate a non-linear increase with increasing FVF. It can be observed that the numerical data are in good agreement with the experiments. In fact, the predictions are within the experimental data scatter. Fig. 2-22 also includes predictions from analytical models, namely, the Mori-Tanaka and Hashin-Shtrikman approaches. It can be seen that both analytical methods deliver similar results yet underpredict the values from experimental and numerical modeling, especially for higher FVF.

Referring to Fig. 2-23, experimental data indicate a reduction in Poisson’s ratio with rising FVF. The numerical predictions follow the experimental values rather well up to a FVF of 0.15, at which point a shift in the experimental data seems to have occurred compared to its initial trend. The original publication [20] from which the experimental data was sourced from did not address nor further investigate this behavior. It is herein speculated that at higher FVF, the sample morphology may have deviated from the assumptions made in the present study, which is a random filler distribution and dispersion, leading to a reduced ability to lower the Poisson's ratio.

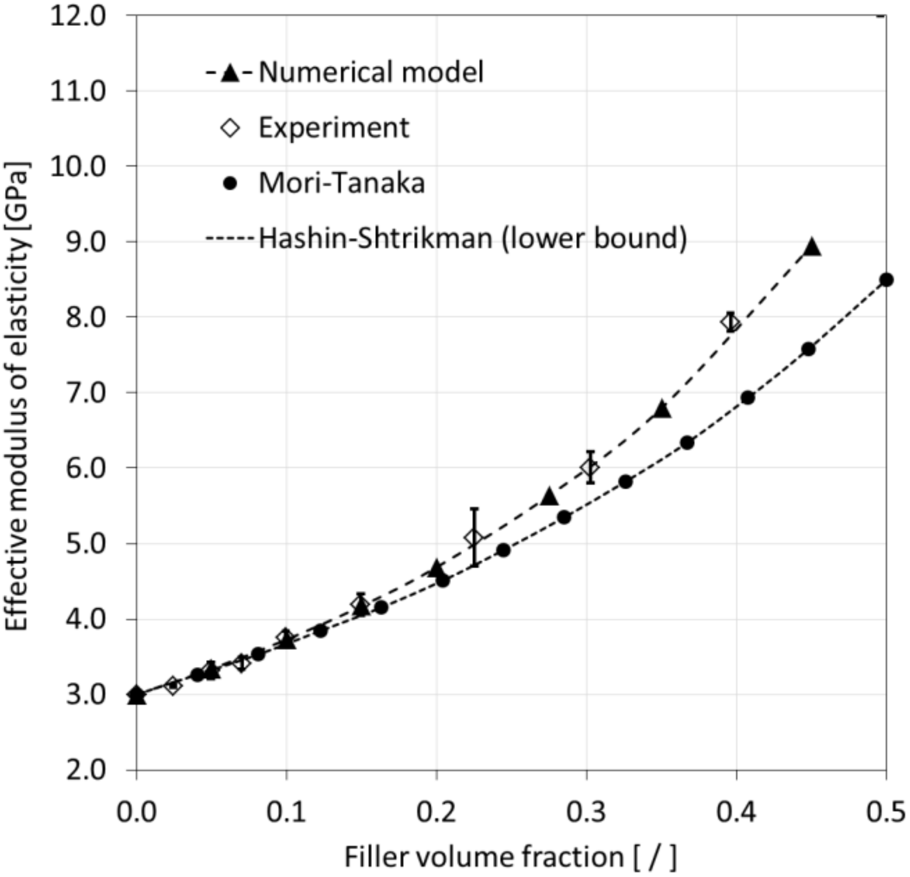


Figure 2-22 Comparison of modeling results for the modulus of elasticity with analytical model predictions and experimental data [19,20]. Error bars express experimental data scatter.

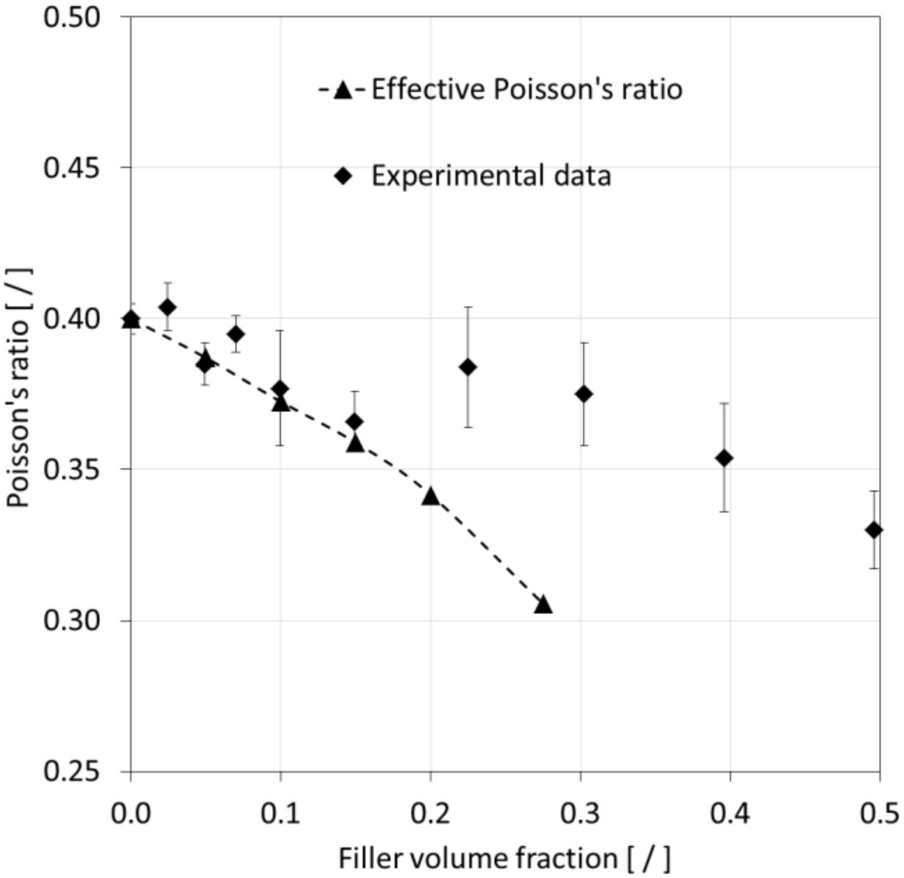


Figure 2-23 Comparison of modeling results for the Poisson's ratio with experimental data [20]. Error bars express experimental data scatter.

Numerical predictions for the effective CTE are depicted in Fig. 2-24. No relevant experimental data could be found for comparison in the technical literature. Alternatively, a basic Voigt model prediction was made as shown in the figure for comparison with the numerical modeling results. Both the numerical and the analytical predictions describe a significant reduction of CTE with increasing FVF. It can be observed that the graph associated with the numerical model is slightly non-linear, predicting a 6% lower CTE for the highest filler loading compared to the Voigt model. Interestingly, the rather basic Voigt model appears to yield acceptable predictions in this context. However, prior to generalizing such statement, it would be prudent to study particle size effects and compare data with experimental results.

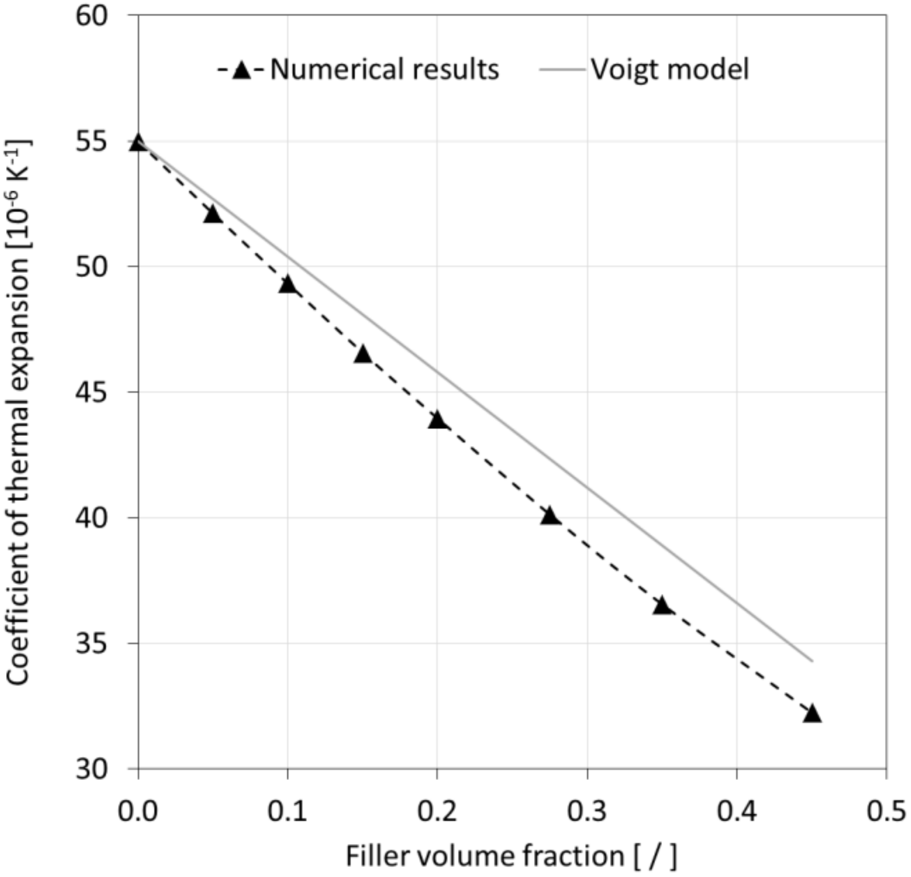


Figure 2-24 Comparison of modeling results for the coefficient of thermal expansion with Voigt model predictions.

Similar to other research work the present study confirms that filler particle and polymer matrix properties as well as filler loading significantly affect the effective composite properties. It has been shown that particle size is another important parameter, since polymer composites created with smaller particles tend to increase the effective modulus of elasticity to a greater extent compared to composites containing larger particles at the same FVF [52]. Most analytical approaches are unable to capture the influence of filler geometry and therefore fail to predict properties accurately [19,16]. Alternatively, empirical models can be fitted to predicting the properties of specific composite configurations [53]. While such an approach may increase the accuracy for material property predictions, it has drawbacks in terms of requiring a range of empirical models with limited applicability. The SFEA framework that was applied in the present study alleviates many of the shortcomings associated with analytical and empirical techniques. The SFEA framework, while being computational more intensive than analytical and empirical methods, was shown to be versatile in terms of filler and matrix material characteristics and geometries. Moreover, the developed numerical technique is capable of uniting the prediction of mechanical as well as other physical properties [48] in one model, further adding to the versatility of the presented approach. The SFEA framework is thus seen as an attractive tool for material designers. Its application in conjunction with experimental validation may thus serve to accelerate material development to meet application requirements.

2.2.5 Conclusions

A stochastic finite element analysis framework was developed enabling the property prediction of particulate filler modified polymer composites. The employed modeling approach is based on Monte Carlo principles and allows for capturing the effects of filler size distribution, filler shapes and orientations, which are important parameters for accurately predicting composite properties. While the developed analysis framework is capable of predicting a variety of mechanical and physical properties, such as thermal conductivity, the present study focused on predicting the effective modulus of elasticity, Poisson's ratio and coefficient of thermal expansion for the case of randomly distributed and dispersed spherical glass particles embedded in an epoxy polymer matrix.

The numerical results obtained from the analysis framework were compared with experimental values and analytical approaches. The numerical data was found to agree well with non-linear trends observed in the experiments, especially for the elastic modulus, in which case predictions fell within the experimental data scatter. Numerical predictions deviated from experimental Poisson's ratio data for filler volume fractions exceeding 0.15. This may be a result of morphology changes in test specimens at higher filler loadings, e.g., an increasing extent of filler agglomerations, which would deviate from assumptions made for the numerical model. Nevertheless, in general, the data from the numerical model were found to predict the experiments rather well while the employed analytical methods were less successful in predicting the test data accurately.

This study shows that numerical techniques, including the current stochastic finite element analysis framework, are attractive and effective for modeling particulate filler modified polymer composites, outperforming analytical methods in terms of versatility, and purely experimental campaigns with regard to time and cost. Given that the developed analysis framework is capable of treating binary, ternary and higher order polymer composites with randomly distributed filler particles, it is seen as an suitable tool for material designers to achieve greater accuracy predictions and use less costly and time-consuming experimentation in order to expedite the development of advanced filler modified polymer composites.

2.3 References

[1] Beret, S.; Prausnitz, J. Densities of liquid polymers at high pressure. Pressure-volume-temperature measurements for polythylene, polyisobutylene, poly(vinyl acetate), and poly(dimethylsiloxane) to 1 kbar. Macromolecules. 1975, 8, 536–538, DOI: 10.1021/ma60046a028.

[2] Manfredi, L.; Rodriguez, E.; Wladyka-Przybylak, M.; Vazquez, A. Thermal degradation and fire resistance of unsaturated polyester modified acrylic resins and their composites with

natural fibres. Polym. Degrad. Stab. 2006, 91, 255-261, DOI: 10.1016/j.polymdegradstab.2005.05.003.

[3] Drury, J.; Mutsaers, C.; Hart, C.; Matters, M.; de Leeuw D. Low-cost all-polymer integrated circuits. Appl. Phys. Lett. 1998, 73, 108-110, DOI: 10.1063/1.121783.

[4] Sheng, N.; Boyce, M.; Parks, D.; Rutledge, G.; Abes, J. Multiscale micromechanical modeling of polymer/clay nanocomposites and the effective clay particle. Polymer. 2004, 45, 487-506, DOI: 10.1016/j.polymer.2003.10.100.

[5] Eshelby, J. The determination of the elastic field of an ellipsoidal inclusion and related problems. Proc. R. Soc. London, Ser. A. 1957, 241, 561–569, DOI: 10.1098/rspa.1957.0133.

[6] Voigt, W. Ueber die Beziehung zwischen den beiden Elasticitätsconstanten isotroper Körper, Annalen der Physik, 1889, 274, 573–587, DOI: 10.1002/andp.18892741206.

[7] Reuss, A. Berechnung der Fließgrenze von Mischkristallen auf Grund der Plastizitätsbedingung für Einkristalle. Zeitschrift für Angewandte Mathematik und Mechanik 1929, 9, 49–58, DOI: 10.1002/zamm.19290090104.

[8] Hashin, Z.; Shtrikman, S. A variational approach to the theory of elastic behavior of multiphase materials. J. Mech. Phys. Solids, 1963, 11, 127–140, DOI: 10.1016/0022-5096(63)90060-7.

[9] Mori, T.; Tanaka, K. Average stress in the matrix and average elastic energy of materials with misfitting inclusions. Acta Metall. 1973, 21, 571–574, DOI: 10.1016/0001-6160(73)90064-3.

[10] Benveniste, Y. A new approach to the application of Mori–Tanaka's theory in composite materials. Mech. Mater. 1987, 6, 147–157, DOI: 10.1016/0167-6636(87)90005-6.

[11] Hill, R. A self-consistent mechanics of composite materials. J. Mech. Phys. Solids. 1965, 13, 213–222, DOI: 10.1016/0022-5096(65)90010-4.

[12] Christensen, R.; Lo, K. Solutions for effective shear properties of three phase sphere and cylinder models. J. Mech. Phys. Solids, 1987, 27, 315–330, DOI: 10.1016/0022-5096(79)90032-2.

[13] Torquato, S. Effective stiffness tensor of composite media: II. Applications to isotropic dispersions. J. Mech. Phys. Solids. 1998, 46, 1411–1440, DOI: 10.1016/S0022-5096(97)00083-5.

[14] Drugan, W.; Willis, J. A micromechanics-based nonlocal constitutive equation and estimates of the representative volume element size for elastic composites. J. Mech. Phys. Solids. 1996, 44, 497–524, DOI: 10.1016/0022-5096(96)00007-5.

[15] Drugan, W. Micromechanics-based variational estimates for a higher-order nonlocal constitutive equation and optimal choice of effective moduli of elastic composites. J. Mech. Phys. Solids. 2000, 48, 1359–1387, DOI: 10.1016/S0022-5096(99)00088-5.

[16] Segurado, J.; Llorca, J. A numerical approximation to the elastic properties of sphere-reinforced composites. J Mech Phys Solids 2002, 50, 2107-2121, DOI: 10.1016/S0022-5096(02)00021-2.

[17] Lopez, O.; Choplin, L.; Tanguy, P. Thermorheological analysis of glass beads-filled polymer melts. Polym Eng Sci 1990, 30, 821-828, DOI: 10.1002/pen.760301404.

[18] Bibikov, S.; Prokof'Ev, M. Composite Materials for some radiophysics applications. Metal, Ceramic and Polymeric Composites for Various Uses, InTech Open, 2011, DOI: 10.5772/18074.

[19] Ju, J.; Yanase, K. Micromechanics and effective elastic moduli of particle-reinforced composites with near-field particle interactions. Acta Mech. , 2010, 215, 135-153, DOI: 10.1007/s00707-010-0337-2.

[20] Smith, J. Experimental values for the elastic constants of a particulate-filled glassy polymer. J. Res. Nat. Bur. Stand. 1976, 80, 45–49, DOI: 10.6028/jres.080A.008.

[21] Ma, P.; Siddiqui, N.; Marom, G.; Kim, J. Dispersion and functionalization of carbon nanotubes for polymer-based nanocomposites: A review. Compos Part A Appl-S 2010, 41, 1345-1367, DOI: 10.1016/j.compositesa.2010.07.003.

[22] Akcora, P.; Liu, H.; Kumar, S.; Moll, J.; Brian, Y.; Benicewicz, C.; Schadler, L.; Acehan, D.; Panagiotopoulos, A.; Pryamitsyn, V.; Ganesan, V.; Ilavsky, J; Thiyagarajan, P; Colby, R; Douglas, J. Anisotropic self-assembly of spherical polymer-grafted nanoparticles. Nat Mater 2009, 8, 354-359, DOI: 10.1038/nmat2404.

[23] Li, Z.; Okamoto, K.; Ohki, Y.; Tanaka, T. Effects of nano-filler addition on partial discharge resistance and dielectric breakdown strength of Micro-Al2O3Epoxy composite. IEEE Trans Dielectr Electr Insul 2010, 17, 653-661, DOI: 10.1109/TDEI.2010.5492235.

[24] Prasad, K.; Das, B.; Maitra, U.; Ramamurty, U.; Rao, C. Extraordinary synergy in the mechanical properties of polymer matrix composites reinforced with 2 nanocarbons. PNAS 2009, 106, 13186-13189, DOI: 10.1073/pnas.0905844106.

[25] Bewlay, B. P.; Jackson, M. R.; Lipsitt, H. A. The balance of mechanical and environmental properties of a multielement niobium-niobium silicide-based in situ composite. Metall Mater Trans A 1996, 27, 3801–3808, DOI: 10.1007/BF02595629.

[26] Rana, D.; Lee, C.; Cho, K.; Lee, B.; Choe, S. Thermal and mechanical properties for binary blends of metallocene polyethylene with conventional polyolefins. J Appl Polym Sci 1998,

69, 2441–2450, DOI: 10.1002/(SICI)1097-4628(19980919)69:12<2441::AID-APP15>3.0.CO;2-%23.

[27] Sathe, S.; Devi, S.; Srinivasa, G.; Rao, K. Relationship between morphology and mechanical properties of binary and compatibilized ternary blends of polypropylene and nylon 6. J Appl Polym Sci 1996, 61, 97–107, DOI: 10.1002/(SICI)1097-4628(19960705)61:1<97::AID-APP11>3.0.CO;2-X.

[28] Halpin J. C.; Kardos, J. L. The Halpin-Tsai equations: A review. Polym Eng Sci 1976, 16, 344–352, DOI: 10.1002/pen.760160512.

[29] Ahmadi Moghaddam, H.; Mertiny, P. Stochastic finite element modeling of char-forming filler addition and alignment – Effects on heat conduction in polymer condensed phase. In Flame Retardants; Zafar, F.; Sharmin, E.; Intech Open: London, United Kingdom, 2018, DOI: 10.5772/intechopen.82878.

[30] Coleman, J.; Khan, U.; Gun'ko, Y. Mechanical reinforcement of polymers using carbon nanotubes. Adv. Mater. 2006, 18, 689-706, DOI: 10.1002/adma.200501851.

[31] Moniruzzaman, M.; Winey, K.I. Polymer nanocomposites containing carbon nanotubes. Macromolecules 2006, 39, 5194-5205, DOI: 10.1021/ma060733p.

[32] Green, M.J.; Behabtu, N.; Pasquali, M.; Adams, W.W. Nano tubes as polymer. Polymer 2009, 50, 4979-4997, DOI: 10.1016/j.polymer.2009.07.044 .

[33] Liu, J.H.; Liu, J.Y.; Yang, L.B.; Chen, X.; Zhang, M.Y.; Meng, F.L.; Luo, T.; Li, M.Q. Nanomaterial-assisted signal enhancement of hybridization for DNA biosensors: a review. Sens 2009, 9, 7343-7364, DOI: 10.3390/s90907343.

[34] Balasubramanian, K.; Burghard, M. Electrochemically functionalized carbon nanotubes for device applications. J Mater Chem 2008, 26, 452, DOI: 10.1039/B718262G.

[35] Harris, P.J.F. Carbon nanotubes and related structures: New materials for the twenty-first century. Am J Phys 2004, 72, 415, DOI: 10.1119/1.1645289.

[36] Dai L. Carbon Nanotechnology: Recent Developments in Chemistry, Physics, Materials Science and Device Applications. 1st ed. Amsterdam: Elsevier; 2006. ISBN-13: 978-0-444-51855-2.

[37] Meincke, O.; Kaempfer, D.; Weickmann, H.; Friedrich, C.; Vathauer, M.; Warth, H. Mechanical properties and electrical conductivity of carbon-nanotube filled polyamide-6 and its blends with acrylonitrile/butadiene/styrene. Polymer, 2004, 45, 739-748, DOI: 10.1016/j.polymer.2003.12.013.

[38] Valavala, P.K.; Odegard, G.M. Modeling techniques for determination of mechanical properties of polymer Nanocomposites. Adv Mater Sci 2005, 9, 34-44, DOI: 10.1.1.543.5381&rep=rep1&type.

[39] Odegard, G.; Clancy, T.; Gates, T. Modeling of the mechanical properties of nanoparticle/polymer composites. Plymer 2005, 46, 553-562, DOI: 10.1016/j.polymer.2004.11.022.

[40] Liu, Y.J.; Chen, X. Evaluations of the effective material properties of carbon nanotube-based composites using a nanoscale representative volume element. Mech Mater 2003, 35, 69–81, DOI: 10.1016/S0167-6636(02)00200-4.

[41] Maybeck, P. Stochastic models, Estimation and control, Mathematic in Science and Engineering, 141-3, ISBN: 0-12-480701-1.

[42] Haldar, A.; Mahadevan, S. Reliability assessment using stochastic finite element analysis. New York, NY, Wiley, 2000, ISBN: 978-0-471-36961-5.

[43] Koduru, S.; Haukaas, T. Feasibility of FORM in finite element reliability analysis. Struct. Saf. 2010, 32, 145–153, DOI: 10.1016/j. strusafe.2009.10.001.

[44] Thacker, B.; Riha, D.; Fitch, S.; Huyse, L.; Pleming, J. Probabilistic engineering analysis using the NESSUS software. Struct. Saf. 2006, 28, 83–107, DOI: 10.1016/j.strusafe.2004.11.003.

[45] Patelli, E.; Panayirci, M.; Broggi, M.; Goller, B.; Beaurepaire, P.; Pradlwarter, H.; Schuëller, G. General purpose software for efficient uncertainty management of large finite element models. Finite Elem. Anal. Des. 2012, 51, 31–48, DOI: 10.1016/j. finel.2011.11.003.

[46] Akula, V. Multiscale reliability analysis of a composite stiffened panel. Compos. Struct. 2014, 116, 432–440, DOI: 10.1016/j. compstruct.2014.06.001.

[47] McKenna, F.; Fenves, G.; Scott, M. Open system for earthquake engineering simulation. Berkeley, CA: University of California. Retrieved from http://opensees.berkeley.edu, 2000.

[48] Ahmadi Moghaddam, H.; Mertiny, P. Stochastic finite element analysis framework for modelling thermal conductivity of particulate modified polymer composites. Results Phys 2018, 11, 905-914, DOI: 10.1016/j.rinp.2018.10.045.

[49] Technical Data Sheet 758, Glass Beads, Polysciences, Inc., 2016.

[50] Jeyranpour, F.; Alahyarizadeh, Gh.; Arab, B. Comparative investigation of thermal and mechanical properties of cross-linked epoxy polymers with different curing agents by molecular dynamics simulation. J Mol Graph Model 2015, 62, 157-164, DOI: 10.1016/j.jmgm.2015.09.012.

[51] West, S. G.; Finch, J.F.; Curran, P. J. Structural equation models with nonnormal variables: Problems and remedies. In Structural Equation Modeling: Concepts, Issues, and

Applications; Hoyle, R.H., Ed.; Sage Publications: Thousand Oaks, CA, USA, 1995, pp. 56-75.

[52] Deng, F.; Van Vliet, K.J. Prediction of elastic properties for polymer–particle nanocomposites exhibiting an interphase. Nanotechnol 2001, 22, 165703, DOI: 10.1088/0957-4484/22/16/165703.

[53] Upadhyay, A.; Singh, R. Prediction of Effective Elastic Modulus of Biphasic Composite Materials. Mod Mech Eng 2012, 2, 6-13, DOI: 10.4236/mme.2012.21002.

Chapter 3

Thermal Characterization of Randomly Distributed Particulate Polymer Composites

As a second part of this research study, a stochastic FEA framework was developed that enabled calculating effective thermal conductivity of spherical particulate polymer composites. This work was published as publication no. 3 listed in the Preface of this thesis document.

Nomenclature

ϕ	filler volume fraction
q	heat flux
k_{f}	filler thermal conductivity
TCC	thermal contact conductance
k_{m}	Matrix thermal conductivity
T_{hot}	hot temperature
k_{eff}	effective thermal conductivity
T_{cold}	cold temperature
k_{fr}	filler replacement thermal conductivity
K_i	node thermal conductivity
d	spherical particle diameter
l	RVE length
h	thermal contact conductance
T_2	RVE front surface temperature

x	random variables
T_1	RVE front surface temperature
$P(x)$	probability density
Q_i	node heat flux
N	sample sets
K_{eff}	Effective thermal conductivity
$E(f)$	quantity of interest
i	node number
$f(x)$	scalar quantity
n	Total number of nodes
$V_{\mathrm{B}m}$	spherical particles volume
m	bin number
$r_{\mathrm{B}m}$	particle size
n	particle size distribution frequency
V_{BT}	particles total volume
$R_{\mathrm{B}m}$	particle volume ratio
$\varphi_{\mathrm{B}m}$	bin volume fraction
R	thermal resistance
$\overline{\Delta T}$	average temperature drop
Q	heat flux

3.1 Introduction

Polymers are used in a broad range of applications due to their attractive properties, including low volumetric mass density [1], high degradation resistance against corrosion, and comparatively low cost [2-3]. However, the performance of polymers is typically limited in terms of mechanical, thermal and electrical properties. One possible approach for overcoming these limitations is to enhance polymer properties by incorporating appropriate fillers [4]. In recent years, considerable progress has been made in terms of developing filler modified polymer composites with well dispersed and distributed particles [5-6]. A large selection of micro- and nano-scale particulate fillers is available to material designers for modifying polymers with different filler morphologies and filler volume fractions. However, designing a filler modified polymer is typically a time-consuming process and a challenging task to perform, given the vast design space encompassing various filler types, materials, volume fractions, polymer matrices, and manufacturing routes that affect filler dispersion, distribution and alignment [7].

The addition of suitable micro- and nano-particulate fillers into a polymer matrix is intuitively associated with changes of the composite mechanical and thermal properties over the pure polymer. A variety of analytical, empirical and experimental approaches have been developed to characterize the properties of filler-modified polymers, such as thermal conductivity [8-10]. In general, adding fillers with high thermal conductivity improves heat conduction in polymer composites. However, the magnitude of thermal conductivity enhancement strongly depends on certain key parameters, i.e., the filler thermal conductivity [11], the state of filler dispersion in the matrix [11-13], the interfacial interaction between filler particles and the matrix polymer [14], and other aspects of composite morphology, such as particle size, shape and orientation [15].

The technical literature provides several examples for applications of modified polymers with enhanced thermal properties, e.g., for removing thermal energy from a heat source [16-21], and for utilization in semiconductor devices [22-23]. Conversely, instead of using highly thermally conductivity fillers that cause the composite effective thermal conductivity to increase, fillers with low thermal conductivity may also be used to reduce the effective thermal conductivity to create thermal insulation materials, for example, thermally durable composites for heat barrier components in heaters [24-25].

Analytical approaches are generally seen as most convenient for predicting material properties. Maxwell pioneered an analytical approach for predicting the effective thermal conductivity of a composite [26] where thermal interaction between particles was ignored [27]. The Maxwell model for predicting the effective thermal conductivity of a continues matrix with embedded spherical particles can be expressed by Eq. (3-1).

$$\frac{k_{\mathrm{eff}}}{k_{\mathrm{m}}} = 1 + \frac{3\phi}{\left(\frac{k_{\mathrm{f}} + 2k_{\mathrm{m}}}{k_{\mathrm{f}} - k_{\mathrm{m}}}\right) - \phi} \tag{3-1}$$

where ϕ is filler volume fraction, and k_{f} and k_{m} are the thermal conductivity of the filler and the polymer matrix, respectively. The Maxwell expression was extended by Eucken [28] for calculating the thermal conductivity of multi-phase filler modified polymers. Other analytical approaches include Burger [29] and Hamilton and Crosser [30], who expanded the analysis to particles shapes other than spherical. Some of the analytical approaches consider an array of particles in a matrix domain. For example, Rayleigh [31] proposed a model with an array of spherical particles embedded in a cubic matrix element. In this model, as expressed in Eq. (3-2), direct thermal interaction between particles can be considered.

$$\frac{k_{\mathrm{eff}}}{k_{\mathrm{m}}}=1+\frac{3\phi}{\left(\frac{k_{\mathrm{f}}+2k_{\mathrm{m}}}{k_{\mathrm{f}}-k_{\mathrm{m}}}\right)-\phi+1.569\left(\frac{k_{\mathrm{f}}-k_{\mathrm{m}}}{3k_{\mathrm{f}}-4k_{\mathrm{m}}}\right)\phi^{\frac{10}{3}}+\cdots} \tag{3-2}$$

Bohm et al. [32] used Mori-Tanaka models based on mean field theories for predicting thermal conductivity of spherical particulate composites with interfacial resistance as shown in Eq. (3-3).

$$k_{\mathrm{eff}}=k_{\mathrm{m}}+\frac{3\phi k_{\mathrm{m}}(k_{\mathrm{fr}}-k_{\mathrm{m}})}{3k_{\mathrm{m}}+(1-\phi)(k_{\mathrm{fr}}-k_{\mathrm{m}})} \tag{3-3}$$

where k_{fr} is replacement thermal conductivity of filler that can be calculated using Eq. (3-4).

$$k_{\mathrm{fr}}=k_{\mathrm{f}}\frac{dh}{dh+2k_{\mathrm{f}}} \tag{3-4}$$

where d and h are the diameter of spherical particle and thermal contact conductance, respectively. The proposed model was equivalent to Hasselman-Johnson model [33] based on Maxwell analytical approach.

Despite their expediency, analytical methods such as the ones described above, are limited in their ability to capture actual composite morphologies and material interactions, and hence, they lack in accuracy, especially for higher filler volume fractions. Alternatively, experimental methods can be used to assess the properties of filler modified composites. However, fabricating and testing samples is time-consuming and costly, and hence, a dependency solely on experimental methods would severely hamper efforts to comprehensively characterize advanced material systems [34]. Moreover, the number of applications that require designing advanced materials is growing rapidly, and therefore, efficient methods that accurately and cost-effectively predict their properties are needed in addition to analytical and experimental approaches [35-37].

Heat is transferred through solid materials by means of phonons, i.e. lattice vibration waves, and possibly by electrons [38-39]. In electrically insulative materials, such as typical polymers, heat transfer occurs solely by phonon transport. Interfacial thermal resistance (ITR), sometimes referred to as Kapitza resistance, may occur at material interfaces where phonon transmission is impeded. ITR is an important phenomenon that impacts thermal flow and conductivity in polymer composites [40]. Most analytical methods are unable to capture ITR effects, and, as a result, are not capable of predicting thermal properties correctly. In addition to ITR, direct thermal interaction between particles may occur, which is ignored in most analytical approaches. Considering thermal interaction between particles would further increase the accuracy of thermal conductivity predictions. Consequently, the development of modeling approaches that encompass ITR and direct thermal interaction between particles is a desirable and significant effort in order to enable accurate thermal conductivity predictions for filler-modified polymer composites.

Stochastic analyses are used for a variety of applications that involve random values and uncertain predictions. Associated models are solved using statistical methods, providing means for estimating outcomes by incorporating input data from probability distributions or random patterns. As a consequence, a stochastic model produces a set of distributed results for a specific problem. In fact, the power of stochastic analysis is its capability of considering a variety of outcomes, which is critical for applications where variables are subject to temporal and/or spatial changes. The stochastic model thus allows for considering uncertainty in the analysis. Notably, the variation of inputs may not be independent from each other for some applications, and therefore, outcomes become challenging to predict. Stochastic models are able to solve such problems by running a large number of simulations that are known as stochastic projections [41]. Typical engineering analyses merely consider explicit input parameters and then deliver a distinct output. However, strictly speaking, explicit values do not exist for any event or parameter, but a probability distribution does, for the design of structures, prediction of material properties, and the like. Therefore, using a method of analysis that enables applying uncertainty to engineering models is compelling. Also, as engineering design is becoming increasingly complex in terms of input parameter variety, the need of developing stochastic analysis methods is becoming more prevalent. Stochastic finite element analysis (SFEA) is a powerful methodology which recently has been developed for analyzing problems that cannot be captured suitably with classic analytical approaches [42-43]. As the name implies, SFEA is capable of applying uncertainty to engineering problems. In the context of thermal conductivity prediction of filler modified polymers, there are several random variables involved in the material modeling, including particle location, orientation and size distribution. SFEA is therefore considered a potent method for predicting thermal conductivity by applying uncertainty to these variables.

Researchers are provided with several options for linking stochastic methods and finite element analysis (FEA). The first option is using FEA software tools that facilitate the modeling process via built-in scripting and stochastic capabilities [44-46]. Performing SFEA is comparatively straightforward using such software products since programming needs are limited and processes are largely automated. The second option is advance scripting using suitable programming languages to link stochastic analysis with FEA. This method provides greater flexibility and control to the researcher for applying uncertainty to the analysis and choosing suitable modeling processes, such as random number generators [47]. In many instances these aspects are important for the success of a SFEA study. In addition to scripting capabilities, high performance computing capabilities are recommended, if not required, for efficient data processing and acceptable processing times.

SFEA further requires performing a suitably large number of model iterations and use of statistical analyses in order to deliver meaningful results and their correct interpretation for a given problem. SFEA are commonly conducted in the context of a Monte Carlo simulation, i.e. the method of forming a ‘scientific guess’ about outcomes by using principles of inferential statistics which yield “a random sample that tends to exhibit the same properties as the population from which it is drawn” [48]. Monte Carlo simulations require sample size and sample variance to be carefully considered for achieving an acceptable level of confidence. In general, for a growing sample variance a larger sample size is needed for maintaining an acceptable level of confidence. This concept is associated to the Law of Large Numbers [49-50] which indicates that repeating the same analysis with random input variables a large number of times increases the chance of achieving a result closer to the population average. As such, a Monte Carlo simulation approximates an output data distribution by using a large number of samples as described by Eq. (3-5). These samples are also used for computing expectations according to Eq. (3-6).

$$P_N(x) = \frac{1}{N}\sum_{i=1}^{N} 1\left(x^{(i)} = x\right) \qquad \overrightarrow{N \to \infty} \qquad P(x) \tag{3-5}$$

$$E_N(f) = \frac{1}{N}\sum_{i=1}^{N} f\left(x^{(i)}\right) \qquad \overrightarrow{N \to \infty} \qquad E(f) = \sum_{x} f(x)p(x) \tag{3-6}$$

where x, $P(x)$ and N are a collection of random variables, a probability density or distribution, and set of samples, respectively. $E(f)$ and $f(x)$ are correspondingly the quantity of interest and scalar quantity determined by x.

Selecting an appropriate algorithm for random number generation is an important step in Monte Carlo simulation for applying randomness or uncertainty to input variables in the model. A pseudo-random number generator (PRNG) may be used for generating a set of numbers in order to estimate properties of sequences of random numbers [51]. PRNG is also called a deterministic random bit generator (DRBG) since a generated set of random numbers depends on an initial value known as “seed”. This concept indicates that using the same seed in every iteration will generate identical sets of random numbers. Therefore, PRNG does not necessarily generate true randomness, and it is required to shuffle the seed before performing a new iteration. It is usually recommended to incorporate a clock function to generate the initial seed. MATLAB (MathWorks, Natick, Massachusetts, USA) is a numerical computing environment that has PRNG capabilities for generating true random numbers that describe specific distribution types. A quasi-random number generator (QRNG) (also known as low-discrepancy sequence method) is another algorithm that generates random numbers that are distant from numbers existing in the set. While this algorithm can prevent clustering of generated random numbers, it increases the chance of generating uniform distributions, and hence, random numbers generated with this method often cannot pass a randomness test. It is therefore recommended to use good-quality a PRNG [52].

Based on the preceding considerations a comprehensive SFEA modeling framework was developed. The overarching objective of this research work is to predict properties of particulate filler modified polymer composites allowing for a variety of composite and filler morphologies. In the context of thermal conductivity it is hypothesized that the properties of particulate filler-modified composites can be predicted with the proposed methodology when considering filler-matrix and particle-to-particle interfacial effects. The developed framework is described in the following, and its capabilities are demonstrated by predicting the thermal conductivity of a polymer composites with spherical particles that are highly thermally conductive and well distributed in the matrix. It should be emphasized that the developed modeling framework is not merely limited to the specifics of the analysis presented in this article. In fact, the framework can be adapted to other filler and composite morphologies and mechanical and physical properties predictions.

3.2 Stochastic FEA Framework

Several programming environments were employed to create the SFEA framework, which elemental structure is schematized in Fig. 3-1. The modules that are depicted in the figure and their interaction for predicting effective thermal conductivity are described in the following.

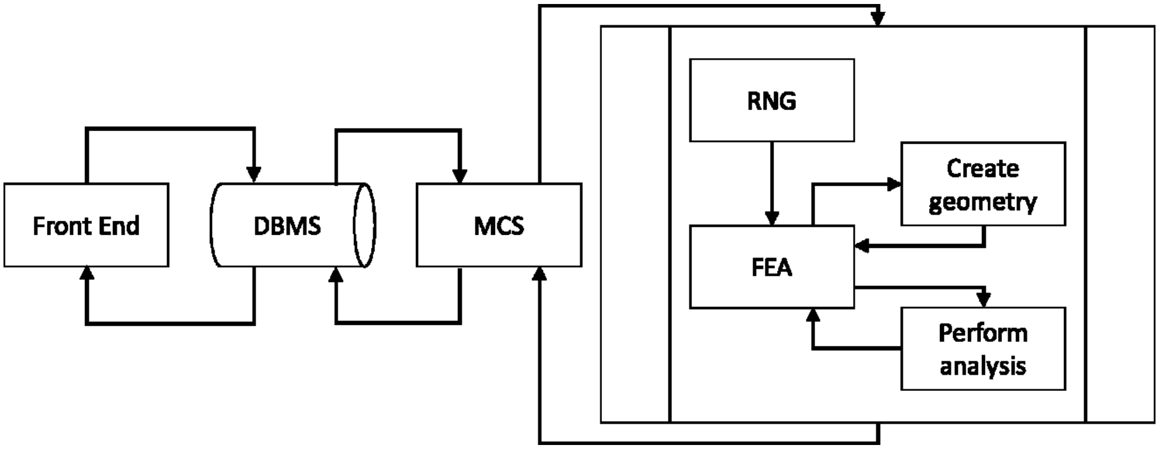


Figure 3-1 Schematic of the SFEA framework showing connections between individual modules.

Visual Basic for Applications (VBA; Microsoft, Redmond, Washington, USA) programming language was employed for creating the main domain that connects the various modules of the framework. A user-facing ‘Front End’ was designed for the input of data required for the analysis. Data captured by the Front End include the size of the representative volume element (RVE) (i.e. the modeling domain), the desired filler volume fraction, the thermal properties of the filler and matrix, information on the filler size distribution, thermal boundary conditions, parameters defining the contact between filler and matrix, and information for FEA model mesh generation. These data are stored in a database. An Open Database Connectivity (ODBC) method was used for accessing the Database Management System (DBMS). This approach enabled creating an operating system that was independent of the database system, thus providing more flexibility in terms of accessing the database at any time in the analysis, which was an important key to successfully performing the SFEA. The DBMS module transfers all required input parameters to the Monte Carlo Simulation (MCS) module for performing the SFEA. The MCS module, built using VBA programming language, was developed in tabulated format, which enables storing input parameters and results calculated by the FEA module as shown in Fig. 3-2. The MSC module repeats the subprocess shown in the schematic in Fig. 3-1 and performs required statistical analyses, i.e. calculating mean values, standard deviations and variances, for accepting final results. Once the exit criterion is satisfied the MCS module stops and transfers the final values to the database, which can be accessed via the Front End. An exit criterion can either be a statistical threshold (e.g. a certain standard deviation) or a certain number of module iterations.

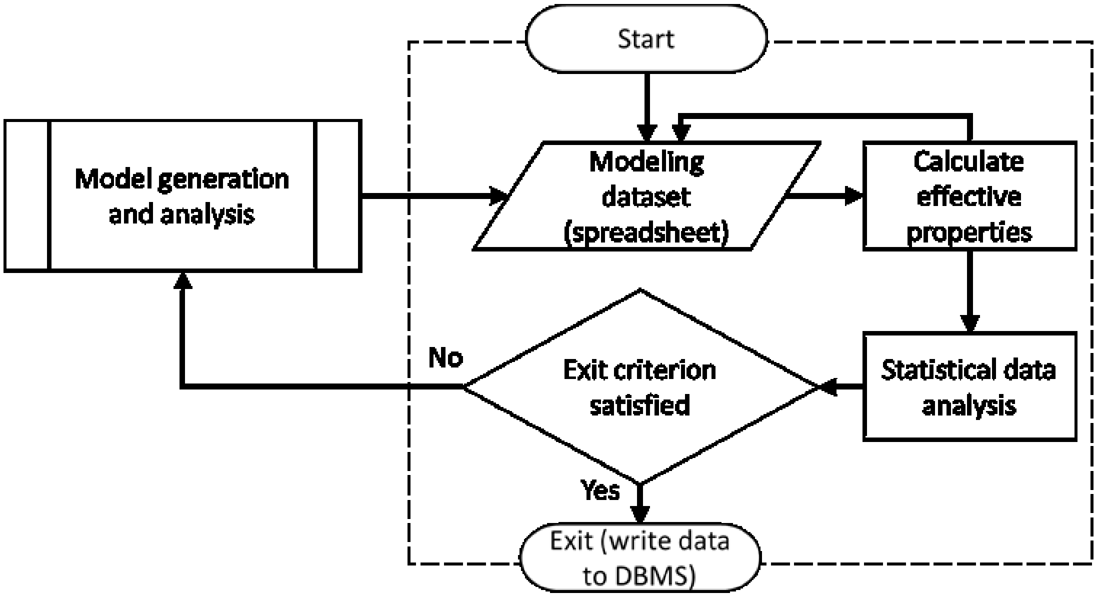

Start
Model generation and analysis
Modeling dataset (spreadsheet)
Calculate effective properties
No
Exit criterion satisfied
Statistical data analysis
Yes
Exit (write data to DBMS)

Figure 3-2 Schematic illustrating the algorithm for the Monte Carlo Simulation (MCS) module.

The subprocess algorithm from Fig. 3-1, depicted in detail in Fig. 3-3, was developed in the general mathematical programming environment MATLAB. It starts with running the Random Number Generator (RNG) module. One of the functions of this module is generating random numbers that are required for creating the composite morphology, which in the present context are particle size and location in the RVE (note that information on particle orientation is not required for spherical particles).

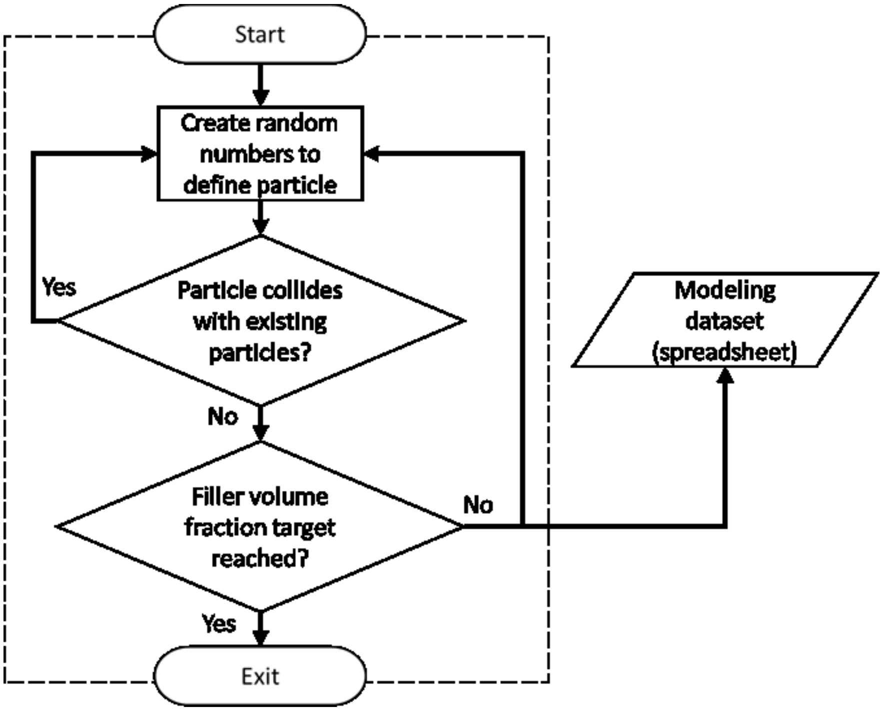

Start
Create random numbers to define particle
Particle collides with existing particles?
Yes
No
Filler volume fraction target reached?
No
Yes
Exit
Modeling dataset (spreadsheet)

Figure 3-3 Schematic illustrating the Random Number Generator (RNG) module.

An objective of the developed analysis framework is to have the generated particle size data conform to a given size distribution. This was accomplished by implementing ‘bins’ where each bin “m” represents an explicit particle size and volume fraction within the RVE. For example, a total number of bins M = 5 is considered acceptable for most modeling tasks, yet a greater number of bins can be implemented if needed. The method used for calculating the volume fraction for each bin is expressed by Eqs. (3-7) to (3-10).

$$V_{\mathrm{B}m} = n \times \frac{4}{3} \pi {r_{\mathrm{B}m}}^3 \qquad (3\text{-}7)$$

where $V_{\mathrm{B}m}$ is the volume of spherical particles in bin m, and $r_{\mathrm{B}m}$ and n are the associated particle size (radius) and frequency in the particle size distribution, respectively. The total volume V_{BT} of all particles in the complete particle distribution is obtained using Eq. (3-8).

$$V_{\mathrm{BT}} = V_{\mathrm{B1}} + V_{\mathrm{B2}} + \cdots + V_{\mathrm{B}m} \tag{3-8}$$

Hence, the ratio of particle volume in bin m with respect to the total particle volume is calculated as expressed in Eq. (3-9).

$$R_{\mathrm{B}m} = \frac{V_{\mathrm{B}m}}{V_{\mathrm{BT}}} \tag{3-9}$$

Finally, the particle volume fraction with respect to the RVE for a certain bin m, $\varphi_{\mathrm{B}m}$, is found as indicated by Eq. (3-10).

$$\varphi_{\mathrm{B}m} = \varphi \times R_{\mathrm{B}m} \tag{3-10}$$

where φ is the particle volume fraction for the RVE. Calculating the particle volume fractions for other bins using Eq. (3-10) and adding them together can be written as Eq. (3-11).

$$\varphi = \varphi_{\mathrm{B1}} + \varphi_{\mathrm{B2}} + \cdots + \varphi_{\mathrm{B}m} \tag{3-11}$$

Different approaches exist for distributing the generated particles within the RVE, i.e. random or size-order addition of particles. Skinner et al. [53] demonstrated that adding particles in descending size order has computational advantages compared to other methods. Therefore, this approach was incorporated in the RNG module, i.e. bins are processed in the order of descending particle size.

The RNG module also performs a collision detection between particles. In the case of particle intersection, the most recent particle is discarded and a new one is created. Once the data describing the set of particles are complete, these data are transferred to the FEA module in tabular format to be used for creating the FEA geometry.

Using scripting in IronPython programming language the fully custom FEA module depicted in Fig. 3-4 was developed employing the FEA software ANSYS Workbench (Version 19, ANSYS Inc., Canonsburg, PA, USA). This module consists of two different components. The first component, which was realized using ANSYS DesignModeler in conjunction with JAVA programming, creates the parametric three-dimensional geometry for the RVE and all filler particles based on the output from the RNG module. Once the model geometry is defined it is transferred to the analysis component, which was also created using JAVA programming. The analysis component automates the process of setting up the finite element analysis, i.e. assigning material properties for particles and matrix, establishing (thermal) boundary conditions, defining interface conditions for simulating ITR as well as direct particle-to-particle heat transfer, and meshing the model. After successful mesh creation the analysis component performs a convergence study by increasing mesh density and re-analyzing the model until results converge satisfactorily. Finally, results are transferred to the MCS module where they are stored and subjected to statistical analyses.

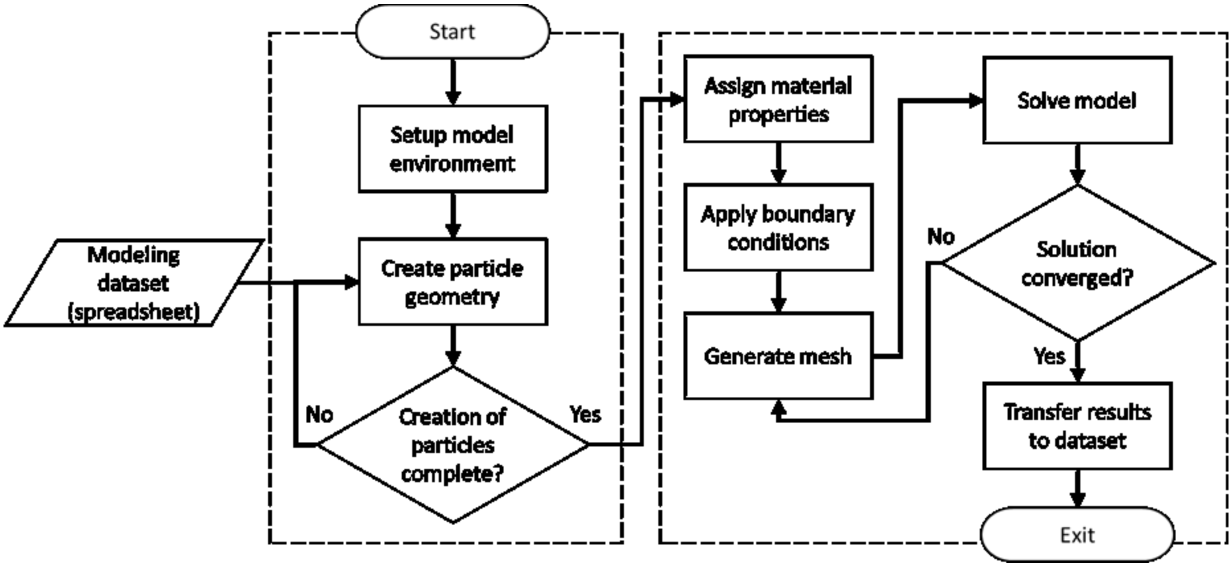

Start
Setup model environment
Modeling dataset (spreadsheet)
Create particle geometry
No
Creation of particles complete?
Yes
Assign material properties
Apply boundary conditions
Generate mesh
Solve model
No
Solution converged?
Yes
Transfer results to dataset
Exit

Figure 3-4 Schematic illustrating the algorithm for the FEA module.

3.3 Steady-State Thermal Numerical Model

The developed SFEA framework was employed to predict the thermal conductivity of polymer composites with randomly distributed spherical particles under steady-state conditions. Specifically, a polyetheretherketone (PEEK) polymer matrix modified with spherical silver nanoparticles was considered. The thermal properties of the materials were adopted from literature [54] and are shown in Table 3-1. Particles were modeled to have a mean diameter of 100 nm with a particle size variation of ±5 percent from the mean. Due to the comparatively narrow range of the particle size distribution, only a single 'bin' was implemented. This bin was populated employing uniformly distributed random numbers. The RVE size was set to 2,000 nm.

Three-dimensional ten-node quadratic tetrahedral thermal solid elements (SOLID87) were used for meshing both the particles and the polymer matrix. This element type provides one degree of freedom in terms of temperature and is well suited for generating irregular and complex meshes. As an example Fig. 3-5 shows the finite element mesh of an RVE and the filler particles included in the RVE devoid of matrix.

Table 3-1 Thermal properties of filler particles and polymer matrix [54].

Property	PEEK	Silver
Density ($kg{\cdot}m^{-1}$)	1,280	10,300
Thermal Conductivity ($W{\cdot}m^{-1}{\cdot}K^{-1}$)	0.23	429

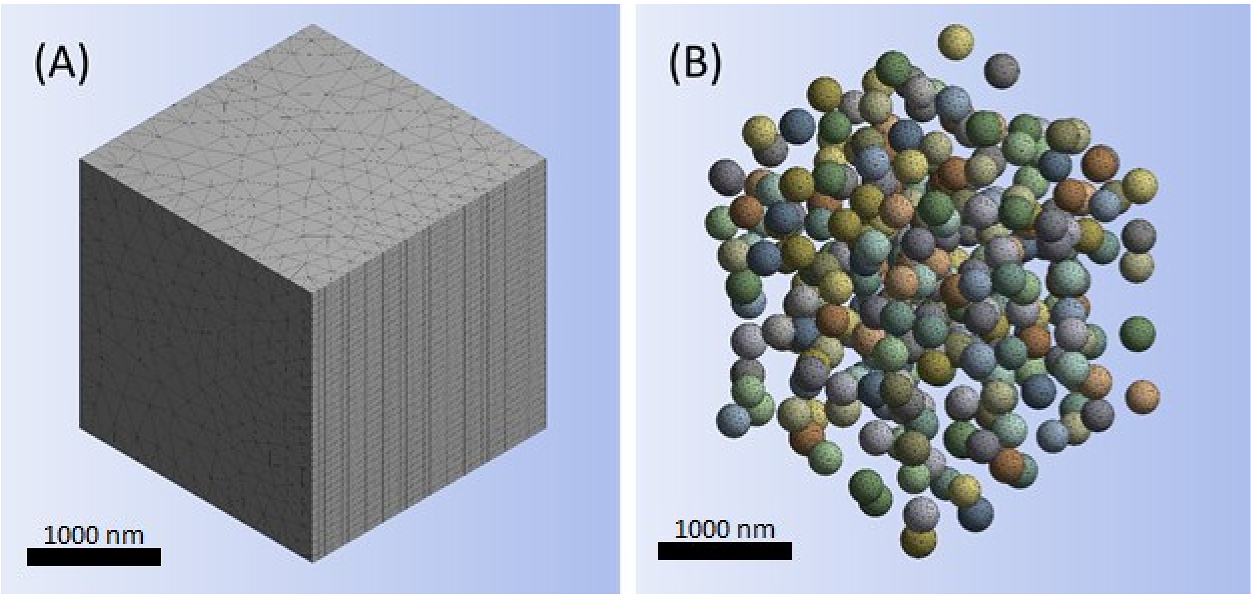


Figure 3-5 RVE finite element mesh (A) and filler particles included in RVE devoid matrix (B).

As mentioned above, modeling the ITR between particles and polymer matrix is crucial to fully capture the material behavior. This was accomplished by generating an interfacial contact between the particles and the polymer matrix using a mesh of three-dimensional six-node quadratic surface-to-surface elements (CONTA174). This approach considers the thermal contact conductance (TCC, in units of W·m^{-2}·K^{-1}). In general, the thermal resistance of a body, R, is a function of surface geometry and can be calculated using Eq. (3-12).

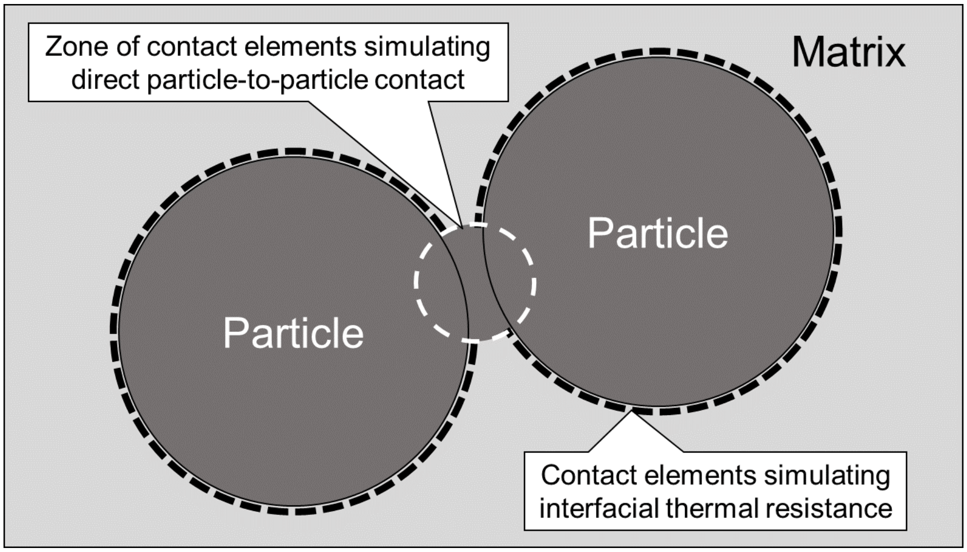
$$R = \frac{\overline{\Delta T}}{Q} \tag{3-12}$$

where $\overline{\Delta T}$ and Q are the average temperature drop on thermal body and the heat flowing through thermal body, respectively. Eq. (3-13) was integrated in order to yield the TCC captured by the surface-to-surface elements.

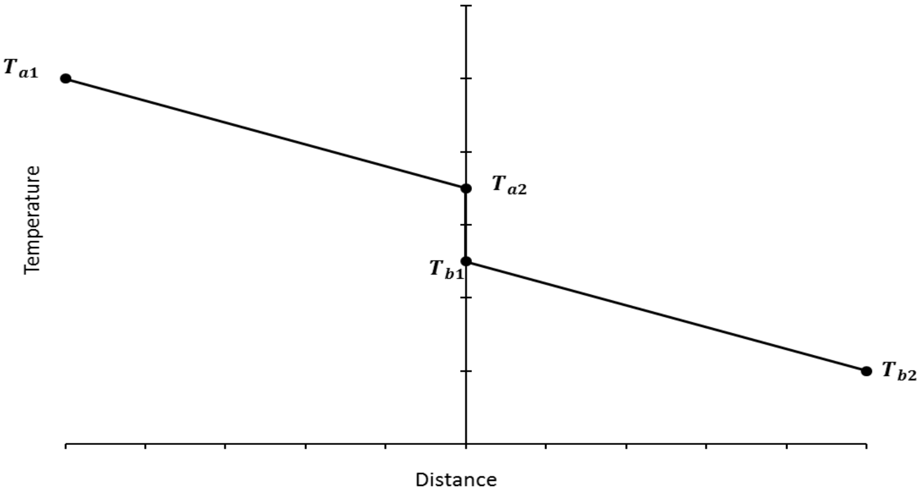
$$q = TCC\ (T_{\mathrm{hot}} - T_{\mathrm{cold}}) \tag{3-13}$$

where T_{hot} and T_{cold} are the temperature of the hot and cold surface, and q is heat flux. The local temperatures are taken as integration points at each element. The modeling framework assumes that the TCC is a constant value for a smooth particle-to-matrix contact, which needs to be obtained from e.g. experiments or other modeling activities. The present study obtained the TCC value from experimental data, which is further described in the 'Results and Discussion' section.

As mention earlier in this text, particles may be located in close proximity suggesting direct particle-to-particle contact and heat transfer. This condition was observed rather frequently in the present case where filler volume fractions ranged up 15.6%. Notably, this phenomenon was observed in other studies as well, e.g. [55]. Since the algorithm of the RNG module does not permit particle collisions, gaps are always created between particles. This constraint prevents direct particle-to-particle heat transfer and may cause artificially reduced thermal conductivity predictions. To overcome this problem an automated method was developed that measures the distance between particles and also between particles and the RVE surfaces. If a measured gap distance is less than a specific threshold value (in this study a distance of 18 nm was selected) a contact permitting direct heat transfer was defined between the identified two objects. This contact was created using a three-dimensional 6-node quadratic surface-to-surface element type (TARGE170), eliminating a temperature drop and transferring heat between the two objects completely. Fig. 3-6 illustrates the modeling approach for capturing ITR and direct particle-to-particle heat transfer. For the latter, a zone with radius equal to the specified gap size threshold value measured from the point of closest proximity between two particles was implemented, see Fig. 3.6A. Figure 3.6B depicts the temperature drop caused by ITR at the contact between particle and matrix (T_{a2} to T_{b1}), while gradual temperature changes T_{a1} to T_{a2} and T_{b1} to T_{b2} are the result of the thermal conductivities of the materials.


Zone of contact elements simulating direct particle-to-particle contact
Matrix
Particle
Particle
Contact elements simulating interfacial thermal resistance

(A)



(B)

Figure 3-6 Illustration of modeling approach for capturing ITR and direct particle-to-particle heat transfer (A), temperature distribution at the contact interface (B).

Thermal boundary conditions for the RVE were applied as follows. Four sides of the RVE were considered adiabatic. A warm and cold temperature was defined on two opposite RVE faces. The warm and cold temperature were 32°C and 22°C, respectively. The applied RVE boundary conditions create a heat flux between the warm and cold RVE faces. As illustrated by Fig. 3-7 this heat flux can be determined for any parallel plane between the warm and cold RVE faces assuming steady-state conditions. In present analysis the effective thermal conductivity of the RVE was calculated according to Eq. (3-14) considering all nodes located in the ‘warm’ RVE face.

$$K_i = \frac{Q_i \times l}{T_2 - T_1} \tag{3-14}$$

where Q_i is the total heat flux at the i^{th} node on the ‘warm’ RVE face, l is the RVE length, and T_1 and T_2 are the temperature on the opposing warm and cold RVE faces, respectively. Finally the effective thermal conductivity, K_{eff}, is obtained using Eq. (3-15).

$$K_{\text{eff}} = \frac{\sum_{i=1}^{n} K_i}{n} \tag{3-15}$$

where K_i and n are the thermal conductivity of the i^{th} node and total number of nodes on the ‘warm’ RVE face.

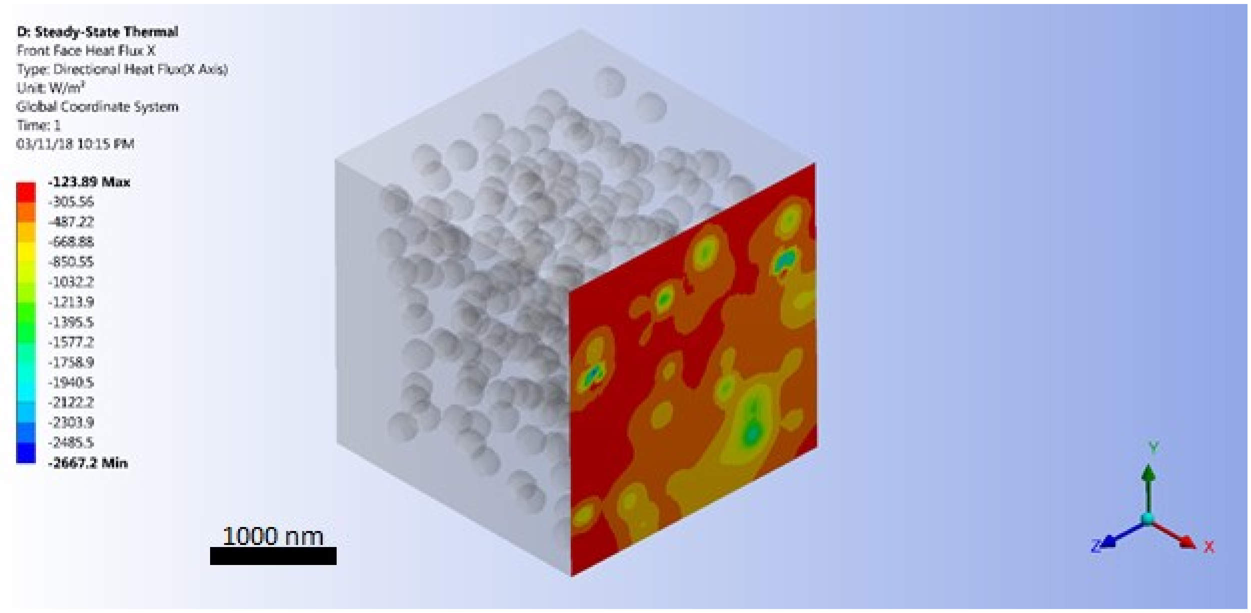


Figure 3-7 Illustration of heat flux across a parallel plane between the warm and cold RVE faces.

3.4 Result and Discussions

Using the above material properties and dimensional information the effective thermal conductivity for a PEEK-matrix/Ag-particle composite was computed for filler volume fractions of 1.30%, 4.87%, 8.00%, 13.2% and 15.6%. First, a convergence study was performed for an 0.08 filler volume fraction in order to check the effective thermal conductivity result accuracy. For this reason, the effective thermal conductivity was calculated and extracted from RVE front surface for different level of mesh refinement and finally results were plotted as shown in Fig. 3-8. Result clearly shows that the change in effective thermal conductivity was less than 5% for an increment of approximately 41,000 degrees of freedom. As the result of this mesh refinement, it was decided to choose the global mesh size of 2.5% of the RVE length. Since in SFEA approach a lot of iterations are needed to be performed, it was decided to keep this configuration for generating the mesh and tried to keep the number of nodes below 300,000 degrees of freedom that enabled minimizing time required for solving the finite element models.

Second, computed data was examined to assess the spatial particle distribution. As illustrated by the sample data in Fig. 3-9, particle centers were observed to be well dispersed and distributed within the RVE, and particle clustering is negligible. The observed composite morphology was anticipated as it is considered representative of actual spherical filler modified composites. The coordinates indicated in Fig. 3-9 correspond to the Cartesian coordinate system that measures the RVE from an origin in one of its corners.

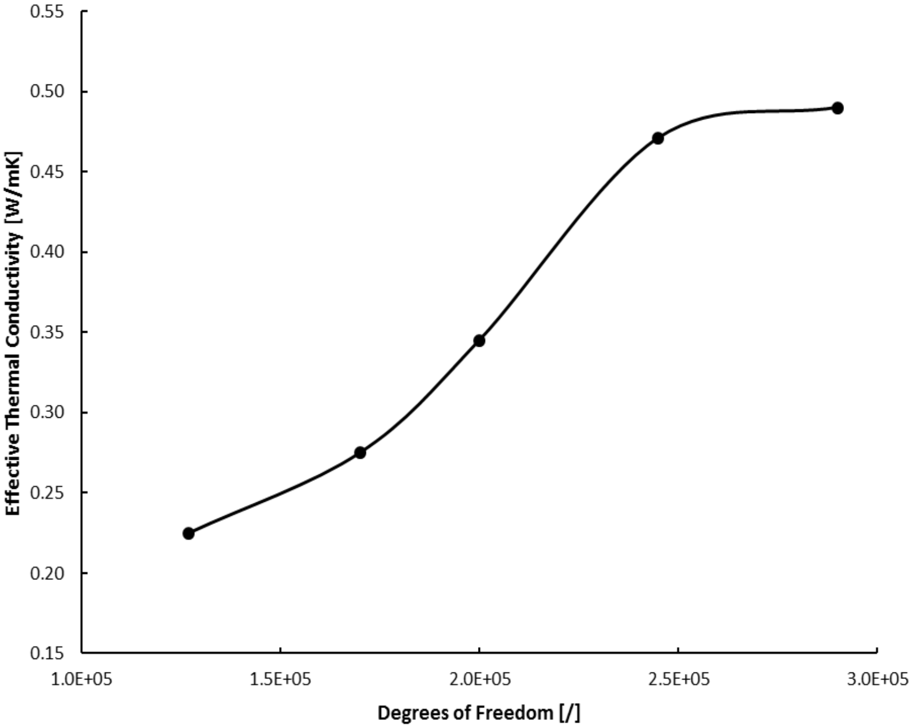


Figure 3-8 The effective thermal conductivity relates to RVE front surface for different level of mesh refinement.

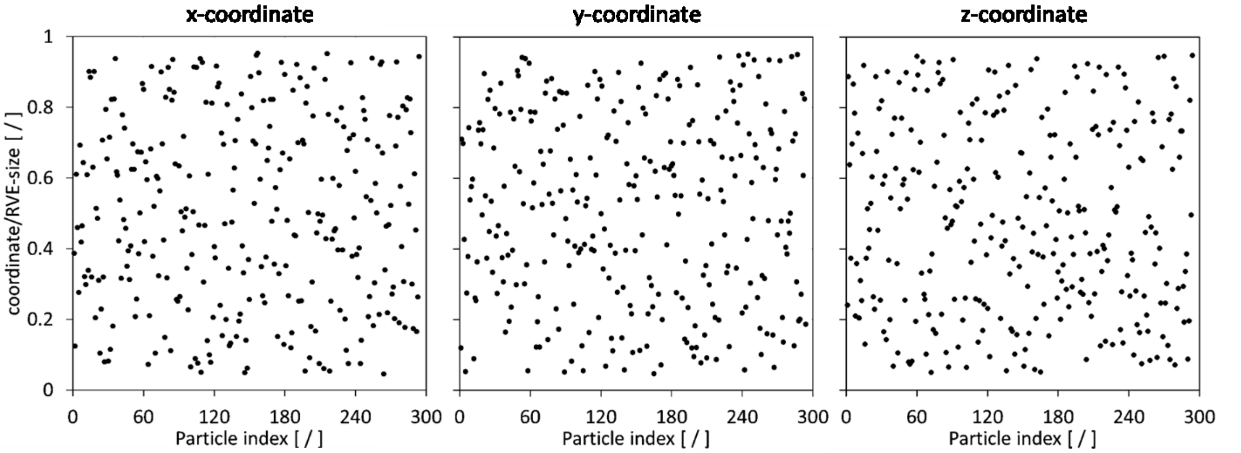


Figure 3-9 Sample spacial distribution of particle centers in RVE for 13.2% filler volume fraction.

While the ITR for the studied material system could be evaluated using experimental means, it was herein decided to determine this parameter iteratively using the developed modeling framework and results from the technical literature. Various modeling iterations were completed for

8.00% filler volume fraction. Setting different values for the TCC enabled tuning this parameter to match the effective thermal conductivity given in [54]. A TCC determined as 5,150 W·m^{-2}K^{-1} was subsequently used to compute the effective thermal conductivities for the remaining filler volume fractions in this study.

In order to assess RVE isotropy, effective thermal conductivities were computed along the different coordinate directions. While the mean values for multiple model iterations indicate isotropic material behavior, it is interesting to note that for a single model, data can be slightly anisotropic as shown in Table 3-2 for a model with 8.00% filler volume fraction. Overall, the investigation of particle spacial distribution and RVE isotropy indicated that the RNG module provided appropriate statistical randomness to the modeling framework.

Table 3-2 Thermal conductivity in mutually perpendicular directions in RVE with 8.00% filler volume fraction.

Property	x-direction	y-direction	z-direction
Thermal conductivity (W·m^{-1}·K^{-1})	0.815	0.800	0.782

As mentioned previously, each FEA solution, i.e. an effective thermal conductivity value is transferred to the MCS module for data storage and statistical analysis, that is, computing the unbiased standard deviation and variance of the effective thermal conductivity results. The MCS module repeats this process until the standard deviation is less than a specified valued or until a set number of iterations is reached. In line with Monte Carlo simulation, the mean of the effective thermal conductivity results was taken as final effective thermal conductivity for a given filler volume fraction. Table 3-3 shows the results and various statistical analyses computed for the present material systems. Notably, standard deviations increase with increasing filler volume fractions from 1.3% to 13.2%, followed by a reduction at 15.6% filler content. This behavior can be explained by the number of model iterations were completed for each case, i.e., over 500 iterations were performed for filler volume fractions of 1.3% to 13.2%, and only 100 iterations were performed for 15.6% filler content due to limitation in computing resources. Clearly, these results indicate the importance of running simulations with a sufficient number of iterations for creating acceptable accuracy.

The simulation results were studied to assess the characteristics of the data that was obtained through the PRNG and overall modeling framework. It was hypothesized that statistical randomness is obtained with numerical results being close to a mean value and following a normal distribution. Fundamentally, an infinite number of iterations could be performed to predict the effective thermal conductivity for each volume fraction, and hence, predictions can be considered continuous random variables for statistical analysis purposes. A Probability Density Function (PDF) was therefore computed for each dataset associated with one of the studied filler volume fractions. Fig. 3-10 depicts the PDF of the simulation data for a filler volume fraction of 13.2%. Note that for better data visualisation the PDF data was normalized with respect to the maximum value of the dataset.

Table 3-3 Results for thermal conductivity simulations and statistical analyses for PEEK-matrix/Ag-particle composites.

Filler volume fraction [%]	1.30	4.78	8.00	13.2	15.6
Number of iterations (/)	512	516	500	504	100
Mean value ($W{\cdot}m^{-1}{\cdot}K^{-1}$)	0.240	0.307	0.490	0.848	1.746
Median value ($W{\cdot}m^{-1}{\cdot}K^{-1}$)	0.240	0.304	0.473	0.847	1.739
Standard deviation ($W{\cdot}m^{-1}{\cdot}K^{-1}$)	0.001	0.022	0.088	0.108	0.071
Variance ($W{\cdot}m^{-1}{\cdot}K^{-1}$)	$3.3{\cdot}10^{-6}$	$5.1{\cdot}10^{-4}$	0.007	0.011	0.005
Skewness (/)	1.242	2.154	0.930	0.053	0.083
Kurtosis (/)	2.220	9.476	0.671	-0.499	1.052
95% confidence value ($W{\cdot}m^{-1}{\cdot}K^{-1}$)	0.0001	0.0015	0.0077	0.0094	0.0142

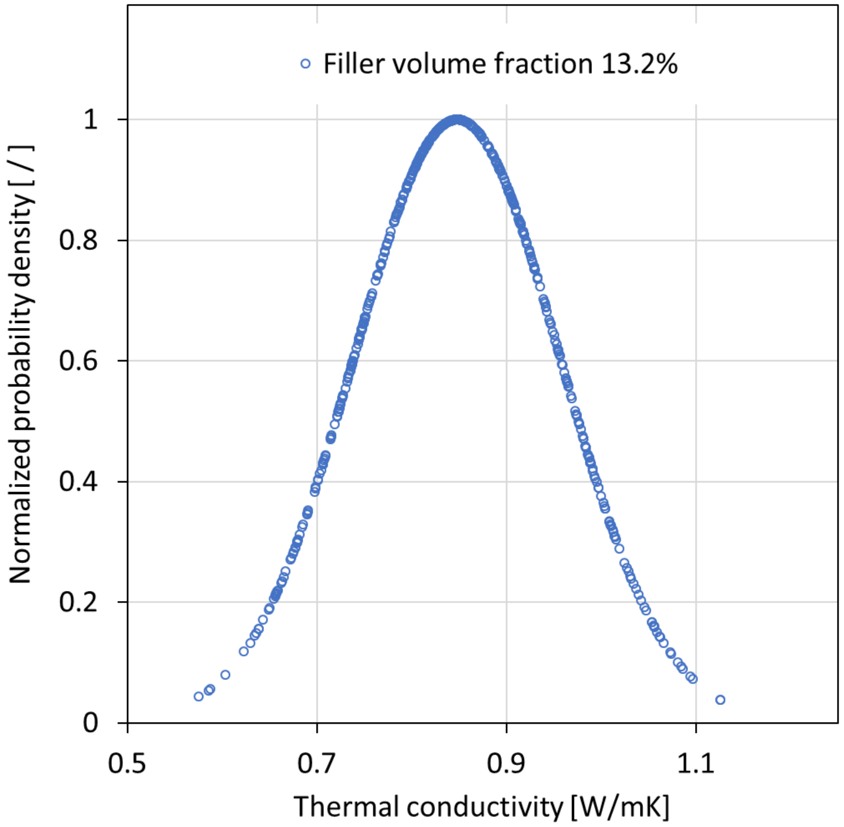


Figure 3-10 Normalized probability density of effective thermal conductivity data for 13.2% filler volume fraction.

The graph in Fig. 3-10 suggests that data is normally distributed. To test this presumption, the mean, median, skewness and kurtosis values were determined for the simulation data. (Skewness and kurtosis are also referred to as asymmetry and 'peakedness' of a distribution, respectively.) These parameters can be used as criteria for accepting or rejecting a dataset to be considered as normally distributed. Notably, mean and median values agree within 4% for all filler volume fraction cases. A skewness value near zero and an absolute kurtosis value of less than 2 are generally considered quality indicators for normally distributed data, and West et al. [56] proposed that an absolute skewness greater than 2 and kurtosis greater than 7 can be considered an indication for significant departure from normality. Referring the skewness and kurtosis values in Table 3-3, it can be concluded that data for 13.2% and 15.6% filler volume fraction closely meet the normality criteria. Data for the lower filler volume fractions of 1.30%, 4.78%, 8.00% do not meet normally conditions. However, this result is to be expected since data for diminishing filler volume fractions is bounded by the thermal conductivity of the polymer matrix (i.e. 0.23 $\mathrm{W{\cdot}m^{-1}K^{-1}}$). An asymmetric PDF for the three lower filler volume fractions is therefore sensible. Fig. 3-11, which depicts normalized probability density data for all filler volume fractions, further illustrates the effect of the lower bound imposed by the thermal conductivity of the polymer matrix. Referring to the method described in [56] a normality test was thus performed by filtering data that are greater than a 'right bound' calculated according to Eq. (3-16).

$$K_{\mathrm{R}} = \tilde{x} + (\tilde{x} - K_{\mathrm{L}}) \tag{3-16}$$

where K_{R} is the 'right bound' of the effective thermal conductivity data, and $\tilde{x}$ and K_{L} are correspondingly the median and minimum effective thermal conductivity. Statistical analyses results from filtered data for the lower filler volume fractions of 1.30%, 4.78%, 8.00% are shown in Table 3-4. These results indicate that filtered simulation data for the lower filler volume fractions pass the normality test.

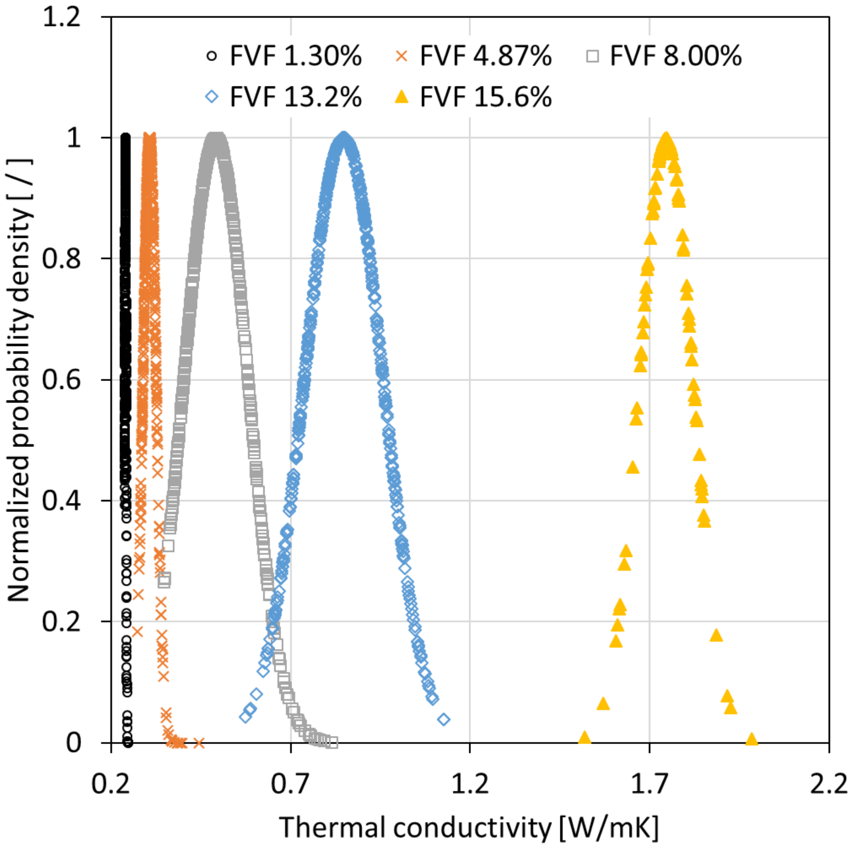


Figure 3-11 Normalized probability density of effective thermal conductivity data for different filler volume fractions (FVF).

Table 3-4 Filtered thermal conductivity simulation data and statistical analyses for PEEK-matrix/Ag-particle composites.

Filler volume fraction [%]	1.30	4.78	8.00
Mean value ($W{\cdot}m^{-1}{\cdot}K^{-1}$)	0.240	0.305	0.476
Median value ($W{\cdot}m^{-1}{\cdot}K^{-1}$)	0.240	0.303	0.465
Skewness (/)	0.261	0.361	0.430
Kurtosis (/)	-1.042	-0.477	-0.712

Since present simulations can be considered continuous random variables it is more appropriate to calculate the probability of an explicit effective thermal conductivity occurring within a specific interval using Eq. (3-17).

$$P(a \leq X \ \leq b) = \int_a^b f(\chi)d\chi \tag{3-17}$$

where P is the probability of an effective thermal conductivity occurring within an interval a and b; X and $f(\chi)$ are a continuous random variable and the PDF, respectively. A Cumulative Distribution Function (CDF) can thus be computed for each of the various filler volume fractions. Corresponding graphs are depicted in Fig. 3-12.

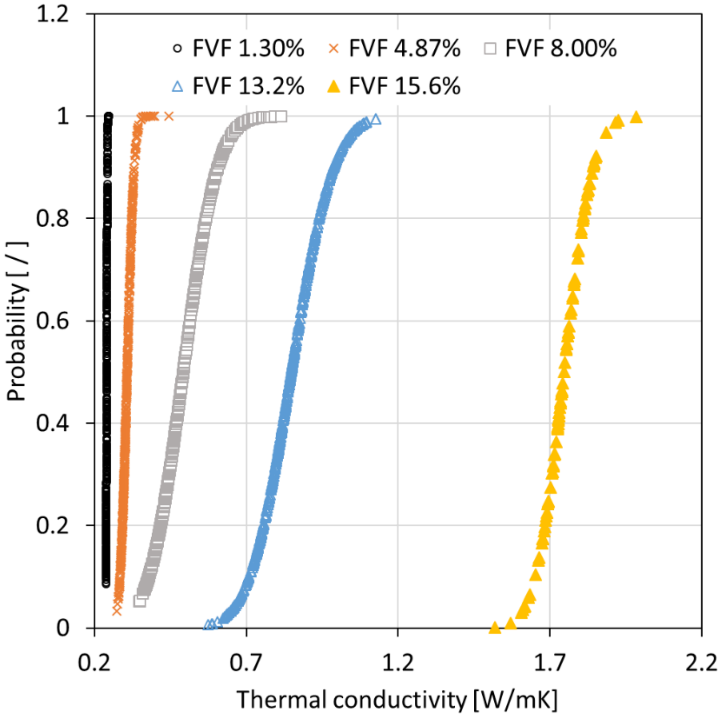


Figure 3-12 Cumulative density function of effective thermal conductivity data for different filler volume fractions (FVF).

Finally, the results from the SFEA framework were compared to the experimental results from [54] and analytical predictions. For the latter the models by Maxwell and Rayleigh were considered as described in the Introduction section of this article. Fig. 3-13 depicts the mean values of the SFEA predictions, the analytical solutions and experimental data for the filler volume fractions of 1.30%, 4.87%, 8.00%, 13.2% and 15.6%. Notably, SFEA results are in excellent agreement with the experimental data. While the analytical models substantially underpredict the experimental values, the results from the SFEA simulations match rather closely the strongly nonlinear trend of the experimental data. The difference between the SFEA results and the experimental values were calculated to be less than 7.5%. It is presumed that the close match is a consequence of the SFEA simulation's ability to include interfacial interactions between particles and particles and matrix. In particular, particle-to-particle interactions are likely becoming increasingly important for increasing filler volume fractions, as indicated by the strong nonlinear increase of experimental data. Based on these observations, the developed SFEA modeling framework is seen as an effective tool to predict the thermal conductivity of particulate filler modified composite materials.

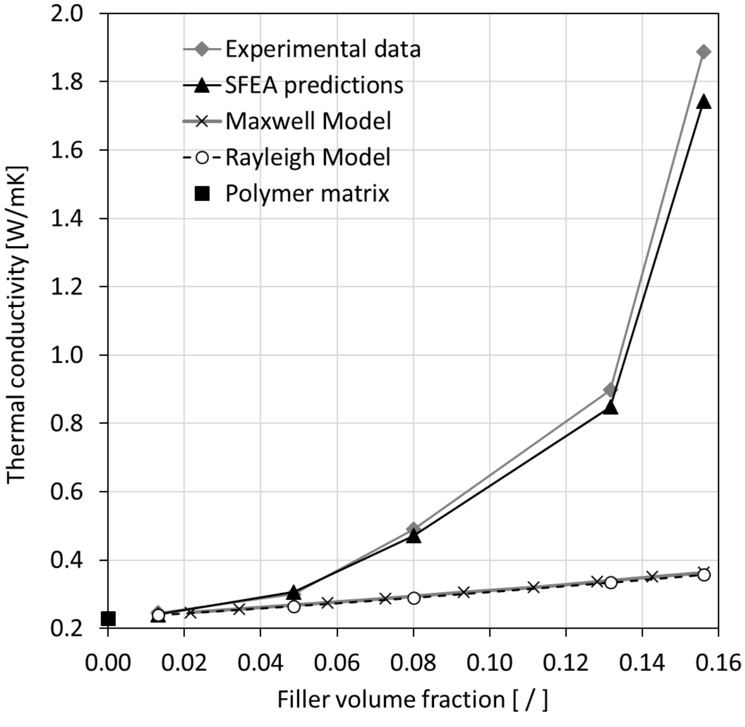


Figure 3-13 Thermal conductivity comparison for Silver Nano particles embedded in PEEK [54].

3.5 Conclusions

A stochastic finite element analysis framework was developed that enables predicting the properties of particulate filler modified polymer composites. The framework comprises a Monte Carlo modeling approach. The modeling framework is versatile in the sense that it can capture filler size distributions and different filler shapes and orientations. In the present study, the effects on thermal conductivity caused by randomly distributing spherical particles in a matrix was investigated. Notably, the developed modeling approach considers the influence of thermal resistance at the filler-matrix interface as well as direct thermal contact between particles. Interfacial thermal properties were considered constant and were determined by tuning the corresponding parameter to an experimental value for a specific filler volume fraction.

The study of numerical model predictions indicated that statistical randomness was obtained. Numerical results were distributed closely to a mean value following a normal distribution. Upon computing effective thermal conductivity values for a broad range of filler volume fractions an excellent agreement between model predictions and experimental data was ascertained. Based on the observations and analyses made in this study, the effectiveness of the developed stochastic finite element analysis framework could be confirmed.

3.6 References

[1] Beret, S.; Prausnitz, J. Densities of liquid polymers at high pressure. Pressure-volume-temperature measurements for polythylene, polyisobutylene, poly(vinyl acetate), and poly(dimethylsiloxane) to 1 kbar. Macromolecules. 1975, 8, 536–538, DOI: 10.1021/ma60046a028.

[2] Manfredi, L.; Rodriguez, E.; Wladyka-Przybylak, M.; Vazquez, A. Thermal degradation and fire resistance of unsaturated polyester modified acrylic resins and their composites with natural fibres. Polym. Degrad. Stab. 2006, 91, 255-261, DOI: 10.1016/j.polymdegradstab.2005.05.003.

[3] Drury, J.; Mutsaers, C.; Hart, C.; Matters, M.; de Leeuw D. Low-cost all-polymer integrated circuits. Appl. Phys. Lett. 1998, 73, 108-110, DOI: 10.1063/1.121783.

[4] Sheng, N.; Boyce, M.; Parks, D.; Rutledge, G.; Abes, J. Multiscale micromechanical modeling of polymer/clay nanocomposites and the effective clay particle. Polymer. 2004, 45, 487-506, DOI: 10.1016/j.polymer.2003.10.100.

[5] Ma, P.; Siddiqui, N.; Marom, G.; Kim, J. Dispersion and functionalization of carbon nanotubes for polymer-based nanocomposites: A review. Composites, Part A Composites. 2010, 41, 1345-1367, DOI: 10.1016/j.compositesa.2010.07.003.

[6] Akcora, P.; Liu, H.; Kumar, S.; Moll, J.; Brian Y.; Benicewicz, C.; Schadler, L.; Acehan, D.; Panagiotopoulos, A.; Pryamitsyn, V.; Ganesan, V.; Ilavsky, J.; Thiyagarajan, P.; Colby, R.; Douglas, J. Anisotropic self-assembly of spherical polymer-grafted nanoparticles, Nat. Mater. 2009, 8, 354–359, DOI: 10.1038/nmat2404.

[7] Li, Z.; Okamoto, K.; Ohki, Y.; Tanaka, T. Effects of nano-filler addition on partial discharge resistance and dielectric breakdown strength of Micro-Al2O3Epoxy composite. IEEE Trans. Dielectr. Electr. Insul. 2010, 17, 653-661, DOI: 10.1109/TDEI.2010.5492235.

[8] Lin, F.; Bhatia, G.; Ford, J. Thermal conductivities of powder-filled epoxy resins. J. Appl. Polym. Sci. 1993, 49, 1901–1908, DOI: 10.1002/app.1993.070491105.

[9] Agari, Y.; Uno, T. Estimation on thermal conductivities of filled polymers. J. Appl. Polym. Sci. 1986, 32, 5705–5712, DOI: 10.1002/app.1986.070320702.

[10] Cheng, S.; Vachon, R. The prediction of the thermal conductivity of two and three phase solid heterogeneous mixtures. Int. J. Heat Mass Transfer. 1969, 12, 249-264, DOI: 10.1016/0017-9310(69)90009-X.

[11] Hasselman, D.; Johnson, L. Effective Thermal Conductivity of Composites with Interfacial Thermal Barrier Resistance. J. Compos. Mater. 1987, 21, 508-515, DOI: 10.1177/002199838702100602.

[12] Song, Y.; Youn, J. Influence of dispersion states of carbon nanotubes on physical properties of epoxy nanocomposites. Carbon. 2005, 43, 1378-1385, DOI: 10.1016/j.carbon.2005.01.007.

[13] Karasek, L.; Sumita, M. Characterization of dispersion state of filler and polymer-filler interactions in rubber-carbon black composites. J. Mater. Sci. 1996, 31, 281–289, DOI: 10.1007/BF01139141.

[14] Yung, K.; Liem, H. Enhanced thermal conductivity of boron nitride epoxy-matrix composite through multi-modal particle size mixing. J. Appl. Polym. Sci. 2007, 106, 3587–3591, DOI: 10.1002/app.27027.

[15] Paul, M.; Alexandre, M.; Degee, P.; Henrist, C.; Rulmont, A.; Dubois, P. New nanocomposite materials based on plasticized poly(l-lactide) and organo-modified montmorillonites: thermal and morphological study. Polymer. 2003, 44, 443-450, DOI: 10.1016/S0032-3861(02)00778-4.

[16] Fan, Z.; Wei, T.; Luo, G.; Wei, F. Fabrication and characterization of multi-walled carbon nanotubes-based ink. J. Mater. Sci. 2005, 40, 5075–5077, DOI: 10.1007/s10853-005-1622-7.

[17] Kordas, K.; Mustonen, T.; Toth, G.; Jantunen, H.; Lajunen, M.; Soldano, C.; Talapatra, S.; Kar, S.; Vajtai, R.; Ajayan, P. Inkjet printing of electrically conductive patterns of carbon nanotubes. Small. 2006, 2, 1021-1025, DOI: 10.1002/smll.200600061.

[18] Small, W.; Panhuis, M. Inkjet printing of transparent electrically conducting single-walled carbon-nanotube composites. Small. 2007, 3, 1500-1503, DOI: 10.1002/smll.200700110.

[19] Wei, T.; Ruan, J.; Fan, Z.; Luo, G.; Wei, F. Preparation of graphene Nano sheet/polymer composites using in situ reduction–extractive dispersion. Carbon. 2007, 45, 2296-2299, DOI: 10.1016/j.carbon.2009.04.030.

[20] Dragoman, M.; Flahaut, E.; Dragoman, D.; Al Ahmad, M.; Plana, R. Writing simple RF electronic devices on paper with carbon nanotube ink. Nanotechnology. 2009, 20, 5203, DOI: 10.1088/0957-4484/20/37/375203/meta.

[21] Khabibullin, A.; Zharov, I. Fuel cell membranes based on polymer-modified silica colloidal crystals and glasses: Proton conductivity and fuel cell performance. Mater. Res. Soc. Symp. Proc. 2013, 1502, DOI: 10.1557/opl.2013.568.

[22] Cao, Q.; Rogers, J. Ultrathin films of single-walled carbon nanotubes for electronics and sensors: a review of fundamental and applied aspects. Adv. Mater. 2009, 21, 29-53, DOI: 10.1002/adma.200801995.

[23] Avouris, P.; Chen, Z.H.; Perebeinos, V. Carbon-based electronics. Nanoscience and Technology. 2009, 2, 174-184, DOI: 10.1142/9789814287005_0018.

[24] Kim, J.; Kang, P.; Chang Nho Y. Positive temperature coefficient behavior of polymer composites having a high melting temperature. J. Appl. Polym. Sci. 2004, 92, 394–401, DOI: 10.1002/app.20064.

[25] Nikkeshi, S.; Kudo, M.; Masuko, T. Dynamic viscoelastic properties and thermal properties of powder-epoxy resin composite. J. Appl. Polym. Sci. 1998, 69, 2593–2598, DOI: 10.1002/(SICI)1097-4628(19980926)69:13<2593::AID-APP9>3.0.CO;2-5.

[26] Maxwell, J. A treatise on electricity and magnetism, volume I, 3rd Edition, Oxford University Press, 1904.

[27] Bird R.; Stewart W.; Lightfoot, E. Transport phenomena,John Wiley & Sons, 2007, ISBN: 0-470-11539-4.

[28] Eucken A. Die Wärmeleitfähigkeit Keramischer, Fester Stoffe– Ihre Berechnung aus der Wärmeleitfähigkeit der Bestandteile, VDI Forschungsheft 353, Beilage zu, Forschung auf dem Ggebiete des Ingenieurwesens, Ausgabe B, Band 3, 1932.

[29] Burger, H. Das lertvermogen verdumter mischkristallfreierlonsungen, Phys. Zs. 20, 1915.

[30] Hamilton, R.; Crosser, O. Thermal conductivity of heterogeneous two component systems, Ind. Eng. Chem. Fundam. 1962, 1, 187-191, DOI: 10.1021/i160003a005.

[31] Lord Rayleigh Sec. R.S. LVI. On the influence of obstacles arrangedin rectangular order upon the properties of a medium. The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science, 1892, 34, 481-502, DOI: 10.1080/14786449208620364.

[32] Bohm, H.; Nogales, S. Mori-Tanaka Model for the thermal conductivity of composites with interfacial resistance and particle size distributions. Compos. Sci. Technol. 2008, 68, 1181-1187, DOI: 10.1016/j.compscitech.2007.06.009.

[33] Hasselman, D.; Johnson, L. Effective Thermal Conductivity of Composites with Interfacial Thermal Barrier Resistance. J. Compos. Mater. 1987, 21, 508-515, DOI: 10.1177/002199838702100602.

[34] Meincke, O.; Kaempfer, D.; Weickmann, H.; Friedrich, C.; Vathauer, M.; Warth, H. Mechanical properties and electrical conductivity of carbon-nanotube filled polyamide-6 and its blends with acrylonitrile/butadiene/styrene. Polymer, 2004, 45, 739-748, DOI: 10.1016/j.polymer.2003.12.013.

[35] Valavala, P.; Odegard, G.; Modeling techniques for determination of mechanical properties of polymer Nanocomposites. Department of Mechanical Engineering, Michigan Technological University, 1400 Townsend Drive, Houghton, MI 49931, USA, 2005

[36] Odegard, G.; Clancy, T.; Gates, T. Modeling of the mechanical properties of nanoparticle/polymer composites. Polymer, 2005, 46, 553-562, DOI: 10.1016/j.polymer.2004.11.022.

[37] Liu, Y.; Chen, X. Evaluations of the effective material properties of carbon nanotube-based composites using a nanoscale representative volume element. Mech. Mater. 2003, 35, 69–81, DOI: 10.1016/S0167-6636(02)00200-4.

[38] Majumdar, C.; Gerner, A. Microscale energy transport. Washington, D.C.: Taylor & Francis., 1998, ISBN: 978-1560324591.

[39] Chen, G. Nanoscale energy transport and conversion: a parallel treatment of electrons, molecules, phonons, and photons, New York: Oxford. ISBN: 978-0195159424, 2004.

[40] Pollack, G. Kapitza Resistance. Rev. Mod. Phys. 1996, 41, 48, DOI: 10.1103/RevModPhys.41.48.

[41] Maybeck, P. Stochastic models, Estimation and control, Mathematic in Science and Engineering, 141-3, ISBN: 0-12-480701-1.

[42] Haldar, A.; Mahadevan, S. Reliability assessment using stochastic finite element analysis. New York, NY, Wiley, 2000, ISBN: 978-0-471-36961-5.

[43] Koduru, S.; Haukaas, T. Feasibility of FORM in finite element reliability analysis. Struct. Saf. 2010, 32, 145–153, DOI: 10.1016/j. strusafe.2009.10.001.

[44] Thacker, B.; Riha, D.; Fitch, S.; Huyse, L.; Pleming, J. Probabilistic engineering analysis using the NESSUS software. Struct. Saf. 2006, 28, 83–107, DOI: 10.1016/j.strusafe.2004.11.003.

[45] Patelli, E.; Panayirci, M.; Broggi, M.; Goller, B.; Beaurepaire, P.; Pradlwarter, H.; Schuëller, G. General purpose software for efficient uncertainty management of large finite element models. Finite Elem. Anal. Des. 2012, 51, 31–48, DOI: 10.1016/j. finel.2011.11.003.

[46] Akula, V. Multiscale reliability analysis of a composite stiffened panel. Compos. Struct. 2014, 116, 432–440, DOI: 10.1016/j. compstruct.2014.06.001.

[47] McKenna, F.; Fenves, G.; Scott, M. Open system for earthquake engineering simulation. Berkeley, CA: University of California. Retrieved from http://opensees.berkeley.edu, 2000.

[48] Wang, J. Industrial design engineering: Inventive Problem Solving. Taylor and Francis group. 2017, ISBN: 9781498709606.

[49] Grimmett, G.; Stirzaker R. Probability and random processes. 2nd Edition. Clarendon Press, Oxford. ISBN: 0-19-853665-8.

[50] Sen, P.; Singer, J. Large sample methods in statistics. Singer. Chapman & Hall, Inc. 1994, ISBN: 0412042215.

[51] Barker, E.; Recommendation for Key Management, NIST Special Publication 800-57. NIST, 2013, DOI: 10.6028/NIST.SP.800-57pt1r4.

[52] Ecuyer, L. Uniform random number generators, International Encyclopedia of Statistical Science, Springer, 2010.

[53] Skinner, M.; Mertiny, P. Effect of Random and Size-Ordered Addition of Spherical Nanoparticles Inside a Representative Volume Element, CAMX conference, 2016.

[54] Rivière, L.; Lonjon, A.; Dantras, E.; Lacabanne, C.; Olivier, P.; Gleizes, N. Silver fillers aspect ratio influence on electrical and thermal conductivity in PEEK/Ag nanocomposites. Eur. Polym. J. 2016, 85, 115-125, DOI: 10.1016/j.eurpolymj.2016.08.003.

[55] Mishra, D.; Satapathy, A.; Patnaik, A. A simulation study on thermal conductivity of glass bead embedded polymeric system, Int. J. Plast. Technol. 2012, 16, 24–38, DOI: 10.1007/s12588-012-9028-2.

[56] West S.; Finch J.; Curran P. Structural equation models with nonnormal variables: problems and remedies, In: Hoyle RH, editor. Structural equation modeling: Concepts, issues and applications, 1995, 56-57.

Chapter 4

Thermal Characterization of Aligned Particulate Polymer Composites

In a third part of this research study, a stochastic FEA framework was developed that enabled calculating effective thermal conductivity of randomly aligned cylindrical particulate polymer composites. This work was published as publication no. 4 listed in the Preface of this thesis document.

Nomenclature

K_i	node thermal conductivity
Q_i	node heat flux
K_i	node thermal conductivity
l	RVE length
T_2	RVE front surface temperature
T_1	RVE front surface temperature
K_{eff}	Effective thermal conductivity
i	node number
n	Total number of nodes

4.1 Introduction

The advent of engineered nanoparticles with high aspect ratio, such as graphene and carbon nanotubes (CNT), and their availability in quantities relevant for industrial production, has greatly expanded opportunities to modify polymers to meet demanding requirements in a broad range of applications. Such nano-additives have been shown to improve polymer mechanical (e.g., stiffness, strength and fracture properties) as well as physical characteristics (e.g., electrical and thermal conductivity), see e.g., [1–6]. The same holds true in the context of flame retardancy. Among the three commonly considered flame retardant approaches (i.e., gas phase flame retardants, endothermic flame retardants, and char-forming flame retardants) nanofillers are typically active via the latter mechanism. Nanofillers operate in the polymer condensed phase where they may provide thermal insulation and a mass transport barrier that mitigates the release of fuel into the gas phase. Nanocomposites with suitable filler morphology and loading were observed to form a coherent filler network layer covering sample surfaces, which significantly reduced peak heat release and radiant heat flux [7]. In addition to the char-forming mechanism, nanofillers were found to reduce the melt flow of polymers. High aspect ratio nano-additives were reported to form jammed network structures causing melt to behave rheologically like a gel, thus inhibiting dripping of flammable material [8].

While the potential of nano-fillers to enhance flame retardancy through increased barrier properties impeding heat flux and fuel release, and altered rheological properties inhibiting flammable drips, has widely been acknowledged in the technical literature, the influence of filler addition on increased thermal conductivity and thus heat transfer into the polymer still requires further study [1]. Carbon-based fillers possess thermal conductivities that exceed those of polymers by several orders of magnitude. For example, thermal conductivity ranging from 2000 to 5000 $Wm^{-1}K^{-1}$ has been reported for CNT and graphene while values for typical polymers are between 0.1 and 0.3 $Wm^{-1}K^{-1}$ [9-10]. Assessing and understanding thermal conduction in nanocomposites with high aspect ratio fillers is particular complex due to the inherent propensity of filler contact, alignment and agglomeration.

Besides randomly oriented and dispersed particles, polymer nanocomposites with purposely aligned particulate fillers have been created, which resulted in improved performance in a variety of applications. Nanocomposites with aligned particles have been used for the design of sensors [11–13] and high-strength modified polymers that require particle alignment in order to achieve specific anisotropic properties [14–16]. Carbon nanotubes, as a 'one- dimensional' high aspect ratio carbon allotrope, are especially suited to create nanocomposites with anisotropic properties, e.g., in terms of heat transfer properties [17–22].

Determining the mechanical and physical properties using experimental methods is typically a time-consuming and costly approach. Analytical methods, on the other hand, are highly efficient for predicting effective material properties of particulate composites [23]. However, analytical methods lack accuracy when predicting properties, especially for higher filler volume fraction modified polymers. Considering these limitations, and in light of a rapidly growing number of applications involving particulate composites, experimental and analytical approaches are not sufficient to address the demands imposed by a vast field of available filler materials and fabrication parameters. Hence, alternative methods for assessing and designing filler modified composites need to be developed [24–26].

Stochastic analysis is one of the most reliable and recognized methods for analyzing complex problems involving many input and output parameters in the field of reliability analysis.

This method can predict accurate outcomes using statistical principles. Stochastic analysis can be used in a variety of applications, e.g., financial forecasting and modeling, where numerous input and output variables need to be considered. Recently, a numerical modeling framework was developed based stochastic analysis to simulate the effective material properties of filler modified composites [27]. Specifically, a stochastic finite element analysis (SFEA) approach was employed that enabled the prediction of the effective thermal conductivity of randomly distributed and disperses spherical particles embedded in a polymer matrix. In the present contribution, aforementioned SFEA approach was adopted to predict the effective thermal conductivity of a polymer matrix containing randomly oriented or aligned rod-shaped filler particles. The particle geometry was adapted to mimic CNT. The study described herein investigates the effect of filler addition and alignment on heat transfer into polymer composites in the context of fire-retardant materials.

4.2 Stochastic Finite Element Analysis Framework

The nature of stochastic analysis, and thus the presented modeling approach, requires performing numerous iterations in order to calculate the effective thermal conductivity of a polymer matrix containing a rod-shaped filler. The SFEA algorithm described in [27] was adopted and employed for the present study. This algorithm provides a framework for connecting a customized stochastic analysis with a parametric finite element analysis (FEA). In this manner, the process of applying uncertainty to input variables, and creating and solving FEA models is automated. The modeling framework, which uses several scripting languages, is briefly summarized in the present section. The interested reader is referred to [27] for additional details on the modeling approach. Fig. 4-1 depicts a flowchart outlining the main domain, i.e., the elemental structure and connections, of the framework's various modules.

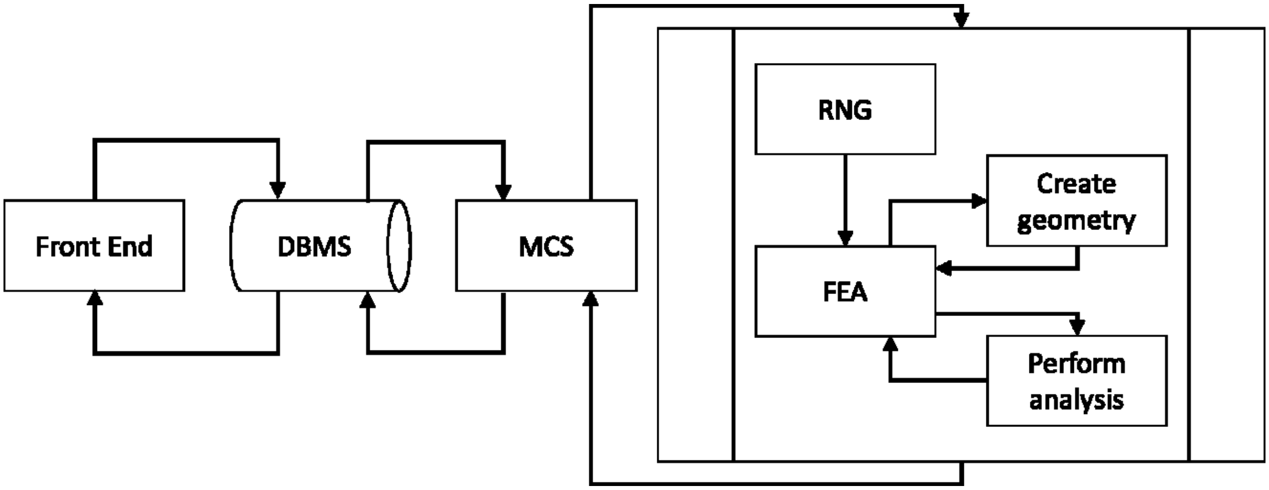


Figure 4-1 Schematic of the SFEA framework showing connections between individual modules [27].

The main domain was developed in Visual Basic for Applications scripting language (VBA; Microsoft, Redmond, Washington, USA). Input parameters are provided via the 'Front End' module. The set of required input parameters comprises (i) the modeling domain that is defined by the size of the considered cubic representative volume element (RVE); (ii) a set of filler volume fractions that are to be analyzed; (iii) the material properties for the filler and matrix; (iv) the filler particle size distribution; (v) information on boundary effects, i.e., particle-to-particle and particle-to-matrix interfacial thermal resistance (ITR) as well as a threshold gap size that defines direct contact between particles and particles to the RVE boundary; (vi) details for the FEA mesh generation; and (vii) details regarding the model output acceptance criteria required for statistical analyses, i.e., standard deviation and variance. The input parameters are transferred to a database with an appropriate management system ('DBMS'), which holds and communicates input and computed data between the various modules.

The Monte Carlo simulation (MCS) module, also developed using VBA scripting language, performs two subprocesses, i.e., the random number generator (RNG) and the FEA modeling. Using the algorithm depicted in Fig. 4-2, the MCS module retrieves needed input parameters from the database, performs iteratively the SFEA, computes statistical data (standard deviation and variance) after each iteration, and finally stores results back into the database. The MCS module repeats the modeling subprocess until the acceptance criteria defined in the database are satisfied. Once results converge according to the criteria specified, the MCS module determines the effective thermal conductivity (by calculating an average value). The MCS module repeats the above processes until all specified filler volume fractions are analyzed.

The RNG module was developed in the numerical computing environment MATLAB (MathWorks, Natick, Massachusetts, USA), which has pseudorandom number generating capabilities. This module retrieves the input data defining the RVE size and the particle size distribution from the database. The RNG module sequentially creates sets of random numbers for anchor points in Cartesian coordinates as well as vectors required for generating the position and orientation of rod-shaped filler particle geometries, respectively. The RNG module also performs a collision detection using a geometrical model to avoid particles interfering with each other as well as with the RVE surfaces. When detecting interference, the RNG rejects the most recently generated particle. In the geometrical model, the rod-shaped particle geometry is represented by a series of spheres (see Fig. 4-3). The distance between each sphere associated with the most recent rod-shaped particle and all preexisting sphere geometries in the RVE, and the RVE boundaries, is evaluated to discern a particle collision. While representing rod-shaped particles using a series of spheres is only an approximation, it pro- vides an expedient means for performing the collision detection algorithm. For the analyses presented herein, as series of 200 spheres was used to represent rod-shaped particles for interference detection.

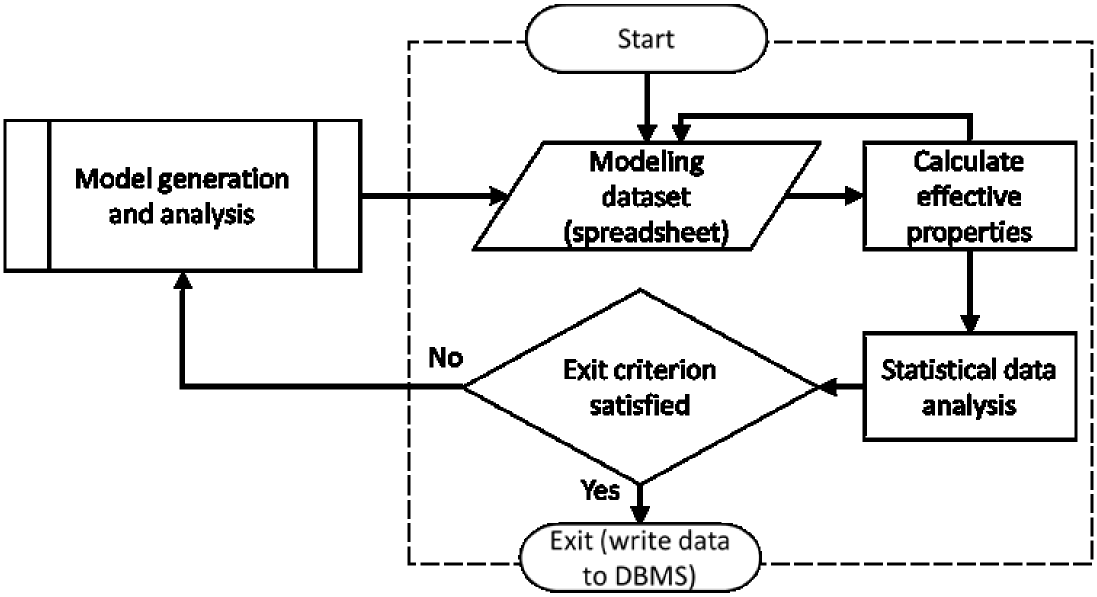

Start
Model generation and analysis
Modeling dataset (spreadsheet)
Calculate effective properties
No
Exit criterion satisfied
Statistical data analysis
Yes
Exit (write data to DBMS)

Figure 4-2 Monte Carlo simulation (MCS) module algorithm [27].

The process performed by the RNG module can be controlled to yield both randomly oriented and aligned rod-shaped particles within the RVE. In the case of aligned particles, constraints are imposed on the vector indicating particle orientation. As shown in Fig. 4-4, after generating the set of random numbers for each particle, these data are stored in the database in a tabulated format for later use by the FEA module. The RNG module iteration is terminated when the required filler volume fraction is reached.

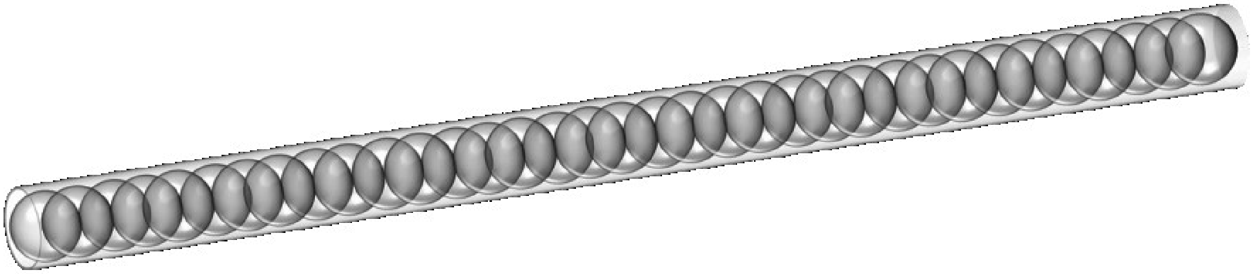


Figure 4-3 Schematic of series of spheres representing rod-shaped particles.

It should be mentioned that the RNG module also has the ability of creating particles that conform to a given size distribution (whilst, this feature was not utilized in the present study). The interested reader is referred to [27] for details on the algorithm that produces particles obeying a certain size distribution, and the effect that different size-ordered particle addition has on computational performance.

The FEA module was developed as a fully customizable parametric FEA platform in ANSYS Workbench (Version 19, ANSYS Inc., Canonsburg, PA, USA) using IronPython scripting language, which enabled applying uncertainties to input parameters required for performing the FEA simulation. This platform consists of a model generation environment, i.e., ANSYS DesignModeler, and a model solution environment, i.e., ANSYS Mechanical, which enable creating the parametric geometry and the finite element model, respectively. JAVA scripting language was used to automate the process of reading input data from the database (i.e., RVE dimensions and particle anchor points and orientation vectors) and creating particle geometries in the model generation environment. The three-dimensional geometry thus created is transferred to the model solution environment for further analysis. Similar to the model generation environment, the model solution environment also uses JAVA scripting language to automate the FEA process. The model solution environment retrieves further inputs parameters from the database, including material properties, information on boundary effects and conditions, and mesh generation parameters, and then constructs the finite element model for each model iteration. After performing the analysis, the FEA results are saved to the database in tabulated format for further statistical analysis. Fig. 4-5 illustrated the algorithm for the FEA module.

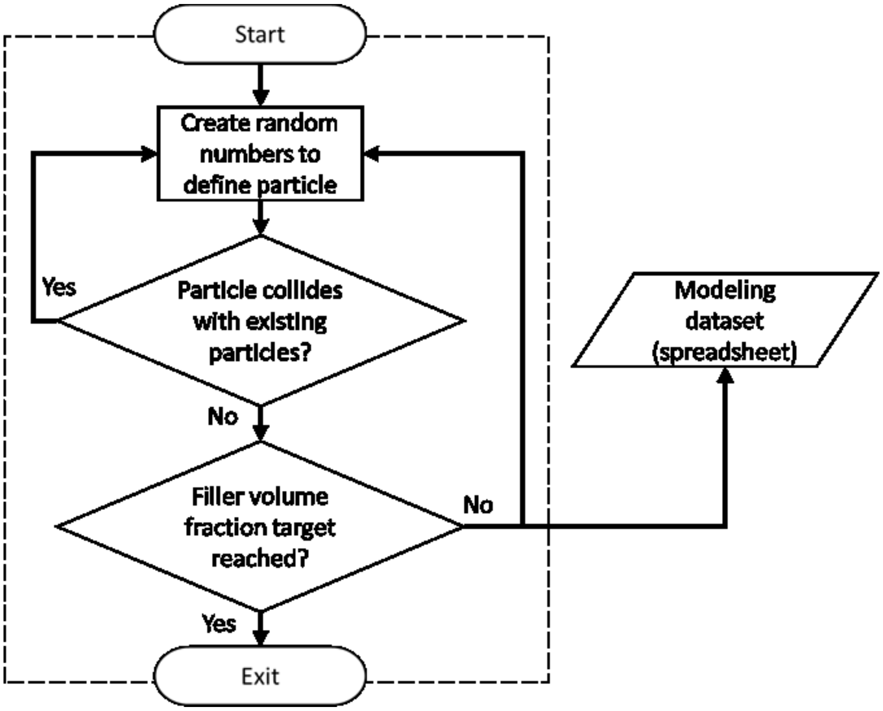

Start
Create random numbers to define particle
Particle collides with existing particles?
Yes
No
Filler volume fraction target reached?
No
Yes
Exit
Modeling dataset (spreadsheet)

Figure 4-4 Random number generator algorithm [27].

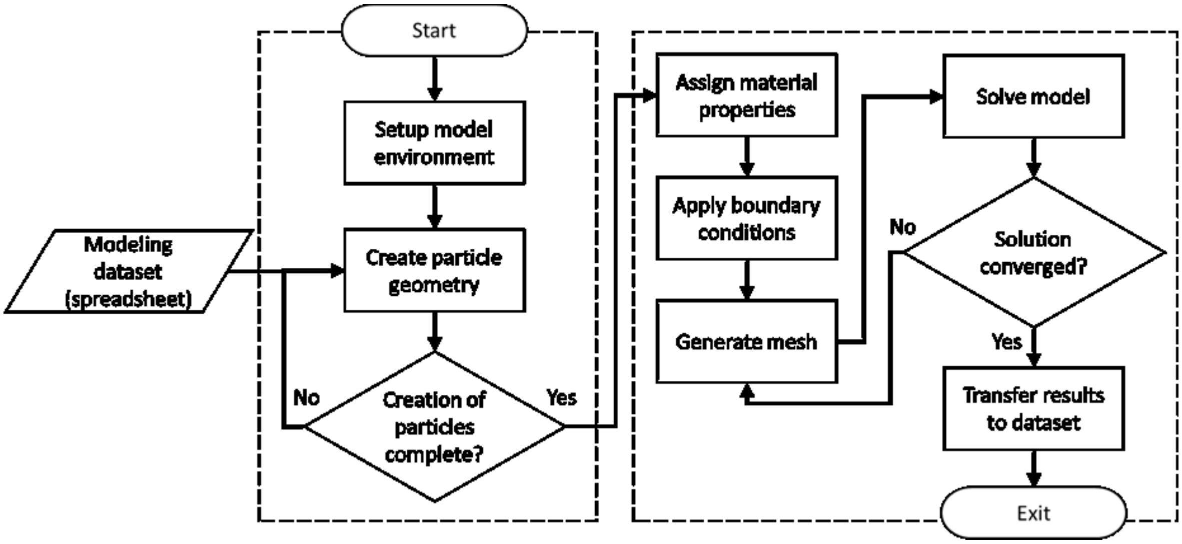

Start
Setup model environment
Modeling dataset (spreadsheet)
Create particle geometry
Creation of particles complete?
No
Yes
Assign material properties
Apply boundary conditions
Generate mesh
Solve model
Solution converged?
No
Yes
Transfer results to dataset
Exit

Figure 4-5 FEA module algorithm [27]. Schematic illustrating the algorithm for the FEA module.

4.3 Steady-state Numerical Modeling

The developed SFEA framework was employed to estimate the effective thermal conductivity under steady-state conditions of filler modified composites with randomly and aligned rod-shaped particles embedded in a polymer matrix. The rod-shaped particles mimic CNT embedded in epoxy polymer in order to elucidate the effect of filler addition and alignment in the context of fire-retardant materials. Table 4-1 shows the CNT longitudinal and lateral thermal properties and volumetric mass density, which were adopted from [9, 28, 29]. The mean particle diameter was set to 2.85 nm with a constant particle aspect ratio of 56, i.e., only a single particle size was utilized in this study to limit the parameter space affecting the results. The RVE size was set 200 nm.

Since CNT have anisotropic thermal properties it is not possible to define their thermal conductivity using global coordinates. Hence, an algorithm was developed in JAVA scripting language that provides dedicated local Cartesian coordinates for each rod-shaped particle. As depicted in Fig. 4-6, the local coordinates (x, y, z) have their origin at one end of a particle with the x-direction aligning with the particle's longitudinal axis. For the case of aligned particles the components describing the vector for each particle's major axis (x) were constrained to remain within upper and lower bounds. For the present analyses, these constraints correspond to a maximum possible angle of approximately 8.5° between a particle's major axis (x) and the global (RVE) X-direction.

Table 4-1 Volumetric mass density and thermal conductivity of polymer and CNT [9, 28, 29].

Material	Epoxy	CNT
Density (kg·m^{-3})	1,250	1,600
Thermal conductivity, longitudinal (W·m^{-1}·K^{-1})	0.25	3,500
Thermal conductivity, lateral (W·m^{-1}·K^{-1})	0.25	1.5

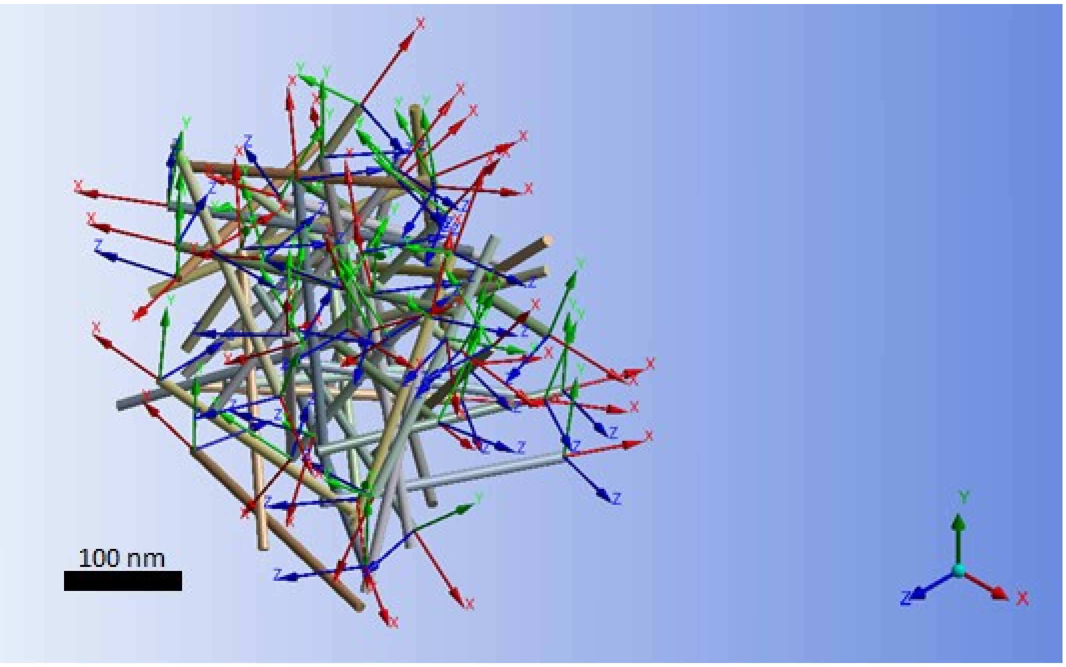


Figure 4-6 Illustration of local coordinate systems for rod-shaped particles.

Three-dimensional ten-node quadratic tetrahedral thermal solid elements, i.e., SOLID87, were used to generate the finite element mesh for both the particles and matrix. This element type, which provides one degree of freedom (temperature), is recommended for meshing irregular geometries. The latter characteristic is desirable in the present context, given that the rod- shaped particles constitute geometries that typically are difficult to mesh. As an example, Fig. 4-7 depicts the meshing generated for the matrix (left-hand side) and randomly distributed and aligned rod-shaped particles occupying the RVE devoid matrix (right-hand side).

As demonstrated in [27], it is essential to model the ITR between particles and the matrix as well as between particles that are in contact with each other in order to achieve a model that realistically captures effective thermal conductivities for different filler loadings. In this study the particle-to-matrix thermal contact conductance (TCC) was adopted from literature [30-32] as $10^8\ Wm^{-2}K^{-1}$. Also, the direct particle-to-particle heat transfer threshold was set to approximately 1 nm. Note that implementing this threshold is necessary since the employed particle collision algorithm prevents true direct particle-to-particle contact. ITR and particle-to-particle thermal contact was implemented using three-dimensional 6-node quadratic surface-to-surface elements, i.e., CONTA174 and TARGE170. For details on the chosen approach to model contact phenomena the readers are referred to [27].

Thermal boundary conditions were applied to the RVE to perform the stead-state thermal analysis and calculate effective thermal conductivities. A temperature 22 and 32°C were defined on opposite sides of the RVE, respectively, with the remaining surfaces considered adiabatic. Note that for the case of aligned filler particles, the alignment direction is referred to as the longitudinal direction, as opposed to the two lateral directions that are defined in a Cartesian coordinate system. Three sets of boundary conditions were also applied for randomly distributed filler particles for determining the effective thermal conductivities along the global Cartesian coordinate directions (X, Y, Z), thus enabling the assessment of isotropy. The applied boundary conditions create a temperature gradient and thus a heat flux between the warm and cold RVE surfaces, as illustrated in Fig. 4-8. The thermal conductivity of nodes K_i located in the warm surface was determined using Eq. (4-1).

$$K_i = \frac{Q_i \times l}{T_2 - T_1} \tag{4-1}$$

where Q is the calculated numerical total heat flux at the ith node located on the warm side, l is the RVE length, and T_1 and T_2 correspond the temperature on the warm and cold surface, respectively. Consequently, Eq. (4-2) yields the effective thermal conductivity, K_{eff}.

$$K_{\text{eff}} = \frac{\sum_{i=1}^{n} K_i}{n} \tag{4-2}$$

where n is total number nodes on the warm surface of the RVE.

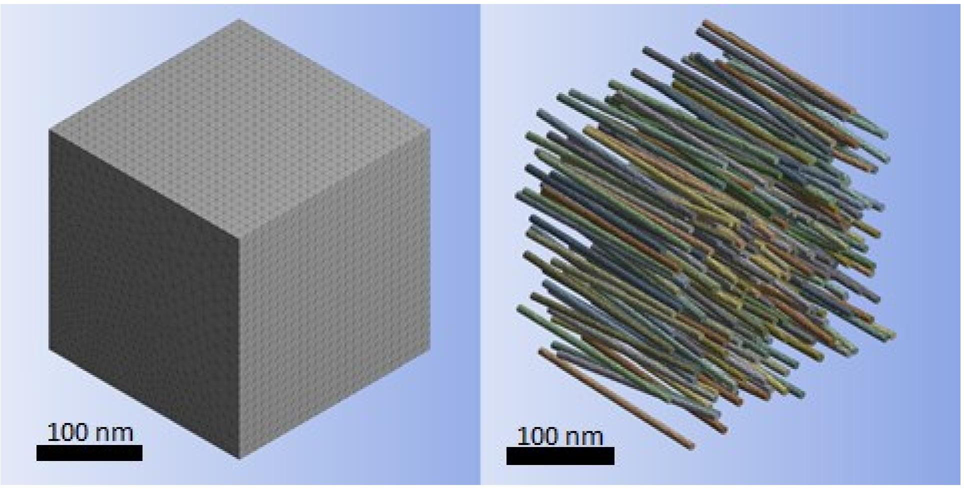


Figure 4-7 Finite element mesh of matrix (left) and aligned rod-shaped particles occupying the RVE devoid matrix (right).

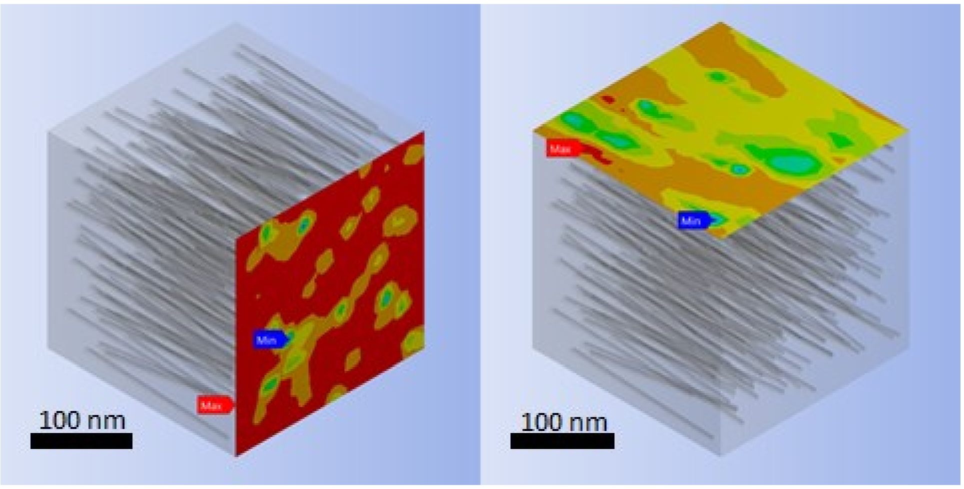


Figure 4-8 Illustration of longitudinal (left) and lateral (right) heat flux between opposing warm and cold RVE surfaces.

4.4 Results and Discussion

The described modeling framework was employed to calculate effective thermal conductivities of composites with randomly distributed particles that were either aligned or had random orientations. The studied filler volume fractions were 2.0, 4.0, 7.5 and 10%. Note that successful mesh generation becomes challenging for high filler aspect ratios, and hence, the particle aspect ratio was limited to 56 in the current study. While this value is comparatively low for CNT it does represent actual (multiwall) CNT structures as indicated in [33]. Moreover, modeling filler volume fractions exceeding 10% was found to demand excessive computational effort, and hence, analyses were limited to 10% filler volume fractions and below. Note that effective enhancement of flame retardant properties was ascertained in CNT-polymer composites that were significantly below the set 10% limit, see e.g., [7].

A convergence study was performed for a composite with randomly distributed and aligned particles at a filler volume fraction of 4.0%. To assess the sensitivity of the computed effective thermal conductivity to mesh refinement, numerical analyses were performed at different levels of mesh refinement. The results are depicted in Fig. 4-9. It was observed that changing mesh density from ~266,000 to ~376,000 degrees of freedom created a change in the effective thermal conductivity result of only less than 3.5%. Consequently, in order to maintain computational efforts within reasonable bounds, mesh generation was controlled to remain below 400,000 degrees of freedom. Therefore, it was decided to choose the global mesh size of 4% of the RVE length.

Fig. 4-10 shows an example of a model with randomly distributed and randomly oriented particles. Ideally, particle spacial distributions for this case should result in isotropic thermal conductivity properties. To further investigate this hypothesis, a study was performed in which effective thermal conductivities were computed along the global RVE directions, i.e., X, Y, Z coordinates. This analysis was completed for the chosen set of filler volume fractions (2.0, 4.0, 7.5 and 10%). Corresponding effective thermal conductivity results for a single model are included in Table 4-2. The given data indicates that thermal conductivity values were essentially isotropic for models with lower filler loading (i.e., 2%) while for increasing filler volume fractions a mild level of anisotropy was sometimes observed. For that reason, the average effective thermal conductivity was computed from the three Cartesian coordinate directions and subsequently used for comparing composites with randomly oriented particles with aligned filler composites.

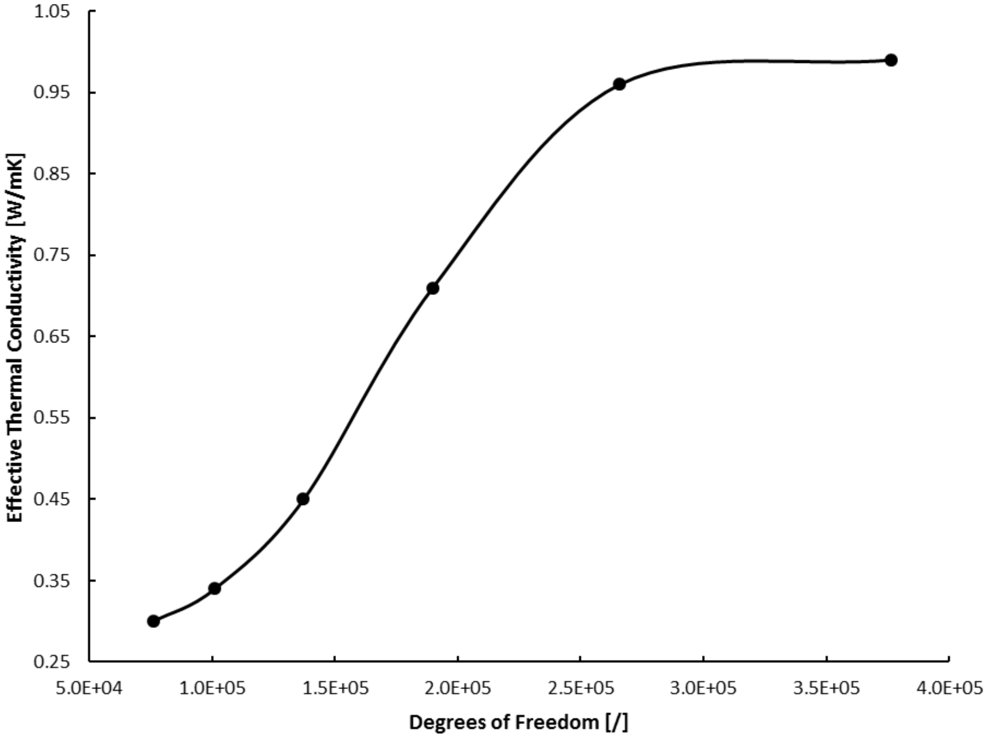


Figure 4-9 Convergence study for composites with 4% filler volume fraction.

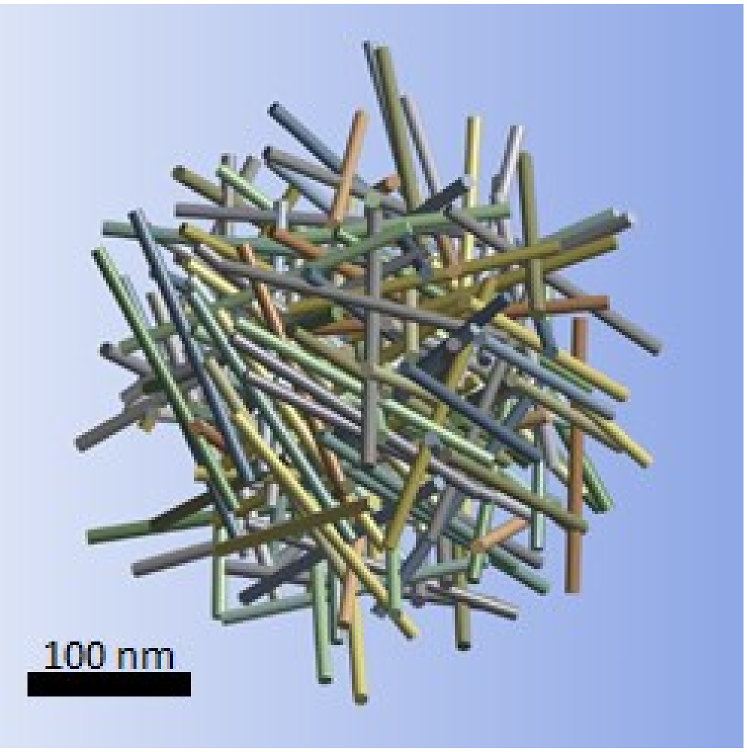


Figure 4-10 Example of randomly distributed and randomly oriented filler particles.

Table 4-2 Average and directional thermal conductivities of composites with randomly distributed and randomly oriented filler particles.

Filler volume fraction [%]	2.0%	4.0%	7.5%	10%
Thermal conductivity, X ($W{\cdot}m^{-1}{\cdot}K^{-1}$)	0.556	0.781	1.08	1.29
Thermal conductivity, Y ($W{\cdot}m^{-1}{\cdot}K^{-1}$)	0.525	0.676	1.14	1.26
Thermal conductivity, Z ($W{\cdot}m^{-1}{\cdot}K^{-1}$)	0.525	0.778	1.13	1.14
Average thermal conductivity ($W{\cdot}m^{-1}{\cdot}K^{-1}$)	0.536	0.745	1.12	1.23

As explained previously, the MCS module of the modeling framework performs numerous iterations for each filler volume fraction and subsequently computes the effective thermal conductivity and stores these data in the database. The MCS module repeats this process until specified acceptance criteria are satisfied. For the presented study, the analysis process was terminated after 100 iterations for each of the set filler volume fractions. (Alternatively, a threshold for the unbiased standard deviation or variance could be defined as a termination criterion.) The effective thermal conductivity for a certain filler volume fraction was then computed from the mean of the results stored in the database. Statistical analyses were also performed on the data in order to assess the quality of the employed stochastic process. Data plots for specific volume fractions suggest that data is normally distributed, as shown in Fig. 4-11 by the normalized probability density of effective thermal conductivity data for the direction lateral to filler alignment in a composite with 4.0% filler volume fraction. Data were computed for normality tests for each volume fraction, including the data mean, median, skewness and kurtosis. Tables 4-3 and 4-4 list corresponding results for the longitudinal and lateral directions of composites with randomly distributed and aligned particles. These results indicate that the effective thermal conductivity data obey normal distributions, e.g., mean and median of effective thermal conductivity results were practically identical (differences are less than 0.04%). Similar to results presented in [27], appropriate randomness of computed data was thus ascertained. Graphs showing normally distributed effective thermal conductivity data for the chosen filler volume fractions are plotted for the longitudinal and lateral case in Fig. 4-12 and 4-13, respectively. It is important to mention since 100 iterations were performed for calculating the standard deviations of all filler volume fractions, the standard deviations show an increasing trend by increasing filler volume fractions which was not the case in [27].

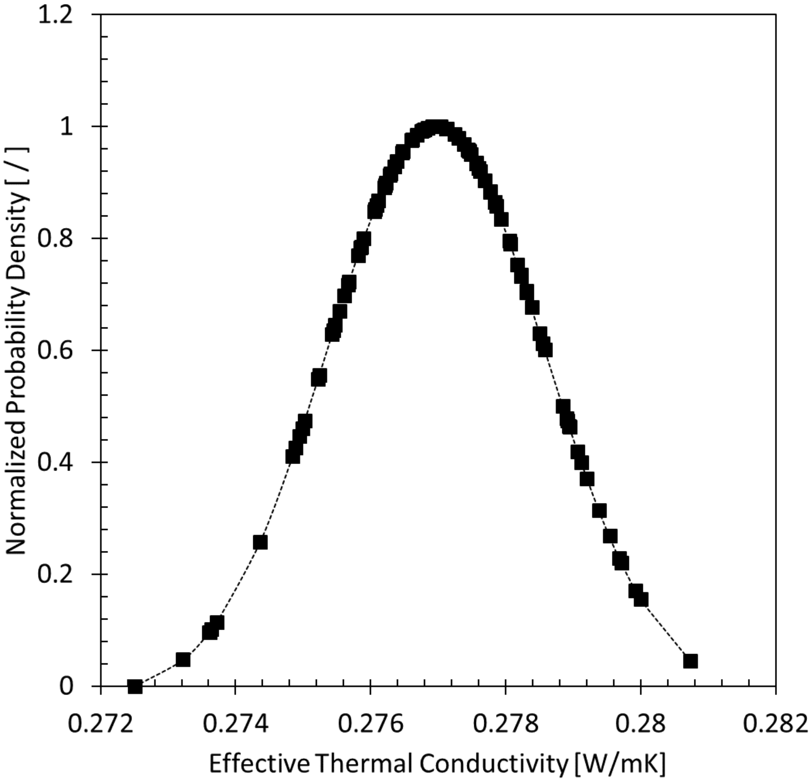


Figure 4-11 Normalized probability density of effective thermal conductivity data for the direction lateral to filler alignment and 4.0% filler volume fraction.

Table 4-3 Results for longitudinal effective thermal conductivity and statistical analyses.

Filler volume fraction [%]	2.0	4.0	7.5	10.0
Number of iterations (/)	100	100	100	100
Mean value ($W{\cdot}m^{-1}{\cdot}K^{-1}$)	0.775	1.027	1.318	1.489
Median value ($W{\cdot}m^{-1}{\cdot}K^{-1}$)	0.774	1.031	1.325	1.486
Standard deviation ($W{\cdot}m^{-1}{\cdot}K^{-1}$)	0.019	0.032	0.037	0.038
Variance ($W{\cdot}m^{-1}{\cdot}K^{-1}$)	$3.7{\cdot}10^{-4}$	$1.0{\cdot}10^{-3}$	$1.3{\cdot}10^{-3}$	$1.4{\cdot}10^{-3}$
Skewness (/)	1.197	-0.159	-0.672	0.083
Kurtosis (/)	-0.079	-0.539	0.607	0.128
95% confidence value ($W{\cdot}m^{-1}{\cdot}K^{-1}$)	0.0038	0.0064	0.0073	0.0076

Table 4-4 Results for lateral effective thermal conductivity and statistical analyses.

Filler volume fraction [%]	2.0	4.0	7.5	10.0
Number of iterations (/)	100	100	100	100
Mean value ($W{\cdot}m^{-1}{\cdot}K^{-1}$)	0.262	0.276	0.306	0.332
Median value ($W{\cdot}m^{-1}{\cdot}K^{-1}$)	0.262	0.276	0.306	0.331
Standard deviation ($W{\cdot}m^{-1}{\cdot}K^{-1}$)	0.001	0.001	0.003	0.004
Variance ($W{\cdot}m^{-1}{\cdot}K^{-1}$)	$1.6{\cdot}10^{-6}$	$2.6{\cdot}10^{-6}$	$9.4{\cdot}10^{-6}$	$2.3{\cdot}10^{-5}$
Skewness (/)	-0.819	-0.267	-0.180	0.216
Kurtosis (/)	2.410	0.031	-0.124	0.344
95% confidence value ($W{\cdot}m^{-1}{\cdot}K^{-1}$)	0.0002	0.0003	0.0006	0.0009

The effective thermal conductivity data calculated by SFEA framework can be considered continuous random variables, and hence, it is recommended to calculate the probability of occurrence of an explicit effective thermal conductivity within an identified interval. This calculation can be performed using Eq. (4-3).

$$P(a \leq X \ \leq b) = \int_a^b f(\chi)d\chi \tag{4-3}$$

where P is the probability of an event of explicit effective thermal conductivity within the interval a and b; X and $f(\chi)$ are correspondingly a continuous random variable and the probability distribution function. A cumulative distribution function (CDF) can be computed from Eq. (4-3) for each of the selected filler volume fractions. Corresponding CDF graphs are depicted in Fig. 4-14 and 15 for the longitudinal and lateral cases of aligned filler composites, respectively.

Finally, the modeling approach was used to achieve the objective of the study, that is, assessing the effect of filler addition and alignment on heat transfer into polymer composites in the context of fire-retardancy. Fig. 4-16 depicts average effective thermal conductivity results for different filler volume fractions for the cases of randomly oriented (isotropic) and aligned rod-shaped particles mimicking CNT. For the aligned filler morphologies, effective thermal conductivity results are shown for the direction of filler alignment (longitudinal) and the corresponding lateral direction. Despite the fact that the modeling approach employed generic material properties and simplifying assumptions for the CNT geometry, the data is in satisfactory agreement with data published in the technical literature (e.g., [32, 34-36]). Experimentally characterized CNT-polymer composites involve a wide range of material properties and fabrication routes. Notwithstanding these differences, the modeled thermal conductivity in the order of 1.0 $Wm^{-2}K^{-1}$ for isotropic composites with filler loadings approaching 10% are shown to be realistic.

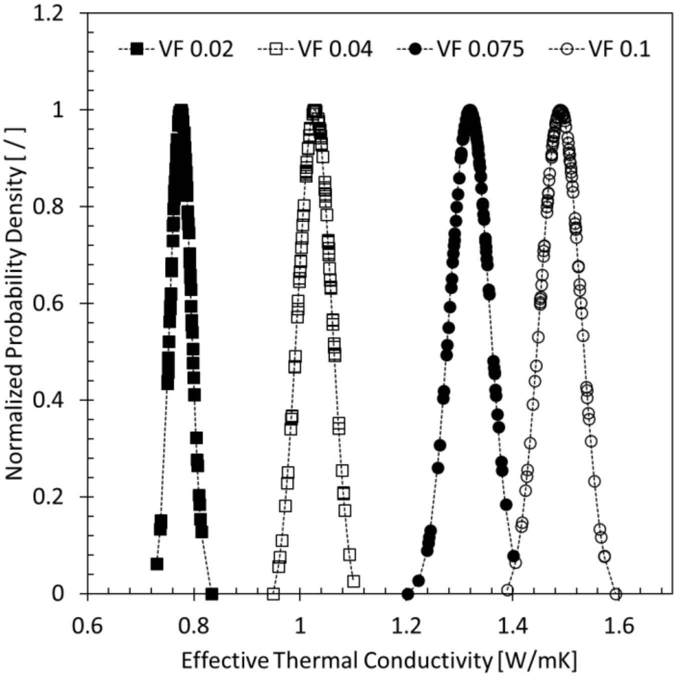


Figure 4-12 Normalized probability density of effective thermal conductivity data for the direction of filler alignment and different filler volume fractions (VF).

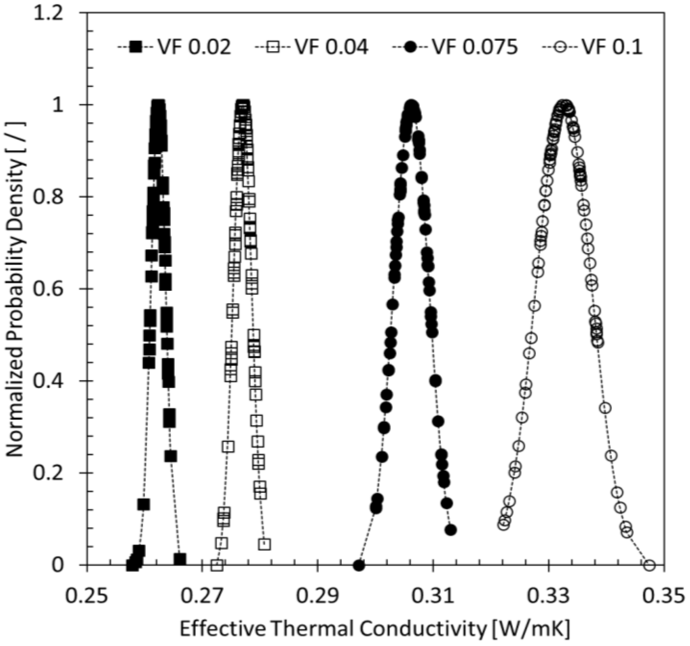


Figure 4-13 Normalized probability density of effective thermal conductivity data for the direction lateral to filler alignment and different filler volume fractions (VF).

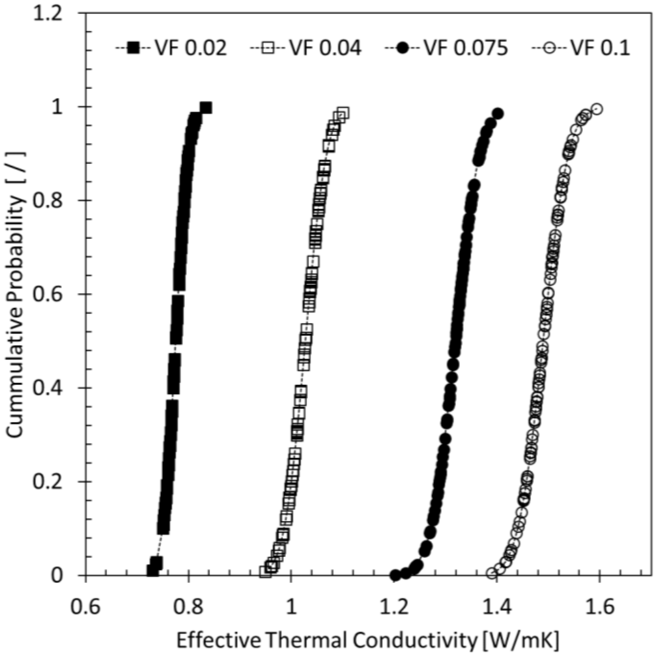


Figure 4-14 Cumulative density function of effective thermal conductivity data for the direction of filler alignment and different filler volume fractions (VF).

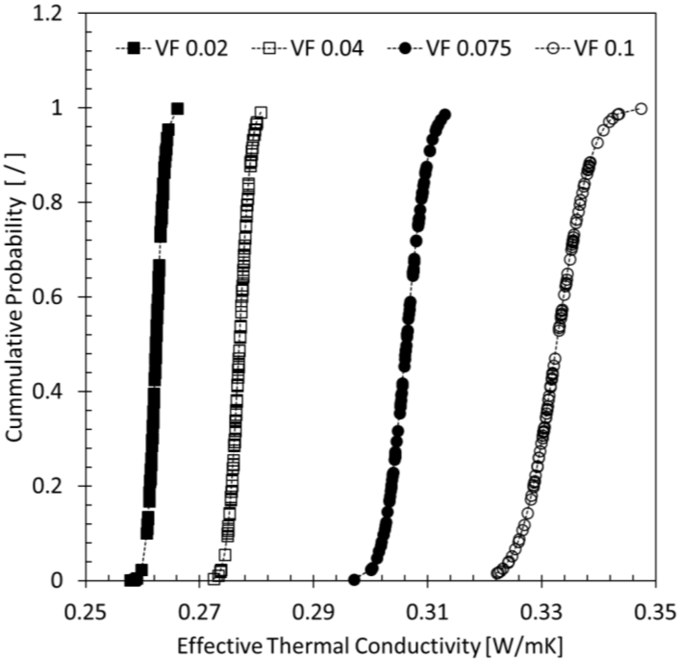


Figure 4-15 Cumulative density function of effective thermal conductivity data for the direction lateral to filler alignment and different filler volume fractions (VF).

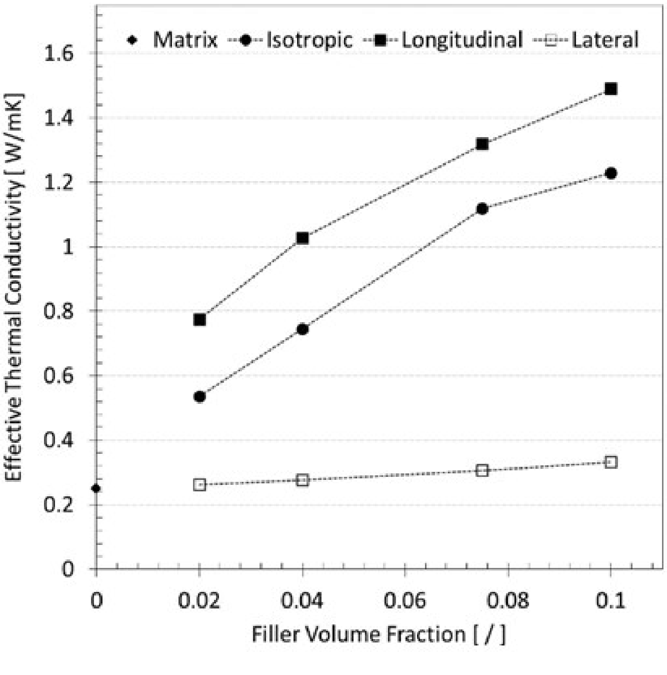


Figure 4-16 Effective thermal conductivity of randomly oriented and aligned rod-shaped particles embedded in epoxy polymer matrix.

The relative increase in effective thermal conductivity for the different filler volume fractions is depicted in the graph in Fig. 4-17 for the cases of randomly oriented (isotropic) and aligned particles (longitudinal and lateral). These data clearly demonstrate that heat transfer into the polymer can greatly be reduced when filler particles are aligned parallel to the surface of a component. For example, while thermal conductivity in an isotropic and aligned filler composite was found to respectively increase almost sixfold and fivefold over the matrix for 10% filler loading, the lateral thermal conductivity in the aligned filler composite rose only by a factor of 1.3. In the context of fire-retardancy it can therefore be concluded that aligning CNT and other high aspect ratio carbon allotrope fillers parallel to the surface of a polymer component may provide an effective means for alleviating heat input into the material while enabling the desired mass transport barrier for mitigating fuel release into the gas phase, and reduced peak heat release and radiant heat flux.

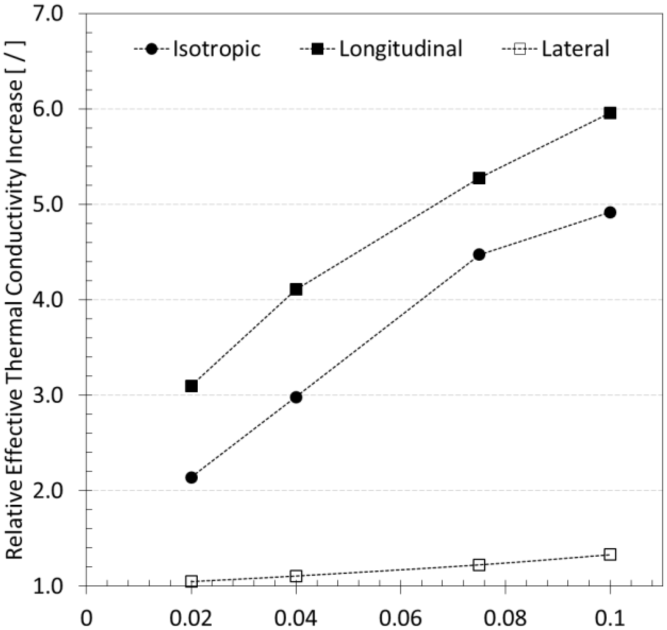


Figure 4-17 Relative increase in effective thermal conductivity of randomly oriented and aligned rod-shaped particles embedded in epoxy polymer matrix.

4.5 Conclusions

A stochastic finite element analysis framework was employed to simulate the effective thermal conductivity of randomly distributed rod-shape particles mimicking carbon nanotubes embedded in a polymer matrix. Particles were either randomly oriented or aligned, creating isotropic or anisotropic thermal conductivity behavior, respectively. The modeling framework that is based on Monte Carlo simulation considers filler-matrix and particle-to-particle interfacial effects.

The numerical study indicated that the effective thermal conductivity is greatly enhanced for aligned filler composites in the alignment direction and isotropic filler modified composites. However, in the direction lateral to filler alignment the increase in thermal conductivity is only modest. Therefore, in order to limit heat input into the material, CNT and other high aspect ratio carbon allotrope fillers may be aligned parallel to the surface of a polymer component. In this manner, the flame-retardancy effectiveness of filler modified polymer composites can be improved, while providing a mass transport barrier that lessens the release of fuel into the gas phase, peak heat release and radiant heat flux, all of which were previously described in the technical literature.

4.6 References

[1] Liu, S.; Chevalia, V.; Xu, Z.; Hui, D.; Wang H. A review of extending performance of epoxy resins using carbon nanomaterials. Composites Part B: Engineering. 2018, 136, 197-214. DOI: 10.1016/j.compositesb.2017.08.020.

[2] Mishra, D.; Mohapatra, S.; Satapathy, A. A detailed investigation on thermal and micro- structural properties of hexagonal boron nitride composites. Materials Today: Proceed- ings. 2018, 5, 19747-19753, DOI: 10.1016/j.matpr.2018.06.337.

[3] Ansari, R.; Rouhi, S.; Ahmadi, M. On the thermal conductivity of carbon nanotube/polypropylene nanocomposites by finite element method. J. Comput. Appl. Mech. 2018, 49, 70-85, DOI: 10.22059/jcamech.2017.243530.195.

[4] Shi, X.; Hassanzadeh Aghdam, M.; Ansari, R. Effect of aluminum carbide interphase on the thermomechanical behavior of carbon nanotube/aluminum nanocomposites. Proceedings of the Institution of Mechanical Engineers, Part L: Journal of Materials: Design and Applications. 2018, 233, DOI: 10.1177/1464420718794716.

[5] Ahmadi, M.; Ansari, R.; Hassanzadeh Aghdam, M. Micromechanical analysis of elastic modulus of carbon nanotube-aluminum nanocomposites with random microstructures. J. Alloys Compd. 2019, 779, 433-439, DOI: 10.1016/j.jallcom.2018.11.326.

[6] Ahmadi, M.; Ansari, R.; Rouhi, S. Investigating the thermal conductivity of concrete/graphene nanocomposite by a multi-scale modeling approach. Int. J. Mod. Phys. B. 2018, 32, 1850171, DOI: 10.1142/S0217979218501710.

[7] Kashiwagi, T.; Grulke, E.; Hilding, J.; Groth, K.; Harris, R.; Butler, K. Thermal and flammability properties of polypropylene/carbon nanotube nanocomposites. Polymer. 2004, 45, 4227-4239, DOI: 10.1016/j.polymer.2004.03.088.

[8] Liu, S.; Yan, H.; Fang, Z.; Wang, H. Effect of graphene nanosheets on morphology, thermal stability and flame retardancy of epoxy resin. Compos. Sci. Technol. 2014, 90, 40-47, DOI: 10.1016/j.compscitech.2013.10.012.

[9] Pop, E.; Mann, D.; Wang, Q.; Goodson, K.; Dai, H. Thermal conductance of an individual single-wall carbon nanotube above room temperature. Nano Letters. 2006, 6, 96-100, DOI: 10.1021/nl052145f.

[10] Balandin, A.; Ghosh, S.; Bao, W.; Calizo, I.; Teweldebrhan, D.; Miao, F. Superior thermal conductivity of single-layer graphene. Nano Letters. 2008, 8, 902-907, DOI: 10.1021/nl0731872.

[11] Liu, J.; Liu, J.; Yang, L.; Chen, X.; Zhang, M.; Meng, F. Nanomaterial-assisted signal enhancement of hybridization for DNA biosensors: A review. Sensors. 2009, 9, 7343-7364, DOI: 10.3390/s90907343.

[12] Balasubramanian, K.; Burghard, M. Biosensors based on carbon nanotubes, Anal. Bioanal. Chem. 2006, 385, 452, DOI: 10.1007/s00216-006-0314-8.

[13] Yan, T.; Wang, Z.; Wang, YQ.; Pan, Z. Carbon/graphene composite nanofiber yarns for highly sensitive strain sensors. Mater. Des. 2018, 143, 214-223, DOI: 10.1016/j.matdes.2018.02. 006.

[14] Coleman, J.; Khan, U.; Gun'ko, Y. Mechanical reinforcement of polymers using carbon nanotubes. Adv. Mater. 2006, 18, 689-706, DOI: 10.1002/adma.200501851.

[15] Moniruzzaman, M.; Winey, K. Polymer nanocomposites containing carbon nanotubes. Macromolecules. 2006, 39, 5194-5205, DOI: 10.1021/ma060733p.

[16] Green, MJ.; Behabtu, N.; Pasquali, M.; Adams, W. Nano tubes as polymer. Polymer. 2009, 50, 4979-4997, DOI: 10.1016/j.polymer.2009.07.044.

[17]Harris, P. Carbon nanotubes and related structures: New materials for the twenty-first century. Am. J. Phys. 2004, 72, 415, DOI: 10.1119/1.1645289.

[18] Dai L. Carbon Nanotechnology: Recent Developments in Chemistry, Physics, Materials Science and Device Applications. 1st ed. Amsterdam: Elsevier; 2006. ISBN-13: 978-0-444-51855-2.

[19] Wu, Q.; Zhu, W.; Zhang, C.; Liang, Z.; Wang, B. Study of fire retardant behavior of carbon nanotube membranes and carbon nanofiber paper in carbon fiber reinforced epoxy com- posites. Carbon. 2010, 48, 1799-1806, DOI: 10.1016/j.carbon.2010.01.023.

[20] Knight, C.; Filbert, I.; Zeng, C.; Zhange, C.; Wang B. A highly efficient fire-retardant nano-material based on carbon nanotubes and magnesium hydroxide. Fire Mater. 2013, 37, 91-99, DOI: 10.1002/fam.2115.

[21] Xiong, L.; Liu, K.; Shuai, J.; Hou, Z.; Zhu, L.; Li, W. Toward high strength and high electrical conductivity in super-aligned carbon nanotubes reinforced copper. Adv. Eng. Mater. 2018, 20, 1700805, DOI: 10.1002/adem.201700805.

[22] Xiong, L.; Shuai, J.; Liu, K.; Hou, Z.; Zhu, L.; Li, W. Enhanced mechanical and electrical properties of super-aligned carbon nanotubes reinforced copper by severe plastic deformation. Composites, Part B. 2018, 160, 315-320, DOI: 10.1016/j.compositesb.2018.10.023.

[23] Meincke, O.; Kaempfer, D.; Weickmann, H.; Friedrich, C.; Vathauer, M.; Warth, H. Mechanical properties and electrical conductivity of carbon-nanotube filled polyamide-6 and its blends with acrylonitrile/butadiene/styrene. Polymer, 2004, 45, 739-748, DOI: 10.1016/j.polymer.2003.12.013.

[24] Valavala, P.; Odegard, G. Modeling techniques for determination of mechanical properties of polymer nanocomposites. Adv. Mater. Sci. Eng. 2005, 9, 34-44, DOI: 10.1.1.543.5381&rep=rep1&type.

[25] Odegard, G.; Clancy, T.; Gates, T. Modeling of the mechanical properties of nanoparticle/ polymer composites. Polymer. 2005, 46, 553-562, DOI: 10.1016/j.polymer.2004.11.022.

[26] Liu, Y.; Chen, X. Evaluations of the effective material properties of carbon nanotube-based composites using a nanoscale representative volume element. Mech. Mater. 2003, 35, 69-81, DOI: 10.1016/S0167-6636(02)00200-4.

[27] Ahmadi Moghaddam, H.; Mertiny, P. Stochastic finite element analysis framework for modelling thermal conductivity of particulate modified polymer composites. Results Phys. 2018, 11, 905-914, DOI: 10.1016/j.rinp.2018.10.045.

[28] Sinha, S.; Barjami, S.; Iannacchione, G.; Schwab, A.; Muench, G. Off-axis thermal properties of carbon nanotube films. J. Nanopart. Res. 2005, 7, 651-657, DOI: 10.1007/ s11051-005-8382-9.

[29] Sugime, H.; Esconjauregui, S.; Yang, J.; D'Arsié, L.; Oliver, R.; Bhardwaj, S. Low temperature growth of ultra-high mass density carbon nanotube forests on conductive supports. Appl. Phys. Lett. 2013, 103, 109901, DOI: 10.1063/1.4818619.

[30] Huxtable, S.; Cahill, D.; Shenogin, S.; Xue, L.; Ozisik, R.; Barone, P. Interfacial heat flow in carbon nanotube suspensions. Nat. Mater. 2003, 2, 731-734, DOI: 10.1038/nmat996.

[31]Shenogin, S.; Xue, L.; Ozisik, R.; Keblinski, P.; Cahill, D. Role of thermal boundary resistance on the heat flow in carbon-nanotube composites. J. Appl. Phys. 2004, 95, 8136-8144, DOI: 10.1063/1.1736328.

[32] Han, Z.; Fina, A. Thermal conductivity of carbon nanotubes and their polymer nanocomposites: A review. Prog. Polym. Sci. 2011, 36, 914-944, DOI: 10.1016/j.progpolymsci.2010.11.004.

[33] Xu, D.; Wang, Z.; Douglas, J. Influence of carbon nanotube aspect ratio on normal stress differences in isotactic polypropylene nanocomposite melts. Macromolecules. 2008, 41, 815-825, DOI: 10.1021/ma702178e.

[34] Jiang, Q.; Wang, X.; Zhu, Y.; Hui, D.; Qiu, Y. Mechanical, electrical and thermal properties of aligned carbon nanotube/polyimide composites. Composites, Part B. 2014, 56, 408-412, DOI: 10.1016/j.compositesb.2013.08.064.

[35] Pan, C.; Hocheng, H. Evaluation of anisotropic thermal conductivity for unidirectional FRP in laser machining. Composites, Part A. 2001, 32, 1657-1667, DOI: 10.1016/S1359-835X(00)00093-2.

[36] Yang, K.; Gu, M.; Guo, Y.; Pan, X.; Mu, G. Effects of carbon nanotube functionalization on the mechanical and thermal properties of epoxy composites. Carbon. 2009, 47, 1723-1737, DOI: 10.1016/j.carbon.2009.02.029.

Chapter 5
Electrical Characterization of Particulate Polymer Composites

In the fourth part of this research study, a stochastic FEA framework was developed that enabled calculating effective electrical properties of randomly distributed and dispersed spherical particulate polymer composites. This work was published as publication no. 5 listed in the Preface of this thesis document.

Nomenclature

e	element
d	tunneling distance
I_i^e	current at element
P	probability
R^e	resistance at element
χ	random variable
V_i	Node i electrical potential
$f(\chi)$	probability density function
V_j	Node j electrical potential
P_{th}	percolation probability
K_{ij}^e	eelectrical stiffness matrix
A	number of iteration percolation occurred

EC_i	node electrical conductivity
I_i	electrical current density
N	total number of iterations
D	RVE length
T	temperature
V_L	RVE front surface voltage
T_0	absolute zero
V_R	RVE back surface voltage
EC_eff	effective electrical conductivity
n	number of nodes
J	current density
ECC	electrical contact conductance
V_t	voltage at target node
V_C	voltage at contact node
ρ_{tunl}	tunneling resistance
m	electron mass
h	Planck’s constant
λ	barrier height
e	quantum of electricity

5.1 Introduction

Since the middle of the 20th century, polymers have seen rapid deployment in consumer products and industrial applications. Concurrently, researchers have sought to improve polymer mechanical, thermal and electrical properties by adding appropriate fillers [1–4]. Carbon black (CB) [5,6], carbon nanotubes (CNT) [7–10] and nano-silver particles [11–15] are some common fillers used for enhancing mechanical, thermal and electrical properties of particulate polymer composites. Industrial applications for such materials include high-voltage and temperature devices, heaters [16–18] and electromagnetic interference (EMI) shielding [19–21]. The vast diversity of particle materials and morphologies (shape, dimensions, size distribution) poses significant challenges for material designers seeking to effectively develop multifunctional particulate polymer composites that meet desired properties. Analytical and experimental methods are available to explore the material design space. However, analytical methods typically have limited detailedness, accuracy and versatility, while experimental methods are associated with substantial time and cost, making them less attractive. Therefore, new methods for predicting the properties of filler-modified polymers are sought [22–29].

Numerical techniques, especially finite element analysis (FEA), have become popular tools for predicting mechanical [30–33] and thermal [34–36] properties of particulate polymer composites using a representative volume element (RVE) concept. Also, several studies explored the electrical properties of particulate composites using numerical approaches. Pike et al. [37] are pioneers, studying electrical properties such as the percolation threshold, i.e., the minimum amount of filler required for establishing a transfer of electrical charge, in polymer composites using two-dimensional numerical models involving Monte Carlo (MC) simulation. Kirkpatrick [38] and Behnam and Ural [39] also developed two-dimensional numerical models that enabled the prediction of electrical properties of randomly oriented and dispersed CNT in conjunction with an MC approach. Various models have been proposed based on resistor networks to facilitate the prediction of the electrical properties of particulate polymer composites [40–48].

The transfer of electrical charge in polymer composites is largely controlled by the quantum mechanical phenomenon of "electron tunneling" [49–53], that is, the transfer of electron electrical charge may occur from one particle to another through an insulator barrier if the distance between the particles is less than an explicit value. This effect brings forth a nonlinear current-voltage relation between two particles. Given a sufficiently high particle concentration and suitable particle dispersion, electrical paths in the form of a continuous conducting structure or network allow electrons and thus electrical current to flow through the material [54]. The percolation model described in Reference [55] considers two types of electrical barriers which are mimicked by electrical resistors, i.e., a tunneling resistance and a contact resistance; the latter relates to particles in direct mechanical contact. The combination of these resistances again gives rise to a nonlinear current-voltage behavior. Notably, it was observed that electrical conduction in particulate polymer composites is affected by temperature [56-57], where an increase in temperature led to an increase in electrical conductivity.

In the context of FEA, the system of equations that represents electrical conductivity, i.e., Ohm's law, can be written for a linear electrical element as

$$I_i^e = \frac{1}{R^e}(V_i - V_j) \tag{5-1}$$

where V_i and V_j are the electrical potential (voltage) at nodes i and j, respectively, I and R are correspondingly current and resistance at element e. This expression can be written in matrix form as follows:

$$\begin{Bmatrix} I_i^e \\ I_j^e \end{Bmatrix} = \left[K_{ij}^e\right] \begin{Bmatrix} V_i \\ V_j \end{Bmatrix} \tag{5-2}$$

where K_{ij}^e is known as the electrical stiffness matrix, which is defined as

$$\left[K_{ij}^e\right] = \frac{1}{R^e}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \tag{5-3}$$

The effective electrical properties, i.e., the effective electrical conductivity of an RVE (representing e.g., a particulate polymer composite), can be calculated as follows [44,45]:

$$EC_i = \frac{I_i \times D}{(V_{\mathrm{L}} - V_{\mathrm{R}})} \tag{5-4}$$

$$EC_{\mathrm{eff}} = \sum_{i=1}^{n} EC_i \tag{5-5}$$

where EC_i and I_i are correspondingly the electrical conductivity, in units of Siemens per meter (S/m), and electrical current density (units A/m^2), at the i^{th} node located on the face of an RVE exposed to an electrical charge. D is the RVE characteristic length. V_{L} and V_{R} are voltages that are applied to opposing RVE faces, i.e., a 'left' and 'right' side. EC_{eff} is the effective electrical conductivity with n being the number of nodes located on each of the RVE faces exposed to an electrical charge.

The distribution of particles in polymer composites, and thus its electrical properties, are statistical in nature. Hence, in this paper, a stochastic FEA (SFEA) framework was employed that enables prediction of the effective electrical conductivity and percolation threshold of particulate polymer composites. Interested readers are referred to Reference [58] for detailed information on the SFEA framework concept, including a consideration associated with MC simulation and random number generation for creating true randomness.

Applying mechanical strain to a polymer modified with a conductive filler such as nano-silver particles may result in the phenomenon known as piezoresistivity, that is, resistivity changes occur as the material is subjected to mechanical strain. This effect is the result of changing distances between conductive particles, as well as changes in particle orientation for cases when orientation matters, i.e., non-spherical particles (e.g., cylindrical, ellipsoidal, and disk-shaped particles). Materials exhibiting piezoresistivity may be good candidates for making sensors that provide deformation-based measurements.

Several numerical and analytical methods have been developed to investigate the piezoresistivity of particulate polymer composites [59–63]. In many of these studies, a resistor network was created to represent the particles and their electrical interaction; the polymer matrix was typically not explicitly modeled as a continuum. Such a modeling approach, while expedient, fails to capture mechanical interactions of the composite constituents, such as the deformation of particles due to arising stress/strain in the composite, and hence, the final model predictions may be compromised. In contrast, the SFEA framework employed in the presented study includes both the matrix material and embedded particles in order to predict piezoresistivity. It is postulated that accurate results can thus be achieved since this approach enables the calculation of particle locations, orientations, and deformations precisely as a result of not only considering the global but also the local mechanical strain. Moreover, the effect of material parameters Poisson's ratio and Young's modulus on piezoresistivity can be investigated.

Employing the SFEA framework, filler-modified polymers were herein modeled, and their electrical properties predicted (i.e., conductivity, percolation and piezoresistivity), including the effect of temperature. The composites comprised randomly distributed and dispersed filler particles. Spherical nano-silver particles embedded in epoxy polymer were considered in this study. Modeling results were compared with values from the technical literature in order to demonstrate the viability of the developed modeling approach.

5.2 Overview of SFEA Framework

In order to predict the effective electrical conductivity and percolation threshold of particulate polymer composite, an SFEA framework was created using multiple programing languages. A schematic of the framework is depicted in Fig. 5-1.

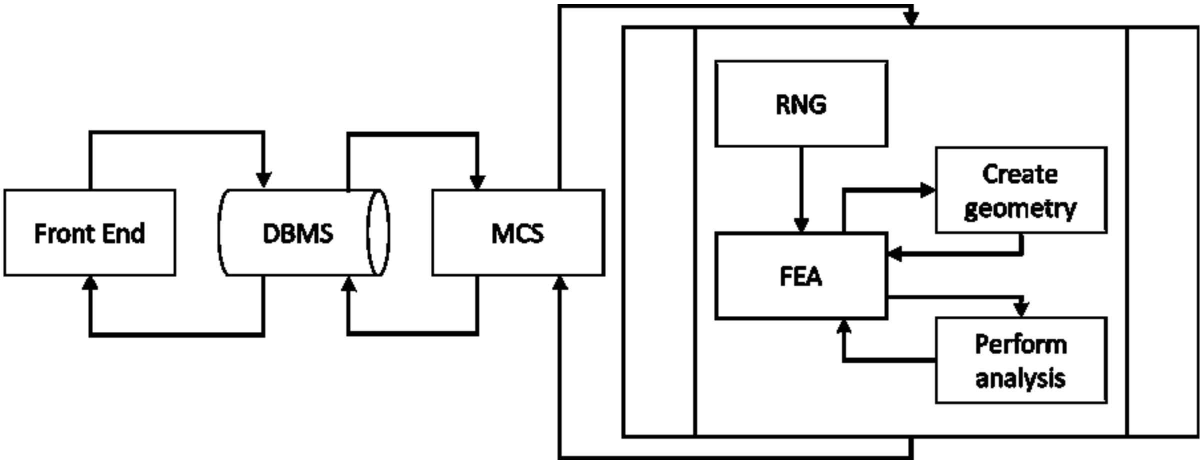


Figure 5-1 Schematic of the SFEA framework showing connections between individual modules [58].

Visual Basic for Applications (VBA; Microsoft, Redmond, WA, USA) programming language was used to create a domain that connects the various modules developed for the framework. The user interacts with the framework via the “Front End”, which was written in VBA programming language. The Front End enables the capture of information required for the analysis, including RVE size, particle size distribution, electrical properties (e.g., the electrical resistivity of polymer matrix as well as filler, electrical conductance between polymer matrix and particles, the tunneling distance and electrical conductance between particles, electrical boundary conditions), and finally parameters defining the mesh for the FEA model. Data captured by the Front End are stored in tabulated format within a database. An Open Database Connectivity concept was used to enable accessing the Database Management System (DBMS) module. This method facilitates access to the database at any time during the numerical analysis. All information saved in the DBMS module is transferred to the Monte Carlo Simulation (MCS) module, which is the core of the SFEA framework. The MCS module was developed in tabulated format using VBA programming language, which enables storing input parameters as well as saving results calculated by SFEA framework as illustrated in Fig. 5-2. The subprocess shown in Fig. 5-1 is iterated as part of the MCS module in order to calculate the effective electrical conductivity and probability of passing the electrical current from one side of an RVE to the other in order to identify the percolation threshold as the volume fraction is increased. Once a user-defined terminating number of iterations is reached, or the standard deviation of the dataset is below a threshold set by the user prior to starting the SFEA framework execution, the MCS module stops iterating, and calculated results are transferred and stored in the database, which can be accessed by the user from the Front End. In typical fashion, increasing the number of iterations will increase the accuracy of predicting the effective electrical conductivity as well as the percolation threshold for the simulated material system.

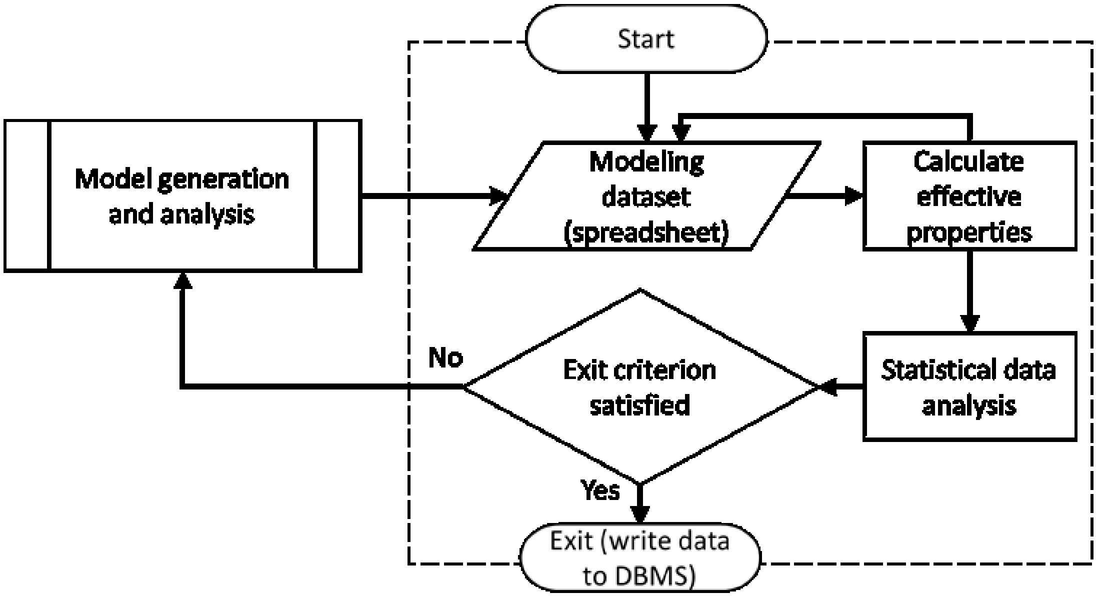

Start
Model generation and analysis
Modeling dataset (spreadsheet)
Calculate effective properties
Statistical data analysis
Exit criterion satisfied
No
Yes
Exit (write data to DBMS)

Figure 5-2 Schematic illustrating the algorithm for the Monte Carlo Simulation (MCS) module [58].

The MC subprocess, as illustrated in Fig. 5-1, starts with the Random Number Generator (RNG) module, for which a schematic is shown in Fig. 5-3. This module facilitates the process of generating random numbers required for creating the particulate polymer composite morphology, e.g., particles' coordinates and size distribution. Interested readers are referred to Reference [58] for additional information on how the RNG module creates random numbers that conform to a given particle size distribution. The RNG module was developed in the general mathematical programming environment MATLAB (MathWorks, Natick, MA, USA) with the goal of creating true randomness in the SFEA framework.

Since in an actual material, particles do not intersect with each other, i.e., they cannot occupy the same space, the RNG module performs a collision detection for the particles contained within the RVE. If a newly added particle intersects with either the RVE surface or other particles already within the RVE, the particle is rejected, and a new particle is created instead. This process continues until the volume fraction defined by the user in the Front End is satisfied. The RNG module stores the data in tabulate format within a database which is accessed through the FEA module for creating the finite element model.

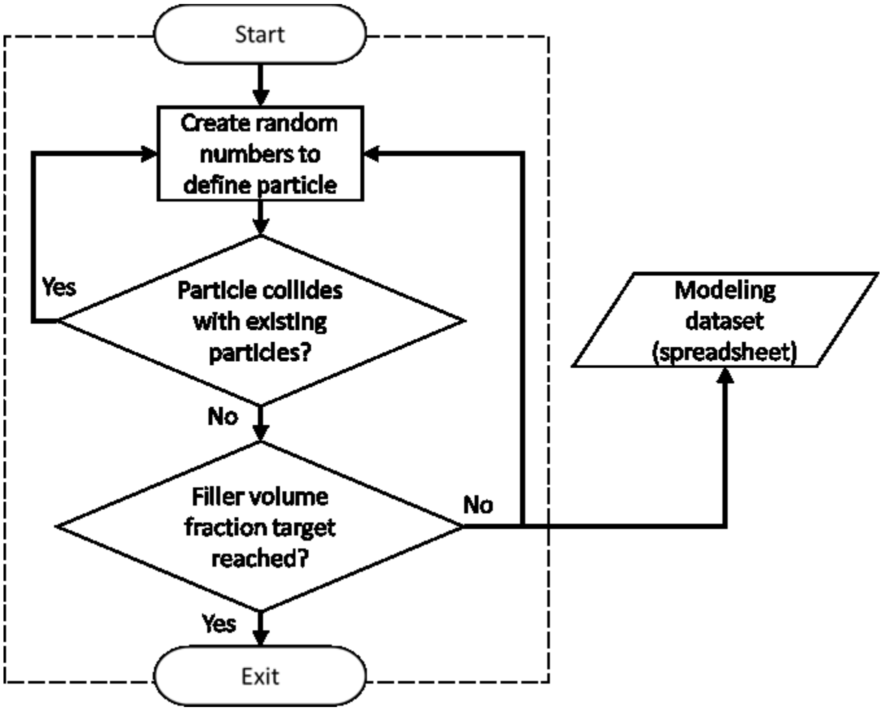

Start
Create random numbers to define particle
Yes
Particle collides with existing particles?
No
Filler volume fraction target reached?
No
Yes
Exit
Modeling dataset (spreadsheet)

Figure 5-3 Schematic illustrating the Random Number Generator (RNG) module [58].

The commercial FEA software package ANSYS Workbench (Version 19, ANSYS Inc., Canonsburg, PA, USA) was employed for creating the material model. IronPython programming language was used for developing a customized FEA module as shown in Fig. 5-4. Since in ANSYS Workbench the model generation environment (ANSYS DesignModeler) is separate from the FEA solution environment (ANSYS Mechanical), two different customized modules were developed using JavaScript programing language. The setup enables automating of the process of reading random numbers from the database, creating the particulate polymer composite geometry (i.e., the RVE in DesignModeler), assigning electrical properties to the polymer matrix and the particles, defining contact between particles and the polymer matrix as well as contact between particles (i.e., ‘contact’ implies the tunneling phenomenon), forming electrical boundary conditions, and setting parameters required for mesh generation. Results in terms of electrical conductivity calculated for each iteration are transferred to the MCS module for storage in tabulated format for further statistical analyses.

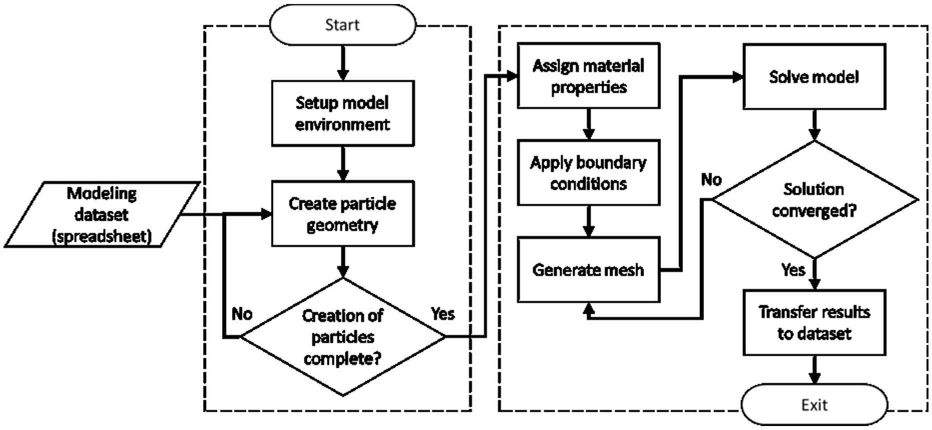

Start
Setup model environment
Modeling dataset (spreadsheet)
Create particle geometry
No
Creation of particles complete?
Yes
Assign material properties
Apply boundary conditions
Generate mesh
Solve model
No
Solution converged?
Yes
Transfer results to dataset
Exit

Figure 5-4 Schematic illustrating the algorithm for the FEA module [58].

5.3 Steady-State Electric Conduction Numerical Model

5.3.1 Electrical Conduction and Percolation Threshold Modeling

Steady-state electric conduction numerical modeling was performed using the SFEA framework for predicting the effective electrical conductivity and electrical percolation threshold of particulate polymer composites. In the present paper, material systems with spherical-shape particles were modeled, and it was decided to predict the electrical properties of silver nano-particles embedded in an epoxy polymer matrix. The electrical properties of particles and matrix as shown in Table 5-1 were considered for the model. Parameters determining particle sizes, for use with the RNG module, were adjusted so that the filler conforms to a distribution with average particle diameters of 3 nm, 5 nm and 7 nm with a size variation of ±5 percent from the mean. These sizes were adopted from TEM images presented in Reference [14], where nano-silver particles were reported to aggregate forming clusters. The present modeling approach could thus serve to explore the properties of nano-silver clusters or an assumed macro nanocomposite with well-dispersed and distributed nano-particles. In terms of RVE size, a desirable dimension would ensure the true randomness of the model. Hence, as suggested in reference [61], the RVE size was set to ten times greater than the particle dimensions, which was found to be large enough to satisfy randomness in the model.

Three-dimensional twenty-node electric solid elements (SOLID231) were used for generating the mesh for two electric charge plates placed at the back and the front-side of the RVE, as illustrated in Fig. 5-5. The chosen element, which is based on an electric scalar potential formulation, has only one degree of freedom (voltage) at each node and can be used for modelling irregular shapes without losing accuracy. Three-dimensional ten-node quadratic tetrahedral electric solid elements (SOLID232), carrying only one degree of freedom (voltage) were used for modelling the polymer matrix as well as the filler. Nodes located in the plane of contact between the polymer matrix and the electrical charge plates were merged to avoid any discontinuity in the model and increase result accuracy.

Table 5-1 Electrical properties of polymer matrix and particles [10].

Property	Epoxy	Silver
Isotropic Electrical Resistivity [ohm.m]	1.00E+10	1.59E-8

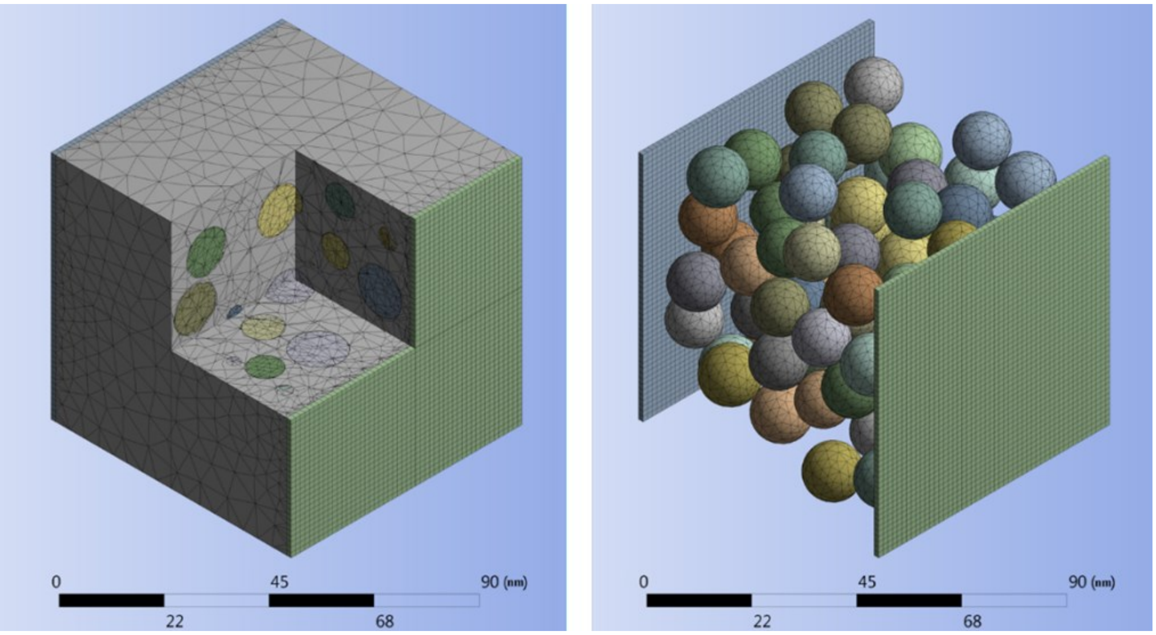


Figure 5-5 A cut-away view of the finite element mesh for RVE and electrical charge plates (left) and an isolated view of electrical charge plates and particles (right) for the 0.3 filler volume fraction.

four forms of contact, i.e., particle-to-matrix, particle-to-particle, particle-to-electric charge plate and electric charge plate-to-matrix, were implemented to essentially establish a numerical resistor network. The modeling concept enabling a direct particle-to-particle electrical current is schematically depicted in Fig. 5-6. Interested readers are referred to Reference [58] for further information on the contact element zone simulating direct particle-to-particle contact (in the context of heat transfer). Three-dimensional six-node quadratic surface-to-surface structural-thermal-electric coupled field elements (CONTA174 and TARGE170) were used for modelling electric current conduction between the RVE constituents. Since structural and thermal aspects were not the focus of the present analysis, KEYOPT (1) was used to set the required degree of freedom for modelling electric contact. The surface electric interaction between the polymer matrix and particles was defined employing the concept of 'electric contact conductance' (ECC) per unit area as described by Eq. (6-6).

$$J = ECC(V_t - V_C) \tag{5-6}$$

where J and ECC are the current density and electric contact conductance for an electric potential (voltage), respectively; V_t and V_C are correspondingly the voltages at the target and contact surfaces. While the ECC can be a function of temperature and pressure existing at the contact, in this study, temperature and pressure effects were neglected. A small ECC of $10^{-4}\ S/m^2$ was used for defining the contact from the polymer matrix to particles and electric charge plates. Due to this contact setting, the polymer matrix has only a minimal contribution to the effective electrical conductivity in the resistor network, which is akin to other works employing a resistor network method [40–48]. However, since the present authors seek to also explore the effects of applied mechanical strain and temperature change on the effective electrical conductivity, representing the matrix in the numerical model is crucial for enabling sequential multiphysics simulations.

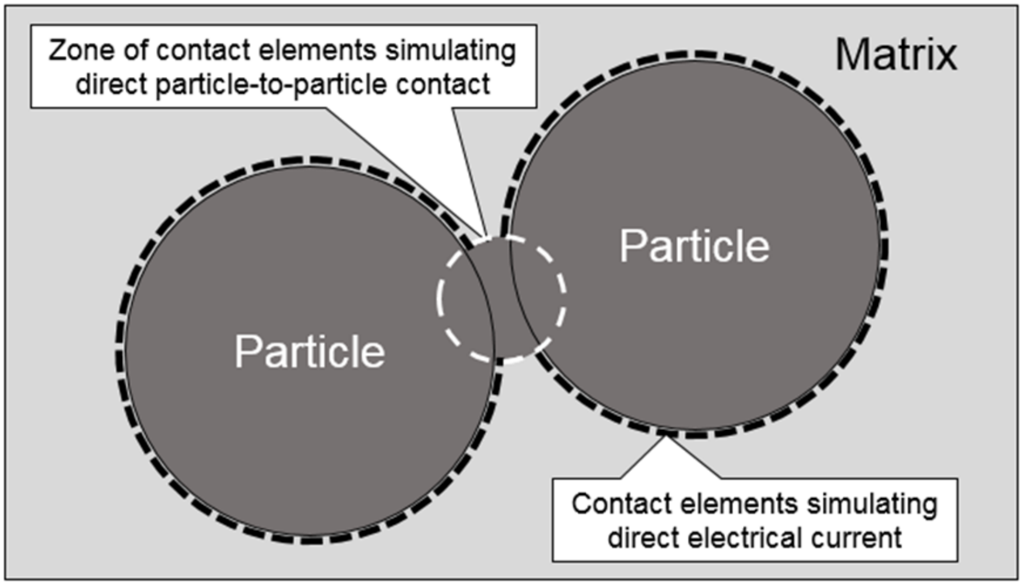

Zone of contact elements simulating direct particle-to-particle contact
Matrix
Particle
Particle
Contact elements simulating direct electrical current

Figure 5-6 A schematic depicting the modeling approach for direct particle-to-particle electrical contact.

As mentioned previously, the electron tunneling effect dominates the electrical conductivity of particulate polymer composites, and hence, modeling this effect with the SFEA framework is a key aspect for predicting the effective electrical conductivity and percolation behavior. A script was written in JavaScript programming language that automatically measures the distance between particles within the RVE, as well as the distance between particles and electrical charge plates, for defining another type of contact in ANSYS Mechanical. If the measured distance was less than an explicit threshold, i.e., the tunneling distance, a contact was defined that permits a transfer of electrical charge. The minimum distance required for transferring a charge can be measured experimentally [11,12]. The 'tunneling' contact was created using three-dimensional six-node quadratic surface-to-surface structural-thermal-electric coupled field elements (CONTA174 and TARGE170), which allow 'direct electrical conduction' (DEC) between particles within the RVE when the distance between particles and the distance between particles and an electric charge plate is less than the tunneling distance. The ECC value was approximated based on Eq. (6-7) [64].

$$\rho_{tunl} = \frac{h^2}{e^2\sqrt{2m\lambda}} exp(\frac{4\pi d}{h}\sqrt{2m\lambda}) \tag{5-7}$$

where m and h are the electron mass and Planck's constant, respectively; λ and e are correspondingly the polymer barrier height and the quantum of electricity; and d is the parameter defining the tunneling distance. Upon the material system reaching percolation, Eq. (6-7) affects the material electrical resistivity, imposing a nonlinear behavior between tunneling distance and electrical resistivity. As suggested in Reference [11], the barrier height can vary from 1 eV to 4 eV. In this study, λ was chosen as 1.5 eV. The technical literature describes a range of experimentally measured tunneling distances [11–13] varying from 0.5 to 5 nm. For the present analysis, a tunneling distance of 1 nm and 1.5 nm was set for assessing percolation behavior and effective electrical conductivity. As shown in Fig. 5-7, electrical conductivity diminishes for tunneling distances greater than 1.5 nm for the considered barrier height value of 1.5 eV.

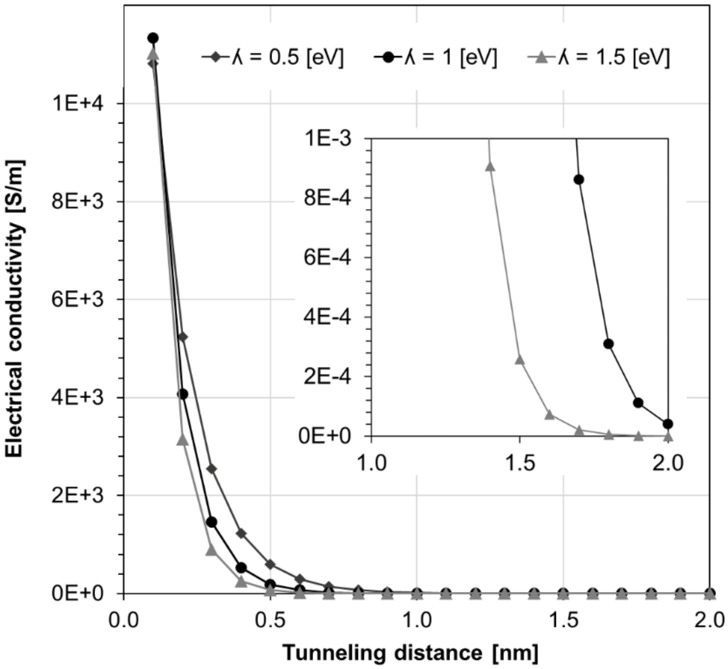


Figure 5-7 Electrical conductivity versus tunneling distance for different polymer barrier heights λ, based on Eq. (6-7).

In terms of RVE boundary conditions, a low electrical potential of 0.2 V was applied between the electric charge plates. In the case of electrical percolation, this boundary condition produces direct current (DC) flowing between the electrical charge plates. It is a necessary condition that the average current density on all nodes located on the sides of the RVE with an electrical charge plate, i.e., the total current entering and exiting the RVE, is equal. Fig. 5-8 depicts an example of an electrical current density distribution that was simulated after a percolation condition was achieved. It is interesting to note that the minimum number of electrical paths required for achieving percolation is one. Therefore, in models with low filler loading, a large portion of particles may not influence the electrical properties of the polymer composite. Particles located in the electrical path are known as “backbone” particles [38] with non-zero current [46].

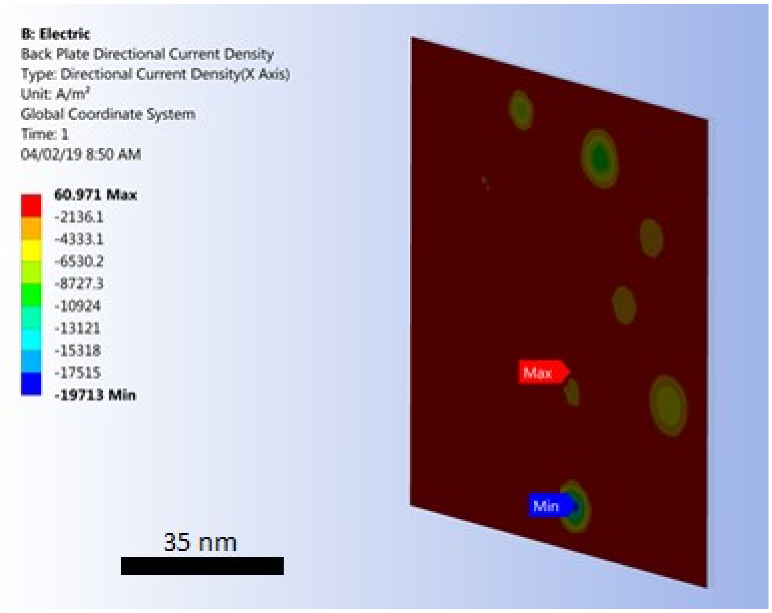


Figure 5-8 Example of simulated electrical current density on an electrical charge plate after achieving percolation conditions.

5.3.2 Electrical Piezoresistivity Modeling

The developed SFEA framework was also employed for exploring electrical piezoresistivity effects of particulate polymer composites. For this part of the study, the average particle diameter was set to 3 nm with a size variation of ±5 percent from the mean. An RVE size of 30 nm was found to be sufficient to achieve random material systems. For the deformation-based analyses, mechanical properties for the filler and matrix were used as shown in Table 5-2.

Table 5-2 Mechanical properties of nano-silver particles and the polymer matrix.

Property	Epoxy	Nano Silver particles
Density $[Kg/m^3]$	1,280	10,300
Modulus of Elasticity $[Gpa]$	3.0	476
Poisson’s Ratio	0.4	0.36

A static structural numerical model was created using a three-dimensional 10-node quadratic tetrahedral structural solid elements (SOLID187) for generating the finite element mesh for particles and the matrix, as depicted in Fig. 5-5. Three-dimensional eight-node surface-to-surface contact elements (CONTA174 and TARGE170) were used for defining the contact between particles and the matrix, restraining any relative displacement between a particle and the surrounding matrix.

The following boundary conditions were applied to the static structural numerical model to simulate the displacement and deformation of particles within the RVE. A Cartesian displacement boundary condition was applied to the RVE that restrained nodes located on one face from moving perpendicular to the face while permitting displacements in the transverse direction. Another uniform Cartesian displacement boundary condition was applied to the opposing face that forced all the nodes located on this surface to displace by an explicit value in the direction perpendicular to the surface, again with the freedom to displace laterally. To restrain the RVE from rigid body motion, a Cartesian zero displacement boundary condition was applied to one of the RVE's corners.

The above boundary conditions enabled applying mechanical strain to the material system and predicting changes in composite morphology. A script was developed using JavaScript programming language that facilitates extracting the geometry data of the deformed body (i.e., particle sizes and location coordinates) and the saving of this information in tabulated format. This data was used for generating a post-deformation steady-state electric conduction numerical model (as explained in the previous section) in order to calculate the effective electrical conductivity of the deformed material system.

5.3.3 Thermal-Electrical Numerical Model

The developed SFEA framework was further used to explore the effects of temperature on the effective electrical conductivity of a particulate polymer composite. The same material system was herein used as described in the section on piezoresistivity modeling above. In addition to mechanical properties, the coefficients of thermal expansion, as shown in Table 5-3, were used for the nano-silver and epoxy material.

Table 5-3 Coefficient of thermal expansion of fillers and matrix.

Property	Epoxy	Nano Silver particles
Coefficient of thermal expansion $[K^{-1}]$	450E-6	18E-6

A sequential structural-thermal numerical model was created for calculating the effective electrical conductivity of particulate polymer composites subjected to temperature change. The same element types, as well as mesh properties described in the previous section, were employed for this modeling approach since the utilized elements possess the degrees of freedom required for considering temperature in the numerical model.

Mechanical boundary conditions were set akin to the piezoresistivity model. In addition, a body temperature was applied to the RVE, enabling the simulation of a temperature change from ambient conditions, i.e., an initial temperature of 22 $°C$, to an elevated temperature. The applied mechanical-thermal boundary conditions thus impose the thermal expansion of both the particles and polymer matrix, and in consequence, changes in the location and size of particles, which in turn may affect the material electrical properties. The JavaScript program described in the previous section was again used for extracting and saving particle locations and sizes for performing a steady-state electric conduction numerical model for calculating the effective electrical conductivity following a temperature change.

5.4 Results and Discussion

5.4.1 Effective Electrical Conductivity and Percolation Behavior

The effective electrical conductivity for an epoxy nanocomposite with nano-silver particles was computed using the aforementioned properties for filler volume fractions ranging from 3 vol% to 30 vol% with an interval of 3 vol%. The first step of the analysis was performing a convergence study where the effective electrical conductivity of the composite was determined for a few different levels of mesh refinement. Results shown in Fig. 5-9 are for the case of 21 vol% filler loading and particle size and tunneling distance 3 nm and 1.5 nm, respectively. It was observed that the effective electrical conductivity changed by less than 5 percent when increasing the number of degrees of freedom from approximately 210,000 to 290,000, and hence, the mesh refinement corresponding to 210,000 degrees of freedom was deemed to be sufficiently fine for being employed in the SFEA framework for all volume fractions. Consequently, it was decided to choose a global mesh size of 2% of the RVE length. Due to the stochastic nature of the chosen modeling approach, numerous model runs are needed for calculating the effective properties; therefore, minimizing the number of nodes in the model is critical for maintaining acceptable computing times required for a problem solution. Note that the spatial particle distribution generated by the SFEA framework was already investigated in Reference [58]. Interested readers are referred to this publication for a discussion on the performance of this modeling framework to generate randomly distributed particles inside the RVE.

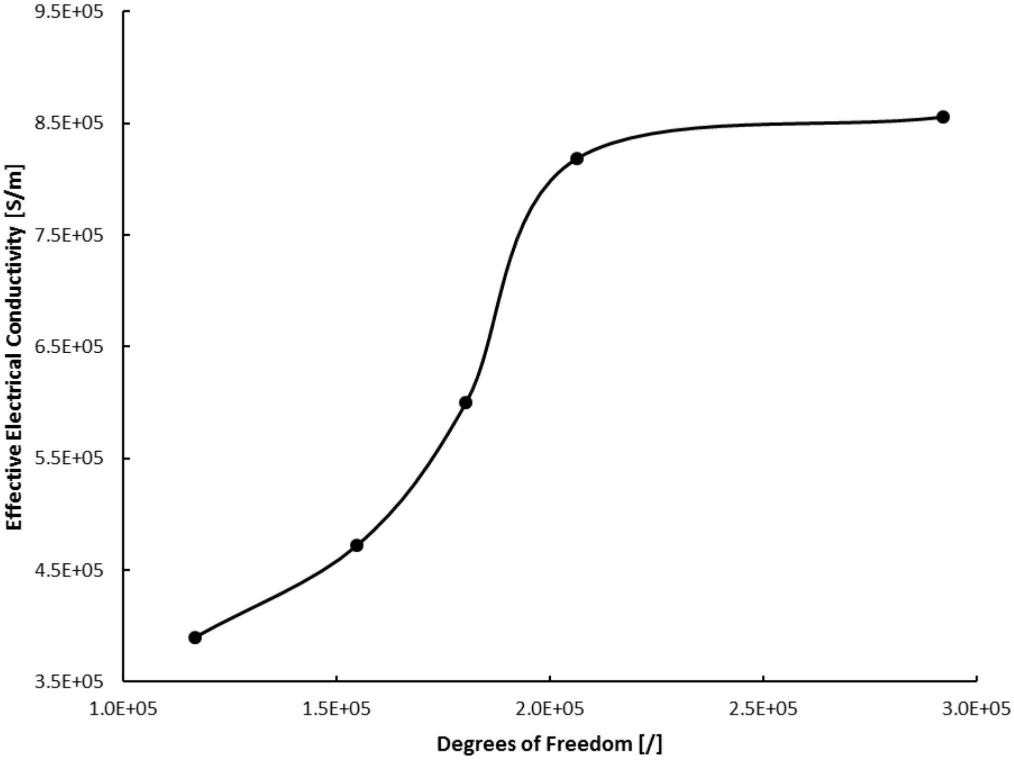


Figure 5-9 Effective electrical conductivity for different model mesh refinements for 21 vol% filler loading.

As mentioned previously, the RVE effective electrical conductivity is computed and stored in each model iteration. This data is used for statistical analyses, such as calculating the unbiased standard deviation and variance for a dataset reflecting the mean effective electrical conductivity for a given material configuration. Abiding by the MC simulation concept, the mean of the effective electrical conductivity results from a set of model iterations was calculated and taken as the final effective electrical conductivity for a given filler volume fraction. For example, Table 5-4 shows the mean effective electrical conductivity and statistical analyses performed for a material system with a filler volume fraction of 30%, filler particle size of 3 nm, and tunneling distance of 1.5 nm. Simulation data are further depicted in Fig. 5-10 in the form of a normalized Probability Distribution

Function (PDF), which suggests that the results are closely normal distributed. Readers are referred to Reference [58] for a discussion on how data calculated by SFEA framework conforms to a normal distribution based on statistical analysis results and acceptance criteria such as mean, median, skewness, and kurtosis values. Note that in the present work, being mindful of required computational resources and solutions times, the number of model iterations was limited to 25 for each material configuration.

Table 5-4 Effective electrical conductivity results and statistical analyses for a silver/epoxy nanocomposite with particle size of 7 nm, tunneling distance of 1.5 nm and 30 vol% filler loading.

Mean value [S·m^{-1}]	855,805
Median value [S·m^{-1}]	849,367
Standard deviation [S·m^{-1}]	321,657
Variance [S·m^{-1}]	1.03E11
Skewness [/]	0.432
Kurtosis [/]	-0.383
95% confidence value [S·m^{-1}]	132,773

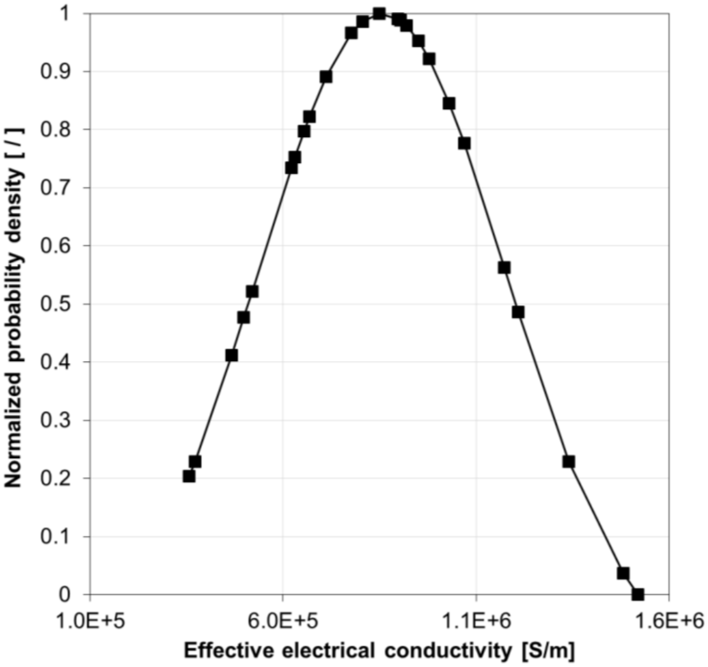


Figure 5-10 A normalized probability density graph for effective electrical conductivity data of a silver/epoxy nanocomposite system with a particle size of 3 nm, tunneling distance of 1.5 nm, and 30 vol% filler loading.

Simulation results can be considered continuous random variables, and therefore, it is possible to calculate the probability of a specific effective electrical conductivity happening within an explicit interval using Eq. (6-8).

$$P(a \leq X \ \leq b) = \int_a^b f(\chi) d\chi \tag{5-8}$$

where P is the probability of an effective electrical conductivity occurring within an interval a and b; $f(\chi)$ and χ are the PDF of the data set and a continuous random variable, respectively. Hence, a Cumulative Distribution Function (CDF) can be computed for each of the various filler volume fractions for a specific material system. For example, Fig. 5-11 depicts the CDF graph for the material system corresponding to Fig. 5-10.

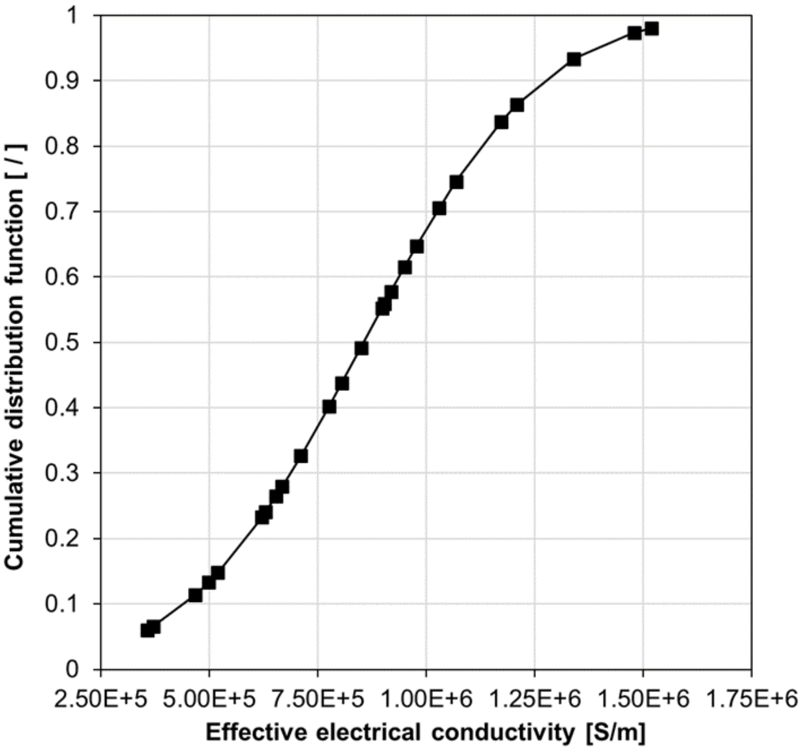


Figure 5-11 The cumulative distribution function for effective electrical conductivity data of a silver/epoxy nanocomposite system with a particle size of 3 nm, tunneling distance of 1.5 nm, and 30 vol% filler loading.

The developed SFEA framework was used for predicting effective electrical conductivities of different material systems, i.e., for particle sizes, D, of 3 nm, 5 nm and 7 nm and tunneling distances, d, of 1 nm and 1.5 nm. The corresponding results are depicted in Fig. 5-12. As mentioned earlier, each predicted data point represents the mean value of 25 model iterations. Fig. 5-12 also shows experimental data, taken from Reference [14], for a specific nano-silver epoxy material before and after thermal treatment. While the experimentally observed percolation threshold was between 5 vol% and 6 vol%, percolation was predicted to occur at higher filler loadings (i.e., >10 vol%) for the simulated material systems. Above percolation, the predicted and experimental data are qualitatively and quantitatively in good agreement, especially for the test data obtained for the nanocomposites after thermal treatment. As mentioned above, a material morphology with clustered silver particles was observed in Reference [14], whereas the present modeling approach generated homogenously distributed and well-dispersed particles, which is likely the cause for the differences in percolation behavior between experiments and modeling results. Nevertheless, despite morphological differences, the SFEA framework was able to simulate material systems that closely mimic actual material behavior.

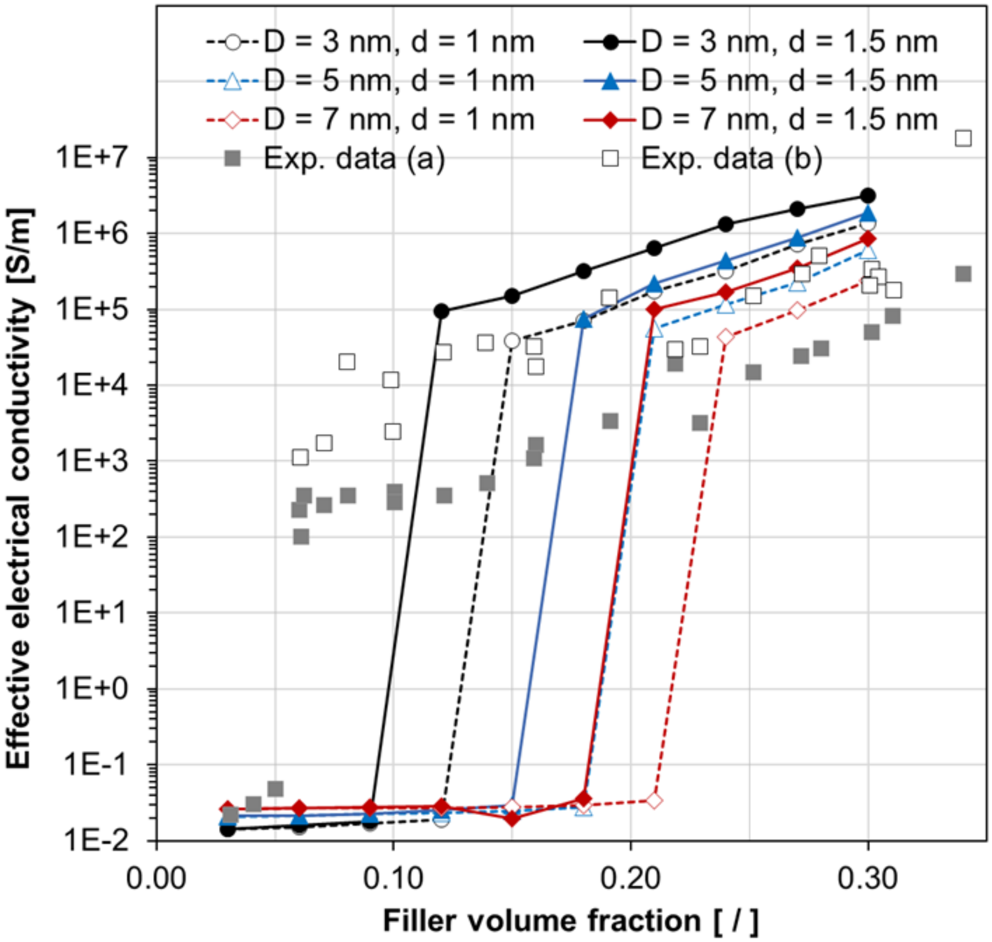


Figure 5-12 Predicted effective electrical conductivity versus filler volume fraction for silver/epoxy nanocomposites with different particle sizes (D) and tunneling distances (d). Experimental (Exp.) data (square symbols) from [14], with (a) and (b) indicating tests before and after thermal treatment, respectively.

Effective electrical conductivity data above percolation were plotted in Fig. 5-13 to explore the influence of the key independent modeling parameters (i.e., filler loading, tunneling distance and particle size). From this graph, it can be inferred that, expectedly, filler loading chiefly influences effective electrical conductivity. For the range of considered particle sizes and tunneling distances, both parameters were found to also have significant influence. Given that the tunneling distance is difficult to quantify compared to particle size and filler loading, and considering its impact on modeling outputs, careful consideration should be given when exploring material designs.

Besides effective electrical conductivity it is also of interest to assess the influence of the key independent modeling parameters on the percolation behavior. An arbitrary yet sensible effective electrical conductivity value of 1.0 S/m was set as the threshold for deciding that electrical conduction through the RVE is established, i.e., electrical percolation is achieved. Furthermore, the probability of reaching the percolation threshold for any given volume fraction was calculated using Eq. (6-9).

$$P_{th}(E) = \frac{A}{N} \tag{5-9}$$

where P_{th} is the probability of reaching the percolation threshold for a given material system; A and N specify the number of model iterations that the effective electrical conductivity was above the threshold value and the total number of iterations, respectively. Corresponding results are summarized in Figs. 5-14 and 5-15. Similar to the electrical conduction behavior, these results indicate the substantial effect that tunneling distance and particle size have on reaching percolation, with particle size being the most significant parameter.

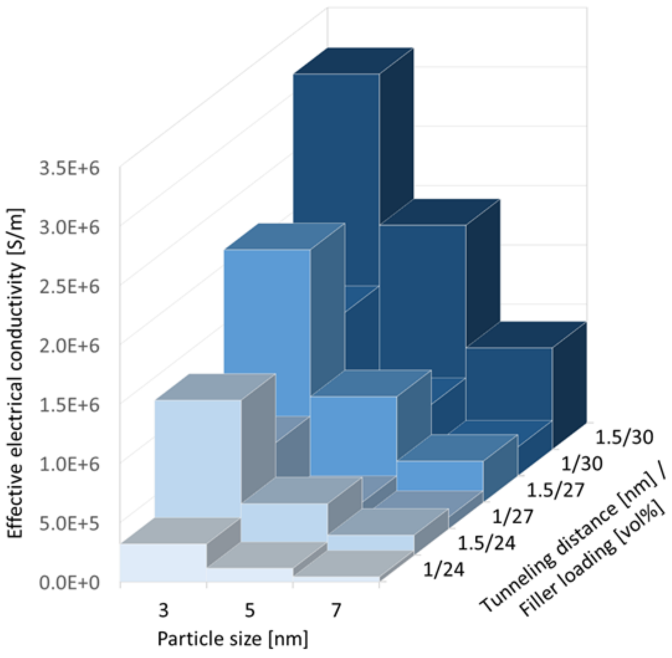


Figure 5-13 The influence of filler loading, tunneling distance, and particle size on the effective electrical conductivity data for silver/epoxy nanocomposites above percolation.

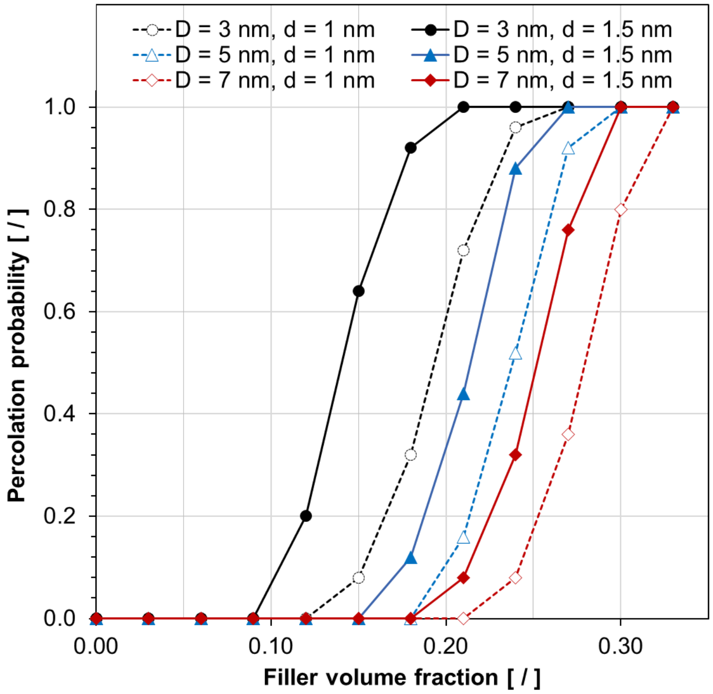


Figure 5-14 The probability of reaching electrical percolation, for an effective electrical conductivity threshold of 1.0 S/m, versus filler volume fraction for silver/epoxy nanocomposites with different particle sizes (D) and tunneling distances (d).

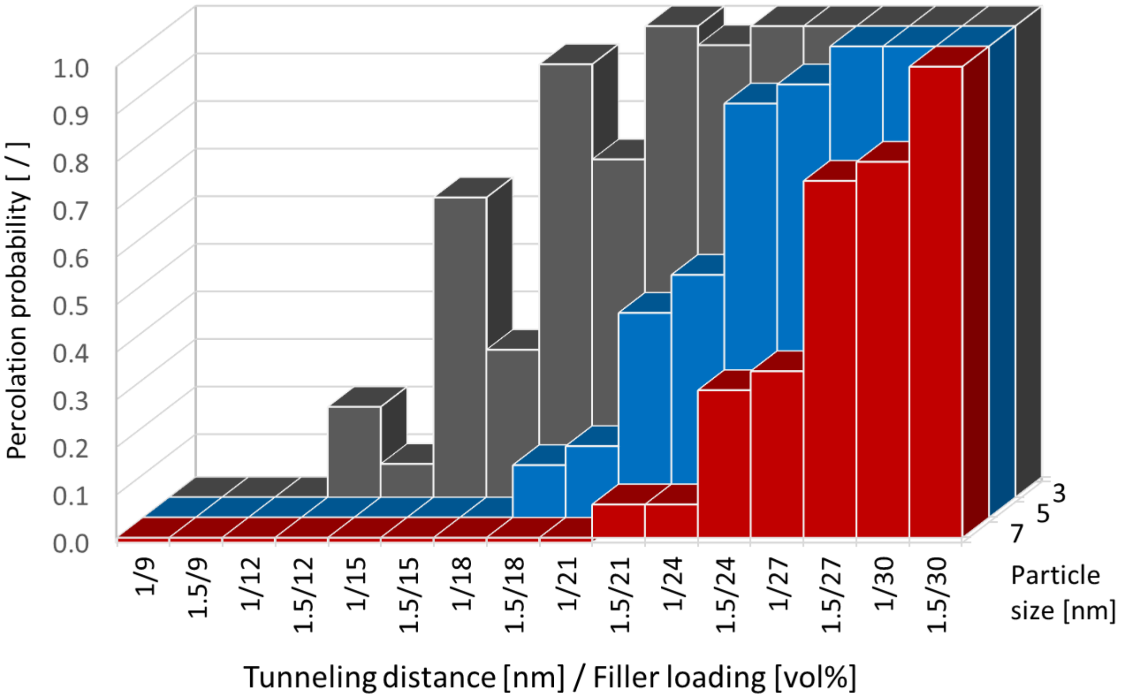


Figure 5-15 The influence of filler loading, tunneling distance and particle size on the probability of for a silver/epoxy nanocomposite reaching electrical percolation, for an effective electrical conductivity threshold of 1.0 S/m.

5.4.2 Piezoresistivity Behavior

A possible application of silver/epoxy nanocomposites are sensors; for example, for measuring deformation. Hence, in the second part of the present research, the SFEA framework was used to investigate the piezoresistivity behavior of these nanocomposites. The piezoresistive behavior of conductive filler modified polymers can be rather complex. For example, the applied tensile strain does not simply cause the distance between filler particles to increase; while filler proximity increases between some particles, it also decreases simultaneously between other particles due to Poisson’s effects in a continuum material, as illustrated in Fig. 5-16.

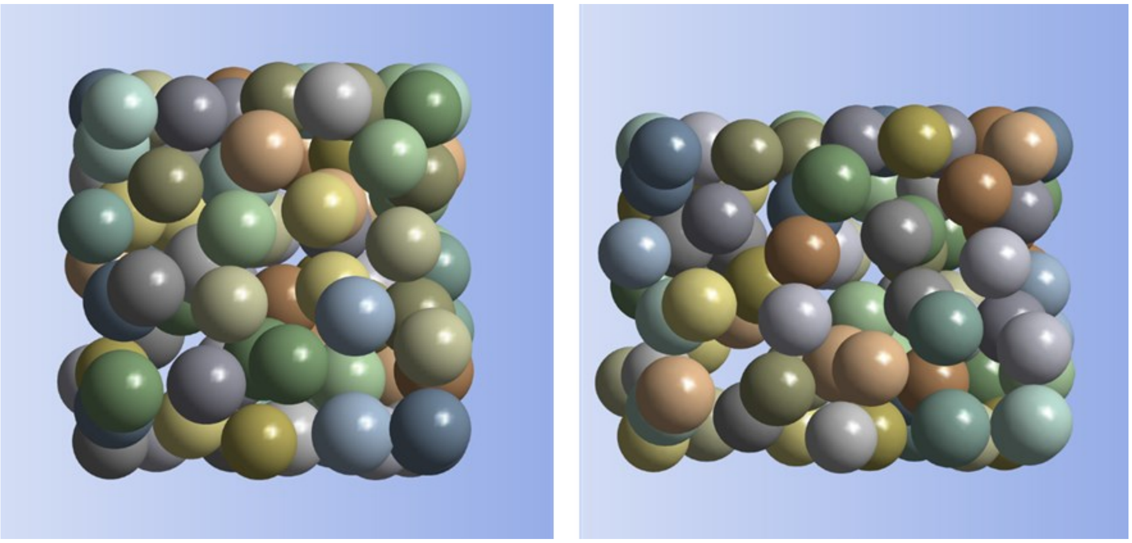


Figure 5-16 An illustration of the applied tensile strain and Poisson’s effects on filler particle proximity: particles in undeformed RVE (left), and particles in deformed RVE (right), for 90,000 microstrain applied to the RVE (horizontal), RVE and particle size of 30 nm and 5 nm, respectively, and filler loading of 30 vol%. For clarity, the matrix material is not shown, and deformations and displacements are depicted with a 150× scaling factor.

An analysis was conducted imposing a mechanical strain of up to 90,000 microstrain upon a material system with a particle size of 3 nm, RVE size of 30 nm, tunneling distance of 1.5 nm, and filler loading of 30 vol%. Electrical resistivity data were computed considering three different Poisson’s ratios for the polymer matrix, i.e., 0.35, 0.4, and 0.45. Corresponding results are shown in Fig. 5-17 along with non-linear Gaussian curve fits. Note that each data point in this figure was computed using a single iteration of the SFEA framework. As shown previously, results for a specific material system are subject to considerable stochastic variation, which becomes rather evident in the plotted datasets. Nevertheless, it can be seen that the material systems are sensitive to applied strain in a non-linear fashion, which is in agreement with previous modeling work on other nanocomposites with conductive platelet fillers [59]. After an initial increase in resistivity by approximately 2.5% at about 20,000 microstrain, a decrease in resistivity by roughly 10% over the initial value was predicted at the maximum applied strain. The data show that the matrix Poisson's ratio has only minor effects over a large portion of the assessed strain range. Presumably, changing particle proximity is the major mechanism that is causing piezoresistivity in conductive spherical filler modified composites. Increasing the average interparticle distance (i.e., the distance between the particle centers) as the result of increasing tensile strain causes an initial increase in electrical resistivity, as shown in Fig. 5-17. However, increasing the tensile strain further also raises the tendency of forming particle clusters that consequently causes a reduction in electrical resistivity. These phenomena were also observed in nano-platelet filler modified polymers [59].

From the presented results, two shortcomings pertaining to a potential sensor material can be noted. First, the sensor response cannot be linked uniquely to a certain strain value as resistivity initially rises before decreasing at higher strain values. Secondly, the considerable stochastic variation in response behavior between different samples of the same material configuration requires careful sensor calibration.

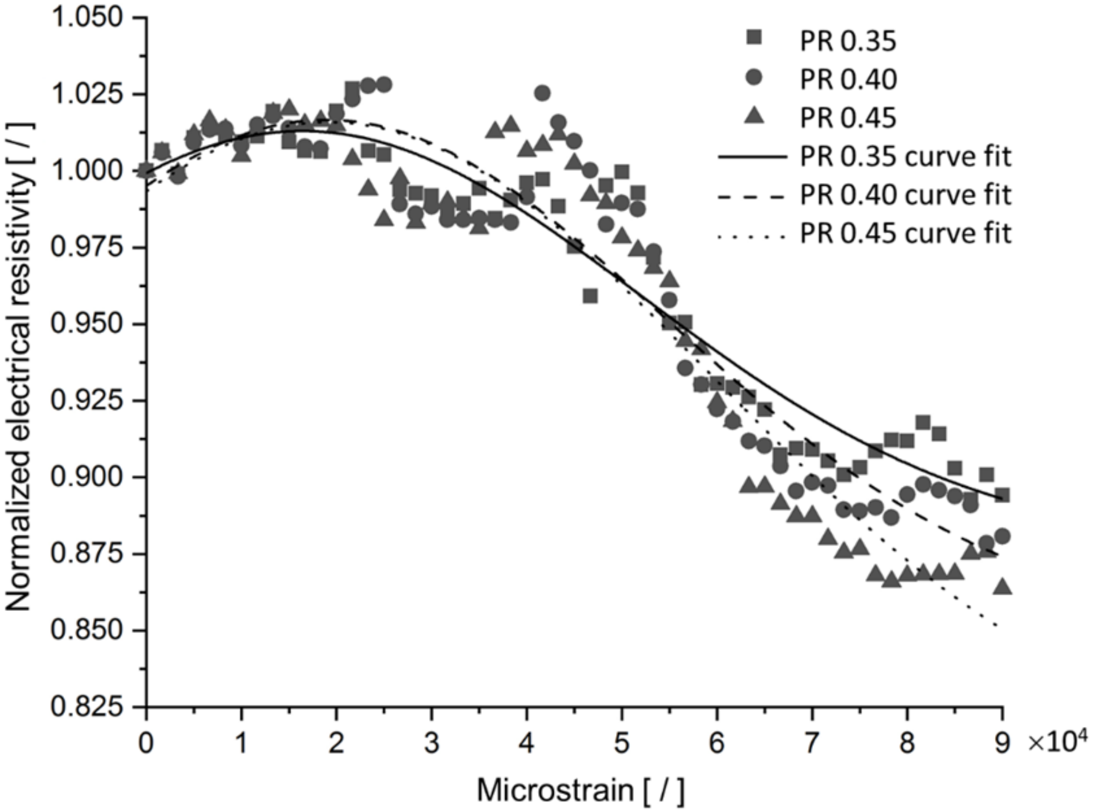


Figure 5-17 Normalized effective electrical resistivity versus applied mechanical strain for silver/epoxy nanocomposites with a particle size of 3 nm, tunneling distance of 1.5 nm, filler loading of 30 vol%, and polymer matrix Poisson's ratio (PR) of 0.35, 0.40, and 0.45.

5.4.3 Temperature Effect on Effective Electrical Conductivity

Assessing the effects of temperature on the effective electrical conductivity is important when considering applications of silver/epoxy nanocomposites. The material response at different temperatures is affected by two physical phenomena: (i) changes in tunneling distance due to thermal expansion effects, and (ii) the effect of temperature on electron activity. With respect to the former phenomenon, a rising temperature changes the material volume, and therefore the distances between particles depending on the thermal-mechanical properties of the matrix and particle filler. The other contributing phenomenon, electron activity changing with temperature, has been described in terms of tunneling current density for intermediate voltages, where $eV \leq \lambda$, as given by

Eq. (6-10) [65].

$$J(V,T) = J(V,T_0)\left\{1 + \left[\frac{3 \times 10^{-9} \times d^2 T^2}{\lambda - \frac{V}{2}}\right]\right\} \tag{5-10}$$

where J, V and T are current density, voltage and temperature in degrees Kelvin, respectively; T_0 represents absolute zero Kelvin (-273.15°C).

Since the developed SFEA framework explicitly considers both the matrix and particulate filler, it enables studying the effects that both phenomena have on a nanocomposite's electrical conductivity. Taking the same material system as in the previous section, a temperature change was applied to the model ranging from ambient 22 °C to 76 °C. Electrical conductivity data were computed, and results are shown in Fig. 5-18 considering only thermal expansion effects, whereas the influence of temperature on electron activity as described by Eq. (6-10) is included in the data shown in Fig. 5-19. As in the previous section, each data point in these figures represents only a single model run in order to demonstrate the extent of stochastic data variation. The data shown in Figs. 5-18 and 5-19 indicate that thermal expansion effects cause a slight reduction in electrical conductivity over the given temperature range (less than 2%). However, when considering electron activity, the model data exhibits a moderate increase in electrical conductivity (by about 7%). Clearly, both thermally-induced phenomena are counteractive, yet, the effect of increasing current density as the result of rising temperature on electrical conductivity is 4.6 times higher than the reduction in electrical conductivity caused by changes in tunneling distance due to thermal expansion effects. Combined, these effects create an approximately linear relation with positive gradient between electrical conductivity and temperature.

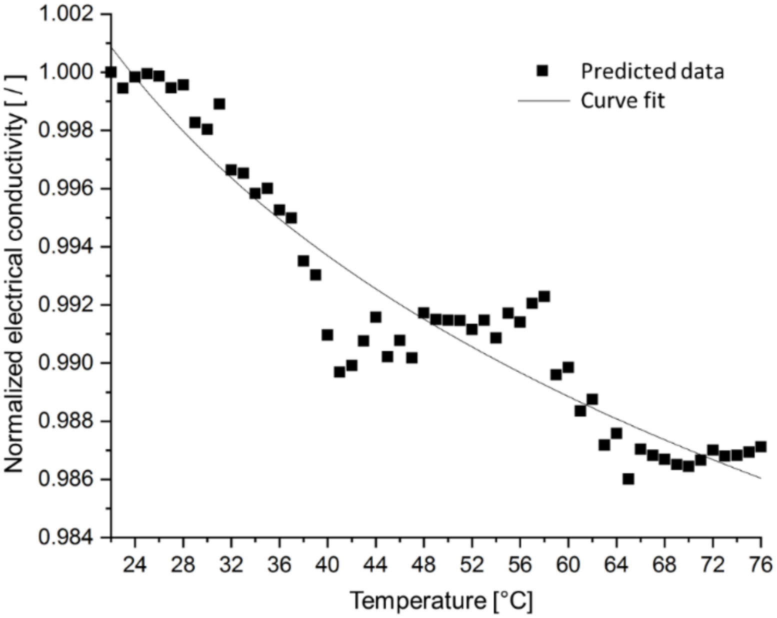


Figure 5-18 Normalized electrical conductivity, considering only thermal expansion effects, versus temperature for silver/epoxy nanocomposites with a particle size of 3 nm, tunneling distance of 1.5 nm, and filler loading of 30 vol%.

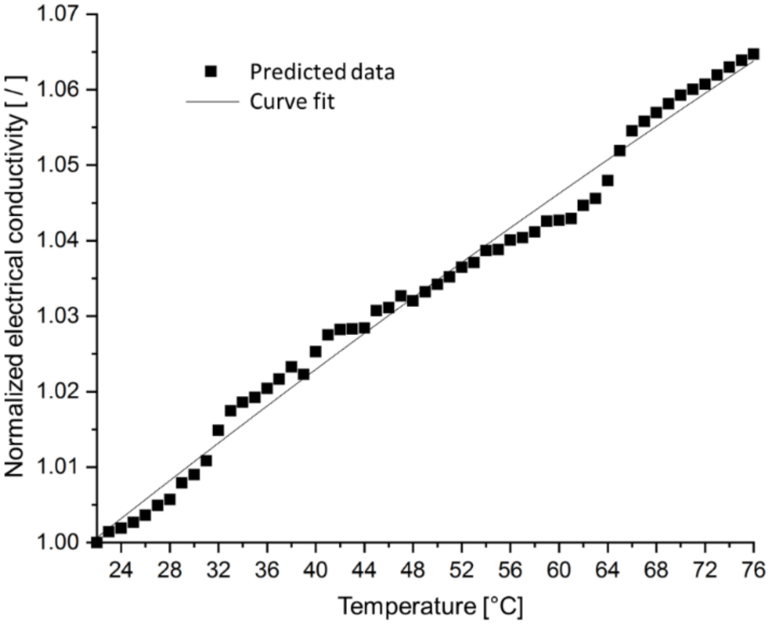


Figure 5-19 Normalized electrical conductivity, considering thermal expansion and electron activity effects, versus temperature for silver/epoxy nanocomposites with a particle size of 3 nm, tunneling distance of 1.5 nm, and filler loading of 30 vol%.

5.5 Conclusions

A stochastic finite element analysis framework was developed that enables predicting the electrical conductivity behavior of polymer composites with electrically conductive fillers. The analysis framework establishes a resistor network that encompasses a continuum representation of both the matrix material and filler particles. As such, the modeling approach enables estimating the composite percolation behavior, and provides a means to simulate piezoresistivity and temperature effects. Due to the parametric nature of the model, the influence of key parameters, such as particle size and tunneling distance, can expediently be explored. The capabilities of the modeling framework were demonstrated considering epoxy nanocomposites reinforced with silver particles. Model outputs were contrasted with available numerical and experimental results, and good qualitative agreement and acceptable quantitative agreement were ascertained. Reasons for quantitative differences are seen in the nanocomposite morphology created by the model, i.e., well-dispersed and homogenously distributed filler particles, which is in contrast with experimental works featuring materials with a typically clustered nanoparticle morphology. Future work will explore the effects of particle clustering, which can be implemented in the model by expanding the particle collision algorithm in the model generation step to not only avoid particle intersection but also enforce particle clustering. The analyses that were performed with the stochastic finite element analysis framework and presented in the present contribution demonstrate the advantages of the developed modeling approach in terms of versatility, time, and cost for exploring different materials systems as compared to experimental campaigns, analytical models, and other numerical techniques.

5.6 References

[1] Beret, S.; Prausnitz, J. Densities of liquid polymers at high pressure. Pressure-volume-temperature measurements for polythylene, polyisobutylene, poly(vinyl acetate), and poly(dimethylsiloxane) to 1 kbar. Macromolecules. 1975, 8, 536–538, DOI: 10.1021/ma60046a028.

[2] Manfredi, L.; Rodriguez, E.; Wladyka-Przybylak, M.; Vazquez, A. Thermal degradation and fire resistance of unsaturated polyester, modified acrylic resins and their composites with natural fibres. Polym. Degrad. Stab. 2006, 91, 255-261, DOI: 10.1016/j.polymdegradstab.2005.05.003.

[3] Drury, C.; Mutsaers, C.; Hart, C.; Matters, M.; Leeuw, D. Low-cost all-polymer integrated circuits. Appl. Phys. Lett. 1998, 73, 108, 1998, DOI: 10.1063/1.121783.

[4] Sheng, N.; Boyce, M.; Parks, D.; Rutledge, G.; Abes, J. Multiscale micromechanical modeling of polymer/clay nanocomposites and the effective clay particle. Polymer. 2004, 45, 487-506, DOI: 10.1016/j.polymer.2003.10.100.

[5] Ghosh, P.; Chakrabarti, A. Conducting carbon black filled EPDM vulcanizates: assessment of dependence of physical and mechanical properties and conducting character on variation of filler loading. Eur. Polym. J. 2000, 36, 1043-1054, DOI: 10.1016/S0014-3057(99)00157-3.

[6] Sohi, N.; Bhadra, S.; Khastgir, D. The effect of different carbon fillers on the electrical conductivity of ethylene vinyl acetate copolymer-based composites and the applicability of different conductivity models. Carbon. 2011, 49, 1349-1361, DOI: 10.1016/j.carbon.2010.12.001.

[7] Natsuki, T.; Tantrakarn, K.; Endo, M. Prediction of elastic properties for single-walled carbon nanotubes. Carbon. 2004, 42, 39–45, DOI: 10.1016/j.carbon.2003.09.011.

[8] Meo, M.; Rossi, M. A molecular-mechanics based finite element model for strength prediction of single wall carbon nanotubes. Mater. Sci. Eng. A. 2007, 454–455, 170–177, DOI: 10.1016/j.msea.2006.11.158.

[9] Tserpes, K.; Papanikos, P.; Tsirkas, S. A progressive fracture model for carbon nanotubes, Composites Part B: Engineering. 2006, 37, 662–669, DOI: 10.1016/j.compositesb.2006.02.024.

[10] Belytschko, T.; Xiao, S.; Schatz, G.; Ruoff, R. Atomistic simulations of nanotube fracture. Physical Review B. 2002, 65, 235430, DOI: 10.1103/PhysRevB.65.235430.

[11] Wu, L.; Tan, S.; Bosman, M.; Yang, J.; Nijhuisbf, C.; Bai, P. Charge transfer plasmon resonances across silver–molecule–silver junctions: estimating the terahertz conductance of molecules at near-infrared frequencies. RSC Adv. 2016, 6, 70884-70894, DOI: 10.1039/C6RA16826D.

[12] Nam, S.; Cho, H.; Kim, T.; Kim, D.; Sung, B.; Lim, S.; Kim, H. Effects of silica particles on the electrical percolation threshold and thermomechanical properties of epoxy/silver nanocomposites. Appl. Phys. Lett. 2011, 99, 043104, DOI: 10.1063/1.3615690.

[13] Untereker, D.; Lyu, S.; Schley, J.; Martinez, G.; Lohstreter, L. Maximum Conductivity of Packed Nanoparticles and Their Polymer Composites. Appl. Mater. Interfaces. 2009, 1, 97–101, DOI: 10.1021/am800038z.

[14] Jiguet, S.; Bertsch, A.; Hofmann, H.; Renaud, P. SU8-Silver Photosensitive Nanocomposites, Adv. Eng. Mater. 2004, 6(9), DOI: 10.1002/adem.200400068.

[15] Jouni, M.; Boudenne, A.; Boiteux, G.; Massardier, V.; Garnier, B.; Serghei, A. Electrical and Thermal Properties of Polyethylene/Silver Nanoparticle Composites. Polym. Compos. 2013, 34, 778-786, DOI: 10.1002/pc.22478.

[16] Feller, J.; Bruzaud, S.; Grohens, Y. Influence of clay nanofiller on electrical and rheological properties of conductive polymer composite. Mater. Lett. 2004, 58, 739-745, DOI: 10.1016/j.matlet.2003.07.010.

[17] Villacorta, B.; Ogale, A.; Hubing, T. Effect of heat treatment of carbon nanofibers on the electromagnetic shielding effectiveness of linear low density polyethylene nanocomposites. Polym. Eng. Sci. 2012, 53, 417, DOI: 10.1002/pen.23276.

[18] Yuan, Q.; Wu, D. Low percolation threshold and high conductivity in carbon black filled polyethylene and polypropylene composites. J. Appl. Polym. Sci. 2009, 115, 369-383, DOI: 10.1002/app.30919.

[19] Mondal, S.; Ganguly, S.; Rahaman, M.; Aldalbahi, A.; Chaki, T.; Khastgir, D.; Das, N. A strategy to achieve enhanced electromagnetic interference shielding at low concentration with a new generation of conductive carbon black in a chlorinated polyethylene elastomeric matrix. Phys. Chem. Chem. Phys. 2016, 18, 24591-24599, DOI: 10.1039/C6CP04274K.

[20] Nisar, M.; Bergmann, C.; Geshev, J.; Quijada, R.; Maraschin, T.; Basso, N.; Barrera, E.; Galland, G. Synthesis of high-density polyethylene/rGO-CNT-Fe nanocomposites with outstanding magnetic and electrical properties. J. Appl. Polym. Sci. 2017, 134, 45382, DOI: 10.1002/app.45382.

[21] Villacorta, B.; Ogale, A. Morphological influence of carbon modifiers on the electromagnetic shielding of their linear low density polyethylene composites. J. Appl. Polym. Sci. 2014, 131, 41055, DOI: 10.1002/app.41055.

[22] Ma, P.; Siddiqui, N.; Marom, G.; Kim, J. Dispersion and functionalization of carbon nanotubes for polymer-based nanocomposites: A review. Composites Part A: Applied Science and Manufacturing. 2010, 41, 1345-1367, DOI: 10.1016/j.compositesa.2010.07.003.

[23] Akcora, P.; Liu, H.; Kumar, S.; Moll, J.; Brian, Y.; Benicewicz, C.; Schadler, L.; Acehan, D.; Panagiotopoulos, A.; Pryamitsyn, V.; Ganesan, V.; Ilavsky, J.; Thiyagarajan, P.; Colby, R.; Douglas, J. Anisotropic self-assembly of spherical polymer-grafted nanoparticles. Nat. Mater. 2009, 8, 354-359, DOI: 10.1038/NMAT2404.

[24] Li, Z.; Okamoto, K.; Ohki, Y.; Tanaka, T. Effects of nano-filler addition on partial discharge resistance and dielectric breakdown strength of Micro-Al2O3Epoxy composite. IEEE Trans. Dielectr. Electr. Insul. 2010, 17, 653-661, DOI: 10.1109/TDEI.2010.5492235.

[25] Seidel, G.; Puydupin-Jamin, A. Analysis of clustering, interphase region, and orientation effects on the electrical conductivity of carbon nanotube–polymer nanocomposites via computational micromechanics. Mech. Mater. 2011, 43, 755–774, DOI: 10.1016/j.mechmat.2011.08.010.

[26] Natsuki, T.; Endo, M.; Takahashi, T. Percolation study of orientated short-fiber composites by a continuum model. Physica A. 2005, 352, 498–508, DOI: 10.1016/j.physa.2004.12.059.

[27] Ma, H.; Gao, X. A three-dimensional Monte Carlo model for electrically conductive polymer matrix composites filled with curved fibers. Polymer. 2008, 49, 4230–4238, DOI: 10.1016/j.polymer.2008.07.034.

[28] Bao, W.; Meguid, S.; Zhu, Z.; Pan, Y.; Weng, G. A novel approach to predict the electrical conductivity of multifunctional nanocomposites. Mech. Mater. 2012, 46, 129–138, DOI: 10.1016/j.mechmat.2011.12.006.

[29] Wang, W.; Jayatissa, A. Computational and experimental study of electrical conductivity of graphene/poly(methyl methacrylate) nanocomposite using Monte Carlo method and percolation theory. Synth. Met. 2015, 204, 141–147, DOI: 10.1016/j.synthmet.2015.03.022.

[30] Li, C.; Chou, T. Multiscale modeling of compressive behavior of carbon nanotube polymer composites. Compos. Sci. Technol. 2006, 66, 2409–2414, DOI: 10.1016/j.compscitech.2006.01.013.

[31] Xiao, J.; Gillespie, J. Nanomechanics of single-walled carbon nanotubes as composite reinforcement. Polym. Eng. Sci. 2006, 46, 1051–1059, DOI: 10.1002/pen.20553.

[32] Shi, D.; Feng, X.; Huang, Y.; Hwang, K.; Gao, H. The effect of nanotube waviness and agglomeration on the elastic property of carbon nanotube-reinforced composites. J. Eng. Mater. Technol. 2004, 126, 250–257, DOI: 10.1115/1.1751182.

[33] Song, Y.; Youn, J. Evaluation of effective thermal conductivity for carbon nanotube/polymer composites using control volume finite element method. Carbon. 2006, 44, 710–717, DOI: 10.1016/j.carbon.2005.09.034.

[34] Xue, Q. Model for thermal conductivity of carbon nanotube-based composites. Phys. Rev. B. 2005, 368, 302–307, DOI:10.1016/j.physb.2005.07.024.

[35] Bagchi, A.; Nomura, S. On the effective thermal conductivity of carbon nanotube reinforced polymer composites. Compos. Sci. Technol. 2006, 66, 1703–1712, DOI: 10.1016/j.compscitech.2005.11.003.

[36] Zhang, J.; Tanaka, M. Systematic study of thermal properties of CNT composites by the fast multipole hybrid boundary node method. Eng. Anal. Bound. Elem. 2007, 31, 388–401, DOI: 10.1016/j.enganabound.2006.07.011.

[37] Pike, G.; Seager, C. Percolation and conductivity: a computer study I. Phys. Rev. B. 1974, 10, 1421–1434, DOI: 10.1103/PhysRevB.10.1421.

[38] Kirkpatrick, S. Percolation and conduction. Rev. Mod. Phys. 1973, 45, 574–588, DOI: 10.1103/ RevModPhys.45.574.

[39] Behnam, A.; Ural, A. Computational study of geometry-dependent resistivity scaling in single-walled carbon nanotube films. Phys. 2007, 75, 125432, DOI: 10.1103/PhysRevB.75.125432.

[40] Dalmas, F.; Dendievel, R.; Chazeau, L.; Cavaillé, J.; Gauthier, C. Carbon nanotube-filled polymer composites. Numerical simulation of electrical conductivity in three-dimensional entangled fibrous networks. Acta Mater. 2006, 54, 2923–2931, DOI: 10.1016/j.actamat.2006.02.028.

[41] Hakobyan, Y.; Papoulia, K.; Grigoriu, M. Physical and geometrical percolations of effective conductivity on a lattice. Phys, Rev. B. 2007, 76, 144205, DOI: 10.1103/PhysRevB.76.144205.

[42] Foygel, M.; Morris, R.; Anex, D.; French, S.; Sobolev, V. Theoretical and computational studies of carbon nanotube composites and suspensions: electrical and thermal conductivity. Phys, Rev. B. 2005, 71, 104201, DOI: 10.1103/PhysRevB.71.104201.

[43] Hu, N.; Masuda, Z.; Yan, C.; Yamamoto, G.; Fukunaga, H.; Hashida, T. The electrical properties of polymer nanocomposites with carbon nanotube fillers. Nanotechnology. 2008, 19, 215701, DOI: 10.1088/0957-4484/19/21/215701.

[44] Jack, D.; Yeh, C.; Liang, Z.; Li, S.; Park, J.; Fielding, J. Electrical conductivity modeling and experimental study of densely packed SWCNT networks. Nanotechnology. 2010, 21, 195703, DOI: 10.1088/0957-4484/21/19/195703.

[45] Li, C.; Chou, T. Electrical conductivities of composites with aligned carbon nanotubes. J. Nanosci. Nanotechnol. 2009, 9, 2518–2524, DOI: 10.1166/jnn.2009.456.

[46] Li, C.; Chou, T. A direct electrifying algorithm for backbone identification. J. Phys. A Math. Gen. 2007, 40, 679-686, DOI: 10.1088/1751-8113/40/49/004.

[47] Sun, X. Highly conductive carbon nanotube/polymer nanocomposites achievable? Macromol. Theor. Simul. 2009, 18, 155–161, DOI: 10.1002/mats.200800091.

[48] Topinka, M.; Rowell, M.; Goldhaber-Gordon, D.; McGehee, M.; Hecht, DS.; Gruner, G. Charge transport in interpenetrating networks of semiconducting and metallic carbon nanotubes. Nano Lett. 2009, 9, 1866–1871, DOI: 10.1021/nl803849e.

[49] Li, C.; Thostenson, E.; Chou, T. Dominant role of tunneling resistance in the electrical conductivity of carbon nanotube-based composites. Appl. Phys. Lett. 2007, 91, 223114, DOI: 10.1063/1.2819690.

[50] Seidel, G.; Lagoudas, D. A micromechanics model for the electrical conductivity of nanotube–polymer nanocomposites. J. Compos. Mater. 2009, 43, 917–941, DOI: 10.1177/ 0021998308105124.51.

[51] Postma, H.; Jonge, M.; Yao, Z.; Dekker, C. Electrical transport through carbon nanotube junctions created by mechanical manipulation. Phys. Rev. B. 2000, 62, R10 653–R10 656, DOI: 10.1103/PhysRevB.62.R10653.

[52] Choi, E.; Brooks, J.; Eaton, D. Enhancement of thermal and electrical properties of carbon nanotube polymer composites by magnetic field processing. J. Appl. Phys. 2003, 94, 6034, DOI: 10.1063/ 1.1616638.

[53] Sandler, J.; Kirk, J.; Kinloch, I.; Shaffer, MSP.; Windle, A. Ultra-low electrical percolation threshold in carbon-nanotube epoxy composites. Polymer. 2003, 44, 5893–5899, DOI: 10.1016/S0032-3861(03)00539-1.

[54] Zhang, W.; Dehghani-Sanij, A.; Blackburn, R. Carbon based conductive polymer composites. J. Mater. Sci. 2007, 42, 3408-3418, DOI: 10.1007/s10853-007-1688-5.

[55] Cai, W.; Tu, ST.; Gong, J. A physically based percolation model of the effective electrical conductivity of particle filled composites. J. Compos. Mater. 2006, 40, 2131–2142, DOI: 10.1177/0021998306062312.

[56] Hindermann-Bischoff, M.; Ehrburger-Dolle, F. Electrical conductivity of carbon black–polyethylene composites: Experimental evidence of the change of cluster connectivity in the PTC effect. Carbon. 2001, 39, 375-382, DOI: 10.1016/S0008-6223(00)00130-5.

[57] Tang, H.; Chen, X.; Luo, Y. Studies on the PTC/NTC effect of carbon black filled low density polyethylene composites. Eur. Polym. J. 1997, 33, 1383-1386, DOI: 10.1016/S0014-3057(96)00221-2.

[58] Ahmadi Moghaddam, H.; Mertiny, P. Stochastic finite element analysis framework for modelling thermal conductivity of particulate modified polymer composites. Results Phys. 2018, 11, 905-914, 2018, DOI: 10.1016/j.rinp.2018.10.045.

[59] Oskouyi, A.; Sundararaj, U.; Mertiny, P. Tunneling conductivity and Piezoresistivity of composites containing randomly dispersed conductive nano-platelets. Materials. 2014, 7, 2501-2521, DOI: 10.3390/ma7042501.

[60] Hicks, J.; Behnam, A.; Ural, A. A computational study of tunneling-percolation electrical transport in graphene-based nanocomposites. Appl. Phys. Lett. 2009, 95, 213103, DOI: 10.1063/1.3267079.

[61] Luheng, W.; Tianhuai, D.; Peng, W. Influence of carbon black concentration on Piezoresistivity for carbon black-filled silicone rubber composite. Carbon. 2009, 47, 3151–3157, DOI: 10.1016/j.carbon.2009.06.050.

[62] Hu, N.; Karube, Y.; Yan, C.; Masuda, Z.; Fukunaga, H. Tunneling effect in a polymer/carbon nanotube nanocomposite strain sensor. Acta Mater. 2008, 56, 2929–2936, DOI: 10.1016/j.actamat.2008.02.030.

[63] Pham, G. Characterization and Modeling of Piezo-Resistive Properties of Carbon Nanotube-Based Conductive Polymer Composites. Ph.D. Thesis, Florida State University, Tallahassee, FL, USA, November 2008.

[64] Simmons, J. G. Generalized Formula for the Electric Tunnel Effect Between Similar Electrodes Separated by a Thin Insulating Film. J. Appl. Phys. 1963, 34, 1793–1803, DOI: 10.1063/1.1702682.

[65] Simmons, J. G., Generalized Thermal J-V Characteristic for the Electric Tunnel Effect, J. Appl. Phys. 1964, 35, 2655–2658, DOI: 10.1063/1.1713820.

Chapter 6
Conclusions

6.1 Conclusions and Discussions

In this doctoral research work, a stochastic finite element analysis (SFEA) framework was developed for predicting mechanical, thermal and electrical properties of randomly distributed and dispersed particulate polymer composites. Developing this framework was motivated by the lack of analytical methods capable of predicting properties of filler modified polymers accurately. While several of the latter are available, they are limited in terms of relevant input parameters for predicting material properties of particulate polymer composites. Besides analytical methods, experimental methods can be used for exploring material properties. However, performing experiments is typically time-consuming and associated with considerable costs. This framework is aimed at alleviating the shortcomings related to analytical and experimental methods.

A review of technical literature published in the context of analytical methods for predicting mechanical properties of filler modified polymers [1-13] (i.e., effective modulus of elasticity, Poisson's ratio and coefficient of thermal expansion) reveals that typically only filler volume fraction is considered for predicting mechanical properties while the effects of particles size distribution and composite morphology are usually neglected. Not including the latter effects can lead to a reduction in the accuracy of predicted material properties. Regarding analytical methods for predicting thermal properties of filler modified polymers (i.e., effective thermal conductivity), with the exception of work presented in [14] that considers "limited" particle size effects, other works perform predictions solely based on filler content [15-20].

Besides analytical models, numerical modeling approaches have been presented for the computation of effective material properties, yet, these approaches typically employ a homogenized morphology and effective material properties that are derived from analytical models [21-22]. These models are subject to similar problems as analytical models and are therefore limited in their ability to predict effective material properties for wide range of filler loadings accurately. In the context of predicting electrical properties of filler modified polymers (i.e., effective electrical conductivity, percolation threshold and piezoresistivity properties), developed numerical methods [23-35], while being able to consider to particle size effects, are often not continuum models (i.e., the matrix is not explicitly included in the model), which reduces the accuracy of electrical properties predictions.

In contrast, the developed SFEA framework constitutes a continuum modeling approach for predicting a broad range of interacting material properties (mechanical, thermal and electrical), and provides a means for applying true randomness to all aspects of the model generation, including particle size distribution and composite morphology (i.e., particle orientation, distribution and dispersion). These aspects underline the effectiveness and novelty the SFEA framework.

The presented work involving the SFEA framework focused largely on spherical particle shapes for demonstrating the concept. However, other particle shapes can be modeled as well, such as disk, ellipsoidal and cylindrical shapes particles which is demonstrated in one part of this study, in order to explore particle alignment effect on polymer composite properties. As mentioned above, this framework enables exploring the effects of a multitude of input parameters, including particle size distribution, particle and matrix material properties, and meshing characteristics, while permitting statistical treatment of results related to mechanical, thermal and electrical properties. Consequently, this framework is not only an effective tool for exploring changes in material properties caused by filler addition, which is similar to an experimental campaign; more importantly, the framework permits comprehensively exploring properties of materials that have not been realized, or are difficult to fabricate, through experimental means, such as certain hybrid materials or composites with highly aligned fillers (as shown in Chapter 3).

At this juncture it should be mentioned that increasing filler loading in the model greatly impacts the computational effort. In fact, increasing filler volume percentage in the RVE increases the model solution time exponentially. Hence, while opportunities exist to model a wide range of composites with high complexity, the SFEA framework may face limitations in terms of computational resources. Nevertheless, ever increasing speed and capacity of computer equipment bode well for the developed modeling approach to enable substantial future insights.

In summary, this doctoral research included the following parts. Initially, a semi-automated SFEA framework was created as an exploratory tool (Section 2.1), forming the foundation and proving the concept for predicting the (mechanical) properties of randomly distributed (spherical) particles embedded in a polymer matrix. The effective properties (modulus of elasticity and Poisson's ratio) were computed, and the results were contrasted against analytical and experimental works. Good agreement between the numerical result and the experimental results published in the literature was ascertained. In a subsequent step (Section 2.2), the SFEA framework concept was expanded to enable the comprehensive study of mechanical properties. Besides modulus of elasticity and Poisson's ratio also the coefficient of thermal expansion of randomly distributed spherical shape particulate polymer composites was predicted. It was confirmed that the stochastic simulation enabled applying true randomness in the model and consequently the prediction of mechanical properties.

The third part of this research (Chapter 3) focused on predicting the effective thermal conductivity of particulate polymer composites by simulating multiple heat transfer mechanisms, i.e., direct particle-to-particle heat transfer as well as defining interface thermal (Kapitza) resistance between particles and the polymer matrix. This modelling approach enabled increasing the fidelity and thus possibly accuracy of thermal property predictions in comparison with other available methods. An expansion of this work (Chapter 4) involved employing the SFEA framework to predict effective thermal conductivity while exploring particle alignment effects, for the case of randomly distributed but aligned cylindrical shape particles. The numerical results demonstrated the SFEA framework's capabilities in capturing alignment effects.

In the final part of this research (Chapter 5), the capabilities of the SFEA framework were further expanded for predicting electrical properties (i.e., effective electrical conductivity, electrical percolation threshold and electrical piezoresistivity effects). In light of the statistical nature of these properties and to accomplish the latter, the framework's ability to consider changes in electrical conductivity as well as strain in response to mechanical deformation and temperature was instrumental. These effects were explored for the case of polymer composites with randomly distributed spherical shaped particles. It should be emphasized that this work constitutes a significant advancement over past and contemporary modeling approaches presented in the technical literature. Most research work done in this field neglects the presence of the matrix material in numerical models, employing instead a network of resistors that represent particles and particle interactions.

In terms of implementation, the SFEA framework incorporates different programming languages. The Visual Basic for Applications programming language was employed to execute the different modules associated with the framework, as well as to create a database for storing input and output parameters. The general mathematical programming environment MATLAB was used for generating random numbers required for the random variables. The IronPython programming language was used to automate the processes in the ANSYS Workbench FEA environment (i.e., reading numerical input parameters and running ANSYS DesignModeler and ANSYS Mechanical). Finally, JavaScript programming enabled automating the process of creating the model geometry using ANSYS DesignModeler, and performing numerical analyses using ANSYS Mechanical, and extracting the numerical results. The various software codes were developed in such a manner as to allow for use with any type of ANSYS license (e.g., commercial or academic licenses).

In closing, this doctoral work proved the stated research hypothesis (i.e., an effective numerical modeling framework can be created that facilitates material design for the large design space that is associated with the wide variety of particulate polymer composites). Moreover, in the process of this doctoral work, all of the stated research objectives were achieved, that is, the development of a SFEA framework for comprehensively predicting the properties of particulate polymer composites, including mechanical properties (modulus of elasticity, Poisson’s ratio, coefficient of thermal expansion), thermal properties, (effective thermal conductivity), electrical properties (effective electrical conductivity, percolation threshold, piezoresistivity effects due to mechanical and thermal deformation).

6.2 Recommendations for Future Work

As mentioned in the introductory parts of this thesis, adding suitable fillers to polymers may improve their mechanical, thermal and electrical properties, yet, considering the lack of comprehensive analytical methods for predicting property changes, designing new material systems is a challenging, time-consuming and thus expensive task for material designers. Therefore, important contributions were made by this doctoral research in the area of particulate polymer composite material characterization, and novel approaches for predicting mechanical, thermal and electrical properties were developed. Nevertheless, work on the stochastic finite element analysis framework can be extended to explore opportunities to further improve results accuracy and expand modeling capabilities. The following recommendations are therefore given.

- The approach developed in this doctoral work for prediction of effective thermal conductivity of particulate polymer composite considers interfacial thermal resistance between particles and polymer matrix as well as a direct heat transfer between particles. However, perfect heat transfer was assumed for the latter, given the complexities related to direct particle-to-particle heat transfer. However, it is known that resistance to heat transfer between particles exists, which depends on complicated conditions such as particle morphology. In this study, the number of direct particle-to-particle contacts is considered small, and hence, results were calculated by tuning the particle to matrix interfacial thermal resistance values for an explicit volume fraction. It is of interest to investigate the effects of this assumption by considering a separate thermal resistance parameter for direct particle-to-particle heat transfer.
- In this study, a fully bonded condition was assumed for defining the contact between filler particles and the surrounding matrix since only a small displacements applied. However, it would be of interest to include other types of modeling interfacial contact for predicting the various materials properties, e.g. contacts with a debonding capability, since such an approach may permit capturing damage effects in case of applying higher mechanical strains (i.e. detachment and relative displacement between filler particles and matrix).
- Recall that the calculated effective electrical conductivity and percolation threshold values in Chapter 5 agreed only marginally with experimental results from the technical literature. The reason for the discrepancies is seen in the difference between the morphology of the numerical model, i.e., well distributed and dispersed filler particles, and the clustered

morphology of composites in the experimental work. Hence, developing a new particle collision detection algorithm that allows for some extent of particle interference and/or attraction of particles may enable capturing clustering effects in particulate polymer composites and thus improve the predictions quantitatively.

6.3 References

[1] Prasad, K.; Das, B.; Maitra, U.; Ramamurty, U.; Rao, C. Extraordinary synergy in the mechanical properties of polymer matrix composites reinforced with 2 nanocarbons. PNAS. 2009, 106, 13186-13189, DOI: 10.1073/pnas.0905844106.

[2] Bewlay, B.; Jackson, M.; Lipsitt, H. The balance of mechanical and environmental properties of a multielement niobium-niobium silicide-basedIn Situ composite. Metall. Mater. Trans. A. 1996, 27, 3801–3808, DOI: 10.1007/BF02595629.

[3] Rana, D.; Lee, C.; Cho, K.; Lee, B.; Choe, S. Thermal and mechanical properties for binary blends of metallocene polyethylene with conventional polyolefins. J. Appl. Polym. Sci. 1998, 69, 2441–2450, DOI: 10.1002/(SICI)1097-4628(19980919)69:12<2441::AID-APP15>3.0.CO;2-%23.

[4] Sathe, S.; Devi, S.; Srinivasa, G.; Rao, K. Relationship between morphology and mechanical properties of binary and compatibilized ternary blends of polypropylene and nylon 6. J. Appl. Polym. Sci. 1996, 61, 97–107, DOI: 10.1002/(SICI)1097-4628(19960705)61:1<97::AID-APP11>3.0.CO;2-X.

[5] Eshelby, J. The determination of the elastic field of an ellipsoidal inclusion and related problems. Proc. R. Soc. London, Ser. A. 1957, 241, 561–569, DOI: 10.1098/rspa.1957.0133.

[6] Voigt, W. Ueber die Beziehung zwischen den beiden Elasticitätsconstanten isotroper Körper, Annalen der Physik, 1889, 274, 573–587, DOI: 10.1002/andp.18892741206.

[7] Halpin, J.; Kardos, J. The Halpin-Tsai equations: A review. Polym. Eng. Sci. 1976, 16, 344–352, DOI: 10.1002/pen.760160512.

[8] Hashin, Z.; Shtrikman, S. A variational approach to the theory of elastic behavior of multiphase materials. J. Mech. Phys. Solids. 1963, 11, 127–140, DOI: 10.1016/0022-5096(63)90060-7.

[9] Mori, T.; Tanaka, K. Average stress in the matrix and average elastic energy of materials with misfitting inclusions. Acta Metall. 1973, 21, 571–574, DOI: 10.1016/0001-6160(73)90064-3.

[10] Benveniste, Y. A new approach to the application of Mori–Tanaka's theory in composite materials. Mech. Mater. 1987, 6, 147–157, DOI: 10.1016/0167-6636(87)90005-6.

[11] Christensen, R.; Lo, K. Solutions for effective shear properties of three phase sphere and cylinder models. J. Mech. Phys. Solids. 1987, 27, 315–330, DOI: 10.1016/0022-5096(79)90032-2.

[12] Hill, R. A self-consistent mechanics of composite materials. J. Mech. Phys. Solids. 1965, 13, 213–222, DOI: 10.1016/0022-5096(65)90010-4.

[13] Torquato, S. Effective stiffness tensor of composite media: II. Applications to isotropic dispersions. J. Mech. Phys. Solids. 1998, 46, 1411–1440, DOI: 10.1016/S0022-5096(97)00083-5.

[14] Bohm, H.; Nogales, S. Mori-Tanaka Model for the thermal conductivity of composites with interfacial resistance and particle size distributions. Compos. Sci. Technol. 2008, 68, 1181-1187, DOI: 10.1016/j.compscitech.2007.06.009.

[15] Maxwell, J. A treatise on electricity and magnetism, volume I, 3rd Edition, Oxford University Press, 1904.

[16] Bird R.; Stewart W.; Lightfoot, E. Transport phenomena,John Wiley & Sons, 2007, ISBN: 0-470-11539-4.

[17] Eucken A. Die Wärmeleitfähigkeit Keramischer, Fester Stoffe– Ihre Berechnung aus der Wärmeleitfähigkeit der Bestandteile, VDI Forschungsheft 353, Beilage zu, Forschung auf dem Ggebiete des Ingenieurwesens, Ausgabe B, Band 3, 1932.

[18] Burger, H. Das lertvermogen verdumter mischkristallfreierlonsungen, Phys. Zs. 20, 1915.

[19] Hamilton, R.; Crosser, O. Thermal conductivity of heterogeneous two component systems, Ind. Eng. Chem. Fundam. 1962, 1, 187-191, DOI: 10.1021/i160003a005.

[20] Lord Rayleigh Sec. R.S. LVI. On the influence of obstacles arrangedin rectangular order upon the properties of a medium. The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science. 1892, 34, 481-502, DOI: 10.1080/14786449208620364.

[21] Soltannia, B.; Haji Gholami, I.; Masajedian, S.; Mertiny, P.; Sameoto, D.; Taheri, F. Parametric Study of Strain Rate Effects on Nanoparticle-Reinforced Polymer Composites. J. Nanomater. 2016, 1–9, DOI: 10.1155/2016/9841972.

[22] Soltannia, B.; Mertiny, P.; Taheri, F. Static and dynamic characteristics of nano-reinforced 3D-fiber metal laminates using non-destructive techniques. Journal of Sandwich Structures & Materials. 2020, DOI: 10.1177/1099636220924585.

[23] Pike, G.; Seager, C. Percolation and conductivity: a computer study I. Phys. Rev. B. 1974, 10, 1421–1434, DOI: 10.1103/PhysRevB.10.1421.

[24] Kirkpatrick, S. Percolation and conduction. Rev. Mod. Phys. 1973, 45, 574–588, DOI: 10.1103/ RevModPhys.45.574.

[25] Behnam, A.; Ural, A. Computational study of geometry-dependent resistivity scaling in single-walled carbon nanotube films. Phys. 2007, 75, 125432, DOI: 10.1103/PhysRevB.75.125432.

[26] Dalmas, F.; Dendievel, R.; Chazeau, L.; Cavaillé, J.; Gauthier, C. Carbon nanotube-filled polymer composites. Numerical simulation of electrical conductivity in three-dimensional

entangled fibrous networks. Acta Mater. 2006, 54, 2923–2931, DOI: 10.1016/j.actamat.2006.02.028.

[27] Hakobyan, Y.; Papoulia, K.; Grigoriu, M. Physical and geometrical percolations of effective conductivity on a lattice. Phys, Rev. B. 2007, 76, 144205, DOI: 10.1103/PhysRevB.76.144205.

[28] Foygel, M.; Morris, R.; Anex, D.; French, S.; Sobolev, V. Theoretical and computational studies of carbon nanotube composites and suspensions: electrical and thermal conductivity. Phys, Rev. B. 2005, 71, 104201, DOI: 10.1103/PhysRevB.71.104201.

[29] Hu, N.; Masuda, Z.; Yan, C.; Yamamoto, G.; Fukunaga, H.; Hashida, T. The electrical properties of polymer nanocomposites with carbon nanotube fillers. Nanotechnology. 2008, 19, 215701, DOI: 10.1088/0957-4484/19/21/215701.

[30] Jack, D.; Yeh, C.; Liang, Z.; Li, S.; Park, J.; Fielding, J. Electrical conductivity modeling and experimental study of densely packed SWCNT networks. Nanotechnology. 2010, 21, 195703, DOI: 10.1088/0957-4484/21/19/195703.

[31] Li, C.; Chou, T. Electrical conductivities of composites with aligned carbon nanotubes. J. Nanosci. Nanotechnol. 2009, 9, 2518–2524, DOI: 10.1166/jnn.2009.456.

[32] Li, C.; Chou, T. A direct electrifying algorithm for backbone identification. J. Phys. A Math. Gen. 2007, 40, 679-686, DOI: 10.1088/1751-8113/40/49/004.

[33] Sun, X. Highly conductive carbon nanotube/polymer nanocomposites achievable? Macromol. Theor. Simul. 2009, 18, 155–161, DOI: 10.1002/mats.200800091.

[34] Topinka, M.; Rowell, M.; Goldhaber-Gordon, D.; McGehee, M.; Hecht, DS.; Gruner, G. Charge transport in interpenetrating networks of semiconducting and metallic carbon nanotubes. Nano Lett. 2009, 9, 1866–1871, DOI: 10.1021/nl803849e.

[35] Oskouyi, A.; Sundararaj, U.; Mertiny, P. Tunneling conductivity and Piezoresistivity of composites containing randomly dispersed conductive nano-platelets. Materials. 2014, 7, 2501-2521, DOI: 10.3390/ma7042501.

Bibliography

[1] Beret, S.; Prausnitz, J. Densities of liquid polymers at high pressure. Pressure-volume-temperature measurements for polythylene, polyisobutylene, poly(vinyl acetate), and poly(dimethylsiloxane) to 1 kbar. Macromolecules. 1975, 8, 536–538, DOI: 10.1021/ma60046a028.

[2] Manfredi, L.; Rodriguez, E.; Wladyka-Przybylak, M.; Vazquez, A. Thermal degradation and fire resistance of unsaturated polyester modified acrylic resins and their composites with natural fibres. Polym. Degrad. Stab. 2006, 91, 255-261, DOI: 10.1016/j.polymdegradstab.2005.05.003.

[3] Drury, J.; Mutsaers, C.; Hart, C.; Matters, M.; de Leeuw D. Low-cost all-polymer integrated circuits. Appl. Phys. Lett. 1998, 73, 108-110, DOI: 10.1063/1.121783.

[4] Sheng, N.; Boyce, M.; Parks, D.; Rutledge, G.; Abes, J. Multiscale micromechanical modeling of polymer/clay nanocomposites and the effective clay particle. Polymer. 2004, 45, 487-506, DOI: 10.1016/j.polymer.2003.10.100.

[5] Chen, H.; Wang, M.; Lin, Y.; Chan, C.; Wu, J. Morphology and mechanical property of binary and ternary polypropylene nanocomposites with nanoclay and CaCo3 particles. J. Appl. Polym. Sci. 2007, 106, 3409–3416, DOI: 10.1002/app.27017.

[6] Ma, P.; Siddiqui, N.; Marom, G.; Kim, J. Dispersion and functionalization of carbon nanotubes for polymer-based nanocomposites: A review. Composites, Part A Composites. 2010, 41, 1345-1367, DOI: 10.1016/j.compositesa.2010.07.003.

[7] Akcora, P.; Liu, H.; Kumar, S.; Moll, J.; Brian Y.; Benicewicz, C.; Schadler, L.; Acehan, D.; Panagiotopoulos, A.; Pryamitsyn, V.; Ganesan, V.; Ilavsky, J.; Thiyagarajan, P.; Colby, R.;

Douglas, J. Anisotropic self-assembly of spherical polymer-grafted nanoparticles, Nat. Mater. 2009, 8, 354–359, DOI: 10.1038/nmat2404.

[8] Li, Z.; Okamoto, K.; Ohki, Y.; Tanaka, T. Effects of nano-filler addition on partial discharge resistance and dielectric breakdown strength of Micro-Al2O3Epoxy composite. IEEE Trans. Dielectr. Electr. Insul. 2010, 17, 653-661, DOI: 10.1109/TDEI.2010.5492235.

[9] Meincke, O.; Kaempfer, D.; Weickmann, H.; Friedrich, C.; Vathauer, M.; Warth, H. Mechanical properties and electrical conductivity of carbon-nanotube filled polyamide-6 and its blends with acrylonitrile/butadiene/styrene. Polymer, 2004, 45, 739-748, DOI: 10.1016/j.polymer.2003.12.013.

[10] Valavala, P.; Odegard, G.; Modeling techniques for determination of mechanical properties of polymer Nanocomposites. Department of Mechanical Engineering, Michigan Technological University, 1400 Townsend Drive, Houghton, MI 49931, USA, 2005

[11] Odegard, G.; Clancy, T.; Gates, T. Modeling of the mechanical properties of nanoparticle/polymer composites. Polymer, 2005, 46, 553-562, DOI: 10.1016/j.polymer.2004.11.022.

[12] Liu, Y.; Chen, X. Evaluations of the effective material properties of carbon nanotube-based composites using a nanoscale representative volume element. Mech. Mater. 2003, 35, 69–81, DOI: 10.1016/S0167-6636(02)00200-4.

[13] Prasad, K.; Das, B.; Maitra, U.; Ramamurty, U.; Rao, C. Extraordinary synergy in the mechanical properties of polymer matrix composites reinforced with 2 nanocarbons. PNAS. 2009, 106, 13186-13189, DOI: 10.1073/pnas.0905844106.

[14] Bewlay, B.; Jackson, M.; Lipsitt, H. The balance of mechanical and environmental properties of a multielement niobium-niobium silicide-basedIn Situ composite. Metall. Mater. Trans. A. 1996, 27, 3801–3808, DOI: 10.1007/BF02595629.

[15] Rana, D.; Lee, C.; Cho, K.; Lee, B.; Choe, S. Thermal and mechanical properties for binary blends of metallocene polyethylene with conventional polyolefins. J. Appl. Polym. Sci. 1998, 69, 2441–2450, DOI: 10.1002/(SICI)1097-4628(19980919)69:12<2441::AID-APP15>3.0.CO;2-%23.

[16] Sathe, S.; Devi, S.; Srinivasa, G.; Rao, K. Relationship between morphology and mechanical properties of binary and compatibilized ternary blends of polypropylene and nylon 6. J. Appl. Polym. Sci. 1996, 61, 97–107, DOI: 10.1002/(SICI)1097-4628(19960705)61:1<97::AID-APP11>3.0.CO;2-X.

[17] Eshelby, J. The determination of the elastic field of an ellipsoidal inclusion and related problems. Proc. R. Soc. London, Ser. A. 1957, 241, 561–569, DOI: 10.1098/rspa.1957.0133.

[18] Voigt, W. Ueber die Beziehung zwischen den beiden Elasticitätsconstanten isotroper Körper, Annalen der Physik, 1889, 274, 573–587, DOI: 10.1002/andp.18892741206.

[19] Halpin, J.; Kardos, J. The Halpin-Tsai equations: A review. Polym. Eng. Sci. 1976, 16, 344–352, DOI: 10.1002/pen.760160512.

[20] Hashin, Z.; Shtrikman, S. A variational approach to the theory of elastic behavior of multiphase materials. J. Mech. Phys. Solids, 1963, 11, 127–140, DOI: 10.1016/0022-5096(63)90060-7.

[21] Mori, T.; Tanaka, K. Average stress in the matrix and average elastic energy of materials with misfitting inclusions. Acta Metall. 1973, 21, 571–574, DOI: 10.1016/0001-6160(73)90064-3.

[22] Benveniste, Y. A new approach to the application of Mori–Tanaka's theory in composite materials. Mech. Mater. 1987, 6, 147–157, DOI: 10.1016/0167-6636(87)90005-6.

[23] Christensen, R.; Lo, K. Solutions for effective shear properties of three phase sphere and cylinder models. J. Mech. Phys. Solids, 1987, 27, 315–330, DOI: 10.1016/0022-5096(79)90032-2.

[24] Hill, R. A self-consistent mechanics of composite materials. J. Mech. Phys. Solids. 1965, 13, 213–222, DOI: 10.1016/0022-5096(65)90010-4.

[25] Torquato, S. Effective stiffness tensor of composite media: II. Applications to isotropic dispersions. J. Mech. Phys. Solids. 1998, 46, 1411–1440, DOI: 10.1016/S0022-5096(97)00083-5.

[26] Coleman, J.; Khan, U.; Gun'ko, Y. Mechanical reinforcement of polymers using carbon nanotubes. Adv. Mater. 2006, 18, 689-706, DOI: 10.1002/adma.200501851.

[27] Moniruzzaman, M.; Winey, K. Polymer nanocomposites containing carbon nanotubes. Macromolecules. 2006, 39, 5194–5205, DOI: 10.1021/ma060733p.

[28] Green, M.J.; Behabtu, N.; Pasquali, M.; Adams, W. Nano tubes as polymer. Polymer. 2009, 50, 4979-4997, DOI: 10.1016/j.polymer.2009.07.044.

[29] Liu, J.; Liu, J.; Yang, L.; Chen, X.; Zhang, M.; Meng, F.; Luo, T.; Li, M. Nanomaterial-assisted signal enhancement of hybridization for DNA biosensors: a review. Sensors. 2009, 9, 7343-7364, DOI: 10.3390/s90907343.

[30] Balasubramanian, K.; Burghard, M. Biosensors based on carbon nanotubes, Anal. Bioanal. Chem. 2006, 385, 452, DOI: 10.1007/s00216-006-0314-8.

[31] Fan, S.S.; Chapline, M.; Franklin, N.; Tombler, T.; Cassell, A.; Dai, H. Self-oriented regular arrays of carbon nanotubes and their field emission properties. Science. 1999, 283, 512-514, DOI: 10.1126/science.283.5401.512.

[32] Harris P. Carbon nanotubes and related structures: New materials for the twenty-first century. American Journal of Physics. 2004;72(3):415. DOI: 10.1119/1.1645289.

[33] Dai L. Carbon Nanotechnology: Recent Developments in Chemistry, Physics, Materials Science and Device Applications. 1st ed. Amsterdam: Elsevier; 2006. ISBN-13: 978-0-444-51855-2.

[34] Lin, F.; Bhatia, G.; Ford, J. Thermal conductivities of powder-filled epoxy resins. J. Appl. Polym. Sci. 1993, 49, 1901–1908, DOI: 10.1002/app.1993.070491105.

[35] Agari, Y.; Uno, T. Estimation on thermal conductivities of filled polymers. J. Appl. Polym. Sci. 1986, 32, 5705–5712, DOI: 10.1002/app.1986.070320702.

[36] Cheng, S.; Vachon, R. The prediction of the thermal conductivity of two and three phase solid heterogeneous mixtures. Int. J. Heat Mass Transfer. 1969, 12, 249-264, DOI: 10.1016/0017-9310(69)90009-X.

[37] Hasselman, D.; Johnson, L. Effective Thermal Conductivity of Composites with Interfacial Thermal Barrier Resistance. J. Compos. Mater. 1987, 21, 508-515, DOI: 10.1177/002199838702100602.

[38] Song, Y.; Youn, J. Influence of dispersion states of carbon nanotubes on physical properties of epoxy nanocomposites. Carbon. 2005, 43, 1378-1385, DOI: 10.1016/j.carbon.2005.01.007.

[39] Karasek, L.; Sumita, M. Characterization of dispersion state of filler and polymer-filler interactions in rubber-carbon black composites. J. Mater. Sci. 1996, 31, 281–289, DOI: 10.1007/BF01139141.

[40] Paul, M.; Alexandre, M.; Degee, P.; Henrist, C.; Rulmont, A.; Dubois, P. New nanocomposite materials based on plasticized poly(l-lactide) and organo-modified montmorillonites: thermal and morphological study. Polymer. 2003, 44, 443-450, DOI: 10.1016/S0032-3861(02)00778-4.

[41] Yung, K.; Liem, H. Enhanced thermal conductivity of boron nitride epoxy-matrix composite through multi-modal particle size mixing. J. Appl. Polym. Sci. 2007, 106, 3587–3591, DOI: 10.1002/app.27027.

[42] Miranda, O.; Yang, R.; Alvarado-Gil, J. A crowding factor model for the thermal conductivity of particulate composites at non-dilute limit. J. Appl. Phys. 2013, 114, 064306, DOI: 10.1063/1.4818409.

[43] Lin, Z.; Zhigilei, L. V.; Celli, V. Electron-phonon coupling and electron heat capacity of metals under conditions of strong electron-phonon non equilibrium. Phys. Rev. B: Condens. Matter. 2008, 77, 075133, DOI: 10.1103/PhysRevB.77.075133.

[44] Fan, Z.; Wei, T.; Luo, G.; Wei, F. Fabrication and characterization of multi-walled carbon nanotubes-based ink. J. Mater. Sci. 2005, 40, 5075–5077, DOI: 10.1007/s10853-005-1622-7.

[45] Kordas, K.; Mustonen, T.; Toth, G.; Jantunen, H.; Lajunen, M.; Soldano, C.; Talapatra, S.; Kar, S.; Vajtai, R.; Ajayan, P. Inkjet printing of electrically conductive patterns of carbon nanotubes. Small. 2006, 2, 1021-1025, DOI: 10.1002/smll.200600061.

[46] Small, W.; Panhuis, M. Inkjet printing of transparent electrically conducting single-walled carbon-nanotube composites. Small. 2007, 3, 1500-1503, DOI: 10.1002/smll.200700110.

[47] Wei, T.; Ruan, J.; Fan, Z.; Luo, G.; Wei, F. Preparation of graphene Nano sheet/polymer composites using in situ reduction–extractive dispersion. Carbon. 2007, 45, 2296-2299, DOI: 10.1016/j.carbon.2009.04.030.

[48] Dragoman, M.; Flahaut, E.; Dragoman, D.; Al Ahmad, M.; Plana, R. Writing simple RF electronic devices on paper with carbon nanotube ink. Nanotechnology. 2009, 20, 5203, DOI: 10.1088/0957-4484/20/37/375203/meta.

[49] Cao, Q.; Rogers, J. Ultrathin films of single-walled carbon nanotubes for electronics and sensors: a review of fundamental and applied aspects. Adv. Mater. 2009, 21, 29-53, DOI: 10.1002/adma.200801995.

[50] Avouris, P.; Chen, Z.H.; Perebeinos, V. Carbon-based electronics. Nanoscience and Technology. 2009, 2, 174-184, DOI: 10.1142/9789814287005_0018.

[51] Khabibullin, A.; Zharov, I. Fuel cell membranes based on polymer-modified silica colloidal crystals and glasses: Proton conductivity and fuel cell performance. Mater. Res. Soc. Symp. Proc. 2013, 1502, DOI: 10.1557/opl.2013.568.

[52] Tripathi, B.; Shahi, K. Prog. Organic–inorganic nanocomposite polymer electrolyte membranes for fuel cell applications. Prog. Polym. Sci. 2011, 36, 945-979, DOI: 10.1016/j.progpolymsci.2010.12.005.

[53] Winter, M.; Brodd, R. What are batteries, fuel cells, and supercapacitors. Chem. Rev. 2004, 104, 4245–4270, DOI: 10.1021/cr020730k.

[54] Gary, F. Book review: Polymer electrolytes. Royal Society of Chemistry, Cambridge, 1997, ISBN 0-85404-557-0, DOI: 10.1002/(SICI)1097-0126(199805)46:1<78::AID-PI16>3.0.CO;2-I.

[55] Kim, J.; Kang, P.; Chang Nho Y. Positive temperature coefficient behavior of polymer composites having a high melting temperature. J. Appl. Polym. Sci. 2004, 92, 394–401, DOI: 10.1002/app.20064.

[56] Nikkeshi, S.; Kudo, M.; Masuko, T. Dynamic viscoelastic properties and thermal properties of powder-epoxy resin composite. J. Appl. Polym. Sci. 1998, 69, 2593–2598, DOI: 10.1002/(SICI)1097-4628(19980926)69:13<2593::AID-APP9>3.0.CO;2-5.

[57] Lia, C.; Thostenson, E.; Chou, T. Dominant role of tunneling resistance in the electrical conductivity of carbon nanotube-based composites. Appl. Phys. Lett. 2007, 91, 223114, DOI: 10.1063/1.2819690.

[58] Seidel G.; Lagoudas D. A micromechanics model for the electrical conductivity of nanotube–polymer nanocomposites. J. Compos. Mater. 2009, 43, 917–941. DOI: 10.1177/ 0021998308105124.

[59] Postma, H.; Jonge, M.; Yao, Z.; Dekker, C. Electrical transport through carbon nanotube junctions created by mechanical manipulation. Phys. Rev. B. 2000, 62, R10 653–R10 656, DOI: 10.1103/PhysRevB.62.R10653. [60] Choi, E.; Brooks, J.; Eaton, D. Enhancement of thermal and electrical properties of carbon nanotube polymer composites by magnetic field processing. J. Appl. Phys. 2003, 94, 6034, DOI: 10.1063/ 1.1616638.

[61] Sandler, J.; Kirk, J.; Kinloch, I.; Shaffer, M.; Windle, A. Ultra-low electrical percolation threshold in carbon-nanotube epoxy composites. Polymer. 2003, 44, 5893–5899, DOI: 10.1016/S0032-3861(03)00539-1.

[62] Zhang, W.; Dehghani-Sanij, A.; Blackburn, R. Carbon based conductive polymer composites. J. Mater. Sci. 2007, 42, 3408–3418, DOI: 10.1007%2Fs10853-007-1688-5.

[63] Cai, W.; Tu, S.; Gong, J. A physically based percolation model of the effective electrical conductivity of particle filled composites. J. Compos. Mater. 2006, 40, 2131–2142, DOI: 10.1177/0021998306062312.

[64] Hindermann-Bischoff, M.; Ehrburger-Dolle, F. Electrical conductivity of carbon black–polyethylene composites: Experimental evidence of the change of cluster connectivity in the PTC effect. Carbon. 2001, 39, 375-382, DOI: 10.1016/S0008-6223(00)00130-5.

[65] Tang, H.; Chen, X.; Luo, Y. Studies on the PTC/NTC effect of carbon black filled low density polyethylene composites. Eur. Polym. J. 1997, 33, 1383-1386, DOI: 10.1016/S0014-3057(96)00221-2.

[66] Seidel, G.; Puydupin-Jamin, A. Analysis of clustering, interphase region, and orientation effects on the electrical conductivity of carbon nanotube–polymer nanocomposites via computational micromechanics. Mech. Mater. 2011, 43, 755–774, DOI: 10.1016/j.mechmat.2011.08.010.

[67] Natsuki, T.; Endo, M.; Takahashi, T. Percolation study of orientated short-fiber composites by a continuum model. Phys. Stat. Mech. Appl. 2005, 352, 498–508 DOI: 10.1016/j.physa.2004.12.059.

[68] Ma, H.; Gao, X. A three-dimensional Monte Carlo model for electrically conductive polymer matrix composites filled with curved fibers. Polymer. 2008, 49, 4230–4238, DOI: 10.1016/j.polymer.2008.07.034.

[69] Bao, W.; Meguid, S.; Zhu, Z.; Pan, Y.; Weng, G. A novel approach to predict the electrical conductivity of multifunctional nanocomposites. Mech. Mater. 2012, 46, 29–138, DOI: 10.1016/j.mechmat.2011.12.006.

[70] Wang, W.; Jayatissa, A. Computational and experimental study of electrical conductivity of graphene/poly(methyl methacrylate) nanocomposite using Monte Carlo method and percolation theory. Synth. Met. 2015, 204, 141–147, DOI: 10.1016/j.synthmet.2015.03.022.

[71] Haldar, A.; Mahadevan, S. Reliability assessment using stochastic finite element analysis. New York, NY, Wiley, 2000, ISBN: 978-0-471-36961-5.

[72] Koduru, S.; Haukaas, T. Feasibility of FORM in finite element reliability analysis. Struct. Saf. 2010, 32, 145–153, DOI: 10.1016/j. strusafe.2009.10.001.

[73] Thacker, B.; Riha, D.; Fitch, S.; Huyse, L.; Pleming, J. Probabilistic engineering analysis using the NESSUS software. Struct. Saf. 2006, 28, 83–107, DOI: 10.1016/j.strusafe.2004.11.003.

[74] Patelli, E.; Panayirci, M.; Broggi, M.; Goller, B.; Beaurepaire, P.; Pradlwarter, H.; Schuëller, G. General purpose software for efficient uncertainty management of large finite element models. Finite Elem. Anal. Des. 2012, 51, 31–48, DOI: 10.1016/j. finel.2011.11.003.

[75] Akula, V. Multiscale reliability analysis of a composite stiffened panel. Compos. Struct. 2014, 116, 432–440, DOI: 10.1016/j. compstruct.2014.06.001.

[76] McKenna, F.; Fenves, G.; Scott, M. Open system for earthquake engineering simulation. Berkeley, CA: University of California. Retrieved from http://opensees.berkeley.edu, 2000.

[77] Wang, J. Industrial design engineering: Inventive Problem Solving. Taylor and Francis group. 2017, ISBN: 9781498709606.

[78] Grimmett, G.; Stirzaker R. Probability and random processes. 2nd Edition. Clarendon Press, Oxford. ISBN: 0-19-853665-8.

[79] Sen, P.; Singer, J. Large sample methods in statistics. Singer. Chapman & Hall, Inc. 1994, ISBN: 0412042215.

[80] Teller, E.; Ulam, S. Heterocatalytic detonations, LAMS-1225, 2011.

[81] Brzenziak, Z. Basic Stochastic Processes: A Course Through Exercises (Springer Undergraduate Mathematics Series), 2000, ISBN: 3540761756,

[82] Zadeh, L. Fuzzy sets. Information and Control. 1965, 8, 338–353, DOI:10.1016/s0019-9958(65)90241-x.

[83] Maybeck, P. Stochastic models, Estimation and control, Mathematic in Science and Engineering, 141-3, ISBN: 0-12-480701-1.

[84] Guoliang, J.; Lin, C.; Changhai, L. The study of probability distribution of mechanical properties of steel used in pressure vessels for petrochemical industries, Journal of Daqing Petroleum Institute, 1991.

[85] Gagniuc, P. Markov Chains: From Theory to Implementation and Experimentation. USA, NJ: John Wiley & Sons. 2017, ISBN: 978-1-119-38755-8.

[86] Kwak, Y.; Ingall, L. Exploring Monte Carlo Simulation Applications for Project Management. Risk Manag. 2007, 9, 44–57, DOI: 10.1057/palgrave.rm.8250017.

[87] Power, D.; Sharda, R. Model-driven decision support systems: Concepts and research directions. Decision Support Systems. 2007, 43, 1044-1061, DOI: 10.1016/j.dss.2005.05.030.

[88] Edjer, T.; Baltussen, R.; Adam, T.; Hutubessy, R.; Acharya, A.; Evans, D.; Murray, C. Making Choices in Health: WHO Guide to Cost-effectiveness Analysis, ISBN 9241546018.

[89] L'Ecuyer, P. Random number generators, Encyclopedia of operations research and management science, Second Edition, 2001, 695-702, DOI: 10.1007/1-4020-0611-X_852.

[90] P. L'Ecuyer, History of uniform random number generation. Proceedings of the 2017 winter simulation conference, 2017, 13, 1–29, DOI: 10.5555/3242181.3242195.

[91] Barker, E.; Recommendation for Key Management, NIST Special Publication 800-57. NIST, 2013, DOI: 10.6028/NIST.SP.800-57pt1r4.

[92] Ecuyer, L. Uniform random number generators, International Encyclopedia of Statistical Science, Springer, 2010.

[93] Random (Java Platform SE 8), Java Platform Standard Edition 8, Documentation.

[94] Pollack, G. Kapitza Resistance. Rev. Mod. Phys. 1996, 41, 48, DOI: 10.1103/RevModPhys.41.48.

[95] ASTM D257-14, Standard Test Methods for DC Resistance or Conductance of Insulating Materials, ASTM International, West Conshohocken, PA, 2014, www.astm.org, DOI: 10.1520/D0257-14.

[96] Bigg, D. Mechanical properties of particulate filled polymers. Polym. Compos. 1987, 8, 115-122, DOI: 10.1002/pc.750080208.

[97] Ju, J.; Yanase, K. Micromechanics and effective elastic moduli of particle-reinforced composites with near-field particle interactions. Acta Mech. , 2010, 215, 135-153, DOI: 10.1007/s00707-010-0337-2.

[98] Smith, J. Experimental values for the elastic constants of a particulate-filled glassy polymer. J. Res. Nat. Bur. Stand. 1976, 80, 45–49, DOI: 10.6028/jres.080A.008.

[99] Kashiwagi, T.; Grulke, E.; Hilding, J.; Groth, K.; Harris, R.; Butler, K. Thermal and flammability properties of polypropylene/carbon nanotube nanocomposites. Polymer. 2004, 45, 4227-4239, DOI: 10.1016/j.polymer.2004.03.088.

[100] Liu, S.; Yan, H.; Fang, Z.; Wang, H. Effect of graphene nanosheets on morphology, thermal stability and flame retardancy of epoxy resin. Compos. Sci. Technol. 2014, 90, 40-47, DOI: 10.1016/j.compscitech.2013.10.012.

[101] Hahn, G. Sample Sizes for Monte Carlo Simulation. IEEE Transactions on Systems, Man and Cybernetics, 1972.

[102] Guoliang, J.; Lin, C.; Jiamei, D. Monte Carlo finite element method of structure reliability analysis. Reliab. Eng. Syst. Saf. 1993, 40, 77-83, DOI: 10.1016/0951-8320(93)90121-E

[103] Pilkey, W.; Pilkey, D. Stress concentration factors. Wiley, Third edition, DOI: 10.1002/9780470211106.

[104] Ostoja-Starzewski, M. Material spatial randomness: from statistical to representative volume element. Probab. Eng. Mech. 2006, 21, 112–32, DOI: 10.1016/j.probengmech.2005.07.007.

[105] Drugan, W.; Willis, J. A micromechanics-based nonlocal constitutive equation and estimates of the representative volume element size for elastic composites. J. Mech. Phys. Solids. 1996, 44, 497–524, DOI: 10.1016/0022-5096(96)00007-5.

[106] Hill, R. Elastic properties of reinforced solids: some theoretical principles. J. Mech. Phys. Solids. 1963, 11, 357–72, DOI: 10.1016/0022-5096(63)90036-X.

[107] Karimi, D.; Milani, A. SRVE modeling of particulate polymer matrix composites with irregularly shaped inclusions: Application to a green stone composite. Compos. Struct. 2019, 228, 111331, DOI: 10.1016/j.compstruct.2019.111331.

[108] Reuss, A. Berechnung der Fließgrenze von Mischkristallen auf Grund der Plastizitätsbedingung für Einkristalle. Zeitschrift für Angewandte Mathematik und Mechanik 1929, 9, 49–58, DOI: 10.1002/zamm.19290090104.

[109] Drugan, W. Micromechanics-based variational estimates for a higher-order nonlocal constitutive equation and optimal choice of effective moduli of elastic composites. J. Mech. Phys. Solids. 2000, 48, 1359–1387, DOI: 10.1016/S0022-5096(99)00088-5.

[110] Segurado, J.; Llorca, J. A numerical approximation to the elastic properties of sphere-reinforced composites. J Mech Phys Solids 2002, 50, 2107-2121, DOI: 10.1016/S0022-5096(02)00021-2.

[111] Lopez, O.; Choplin, L.; Tanguy, P. Thermorheological analysis of glass beads-filled polymer melts. Polym Eng Sci 1990, 30, 821-828, DOI: 10.1002/pen.760301404.

[112] Bibikov, S.; Prokof'Ev, M. Composite Materials for some radiophysics applications. Metal, Ceramic and Polymeric Composites for Various Uses, InTech Open, 2011, DOI: 10.5772/18074.

[113] Ahmadi Moghaddam, H.; Mertiny, P. Stochastic finite element modeling of char-forming filler addition and alignment – Effects on heat conduction in polymer condensed phase. In Flame Retardants; Zafar, F.; Sharmin, E.; Intech Open: London, United Kingdom, 2018, DOI: 10.5772/intechopen.82878.

[114] Balasubramanian, K.; Burghard, M. Electrochemically functionalized carbon nanotubes for device applications. J Mater Chem 2008, 26, 452, DOI: 10.1039/B718262G.

[115] Ahmadi Moghaddam, H.; Mertiny, P. Stochastic finite element analysis framework for modelling thermal conductivity of particulate modified polymer composites. Results Phys 2018, 11, 905-914, DOI: 10.1016/j.rinp.2018.10.045.

[116] Technical Data Sheet 758, Glass Beads, Polysciences, Inc., 2016.

[117] Jeyranpour, F.; Alahyarizadeh, Gh.; Arab, B. Comparative investigation of thermal and mechanical properties of cross-linked epoxy polymers with different curing agents by molecular dynamics simulation. J Mol Graph Model 2015, 62, 157-164, DOI: 10.1016/j.jmgm.2015.09.012.

[118] West, S. G.; Finch, J.F.; Curran, P. J. Structural equation models with nonnormal variables: Problems and remedies. In Structural Equation Modeling: Concepts, Issues, and Applications; Hoyle, R.H., Ed.; Sage Publications: Thousand Oaks, CA, USA, 1995, pp. 56-75.

[119] Deng, F.; Van Vliet, K.J. Prediction of elastic properties for polymer–particle nanocomposites exhibiting an interphase. Nanotechnol 2001, 22, 165703, DOI: 10.1088/0957-4484/22/16/165703.

[120] Upadhyay, A.; Singh, R. Prediction of Effective Elastic Modulus of Biphasic Composite Materials. Mod Mech Eng 2012, 2, 6-13, DOI: 10.4236/mme.2012.21002.

[121] Maxwell, J. A treatise on electricity and magnetism, volume I, 3rd Edition, Oxford University Press, 1904.

[122] Bird R.; Stewart W.; Lightfoot, E. Transport phenomena,John Wiley & Sons, 2007, ISBN: 0-470-11539-4.

[123] Eucken A. Die Wärmeleitfähigkeit Keramischer, Fester Stoffe– Ihre Berechnung aus der Wärmeleitfähigkeit der Bestandteile, VDI Forschungsheft 353, Beilage zu, Forschung auf dem Ggebiete des Ingenieurwesens, Ausgabe B, Band 3, 1932.

[124] Burger, H. Das lertvermogen verdumter mischkristallfreierlonsungen, Phys. Zs. 20, 1915.

[125] Hamilton, R.; Crosser, O. Thermal conductivity of heterogeneous two component systems, Ind. Eng. Chem. Fundam. 1962, 1, 187-191, DOI: 10.1021/i160003a005.

[126] Lord Rayleigh Sec. R.S. LVI. On the influence of obstacles arrangedin rectangular order upon the properties of a medium. The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science, 1892, 34, 481-502, DOI: 10.1080/14786449208620364.

[127] Bohm, H.; Nogales, S. Mori-Tanaka Model for the thermal conductivity of composites with interfacial resistance and particle size distributions. Compos. Sci. Technol. 2008, 68, 1181-1187, DOI: 10.1016/j.compscitech.2007.06.009.

[128] Majumdar, C.; Gerner, A. Microscale energy transport. Washington, D.C.: Taylor & Francis., 1998, ISBN: 978-1560324591.

[129] Chen, G. Nanoscale energy transport and conversion: a parallel treatment of electrons, molecules, phonons, and photons, New York: Oxford. ISBN: 978-0195159424, 2004.

[130] Skinner, M.; Mertiny, P. Effect of Random and Size-Ordered Addition of Spherical Nanoparticles Inside a Representative Volume Element, CAMX conference, 2016.

[131] Rivière, L.; Lonjon, A.; Dantras, E.; Lacabanne, C.; Olivier, P.; Gleizes, N. Silver fillers aspect ratio influence on electrical and thermal conductivity in PEEK/Ag nanocomposites. Eur. Polym. J. 2016, 85, 115-125, DOI: 10.1016/j.eurpolymj.2016.08.003.

[132] Mishra, D.; Satapathy, A.; Patnaik, A. A simulation study on thermal conductivity of glass bead embedded polymeric system, Int. J. Plast. Technol. 2012, 16, 24–38, DOI: 10.1007/s12588-012-9028-2.

[133] Liu, S.; Chevalia, V.; Xu, Z.; Hui, D.; Wang H. A review of extending performance of epoxy resins using carbon nanomaterials. Composites Part B: Engineering. 2018, 136, 197-214. DOI: 10.1016/j.compositesb.2017.08.020.

[134] Mishra, D.; Mohapatra, S.; Satapathy, A. A detailed investigation on thermal and micro-structural properties of hexagonal boron nitride composites. Materials Today: Proceed- ings. 2018, 5, 19747-19753, DOI: 10.1016/j.matpr.2018.06.337.

[135] Ansari, R.; Rouhi, S.; Ahmadi, M. On the thermal conductivity of carbon nanotube/polypropylene nanocomposites by finite element method. J. Comput. Appl. Mech. 2018, 49, 70-85, DOI: 10.22059/jcamech.2017.243530.195.

[136] Shi, X.; Hassanzadeh Aghdam, M.; Ansari, R. Effect of aluminum carbide interphase on the thermomechanical behavior of carbon nanotube/aluminum nanocomposites. Proceedings of the Institution of Mechanical Engineers, Part L: Journal of Materials: Design and Applications. 2018, 233, DOI: 10.1177/1464420718794716.

[137] Ahmadi, M.; Ansari, R.; Hassanzadeh Aghdam, M. Micromechanical analysis of elastic modulus of carbon nanotube-aluminum nanocomposites with random microstructures. J. Alloys Compd. 2019, 779, 433-439, DOI: 10.1016/j.jallcom.2018.11.326.

[138] Ahmadi, M.; Ansari, R.; Rouhi, S. Investigating the thermal conductivity of concrete/graphene nanocomposite by a multi-scale modeling approach. Int. J. Mod. Phys. B. 2018, 32, 1850171, DOI: 10.1142/S0217979218501710.

[139] Pop, E.; Mann, D.; Wang, Q.; Goodson, K.; Dai, H. Thermal conductance of an individual single-wall carbon nanotube above room temperature. Nano Letters. 2006, 6, 96-100, DOI: 10.1021/nl052145f.

[140] Balandin, A.; Ghosh, S.; Bao, W.; Calizo, I.; Teweldebrhan, D.; Miao, F. Superior thermal conductivity of single-layer graphene. Nano Letters. 2008, 8, 902-907, DOI: 10.1021/nl0731872.

[141] Yan, T.; Wang, Z.; Wang, YQ.; Pan, Z. Carbon/graphene composite nanofiber yarns for highly sensitive strain sensors. Mater. Des. 2018, 143, 214-223, DOI: 10.1016/j.matdes.2018.02. 006.

[142] Wu, Q.; Zhu, W.; Zhang, C.; Liang, Z.; Wang, B. Study of fire retardant behavior of carbon nanotube membranes and carbon nanofiber paper in carbon fiber reinforced epoxy com- posites. Carbon. 2010, 48, 1799-1806, DOI: 10.1016/j.carbon.2010.01.023.

[143] Knight, C.; Filbert, I.; Zeng, C.; Zhange, C.; Wang B. A highly efficient fire-retardant nanomaterial based on carbon nanotubes and magnesium hydroxide. Fire Mater. 2013, 37, 91-99, DOI: 10.1002/fam.2115.

[144] Xiong, L.; Liu, K.; Shuai, J.; Hou, Z.; Zhu, L.; Li, W. Toward high strength and high electrical conductivity in super-aligned carbon nanotubes reinforced copper. Adv. Eng. Mater. 2018, 20, 1700805, DOI: 10.1002/adem.201700805.

[145] Xiong, L.; Shuai, J.; Liu, K.; Hou, Z.; Zhu, L.; Li, W. Enhanced mechanical and electrical properties of super-aligned carbon nanotubes reinforced copper by severe plastic deformation. Composites, Part B. 2018, 160, 315-320, DOI: 10.1016/j.compositesb.2018.10.023.

[146] Sinha, S.; Barjami, S.; Iannacchione, G.; Schwab, A.; Muench, G. Off-axis thermal properties of carbon nanotube films. J. Nanopart. Res. 2005, 7, 651-657, DOI: 10.1007/ s11051-005-8382-9.

[147] Sugime, H.; Esconjauregui, S.; Yang, J.; D'Arsié, L.; Oliver, R.; Bhardwaj, S. Low temperature growth of ultra-high mass density carbon nanotube forests on conductive supports. Appl. Phys. Lett. 2013, 103, 109901, DOI: 10.1063/1.4818619.

[148] Huxtable, S.; Cahill, D.; Shenogin, S.; Xue, L.; Ozisik, R.; Barone, P. Interfacial heat flow in carbon nanotube suspensions. Nat. Mater. 2003, 2, 731-734, DOI: 10.1038/nmat996.

[149] Shenogin, S.; Xue, L.; Ozisik, R.; Keblinski, P.; Cahill, D. Role of thermal boundary resistance on the heat flow in carbon-nanotube composites. J. Appl. Phys. 2004, 95, 8136-8144, DOI: 10.1063/1.1736328.

[150] Han, Z.; Fina, A. Thermal conductivity of carbon nanotubes and their polymer nanocomposites: A review. Prog. Polym. Sci. 2011, 36, 914-944, DOI: 10.1016/j.progpolymsci.2010.11.004.

[151] Xu, D.; Wang, Z.; Douglas, J. Influence of carbon nanotube aspect ratio on normal stress differences in isotactic polypropylene nanocomposite melts. Macromolecules. 2008, 41, 815-825, DOI: 10.1021/ma702178e.

[152] Jiang, Q.; Wang, X.; Zhu, Y.; Hui, D.; Qiu, Y. Mechanical, electrical and thermal properties of aligned carbon nanotube/polyimide composites. Composites, Part B. 2014, 56, 408-412, DOI: 10.1016/j.compositesb.2013.08.064.

[153] Pan, C.; Hocheng, H. Evaluation of anisotropic thermal conductivity for unidirectional FRP in laser machining. Composites, Part A. 2001, 32, 1657-1667, DOI: 10.1016/S1359-835X(00)00093-2.

[154] Yang, K.; Gu, M.; Guo, Y.; Pan, X.; Mu, G. Effects of carbon nanotube functionalization on the mechanical and thermal properties of epoxy composites. Carbon. 2009, 47, 1723-1737, DOI: 10.1016/j.carbon.2009.02.029.

[155] Ghosh, P.; Chakrabarti, A. Conducting carbon black filled EPDM vulcanizates: assessment of dependence of physical and mechanical properties and conducting character on variation of filler loading. Eur. Polym. J. 2000, 36, 1043-1054, DOI: 10.1016/S0014-3057(99)00157-3.

[156] Sohi, N.; Bhadra, S.; Khastgir, D. The effect of different carbon fillers on the electrical conductivity of ethylene vinyl acetate copolymer-based composites and the applicability of different conductivity models. Carbon. 2011, 49, 1349-1361, DOI: 10.1016/j.carbon.2010.12.001.

[157] Natsuki, T.; Tantrakarn, K.; Endo, M. Prediction of elastic properties for single-walled carbon nanotubes. Carbon. 2004, 42, 39–45, DOI: 10.1016/j.carbon.2003.09.011.

[158] Meo, M.; Rossi, M. A molecular-mechanics based finite element model for strength prediction of single wall carbon nanotubes. Mater. Sci. Eng. A. 2007, 454–455, 170–177, DOI: 10.1016/j.msea.2006.11.158.

[159] Tserpes, K.; Papanikos, P.; Tsirkas, S. A progressive fracture model for carbon nanotubes, Composites Part B: Engineering. 2006, 37, 662–669, DOI: 10.1016/j.compositesb.2006.02.024.

[160] Belytschko, T.; Xiao, S.; Schatz, G.; Ruoff, R. Atomistic simulations of nanotube fracture. Physical Review B. 2002, 65, 235430, DOI: 10.1103/PhysRevB.65.235430.

[161] Wu, L.; Tan, S.; Bosman, M.; Yang, J.; Nijhuisbf, C.; Bai, P. Charge transfer plasmon resonances across silver–molecule–silver junctions: estimating the terahertz conductance of molecules at near-infrared frequencies. RSC Adv. 2016, 6, 70884-70894, DOI: 10.1039/C6RA16826D.

[162] Nam, S.; Cho, H.; Kim, T.; Kim, D.; Sung, B.; Lim, S.; Kim, H. Effects of silica particles on the electrical percolation threshold and thermomechanical properties of epoxy/silver nanocomposites. Appl. Phys. Lett. 2011, 99, 043104, DOI: 10.1063/1.3615690.

[163] Untereker, D.; Lyu, S.; Schley, J.; Martinez, G.; Lohstreter, L. Maximum Conductivity of Packed Nanoparticles and Their Polymer Composites. Appl. Mater. Interfaces. 2009, 1, 97–101, DOI: 10.1021/am800038z.

[164] Jiguet, S.; Bertsch, A.; Hofmann, H.; Renaud, P. SU8-Silver Photosensitive Nanocomposites, Adv. Eng. Mater. 2004, 6(9), DOI: 10.1002/adem.200400068.

[165] Jouni, M.; Boudenne, A.; Boiteux, G.; Massardier, V.; Garnier, B.; Serghei, A. Electrical and Thermal Properties of Polyethylene/Silver Nanoparticle Composites. Polym. Compos. 2013, 34, 778-786, DOI: 10.1002/pc.22478.

[166] Feller, J.; Bruzaud, S.; Grohens, Y. Influence of clay nanofiller on electrical and rheological properties of conductive polymer composite. Mater. Lett. 2004, 58, 739-745, DOI: 10.1016/j.matlet.2003.07.010.

[167] Villacorta, B.; Ogale, A.; Hubing, T. Effect of heat treatment of carbon nanofibers on the electromagnetic shielding effectiveness of linear low density polyethylene nanocomposites. Polym. Eng. Sci. 2012, 53, 417, DOI: 10.1002/pen.23276.

[168] Yuan, Q.; Wu, D. Low percolation threshold and high conductivity in carbon black filled polyethylene and polypropylene composites. J. Appl. Polym. Sci. 2009, 115, 369-383, DOI: 10.1002/app.30919.

[169] Mondal, S.; Ganguly, S.; Rahaman, M.; Aldalbahi, A.; Chaki, T.; Khastgir, D.; Das, N. A strategy to achieve enhanced electromagnetic interference shielding at low concentration with a new generation of conductive carbon black in a chlorinated polyethylene elastomeric matrix. Phys. Chem. Chem. Phys. 2016, 18, 24591-24599, DOI: 10.1039/C6CP04274K.

[170] Nisar, M.; Bergmann, C.; Geshev, J.; Quijada, R.; Maraschin, T.; Basso, N.; Barrera, E.; Galland, G. Synthesis of high-density polyethylene/rGO-CNT-Fe nanocomposites with

outstanding magnetic and electrical properties. J. Appl. Polym. Sci. 2017, 134, 45382, DOI: 10.1002/app.45382.

[171] Villacorta, B.; Ogale, A. Morphological influence of carbon modifiers on the electromagnetic shielding of their linear low density polyethylene composites. J. Appl. Polym. Sci. 2014, 131, 41055, DOI: 10.1002/app.41055.

[172] Li, C.; Chou, T. Multiscale modeling of compressive behavior of carbon nanotube polymer composites. Compos. Sci. Technol. 2006, 66, 2409–2414, DOI: 10.1016/j.compscitech.2006.01.013.

[173] Xiao, J.; Gillespie, J. Nanomechanics of single-walled carbon nanotubes as composite reinforcement. Polym. Eng. Sci. 2006, 46, 1051–1059, DOI: 10.1002/pen.20553.

[174] Shi, D.; Feng, X.; Huang, Y.; Hwang, K.; Gao, H. The effect of nanotube waviness and agglomeration on the elastic property of carbon nanotube-reinforced composites. J. Eng. Mater. Technol. 2004, 126, 250–257, DOI: 10.1115/1.1751182.

[175] Song, Y.; Youn, J. Evaluation of effective thermal conductivity for carbon nanotube/polymer composites using control volume finite element method. Carbon. 2006, 44, 710–717, DOI: 10.1016/j.carbon.2005.09.034.

[176] Xue, Q. Model for thermal conductivity of carbon nanotube-based composites. Phys. Rev. B. 2005, 368, 302–307, DOI:10.1016/j.physb.2005.07.024.

[177] Bagchi, A.; Nomura, S. On the effective thermal conductivity of carbon nanotube reinforced polymer composites. Compos. Sci. Technol. 2006, 66, 1703–1712, DOI: 10.1016/j.compscitech.2005.11.003.

[178] Zhang, J.; Tanaka, M. Systematic study of thermal properties of CNT composites by the fast multipole hybrid boundary node method. Eng. Anal. Bound. Elem. 2007, 31, 388–401, DOI: 10.1016/j.enganabound.2006.07.011.

[179] Pike, G.; Seager, C. Percolation and conductivity: a computer study I. Phys. Rev. B. 1974, 10, 1421–1434, DOI: 10.1103/PhysRevB.10.1421.

[180] Kirkpatrick, S. Percolation and conduction. Rev. Mod. Phys. 1973, 45, 574–588, DOI: 10.1103/RevModPhys.45.574.

[181] Behnam, A.; Ural, A. Computational study of geometry-dependent resistivity scaling in single-walled carbon nanotube films. Phys. 2007, 75, 125432, DOI: 10.1103/PhysRevB.75.125432.

[182] Dalmas, F.; Dendievel, R.; Chazeau, L.; Cavaillé, J.; Gauthier, C. Carbon nanotube-filled polymer composites. Numerical simulation of electrical conductivity in three-dimensional entangled fibrous networks. Acta Mater. 2006, 54, 2923–2931, DOI: 10.1016/j.actamat.2006.02.028.

[183] Hakobyan, Y.; Papoulia, K.; Grigoriu, M. Physical and geometrical percolations of effective conductivity on a lattice. Phys, Rev. B. 2007, 76, 144205, DOI: 10.1103/PhysRevB.76.144205.

[184] Foygel, M.; Morris, R.; Anex, D.; French, S.; Sobolev, V. Theoretical and computational studies of carbon nanotube composites and suspensions: electrical and thermal conductivity. Phys, Rev. B. 2005, 71, 104201, DOI: 10.1103/PhysRevB.71.104201.

[185] Hu, N.; Masuda, Z.; Yan, C.; Yamamoto, G.; Fukunaga, H.; Hashida, T. The electrical properties of polymer nanocomposites with carbon nanotube fillers. Nanotechnology. 2008, 19, 215701, DOI: 10.1088/0957-4484/19/21/215701.

[186] Jack, D.; Yeh, C.; Liang, Z.; Li, S.; Park, J.; Fielding, J. Electrical conductivity modeling and experimental study of densely packed SWCNT networks. Nanotechnology. 2010, 21, 195703, DOI: 10.1088/0957-4484/21/19/195703.

[187] Li, C.; Chou, T. Electrical conductivities of composites with aligned carbon nanotubes. J. Nanosci. Nanotechnol. 2009, 9, 2518–2524, DOI: 10.1166/jnn.2009.456.

[188] Li, C.; Chou, T. A direct electrifying algorithm for backbone identification. J. Phys. A Math. Gen. 2007, 40, 679-686, DOI: 10.1088/1751-8113/40/49/004.

[189] Sun, X. Highly conductive carbon nanotube/polymer nanocomposites achievable? Macromol. Theor. Simul. 2009, 18, 155–161, DOI: 10.1002/mats.200800091.

[190] Topinka, M.; Rowell, M.; Goldhaber-Gordon, D.; McGehee, M.; Hecht, DS.; Gruner, G. Charge transport in interpenetrating networks of semiconducting and metallic carbon nanotubes. Nano Lett. 2009, 9, 1866–1871, DOI: 10.1021/nl803849e.

[191] Oskouyi, A.; Sundararaj, U.; Mertiny, P. Tunneling conductivity and Piezoresistivity of composites containing randomly dispersed conductive nano-platelets. Materials. 2014, 7, 2501-2521, DOI: 10.3390/ma7042501.

[192] Hicks, J.; Behnam, A.; Ural, A. A computational study of tunneling-percolation electrical transport in graphene-based nanocomposites. Appl. Phys. Lett. 2009, 95, 213103, DOI: 10.1063/1.3267079.

[193] Luheng, W.; Tianhuai, D.; Peng, W. Influence of carbon black concentration on Piezoresistivity for carbon black-filled silicone rubber composite. Carbon. 2009, 47, 3151–3157, DOI: 10.1016/j.carbon.2009.06.050.

[194] Hu, N.; Karube, Y.; Yan, C.; Masuda, Z.; Fukunaga, H. Tunneling effect in a polymer/carbon nanotube nanocomposite strain sensor. Acta Mater. 2008, 56, 2929–2936, DOI: 10.1016/j.actamat.2008.02.030.

[195] Pham, G. Characterization and Modeling of Piezo-Resistive Properties of Carbon Nanotube-Based Conductive Polymer Composites. Ph.D. Thesis, Florida State University, Tallahassee, FL, USA, November 2008.

[196] Simmons, J. G. Generalized Formula for the Electric Tunnel Effect Between Similar Electrodes Separated by a Thin Insulating Film. J. Appl. Phys. 1963, 34, 1793–1803, DOI: 10.1063/1.1702682.

[197] Simmons, J. G., Generalized Thermal J-V Characteristic for the Electric Tunnel Effect, J. Appl. Phys. 1964, 35, 2655–2658, DOI: 10.1063/1.1713820.

[198] Soltannia, B.; Haji Gholami, I.; Masajedian, S.; Mertiny, P.; Sameoto, D.; Taheri, F. Parametric Study of Strain Rate Effects on Nanoparticle-Reinforced Polymer Composites. J. Nanomater. 2016, 1–9, DOI: 10.1155/2016/9841972.

[199] Soltannia, B.; Mertiny, P.; Taheri, F. Static and dynamic characteristics of nano-reinforced 3D-fiber metal laminates using non-destructive techniques. Journal of Sandwich Structures & Materials. 2020, DOI: 10.1177/1099636220924585.

Appendix I

The following is the APDL code that was used for creating particle and RVE geometries in Section 2.1 of this thesis.

```
FINISH
/CLEAR
! Set name for this model
!/filnam,teeterbox4
/PREP7
*ASK,RAW,NUMBER OF THE PARTICLES PLEASE,3
COLUMN=7
KCN=10
*DIM,ATAB,table,RAW,COLUMN
*TREAD,ATAB,coordinates,csv,c:\\Samples
*DIM,AARR,array,RAW,COLUMN+1
*MFUN,AARR(1,1),copy,ATAB(0,0)
*DEL,ATAB(1)
!*status,ATAB,,raw,,column
!*status,AARR,,raw,,column
*DO,i,2,raw,1
KCN=KCN+1
LOCAL,KCN,0,AARR(i,2),AARR(i,3),AARR(i,4),AARR(i,5),AARR(i,6),AARR(i,7)
```

```
WPCSYS,1,KCN

SPHERE,0,AARR(i,8),0,360

*enddo

Finish
```

Appendix II

The following sample VBA code enables running the SFEA framework in Microsoft Access and generating the outcomes presented in Chapter 2 and 3. Similar codes were utilized for the studies presented in Chapters 4 and 5.

```
Option Compare Database
'Ansys Macro Version 1.0
'Developed By: Hamid Ahmadi
'First row is considered for programming purpose Database is available from
second row (ID = 2)
Declare Sub Sleep Lib "kernel32" _
(ByVal dwMilliseconds As Long)
Private Type settings
    Data1 As String
    Data2 As String
End Type
Public DatabasePath As String
Public TableName As String
Public ExcelPath1 As String
Public AnsysEXE As String
Public MatlabEXE As String
Public Record As String
Public ColumnName As String
Public ID As Long
Public StopID As Long
Public Progress As Long
Public Inc As Long
Sub Main()
    Dim FieldName As String
    Dim ExcelFilePath As String
    Dim ExeFilePath As String
    Dim NewDataEntry As Boolean
    Dim LoopCheck As Boolean
    Dim MessageTo As String
    Dim Subject As String
    Dim Hyperlink As String
    Dim LastAnalysis As Double
    Dim CompletedIteration As Double
    Dim CurrentVolumeFraction As Double
    Dim CurrentIteration As Double
    Dim VFPath As String
    Dim GeometriesPath As String
    Dim ResultsPath As String
    Dim FinalResultsPath As String
    Dim FinalResultName As String
    Dim FinalResultfile As String
    Dim RVESize As Double
    Dim Displacement As Double
    Dim Temperature1 As Double
    Dim Temperature2 As Double
    Dim VF1 As Double
    Dim VF2 As Double
    Dim VF3 As Double
    Dim VF4 As Double
    Dim VF5 As Double
    Dim VF6 As Double
    Dim VF7 As Double
    Dim VF8 As Double
    Dim VF9 As Double
    Dim VF10 As Double
    Dim VF11 As Double
    Dim VF12 As Double
    Dim MechAcceptanceCriteria As Double
    Dim ThermAcceptanceCriteria As Double
    Dim MechanicalCheck As Boolean
    Dim ThermalCheck As Boolean
    Dim CurrentMechSTD As Double
    Dim CurrentThermSTD As Double
    Dim SDX1 As Double                  'Biggest size distribution "X1" from
database
    Dim Pi As Double
    Dim DestPath As String


    'Defining valriables.
    DatabasePath = "C:\Users\hahmadi\Google
Drive\Codes\Project1\Database.accdb"
    TableName = "tblDatabase"
    ColumnName = "AnalysisID"
    Progress = 0
    MechAcceptanceCriteria = 0.000000000000001
    ThermAcceptanceCriteria = 0.000000000000001
    CurrentMechSTD = MechAcceptanceCriteria * 0.9
    CurrentThermSTD = ThermAcceptanceCriteria * 0.9
    Pi = 4 * Atn(1)


    'Finding the last completed analysis in the Database and setting VF and
Iteration varabales based on that.
    LastAnalysis = GettingLastCompletedAnalysisRecord("tblDatabase",
"AnalysisFinished", _
    "C:\Users\hahmadi\Google Drive\Codes\Project1\Database.accdb")
    'Finding ID of first analysis to get worked on
    ID = LastAnalysis + 1
    'Finding completed Volume Fraction of first analysis to work on
    CurrentVolumeFraction = ReadFromAccessTable("tblDatabase", "AnalysisID",
ID, _
    "VFCompleted", "C:\Users\hahmadi\Google
Drive\Codes\Project1\Database.accdb")
    'Finding completed Iteration of first analysis to work on
    CompletedIteration = ReadFromAccessTable("tblDatabase", "AnalysisID", ID, _
    "IterCompleted", "C:\Users\hahmadi\Google
Drive\Codes\Project1\Database.accdb")
    'Calculating Current Iteration to work on
    CurrentIteration = CompletedIteration + 1

    'Setting variavles needed to be checked in the loop
    StopID = 1
    LoopCheck = True



    'Defining master excel file path
    ExcelPath1 = "C:\Users\hahmadi\Google
Drive\Codes\Project1\MaterialDatabase.xlsm"

    'Defining ANSYS macros path
    AnsysEXE = "C:\Users\hahmadi\Google Drive\Codes\Project1\WBFile.exe"

    'Defining Matlab macros path
    MatlabEXE = "C:\Users\hahmadi\Google
Drive\Codes\Project1\SphereColisionDetectionMacro.exe"

    'Setting macro Status to "Running".
    Forms!frmRunMacro.txtStatus.Value = "Macro is running."

    'Setting Project name and project status in Macro form
    If CurrentVolumeFraction = 0 Then
        Forms!frmRunMacro.txtProjectName.Value = "Analyses were completed and
macro has nothing to work on!!! "
        Forms!frmRunMacro.txtProjectStatus.Value = "Please enter new data to
database."
    Else
        Forms!frmRunMacro.txtProjectName.Value =
ReadFromAccessTable("tblDatabase", "AnalysisID", ID, "ProjectName", _
        "C:\Users\hahmadi\Google Drive\Codes\Project1\Database.accdb")
        Forms!frmRunMacro.txtProjectStatus.Value = "Next analysis will be
volume fraction: " & CurrentVolumeFraction _
        & " ,Iteration: " & CurrentIteration & "."
    End If
    Sleep (2500)

    'Main loop
    Do While LoopCheck = True
        'Checking the new data entry.
        Do While NewDataEntry = False
            'Finding the last completed analysis in the Database and setting VF
and Iteration varabales based on that.
            LastAnalysis = GettingLastCompletedAnalysisRecord("tblDatabase",
"AnalysisFinished", _
            "C:\Users\hahmadi\Google Drive\Codes\Project1\Database.accdb")
            'Finding ID of first analysis to get worked on
            ID = LastAnalysis + 1
            'Finding completed Volume Fraction of first analysis to work on
            CurrentVolumeFraction = ReadFromAccessTable("tblDatabase",
"AnalysisID", ID, _
```

```
            "VFCompleted", "C:\Users\hahmadi\Google
Drive\Codes\Project1\Database.accdb")
            'Finding completed Iteration of first analysis to work on
            CompletedIteration = ReadFromAccessTable("tblDatabase",
"AnalysisID", ID, _
            "IterCompleted", "C:\Users\hahmadi\Google
Drive\Codes\Project1\Database.accdb")
            'Calculating Current Iteration to work on
            CurrentIteration = CompletedIteration + 1
            'Stop Macro from functioning if licence is set to "Stop"
            If ReadFromAccessTable(TableName, "AnalysisID", StopID,
"StopTheLoop", DatabasePath) = "Stop" Then
                MsgBox "Ansys macro Stopped working." & vbNewLine & _
                "Verify your licence status."
                'Changing Form status to reflect macro is Stopped.
                Forms!frmRunMacro.txtStatus.Value = "Macro stopped working."
                Forms!frmRunMacro.txtProjectName.Value = "Analyses were
completed and macro has nothing to work on!!! "
                Forms!frmRunMacro.txtProjectStatus.Value = "Please enter new
data to database."
                End
            End If
            'Check last row volume fraction  in the database every 20 secends.
            If CurrentVolumeFraction <> 0 Then
                NewDataEntry = True
                Forms!frmRunMacro.txtProjectName.Value =
ReadFromAccessTable("tblDatabase", "AnalysisID", ID, "ProjectName", _
                "C:\Users\hahmadi\Google Drive\Codes\Project1\Database.accdb")
                Forms!frmRunMacro.txtProjectStatus.Value = "Next analysis will
be volume fraction: " & CurrentVolumeFraction _
                & " ,Iteration: " & CurrentIteration & "."
                Sleep (2500)
            Else
                If CheckNewDataEntry(TableName, "AnalysisID", ID, DatabasePath)
Then
                    NewDataEntry = True
                Else
                    Sleep (15000)
                End If
            End If
        Loop

        'Reading Raw information from master excel file.
        RVESize = ReadFromAccessTable(TableName, ColumnName, ID, "RVESize",
DatabasePath)
        Displacement = ReadFromAccessTable(TableName, ColumnName, ID,
"MechDisplacment", DatabasePath)
        Temperature1 = ReadFromAccessTable(TableName, ColumnName, ID, "Temp1",
DatabasePath)
        Temperature2 = ReadFromAccessTable(TableName, ColumnName, ID, "Temp2",
DatabasePath)
        VF1 = ReadFromAccessTable(TableName, ColumnName, ID, "VF1",
DatabasePath)
        VF2 = ReadFromAccessTable(TableName, ColumnName, ID, "VF2",
DatabasePath)
```

```
        VF3 = ReadFromAccessTable(TableName, ColumnName, ID, "VF3",
DatabasePath)
        VF4 = ReadFromAccessTable(TableName, ColumnName, ID, "VF4",
DatabasePath)
        VF5 = ReadFromAccessTable(TableName, ColumnName, ID, "VF5",
DatabasePath)
        VF6 = ReadFromAccessTable(TableName, ColumnName, ID, "VF6",
DatabasePath)
        VF7 = ReadFromAccessTable(TableName, ColumnName, ID, "VF7",
DatabasePath)
        VF8 = ReadFromAccessTable(TableName, ColumnName, ID, "VF8",
DatabasePath)
        VF9 = ReadFromAccessTable(TableName, ColumnName, ID, "VF9",
DatabasePath)
        VF10 = ReadFromAccessTable(TableName, ColumnName, ID, "VF10",
DatabasePath)
        VF11 = ReadFromAccessTable(TableName, ColumnName, ID, "VF11",
DatabasePath)
        VF12 = ReadFromAccessTable(TableName, ColumnName, ID, "VF12",
DatabasePath)
        SDX1 = ReadFromAccessTable(TableName, ColumnName, ID,
"SizeDistributionX1", DatabasePath)
        DestPath = ReadFromAccessTable(TableName, ColumnName, ID, "Link",
DatabasePath)

        'Calculating RVE size based on the biggest size distribution and volume
fraction. Ratio 3.2 was used.
        'RVESize = ((4 * Pi * SDX1 ^ 3 * 3.2) / (3 * CurrentVolumeFraction)) ^
(1 / 3)
        'RVESize = Round(RVESize, 2)

        'Setting the project name.
        Call ProgressBar(Progress, 3)
        Forms!frmRunMacro.txtProjectName.Value = ReadFromAccessTable(TableName,
"AnalysisID", ID, "ProjectName", DatabasePath)
        Call ProgressBar(Progress, 3)
        Forms!frmRunMacro.txtProjectStatus.Value = "Start processing volume
fraction: " & CurrentVolumeFraction _
        & " ,Iteration: " & CurrentIteration & "."
        Call ProgressBar(Progress, 3)
        Sleep (100)

        'Setting Project Status.
        Forms!frmRunMacro.txtProjectStatus.Value = "Creating all required
variables."
        Call ProgressBar(Progress, 5)
        Sleep (100)

        'Creating Path variables
        VFPath = "C:\Users\hahmadi\Google Drive\Codes\Project1\VF" &
CurrentVolumeFraction
        GeometriesPath = VFPath & "\Geometries"
        ResultsPath = VFPath & "\Results"
        FinalResultsPath = VFPath & "\FinalResults"
        'Crating Required folders if they don't exist
        Call MakeMyFolder(VFPath)
        Call MakeMyFolder(GeometriesPath)
        Call MakeMyFolder(ResultsPath)
        Call MakeMyFolder(FinalResultsPath)

        'Making Variable for FinalResult excel file name located in VF folder
rooth
        FinalResultName = "VF" & CurrentVolumeFraction & "_FinalResults.xlsm"

        'Making Variable for Final result excel file and making the folder if
does not exist
        FinalResultfile = VFPath & "\" & FinalResultName


        'Creating FinalResult excel file at iteration 1 for new volume fraction
        'If CheckOpenCloseExistOfFile(FinalResultfile) = 53 Then
        If CurrentIteration = 1 Then
            Call NewMacroEnabledExcelFile(VFPath, FinalResultName)
            Call ExcelResultFilePrep(FinalResultfile)
        End If

        'Setting Projetc Status.
        Forms!frmRunMacro.txtProjectStatus.Value = "Writing Current VF and
Iteration to master Excel file."
        Call ProgressBar(Progress, 5)

        'Writing Current VF, Iteration, mechanical status and thermal statuse
in master Excel file$$$$$$$$$
        Call WriteCurrentDataToMasterExcelFile(ExcelPath1,
CurrentVolumeFraction, CurrentIteration, _
        ReadFromAccessTable(TableName, ColumnName, ID, "RVESize",
DatabasePath), _
        ReadFromAccessTable(TableName, ColumnName, ID,
"MechanicalAnalysisCheck", DatabasePath), _
        ReadFromAccessTable(TableName, ColumnName, ID, "ThermalAnalysisCheck",
DatabasePath))

        'Reading Analysis status from access and set analysis status variable
        If ReadFromAccessTable(TableName, ColumnName, ID,
"MechanicalAnalysisCheck", DatabasePath) = -1 Then
            MechanicalCheck = True
        Else
            MechanicalCheck = False
        End If
        If ReadFromAccessTable(TableName, ColumnName, ID,
"ThermalAnalysisCheck", DatabasePath) = -1 Then
            ThermalCheck = True
        Else
            ThermalCheck = False
        End If

        'Setting Projetc Status.
        Forms!frmRunMacro.txtProjectStatus.Value = "Starting Matlab to create
new sets of random numbers."
```

```
        Call ProgressBar(Progress, 5)

        'Starting Matlab to create new sets of random numbers.
        'Call RunMatlab(MatlabEXE)

        'Setting Projetc Status.
        Forms!frmRunMacro.txtProjectStatus.Value = "Checking Random number
excel file is created or not."
        Call ProgressBar(Progress, 5)

        'Checking Random number excel file is created or not.
        Call ExcelFileStatusCheckLoop(GeometriesPath & "\VF" &
CurrentVolumeFraction & "_Geometry_" & CurrentIteration & ".xlsm")

        'Setting Projetc Status.
        Forms!frmRunMacro.txtProjectStatus.Value = "Ansys is perfroming
Mechanical and Thermal analysis."
        Call ProgressBar(Progress, 5)

        'Ansys is perfroming Mechanical and Thermal analysis.
        Call StartEXEWithArgument(AnsysEXE)

        'Setting Projetc Status.
        Forms!frmRunMacro.txtProjectStatus.Value = "Checking Ansys raw results
files are created or not."
        Call ProgressBar(Progress, 5)

        'Checking Ansys raw results files are created or not.
        If MechanicalCheck = True Then
            Call ExcelFileStatusCheckLoop(ResultsPath & "\VF" &
CurrentVolumeFraction & "_MechNormalStrainX_" & CurrentIteration & ".xls")
        End If

        If MechanicalCheck = True Then
            Call ExcelFileStatusCheckLoop(ResultsPath & "\VF" &
CurrentVolumeFraction & "_MechDeformationZ_" & CurrentIteration & ".xls")
        End If

        If ThermalCheck = True Then
            Call ExcelFileStatusCheckLoop(ResultsPath & "\VF" &
CurrentVolumeFraction & "_ThermalHeatFluxX_" & CurrentIteration & ".xls")
        End If

        'Setting Projetc Status.
        Forms!frmRunMacro.txtProjectStatus.Value = "Copping Ansys results files
to FinalResults Folder."
        Call ProgressBar(Progress, 5)

        'Copping Ansys results files to FinalResults Folder.
        If MechanicalCheck = True Then
            Call SaveAsxlsmFileAndClose(ResultsPath, "VF" &
CurrentVolumeFraction & "_MechNormalStress_" & CurrentIteration & ".xls",
FinalResultsPath, _
            "VF" & CurrentVolumeFraction & "_MechNormalStress_" &
CurrentIteration & ".xlsm")
```

```
            Call SaveAsxlsmFileAndClose(ResultsPath, "VF" &
CurrentVolumeFraction & "_MechNormalStrainY_" & CurrentIteration & ".xls",
FinalResultsPath, _
            "VF" & CurrentVolumeFraction & "_MechNormalStrainY_" &
CurrentIteration & ".xlsm")

            Call SaveAsxlsmFileAndClose(ResultsPath, "VF" &
CurrentVolumeFraction & "_MechNormalStrainX_" & CurrentIteration & ".xls",
FinalResultsPath, _
            "VF" & CurrentVolumeFraction & "_MechNormalStrainX_" &
CurrentIteration & ".xlsm")

            Call SaveAsxlsmFileAndClose(ResultsPath, "VF" &
CurrentVolumeFraction & "_MechDeformationZ_" & CurrentIteration & ".xls",
FinalResultsPath, _
            "VF" & CurrentVolumeFraction & "_MechDeformationZ_" &
CurrentIteration & ".xlsm")
        End If

        Call ProgressBar(Progress, 5)

        If ThermalCheck = True Then
            Call SaveAsxlsmFileAndClose(ResultsPath, "VF" &
CurrentVolumeFraction & "_ThermalHeatFluxX_" & CurrentIteration & ".xls",
FinalResultsPath, _
            "VF" & CurrentVolumeFraction & "_ThermalHeatFluxX_" &
CurrentIteration & ".xlsm")
        End If

        'Setting Projetc Status.
        Forms!frmRunMacro.txtProjectStatus.Value = "Modifing Final excel result
and calculating Averaged results."
        Call ProgressBar(Progress, 5)
        Sleep (100)

        'Modifing Final excel results and calculating Averaged results.
        If MechanicalCheck = True Then
            Call PrepairModulusData(FinalResultsPath, "VF" &
CurrentVolumeFraction & "_MechNormalStress_" & CurrentIteration & ".xlsm", _
            "VF" & CurrentVolumeFraction & "_MechNormalStress_" &
CurrentIteration, RVESize, Displacement)

            'This function was obsoleted after I figured out the calculation on
Poisson's ratio is not good.
            'Call PrepairPoissonData(FinalResultsPath, "VF" &
CurrentVolumeFraction & "_MechNormalStrainY_" & CurrentIteration & ".xlsm", _
            "VF" & CurrentVolumeFraction & "_MechNormalStrainY_" &
CurrentIteration, RVESize, Displacement)

            'new function created for Poisson's ratio
            Call PrepairPoissonData1(FinalResultsPath, "VF" &
CurrentVolumeFraction & "_MechNormalStrainY_" & CurrentIteration & ".xlsm", _
```

```
            "VF" & CurrentVolumeFraction & "_MechNormalStrainY_" &
CurrentIteration, "VF" & CurrentVolumeFraction & "_MechNormalStrainX_" &
CurrentIteration & ".xlsm", _
            "VF" & CurrentVolumeFraction & "_MechNormalStrainX_" &
CurrentIteration)

            Call PrepairThermalExpansionData(FinalResultsPath, "VF" &
CurrentVolumeFraction & "_MechDeformationZ_" & CurrentIteration & ".xlsm", _
            "VF" & CurrentVolumeFraction & "_MechDeformationZ_" &
CurrentIteration, RVESize, Temperature2)
        End If

        Call ProgressBar(Progress, 5)

        If ThermalCheck = True Then
            Call PrepairConductivityData(FinalResultsPath, "VF" &
CurrentVolumeFraction & "_ThermalHeatFluxX_" & CurrentIteration & ".xlsm", _
            "VF" & CurrentVolumeFraction & "_ThermalHeatFluxX_" &
CurrentIteration, RVESize, Temperature1, Temperature2)
        End If

        'Setting Projetc Status.
        Forms!frmRunMacro.txtProjectStatus.Value = "Copying Effective Results
to final Results."
        Call ProgressBar(Progress, 5)
        Sleep (100)

        'Copying Effective Results to final Results.
        If MechanicalCheck = True Then
            Call ReadDataFromCloseFile(FinalResultsPath, "VF" &
CurrentVolumeFraction & "_MechNormalStress_" & CurrentIteration & ".xlsm", _
            "VF" & CurrentVolumeFraction & "_MechNormalStress_" &
CurrentIteration, VFPath, "VF" & CurrentVolumeFraction & "_FinalResults.xlsm",
_
            "Sheet1", "F4", "A" & CurrentIteration + 1)

            Call ReadDataFromCloseFile(FinalResultsPath, "VF" &
CurrentVolumeFraction & "_MechNormalStrainY_" & CurrentIteration & ".xlsm", _
            "VF" & CurrentVolumeFraction & "_MechNormalStrainY_" &
CurrentIteration, VFPath, "VF" & CurrentVolumeFraction & "_FinalResults.xlsm",
_
            "Sheet1", "F4", "B" & CurrentIteration + 1)

            Call ReadDataFromCloseFile(FinalResultsPath, "VF" &
CurrentVolumeFraction & "_MechDeformationZ_" & CurrentIteration & ".xlsm", _
            "VF" & CurrentVolumeFraction & "_MechDeformationZ_" &
CurrentIteration, VFPath, "VF" & CurrentVolumeFraction & "_FinalResults.xlsm",
_
            "Sheet1", "F4", "D" & CurrentIteration + 1)
        End If

        Call ProgressBar(Progress, 5)

        If ThermalCheck = True Then
```

```
            Call ReadDataFromCloseFile(FinalResultsPath, "VF" &
CurrentVolumeFraction & "_ThermalHeatFluxX_" & CurrentIteration & ".xlsm", _
            "VF" & CurrentVolumeFraction & "_ThermalHeatFluxX_" &
CurrentIteration, VFPath, "VF" & CurrentVolumeFraction & "_FinalResults.xlsm",
_
            "Sheet1", "F4", "C" & CurrentIteration + 1)
        End If

        Forms!frmRunMacro.txtProjectStatus.Value = "Checking wheather the
criteria is satisfied."
        Call ProgressBar(Progress, 5)
        Sleep (100)

        'Checking wheather the criteria is satisfied
        'For Iteration 1 Since only one value entered in the system
"CurrentMechSTD" manualy set
            'to 1 to avoid error
        If MechanicalCheck = True Then
            If CurrentIteration = 1 Then
                CurrentMechSTD = MechAcceptanceCriteria * 5
            Else
                CurrentMechSTD = ReadingExcelCell(VFPath, FinalResultName,
"Sheet1", "F3")
            End If
        End If
        'Checking Mechanical analysis criteria satisfied.
        If MechanicalCheck = True Then
            If CurrentMechSTD <= MechAcceptanceCriteria Then
            'If CurrentMechSTD <= 200000000 Then
                'change the status of mechanical check to 0 in both access and
master Excel file.
                'Mechanical Check status will be set to False.
                Call WritevalueToExcelCell("C:\Users\hahmadi\Google
Drive\Codes\Project1", "MaterialDatabase.xlsm", _
                "Sheet1", "B36", 0)
                Call WriteToAccessTable(TableName, ColumnName, ID,
"MechanicalAnalysisCheck", 0, DatabasePath)
                MechanicalCheck = False
            End If
        End If
        Call ProgressBar(Progress, 5)

        'For Iteration 1 Since only one value entered in the system
"CurrentThermSTD" manualy set
            'to 1 to avoid error
        If ThermalCheck = True Then
            If CurrentIteration = 1 Then
                CurrentThermSTD = ThermAcceptanceCriteria * 5
            Else
                CurrentThermSTD = ReadingExcelCell(VFPath, FinalResultName,
"Sheet1", "H3")
            End If
        End If
        Call ProgressBar(Progress, 5)
        'Checking Thermal analysis criteria satisfied.
        If ThermalCheck = True Then
            If CurrentThermSTD <= ThermAcceptanceCriteria Then
            'If CurrentThermSTD <= 0.5 Then
                'change the status of mechanical check to 0 in both access and
master Excel file.
                'Thermal Check status will be set to False.
                Call WritevalueToExcelCell("C:\Users\hahmadi\Google
Drive\Codes\Project1", "MaterialDatabase.xlsm", _
                "Sheet1", "B37", 0)
                Call WriteToAccessTable(TableName, ColumnName, ID,
"ThermalAnalysisCheck", 0, DatabasePath)
                ThermalCheck = False
            End If
        End If
        Call ProgressBar(Progress, 6)

        'If both mechanical and thermal analysis are satisfied, reset the new
volume fraction and iteration
        'Reset mechanical and thermal analysis statuse back to -1
        If CurrentIteration = 10 Then
            'Moving Created VF folder to destination path
            'Call Move_Rename_Folder("C:\Users\hahmadi\Google
Drive\Codes\Project1\VF" & CurrentVolumeFraction, DestPath)

            'Writing new VF to Access
            Call WriteToAccessTable(TableName, ColumnName, ID, "VFCompleted",
FindingNextVolumeFraction(VF1, _
            VF2, VF3, VF4, VF5, VF6, VF7, VF8, VF9, VF10, VF11, VF12,
CurrentVolumeFraction), DatabasePath)

            'Writing new Iteration to Access
            Call WriteToAccessTable(TableName, ColumnName, ID, "IterCompleted",
0, DatabasePath)

            'Changing Mechanical Status for new VF if user asked for Mechanical
analysis
            If ReadFromAccessTable(TableName, ColumnName, ID,
"MainMechanicalAnalysisCheck", DatabasePath) = -1 Then
                Call WriteToAccessTable(TableName, ColumnName, ID,
"MechanicalAnalysisCheck", -1, DatabasePath)
            End If

            'Changing Thermal Status for new VF if user asked for Thermal
analysis
            If ReadFromAccessTable(TableName, ColumnName, ID,
"MainThermalAnalysisCheck", DatabasePath) = -1 Then
                Call WriteToAccessTable(TableName, ColumnName, ID,
"ThermalAnalysisCheck", -1, DatabasePath)
            End If
        Else
            Call WriteToAccessTable(TableName, ColumnName, ID, "IterCompleted",
CurrentIteration, DatabasePath)
        End If
        Call ProgressBar(Progress, 5)

        'Checking if the last volume fraction is 0 "Means there is no VF to
work on"
        If ReadFromAccessTable(TableName, ColumnName, ID, "VFCompleted",
DatabasePath) = 0 Then
            'Setting Projetc Status.
            Forms!frmRunMacro.txtProjectStatus.Value = "All volume fractions
are anlayzed."

            'Setting Projetc Status.
            Forms!frmRunMacro.txtProjectStatus.Value = "Macro is sending email
to job initiator."

            'Setting the status to 'Done'.
            Call WriteToAccessTable(TableName, ColumnName, ID,
"AnalysisFinished", "Done!", DatabasePath)
            'Sending Email to the person who initiated the analysis.
            Call SendEmailWithOutlook(ReadFromAccessTable(TableName,
ColumnName, ID, "InitiatorEmail", DatabasePath), "Project " & _
            ReadFromAccessTable(TableName, ColumnName, ID, "ProjectName",
DatabasePath), _
            "<body><p>The Job you submitted is done." & _
            vbNewLine & "<body><p>Please verify the result carefully.")
        End If

        Forms!frmRunMacro.txtProjectStatus.Value = "Volume Fraction " &
CurrentVolumeFraction & " iteration " & _
        CurrentIteration & " completed."
        Call ProgressBar(Progress, 5)
        Sleep (100)

        'Changing the status back regarding new data entered for performing
another loop.
        NewDataEntry = False
        Progress = 0
    Loop
End Sub
Sub WriteCurrentDataToMasterExcelFile(ExcelFilePath, CurrentVF,
CurrentIteration, RVESize, MechCheck, ThermCheck)
    Dim ExcelApp As Object
    Dim ExcelWB As Object
    Dim ExcelSH As Object

    Set ExcelApp = CreateObject("Excel.Application")
    ExcelApp.Visible = False
    Set ExcelWB = ExcelApp.Workbooks.Open(ExcelFilePath)
    Set ExcelSH = ExcelWB.Worksheets(1)

    ExcelSH.Range("B25").Value = CurrentVF
    ExcelSH.Range("B26").Value = CurrentIteration

    ExcelSH.Range("B11").Value = RVESize

    ExcelSH.Range("B36").Value = MechCheck
    ExcelSH.Range("B37").Value = ThermCheck

    ExcelWB.Save
    ExcelWB.Close
    ExcelApp.Quit
    Set ExcelApp = Nothing
End Sub
Function GetDataFromExcel(ExcelFilePath) As settings
    Dim ExcelApp As Object
    Dim ExcelWB As Object
    Dim ExcelSH As Object

    Set ExcelApp = CreateObject("Excel.Application")
    ExcelApp.Visible = False
    Set ExcelWB = ExcelApp.Workbooks.Open(ExcelFilePath)
    Set ExcelSH = ExcelWB.Worksheets(1)

    'Define settings for on the function
    With GetDataFromExcel
        .Data1 = ExcelSH.Range("B49").Value
        .Data2 = ExcelSH.Range("F49").Value
    End With

    ExcelWB.Save
    ExcelWB.Close
    ExcelApp.Quit
    Set ExcelApp = Nothing
End Function
Function CheckNewDataEntry(TableName, ColumnName, ID, DatabasePath)
    Dim DataBase As Object
    Dim Table As Object
    Dim fldEnumerator As Object
    Dim fldColumns As Object
    Set DataBase = OpenDatabase(DatabasePath)
    Set Table = DataBase.OpenRecordset(TableName, dbOpenTable)
    Set fldColumns = Table.Fields
    'Setting pre condition for Function.
    CheckNewDataEntry = False
    ' Scan the records from beginning to each.
    Do While Not Table.EOF
        ' Checking column and row required.
        For Each fldEnumerator In Table.Fields
            ' If the column is named whatever.
            If fldEnumerator.Name = ColumnName Then
                ' If the title of the current record is whatever.
                If fldEnumerator.Value = ID Then
                    ' then change its value.
                    CheckNewDataEntry = True
                    Exit Do
                End If
            End If
        Next
        ' Move to the next record and continue the same approach.
        Table.MoveNext
    Loop
    Table.Close
    DataBase.Close
End Function
Function ReadFromAccessTable(TableName, ColumnName, ID, FieldName,
DatabasePath)
    Dim DataBase As Object
    Dim Table As Object
    Dim fldEnumerator As Object
    Dim fldColumns As Object
    Set DataBase = OpenDatabase(DatabasePath)
    Set Table = DataBase.OpenRecordset(TableName, dbOpenTable)
    Set fldColumns = Table.Fields
    ' Scan the records from beginning to each.
    Do While Not Table.EOF
        ' Checking column and row required.
        For Each fldEnumerator In Table.Fields
            ' If the column is named whatever.
            If fldEnumerator.Name = ColumnName Then
                ' If the title of the current record is whatever.
                If fldEnumerator.Value = ID Then
                    ' Read the record from table.
                    ReadFromAccessTable = Table(FieldName).Value
                    Exit Do
                End If
            End If
        Next
        ' Move to the next record and continue the same approach.
        Table.MoveNext
    Loop
    Table.Close
    DataBase.Close
End Function
Sub WriteToAccessTable(TableName, ColumnName, ID, FieldName, Record,
DatabasePath)
    Dim DataBase As Object
    Dim Table As Object
    Dim fldEnumerator As Object
    Dim fldColumns As Object

    ' Turn error checking off.
    On Error Resume Next
    Set DataBase = OpenDatabase(DatabasePath)
    Set Table = DataBase.OpenRecordset(TableName, dbOpenTable)
    Set fldColumns = Table.Fields
    ' Scan the records from beginning to each.
    Do While Not Table.EOF
        ' Checking column and row required.
        For Each fldEnumerator In Table.Fields
            ' If the column is named whatever.
            If fldEnumerator.Name = ColumnName Then
                ' If the title of the current record is whatever.
                If fldEnumerator.Value = ID Then
                    ' then change its value.
                    Table.Edit
                    Table(FieldName).Value = Record
                    Table.Update
                    ' Table refresh.
                    DataBase.TableDefs.Refresh
                    Exit Do
                End If
            End If
        Next
        ' Move to the next record and continue the same approach.
        Table.MoveNext
    Loop
    Table.Close
    DataBase.Close
End Sub
Sub ExcelFileStatusCheckLoop(ExcelFilePath)
    Dim ExcelStatus As Boolean
    Do While ExcelStatus = False
        ' Test to see if the file is open.
        If IsExcelFileOpen(ExcelFilePath) Then
            ' Checking Frequency in Milli Seconds.
            Sleep (5000)
        Else
            ' Stoping the loop
            ExcelStatus = True
        End If
    Loop
End Sub
Function IsExcelFileOpen(ExcelFilePath)
    Dim filenum As Integer
    Dim errnum As Integer

    On Error Resume Next   ' Turn error checking off.
    filenum = FreeFile()   ' Get a free file number.
    ' Attempt to open the file and lock it.
    Open ExcelFilePath For Input Lock Read As #filenum
    Close (filenum)        ' Close the file.
    errnum = Err           ' Save the error number that occurred.
    On Error GoTo 0        ' Turn error checking back on.

    ' Check to see which error occurred.
    Select Case errnum

        ' No error occurred.
        ' File is NOT already open by another user.
        Case 0
            IsExcelFileOpen = False

            ' Error number for "Permission Denied."
            ' File is already opened by another user.
        Case 70
            IsExcelFileOpen = True

            'If the file does not exist.
        Case 53
            IsExcelFileOpen = True
            ' Another error occurred.
        Case Else
            Error errnum
    End Select
End Function
Sub StartEXEWithArgument(ExeFilePath)
    Dim strProgramName As String
    Dim strArgument As String

    strProgramName = ExeFilePath
    strArgument = "/G"

    Call Shell("""" & strProgramName & """ """ & strArgument & """",
vbNormalFocus)
End Sub
Function SendEmailWithOutlook(MessageTo, Subject, Hyperlink)

    ' Define app variable and get Outlook using the "New" keyword
    Dim olApp As Object
    Dim olMail As Object  ' An Outlook Mail item

    Set olApp = GetObject(, "Outlook.Application")
        If Err Then
        ' Create a new email object
        Set olApp = CreateObject("Outlook.Application")
        End If

    Set olMail = olApp.CreateItem(olMailItem)

    ' Add the To/Subject/Body to the message and display the message
    With olMail
        .To = MessageTo
        .Subject = Subject
        .HTMLBody = Hyperlink
        .Send       ' Send the message immediately
    End With

    'Release all object variables.
    Set olMailItem = Nothing
    Set olApp = Nothing

End Function
Function GettingLastRecord(TableName, ColumnName, DatabasePath)
    Dim DataBase As Object
    Dim Table As Object

    Set DataBase = OpenDatabase(DatabasePath)
    Set Table = DataBase.OpenRecordset(TableName, dbOpenTable)

    Table.MoveLast
    GettingLastRecord = Table.Fields(ColumnName)

    Table.Close
    DataBase.Close
End Function
```

```
Function GettingLastCompletedAnalysisRecord(TableName, ColumnName,
DatabasePath)
    Dim DataBase As Object
    Dim Table As Object
    Dim fldEnumerator As Object
    Dim fldColumns As Object
    Dim NUM As Long
    Set DataBase = OpenDatabase(DatabasePath)
    Set Table = DataBase.OpenRecordset(TableName, dbOpenTable)
    Set fldColumns = Table.Fields
    NUM = 0
    ' Scan the records from beginning to each.
    Do While Not Table.EOF
        ' Checking column and row required.
        For Each fldEnumerator In Table.Fields
            ' If the column is named whatever.
            If fldEnumerator.Name = ColumnName Then
                ' Checking the status if it is "Done!" or not
                If Table(ColumnName).Value = "Done!" Then
                    ' Read the record from table.
                    NUM = NUM + 1
                    GettingLastCompletedAnalysisRecord = NUM
                    'Exit Do
                End If
            End If
        Next
        ' Move to the next record and continue the same approach.
        Table.MoveNext
    Loop
    Table.Close
    DataBase.Close
End Function
Sub ClosingExcelFileIfRequired(ExcelFilePath)
    Dim ExcelStatus As Boolean
    Do While ExcelStatus = False
        ' Test to see if the file is open.
        If IsExcelFileOpen(ExcelFilePath) Then
            ' Prompt to close the Excel file
            closeexcel = MsgBox("Please verify Required Excel file is closed
and press OK." & vbNewLine & _
            "If you want to terminate the macro press Cancel.", 1, "Checking
ANSYS Excel file")
            If closeexcel = 2 Then
                End
            End If
        Else
            ' Stoping the loop
            ExcelStatus = True
        End If
    Loop
End Sub
Function ProgressBar(Progress, Inc)
    Dim i As Integer
    For i = Progress To Progress + Inc Step 1
    Forms!frmRunMacro.boxProgressBar.Width = "" & 21.6 * i & ""
    Forms!frmRunMacro.lblProgressPercentage.Caption = "" & i & "%"
    DoEvents
    Sleep (350)
    Next i
    Progress = Progress + Inc
End Function
Sub MakeMyFolder(FolderPath)
    Dim fdObj As Object


    Set fdObj = CreateObject("Scripting.FileSystemObject")
    If fdObj.FolderExists(FolderPath) Then

    Else
        fdObj.CreateFolder (FolderPath)
    End If

End Sub
Function CheckOpenCloseExistOfFile(ExcelFilePath)
    Dim filenum As Integer
    Dim errnum As Integer

    On Error Resume Next    ' Turn error checking off.
    filenum = FreeFile()    ' Get a free file number.
    ' Attempt to open the file and lock it.
    Open ExcelFilePath For Input Lock Read As #filenum
    Close (filenum)         ' Close the file.
    errnum = Err            ' Save the error number that occurred.
    On Error GoTo 0         ' Turn error checking back on.

    ' Check to see which error occurred.
    Select Case errnum

        ' No error occurred.
        ' File is NOT already open by another user.
        Case 0
            CheckOpenCloseExistOfFile = 0

            ' Error number for "Permission Denied."
            ' File is already opened by another user.
        Case 70
            CheckOpenCloseExistOfFile = 70

            'If the file does not exist.
        Case 53
            CheckOpenCloseExistOfFile = 53

            ' Another error occurred.
        Case Else
            Error errnum
    End Select
End Function
Sub NewMacroEnabledExcelFile(Path, FileName)
    'Path = "C:\Users\hahmadi\Google Drive\whatever"
    'FileName = "Test1.xlsm"
    Dim app As New Excel.Application
    Dim NewExcel As Excel.Workbook

    app.Visible = False
    app.ScreenUpdating = False

    Set NewExcel = Workbooks.Add
    'Application.DisplayAlerts = False
    NewExcel.SaveAs FileName:=Path & "\" & FileName, _
        FileFormat:=xlOpenXMLWorkbookMacroEnabled, _
        Password:="", _
        WriteResPassword:="", _
        ReadOnlyRecommended:=False, _
        CreateBackup:=False

    NewExcel.Close
    app.Quit
    Set app = Nothing
End Sub
Sub ExcelResultFilePrep(ExcelFilePath)
    Dim ExcelApp As New Excel.Application
    Dim ExcelWB As Excel.Workbook
    Dim ExcelSH As Excel.Sheets


    ExcelApp.Visible = False
    Workbooks.Open (ExcelFilePath)


    Range("A1").Select
    ActiveCell.FormulaR1C1 = "Effective Modulus of Elasticity"
    Range("B1").Select
    Columns("A:A").EntireColumn.AutoFit
    ActiveCell.FormulaR1C1 = "Effective Poisson Ratio"
    Range("C1").Select
    Columns("B:B").EntireColumn.AutoFit
    ActiveCell.FormulaR1C1 = "Effective Thermal Conductivity"
    Range("D1").Select
    Columns("C:C").EntireColumn.AutoFit
    ActiveCell.FormulaR1C1 = "Effective Thermal Expansion"
    Columns("D:D").EntireColumn.AutoFit
    Range("E3").Select
    ActiveCell.FormulaR1C1 = "Standard Deviation"
    Range("E4").Select
    ActiveCell.FormulaR1C1 = "Variance"
    Range("E5").Select
    ActiveCell.FormulaR1C1 = "Average"
    Range("F2").Select
    Columns("E:E").EntireColumn.AutoFit
    ActiveCell.FormulaR1C1 = "Modulus of Elasticity"
    Range("G2").Select
    Columns("F:F").EntireColumn.AutoFit
    ActiveCell.FormulaR1C1 = "Poisson Ratio"
    Range("H2").Select
    Columns("G:G").EntireColumn.AutoFit
    ActiveCell.FormulaR1C1 = "Thermal Conductivity"
    Range("I2").Select
    Columns("H:H").EntireColumn.AutoFit
    ActiveCell.FormulaR1C1 = "Thermal Expansion"
    Columns("I:I").EntireColumn.AutoFit
    Range("E2:I5").Select
    Selection.Borders(xlDiagonalDown).LineStyle = xlNone
    Selection.Borders(xlDiagonalUp).LineStyle = xlNone
    With Selection.Borders(xlEdgeLeft)
        .LineStyle = xlContinuous
        .ColorIndex = 0
        .TintAndShade = 0
        .Weight = xlThin
    End With
    With Selection.Borders(xlEdgeTop)
        .LineStyle = xlContinuous
        .ColorIndex = 0
        .TintAndShade = 0
        .Weight = xlThin
    End With
    With Selection.Borders(xlEdgeBottom)
        .LineStyle = xlContinuous
        .ColorIndex = 0
        .TintAndShade = 0
        .Weight = xlThin
    End With
    With Selection.Borders(xlEdgeRight)
        .LineStyle = xlContinuous
        .ColorIndex = 0
        .TintAndShade = 0
        .Weight = xlThin
    End With
    With Selection.Borders(xlInsideVertical)
        .LineStyle = xlContinuous
        .ColorIndex = 0
        .TintAndShade = 0
        .Weight = xlThin
    End With
    With Selection.Borders(xlInsideHorizontal)
        .LineStyle = xlContinuous
        .ColorIndex = 0
        .TintAndShade = 0
        .Weight = xlThin
    End With

    'Creating Dynamic Tables
    'Mechanical Dynamic Table and set standard deviation, variance and average
function
    ActiveWorkbook.Names.Add Name:="MechanicalDynamicTable", RefersToR1C1:= _
        "=OFFSET('Sheet1'!R2C1,,,COUNTA('Sheet1'!C1)-1)"
    ActiveWorkbook.Names("MechanicalDynamicTable").Comment = ""
    Range("F3").Select
    ActiveCell.FormulaR1C1 = "=STDEV.S(MechanicalDynamicTable)"
    Range("F4").Select
    ActiveCell.FormulaR1C1 = "=VAR.S(MechanicalDynamicTable)"
    Range("F5").Select
    ActiveCell.FormulaR1C1 = "=AVERAGE(MechanicalDynamicTable)"

    'Poisson Dynamic Table and set standard deviation, variance and average
function
    ActiveWorkbook.Names.Add Name:="PoissonDynamicTable", RefersToR1C1:= _
        "=OFFSET('Sheet1'!R2C2,,,COUNTA('Sheet1'!C2)-1)"
    ActiveWorkbook.Names("PoissonDynamicTable").Comment = ""
    Range("G3").Select
    ActiveCell.FormulaR1C1 = "=STDEV.S(PoissonDynamicTable)"
    Range("G4").Select
    ActiveCell.FormulaR1C1 = "=VAR.S(PoissonDynamicTable)"
    Range("G5").Select
    ActiveCell.FormulaR1C1 = "=AVERAGE(PoissonDynamicTable)"


    'Thermal Dynamic Table and set standard deviation, variance and average
function
    ActiveWorkbook.Names.Add Name:="ThermalDynamicTable", RefersToR1C1:= _
        "=OFFSET('Sheet1'!R2C3,,,COUNTA('Sheet1'!C3)-1)"
    ActiveWorkbook.Names("ThermalDynamicTable").Comment = ""
    Range("H3").Select
    ActiveCell.FormulaR1C1 = "=STDEV.S(ThermalDynamicTable)"
    Range("H4").Select
    ActiveCell.FormulaR1C1 = "=VAR.S(ThermalDynamicTable)"
    Range("H5").Select
    ActiveCell.FormulaR1C1 = "=AVERAGE(ThermalDynamicTable)"

    'Thermal Expansion Dynamic Table and set standard deviation, variance and
average function
    ActiveWorkbook.Names.Add Name:="ThermalExpansionDynamicTable",
RefersToR1C1:= _
        "=OFFSET('Sheet1'!R2C4,,,COUNTA('Sheet1'!C4)-1)"
    ActiveWorkbook.Names("ThermalExpansionDynamicTable").Comment = ""
    Range("I3").Select
    ActiveCell.FormulaR1C1 = "=STDEV.S(ThermalExpansionDynamicTable)"
    Range("I4").Select
    ActiveCell.FormulaR1C1 = "=VAR.S(ThermalExpansionDynamicTable)"
    Range("I5").Select
    ActiveCell.FormulaR1C1 = "=AVERAGE(ThermalExpansionDynamicTable)"

    ActiveWorkbook.Save
    ActiveWorkbook.Close
    ExcelApp.Quit
    Set ExcelApp = Nothing
End Sub
Sub SaveAsxlsmFileAndClose(OldFilePath, OldFileName, NewFilePath, NewFileName)
    'Samples
    'OldFilePath = "C:\Users\hahmadi\Desktop\TT"
    'OldFileName = "1_Equivalent Stress.xls"
    'NewFilePath = "C:\Users\hahmadi\Desktop\TT"
    'NewFileName = "hh.xlsm"

    Dim ExcelApp As New Excel.Application
    ExcelApp.Visible = False

    Workbooks.Open (OldFilePath & "\" & OldFileName)


    ActiveWorkbook.SaveAs FileName:=NewFilePath & "\" & NewFileName, _
        FileFormat:=xlOpenXMLWorkbookMacroEnabled, _
        Password:="", _
        WriteResPassword:="", _
        ReadOnlyRecommended:=False, _
        CreateBackup:=False

    ActiveWorkbook.Save
    ActiveWorkbook.Close
    ExcelApp.Quit
    Set ExcelApp = Nothing
End Sub
Sub PrepairModulusData(FilePath, FileName, SheetName, RVESize, Displacement)
    Dim ExcelApp As New Excel.Application
    Dim ExcelWB As Workbooks
    Dim ExcelSH As Worksheet
    Dim LastRow As Long
    Dim LastColumn As Long
    Dim StartCell As Range
    Dim objTable As ListObject


    ExcelApp.Visible = False
    Workbooks.Open (FilePath & "\" & FileName)
    Set ExcelSH = ActiveWorkbook.Worksheets(SheetName)

    Set StartCell = Range("A1")

    Columns("A:A").EntireColumn.AutoFit
    Columns("B:B").EntireColumn.AutoFit
    Range("C1").Select
    ActiveCell.FormulaR1C1 = "Modulus of Elasticity"
    Range("D1").Select
    Columns("C:C").EntireColumn.AutoFit

    'Find Last Row and Column
    LastRow = ExcelSH.Cells(ExcelSH.Rows.Count, StartCell.Column).End(xlUp).Row
    LastColumn = ExcelSH.Cells(StartCell.Row,
ExcelSH.Columns.Count).End(xlToLeft).Column
    'Select Range
    ExcelSH.Range(StartCell, ExcelSH.Cells(LastRow, LastColumn)).Select
    'Inserting Table
    Set objTable = ActiveSheet.ListObjects.Add(xlSrcRange, Selection, , xlYes)
    objTable.TableStyle = "TableStyleMedium2"

    Range("E1").Select
    ActiveCell.FormulaR1C1 = "RVESize"
    Range("E2").Select
    ActiveCell.FormulaR1C1 = "Displacement"
    Range("E3").Select
```

```
ActiveCell.FormulaR1C1 = "Strain"
Range("E4").Select
ActiveCell.FormulaR1C1 = "Effective Modulus of Elasticity"
Range("F1").Select
ActiveCell.FormulaR1C1 = RVESize
Range("F2").Select
ActiveCell.FormulaR1C1 = Displacement
Range("F3").Select
ActiveCell.FormulaR1C1 = "=LN((R[-2]C+R[-1]C)/R[-2]C)"
Range("F4").Select
Columns("E:E").EntireColumn.AutoFit
Range("C2").Select
ActiveCell.FormulaR1C1 = "=[@[Normal Stress (Pa)]]/R3C6"
Range("C3").Select

'Creating required border lines
Range("E1:F4").Select
Selection.Borders(xlDiagonalDown).LineStyle = xlNone
Selection.Borders(xlDiagonalUp).LineStyle = xlNone
With Selection.Borders(xlEdgeLeft)
    .LineStyle = xlContinuous
    .ColorIndex = 0
    .TintAndShade = 0
    .Weight = xlThin
End With
With Selection.Borders(xlEdgeTop)
    .LineStyle = xlContinuous
    .ColorIndex = 0
    .TintAndShade = 0
    .Weight = xlThin
End With
With Selection.Borders(xlEdgeBottom)
    .LineStyle = xlContinuous
    .ColorIndex = 0
    .TintAndShade = 0
    .Weight = xlThin
End With
With Selection.Borders(xlEdgeRight)
    .LineStyle = xlContinuous
    .ColorIndex = 0
    .TintAndShade = 0
    .Weight = xlThin
End With
With Selection.Borders(xlInsideVertical)
    .LineStyle = xlContinuous
    .ColorIndex = 0
    .TintAndShade = 0
    .Weight = xlThin
End With
With Selection.Borders(xlInsideHorizontal)
    .LineStyle = xlContinuous
    .ColorIndex = 0
    .TintAndShade = 0
    .Weight = xlThin
End With
```

```
    'Dynamic Table and set Average function
    ActiveWorkbook.Names.Add Name:="AverageDynamicTable", RefersToR1C1:= _
        "=OFFSET('" & SheetName & "'!R2C3,,,COUNTA('" & SheetName & "'!C3)-1)"
    ActiveWorkbook.Names("AverageDynamicTable").Comment = ""
    Range("F4").Select
    ActiveCell.FormulaR1C1 = "=AVERAGE(AverageDynamicTable)"

    ActiveWorkbook.Save
    ActiveWorkbook.Close
    ExcelApp.Quit
    Set ExcelApp = Nothing
End Sub
Function ReadingDataFromExcel(ExcelFilePath, CellName)
    Dim ExcelApp As Object
    Dim ExcelWB As Object
    Dim ExcelSH As Object

    Set ExcelApp = CreateObject("Excel.Application")
    ExcelApp.Visible = False
    Set ExcelWB = ExcelApp.Workbooks.Open(ExcelFilePath)
    Set ExcelSH = ExcelWB.Worksheets(1)

    'Define settings for on the function
    ReadingDataFromExcel = ExcelSH.Range(CellName).Value

    ExcelWB.Save
    ExcelWB.Close
    ExcelApp.Quit
    Set ExcelApp = Nothing

End Function
Sub PrepairPoissonData(FilePath, FileName, SheetName, RVESize, Displacement)
    Dim ExcelApp As New Excel.Application
    Dim ExcelWB As Workbooks
    Dim ExcelSH As Worksheet
    Dim LastRow As Long
    Dim LastColumn As Long
    Dim StartCell As Range
    Dim objTable As ListObject


    ExcelApp.Visible = False
    Workbooks.Open (FilePath & "\" & FileName)
    Set ExcelSH = ActiveWorkbook.Worksheets(SheetName)

    Set StartCell = Range("A1")

    Columns("A:A").EntireColumn.AutoFit
    Columns("B:B").EntireColumn.AutoFit
    Range("C1").Select
    ActiveCell.FormulaR1C1 = "Poisson Ratio"
    Range("D1").Select
    Columns("C:C").EntireColumn.AutoFit
    'Find Last Row and Column
    LastRow = ExcelSH.Cells(ExcelSH.Rows.Count, StartCell.Column).End(xlUp).Row
    LastColumn = ExcelSH.Cells(StartCell.Row,
ExcelSH.Columns.Count).End(xlToLeft).Column
    'Select Range
    ExcelSH.Range(StartCell, ExcelSH.Cells(LastRow, LastColumn)).Select
    'Inserting Table
    Set objTable = ActiveSheet.ListObjects.Add(xlSrcRange, Selection, , xlYes)
    objTable.TableStyle = "TableStyleMedium2"

    Range("E1").Select
    ActiveCell.FormulaR1C1 = "RVESize"
    Range("E2").Select
    ActiveCell.FormulaR1C1 = "Displacement"
    Range("E3").Select
    ActiveCell.FormulaR1C1 = "Strain"
    Range("E4").Select
    ActiveCell.FormulaR1C1 = "Effective Poisson Ratio"
    Range("F1").Select
    ActiveCell.FormulaR1C1 = RVESize
    Range("F2").Select
    ActiveCell.FormulaR1C1 = Displacement
    Range("F3").Select
    ActiveCell.FormulaR1C1 = "=LN((R[-2]C+R[-1]C)/R[-2]C)"
    Range("F4").Select
    Columns("E:E").EntireColumn.AutoFit
    Range("C2").Select
    ActiveCell.FormulaR1C1 = "=[@[Normal Elastic Strain (m/m)]]/(R3C6*(-1))"
    Range("C3").Select

    'Creating required border lines
    Range("E1:F4").Select
    Selection.Borders(xlDiagonalDown).LineStyle = xlNone
    Selection.Borders(xlDiagonalUp).LineStyle = xlNone
    With Selection.Borders(xlEdgeLeft)
        .LineStyle = xlContinuous
        .ColorIndex = 0
        .TintAndShade = 0
        .Weight = xlThin
    End With
    With Selection.Borders(xlEdgeTop)
        .LineStyle = xlContinuous
        .ColorIndex = 0
        .TintAndShade = 0
        .Weight = xlThin
    End With
    With Selection.Borders(xlEdgeBottom)
        .LineStyle = xlContinuous
        .ColorIndex = 0
        .TintAndShade = 0
        .Weight = xlThin
    End With
    With Selection.Borders(xlEdgeRight)
        .LineStyle = xlContinuous
        .ColorIndex = 0
        .TintAndShade = 0
        .Weight = xlThin
    End With
    With Selection.Borders(xlInsideVertical)
        .LineStyle = xlContinuous
        .ColorIndex = 0
        .TintAndShade = 0
        .Weight = xlThin
    End With
    With Selection.Borders(xlInsideHorizontal)
        .LineStyle = xlContinuous
        .ColorIndex = 0
        .TintAndShade = 0
        .Weight = xlThin
    End With

    'Dynamic Table and set Average function
    ActiveWorkbook.Names.Add Name:="AverageDynamicTable", RefersToR1C1:= _
        "=OFFSET('" & SheetName & "'!R2C3,,,COUNTA('" & SheetName & "'!C3)-1)"
    ActiveWorkbook.Names("AverageDynamicTable").Comment = ""
    Range("F4").Select
    ActiveCell.FormulaR1C1 = "=AVERAGE(AverageDynamicTable)"

    ActiveWorkbook.Save
    ActiveWorkbook.Close
    ExcelApp.Quit
    Set ExcelApp = Nothing
End Sub
Sub PrepairThermalExpansionData(FilePath, FileName, SheetName, RVESize,
Temperature2)
    Dim ExcelApp As New Excel.Application
    Dim ExcelWB As Workbooks
    Dim ExcelSH As Worksheet
    Dim LastRow As Long
    Dim LastColumn As Long
    Dim StartCell As Range
    Dim objTable As ListObject


    ExcelApp.Visible = False
    Workbooks.Open (FilePath & "\" & FileName)
    Set ExcelSH = ActiveWorkbook.Worksheets(SheetName)

    Set StartCell = Range("A1")

    Columns("A:A").EntireColumn.AutoFit
    Columns("B:B").EntireColumn.AutoFit
    Range("C1").Select
    ActiveCell.FormulaR1C1 = "Thermal Expansion"
    Range("D1").Select
    Columns("C:C").EntireColumn.AutoFit

    'Find Last Row and Column
    LastRow = ExcelSH.Cells(ExcelSH.Rows.Count, StartCell.Column).End(xlUp).Row
    LastColumn = ExcelSH.Cells(StartCell.Row,
ExcelSH.Columns.Count).End(xlToLeft).Column
    'Select Range
    ExcelSH.Range(StartCell, ExcelSH.Cells(LastRow, LastColumn)).Select
    'Inserting Table
    Set objTable = ActiveSheet.ListObjects.Add(xlSrcRange, Selection, , xlYes)
    objTable.TableStyle = "TableStyleMedium2"

    Range("E1").Select
    ActiveCell.FormulaR1C1 = "RVESize"
    Range("E2").Select
    ActiveCell.FormulaR1C1 = "Temperature2"
    Range("E3").Select
    ActiveCell.FormulaR1C1 = "Thermal Expansion Ratio"
    Range("E4").Select
    ActiveCell.FormulaR1C1 = "Effective Thermal Expansion"
    Range("F1").Select
    ActiveCell.FormulaR1C1 = RVESize
    Range("F2").Select
    ActiveCell.FormulaR1C1 = Temperature2
    Range("F3").Select
    ActiveCell.FormulaR1C1 = "=1/(R[-2]C*(R[-1]C-22))"
    Range("F4").Select
    Columns("E:E").EntireColumn.AutoFit
    Range("C2").Select
    ActiveCell.FormulaR1C1 = "=[@[Directional Deformation (m)]]*R3C6"
    Range("C3").Select

    'Creating required border lines
    Range("E1:F4").Select
    Selection.Borders(xlDiagonalDown).LineStyle = xlNone
    Selection.Borders(xlDiagonalUp).LineStyle = xlNone
    With Selection.Borders(xlEdgeLeft)
        .LineStyle = xlContinuous
        .ColorIndex = 0
        .TintAndShade = 0
        .Weight = xlThin
    End With
    With Selection.Borders(xlEdgeTop)
        .LineStyle = xlContinuous
        .ColorIndex = 0
        .TintAndShade = 0
        .Weight = xlThin
    End With
    With Selection.Borders(xlEdgeBottom)
        .LineStyle = xlContinuous
        .ColorIndex = 0
        .TintAndShade = 0
        .Weight = xlThin
    End With
    With Selection.Borders(xlEdgeRight)
        .LineStyle = xlContinuous
        .ColorIndex = 0
        .TintAndShade = 0
        .Weight = xlThin
    End With
    With Selection.Borders(xlInsideVertical)
        .LineStyle = xlContinuous
        .ColorIndex = 0
        .TintAndShade = 0
        .Weight = xlThin
    End With
    With Selection.Borders(xlInsideHorizontal)
        .LineStyle = xlContinuous
        .ColorIndex = 0
        .TintAndShade = 0
        .Weight = xlThin
    End With

    'Dynamic Table and set Average function
    ActiveWorkbook.Names.Add Name:="AverageDynamicTable", RefersToR1C1:= _
        "=OFFSET('" & SheetName & "'!R2C3,,,COUNTA('" & SheetName & "'!C3)-1)"
    ActiveWorkbook.Names("AverageDynamicTable").Comment = ""
    Range("F4").Select
    ActiveCell.FormulaR1C1 = "=AVERAGE(AverageDynamicTable)"

    ActiveWorkbook.Save
    ActiveWorkbook.Close
    ExcelApp.Quit
    Set ExcelApp = Nothing
End Sub
Sub PrepairConductivityData(FilePath, FileName, SheetName, RVESize,
Temperature1, Temperature2)
    Dim ExcelApp As New Excel.Application
    Dim ExcelWB As Workbooks
    Dim ExcelSH As Worksheet
    Dim LastRow As Long
    Dim LastColumn As Long
    Dim StartCell As Range
    Dim objTable As ListObject


    ExcelApp.Visible = False
    Workbooks.Open (FilePath & "\" & FileName)
    Set ExcelSH = ActiveWorkbook.Worksheets(SheetName)

    Set StartCell = Range("A1")

    Range("B1").Select
    ActiveCell.FormulaR1C1 = "Directional Heat Flux (W/m^2)"
    Columns("A:A").EntireColumn.AutoFit
    Columns("B:B").EntireColumn.AutoFit
    Range("C1").Select
    ActiveCell.FormulaR1C1 = "Thermal Conductivity"
    Range("D1").Select
    Columns("C:C").EntireColumn.AutoFit

    'Find Last Row and Column
    LastRow = ExcelSH.Cells(ExcelSH.Rows.Count, StartCell.Column).End(xlUp).Row
    LastColumn = ExcelSH.Cells(StartCell.Row,
ExcelSH.Columns.Count).End(xlToLeft).Column
    'Select Range
    ExcelSH.Range(StartCell, ExcelSH.Cells(LastRow, LastColumn)).Select
    'Inserting Table
    Set objTable = ActiveSheet.ListObjects.Add(xlSrcRange, Selection, , xlYes)
    objTable.TableStyle = "TableStyleMedium2"

    Range("E1").Select
    ActiveCell.FormulaR1C1 = "RVESize"
    Range("E2").Select
    ActiveCell.FormulaR1C1 = "Temperature 1"
    Range("E3").Select
    ActiveCell.FormulaR1C1 = "Temperature 2"
    Range("E4").Select
    ActiveCell.FormulaR1C1 = "Effective Thermal Conductivity"
    Range("F1").Select
    ActiveCell.FormulaR1C1 = RVESize
    Range("F2").Select
    ActiveCell.FormulaR1C1 = Temperature1
    Range("F3").Select
    ActiveCell.FormulaR1C1 = Temperature2
    Range("F4").Select
    Columns("E:E").EntireColumn.AutoFit
    Range("C2").Select
    ActiveCell.FormulaR1C1 = "=([@[Directional Heat Flux (W/m^2)]]*R1C6*(-
1))/(R3C6-R2C6)"
    Range("C3").Select

    'Creating required border lines
    Range("E1:F4").Select
    Selection.Borders(xlDiagonalDown).LineStyle = xlNone
    Selection.Borders(xlDiagonalUp).LineStyle = xlNone
    With Selection.Borders(xlEdgeLeft)
        .LineStyle = xlContinuous
        .ColorIndex = 0
        .TintAndShade = 0
        .Weight = xlThin
    End With
    With Selection.Borders(xlEdgeTop)
        .LineStyle = xlContinuous
        .ColorIndex = 0
        .TintAndShade = 0
        .Weight = xlThin
    End With
    With Selection.Borders(xlEdgeBottom)
        .LineStyle = xlContinuous
        .ColorIndex = 0
        .TintAndShade = 0
        .Weight = xlThin
    End With
    With Selection.Borders(xlEdgeRight)
        .LineStyle = xlContinuous
        .ColorIndex = 0
        .TintAndShade = 0
        .Weight = xlThin
    End With
    With Selection.Borders(xlInsideVertical)
        .LineStyle = xlContinuous
        .ColorIndex = 0
        .TintAndShade = 0
        .Weight = xlThin
    End With
    With Selection.Borders(xlInsideHorizontal)
        .LineStyle = xlContinuous
        .ColorIndex = 0
        .TintAndShade = 0
        .Weight = xlThin
    End With

    'Dynamic Table and set Average function
    ActiveWorkbook.Names.Add Name:="AverageDynamicTable", RefersToR1C1:= _
        "=OFFSET('" & SheetName & "'!R2C3,,,COUNTA('" & SheetName & "'!C3)-1)"
    ActiveWorkbook.Names("AverageDynamicTable").Comment = ""
    Range("F4").Select
    ActiveCell.FormulaR1C1 = "=AVERAGE(AverageDynamicTable)"

    ActiveWorkbook.Save
    ActiveWorkbook.Close
    ExcelApp.Quit
    Set ExcelApp = Nothing
End Sub
Sub ReadDataFromCloseFile(srcPath, srcFileName, srcSheet, DestPath,
destFilename, destsheet, srcCell, destCell)
    'srcPath = "C:\Users\hahmadi\Desktop\TT"
    'srcFileName = "2_Equivalent Elastic Strain.xls"
    'srcSheet = "2_Equivalent Elastic Strain"
    'destPath = "C:\Users\hahmadi\Desktop\TT"
    'destFilename = "1_Equivalent Stress.xls"
    'destsheet = "1_Equivalent Stress"
    'srcCell = "B2"
    'destCell = "D2"

    Dim ExcelApp As New Excel.Application
    Dim src As Workbooks
    Dim ExcelSH As Worksheet
    Dim dest As Workbooks
    Dim Result As Double

    ExcelApp.Visible = False


    ' OPEN THE SOURCE EXCEL WORKBOOK IN AND READ INFORMATION FROM A CELL.
    Workbooks.Open (srcPath & "\" & srcFileName)

    Set ExcelSH = ActiveWorkbook.Worksheets(srcSheet)

    Result = ExcelSH.Range(srcCell).Value

    ' CLOSE THE SOURCE FILE.
    ActiveWorkbook.Save
    ActiveWorkbook.Close


    'Open the destination file and WRITE THE INFROMATION IN A CELL
    ExcelApp.Visible = False
    Workbooks.Open (DestPath & "\" & destFilename)
    Set ExcelSH = ActiveWorkbook.Worksheets(destsheet)

    ' COPY DATA FROM SOURCE (CLOSE WORKGROUP) TO THE DESTINATION WORKBOOK.
    ExcelSH.Range(destCell).Value = Result


    'Saving the destination file and close it.
    ActiveWorkbook.Save
    ActiveWorkbook.Close
    ExcelApp.Quit
    Set ExcelApp = Nothing

End Sub
Function ReadingExcelCell(srcPath, srcFileName, srcSheet, srcCell)
    Dim ExcelApp As New Excel.Application
    Dim src As Workbooks
    Dim ExcelSH As Worksheet

    ExcelApp.Visible = False

    ' OPEN THE SOURCE EXCEL WORKBOOK AND READ INFORMATION FROM A CELL.
    Workbooks.Open (srcPath & "\" & srcFileName)
    Set ExcelSH = ActiveWorkbook.Worksheets(srcSheet)
    ReadingExcelCell = ExcelSH.Range(srcCell).Value

    'Saving the destination file and close it.
    ActiveWorkbook.Save
    ActiveWorkbook.Close
    ExcelApp.Quit
    Set ExcelApp = Nothing

End Function
Sub WritevalueToExcelCell(srcPath, srcFileName, srcSheet, srcCell, CellValue)
    Dim ExcelApp As New Excel.Application
    Dim ExcelSH As Worksheet

    ExcelApp.Visible = False

    ' OPEN THE SOURCE EXCEL WORKBOOK AND READ INFORMATION FROM A CELL.
    Workbooks.Open (srcPath & "\" & srcFileName)
    Set ExcelSH = ActiveWorkbook.Worksheets(srcSheet)

    ExcelSH.Range(srcCell).Value = CellValue

    ActiveWorkbook.Save
    ActiveWorkbook.Close
    ExcelApp.Quit
    Set ExcelApp = Nothing
```

```
End Sub
Function FindingNextVolumeFraction(F1, F2, F3, F4, F5, F6, F7, F8, F9, F10,
F11, F12, CF)
    'F1, F2, ... are volumefractions
    'CF is current Volume Fraction
    'NF is Next volume fraction

    If F1 = CF Then
        FindingNextVolumeFraction = F2
    ElseIf F2 = CF Then
        FindingNextVolumeFraction = F3
    ElseIf F3 = CF Then
        FindingNextVolumeFraction = F4
    ElseIf F4 = CF Then
        FindingNextVolumeFraction = F5
    ElseIf F5 = CF Then
        FindingNextVolumeFraction = F6
    ElseIf F6 = CF Then
        FindingNextVolumeFraction = F7
    ElseIf F7 = CF Then
        FindingNextVolumeFraction = F8
    ElseIf F8 = CF Then
        FindingNextVolumeFraction = F9
    ElseIf F9 = CF Then
        FindingNextVolumeFraction = F10
    ElseIf F10 = CF Then
        FindingNextVolumeFraction = F11
    ElseIf F11 = CF Then
        FindingNextVolumeFraction = F12
    End If
End Function
Sub RunMatlab(FilePath)
    Dim RetVal
    RetVal = Shell(FilePath, 1)
End Sub
Sub Move_Rename_Folder(SourcePath, DestinationPath)
'This example move the folder from FromPath to ToPath.
    Dim FSO As Object
    Dim FromPath As String
    Dim ToPath As String

    FromPath = SourcePath  '<< Change
    ToPath = DestinationPath   '<< Change
    'Note: It is not possible to use a folder that exist in ToPath

    If Right(FromPath, 1) = "\" Then
        FromPath = Left(FromPath, Len(FromPath) - 1)
    End If

    If Right(ToPath, 1) = "\" Then
        ToPath = Left(ToPath, Len(ToPath) - 1)
    End If

    Set FSO = CreateObject("scripting.filesystemobject")
    If FSO.FolderExists(FromPath) = False Then
        MsgBox FromPath & " doesn't exist"
        Exit Sub
    End If

    If FSO.FolderExists(ToPath) = True Then
        MsgBox ToPath & " exist, not possible to move to a existing folder"
        Exit Sub
    End If

    FSO.MoveFolder Source:=FromPath, Destination:=ToPath
End Sub
Sub PrepairPoissonData1(FilePath, FileName1, SheetName1, FileName2, SheetName2)
    Dim ExcelApp1 As New Excel.Application
    Dim ExcelWB1 As Object
    Dim ExcelSH1 As Worksheet
    Dim ExcelApp2 As New Excel.Application
    Dim ExcelWB2 As Object
    Dim ExcelSH2 As Worksheet
    Dim LastRow As Long
    Dim LastColumn As Long
    Dim StartCell As Range
    Dim objTable As ListObject
    Dim BruteForce As String
    Dim i As Integer
    Dim x As Integer
    Dim arrExcelValues(10000000) As Variant

    'BruteForce = "TASKKILL /F /IM excel.exe"
    'Shell BruteForce, vbHide

    i = 2
    x = 0
    j = 0
    k = 2

    Set ExcelApp1 = CreateObject("Excel.Application")
    ExcelApp1.Visible = False
    Set ExcelWB1 = Workbooks.Open(FilePath & "\" & FileName1)
    Set ExcelSH1 = ActiveWorkbook.Worksheets(SheetName1)

    ExcelApp2.Visible = False
    Set ExcelWB2 = Workbooks.Open(FilePath & "\" & FileName2)
    Set ExcelSH2 = ActiveWorkbook.Worksheets(SheetName2)

    Do Until Range("B" & i).Value = ""
        arrExcelValues(x) = Range("B" & i).Value
        i = i + 1
        x = x + 1
    Loop

    ExcelSH2.Activate
    ActiveWorkbook.Save
    ActiveWorkbook.Close
    ExcelApp2.Quit
    Set ExcelApp2 = Nothing

    ExcelSH1.Activate
    Range("C1").Value = "Normal Elastic Strain (m/m)2"
    Do Until j = x
        Range("C" & k).Value = arrExcelValues(j)
        j = j + 1
        k = k + 1
    Loop

    ExcelSH1.Activate
    Set StartCell = Range("A1")
    Range("D1").Select
    'Find Last Row and Column
    LastRow = ExcelSH1.Cells(ExcelSH1.Rows.Count,
StartCell.Column).End(xlUp).Row
    LastColumn = ExcelSH1.Cells(StartCell.Row,
ExcelSH1.Columns.Count).End(xlToLeft).Column
    'Select Range
    ExcelSH1.Range(StartCell, ExcelSH1.Cells(LastRow, LastColumn)).Select
    'Inserting Table
    Set objTable = ActiveSheet.ListObjects.Add(xlSrcRange, Selection, , xlYes)
    objTable.TableStyle = "TableStyleMedium2"

    Range("D2").Select
    ActiveCell.FormulaR1C1 = _
        "=[@[Normal Elastic Strain (m/m)]]/([@[Normal Elastic Strain
(m/m)2]]*(-1))"

    'Dynamic Table and set Average function
    ActiveWorkbook.Names.Add Name:="AverageDynamicTable", RefersToR1C1:= _
        "=OFFSET('" & SheetName1 & "'!R2C4,,,COUNTA('" & SheetName1 & "'!C4)-
1)"
    ActiveWorkbook.Names("AverageDynamicTable").Comment = ""

    Range("F4").Select
    ActiveCell.FormulaR1C1 = "=AVERAGE(AverageDynamicTable)"

    ActiveWorkbook.Save
    ActiveWorkbook.Close
    ExcelApp1.Quit
    Set ExcelApp1 = Nothing
End Sub
```

Appendix III

The following is the IronPython code that was used in ANSYS Workbench to automate the FEA process. This code was utilized to generate the outcomes presented in Chapters 2 and 3.

```
#Setting Module needed for this script
import shutil
import time
print ("")
print ("################")
print ("Spherical Particles Mechanical and Thermal Characterization.")
print ("Designed By: Hamidreza Ahmadimoghaddam")
print ("################")
print ("")
time.sleep(0.3)
print ("")
print ("")


print ("")
print ("################")
print ("Creating Workbench Layout.")
print ("################")
print ("")
time.sleep(0.3)
template1 = GetTemplate(TemplateName="Geometry")
system1 = template1.CreateSystem()
template2 = GetTemplate(TemplateName="EngData")
system2 = template2.CreateSystem(
    Position="Below",
    RelativeTo=system1)
favorites1 = EngData.LoadFavoriteItems()
library1 = EngData.OpenLibrary(
    Name="Custom Materials",
    Source="C:/FEA Analysis/v19/Custom Materials.xml")
engineeringData1 = system2.GetContainer(ComponentName="Engineering Data")
matl1 = engineeringData1.ImportMaterial(
    Name="MAT1",
    Source="C:/FEA Analysis/v19/Custom Materials.xml")
matl2 = engineeringData1.ImportMaterial(
    Name="MAT2",
    Source="C:/FEA Analysis/v19/Custom Materials.xml")
matlProp1 = matl1.GetProperty(Name="Density")
materialPropertyData1 = matlProp1.GetPropertyData(
    Name="Density",
    Qualifiers={"Definition": "", "Behavior": ""})
materialVariable1 = materialPropertyData1.GetVariable(Name="Density")
parameter1 = Parameters.CreateParameter(
    Entity=materialVariable1,
    PropertyName="ParameterValue",
    DisplayText="Density")
parameter1.ValueQuantityName = "Density"
matlProp2 = matl1.GetProperty(Name="Coefficient of Thermal Expansion")
materialPropertyData2 = matlProp2.GetPropertyData(
    Name="Coefficient of Thermal Expansion",
    Qualifiers={"Definition": "Secant", "Behavior": "Isotropic"})
materialVariable2 = materialPropertyData2.GetVariable(Name="Coefficient of
Thermal Expansion")
parameter2 = Parameters.CreateParameter(
    Entity=materialVariable2,
    PropertyName="ParameterValue",
    DisplayText="Coefficient of Thermal Expansion")
parameter2.ValueQuantityName = "InvTemp1"
matlProp3 = matl1.GetProperty(Name="Elasticity")
materialPropertyData3 = matlProp3.GetPropertyData(
    Name="Elasticity",
    Qualifiers={"Definition": "", "Behavior": "Isotropic", "Derive from":
"Young's Modulus and Poisson's Ratio"})
materialVariable3 = materialPropertyData3.GetVariable(Name="Young's Modulus")
parameter3 = Parameters.CreateParameter(
    Entity=materialVariable3,
    PropertyName="ParameterValue",
    DisplayText="Young's Modulus")
parameter3.ValueQuantityName = "Stress"
materialVariable4 = materialPropertyData3.GetVariable(Name="Poisson's Ratio")
parameter4 = Parameters.CreateParameter(
    Entity=materialVariable4,
    PropertyName="ParameterValue",
    DisplayText="Poisson's Ratio")
parameter4.ValueQuantityName = ""
matlProp4 = matl1.GetProperty(Name="Thermal Conductivity")
materialPropertyData4 = matlProp4.GetPropertyData(
    Name="Thermal Conductivity",
    Qualifiers={"Definition": "", "Behavior": "Isotropic"})
materialVariable5 = materialPropertyData4.GetVariable(Name="Thermal
Conductivity")
parameter5 = Parameters.CreateParameter(
    Entity=materialVariable5,
    PropertyName="ParameterValue",
    DisplayText="Thermal Conductivity")
parameter5.ValueQuantityName = "Thermal Conductivity"
matlProp5 = matl2.GetProperty(Name="Density")
materialPropertyData5 = matlProp5.GetPropertyData(
    Name="Density",
    Qualifiers={"Definition": "", "Behavior": ""})
materialVariable6 = materialPropertyData5.GetVariable(Name="Density")
parameter6 = Parameters.CreateParameter(
    Entity=materialVariable6,
    PropertyName="ParameterValue",
    DisplayText="Density")
parameter6.ValueQuantityName = "Density"
matlProp6 = matl2.GetProperty(Name="Coefficient of Thermal Expansion")
materialPropertyData6 = matlProp6.GetPropertyData(
    Name="Coefficient of Thermal Expansion",
    Qualifiers={"Definition": "Secant", "Behavior": "Isotropic"})
materialVariable7 = materialPropertyData6.GetVariable(Name="Coefficient of
Thermal Expansion")
parameter7 = Parameters.CreateParameter(
    Entity=materialVariable7,
    PropertyName="ParameterValue",
    DisplayText="Coefficient of Thermal Expansion")
parameter7.ValueQuantityName = "InvTemp1"
matlProp7 = matl2.GetProperty(Name="Elasticity")
materialPropertyData7 = matlProp7.GetPropertyData(
    Name="Elasticity",
    Qualifiers={"Definition": "", "Behavior": "Isotropic", "Derive from":
"Young's Modulus and Poisson's Ratio"})
materialVariable8 = materialPropertyData7.GetVariable(Name="Young's Modulus")
parameter8 = Parameters.CreateParameter(
    Entity=materialVariable8,
    PropertyName="ParameterValue",
    DisplayText="Young's Modulus")
parameter8.ValueQuantityName = "Stress"
materialVariable9 = materialPropertyData7.GetVariable(Name="Poisson's Ratio")
parameter9 = Parameters.CreateParameter(
    Entity=materialVariable9,
    PropertyName="ParameterValue",
    DisplayText="Poisson's Ratio")
parameter9.ValueQuantityName = ""
matlProp8 = matl2.GetProperty(Name="Thermal Conductivity")
materialPropertyData8 = matlProp8.GetPropertyData(
    Name="Thermal Conductivity",
    Qualifiers={"Definition": "", "Behavior": "Isotropic"})
materialVariable10 = materialPropertyData8.GetVariable(Name="Thermal
Conductivity")
parameter10 = Parameters.CreateParameter(
    Entity=materialVariable10,
    PropertyName="ParameterValue",
    DisplayText="Thermal Conductivity")
parameter10.ValueQuantityName = "Thermal Conductivity"
template3 = GetTemplate(
    TemplateName="Static Structural",
    Solver="ANSYS")
system3 = template3.CreateSystem(
    Position="Right",
    RelativeTo=system1)
template4 = GetTemplate(
    TemplateName="Steady-State Thermal",
    Solver="ANSYS")
system4 = template4.CreateSystem(
    Position="Right",
    RelativeTo=system2)
geometryComponent1 = system3.GetComponent(Name="Geometry")
geometryComponent2 = system1.GetComponent(Name="Geometry")
geometryComponent1.ReplaceWithShare(
    TargetSystem=system3,
    ComponentToShare=geometryComponent2,
    SourceSystem=system1)
geometryComponent3 = system4.GetComponent(Name="Geometry")
```

```
geometryComponent3.ReplaceWithShare(
    TargetSystem=system4,
    ComponentToShare=geometryComponent2,
    SourceSystem=system1)
engineeringDataComponent1 = system3.GetComponent(Name="Engineering Data")
engineeringDataComponent2 = system2.GetComponent(Name="Engineering Data")
engineeringDataComponent1.ReplaceWithShare(
    TargetSystem=system3,
    ComponentToShare=engineeringDataComponent2,
    SourceSystem=system2)
engineeringDataComponent3 = system4.GetComponent(Name="Engineering Data")
engineeringDataComponent3.ReplaceWithShare(
    TargetSystem=system4,
    ComponentToShare=engineeringDataComponent2,
    SourceSystem=system2)

print ("")
print ("################")
print ("Workbench Layout was created Successfully!!!")
print ("################")
print ("")
time.sleep(0.3)
print ("")
print ("")


print ("")
print ("################")
print ("Start reading from Master file.")
print ("################")
print ("")
time.sleep(0.3)
# IronPython imports to enable Excel interop
import clr
clr.AddReference("Microsoft.Office.Interop.Excel")
import Microsoft.Office.Interop.Excel as Excel

# Open Excel and the PassWorkbook
MasterExcelFile = Excel.ApplicationClass()
MasterExcelFile.Visible = False
MasterExcelWorkbook = MasterExcelFile.Workbooks.Open("C:/Users/hahmadi/Google Drive/Codes/Project1/MaterialDatabase.xlsm")
MasterExcelWorksheet1 = MasterExcelWorkbook.Worksheets(1)

#Reading required Parameters from master excel file.
Density1 = MasterExcelWorksheet1.Range["B3"]
CoefThermalExpansion1 = MasterExcelWorksheet1.Range["B4"]
YoungsModulus1 = MasterExcelWorksheet1.Range["B5"]
PoissonRatio1 = MasterExcelWorksheet1.Range["B6"]
ThermalConductivity1 = MasterExcelWorksheet1.Range["B7"]
Density2 = MasterExcelWorksheet1.Range["C3"]
CoefThermalExpansion2 = MasterExcelWorksheet1.Range["C4"]
YoungsModulus2 = MasterExcelWorksheet1.Range["C5"]
PoissonRatio2 = MasterExcelWorksheet1.Range["C6"]
ThermalConductivity2 = MasterExcelWorksheet1.Range["C7"]
```

```
#Reading Analysis Parameter from master excel file.
MechanicalAnalysis = MasterExcelWorksheet1.Range["B36"]
ThermalAnalysis = MasterExcelWorksheet1.Range["B37"]

#Setting analysis variables
MechanicalAnalysisCheck = MechanicalAnalysis.Value2
ThermalAnalysisCheck = ThermalAnalysis.Value2

print ("")
print ("################")
print ("Setting Materials parameters in the analysis")
print ("################")
print ("")
time.sleep(0.3)
# Get the Workbench Parameters
P1Param = Parameters.GetParameter(Name="P1")
P2Param = Parameters.GetParameter(Name="P2")
P3Param = Parameters.GetParameter(Name="P3")
P4Param = Parameters.GetParameter(Name="P4")
P5Param = Parameters.GetParameter(Name="P5")
P6Param = Parameters.GetParameter(Name="P6")
P7Param = Parameters.GetParameter(Name="P7")
P8Param = Parameters.GetParameter(Name="P8")
P9Param = Parameters.GetParameter(Name="P9")
P10Param = Parameters.GetParameter(Name="P10")

# Assign values to the input parameters
P1Param.Expression = Density1.Value2.ToString() + " [kg m^-3]"
P2Param.Expression = CoefThermalExpansion1.Value2.ToString() + " [C^-1]"
P3Param.Expression = YoungsModulus1.Value2.ToString() + " [Pa]"
P4Param.Expression = PoissonRatio1.Value2.ToString()
P5Param.Expression = ThermalConductivity1.Value2.ToString() + " [W m^-1 C^-1]"
P6Param.Expression = Density2.Value2.ToString() + " [kg m^-3]"
P7Param.Expression = CoefThermalExpansion2.Value2.ToString() + " [C^-1]"
P8Param.Expression = YoungsModulus2.Value2.ToString() + " [Pa]"
P9Param.Expression = PoissonRatio2.Value2.ToString()
P10Param.Expression = ThermalConductivity2.Value2.ToString() + " [W m^-1 C^-1]"

#Quit from Master Excel file.
MasterExcelWorkbook.Close()
MasterExcelFile.Quit()
del MasterExcelFile

print ("")
print ("################")
print ("Reading from Master file was terminared Successfully!!!")
print ("################")
print ("")
time.sleep(0.3)

print ("")
print ("################")
print ("Materials parameters were set Successfully!!!")
print ("################")
print ("")
time.sleep(0.3)



print ("")
print ("################")
print ("Creating Geometry.")
print ("################")
print ("")
#Creating Geometry
#Getting geometry container
system1 = GetSystem(Name="Geom")
geometry1 = system1.GetContainer(ComponentName="Geometry")

#Oppening DesignModeler
geometry1.Edit(Interactive=True)

#Sending Javascript command to DesignModeler
geometry1.SendCommand(Command="""WB.AppletList.Applet(\"DSApplet\").App.Script.
doToolsRunMacro(\"C:/Users/hahmadi/Google Drive/Codes/Project1/
SphereParticlesGeometry.js\")""")

#Closing DesignModeler
geometry1.Exit()

print ("")
print ("################")
print ("Geometry Created Successfully!!!")
print ("################")
print ("")
time.sleep(0.3)

#Refresh Workbench before openning the mechanical.
Refresh()

#Defining wheather Mechanical analysis should be done or not.
if MechanicalAnalysisCheck == -1:

        print ("")
        print ("################")
        print ("Starting Mechanical Analysis.")
        print ("################")
        print ("")
        time.sleep(0.3)
        #Performing Mechanical Analysis
        #Getting Mechanical Model container
        system2 = GetSystem(Name="SYS")
        model1 = system2.GetContainer(ComponentName="Model")

        #Oppening Mechanical GUI
        model1.Edit(Interactive=True)

        print ("")
        print ("################")
        print ("Prepairing Mechanical FE model.")
        print ("################")
        print ("")
        time.sleep(0.3)

        #Sending Pre Javascript macro to mechanical model
        model1.SendCommand(Command="""WB.AppletList.Applet(\"DSApplet\").App.Scr
ipt.doToolsRunMacro(\"C:/Users/hahmadi/Google
Drive/Codes/Project1/MechanicalMacro_Pre.js\")""")

        print ("")
        print ("################")
        print ("Solving Mechanical FE model.")
        print ("################")
        print ("")
        time.sleep(0.3)

        #Sending Post Javascript macro to mechanical model
        model1.SendCommand(Command="""WB.AppletList.Applet(\"DSApplet\").App.Scr
ipt.doToolsRunMacro(\"C:/Users/hahmadi/Google
Drive/Codes/Project1/MechanicalMacro_Post.js\")""")

        print ("")
        print ("################")
        print ("Solving Thermal Expansion FE model.")
        print ("################")
        print ("")
        time.sleep(0.3)

        #Sending Post Javascript macro to mechanical model
        model1.SendCommand(Command="""WB.AppletList.Applet(\"DSApplet\").App.Scr
ipt.doToolsRunMacro(\"C:/Users/hahmadi/Google
Drive/Codes/Project1/MechanicalMacro_Post2.js\")""")

        #Closing Mechanical GUI
        model1.Exit()

        print ("")
        print ("################")
        print ("Mechanical result were extracted and analysis was terminated
successfully!!!")
        print ("################")
        print ("")
        time.sleep(0.5)


#Defining wheather Thermal analysis should be done or not.
if ThermalAnalysisCheck == -1:

        print ("")
        print ("################")
        print ("Starting Thermal Analysis.")
        print ("################")
        print ("")
        #Performing Thermal Analysis
        #Getting Thermal Model container
        system3 = GetSystem(Name="SYS 1")
        model2 = system3.GetContainer(ComponentName="Model")

        #Oppening Mechanical GUI
        model2.Edit(Interactive=True)

        print ("")
        print ("################")
        print ("Prepairing Thermal FE model.")
        print ("################")
        print ("")
        time.sleep(0.3)

        #Sending Pre Javascript macro to Thermal model
        model2.SendCommand(Command="""WB.AppletList.Applet(\"DSApplet\").App.Scr
ipt.doToolsRunMacro(\"C:/Users/hahmadi/Google
Drive/Codes/Project1/ThermalMacro_Pre.js\")""")

        print ("")
        print ("################")
        print ("Solving Thermal FE model.")
        print ("################")
        print ("")
        time.sleep(0.3)

        #Sending Post Javascript macro to Thermal model
        model2.SendCommand(Command="""WB.AppletList.Applet(\"DSApplet\").App.Scr
ipt.doToolsRunMacro(\"C:/Users/hahmadi/Google
Drive/Codes/Project1/ThermalMacro_Post.js\")""")

        #Closing Mechanical GUI
        model2.Exit()

        print ("")
        print ("################")
        print ("Thermal result were extracted and analysis was terminated
successfully!!!")
        print ("################")
        print ("")
        time.sleep(1)
```

Appendix IV

The following IronPython code was used in ANSYS Workbench to automate the FEA process. This code was utilized to generate the outcomes presented in Chapter 4.

```
#Setting Module needed for this script
import shutil
import time
print ("")
print ("################")
print ("Cylindrical Particles Mechanical and Thermal Characterization.")
print ("Designed By: Hamidreza Ahmadimoghaddam")
print ("################")
print ("")
time.sleep(0.3)
print ("")
print ("")


print ("")
print ("################")
print ("Creating Workbench Layout.")
print ("################")
print ("")
time.sleep(0.3)

template1 = GetTemplate(TemplateName="Geometry")
system1 = template1.CreateSystem()
template2 = GetTemplate(TemplateName="EngData")
system2 = template2.CreateSystem(
        Position="Below",
        RelativeTo=system1)
engineeringData1 = system2.GetContainer(ComponentName="Engineering Data")
matl1 = engineeringData1.ImportMaterial(
        Name="MAT1",
        Source="C:/FEA Analysis/v19/Custom Materials.xml")
matl2 = engineeringData1.ImportMaterial(
        Name="MAT3",
        Source="C:/FEA Analysis/v19/Custom Materials.xml")
matlProp1 = matl1.GetProperty(Name="Density")
materialPropertyData1 = matlProp1.GetPropertyData(
        Name="Density",
        Qualifiers={"Definition": "", "Behavior": ""})
materialVariable1 = materialPropertyData1.GetVariable(Name="Density")
parameter1 = Parameters.CreateParameter(
        Entity=materialVariable1,
        PropertyName="ParameterValue",
        DisplayText="Density")
parameter1.ValueQuantityName = "Density"
```

```
matlProp2 = matl1.GetProperty(Name="Coefficient of Thermal Expansion")
materialPropertyData2 = matlProp2.GetPropertyData(
        Name="Coefficient of Thermal Expansion",
        Qualifiers={"Definition": "Secant", "Behavior": "Isotropic"})
materialVariable2 = materialPropertyData2.GetVariable(Name="Coefficient of
Thermal Expansion")
parameter2 = Parameters.CreateParameter(
        Entity=materialVariable2,
        PropertyName="ParameterValue",
        DisplayText="Coefficient of Thermal Expansion")
parameter2.ValueQuantityName = "InvTemp1"
matlProp3 = matl1.GetProperty(Name="Elasticity")
materialPropertyData3 = matlProp3.GetPropertyData(
        Name="Elasticity",
        Qualifiers={"Definition": "", "Behavior": "Isotropic", "Derive from":
"Young's Modulus and Poisson's Ratio"})
materialVariable3 = materialPropertyData3.GetVariable(Name="Young's Modulus")
parameter3 = Parameters.CreateParameter(
        Entity=materialVariable3,
        PropertyName="ParameterValue",
        DisplayText="Young's Modulus")
parameter3.ValueQuantityName = "Stress"
materialVariable4 = materialPropertyData3.GetVariable(Name="Poisson's Ratio")
parameter4 = Parameters.CreateParameter(
        Entity=materialVariable4,
        PropertyName="ParameterValue",
        DisplayText="Poisson's Ratio")
parameter4.ValueQuantityName = ""
matlProp4 = matl1.GetProperty(Name="Thermal Conductivity")
materialPropertyData4 = matlProp4.GetPropertyData(
        Name="Thermal Conductivity",
        Qualifiers={"Definition": "", "Behavior": "Isotropic"})
materialVariable5 = materialPropertyData4.GetVariable(Name="Thermal
Conductivity")
parameter5 = Parameters.CreateParameter(
        Entity=materialVariable5,
        PropertyName="ParameterValue",
        DisplayText="Thermal Conductivity")
parameter5.ValueQuantityName = "Thermal Conductivity"
matlProp5 = matl2.GetProperty(Name="Density")
materialPropertyData5 = matlProp5.GetPropertyData(
        Name="Density",
        Qualifiers={"Definition": "", "Behavior": ""})
materialVariable6 = materialPropertyData5.GetVariable(Name="Density")
parameter6 = Parameters.CreateParameter(
        Entity=materialVariable6,
        PropertyName="ParameterValue",
        DisplayText="Density")
parameter6.ValueQuantityName = "Density"
matlProp6 = matl2.GetProperty(Name="Coefficient of Thermal Expansion")
materialPropertyData6 = matlProp6.GetPropertyData(
        Name="Coefficient of Thermal Expansion",
        Qualifiers={"Definition": "Secant", "Behavior": "Isotropic"})
materialVariable7 = materialPropertyData6.GetVariable(Name="Coefficient of
Thermal Expansion")
```

```
parameter7 = Parameters.CreateParameter(
        Entity=materialVariable7,
        PropertyName="ParameterValue",
        DisplayText="Coefficient of Thermal Expansion")
parameter7.ValueQuantityName = "InvTemp1"
matlProp7 = matl2.GetProperty(Name="Elasticity")
materialPropertyData7 = matlProp7.GetPropertyData(
        Name="Elasticity",
        Qualifiers={"Definition": "", "Behavior": "Isotropic", "Derive from":
"Young's Modulus and Poisson's Ratio"})
materialVariable8 = materialPropertyData7.GetVariable(Name="Young's Modulus")
parameter8 = Parameters.CreateParameter(
        Entity=materialVariable8,
        PropertyName="ParameterValue",
        DisplayText="Young's Modulus")
parameter8.ValueQuantityName = "Stress"
materialVariable9 = materialPropertyData7.GetVariable(Name="Poisson's Ratio")
parameter9 = Parameters.CreateParameter(
        Entity=materialVariable9,
        PropertyName="ParameterValue",
        DisplayText="Poisson's Ratio")
parameter9.ValueQuantityName = ""
matlProp8 = matl2.GetProperty(Name="Thermal Conductivity")
materialPropertyData8 = matlProp8.GetPropertyData(
        Name="Thermal Conductivity",
        Qualifiers={"Definition": "", "Behavior": "Orthotropic"})
materialVariable10 = materialPropertyData8.GetVariable(Name="Thermal
Conductivity X direction")
parameter10 = Parameters.CreateParameter(
        Entity=materialVariable10,
        PropertyName="ParameterValue",
        DisplayText="Thermal Conductivity X direction")
parameter10.ValueQuantityName = "Thermal Conductivity"
materialVariable11 = materialPropertyData8.GetVariable(Name="Thermal
Conductivity Y direction")
parameter11 = Parameters.CreateParameter(
        Entity=materialVariable11,
        PropertyName="ParameterValue",
        DisplayText="Thermal Conductivity Y direction")
parameter11.ValueQuantityName = "Thermal Conductivity"
materialVariable12 = materialPropertyData8.GetVariable(Name="Thermal
Conductivity Z direction")
parameter12 = Parameters.CreateParameter(
        Entity=materialVariable12,
        PropertyName="ParameterValue",
        DisplayText="Thermal Conductivity Z direction")
parameter12.ValueQuantityName = "Thermal Conductivity"
template3 = GetTemplate(
        TemplateName="Static Structural",
        Solver="ANSYS")
system3 = template3.CreateSystem(
        Position="Right",
        RelativeTo=system1)
template4 = GetTemplate(
        TemplateName="Steady-State Thermal",
        Solver="ANSYS")
system4 = template4.CreateSystem(
        Position="Right",
        RelativeTo=system2)
geometryComponent1 = system3.GetComponent(Name="Geometry")
geometryComponent2 = system1.GetComponent(Name="Geometry")
geometryComponent1.ReplaceWithShare(
        TargetSystem=system3,
        ComponentToShare=geometryComponent2,
        SourceSystem=system1)
geometryComponent3 = system4.GetComponent(Name="Geometry")
geometryComponent3.ReplaceWithShare(
        TargetSystem=system4,
        ComponentToShare=geometryComponent2,
        SourceSystem=system1)
engineeringDataComponent1 = system3.GetComponent(Name="Engineering Data")
engineeringDataComponent2 = system2.GetComponent(Name="Engineering Data")
engineeringDataComponent1.ReplaceWithShare(
        TargetSystem=system3,
        ComponentToShare=engineeringDataComponent2,
        SourceSystem=system2)
engineeringDataComponent3 = system4.GetComponent(Name="Engineering Data")
engineeringDataComponent3.ReplaceWithShare(
        TargetSystem=system4,
        ComponentToShare=engineeringDataComponent2,
        SourceSystem=system2)

print ("")
print ("################")
print ("Workbench Layout was created Successfully!!!")
print ("################")
print ("")
time.sleep(0.3)
print ("")
print ("")


print ("")
print ("################")
print ("Start reading from Master file.")
print ("################")
print ("")
time.sleep(0.3)
# IronPython imports to enable Excel interop
import clr
clr.AddReference("Microsoft.Office.Interop.Excel")
import Microsoft.Office.Interop.Excel as Excel

# Open Excel and the PassWorkbook
MasterExcelFile = Excel.ApplicationClass()
MasterExcelFile.Visible = False
MasterExcelWorkbook = MasterExcelFile.Workbooks.Open("C:/Users/hahmadi/Google
Drive/Codes/Project3/MaterialDatabase.xlsm")
MasterExcelWorksheet1 = MasterExcelWorkbook.Worksheets(1)
```

```
#Reading required Parameters from master excel file.
Density1 = MasterExcelWorksheet1.Range["B3"]
CoefThermalExpansion1 = MasterExcelWorksheet1.Range["B4"]
YoungsModulus1 = MasterExcelWorksheet1.Range["B5"]
PoissonRatio1 = MasterExcelWorksheet1.Range["B6"]
ThermalConductivity1 = MasterExcelWorksheet1.Range["B7"]
Density2 = MasterExcelWorksheet1.Range["C3"]
CoefThermalExpansion2 = MasterExcelWorksheet1.Range["C4"]
YoungsModulus2 = MasterExcelWorksheet1.Range["C5"]
PoissonRatio2 = MasterExcelWorksheet1.Range["C6"]
ThermalConductivity2X = MasterExcelWorksheet1.Range["C7"]
ThermalConductivity2Y = MasterExcelWorksheet1.Range["C8"]
ThermalConductivity2Z = MasterExcelWorksheet1.Range["C9"]

#Reading Analysis Parameter from master excel file.
MechanicalAnalysis = MasterExcelWorksheet1.Range["B36"]
ThermalAnalysis = MasterExcelWorksheet1.Range["B37"]

#Setting analysis variables
MechanicalAnalysisCheck = MechanicalAnalysis.Value2
ThermalAnalysisCheck = ThermalAnalysis.Value2

print ("")
print ("################")
print ("Setting Materials parameters in the analysis")
print ("################")
print ("")
time.sleep(0.3)
# Get the Workbench Parameters
P1Param = Parameters.GetParameter(Name="P1")
P2Param = Parameters.GetParameter(Name="P2")
P3Param = Parameters.GetParameter(Name="P3")
P4Param = Parameters.GetParameter(Name="P4")
P5Param = Parameters.GetParameter(Name="P5")
P6Param = Parameters.GetParameter(Name="P6")
P7Param = Parameters.GetParameter(Name="P7")
P8Param = Parameters.GetParameter(Name="P8")
P9Param = Parameters.GetParameter(Name="P9")
P10Param = Parameters.GetParameter(Name="P10")
P11Param = Parameters.GetParameter(Name="P11")
P12Param = Parameters.GetParameter(Name="P12")

# Assign values to the input parameters
P1Param.Expression = Density1.Value2.ToString() + " [kg m^-3]"
P2Param.Expression = CoefThermalExpansion1.Value2.ToString() + " [C^-1]"
P3Param.Expression = YoungsModulus1.Value2.ToString() + " [Pa]"
P4Param.Expression = PoissonRatio1.Value2.ToString()
P5Param.Expression = ThermalConductivity1.Value2.ToString() + " [W m^-1 C^-1]"
P6Param.Expression = Density2.Value2.ToString() + " [kg m^-3]"
P7Param.Expression = CoefThermalExpansion2.Value2.ToString() + " [C^-1]"
P8Param.Expression = YoungsModulus2.Value2.ToString() + " [Pa]"
P9Param.Expression = PoissonRatio2.Value2.ToString()
P10Param.Expression = ThermalConductivity2X.Value2.ToString() + " [W m^-1 C^-
1]"
```

```
P11Param.Expression = ThermalConductivity2Y.Value2.ToString() + " [W m^-1 C^-
1]"
P12Param.Expression = ThermalConductivity2Z.Value2.ToString() + " [W m^-1 C^-
1]"

#Quit from Master Excel file.
MasterExcelWorkbook.Close()
MasterExcelFile.Quit()
del MasterExcelFile

print ("")
print ("################")
print ("Reading from Master file was terminared Successfully!!!")
print ("################")
print ("")
time.sleep(0.3)

print ("")
print ("################")
print ("Materials parameters were set Successfully!!!")
print ("################")
print ("")
time.sleep(0.3)



print ("")
print ("################")
print ("Creating Geometry.")
print ("################")
print ("")
#Creating Geometry
#Getting geometry container
system1 = GetSystem(Name="Geom")
geometry1 = system1.GetContainer(ComponentName="Geometry")

#Oppening DesignModeler
geometry1.Edit(Interactive=True)

#Sending Javascript command to DesignModeler
geometry1.SendCommand(Command="""WB.AppletList.Applet(\"DSApplet\").App.Script.
doToolsRunMacro(\"C:/Users/hahmadi/Google Drive/Codes/Project3/
CylinderParticlesGeometry.js\")""")

#Closing DesignModeler
geometry1.Exit()

print ("")
print ("################")
print ("Geometry Created Successfully!!!")
print ("################")
print ("")
time.sleep(0.3)

#Refresh Workbench before openning the mechanical.
```

```
Refresh()

#Defining wheather Mechanical analysis should be done or not.
if MechanicalAnalysisCheck == -1:

        print ("")
        print ("################")
        print ("Starting Mechanical Analysis.")
        print ("################")
        print ("")
        time.sleep(0.3)
        #Performing Mechanical Analysis
        #Getting Mechanical Model container
        system2 = GetSystem(Name="SYS")
        model1 = system2.GetContainer(ComponentName="Model")

        #Oppening Mechanical GUI
        model1.Edit(Interactive=True)

        print ("")
        print ("################")
        print ("Prepairing Mechanical FE model.")
        print ("################")
        print ("")
        time.sleep(0.3)

        #Sending Pre Javascript macro to mechanical model
        model1.SendCommand(Command="""WB.AppletList.Applet(\"DSApplet\").App.Scr
ipt.doToolsRunMacro(\"C:/Users/hahmadi/Google
Drive/Codes/Project3/MechanicalMacro_Pre.js\")""")

        print ("")
        print ("################")
        print ("Solving Mechanical FE model.")
        print ("################")
        print ("")
        time.sleep(0.3)

        #Sending Post Javascript macro to mechanical model
        model1.SendCommand(Command="""WB.AppletList.Applet(\"DSApplet\").App.Scr
ipt.doToolsRunMacro(\"C:/Users/hahmadi/Google
Drive/Codes/Project3/MechanicalMacro_Post.js\")""")

        print ("")
        print ("################")
        print ("Solving Thermal Expansion FE model.")
        print ("################")
        print ("")
        time.sleep(0.3)

        #Sending Post Javascript macro to mechanical model
        model1.SendCommand(Command="""WB.AppletList.Applet(\"DSApplet\").App.Scr
ipt.doToolsRunMacro(\"C:/Users/hahmadi/Google
Drive/Codes/Project3/MechanicalMacro_Post2.js\")""")
```

```
        #Closing Mechanical GUI
        model1.Exit()

        print ("")
        print ("################")
        print ("Mechanical result were extracted and analysis was terminated
successfully!!!")
        print ("################")
        print ("")
        time.sleep(0.5)


#Defining wheather Thermal analysis should be done or not.
if ThermalAnalysisCheck == -1:

        print ("")
        print ("################")
        print ("Starting Thermal Analysis.")
        print ("################")
        print ("")
        #Performing Thermal Analysis
        #Getting Thermal Model container
        system3 = GetSystem(Name="SYS 1")
        model2 = system3.GetContainer(ComponentName="Model")

        #Oppening Mechanical GUI
        model2.Edit(Interactive=True)

        print ("")
        print ("################")
        print ("Prepairing Thermal FE model.")
        print ("################")
        print ("")
        time.sleep(0.3)

        #Sending Pre Javascript macro to Thermal model
        model2.SendCommand(Command="""WB.AppletList.Applet(\"DSApplet\").App.Scr
ipt.doToolsRunMacro(\"C:/Users/hahmadi/Google
Drive/Codes/Project3/ThermalMacro_Pre.js\")""")

        print ("")
        print ("################")
        print ("Solving Directional X Thermal FE model.")
        print ("################")
        print ("")
        time.sleep(0.3)

        #Sending Post Javascript macro to Thermal model
        model2.SendCommand(Command="""WB.AppletList.Applet(\"DSApplet\").App.Scr
ipt.doToolsRunMacro(\"C:/Users/hahmadi/Google
Drive/Codes/Project3/ThermalMacro_Post.js\")""")

        print ("")
        print ("################")
        print ("Solving Directional Y Thermal FE model.")
        print ("################")
        print ("")
        time.sleep(0.3)

        #Sending Post Javascript macro to Thermal model
        model2.SendCommand(Command="""WB.AppletList.Applet(\"DSApplet\").App.Scr
ipt.doToolsRunMacro(\"C:/Users/hahmadi/Google
Drive/Codes/Project3/ThermalMacro_Post2.js\")""")

        #Closing Mechanical GUI
        model2.Exit()

        print ("")
        print ("################")
        print ("Thermal result were extracted and analysis was terminated
successfully!!!")
        print ("################")
        print ("")
        time.sleep(1)
```

Appendix V

The following IronPython code was used in ANSYS Workbench to automate the FEA process. This code was utilized to generate the explore effective electrical conductivity and percolation behavior a presented in Chapter 5.

```
#Setting Module needed for this script
import shutil
import time
print ("")
print ("################")
print ("Spherical Particles Electrical Characterization.")
print ("Designed By: Hamidreza Ahmadimoghaddam")
print ("################")
print ("")
time.sleep(0.3)
print ("")
print ("")


print ("")
print ("################")
print ("Creating Workbench Layout.")
print ("################")
print ("")
time.sleep(0.3)

template1 = GetTemplate(TemplateName="Geometry")
system1 = template1.CreateSystem()
template2 = GetTemplate(
    TemplateName="Electric",
    Solver="ANSYS")
system2 = template2.CreateSystem(
    Position="Right",
    RelativeTo=system1)
geometryComponent1 = system2.GetComponent(Name="Geometry")
geometryComponent2 = system1.GetComponent(Name="Geometry")
geometryComponent1.ReplaceWithShare(
    TargetSystem=system2,
    ComponentToShare=geometryComponent2,
    SourceSystem=system1)
favorites1 = EngData.LoadFavoriteItems()
library1 = EngData.OpenLibrary(
    Name="Custom Materials",
    Source="C:/FEA Analysis/v19/Custom Materials.xml")
engineeringData1 = system2.GetContainer(ComponentName="Engineering Data")
matl1 = engineeringData1.ImportMaterial(
    Name="MAT1",
    Source="C:/FEA Analysis/v19/Custom Materials.xml")
matl2 = engineeringData1.ImportMaterial(
    Name="MAT2",
    Source="C:/FEA Analysis/v19/Custom Materials.xml")
matlProp1 = matl1.GetProperty(Name="Resistivity")
materialPropertyData1 = matlProp1.GetPropertyData(
    Name="Resistivity",
    Qualifiers={"Definition": "", "Behavior": "Isotropic"})
materialVariable1 = materialPropertyData1.GetVariable(Name="Resistivity")
parameter1 = Parameters.CreateParameter(
    Entity=materialVariable1,
    PropertyName="ParameterValue",
    DisplayText="Resistivity")
parameter1.ValueQuantityName = "Electrical Resistivity"
matlProp2 = matl2.GetProperty(Name="Resistivity")
materialPropertyData2 = matlProp2.GetPropertyData(
    Name="Resistivity",
    Qualifiers={"Definition": "", "Behavior": "Isotropic"})
materialVariable2 = materialPropertyData2.GetVariable(Name="Resistivity")
parameter2 = Parameters.CreateParameter(
    Entity=materialVariable2,
    PropertyName="ParameterValue",
    DisplayText="Resistivity")
parameter2.ValueQuantityName = "Electrical Resistivity"

print ("")
print ("################")
print ("Workbench Layout was created Successfully!!!")
print ("################")
print ("")
time.sleep(0.3)
print ("")
print ("")


print ("")
print ("################")
print ("Start reading from Master file.")
print ("################")
print ("")
time.sleep(0.3)
# IronPython imports to enable Excel interop
import clr
clr.AddReference("Microsoft.Office.Interop.Excel")
import Microsoft.Office.Interop.Excel as Excel

# Open Excel and the PassWorkbook
MasterExcelFile = Excel.ApplicationClass()
MasterExcelFile.Visible = False
MasterExcelWorkbook = MasterExcelFile.Workbooks.Open("C:/Users/hahmadi/Google
Drive/Codes/Project4/MaterialDatabase.xlsm")
MasterExcelWorksheet1 = MasterExcelWorkbook.Worksheets(1)

#Reading required Parameters from master excel file.
```

```
ElectricalResistance1 = MasterExcelWorksheet1.Range["B8"]
ElectricalResistance2 = MasterExcelWorksheet1.Range["C8"]

print ("")
print ("################")
print ("Setting Materials parameters in the analysis")
print ("################")
print ("")
time.sleep(0.3)

# Get the Workbench Parameters
P1Param = Parameters.GetParameter(Name="P1")
P2Param = Parameters.GetParameter(Name="P2")


# Assign values to the input parameters
P1Param.Expression = ElectricalResistance1.Value2.ToString() + " [ohm m]"
P2Param.Expression = ElectricalResistance2.Value2.ToString() + " [ohm m]"


#Quit from Master Excel file.
MasterExcelWorkbook.Close()
MasterExcelFile.Quit()
del MasterExcelFile

print ("")
print ("################")
print ("Reading from Master file was terminared Successfully!!!")
print ("################")
print ("")
time.sleep(0.3)

print ("")
print ("################")
print ("Materials parameters were set Successfully!!!")
print ("################")
print ("")
time.sleep(0.3)



print ("")
print ("################")
print ("Creating Geometry.")
print ("################")
print ("")
#Creating Geometry
#Getting geometry container
system1 = GetSystem(Name="Geom")
geometry1 = system1.GetContainer(ComponentName="Geometry")

#Oppening DesignModeler
geometry1.Edit(Interactive=True)

#Sending Javascript command to DesignModeler
```

```
geometry1.SendCommand(Command="""WB.AppletList.Applet(\"DSApplet\").App.Script.
doToolsRunMacro(\"C:/Users/hahmadi/Google Drive/Codes/Project4/
SphereParticlesGeometry.js\")""")

#Closing DesignModeler
geometry1.Exit()

print ("")
print ("################")
print ("Geometry Created Successfully!!!")
print ("################")
print ("")
time.sleep(0.3)

#Refresh Workbench before openning the mechanical.
Refresh()

print ("")
print ("################")
print ("Starting Electrical Analysis.")
print ("################")
print ("")
#Performing Electrical Analysis
#Getting Electrical Model container
system1 = GetSystem(Name="SYS")
model1 = system1.GetContainer(ComponentName="Model")

#Oppening Mechanical GUI
model1.Edit(Interactive=True)

print ("")
print ("################")
print ("Prepairing Electrical FE model.")
print ("################")
print ("")
time.sleep(0.3)

#Sending Pre Javascript macro to Electrical model
model1.SendCommand(Command="""WB.AppletList.Applet(\"DSApplet\").App.Script.doT
oolsRunMacro(\"C:/Users/hahmadi/Google
Drive/Codes/Project4/ElectricalMacro_Pre.js\")""")

print ("")
print ("################")
print ("Solving Electrical FE model.")
print ("################")
print ("")
time.sleep(0.3)

#Sending Post Javascript macro to Thermal model
model1.SendCommand(Command="""WB.AppletList.Applet(\"DSApplet\").App.Script.doT
oolsRunMacro(\"C:/Users/hahmadi/Google
Drive/Codes/Project4/ElectricalMacro_Post.js\")""")

#Closing Mechanical GUI
```

```
model1.Exit()

print ("")
print ("################")
print ("Electrical result were extracted and analysis was terminated
successfully!!!")
print ("################")
print ("")
time.sleep(1)
```

Appendix VI

The following IronPython code was used in ANSYS Workbench to automate the FEA process. This code was utilized to study piezoresistivity properties as a result of mechanical strain as presented in Chapter 5.

```
#Setting Module needed for this script
import shutil
import time
print ("")
print ("################")
print ("Spherical Particles Mechanical-Electrical Characterization.")
print ("Designed By: Hamidreza Ahmadimoghaddam")
print ("################")
print ("")
time.sleep(0.3)
print ("")
print ("")


print ("")
print ("################")
print ("Creating Workbench Layout.")
print ("################")
print ("")
time.sleep(0.3)

template1 = GetTemplate(TemplateName="Geometry")
system1 = template1.CreateSystem()
template2 = GetTemplate(TemplateName="EngData")
system2 = template2.CreateSystem(
    Position="Below",
    RelativeTo=system1)
favorites1 = EngData.LoadFavoriteItems()
library1 = EngData.OpenLibrary(
    Name="Custom Materials",
    Source="C:/FEA Analysis/v19/Custom Materials.xml")
engineeringData1 = system2.GetContainer(ComponentName="Engineering Data")
matl1 = engineeringData1.ImportMaterial(
    Name="MAT1",
    Source="C:/FEA Analysis/v19/Custom Materials.xml")
matl2 = engineeringData1.ImportMaterial(
    Name="MAT2",
    Source="C:/FEA Analysis/v19/Custom Materials.xml")
matlProp1 = matl1.GetProperty(Name="Density")
materialPropertyData1 = matlProp1.GetPropertyData(
    Name="Density",
    Qualifiers={"Definition": "", "Behavior": ""})
materialVariable1 = materialPropertyData1.GetVariable(Name="Density")
parameter1 = Parameters.CreateParameter(
    Entity=materialVariable1,
    PropertyName="ParameterValue",
    DisplayText="Density")
parameter1.ValueQuantityName = "Density"
matlProp2 = matl1.GetProperty(Name="Coefficient of Thermal Expansion")
materialPropertyData2 = matlProp2.GetPropertyData(
    Name="Coefficient of Thermal Expansion",
    Qualifiers={"Definition": "Secant", "Behavior": "Isotropic"})
materialVariable2 = materialPropertyData2.GetVariable(Name="Coefficient of
Thermal Expansion")
parameter2 = Parameters.CreateParameter(
    Entity=materialVariable2,
    PropertyName="ParameterValue",
    DisplayText="Coefficient of Thermal Expansion")
parameter2.ValueQuantityName = "InvTemp1"
matlProp3 = matl1.GetProperty(Name="Elasticity")
materialPropertyData3 = matlProp3.GetPropertyData(
    Name="Elasticity",
    Qualifiers={"Definition": "", "Behavior": "Isotropic", "Derive from":
"Young's Modulus and Poisson's Ratio"})
materialVariable3 = materialPropertyData3.GetVariable(Name="Young's Modulus")
parameter3 = Parameters.CreateParameter(
    Entity=materialVariable3,
    PropertyName="ParameterValue",
    DisplayText="Young's Modulus")
parameter3.ValueQuantityName = "Stress"
materialVariable4 = materialPropertyData3.GetVariable(Name="Poisson's Ratio")
parameter4 = Parameters.CreateParameter(
    Entity=materialVariable4,
    PropertyName="ParameterValue",
    DisplayText="Poisson's Ratio")
parameter4.ValueQuantityName = ""
matlProp4 = matl1.GetProperty(Name="Thermal Conductivity")
materialPropertyData4 = matlProp4.GetPropertyData(
    Name="Thermal Conductivity",
    Qualifiers={"Definition": "", "Behavior": "Isotropic"})
materialVariable5 = materialPropertyData4.GetVariable(Name="Thermal
Conductivity")
parameter5 = Parameters.CreateParameter(
    Entity=materialVariable5,
    PropertyName="ParameterValue",
    DisplayText="Thermal Conductivity")
parameter5.ValueQuantityName = "Thermal Conductivity"
matlProp5 = matl1.GetProperty(Name="Resistivity")
materialPropertyData5 = matlProp5.GetPropertyData(
    Name="Resistivity",
    Qualifiers={"Definition": "", "Behavior": "Isotropic"})
materialVariable6 = materialPropertyData5.GetVariable(Name="Resistivity")
parameter6 = Parameters.CreateParameter(
    Entity=materialVariable6,
    PropertyName="ParameterValue",
    DisplayText="Resistivity")
```

```
parameter6.ValueQuantityName = "Electrical Resistivity"
matlProp6 = matl2.GetProperty(Name="Density")
materialPropertyData6 = matlProp6.GetPropertyData(
    Name="Density",
    Qualifiers={"Definition": "", "Behavior": ""})
materialVariable7 = materialPropertyData6.GetVariable(Name="Density")
parameter7 = Parameters.CreateParameter(
    Entity=materialVariable7,
    PropertyName="ParameterValue",
    DisplayText="Density")
parameter7.ValueQuantityName = "Density"
matlProp7 = matl2.GetProperty(Name="Coefficient of Thermal Expansion")
materialPropertyData7 = matlProp7.GetPropertyData(
    Name="Coefficient of Thermal Expansion",
    Qualifiers={"Definition": "Secant", "Behavior": "Isotropic"})
materialVariable8 = materialPropertyData7.GetVariable(Name="Coefficient of
Thermal Expansion")
parameter8 = Parameters.CreateParameter(
    Entity=materialVariable8,
    PropertyName="ParameterValue",
    DisplayText="Coefficient of Thermal Expansion")
parameter8.ValueQuantityName = "InvTemp1"
matlProp8 = matl2.GetProperty(Name="Elasticity")
materialPropertyData8 = matlProp8.GetPropertyData(
    Name="Elasticity",
    Qualifiers={"Definition": "", "Behavior": "Isotropic", "Derive from":
"Young's Modulus and Poisson's Ratio"})
materialVariable9 = materialPropertyData8.GetVariable(Name="Young's Modulus")
parameter9 = Parameters.CreateParameter(
    Entity=materialVariable9,
    PropertyName="ParameterValue",
    DisplayText="Young's Modulus")
parameter9.ValueQuantityName = "Stress"
materialVariable10 = materialPropertyData8.GetVariable(Name="Poisson's Ratio")
parameter10 = Parameters.CreateParameter(
    Entity=materialVariable10,
    PropertyName="ParameterValue",
    DisplayText="Poisson's Ratio")
parameter10.ValueQuantityName = ""
matlProp9 = matl2.GetProperty(Name="Thermal Conductivity")
materialPropertyData9 = matlProp9.GetPropertyData(
    Name="Thermal Conductivity",
    Qualifiers={"Definition": "", "Behavior": "Isotropic"})
materialVariable11 = materialPropertyData9.GetVariable(Name="Thermal
Conductivity")
parameter11 = Parameters.CreateParameter(
    Entity=materialVariable11,
    PropertyName="ParameterValue",
    DisplayText="Thermal Conductivity")
parameter11.ValueQuantityName = "Thermal Conductivity"
matlProp10 = matl2.GetProperty(Name="Resistivity")
materialPropertyData10 = matlProp10.GetPropertyData(
    Name="Resistivity",
    Qualifiers={"Definition": "", "Behavior": "Isotropic"})
materialVariable12 = materialPropertyData10.GetVariable(Name="Resistivity")
```

```
parameter12 = Parameters.CreateParameter(
    Entity=materialVariable12,
    PropertyName="ParameterValue",
    DisplayText="Resistivity")
parameter12.ValueQuantityName = "Electrical Resistivity"
template3 = GetTemplate(
    TemplateName="Static Structural",
    Solver="ANSYS")
system3 = template3.CreateSystem(
    Position="Right",
    RelativeTo=system1)
geometryComponent1 = system3.GetComponent(Name="Geometry")
geometryComponent2 = system1.GetComponent(Name="Geometry")
geometryComponent1.ReplaceWithShare(
    TargetSystem=system3,
    ComponentToShare=geometryComponent2,
    SourceSystem=system1)
engineeringDataComponent1 = system3.GetComponent(Name="Engineering Data")
engineeringDataComponent2 = system2.GetComponent(Name="Engineering Data")
engineeringDataComponent1.ReplaceWithShare(
    TargetSystem=system3,
    ComponentToShare=engineeringDataComponent2,
    SourceSystem=system2)
template4 = GetTemplate(
    TemplateName="Electric",
    Solver="ANSYS")
system4 = template4.CreateSystem(
    Position="Right",
    RelativeTo=system3)
engineeringDataComponent3 = system4.GetComponent(Name="Engineering Data")
engineeringDataComponent3.ReplaceWithShare(
    TargetSystem=system4,
    ComponentToShare=engineeringDataComponent2,
    SourceSystem=system2)
solutionComponent1 = system3.GetComponent(Name="Solution")
modelComponent1 = system4.GetComponent(Name="Model")
solutionComponent1.TransferData(TargetComponent=modelComponent1)
system5 = template1.CreateSystem(
    Position="Right",
    RelativeTo=system4)
system6 = template4.CreateSystem(
    Position="Right",
    RelativeTo=system5)
geometryComponent3 = system6.GetComponent(Name="Geometry")
geometryComponent4 = system5.GetComponent(Name="Geometry")
geometryComponent3.ReplaceWithShare(
    TargetSystem=system6,
    ComponentToShare=geometryComponent4,
    SourceSystem=system5)
engineeringDataComponent4 = system6.GetComponent(Name="Engineering Data")
engineeringDataComponent4.ReplaceWithShare(
    TargetSystem=system6,
    ComponentToShare=engineeringDataComponent2,
    SourceSystem=system2)
```

```
print ("")
print ("################")
print ("Workbench Layout was created Successfully!!!")
print ("################")
print ("")
time.sleep(0.3)
print ("")
print ("")


print ("")
print ("################")
print ("Start reading from Master file.")
print ("################")
print ("")
time.sleep(0.3)
# IronPython imports to enable Excel interop
import clr
clr.AddReference("Microsoft.Office.Interop.Excel")
import Microsoft.Office.Interop.Excel as Excel

# Open Excel and the PassWorkbook
MasterExcelFile = Excel.ApplicationClass()
MasterExcelFile.Visible = False
MasterExcelWorkbook = MasterExcelFile.Workbooks.Open("C:/Users/hahmadi/Google Drive/Codes/Project5/MaterialDatabase.xlsm")
MasterExcelWorksheet1 = MasterExcelWorkbook.Worksheets(1)

#Reading required Parameters from master excel file.
Density1 = MasterExcelWorksheet1.Range["B3"]
CoefThermalExpansion1 = MasterExcelWorksheet1.Range["B4"]
YoungsModulus1 = MasterExcelWorksheet1.Range["B5"]
PoissonRatio1 = MasterExcelWorksheet1.Range["B6"]
ThermalConductivity1 = MasterExcelWorksheet1.Range["B7"]
ElectricalResistance1 = MasterExcelWorksheet1.Range["B8"]
Density2 = MasterExcelWorksheet1.Range["C3"]
CoefThermalExpansion2 = MasterExcelWorksheet1.Range["C4"]
YoungsModulus2 = MasterExcelWorksheet1.Range["C5"]
PoissonRatio2 = MasterExcelWorksheet1.Range["C6"]
ThermalConductivity2 = MasterExcelWorksheet1.Range["C7"]
ElectricalResistance2 = MasterExcelWorksheet1.Range["C8"]

print ("")
print ("################")
print ("Setting Materials parameters in the analysis")
print ("################")
print ("")
time.sleep(0.3)
# Get the Workbench Parameters
P1Param = Parameters.GetParameter(Name="P1")
P2Param = Parameters.GetParameter(Name="P2")
P3Param = Parameters.GetParameter(Name="P3")
P4Param = Parameters.GetParameter(Name="P4")
P5Param = Parameters.GetParameter(Name="P5")
P6Param = Parameters.GetParameter(Name="P6")
```

```
P7Param = Parameters.GetParameter(Name="P7")
P8Param = Parameters.GetParameter(Name="P8")
P9Param = Parameters.GetParameter(Name="P9")
P10Param = Parameters.GetParameter(Name="P10")
P11Param = Parameters.GetParameter(Name="P11")
P12Param = Parameters.GetParameter(Name="P12")

# Assign values to the input parameters
P1Param.Expression = Density1.Value2.ToString() + " [kg m^-3]"
P2Param.Expression = CoefThermalExpansion1.Value2.ToString() + " [C^-1]"
P3Param.Expression = YoungsModulus1.Value2.ToString() + " [Pa]"
P4Param.Expression = PoissonRatio1.Value2.ToString()
P5Param.Expression = ThermalConductivity1.Value2.ToString() + " [W m^-1 C^-1]"
P6Param.Expression = ElectricalResistance1.Value2.ToString() + " [ohm m]"
P7Param.Expression = Density2.Value2.ToString() + " [kg m^-3]"
P8Param.Expression = CoefThermalExpansion2.Value2.ToString() + " [C^-1]"
P9Param.Expression = YoungsModulus2.Value2.ToString() + " [Pa]"
P10Param.Expression = PoissonRatio2.Value2.ToString()
P11Param.Expression = ThermalConductivity2.Value2.ToString() + " [W m^-1 C^-1]"
P12Param.Expression = ElectricalResistance2.Value2.ToString() + " [ohm m]"

#Quit from Master Excel file.
MasterExcelWorkbook.Close()
MasterExcelFile.Quit()
del MasterExcelFile

print ("")
print ("################")
print ("Reading from Master file was terminared Successfully!!!")
print ("################")
print ("")
time.sleep(0.3)

print ("")
print ("################")
print ("Materials parameters were set Successfully!!!")
print ("################")
print ("")
time.sleep(0.3)

print ("")
print ("################")
print ("Creating Structural Geometry.")
print ("################")
print ("")
#Creating Geometry
#Getting geometry container
system1 = GetSystem(Name="Geom")
geometry1 = system1.GetContainer(ComponentName="Geometry")

#Oppening DesignModeler
geometry1.Edit(Interactive=True)

#Sending Javascript command to DesignModeler
```

```
geometry1.SendCommand(Command="""WB.AppletList.Applet(\"DSApplet\").App.Script.
doToolsRunMacro(\"C:/Users/hahmadi/Google Drive/Codes/Project5/
SphereParticlesGeometryPre.js\")""")

#Closing DesignModeler
geometry1.Exit()

print ("")
print ("################")
print ("Geometry Created Successfully!!!")
print ("################")
print ("")
time.sleep(0.3)

#Refresh Workbench before openning the mechanical.
Refresh()

print ("")
print ("################")
print ("Starting Mechanical Analysis.")
print ("################")
print ("")
time.sleep(0.3)
#Performing Mechanical Analysis
#Getting Mechanical Model container
system2 = GetSystem(Name="SYS")
model1 = system2.GetContainer(ComponentName="Model")

#Oppening Mechanical GUI
model1.Edit(Interactive=True)

print ("")
print ("################")
print ("Prepairing Mechanical FE model.")
print ("################")
print ("")
time.sleep(0.3)

#Sending Pre Javascript macro to mechanical model
model1.SendCommand(Command="""WB.AppletList.Applet(\"DSApplet\").App.Script.doT
oolsRunMacro(\"C:/Users/hahmadi/Google
Drive/Codes/Project5/MechanicalMacro_Pre.js\")""")

print ("")
print ("################")
print ("Solving Mechanical FE model.")
print ("################")
print ("")
time.sleep(0.3)

#Sending Post Javascript macro to mechanical model
model1.SendCommand(Command="""WB.AppletList.Applet(\"DSApplet\").App.Script.doT
oolsRunMacro(\"C:/Users/hahmadi/Google
Drive/Codes/Project5/MechanicalMacro_Post.js\")""")
```

```
#Closing Mechanical GUI
model1.Exit()

#Refresh Workbench before openning the Electrical analysis.
Refresh()
Update()

print ("")
print ("################")
print ("Starting Electrical Analysis.")
print ("################")
print ("")
#Performing Electrical Analysis
#Getting Electrical Model container
system3 = GetSystem(Name="SYS 1")
model2 = system3.GetContainer(ComponentName="Model")

#Oppening Mechanical GUI
model2.Edit(Interactive=True)

print ("")
print ("################")
print ("Extracting Geometry Data.")
print ("################")
print ("")
time.sleep(0.3)

#Sending Pre Javascript macro to Electrical model
model2.SendCommand(Command="""WB.AppletList.Applet(\"DSApplet\").App.Script.doT
oolsRunMacro(\"C:/Users/hahmadi/Google
Drive/Codes/Project5/ExtractingGeometryData.js\")""")

#Closing Mechanical GUI
model2.Exit()

print ("")
print ("################")
print ("Creating Electrical Geometry.")
print ("################")
print ("")
#Creating Geometry
#Getting geometry container
system4 = GetSystem(Name="Geom 1")
geometry2 = system4.GetContainer(ComponentName="Geometry")

#Oppening DesignModeler
geometry2.Edit(Interactive=True)

#Sending Javascript command to DesignModeler
geometry2.SendCommand( Command = """var DM =
ag.wb.AppletList.Applet("AGApplet").App;
var lv = ag.listview

//Changing the unit to Meter and Degree
var Unit = agb.SetSessionUnits(agc.UnitMeter, agc.UnitDegree, agc.N0);
```

```
//Reading volume fraction and iteration information from material database
Excel file.
var MasterExcel = new ActiveXObject("Excel.Application");
MasterExcel.Visible = false;
var MasterExcel_file = MasterExcel.Workbooks.Open("C:/Users/hahmadi/Google
Drive/Codes/Project5/MaterialDatabase.xlsm");
var MasterExcel_sheet = MasterExcel_file.Worksheets("Sheet1");

//Setting current Volume fraction
var CurrentVolumeFraction = MasterExcel_sheet.Cells(25, 2).Value;

//Setting current Iteration
var CurrentIteration = MasterExcel_sheet.Cells(26, 2).Value;

//Setting RVE size
var RVESize = MasterExcel_sheet.Cells(11, 2).Value;

//Setting Plate Thickness
var PlateThickness = MasterExcel_sheet.Cells(10, 2).Value;

//Extracting Mechanical-Electrical Iteration
var Iteration = MasterExcel_sheet.Cells(27, 2).Value;

//Closing Master Excel file
MasterExcel_file.Close();
MasterExcel.Application.Quit()

//Reading Displacement from master excelfile information from master Excel
file.
var MasterExcel2 = new ActiveXObject("Excel.Application");
MasterExcel2.Visible = false;
var MasterExcel2_file = MasterExcel2.Workbooks.Open("C:/Users/hahmadi/Google
Drive/Codes/Project5/MasterResultFile.xlsx");
var MasterExcel2_sheet = MasterExcel2_file.Worksheets("Sheet1");

//Setting RVE displacement
var RVESizeDisp = MasterExcel2_sheet.Cells(Iteration + 1, 1).Value;

//Closing Master Excel file
MasterExcel2_file.Close();
MasterExcel2.Application.Quit()

//RVE total lenght
var RVETotalLenght = RVESize + RVESizeDisp

//Reading RVE height from Excel file.
var RandomNumberExcel = new ActiveXObject("Excel.Application");
RandomNumberExcel.Visible = false;
var RandomNumberExcel_file =
RandomNumberExcel.Workbooks.Open("C:/Users/hahmadi/Google
Drive/Codes/Project5/ParticlesLocation/Iteration" + Iteration + ".xlsx");
var RandomNumberExcel_sheet = RandomNumberExcel_file.Worksheets("Sheet1");

//Creating path for saving random number excel file
```

```
var Path = "C:/Users/hahmadi/Google Drive/Codes/Project5/VF" +
CurrentVolumeFraction + "/Geometries/VF" + CurrentVolumeFraction + "_Geometry_"
+ CurrentIteration + ".xlsm"

// Set random number Excel file
var RandomNumberExcel2 = new ActiveXObject("Excel.Application");
RandomNumberExcel2.Visible = false;
var RandomNumberExcel2_file = RandomNumberExcel2.Workbooks.Open(Path);
var RandomNumberExcel2_sheet = RandomNumberExcel2_file.Worksheets("Sheet1");

//Extracting RVE height and wide (Added 0.002 for making sure that all
particles will fit within RVE)
var RVEHeight = RandomNumberExcel_sheet.Cells(1, 8).Value;

//Extracting number of Components from excel file created in previous step
var NumberOfComponents = RandomNumberExcel_sheet.Cells(1, 9).Value;
var NumberOfParticles = NumberOfComponents - 3;
//Creating RVE
CreatingBox(RVEHeight, RVETotalLenght);

//Creating Back Plate
CreatingBox2(RVEHeight, PlateThickness)

//Creating Front Plate
CreatingBox3(RVEHeight, PlateThickness, RVETotalLenght)

//Raw numerator in random number Excel file
var j = 1
while (RandomNumberExcel_sheet.Cells(j, 1).Value != 0) {
    ReadDataAndCreateSphere(j);
    j = j + 1;
}

//Closing random number Excel file
RandomNumberExcel_file.Close();
RandomNumberExcel.Application.Quit()

//Closing random number Excel file
RandomNumberExcel2_file.Close();
RandomNumberExcel2.Application.Quit()

//Generate all the bodies
ag.b.Regen();

//creating Boolean from bodies
var Pat = ag.gui.CreateBoolean()
ag.listview.ActivateItem("Operation");
ag.listview.ItemValue = "Subtract";

//Selecting body for being selected as target
ag.bodyPick;
Select = ag.m.Select3dBody(0, 0, 0, 0);

//Modifing Target section of boolean
ag.listview.ActivateItem("Target Bodies");
ag.listview.ItemValue = "Apply";

//Selecting Bodies for being selected as tools
ag.bodyPick;
for (i = 3; i <= j + 2; i++) {
    ag.m.Select3dBody(i, 0, 0, 0);
}

//Modifing Tooll section of boolean
ag.listview.ActivateItem("Tool Bodies");
ag.listview.ItemValue = "Apply";

//Modifiing Section to preserve all bodies.
ag.listview.ActivateItem("Preserve Tool Bodies?");
ag.listview.ItemValue = "Yes";

ag.b.Regen();


function ReadDataAndCreateSphere(x) {
    var Raw = x
    var Sphere = [0, 0, 0, 0, 0, 0, 0]
    for (i = 1; i < 8; i++) {
        if (i != 7) {
            Sphere[i - 1] = RandomNumberExcel_sheet.Cells(x, i).Value;
        }
        else {
            Sphere[i - 1] = RandomNumberExcel2_sheet.Cells(NumberOfParticles +
1 - x, i).Value;
        }
    }
    feature = ag.gui.CreatePrimitive(1);
    lv.ActivateItem("Operation");
    lv.ItemValue = "Add Frozen";
    lv.ActivateItem("FD3, Origin X Coordinate");
    lv.ItemValue = "" + Sphere[0];
    lv.ActivateItem("FD4, Origin Y Coordinate");
    lv.ItemValue = "" + Sphere[1];
    lv.ActivateItem("FD5, Origin Z Coordinate");
    lv.ItemValue = "" + Sphere[2];
    lv.ActivateItem("FD6, Radius (>0)");
    lv.ItemValue = "" + Sphere[6];


}
function selectNode(target) {
    var Nodes = DS.Tree.AllObjects;
    var count = Nodes.Count;
    var name, current, ID;
    for (var i = 1; i <= count; i++) {
        current = Nodes(i);
        name = current.Name;
        if (name == target) {
            ID = current.ID
            DS.Tree.FirstActiveObjectID = ID
            //reset mechanical to adjust ListView
            SC.wb_OnActivate()
            return;
        }
    }
    return false;
}

//Creating RVE
function CreatingBox(x, y) {
    var Box = [0, 0, 0, y, x, x]

    feature = ag.gui.CreatePrimitive(2);
    lv.ActivateItem("FD3, Point 1 X Coordinate");
    lv.ItemValue = "" + Box[0];
    lv.ActivateItem("FD4, Point 1 Y Coordinate");
    lv.ItemValue = "" + Box[1];
    lv.ActivateItem("FD5, Point 1 Z Coordinate");
    lv.ItemValue = "" + Box[2];
    lv.ActivateItem("FD6, Diagonal X Component");
    lv.ItemValue = "" + Box[3];
    lv.ActivateItem("FD7, Diagonal Y Component");
    lv.ItemValue = "" + Box[4];
    lv.ActivateItem("FD8, Diagonal Z Component");
    lv.ItemValue = "" + Box[5];

}

//Creating Back Plate
//X is RVE size and Y is Plate thickness
function CreatingBox2(x, y) {
    var Box = [-1 * y, 0, 0, y, x, x]

    feature = ag.gui.CreatePrimitive(2);
    lv.ActivateItem("Operation");
    lv.ItemValue = "Add Frozen";
    lv.ActivateItem("FD3, Point 1 X Coordinate");
    lv.ItemValue = "" + Box[0];
    lv.ActivateItem("FD4, Point 1 Y Coordinate");
    lv.ItemValue = "" + Box[1];
    lv.ActivateItem("FD5, Point 1 Z Coordinate");
    lv.ItemValue = "" + Box[2];
    lv.ActivateItem("FD6, Diagonal X Component");
    lv.ItemValue = "" + Box[3];
    lv.ActivateItem("FD7, Diagonal Y Component");
    lv.ItemValue = "" + Box[4];
    lv.ActivateItem("FD8, Diagonal Z Component");
    lv.ItemValue = "" + Box[5];

}

//Creating Front Plate
//X is RVE size and Y is Plate thickness
function CreatingBox3(x, y, z) {
    var Box = [z, 0, 0, y, x, x]
    feature = ag.gui.CreatePrimitive(2);
    lv.ActivateItem("Operation");
    lv.ItemValue = "Add Frozen";
    lv.ActivateItem("FD3, Point 1 X Coordinate");
    lv.ItemValue = "" + Box[0];
    lv.ActivateItem("FD4, Point 1 Y Coordinate");
    lv.ItemValue = "" + Box[1];
    lv.ActivateItem("FD5, Point 1 Z Coordinate");
    lv.ItemValue = "" + Box[2];
    lv.ActivateItem("FD6, Diagonal X Component");
    lv.ItemValue = "" + Box[3];
    lv.ActivateItem("FD7, Diagonal Y Component");
    lv.ItemValue = "" + Box[4];
    lv.ActivateItem("FD8, Diagonal Z Component");
    lv.ItemValue = "" + Box[5];

}""")

#Closing DesignModeler
geometry2.Exit()

print ("")
print ("################")
print ("Geometry Created Successfully!!!")
print ("################")
print ("")
time.sleep(0.3)

#Refresh Workbench before openning the mechanical.
Refresh()


print ("")
print ("################")
print ("Starting Electrical Analysis.")
print ("################")
print ("")
time.sleep(0.3)
#Performing Mechanical Analysis
#Getting Mechanical Model container
system5 = GetSystem(Name="SYS 2")
model3 = system5.GetContainer(ComponentName="Model")

#Oppening Mechanical GUI
model3.Edit(Interactive=True)

print ("")
print ("################")
print ("Prepairing Electrical FE model.")
print ("################")
print ("")
time.sleep(0.3)

#Sending Pre Javascript macro to mechanical model
```

```
model3.SendCommand(Command="""WB.AppletList.Applet(\"DSApplet\").App.Script.doT
oolsRunMacro(\"C:/Users/hahmadi/Google
Drive/Codes/Project5/ElectricalMacro_Pre.js\")""")

print ("")
print ("################")
print ("Solving Electrical FE model.")
print ("################")
print ("")
time.sleep(0.3)

#Sending Post Javascript macro to mechanical model
model3.SendCommand(Command="""WB.AppletList.Applet(\"DSApplet\").App.Script.doT
oolsRunMacro(\"C:/Users/hahmadi/Google
Drive/Codes/Project5/ElectricalMacro_Post.js\")""")

#Closing Mechanical GUI
model3.Exit()

#Refresh Workbench before openning the Electrical analysis.
Refresh()
Update()
```

Appendix VII

The following IronPython code was used in ANSYS Workbench to automate the FEA process. This code was utilized to study piezoresistivity properties as the result of changing temperature as presented in Chapter 5.

```
#Setting Module needed for this script
import shutil
import time
print ("")
print ("################")
print ("Spherical Particles Mechanical-Electrical Characterization.")
print ("Designed By: Hamidreza Ahmadimoghaddam")
print ("################")
print ("")
time.sleep(0.3)
print ("")
print ("")


print ("")
print ("################")
print ("Creating Workbench Layout.")
print ("################")
print ("")
time.sleep(0.3)

template1 = GetTemplate(TemplateName="Geometry")
system1 = template1.CreateSystem()
template2 = GetTemplate(TemplateName="EngData")
system2 = template2.CreateSystem(
    Position="Below",
    RelativeTo=system1)
favorites1 = EngData.LoadFavoriteItems()
library1 = EngData.OpenLibrary(
    Name="Custom Materials",
    Source="C:/FEA Analysis/v19/Custom Materials.xml")
engineeringData1 = system2.GetContainer(ComponentName="Engineering Data")
matl1 = engineeringData1.ImportMaterial(
    Name="MAT1",
    Source="C:/FEA Analysis/v19/Custom Materials.xml")
matl2 = engineeringData1.ImportMaterial(
    Name="MAT2",
    Source="C:/FEA Analysis/v19/Custom Materials.xml")
matlProp1 = matl1.GetProperty(Name="Density")
materialPropertyData1 = matlProp1.GetPropertyData(
    Name="Density",
    Qualifiers={"Definition": "", "Behavior": ""})
materialVariable1 = materialPropertyData1.GetVariable(Name="Density")
parameter1 = Parameters.CreateParameter(
    Entity=materialVariable1,
    PropertyName="ParameterValue",
    DisplayText="Density")
parameter1.ValueQuantityName = "Density"
matlProp2 = matl1.GetProperty(Name="Coefficient of Thermal Expansion")
materialPropertyData2 = matlProp2.GetPropertyData(
    Name="Coefficient of Thermal Expansion",
    Qualifiers={"Definition": "Secant", "Behavior": "Isotropic"})
materialVariable2 = materialPropertyData2.GetVariable(Name="Coefficient of
Thermal Expansion")
parameter2 = Parameters.CreateParameter(
    Entity=materialVariable2,
    PropertyName="ParameterValue",
    DisplayText="Coefficient of Thermal Expansion")
parameter2.ValueQuantityName = "InvTemp1"
matlProp3 = matl1.GetProperty(Name="Elasticity")
materialPropertyData3 = matlProp3.GetPropertyData(
    Name="Elasticity",
    Qualifiers={"Definition": "", "Behavior": "Isotropic", "Derive from":
"Young's Modulus and Poisson's Ratio"})
materialVariable3 = materialPropertyData3.GetVariable(Name="Young's Modulus")
parameter3 = Parameters.CreateParameter(
    Entity=materialVariable3,
    PropertyName="ParameterValue",
    DisplayText="Young's Modulus")
parameter3.ValueQuantityName = "Stress"
materialVariable4 = materialPropertyData3.GetVariable(Name="Poisson's Ratio")
parameter4 = Parameters.CreateParameter(
    Entity=materialVariable4,
    PropertyName="ParameterValue",
    DisplayText="Poisson's Ratio")
parameter4.ValueQuantityName = ""
matlProp4 = matl1.GetProperty(Name="Thermal Conductivity")
materialPropertyData4 = matlProp4.GetPropertyData(
    Name="Thermal Conductivity",
    Qualifiers={"Definition": "", "Behavior": "Isotropic"})
materialVariable5 = materialPropertyData4.GetVariable(Name="Thermal
Conductivity")
parameter5 = Parameters.CreateParameter(
    Entity=materialVariable5,
    PropertyName="ParameterValue",
    DisplayText="Thermal Conductivity")
parameter5.ValueQuantityName = "Thermal Conductivity"
matlProp5 = matl1.GetProperty(Name="Resistivity")
materialPropertyData5 = matlProp5.GetPropertyData(
    Name="Resistivity",
    Qualifiers={"Definition": "", "Behavior": "Isotropic"})
materialVariable6 = materialPropertyData5.GetVariable(Name="Resistivity")
parameter6 = Parameters.CreateParameter(
    Entity=materialVariable6,
    PropertyName="ParameterValue",
    DisplayText="Resistivity")
```

```
parameter6.ValueQuantityName = "Electrical Resistivity"
matlProp6 = matl2.GetProperty(Name="Density")
materialPropertyData6 = matlProp6.GetPropertyData(
    Name="Density",
    Qualifiers={"Definition": "", "Behavior": ""})
materialVariable7 = materialPropertyData6.GetVariable(Name="Density")
parameter7 = Parameters.CreateParameter(
    Entity=materialVariable7,
    PropertyName="ParameterValue",
    DisplayText="Density")
parameter7.ValueQuantityName = "Density"
matlProp7 = matl2.GetProperty(Name="Coefficient of Thermal Expansion")
materialPropertyData7 = matlProp7.GetPropertyData(
    Name="Coefficient of Thermal Expansion",
    Qualifiers={"Definition": "Secant", "Behavior": "Isotropic"})
materialVariable8 = materialPropertyData7.GetVariable(Name="Coefficient of
Thermal Expansion")
parameter8 = Parameters.CreateParameter(
    Entity=materialVariable8,
    PropertyName="ParameterValue",
    DisplayText="Coefficient of Thermal Expansion")
parameter8.ValueQuantityName = "InvTemp1"
matlProp8 = matl2.GetProperty(Name="Elasticity")
materialPropertyData8 = matlProp8.GetPropertyData(
    Name="Elasticity",
    Qualifiers={"Definition": "", "Behavior": "Isotropic", "Derive from":
"Young's Modulus and Poisson's Ratio"})
materialVariable9 = materialPropertyData8.GetVariable(Name="Young's Modulus")
parameter9 = Parameters.CreateParameter(
    Entity=materialVariable9,
    PropertyName="ParameterValue",
    DisplayText="Young's Modulus")
parameter9.ValueQuantityName = "Stress"
materialVariable10 = materialPropertyData8.GetVariable(Name="Poisson's Ratio")
parameter10 = Parameters.CreateParameter(
    Entity=materialVariable10,
    PropertyName="ParameterValue",
    DisplayText="Poisson's Ratio")
parameter10.ValueQuantityName = ""
matlProp9 = matl2.GetProperty(Name="Thermal Conductivity")
materialPropertyData9 = matlProp9.GetPropertyData(
    Name="Thermal Conductivity",
    Qualifiers={"Definition": "", "Behavior": "Isotropic"})
materialVariable11 = materialPropertyData9.GetVariable(Name="Thermal
Conductivity")
parameter11 = Parameters.CreateParameter(
    Entity=materialVariable11,
    PropertyName="ParameterValue",
    DisplayText="Thermal Conductivity")
parameter11.ValueQuantityName = "Thermal Conductivity"
matlProp10 = matl2.GetProperty(Name="Resistivity")
materialPropertyData10 = matlProp10.GetPropertyData(
    Name="Resistivity",
    Qualifiers={"Definition": "", "Behavior": "Isotropic"})
materialVariable12 = materialPropertyData10.GetVariable(Name="Resistivity")
```

```
parameter12 = Parameters.CreateParameter(
    Entity=materialVariable12,
    PropertyName="ParameterValue",
    DisplayText="Resistivity")
parameter12.ValueQuantityName = "Electrical Resistivity"
template3 = GetTemplate(
    TemplateName="Static Structural",
    Solver="ANSYS")
system3 = template3.CreateSystem(
    Position="Right",
    RelativeTo=system1)
geometryComponent1 = system3.GetComponent(Name="Geometry")
geometryComponent2 = system1.GetComponent(Name="Geometry")
geometryComponent1.ReplaceWithShare(
    TargetSystem=system3,
    ComponentToShare=geometryComponent2,
    SourceSystem=system1)
engineeringDataComponent1 = system3.GetComponent(Name="Engineering Data")
engineeringDataComponent2 = system2.GetComponent(Name="Engineering Data")
engineeringDataComponent1.ReplaceWithShare(
    TargetSystem=system3,
    ComponentToShare=engineeringDataComponent2,
    SourceSystem=system2)
template4 = GetTemplate(
    TemplateName="Electric",
    Solver="ANSYS")
system4 = template4.CreateSystem(
    Position="Right",
    RelativeTo=system3)
engineeringDataComponent3 = system4.GetComponent(Name="Engineering Data")
engineeringDataComponent3.ReplaceWithShare(
    TargetSystem=system4,
    ComponentToShare=engineeringDataComponent2,
    SourceSystem=system2)
solutionComponent1 = system3.GetComponent(Name="Solution")
modelComponent1 = system4.GetComponent(Name="Model")
solutionComponent1.TransferData(TargetComponent=modelComponent1)
system5 = template1.CreateSystem(
    Position="Right",
    RelativeTo=system4)
system6 = template4.CreateSystem(
    Position="Right",
    RelativeTo=system5)
geometryComponent3 = system6.GetComponent(Name="Geometry")
geometryComponent4 = system5.GetComponent(Name="Geometry")
geometryComponent3.ReplaceWithShare(
    TargetSystem=system6,
    ComponentToShare=geometryComponent4,
    SourceSystem=system5)
engineeringDataComponent4 = system6.GetComponent(Name="Engineering Data")
engineeringDataComponent4.ReplaceWithShare(
    TargetSystem=system6,
    ComponentToShare=engineeringDataComponent2,
    SourceSystem=system2)
```

```
print ("")
print ("################")
print ("Workbench Layout was created Successfully!!!")
print ("################")
print ("")
time.sleep(0.3)
print ("")
print ("")


print ("")
print ("################")
print ("Start reading from Master file.")
print ("################")
print ("")
time.sleep(0.3)
# IronPython imports to enable Excel interop
import clr
clr.AddReference("Microsoft.Office.Interop.Excel")
import Microsoft.Office.Interop.Excel as Excel

# Open Excel and the PassWorkbook
MasterExcelFile = Excel.ApplicationClass()
MasterExcelFile.Visible = False
MasterExcelWorkbook = MasterExcelFile.Workbooks.Open("C:/Users/hahmadi/Google
Drive/Codes/Project5/MaterialDatabase.xlsm")
MasterExcelWorksheet1 = MasterExcelWorkbook.Worksheets(1)

#Reading required Parameters from master excel file.
Density1 = MasterExcelWorksheet1.Range["B3"]
CoefThermalExpansion1 = MasterExcelWorksheet1.Range["B4"]
YoungsModulus1 = MasterExcelWorksheet1.Range["B5"]
PoissonRatio1 = MasterExcelWorksheet1.Range["B6"]
ThermalConductivity1 = MasterExcelWorksheet1.Range["B7"]
ElectricalResistance1 = MasterExcelWorksheet1.Range["B8"]
Density2 = MasterExcelWorksheet1.Range["C3"]
CoefThermalExpansion2 = MasterExcelWorksheet1.Range["C4"]
YoungsModulus2 = MasterExcelWorksheet1.Range["C5"]
PoissonRatio2 = MasterExcelWorksheet1.Range["C6"]
ThermalConductivity2 = MasterExcelWorksheet1.Range["C7"]
ElectricalResistance2 = MasterExcelWorksheet1.Range["C8"]

print ("")
print ("################")
print ("Setting Materials parameters in the analysis")
print ("################")
print ("")
time.sleep(0.3)
# Get the Workbench Parameters
P1Param = Parameters.GetParameter(Name="P1")
P2Param = Parameters.GetParameter(Name="P2")
P3Param = Parameters.GetParameter(Name="P3")
P4Param = Parameters.GetParameter(Name="P4")
P5Param = Parameters.GetParameter(Name="P5")
P6Param = Parameters.GetParameter(Name="P6")
```

```
P7Param = Parameters.GetParameter(Name="P7")
P8Param = Parameters.GetParameter(Name="P8")
P9Param = Parameters.GetParameter(Name="P9")
P10Param = Parameters.GetParameter(Name="P10")
P11Param = Parameters.GetParameter(Name="P11")
P12Param = Parameters.GetParameter(Name="P12")

# Assign values to the input parameters
P1Param.Expression = Density1.Value2.ToString() + " [kg m^-3]"
P2Param.Expression = CoefThermalExpansion1.Value2.ToString() + " [C^-1]"
P3Param.Expression = YoungsModulus1.Value2.ToString() + " [Pa]"
P4Param.Expression = PoissonRatio1.Value2.ToString()
P5Param.Expression = ThermalConductivity1.Value2.ToString() + " [W m^-1 C^-1]"
P6Param.Expression = ElectricalResistance1.Value2.ToString() + " [ohm m]"
P7Param.Expression = Density2.Value2.ToString() + " [kg m^-3]"
P8Param.Expression = CoefThermalExpansion2.Value2.ToString() + " [C^-1]"
P9Param.Expression = YoungsModulus2.Value2.ToString() + " [Pa]"
P10Param.Expression = PoissonRatio2.Value2.ToString()
P11Param.Expression = ThermalConductivity2.Value2.ToString() + " [W m^-1 C^-1]"
P12Param.Expression = ElectricalResistance2.Value2.ToString() + " [ohm m]"

#Quit from Master Excel file.
MasterExcelWorkbook.Close()
MasterExcelFile.Quit()
del MasterExcelFile

print ("")
print ("################")
print ("Reading from Master file was terminared Successfully!!!")
print ("################")
print ("")
time.sleep(0.3)

print ("")
print ("################")
print ("Materials parameters were set Successfully!!!")
print ("################")
print ("")
time.sleep(0.3)

print ("")
print ("################")
print ("Creating Structural Geometry.")
print ("################")
print ("")
#Creating Geometry
#Getting geometry container
system1 = GetSystem(Name="Geom")
geometry1 = system1.GetContainer(ComponentName="Geometry")

#Oppening DesignModeler
geometry1.Edit(Interactive=True)

#Sending Javascript command to DesignModeler
```

```
geometry1.SendCommand(Command="""WB.AppletList.Applet(\"DSApplet\").App.Script.
doToolsRunMacro(\"C:/Users/hahmadi/Google Drive/Codes/Project5/
SphereParticlesGeometryPre.js\")""")


#Closing DesignModeler
geometry1.Exit()

print ("")
print ("################")
print ("Geometry Created Successfully!!!")
print ("################")
print ("")
time.sleep(0.3)

#Refresh Workbench before openning the mechanical.
Refresh()

print ("")
print ("################")
print ("Starting Mechanical Analysis.")
print ("################")
print ("")
time.sleep(0.3)
#Performing Mechanical Analysis
#Getting Mechanical Model container
system2 = GetSystem(Name="SYS")
model1 = system2.GetContainer(ComponentName="Model")

#Oppening Mechanical GUI
model1.Edit(Interactive=True)

print ("")
print ("################")
print ("Prepairing Mechanical FE model.")
print ("################")
print ("")
time.sleep(0.3)

#Sending Pre Javascript macro to mechanical model
model1.SendCommand(Command="""WB.AppletList.Applet(\"DSApplet\").App.Script.doT
oolsRunMacro(\"C:/Users/hahmadi/Google
Drive/Codes/Project5/MechanicalMacro_Pre.js\")""")

print ("")
print ("################")
print ("Solving Mechanical FE model.")
print ("################")
print ("")
time.sleep(0.3)

#Sending Post Javascript macro to mechanical model
model1.SendCommand(Command="""WB.AppletList.Applet(\"DSApplet\").App.Script.doT
oolsRunMacro(\"C:/Users/hahmadi/Google
Drive/Codes/Project5/MechanicalMacro_Post.js\")""")
```

```
#Closing Mechanical GUI
model1.Exit()

#Refresh Workbench before openning the Electrical analysis.
Refresh()
Update()

print ("")
print ("################")
print ("Starting Electrical Analysis.")
print ("################")
print ("")
#Performing Electrical Analysis
#Getting Electrical Model container
system3 = GetSystem(Name="SYS 1")
model2 = system3.GetContainer(ComponentName="Model")

#Oppening Mechanical GUI
model2.Edit(Interactive=True)

print ("")
print ("################")
print ("Extracting Geometry Data.")
print ("################")
print ("")
time.sleep(0.3)

#Sending Pre Javascript macro to Electrical model
model2.SendCommand(Command="""WB.AppletList.Applet(\"DSApplet\").App.Script.doT
oolsRunMacro(\"C:/Users/hahmadi/Google
Drive/Codes/Project5/ExtractingGeometryData.js\")""")

#Closing Mechanical GUI
model2.Exit()

print ("")
print ("################")
print ("Creating Electrical Geometry.")
print ("################")
print ("")
#Creating Geometry
#Getting geometry container
system4 = GetSystem(Name="Geom 1")
geometry2 = system4.GetContainer(ComponentName="Geometry")

#Oppening DesignModeler
geometry2.Edit(Interactive=True)

#Sending Javascript command to DesignModeler
geometry2.SendCommand( Command = """var DM =
ag.wb.AppletList.Applet("AGApplet").App;
var lv = ag.listview

//Changing the unit to Meter and Degree
```

```
var Unit = agb.SetSessionUnits(agc.UnitMeter, agc.UnitDegree, agc.N0);

//Reading volume fraction and iteration information from material database
Excel file.
var MasterExcel = new ActiveXObject("Excel.Application");
MasterExcel.Visible = false;
var MasterExcel_file = MasterExcel.Workbooks.Open("C:/Users/hahmadi/Google
Drive/Codes/Project5/MaterialDatabase.xlsm");
var MasterExcel_sheet = MasterExcel_file.Worksheets("Sheet1");

//Setting current Volume fraction
var CurrentVolumeFraction = MasterExcel_sheet.Cells(25, 2).Value;

//Setting current Iteration
var CurrentIteration = MasterExcel_sheet.Cells(26, 2).Value;

//Setting RVE size
var RVESize = MasterExcel_sheet.Cells(11, 2).Value;

//Setting Plate Thickness
var PlateThickness = MasterExcel_sheet.Cells(10, 2).Value;

//Extracting Mechanical-Electrical Iteration
var Iteration = MasterExcel_sheet.Cells(27, 2).Value;

//Closing Master Excel file
MasterExcel_file.Close();
MasterExcel.Application.Quit()

//Reading Displacement from master excelfile information from master Excel
file.
var MasterExcel2 = new ActiveXObject("Excel.Application");
MasterExcel2.Visible = false;
var MasterExcel2_file = MasterExcel2.Workbooks.Open("C:/Users/hahmadi/Google
Drive/Codes/Project5/MasterResultFile.xlsx");
var MasterExcel2_sheet = MasterExcel2_file.Worksheets("Sheet1");

//Setting RVE displacement
var RVESizeDisp = MasterExcel2_sheet.Cells(Iteration + 1, 1).Value;

//Closing Master Excel file
MasterExcel2_file.Close();
MasterExcel2.Application.Quit()

//RVE total lenght
var RVETotalLenght = RVESize + RVESizeDisp

//Reading RVE height from Excel file.
var RandomNumberExcel = new ActiveXObject("Excel.Application");
RandomNumberExcel.Visible = false;
var RandomNumberExcel_file =
RandomNumberExcel.Workbooks.Open("C:/Users/hahmadi/Google
Drive/Codes/Project5/ParticlesLocation/Iteration" + Iteration + ".xlsx");
var RandomNumberExcel_sheet = RandomNumberExcel_file.Worksheets("Sheet1");
```

```
//Creating path for saving random number excel file
var Path = "C:/Users/hahmadi/Google Drive/Codes/Project5/VF" +
CurrentVolumeFraction + "/Geometries/VF" + CurrentVolumeFraction + "_Geometry_"
+ CurrentIteration + ".xlsm"

// Set random number Excel file
var RandomNumberExcel2 = new ActiveXObject("Excel.Application");
RandomNumberExcel2.Visible = false;
var RandomNumberExcel2_file = RandomNumberExcel2.Workbooks.Open(Path);
var RandomNumberExcel2_sheet = RandomNumberExcel2_file.Worksheets("Sheet1");

//Extracting RVE height and wide (Added 0.002 for making sure that all
particles will fit within RVE)
var RVEHeight = RandomNumberExcel_sheet.Cells(1, 8).Value;

//Extracting number of Components from excel file created in previous step
var NumberOfComponents = RandomNumberExcel_sheet.Cells(1, 9).Value;
var NumberOfParticles = NumberOfComponents - 3;
//Creating RVE
CreatingBox(RVEHeight, RVETotalLenght);

//Creating Back Plate
CreatingBox2(RVEHeight, PlateThickness)

//Creating Front Plate
CreatingBox3(RVEHeight, PlateThickness, RVETotalLenght)

//Raw numerator in random number Excel file
var j = 1
while (RandomNumberExcel_sheet.Cells(j, 1).Value != 0) {
    ReadDataAndCreateSphere(j);
    j = j + 1;
}

//Closing random number Excel file
RandomNumberExcel_file.Close();
RandomNumberExcel.Application.Quit()

//Closing random number Excel file
RandomNumberExcel2_file.Close();
RandomNumberExcel2.Application.Quit()

//Generate all the bodies
ag.b.Regen();

//creating Boolean from bodies
var Pat = ag.gui.CreateBoolean()
ag.listview.ActivateItem("Operation");
ag.listview.ItemValue = "Subtract";

//Selecting body for being selected as target
ag.bodyPick;
Select = ag.m.Select3dBody(0, 0, 0, 0);

//Modifing Target section of boolean
```

```
ag.listview.ActivateItem("Target Bodies");
ag.listview.ItemValue = "Apply";

//Selecting Bodies for being selected as tools
ag.bodyPick;
for (i = 3; i <= j + 2; i++) {
    ag.m.Select3dBody(i, 0, 0, 0);
}

//Modifing Tooll section of boolean
ag.listview.ActivateItem("Tool Bodies");
ag.listview.ItemValue = "Apply";

//Modifiing Section to preserve all bodies.
ag.listview.ActivateItem("Preserve Tool Bodies?");
ag.listview.ItemValue = "Yes";

ag.b.Regen();


function ReadDataAndCreateSphere(x) {
    var Raw = x
    var Sphere = [0, 0, 0, 0, 0, 0, 0]
    for (i = 1; i < 8; i++) {
        if (i != 7) {
            Sphere[i - 1] = RandomNumberExcel_sheet.Cells(x, i).Value;
        }
        else {
            Sphere[i - 1] = RandomNumberExcel_sheet.Cells(x, i).Value;
        }
    }
    feature = ag.gui.CreatePrimitive(1);
    lv.ActivateItem("Operation");
    lv.ItemValue = "Add Frozen";
    lv.ActivateItem("FD3, Origin X Coordinate");
    lv.ItemValue = "" + Sphere[0];
    lv.ActivateItem("FD4, Origin Y Coordinate");
    lv.ItemValue = "" + Sphere[1];
    lv.ActivateItem("FD5, Origin Z Coordinate");
    lv.ItemValue = "" + Sphere[2];
    lv.ActivateItem("FD6, Radius (>0)");
    lv.ItemValue = "" + Sphere[6];


}
function selectNode(target) {
    var Nodes = DS.Tree.AllObjects;
    var count = Nodes.Count;
    var name, current, ID;
    for (var i = 1; i <= count; i++) {
        current = Nodes(i);
        name = current.Name;
        if (name == target) {
            ID = current.ID
            DS.Tree.FirstActiveObjectID = ID
            //reset mechanical to adjust ListView
            SC.wb_OnActivate()
            return;
        }
    }
    return false;
}

//Creating RVE
function CreatingBox(x, y) {
    var Box = [0, 0, 0, y, x, x]

    feature = ag.gui.CreatePrimitive(2);
    lv.ActivateItem("FD3, Point 1 X Coordinate");
    lv.ItemValue = "" + Box[0];
    lv.ActivateItem("FD4, Point 1 Y Coordinate");
    lv.ItemValue = "" + Box[1];
    lv.ActivateItem("FD5, Point 1 Z Coordinate");
    lv.ItemValue = "" + Box[2];
    lv.ActivateItem("FD6, Diagonal X Component");
    lv.ItemValue = "" + Box[3];
    lv.ActivateItem("FD7, Diagonal Y Component");
    lv.ItemValue = "" + Box[4];
    lv.ActivateItem("FD8, Diagonal Z Component");
    lv.ItemValue = "" + Box[5];

}

//Creating Back Plate
//X is RVE size and Y is Plate thickness
function CreatingBox2(x, y) {
    var Box = [-1 * y, 0, 0, y, x, x]

    feature = ag.gui.CreatePrimitive(2);
    lv.ActivateItem("Operation");
    lv.ItemValue = "Add Frozen";
    lv.ActivateItem("FD3, Point 1 X Coordinate");
    lv.ItemValue = "" + Box[0];
    lv.ActivateItem("FD4, Point 1 Y Coordinate");
    lv.ItemValue = "" + Box[1];
    lv.ActivateItem("FD5, Point 1 Z Coordinate");
    lv.ItemValue = "" + Box[2];
    lv.ActivateItem("FD6, Diagonal X Component");
    lv.ItemValue = "" + Box[3];
    lv.ActivateItem("FD7, Diagonal Y Component");
    lv.ItemValue = "" + Box[4];
    lv.ActivateItem("FD8, Diagonal Z Component");
    lv.ItemValue = "" + Box[5];

}

//Creating Front Plate
//X is RVE size and Y is Plate thickness
function CreatingBox3(x, y, z) {
    var Box = [z, 0, 0, y, x, x]
    feature = ag.gui.CreatePrimitive(2);
    lv.ActivateItem("Operation");
    lv.ItemValue = "Add Frozen";
    lv.ActivateItem("FD3, Point 1 X Coordinate");
    lv.ItemValue = "" + Box[0];
    lv.ActivateItem("FD4, Point 1 Y Coordinate");
    lv.ItemValue = "" + Box[1];
    lv.ActivateItem("FD5, Point 1 Z Coordinate");
    lv.ItemValue = "" + Box[2];
    lv.ActivateItem("FD6, Diagonal X Component");
    lv.ItemValue = "" + Box[3];
    lv.ActivateItem("FD7, Diagonal Y Component");
    lv.ItemValue = "" + Box[4];
    lv.ActivateItem("FD8, Diagonal Z Component");
    lv.ItemValue = "" + Box[5];

}""")

#Closing DesignModeler
geometry2.Exit()

print ("")
print ("################")
print ("Geometry Created Successfully!!!")
print ("################")
print ("")
time.sleep(0.3)

#Refresh Workbench before openning the mechanical.
Refresh()


print ("")
print ("################")
print ("Starting Electrical Analysis.")
print ("################")
print ("")
time.sleep(0.3)
#Performing Mechanical Analysis
#Getting Mechanical Model container
system5 = GetSystem(Name="SYS 2")
model3 = system5.GetContainer(ComponentName="Model")

#Oppening Mechanical GUI
model3.Edit(Interactive=True)

print ("")
print ("################")
print ("Prepairing Electrical FE model.")
print ("################")
print ("")
time.sleep(0.3)

#Sending Pre Javascript macro to mechanical model
```

```
model3.SendCommand(Command="""WB.AppletList.Applet(\"DSApplet\").App.Script.doT
oolsRunMacro(\"C:/Users/hahmadi/Google
Drive/Codes/Project5/ElectricalMacro_Pre.js\")""")

print ("")
print ("################")
print ("Solving Electrical FE model.")
print ("################")
print ("")
time.sleep(0.3)

#Sending Post Javascript macro to mechanical model
model3.SendCommand(Command="""WB.AppletList.Applet(\"DSApplet\").App.Script.doT
oolsRunMacro(\"C:/Users/hahmadi/Google
Drive/Codes/Project5/ElectricalMacro_Post.js\")""")

#Closing Mechanical GUI
model3.Exit()

#Refresh Workbench before openning the Electrical analysis.
Refresh()
Update()
```