Abstract

SRISAWADI, SASITORN. Computational Design and Rapid Tooling for Preferentially Aligned Short Fiber Composite Component Fabrication. (Under the direction of Ola L A Harrysson and Denis R Cormier.)

It is well understood that mechanical properties in discontinuously reinforced polymer matrix composite materials are strongly correlated with the degree of fiber alignment in the matrix. To make the greatest use of the reinforcing phase's strength, the fibers must be aligned with the stress or strain axis. The field-aided microtailoring (FAiMTa) process is one promising method for doing this. FAiMTa uses principles of dielectrophoresis to preferentially align particles or fibers within a matrix. To achieve the preferred fiber orientation, an interdigitated electrode network must be integrated into the mold halves which can be fabricated by additive manufacturing (AM) processes. However, the process of determining the preferred fiber arrangements and electrode locations is very challenging. This dissertation presents algorithms to semi-automate the interdigitated electrode design process. The algorithm has been implemented in the Solidworks CAD system and is demonstrated as well.

In order to validate the benefit of preferential fiber alignment, finite element analysis (FEA) models have been created in which the geometric elements are assigned orthotropic stiffness values representative of what would be expected when reinforcing fibers for that element are preferentially aligned to best resist expected loading patterns. These models are compared with the initial models in which the fibers throughout the component are randomly aligned. In this study, Halpin-Tsai model was used to predict the stiffness of unidirectional

short fiber reinforced composites. A mixed model was used to predict the stiffness of randomly aligned composite used in the initial models and the regions of the final models that do not require fiber alignment. The procedures have been implemented on three models and are demonstrated in this paper. Strain magnitude histograms are used to evaluate the effectiveness of preferential fiber alignment.

Rapid tooling (RT) is an alternative to conventional tool making. RT can significantly reduce lead times, and can also be used to produce parts with complex geometries via assembled tooling. This dissertation explores the use of RT to produce short fiber reinforced composite components with preferentially aligned fiber orientation. Specifically, the objective is to employ the direct-write process in order to print conformal electrode-integrated networks onto the customized RT molds.

Computational Design and Rapid Tooling for Preferentially Aligned Short Fiber Composite Component Fabrication

by
Sasitorn Srisawadi

A dissertation submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the Degree of
Doctor of Philosophy

Industrial Engineering

Raleigh, North Carolina

2011

APPROVED BY:

Dr. Ola L A Harrysson Chair of Advisory Committee	Dr. Denis R Cormier Co-Chair of Advisory Committee
Dr. Harvey A West II	Dr. Jagannadham Kasichainula

Dedication

To my parents

Biography

Sasitorn Srisawadi was born on October 8, 1981 in Bangkok, Thailand. In 2001, she graduated from Chulalongkorn University with a Bachelor of Science in Industrial Engineering. After graduation, she went to work at Textile Prestige Company in Bangkok, Thailand. A year later, Sasitorn was awarded with a full scholarship by the Thai Government to pursue studying at the master's and doctoral levels in Industrial Engineering.

In 2003, she moved to Raleigh, North Carolina to pursue her graduate studies. At North Carolina State University, she received an M.S. in Industrial Engineering in 2006. Afterwards, Sasitorn continued with her Ph.D. study under the guidance of Dr. Denis R Cormier and Dr. Ola L A Harrysson. During her study, she was also supported through research assistantships and teaching assistantships with Edward P. Fitts Department of Industrial and Systems Engineering. Her research interests include additive manufacturing, rapid tooling, composite materials, and manufacturing process development.

Acknowledgements

I would like to thank my advisors, Dr. Denis R Cormier and Dr. Ola L A Harrysson for their support and the insights they have, which made this dissertation possible. I am grateful for the time that they spent guiding my research and being patient with my dissertation writing. I would also like to thank the members of my committee: Dr. Harvey A West II and Dr. Jagannadham Kasichainula for their contributions. In addition, I thank Dr. Denis Marcellin-Little and Bryan Laffitte for their time. Appreciation is also extended to Edward P. Fitts Department of Industrial and Systems Engineering for providing me the opportunities to be here.

Most importantly, I would like to thank my parents for their unconditional love and infinite willingness to give me their best. Even though they are 9,000 miles away, they are always with me. In addition, I would like to thank Uncle Tum and Aunt Tam for their care and generous support. I also would like to thank my sister Kade for always being there for me and literally being here with me in my first week in Raleigh.

Lastly, I would like to express my thanks to my wonderful friends: Na, Nat, Ning, Yi, the Leksrisompong family, Pat, Nina, Orm, Chelsea, Kiddee, Joey, and Wang for being by my side in good times and bad times. I also would like to thank my friends in ISE department: Big, May, Top, Pom, Noom, Guha, Li, Tim and Hazman for their advice and the time we spent together on struggling with projects and exams. My life in Raleigh could not be completed without all of them.

Table of Contents

List of Tables

List of Figures

Chapter 1 Introduction

As the demand for high strength light weight materials continues to increase, so does the demand for development of high performance materials. Of particular interest are development efforts to improve polymer composite materials which are known to have very high strength-to-weight ratios. The high strength of composites is primarily attributable to a very strong reinforcement phase which is surrounded by the matrix material. Polymer matrix composites have begun to emerge as candidates to replace heavier metallic alloys used in structural components.

Military and commercial aviation is a good example of an industry aggressively seeking out materials having high specific strength. Composite use in both skins and structural components has increased with the aim of weight reduction, improved fuel economy, and decreased carbon emissions. Inexpensive metal alloys, such as aluminum 2024 and 7075 have historically been used for construction of aircraft elements such as skins and structural elements. However, those metals are relatively heavy compared to polymer matrix composites. As a result, light-weight composites have become a preferred option for these applications.

Composite materials consist of a discrete constituent (the reinforcement) distributed in a continuous phase (the matrix). They derive their distinguishing characteristics from the properties and behavior of their constituents, from the geometry and arrangement of the constituents, and from the properties of the boundaries (interfaces) between the constituents. Exceptional specific strength and stiffness are not the only advantages of composites. Corrosion resistance is another possibility in marine application for lower maintenance requirements [1]. Composites with the proper combination of matrix and reinforcement can also provide better thermal and electrical insulation, energy absorption [2], thermal dimensional accuracy [3], and so on.

Typically, the material properties needed by any particular applications can be tailored by the types of reinforcement, matrix and optional coupling agents. One of the most common forms of a composite is the polymer matrix composite (PMC). PMC’s are particularly useful in applications where weight reduction is crucial such as aircraft components, bicycle frames, and automobile bodies. In addition, extensive choices for the polymer matrix allow the designers to modify the properties of the composite.

Polymer matrix composites (PMC) can be defined as composites with polymer as the continuous phase that surrounds and supports the reinforcing phase. PMC’s can be classified based on the types of polymer matrix available. These include thermoplastics, thermosets, and elastomers. PMC’s can also be classified on the basis of the nature of reinforcing phase as follows: particle reinforced, fiber reinforced, dispersion strengthened, laminated, etc. **Figure 1** illustrates the common types of reinforcement phase. When a composite’s reinforcing phase is aligned in a particular direction, the strength and stiffness of the composite in the direction of alignment is much higher than it would be if the fibers were randomly oriented. However, the random arrangement of particles/short fibers reinforcement results in parts whose properties are nearly equal in all directions [4].

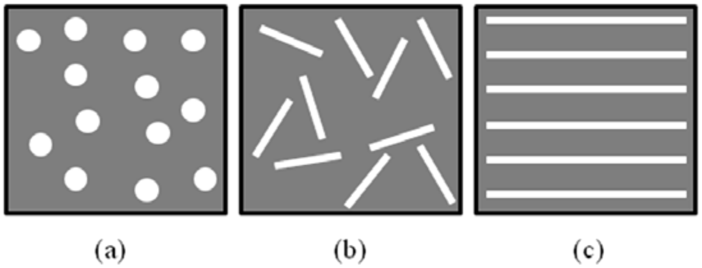


Figure 1 Polymer matrix composites reinforced with (a) random particles; (b) random short fibers; (c) aligned continuous fibers.

A relatively soft material such as polymer can be strengthened by incorporating carbon fibers which are notable for their superior mechanical properties. Carbon fibers are commercially available with a variety of tensile moduli, ranging from 270 GPa on the low side to 517 GPa on the high side [4]. Typical epoxy resins used as the matrix material have tensile moduli of 2.75 to 4.10 GPa [4]. Carbon fiber composites typically have numerous advantages over conventional materials, mostly because of the exceptionally high specific strengths of the carbon fibers. In addition, the materials also possess good fatigue resistance, creep resistance, vibration damping ability, and high electromagnetic interference (EMI) shielding effectiveness [5]. As a result, carbon fiber composites are used for many applications.

Carbon fiber composites have been used for many years in the aerospace industry where the high cost of carbon fiber could be offset by increased fuel efficiency. As carbon fiber production techniques have improved and the price of carbon fibers has come down, the use of carbon fibers has expanded to other industries including automotive, sports, biomedical, construction and so on. For each particular application, the choice of matrix material and reinforcing phase is customized to serve the specific needs of the application.

For modern commercial aircraft (e.g. the fin of the Airbus A310, the horizontal stabilizer of the Airbus A340, and the Boeing 777 [6]), the polymer matrix composite is designed to reduce the aircraft weight and epoxy resins are most widely used as the matrix material. Epoxy-based composites, usually with continuous carbon fiber, are also very common for automotive parts and sports equipment for weight reduction purposes. In addition, several types of polymer matrix composites have been used in biomedical applications [7] and civil engineering and construction applications [8].

The combination of carbon fiber reinforcement phase in an epoxy matrix has several appealing properties. One of the most important motivations to use these composites is the outstanding specific strength. The unidirectional mechanical properties of continuous carbon fiber/epoxy composite, including tensile strength and stiffness, greatly exceed those of 7075-T6 aluminum and Ti-6Al-4V as shown in **Table 1**. In addition to having high strength, carbon fiber/epoxy composites have a density that is approximately 40% less than aluminum.

Continuous fibers certainly offer the best mechanical properties among other forms of reinforcements. However, the fabrication processes for such materials are complex and challenging for mass production due to the high labor content and long curing cycles. Having short or discontinuous fibers rather than continuous ones can be very beneficial in terms of production cost and time when components must be mass produced in various processes including injection molding, extrusion, and resin transfer molding. Due to the random orientation of short fibers, short fiber composites have lower strength and stiffness values than continuously reinforced composites. To compensate for the shortcoming, researchers have sought out ways to improve the mechanical properties of the composites with short carbon fibers.

Table 1 Mechanical properties and density of unidirectional carbon fiber/epoxy composite compared to metals [5]

Material	Strength (MPa)		Tensile modulus (GPa)	Density (g/cm^3)
	Tension	Compression		
Epoxy/carbon fibers AS-4	1482	1227	145	1.55
Epoxylcarbon fibers HMS	1276	1020	207	1.63
Aluminum (7075-T6)	572	-	69	2.76
Titanium (6A1-4V)	1103	-	114	4.43

When a discontinuous fiber-reinforced composite has tensile load applied, the load is transferred to the fiber by a shearing mechanism between fibers and matrix. The distributions of longitudinal stresses for different fiber length are illustrated in **Figure 2**. When the fiber length is smaller than the minimum length as shown in **Figure 2(a)**, the composite will fail at a stress lower than the fiber's ultimate strength. The most optimized case possible is demonstrated in **Figure 2(b)** where the maximum stress reaches the ultimate fiber strength. However, this situation is still less than ideal because the majority of the fiber length is bearing a stress that is much lower than the fiber's ultimate strength. **Figure 2(c)** shows the situation where the fiber length is larger than the minimum length. This allows a greater portion of the fiber to be loaded at the ultimate strength for effective fiber reinforcement.

Therefore, increasing the fiber length to the extent possible is preferable to achieve a stronger composite system for any specific type of fiber material and diameter.

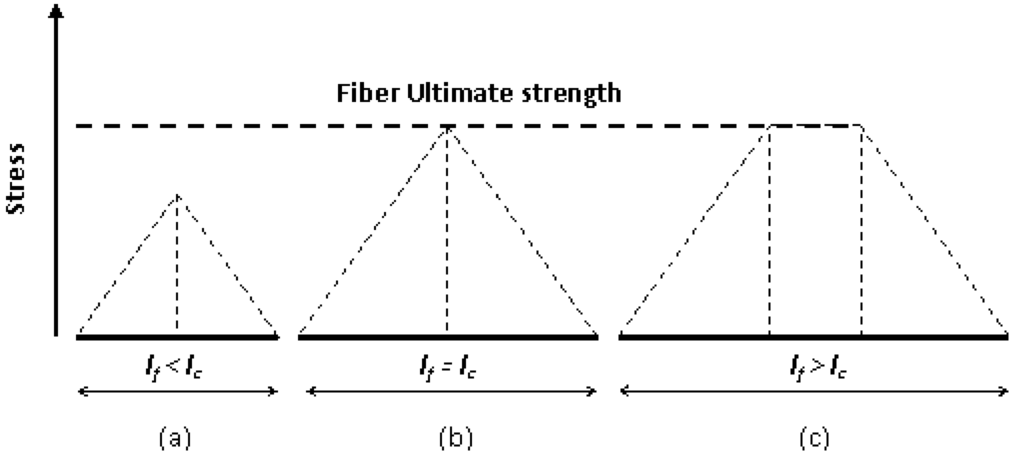
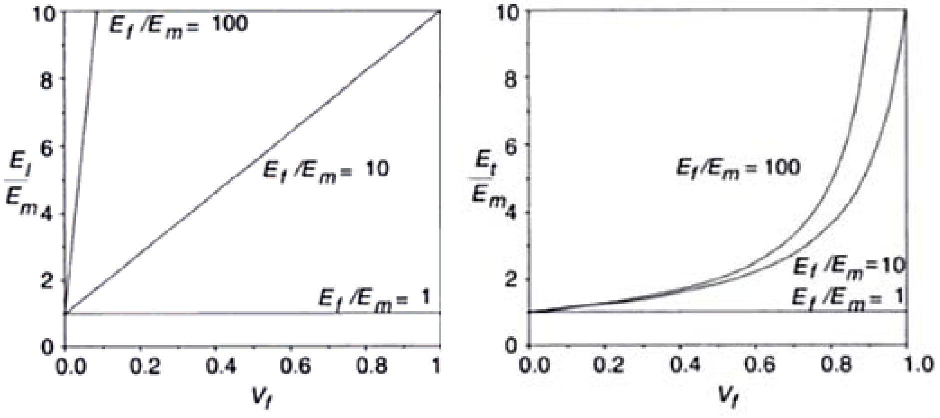


Figure 2 Significance of critical fiber length on the longitudinal stresses

An ideal transfer of load between the fibers and the matrix is needed to successfully utilize the strength of the fibers. Therefore, the chemical and strength characteristics of the interface between fiber and matrix are particularly important in determining the properties of a composite [9]. There are two principle failure mechanisms that occur at the interface of the fiber and the matrix, namely fiber pull out and debonding [10]. Better adhesion between the fiber and the matrix contributes to better mechanical properties as the fiber and the matrix are bonded tightly, thus preventing the premature failure at the interface. Adhesion can be improved by surface treatments such as grafting, coatings, oxidation, and plasma polymerization [11].

The strength of a composite material can also be estimated from the predicted modulus of elasticity based on the rule of mixtures. Consequently, a parallel model can be used to constitute an upper bound for the modulus in the longitudinal and transverse direction [12]. The effect of increasing volume fraction is more noticeable if the fiber modulus is drastically larger than that of the matrix, as shown in **Figure 3**. Although increasing fiber volume fraction is rewarding, it can greatly increase the complexity of the fabrication processes because the viscosity of the mixture rapidly rises when more fiber is added.



(a) (b)

Figure 3 Normalized elastic moduli in (a) longitudinal and (b) transverse directions [12]

The fiber orientation distribution also has a strong influence over the properties of discontinuous fiber-reinforced composites [13, 14]. Experimental data, as shown in **Figure 4**, confirms that the strongest glass fiber/epoxy composite can be obtained by aligning the short fibers along the stress axis [15]. The decrease in modulus as the misorientation angle increases reflects the reduction in reinforcing efficiency. The reinforcing efficiency of the three-dimensional randomly aligned fiber is theoretically just 20% of the longitudinally aligned one [9].

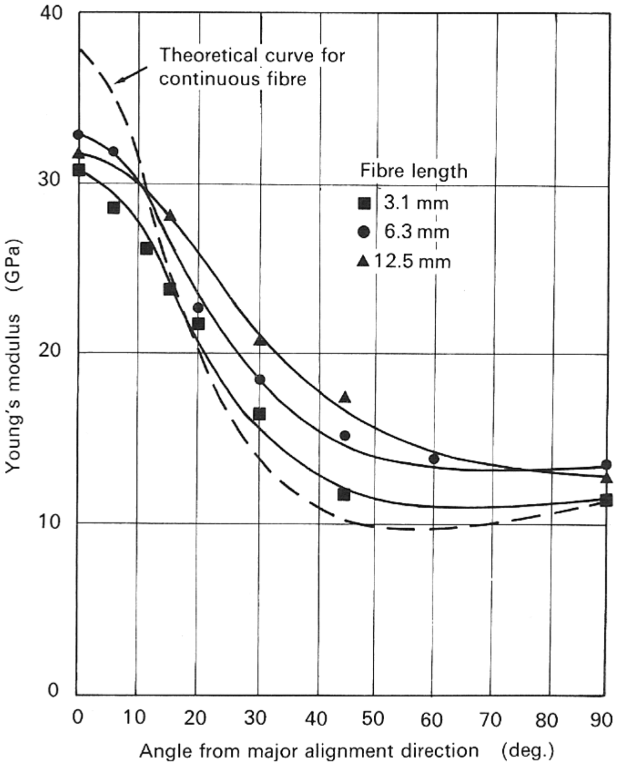


Figure 4 Young's modulus as a function of the fiber orientation angle [15]

Not only are the mechanical properties affected by the degrees of fiber orientation, but the thermal and electrical properties are likewise affected [16, 17]. Due to the nature of common fabrication processes, the fiber orientation distribution is typically random. As a result, the process yields composite parts with lower strength and stiffness than would be the case if the fibers could be aligned. Hence, the challenge of producing preferentially aligned fiber reinforced composite arises, primarily to favor the load transfer. To achieve the capability of precisely controlling the material properties, a process must be developed to manipulate the short carbon fibers within the composite during processing such that a preferred fiber orientation can be achieved.

A relatively new method of producing molded parts with oriented short fibers via the process of dielectrophoresis has recently been described [18]. This method employs an interdigitated electrode network embedded in two halves of the mold. The interdigitated electrode network is connected to a high voltage supply to induce a non-uniform electric field which is intended to manipulate the short fibers by the dielectrophoresis effect. While this approach shows considerable promise, the process of designing both the part and the tooling is complex. This dissertation focuses on the challenges associated with computational design of load-bearing components with oriented short fibers as well as the modified tooling capable of producing these parts. Chapter 2 describes the geometric algorithms that design the electrode networks, which are customized according to the working conditions of the components. The chapter includes the demonstration of the semi-automated procedure in SolidWorks®. Chapter 3 describes the methodology of the rapid tooling technique for the designed mold halves. The electrode networks that were designed by the algorithms in Chapter 2 were physically fabricated. Chapter 4 presents the validation of the effect of the locally aligned fibers using FEA, which incorporate the orthotropic strength resulting from fiber alignment. Finally, Chapter 5 concludes that the algorithms successfully design the electrode and the molds can be readily fabricated. The FEA proves that the components are improved by the preferentially aligned fibers. Future research was also suggested in this chapter.

References

1. Mouritz A, Gellert E, Burchill P, Challis K. Review of advanced composite structures for naval ships and submarines. Composite Structures 2001;53(1):21-42.

2. Hussain F, Hojjati M, Okamoto M, Gorga RE. Review article: polymer-matrix nanocomposites, processing, manufacturing, and application: an overview. J Composite Mater 2006;40(17):1511.

3. Kim RY, Crasto AS, Schoeppner GA. Dimensional stability of composite in a space thermal environment. Composites Sci Technol 2000;60(12-13):2601-2608.

4. Mallick P. Fiber-reinforced composites: materials, manufacturing, and design: CRC Press, 1993.

5. Chung DDL. Carbon fiber composites: Butterworth-Heinemann, 1994.

6. Williams JC, Starke EA. Progress in structural materials for aerospace systems1. Acta Materialia 2003;51(19):5775-5799.

7. Ramakrishna S, Mayer J, Wintermantel E, Leong KW. Biomedical applications of polymer-composite materials: a review. Composites Sci Technol 2001;61(9):1189-1224.

8. Bakis CE, Cosenza E, Lesko J, Machida A. Fiber-reinforced polymer composites for construction—state-of-the-art review. J Composite Constr 2002;6:73.

9. Matthews FL, Rawlings RD. Composite materials: engineering and science: Woodhead Publishing, 1999.

10. Kelly A. Interface effects and the work of fracture of a fibrous composite. Proceedings of the Royal Society of London.A.Mathematical and Physical Sciences 1970;319(1536):95.

11. Dilsiz N, Ebert E, Weisweiler W, Akovali G. Effect of plasma polymerization on carbon fibers used for fiber/epoxy composites. J Colloid Interface Sci 1995;170(1):241-248.

12. Astrom BT. Manufacturing of polymer composites: CRC, 1997.

13. Pipes RB, McCullough RL, Taggart DG. Behavior of discontinuous fiber composites: Fiber orientation. Polymer Composites 1982;3(1):34-39.

14. De SK, White JR. Short fibre-polymer composites. Cambridge, England: Woodhead Pub., 1996.

15. Kacir L, Ishai O, Narkis M. Oriented short glass-fiber composites. IV. Dependence of mechanical properties on the distribution of fiber orientations. Polymer Engineering & Science 1978;18(1):45-52.

16. Taipalus R, Harmia T, Zhang M, Friedrich K. The electrical conductivity of carbon-fibre-reinforced polypropylene/polyaniline complex-blends: experimental characterisation and modelling. Composites Sci Technol 2001;61(6):801-814.

17. Tezvergil A, Lassila LVJ, Vallittu PK. The effect of fiber orientation on the thermal expansion coefficients of fiber-reinforced composites. Dental Materials 2003;19(6):471-477.

18. Modak S. Freeform fiber alignment in a polymer matrix using conformal electrode field networks. 2008:71.

Chapter 2 Computational Design of Electrode Networks for Preferentially Aligned Short Fiber Composite Component Fabrication via Dielectrophoresis

Abstract

Finite Element Analysis (FEA) is often used to identify local stress/strain concentrations where a component is likely to fail. In order to reduce the degree of strain concentration, component thickness can be increased in those regions, or a stronger material can be used. In short fiber reinforced composite materials, strength and stiffness can be increased through proper fiber alignment. The field-aided microtailoring (FAiMTa) process is one promising method for doing this. FAiMTa uses principles of dielectrophoresis to preferentially align particles or fibers within a matrix. To achieve the preferred fiber orientation, an interdigitated electrode network must be integrated into the mold halves which can be fabricated by additive manufacturing (AM) processes. However, the process of determining the preferred fiber arrangements and electrode locations can be very challenging. This dissertation presents algorithms to semi-automate the interdigitated electrode design process. The algorithm has been implemented in the Solidworks CAD system and is demonstrated.

2.1. Introduction

Polymer matrix composite (PMC) materials are commonly specified due to their excellent mechanical properties and light weight. While continuous fibers certainly offer the best mechanical properties among other forms of reinforcements, the use of short or discontinuous fibers can be advantageous when components must be mass produced. Where strength and stiffness are concerned, experiments and theoretical studies have shown that the strongest and stiffest short fiber/epoxy composites are obtained when the short fibers are aligned with one another. Conversely, there is a marked decrease in modulus as the degree of fiber alignment decreases [1-4]. The ideal process is one that produces the excellent material properties with the higher production rates normally associated with short fiber reinforced PMC's.

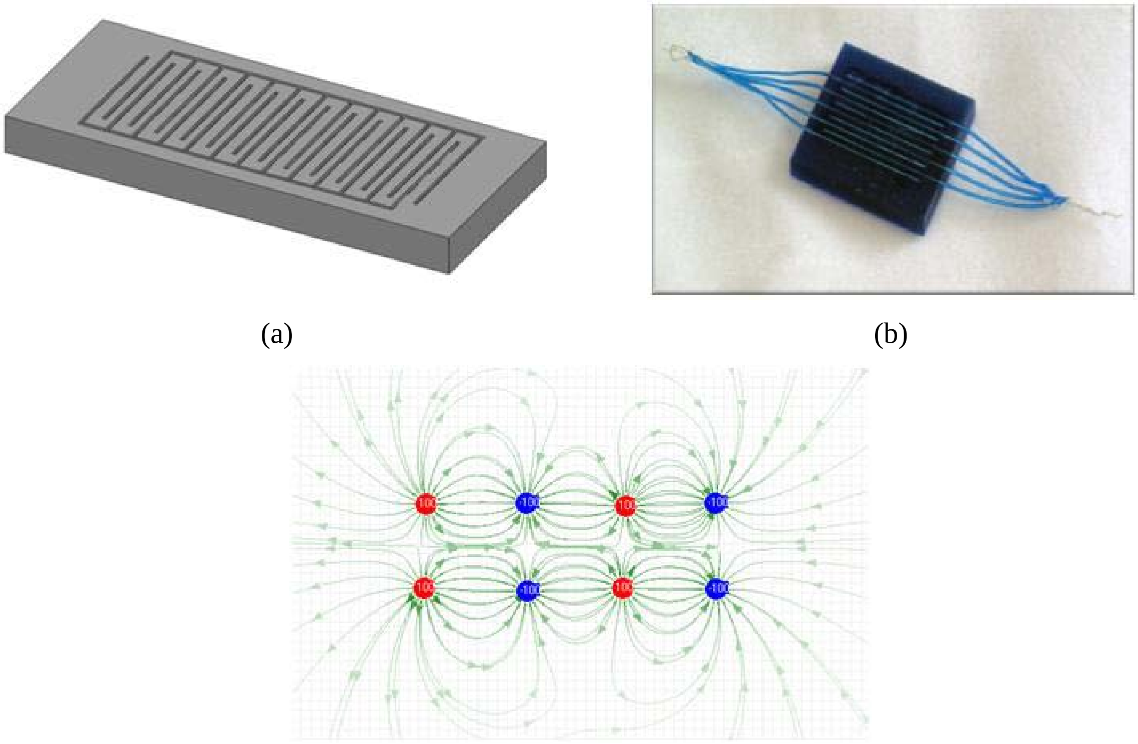
Many attempts have been made, both theoretically and experimentally, to explore the feasibility of controlling short fiber orientation within a matrix. For instance, researchers have long observed flow-induced short fiber alignment when polymer composites are injection molded [5]. Although fiber orientation can somewhat be controlled through gate location, achieving complete control of fiber alignment throughout a part is not generally feasible.

More recently, Calvert et al. capitalized on the phenomenon of fiber alignment during extrusion to create a useful variation of the fused deposition modeling process known as Extrusion Freeform Fabrication (EFF) [6,7]. The process extrudes a liquid polymer and fiber mixture through a syringe attached to an X-Y-Z stage. The syringe tip can be rastered back and forth during extrusion to print one layer of material on top of the next. Given appropriate processing conditions, the resulting 3-dimensional part will have fibers aligned according to the chosen toolpath. The authors demonstrated composite parts reinforced with short carbon fiber at average lengths of 0.38 to 0.085 mm. The study reported a high degree of orientation of the fibers parallel to the direction of writing motion. Although the process is quite promising, the authors sometimes observed a lack of fusion between layers [8]. It should also be pointed out that printing a part by raster extruding one layer of material on top of the next is considerably slower than conventional mold filling processes.

Another approach is to align fibers after the PMC has entered the mold but before solidification. One example of this technique uses shaped magnetic fields along with a magnetic reinforcing phase in the PMC. A numerical model of magnetic fiber manipulation has been developed and experimentally verified [9,10]. This approach is suitable for certain classes of materials in which it is acceptable to coat fibers with a ferromagnetic material.

Kim et al. [11,12] developed the field-aided microtailoring (FAiMTa) process which uses principles of dielectrophoresis to locally control the position and orientation of micro/nano scale inclusions in a PMC. With FAiMTa, a network of interdigitated electrodes is connected to a low frequency alternating current power supply. The resulting non-uniform electric field exerts a field induced torque on fibers in a dielectric fluid (e.g. epoxy) such that the fibers tend to rotate and align with the field lines. Experiments were conducted to prove the concept using different types of inclusions including graphite particles, ceramic particles and glass fibers. The glass fiber/epoxy composite tensile specimens were fabricated and classified by the fiber orientation: aligned randomly, in parallel, and perpendicular to the stress axis. As expected, the modulus and strength of the longitudinally oriented specimens were dramatically higher than those of the perpendicularly and randomly oriented ones. The modulus of elasticity of the longitudinally oriented specimens is 69.6% higher than the random-structured specimens and 115.9% higher than the transverse-structured ones.

The FAiMTa process opens up the possibility of mass production of high performance composites through integration of electrode networks just beneath a mold surface. **Figure 5(a)** shows a CAD model of an interdigitated electrode network. A manually constructed implementation of this design is shown in **Figure 5(b)**. The interdigitated electrode network can be constructed such that the alignment of fibers is optimized with respect to mechanical loading on the part. **Figure 5(c)** shows electric field lines resulting from a given electrode arrangement. Since the preferential alignment of the reinforcement phase varies from one part to another, the electrode configurations that produce field lines aligned with the preferred fiber orientation must also be customized for each part.



(a) (b)

(c)

Figure 5 Integrated electrode network (a) on a CAD model, (b) on a manually constructed mold, and (c) the resulting electric field

The feasibility of freeform fiber alignment via rapid tooling with embedded conformal electrode networks has been demonstrated elsewhere [13]. Micrographs and preliminary mechanical tests were performed to verify the fiber alignment. However, only simple part geometries were studied, and the interdigitated electrodes were manually designed to produce electric field lines along the preferred direction of fiber alignment. The design and fabrication of mold halves can be very challenging when it involves complex geometries and/or curved surfaces.

Prior to fabrication of the molds, the part must be computationally analyzed based on its geometry and functional requirements. Specifically, the material strength and mechanical requirements must be taken into account when indentifying the spatial concentrations and orientations of reinforcing fibers. Finite element analysis (FEA) can be used to determine regions with critically high stress/strain concentrations. Regions in which the stress exceeds the material's yield strength are candidates for reinforcement via preferential fiber alignment.

In this study, geometric algorithms were developed to automate mold and interdigitated electrode design based on results of the FEA study. Following application of these design algorithms, techniques such as direct-write printing can be used to print the conformal electrode networks in conjunction with rapid tooling (RT) techniques. The completed molds are then ready for component fabrication via the FAiMTa technique.

The computational design algorithms presented here consist of (i) an analysis of stress/strain concentrations based on part geometries and load conditions, (ii) a determination of the best possible fiber orientations for critical geometric locations, and (iii) the design of the interdigitated electrode networks. The algorithms were implemented using Visual Basic for Applications (VBA) with in the Solidworks® CAD system.

2.2. Design of Interdigitated Electrode Networks

2.2.1. Procedure Overview

The first step in the design process is to identify critically loaded regions in a component. This is traditionally done via FEA. The rotational force acting upon any given fiber during the alignment process is proportional to the magnitude of the electric field produced by the electrodes. The field intensity rapidly decreases as the distance from the electrode increases [13]. For that reason, the procedure employed here focuses on the outer few millimeters of part thickness. Thus, shell elements are used in FEA static studies to calculate strains based on the intrinsic material properties, the model geometry, and the service environment. The calculated magnitude of normal and shear strains at the center of each FEA element is recorded. The objective of the study is to strengthen regions with critically high strain concentrations, thus a strain threshold is set by the designer. Only those regions whose strains exceed the user defined threshold value actually require optimization of fiber orientation. The overall stress/strain analysis sequence is shown in **Figure 6**.

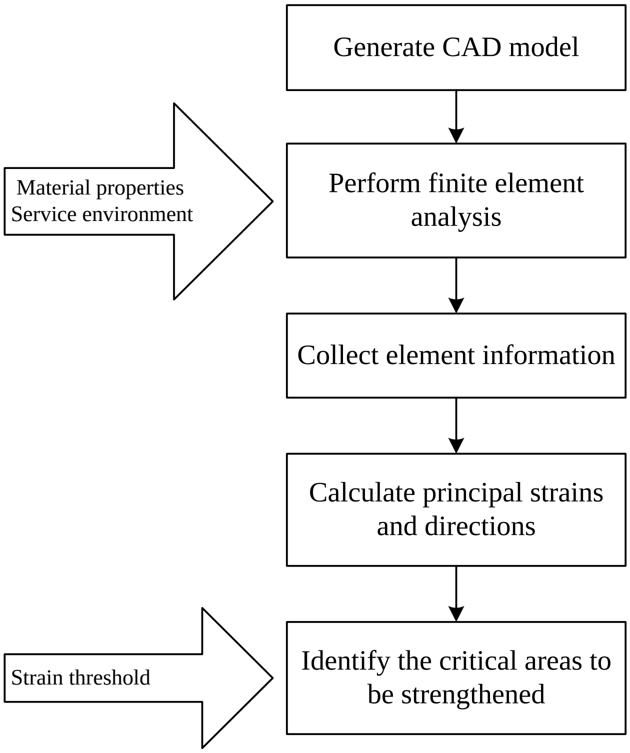


Figure 6 Steps for strain and critical areas analysis

FEA elements whose strains exceed the user defined threshold will be strengthened by aligning fibers with the preferred orientation as determined by the direction of the strains. Since a component can have multiple regions requiring reinforcement, it is necessary to generate an electrode network for each separate region. This is done by checking each critical element to determine whether it has neighboring elements that also require reinforcement. If two critical elements share a node, then it follows that they are neighboring elements. The algorithm uses this fact to identify all elements belonging to a single reinforcement region. It then marches to the next region and identifies all critical elements belonging to that region. The process is repeated until all critical elements are assigned to a specific region requiring reinforcement. Each of these regions requires its own network of interdigitated electrode fingers that are used to align fibers within that region.

As the main line of the electrode must be generated prior to branching into each region’s electrode fingers, the location of the main electrode trunk line on the outer edge of the region must be determined. Then, electrode fingers are branched off of the main line in a direction normal to the strain of the particular element, starting from the elements on the main lines and continuing to the adjacent elements. The algorithm continues the branching method until every element is covered and one half of the electrode network is completed. A duplicate of the electrode fingers is then created to obtain the opposing half of the network. Throughout the design steps, the electrode locations are precisely calculated based on the element locations and the strain directions. The sequence of steps needed to generate the interdigitated electrode network is shown in **Figure 7**, and specific details on how each step is implemented are provided in the next section.

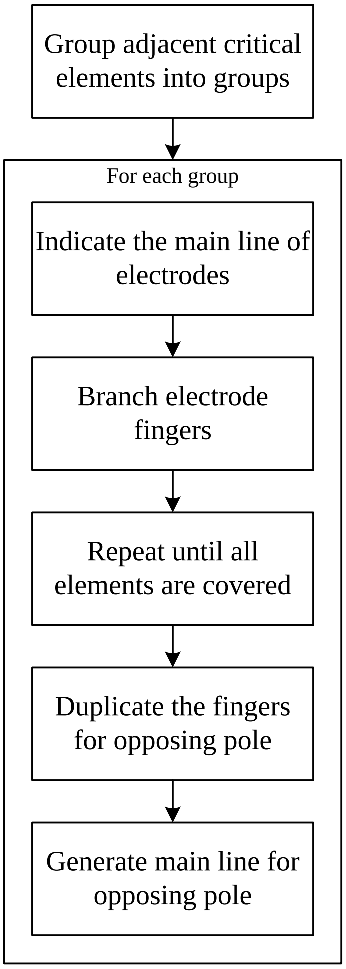


Figure 7 Steps for electrode network design

2.2.2. FEA Study

The first step in the FEA is to define part geometry, material properties (e.g. modulus of elasticity and Poisson’s ratio), and the loads and boundary conditions acting upon the part. **Figure 8(a)** illustrates an example of the FEA modeling in which the pressure is applied on the circumference of the hole on the right and the part is fixed by the hole on the left. Shell meshing is then performed. For this application, the element size is specifically chosen to be equal to the nominal distance between adjacent electrode fingers. In practice, this distance is dictated by the resolution of the fabrication method used to print the electrodes within the mold. By specifying the element size equal to the electrode finger spacing, the FEA elements themselves can be directly used in the algorithm to design the electrode fingers. After the FEA study is conducted, various types of information are available including a graphical strain plot, as shown in **Figure 8 (b)**. Strain data for each mesh element can also be extracted in tabular form.

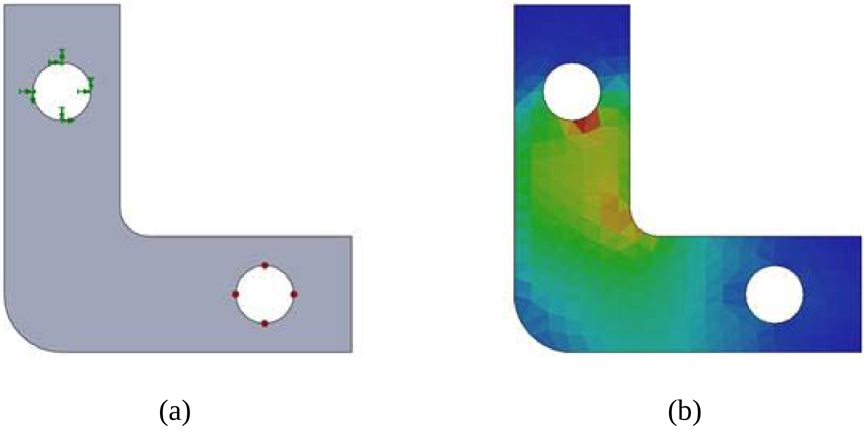

(a) (b)

Figure 8 Model in COSMOSWorks® with (a) the applied load and restraints and (b) the elemental strain plot

The next step in the algorithm is to retrieve element and node coordinates as well as the strain associated with each element. This data is organized into tables for use in subsequent steps of the algorithm. Next, the strain magnitudes and directions are calculated for each element.

In order to identify regions in which fiber alignment is deemed necessary, the user is asked to specify a critical strain threshold. This value can be set as a percentage of the maximum strain or fixed value of strain. Once this is done, the part geometry is segmented into distinct regions requiring fiber alignment. Each critical region is defined by a group of adjacent elements whose strains exceed the user defined threshold. Elements from shell meshing have a 2-dimensional triangular shape with straight or curved edges depending on the model geometries. Two triangular elements are characterized as neighbors if and only if they share at least one common vertex node. Pseudocode for the algorithm is as follows:

Critical Region Procedure

```
Set CriticalRegion to be 0
For each critical element
    If the element is not assigned to any critical region then
        Increase CriticalRegion by 1 to start assigning next region
        Assign the element to CriticalRegion
        Run FindNeighbor Function to find if the element has any adjacent critical element
    Else
        Retrieve CriticalRegion of this element
        Run FindNeighbor Function
    End if
Next critical element
```

FindNeighbor Function

```
For each associated node of the element
    For each associated element of the node
        If the element is a critical element and is not assigned to any CriticalRegion then
            Assign the element to the CriticalRegion
        End if
    Next associated element
Next associated node
```

2.2.3. Electrode Design

Within each region of strain concentration, an interdigitated electrode network must be created. As seen in **Figure 5(a)**, each electrode network requires two main lines which are attached to opposing terminals on the high voltage power supply. Once these main lines are generated, the network of branching interdigitated electrode fingers must be designed that produce electric field lines corresponding to the desired fiber orientations. The process is illustrated with an example in **Figure 9**. The white triangles represent mesh elements that are subject to critically high strain in the vertical direction, whereas the shaded triangles represent elements whose strain is below the critical value. The first step is to identify the location of the first main line on the outer edge of the region. Then, electrode fingers are branched off of the main line in a direction normal to the strain direction of the particular element.

To make a list of the critical elements on the outer edges, the edges of the FEA shell can be selected by the user as the entities under consideration so that the list of corresponding nodes, called edge nodes, is generated. For each critical element, the algorithms browse the three associated nodes and count the number of nodes that are listed on the edges. Subsequently, the elements with two or more edge nodes are recognized as the critical elements on the outer edge and will be used to construct the main line of the electrode.

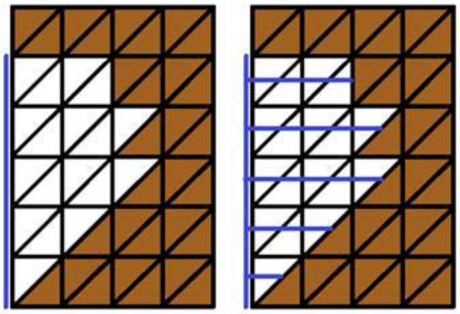


Figure 9 The simplified interdigitated electrode (a) main line and (b) branched trunks

Generating the First Electrode Network Half

Following identification of the edge element for each critical region, the algorithm then seeks to create a series of electrode trunks that are oriented normal to the direction of strain. The basic procedure is as follows:

Start Electrode Procedure

```
For each critical region
    For each critical element on the outer edge
        Retrieve the element's three nodes
        Assign the ones on the outer edge as A and B, and the last one as C
        Retrieve coordinates of the element’s three nodes (A, B, and C) where $\overrightarrow{AB}$ lies on the primary trunk line.
        Create a vector ($s$) which is the projection of the element’s strain components onto the plane defined by element nodes A, B, and C.
        Calculate the coordinates of the midpoint (D) along $\overrightarrow{AB}$.
        Create a vector ($s'$) whose base is at D and whose direction is perpendicular to ($s$).
        Compute the coordinates of point E where vector $s'$ intersects either edge $\overrightarrow{AC}$ or $\overrightarrow{BC}$ in the direction of $s'$.
        Record the point and the edge where the vector $s'$ intersects to be used in the next procedure
    Next element on trunk
Next critical region
```

Figure 10 illustrates the outcomes of application of the algorithm to the element defined by vertices **A**, **B**, and **C**. Vertices **A** and **B** lie on the outer edge of the critical region. The projection of the strain components for this element as determined by the FEA is denoted by $\boldsymbol{s}$, and $\boldsymbol{s'}$ is a vector perpendicular to $\boldsymbol{s}$. The electrode branch emanating from the outer edge starts at the midpoint (**D**) between **A** and **B**, and it extends in a direction parallel to $\boldsymbol{s}$ up to the edge of the FEA element.

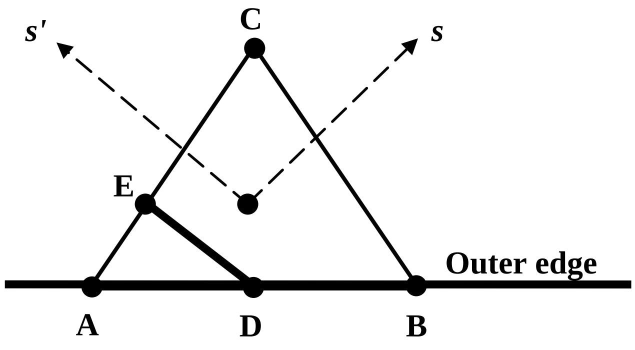


Figure 10 Schematic of electrode branching

Once the branch electrodes have been created for every critical element on the outer edges, the next step is to determine whether the adjacent critical element exists for the electrode to extend. For the example in **Figure 10**, the electrode trunk terminates on the edge $\overrightarrow{AC}$. In consequence, the algorithms search for a critical element that has node **A** and **C** as its vertices. If the particular element exists, then a similar procedure is followed for creating an $\boldsymbol{s'}$ vector that determines the direction the branch should follow in that adjacent element. The only variant for this step is that node **D** is the midpoint of the edge $\overrightarrow{AC}$ of the element where the branch originates. In this manner, each electrode branch is extended until it terminates at a boundary of the critical region. The sequence of branching conditions is illustrated in **Figure 11**. This step is repeated until every critical element has been used to generate an electrode finger.

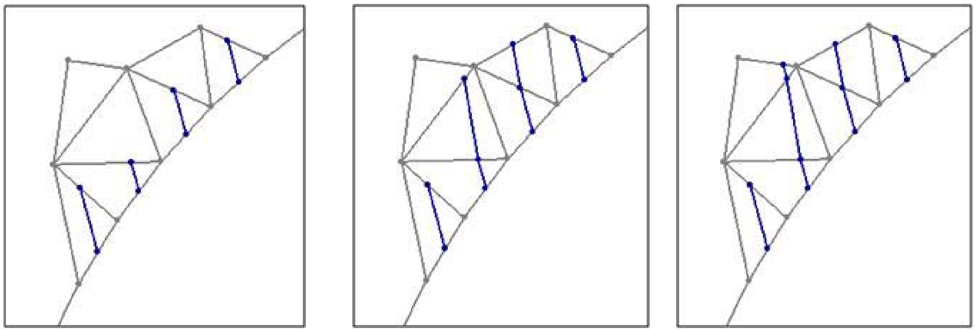


Figure 11 Branching the electrode finger into the neighboring element

Generating the Opposing Electrodes

To produce the desired electric field, the opposing half of the interdigitated electrode network must be generated. The second set of electrodes is simply the duplication of the first one. The duplicated electrode branch is translated so that it is (1) located between a pair of the first set of electrode branches and (2) offset away from the main trunk of the first set. The basic procedure is as follows:

Translate Procedure

```
For each electrode branch
    Retrieve the section of the electrode branch that emanates from the outer edge ($\overrightarrow{A_1A_2}$)
    If the next branch in the same critical region ($\overrightarrow{B_1B_2}$) exists then
        Retrieve the coordinates of the starting and end points of the current and the next branch of the same critical region ($\mathbf{A_1}$, $\mathbf{A_2}$, $\mathbf{B_1}$ and $\mathbf{B_2}$)
        Calculate the length of the electrode
        Create a vector ($\boldsymbol{s}$) whose direction is parallel to the electrode
        Calculate the coordinates of the midpoint ($\mathbf{C_1}$)between the starting points
        Translate $\mathbf{C_1}$ in the direction of $\boldsymbol{s}$, up to the distance of half of the distance between $\mathbf{A_1}$ and $\mathbf{B_1}$
        Calculate the coordinates of a point ($\mathbf{C_2}$)so that the new electrode has the same length and direction
    End if
Next electrode branch
```

Figure 12 illustrates the outcomes of application of the algorithm. The section $\overrightarrow{A_1A_2}$ of the branch is immediately emanated from the outer edge of the region. The vector representing the direction of $\overrightarrow{A_1A_2}$ is denoted by $\boldsymbol{s}$. The electrode for the opposing pole, $\overrightarrow{C_1C_2}$, is a reproduction of $\overrightarrow{A_1A_2}$ which is repositioned by the described procedure.

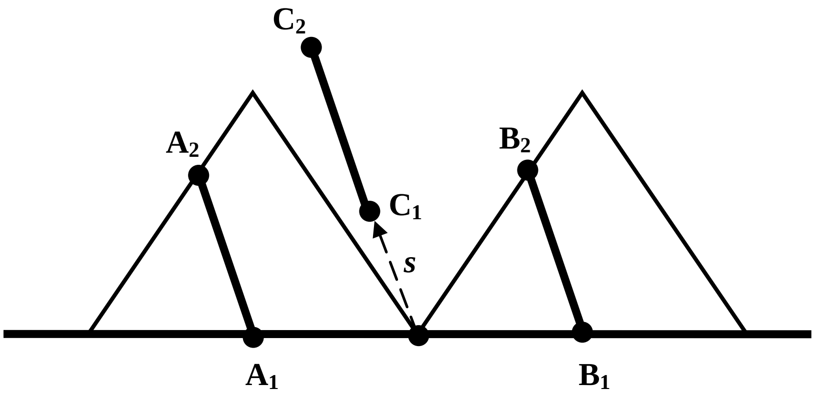


Figure 12 Schematic of electrode duplicating and translating

Subsequently, the main lines for the electrode networks are generated. As previous algorithms record every coordinate of every section of the electrode branches, the main lines are precisely created to connect the electrode branches of the same pole. The main line of the first set of electrodes, shown as blue lines in **Figure 13**, is constructed by chaining together adjacent starting points. On the contrary, the other main line is generated by connecting the ending points of the opposing electrodes, shown as red lines in **Figure 13**.

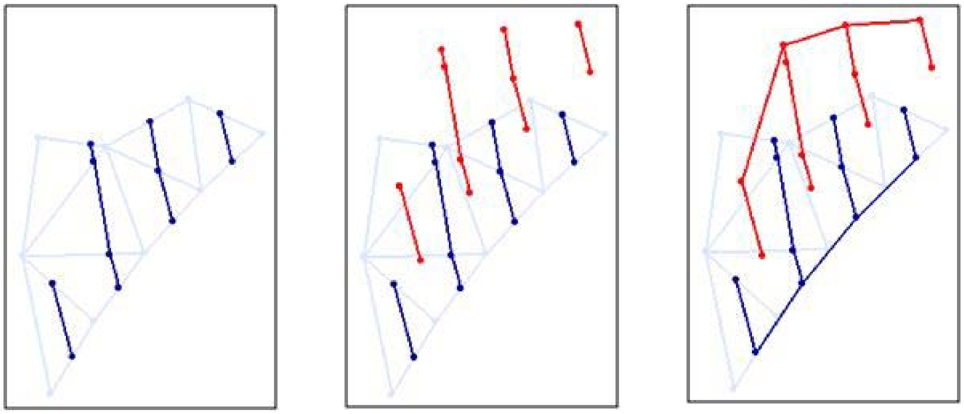
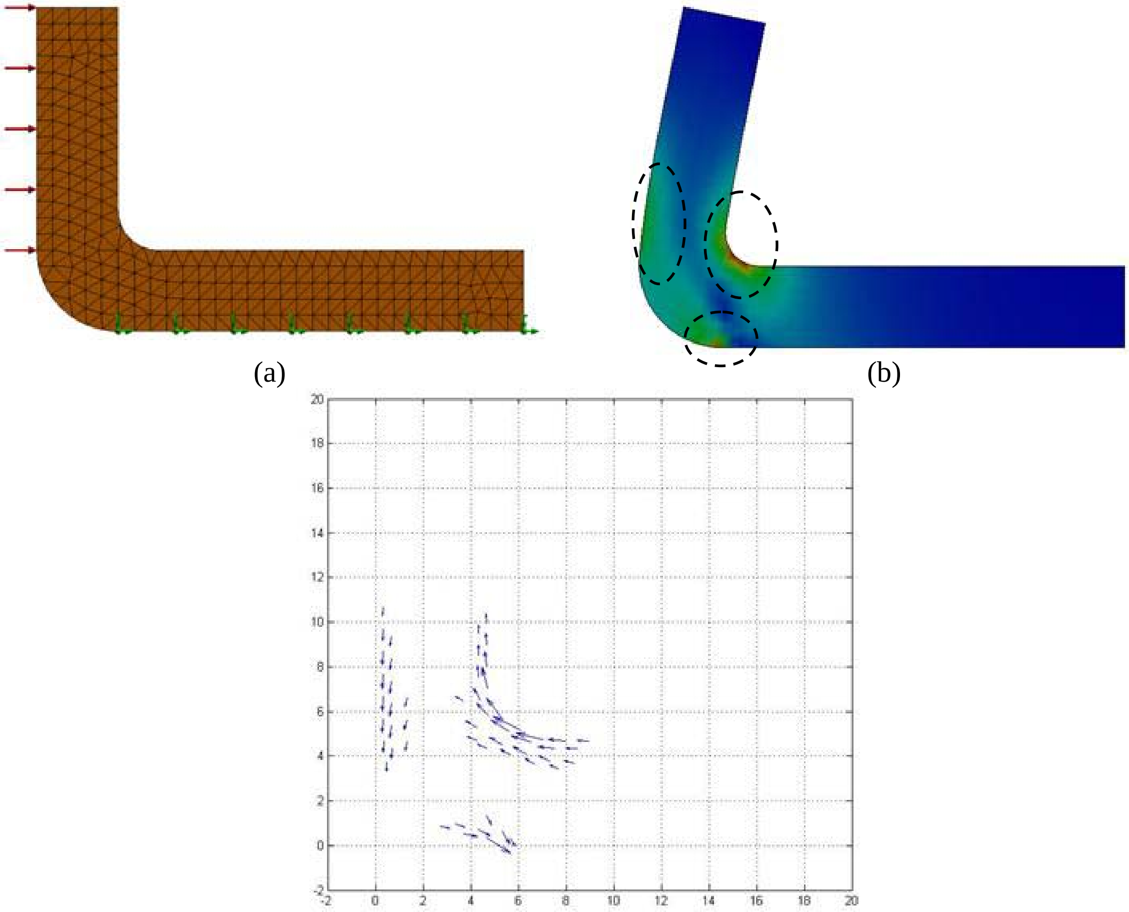


Figure 13 Generating the opposing electrode networks and main trunks

2.3. Outcomes and Discussion

The preceding algorithms were implemented in the Solidworks® CAD package while using COSMOSWorks® for the FEA and MATLAB for various plotting routines and vector operations. The routine is illustrated here via analysis of an L-shaped part. The material properties used in the analysis are similar to those of epoxy resin (i.e. modulus of elasticity = 6.0 GPa and Poisson's ratio = 0.4). The results were evaluated by comparing the electrode networks with the graphical results from COSMOSWorks® as well as strain direction plots generated using MATLAB. The specimen was subjected to the load and boundary conditions as shown in **Figure 14(a)** where the part was fixed at the bottom surface and a pressure of 1 psi was applied along the upright edge. The part was meshed at the size of 1 mm such that the spacing between the electrode fingers would be 0.5 mm. This is a reasonable value based on direct-write processes used to print electrodes. A strain plot from the FEA, as shown in **Figure 14 (b),** was generated illustrating three regions of high strain concentration as circled. The processed data was imported into MATLAB to create a plot of the projection of the strain components in the critical regions as shown in **Figure 14 (c)**.



(a) (b)

(c)

Figure 14 L-shaped specimen with (a) the load and boundary conditions, (b) the equivelent strain plot, and (c) the projection of the strain components

The series of algorithms implemented in Solidworks® using the API were executed to design the interdigitated electrode based on the processed information. An electrode network was generated as a series of sketches in SolidWorks®, illustrated in **Figure 15**. When the electrode network sketches in **Figure 15** are compared with the strain plot illustrated in **Figure 15 (b)**, the algorithms successfully located the areas with high strain concentrations where the fibers should be aligned along the strain lines. Considering the direction of strains in **Figure 15 (c)**, the process effectively located the electrode network in the manner that the electric field lines up with the direction of strains.

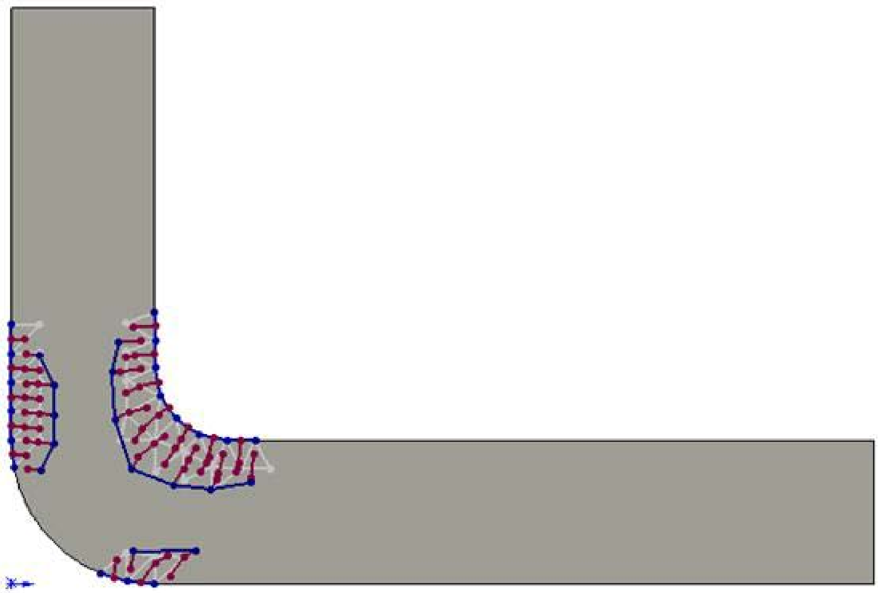


Figure 15 Interdigitated electrode network for the l-shaped specimen

Once a collection of sketches was made to represent the preferred design of the interdigitated electrode network, the preferred fiber arrangement can be achieved through the fabrication of the mold halves with integrated electrode networks. One possibility is to use any AM processes to fabricate the mold halves as a part of RT processes, and the electrode network can be printed on the back of the mold halves using a direct-write process such as an nScrypt micro-dispensing system or an Optomec Aerosol Jet. These machines use direct-print dispensing techniques in which electrically conducting inks can be conformably deposited on various substrates. The dispensing nozzles are controlled by a computer-aided-design (CAD) program that offers the accuracy and repeatability of the XYZ positioning with a resolution within a few microns [14]. It is particularly important to note that these processes allow one to print electrode networks onto non-planar surfaces, such as mold surfaces.

2.4. Summary and Conclusions

Reinforcing a component with discontinuous carbon or glass fibers is common across a broad range of applications. However, it is not generally possible to precisely control the alignment of fibers as desired over the entire part surface when using molding techniques. Ideally, one would like to see fibers aligned in order to impart additional strength and stiffness. One promising method of doing this via the influence of electric field (FAiMTa) has been proposed elsewhere. However, the manual analysis and design of complex interdigitated electrode networks can be very difficult. In this paper, a complete set of geometric algorithms for the computational design of mold halves used to fabricate aligned discontinuous fiber composite components via FAiMTa has been presented. The study focuses on the challenges associated with computational design of load-bearing components with oriented short fibers as well as the CAD implementation.

Starting from gathering the necessary strain information from FEA, the ideal fiber arrangement and series of sketches representing the interdigitated electrode networks are generated. The proposed algorithm has been implemented in the Solidworks® CAD system using the Solidworks® API with the VBA programming environment. FEA is conducted in COSMOSWorks® using expected loading conditions. From the results, the Solidworks® implementation is used to sketch the location of embedded electrode networks to create electric field lines parallel to the desired fiber alignment direction.

The results show a promising outcome of the proposed algorithms. The design of the electrode networks agrees with the strain plot regarding the area with high strain concentration. The projection of the strain components plot also confirms the validation of the algorithms to design the interdigitated electrode network for preferred fiber alignment.

References

1. Pipes RB, McCullough RL, Taggart DG. Behavior of discontinuous fiber composites: Fiber orientation. Polymer Composites 1982;3(1):34-39.

2. Chung DDL. Carbon fiber composites. Boston: Butterworth-Heinemann, 1994.

3. De SK, White JR. Short fibre-polymer composites. Cambridge, England: Woodhead Pub., 1996.

4. Matthews FL, Rawlings RD. Composite materials : engineering and science. Boca Raton; Cambridge, England: CRC Press; Woodhead Publishing, 1999.

5. Papathanasiou T. Flow-induced alignment in injection molding of fiber-reinforced polymer composites. Flow-induced alignment in composite materials 1997:112.

6. Calvert P, Crockett R. Chemical solid free-form fabrication: making shapes without molds. Chem.Mater 1997;9(3):650-663.

7. Calvert P, Lin TL, Martin H. Extrusion freeform fabrication of chopped-fibre reinforced composites. High Perform Polymers 1997;9(4):449.

8. Shofner M, Rodriguez-Macias F, Vaidyanathan R, Barrera E. Single wall nanotube and vapor grown carbon fiber reinforced polymers processed by extrusion freeform fabrication. Composites Part A: Applied Science and Manufacturing 2003;34(12):1207-1217.

9. Hatta H, Yamashita S. Fiber orientation control by means of magnetic moment. J Composite Mater 1988;22(5):484.

10. Yamashita S, Hatta H, Sugano T, Murayama K. Fiber orientation control of short fiber composites: experiment. J Composite Mater 1989;23(1):32.

11. Kim G, Moeller D, Shkel Y. Orthotropic Polymeric Composites with Microstructure Tailored by Electric Field. J Composite Mater 2004;38(21):1895.

12. Kim GH, Shkel YM. Polymeric composites tailored by electric field. J Mater Res 2004;19(4):1164-1174.

13. Modak S. Freeform fiber alignment in a polymer matrix using conformal electrode field networks. 2008:71.

14. Li B, Clark P, Church K. Robust Direct-Write Dispensing Tool and Solutions for Micro/Meso-Scale Manufacturing and Packaging. Proceedings of the ASME International Manufacturing Science and Engineering Conference.

Chapter 3 Finite Element Analysis of Polymer Matrix Composites with Preferentially Aligned Short Fiber

Abstract

It is well understood that mechanical properties in discontinuously reinforced polymer matrix composite materials are strongly correlated with the degree of fiber alignment in the matrix. To make the greatest use of the reinforcing phase's strength, the fibers must be aligned with the stress or strain axis. This can be done in a variety of ways such as the field-aided micro tailoring (FAiMTa) approach which uses the effect of non-uniform electric field produced by interdigitated electrodes to align fibers. Geometric algorithms that automate the process of electrode design have been developed to preferentially align the fibers according to the loading conditions of the component being produced. In order to validate the benefit of preferential fiber alignment, finite element analysis (FEA) models have been created in which the geometric elements are assigned orthotropic stiffness values representative of what would be expected when reinforcing fibers for that element are preferentially aligned to best resist expected loading patterns. These models are compared with the initial models in which the fibers throughout the component are randomly aligned. In this study, micromechanics models were used to predict the modulus of elasticity and Poisson's ratios. The Halpin-Tsai model was used to predict the stiffness of unidirectional short fiber reinforced composites. The prediction was then used to define the material properties of the models specifically in the regions that are subjected to fiber alignment. A mixed model was used to predict the stiffness of randomly aligned composite used in the initial models and the regions of the final models that do not require fiber alignment. The procedures have been implemented on three models and are demonstrated in this paper. Strain magnitude histograms are used to evaluate the effectiveness of preferential fiber alignment.

3.1. Introduction

Composites serve the needs of high-performance products requiring high strength-to-weight ratios. Composites reinforced with short carbon fibers are very common across a broad range of industries, such as aerospace, automotive, and sporting goods. For additional strength and stiffness, fibers can be preferentially aligned. Carbon fiber is one of the most attractive materials due to its excellent mechanical properties. However, some properties are strongly orientation dependent, including strength and modulus. **Table 2** shows the mechanical properties of unidirectional epoxy reinforced with continuous carbon fibers [1]. The test orientation corresponds to the angle between the carbon fiber orientation and the stress axis. It explicitly indicates a significant decrease in modulus and strength when the fibers are perpendicular to the test axis.

Table 2 Mechanical properties of unidirectional carbon fiber/epoxy composite [1]

Test Orientation	**Tensile**		**Compressive**	
	Modulus (GPa)	**Strength (MPa)**	**Modulus (GPa)**	**Strength (MPa)**
0 degree	310	999	229	421
90 degrees	7.6	21.3	7.6	129

Analysis of short fiber reinforced composites (SFRP's) is more complicated due to the random alignment of fibers. The fiber orientation has a strong influence over the properties of discontinuous fiber-reinforced composites. The decrease in modulus as the degree of orientation decreases reflects the reduction in reinforcing efficiency [2-4]. Experimental data also confirms that the strongest glass fiber/epoxy composite can be obtained by aligning the short fibers along the stress axis [5]. A mathematical model predicting the elastic moduli of short fiber reinforced thermoplastics has been proposed [6-8]. The modulus of elasticity of short carbon fiber-reinforced polymers based on the distribution of fiber length and orientation in three dimensions has also been studied [9]. The study illustrated how both fiber volume fraction and the mean orientation angle have a significant influence on the elastic modulus of SFRPs. In addition, the effect of mean fiber orientation angle becomes more critical when the composite is loaded with a higher fiber volume fraction.

In 2004, Kim and Shkel developed a process that is capable of aligning short fibers through application of a high potential electric field to the mixture of uncured epoxy resin and reinforcement phase. This methodology, called field-aided micro-tailoring (FAiMTa) [10], has shown promise as a means of manipulating the inclusions in liquid epoxy. Previous research has demonstrated the feasibility of manipulating both spherical and fibrous particles suspended in a liquid polymer matrix prior to curing. Moreover, FAiMTa opens up the possibility of mass production of high performance composites by integrating electrode networks into mold halves.

Although the rough design of interdigitated electrode configuration may be accurate and straightforward for simple geometries, the process is very difficult for complex geometries. Prior to mold fabrication, the molded part geometry must be computationally analyzed based on the part’s functional requirements. Specifically, the expected loading conditions and material properties must be taken into account in order to obtain the solution method indentifying the preferred locations and directions of reinforcing fibers. Geometric algorithms, which semi-automate the processes of electrode design, have been developed as described in Chapter 2. Subsequently, the composite parts may be fabricated under the influence of non-uniform electric field created by the electrode networks. Short fiber inclusions are manipulated by dielectrophoresis such that the component has tailored mechanical properties while maintaining the continuity of the bulk material.

In this chapter, a validation of the effect of preferentially-aligned fibers is performed using FEA. The result from geometric design of the electrode networks is used to identify the critical regions where preferential fiber alignment is needed. The CAD model of the component is sectioned so that each critical region is assigned orthotropic material properties according to the direction of the aligned fibers. Non-critical regions of the component that are not covered by the interdigitated electrode are assigned isotropic material properties consistent with what would be found in randomly-aligned fiber composite materials. Histograms of magnitude of strains are shown to evaluate the effect of fiber alignment in the strain distribution.

3.2. Stress/strain Concentrations in FEA

Recently, several methods have been developed to optimize the performance of fiber reinforced composites [11-14]. There are several numerical methods available to improve the composites’ load bearing capacity and to reduce stress/strain concentrations by optimizing the fiber orientation. In this study, the optimization of fiber orientation was determined using FEA which considers the structure as an assembly of small finite-sized elements, as shown in

Figure 16. The elements are called finite elements, and the points connecting the elements are called nodes.

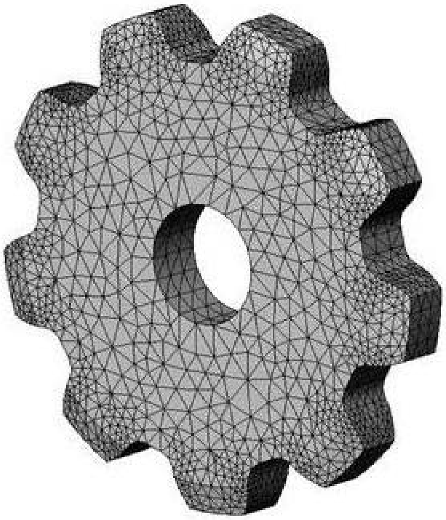


Figure 16 Finite element analysis meshing

The FEA process begins with a model of the geometry which is discritized into finite elements. The element stiffness matrices are assembled under the overall equilibrium condition. Users have the ability to define externally applied loads and boundary conditions prior to the application of a system of simultaneous algebraic equations [15]. Typically, FEA offers useful information such as stresses, reactions, and displacements. As an example, **Figure 17** shows a graphical strain plot derived from the FEA software package COSMOSWorks®. Geometric features, such as fillets, shoulders, grooves, holes, keyways, threads, and so on, result in modification of the simple stress/strain distributions and can cause high localized stresses/strains. **Figure 17** illustrates strain concentrations introduced by a cross-sectional change on a cylindrical tensile specimen under an axial load.

Figure 17 Equivalent strain plot of a cylindrical tensile specimen under an axial load

The material begins to fail if the stress in any given region exceeds the material's yield strength. In regions where this is the case, it is preferred to strengthen the material locally to increase the maximum load capacity. Although this can be done by adding more material or by selecting a different material, it can also be done through proper fiber alignment in short fiber reinforced composites. The anisotropic behavior of aligned-fiber composites is the key to improving the mechanical performance of the composite systems.

3.3. Local Fiber Alignment and Customized Molds

Composite systems with aligned fibers possess anisotropic mechanical properties where the strength and stiffness in the direction parallel to the fiber axis is significantly higher. Thus, the regions with higher strain should locally be strengthened by arranging the reinforcing fibers such that the fiber axes are aligned with the strain directions. The FAiMTa process offers the possibility of locally manipulating the orientation of the reinforcing phase. The electrode design procedures are semi-automated by geometric algorithms that integrate the required part and tooling design steps. Those procedures include (1) the finite element analysis (FEA) of the part according to the external load applied, (2) the determination of the preferred fiber direction, and (3) the interdigitated electrode network generation in the CAD model. Subsequent to use of these algorithms, mold halves with interdigitated electrodes are fabricated via rapid tooling (RT). Specifically, the molds may be produced by additive manufacturing (AM) processes, and the electrode networks may be printed on the back of the molds using a direct-write process such as nScrypt's micro-dispensing technique, Optomec's Aerosol Jet process, and nano silver ink dispensing platform.

The cast composite component fabricated from the customized mold halves possesses heterogeneous material properties. In other words, mechanical properties, such as modulus of elasticity and Poisson's ratio, vary through the part due to locally-aligned fibers. The sections of the component that experience the influence of the electric field will have orthotropic material properties with the highest strength and stiffness corresponding to the preferred fiber alignment. The performance of the tailored-microstructure composite components is expected to surpass that associated with components having randomly aligned fibers. Specifically, the magnitude of strain should decrease. To validate this assumption, FEA was performed for the locally-aligned fiber reinforced components.

3.4. Stiffness of Short-fiber Composites

The elastic behavior of short-fiber composite materials is strongly influenced by the fiber aspect ratio, the fiber volume fraction, and the fiber orientation within the matrix. To determine the stiffness of the composites, two fiber orientation conditions are considered in this study. The first involves a composite with unidirectionally aligned fibers, and the second involves composites with randomly aligned fibers.

3.4.1. Unidirectionally Aligned Short Fiber Composites

Unidirectional fiber composites, as shown in **Figure 18**, are described as orthotropic materials with two axes of symmetry. When fibers are aligned with the x-axis, the strength and stiffness are considerably greater in the x-axis than in the transverse directions, i.e. y- and z-axes. To calculate the modulus of elasticity and Poisson's ratio in this case, the directional stiffness prediction is performed using micromechanics models.

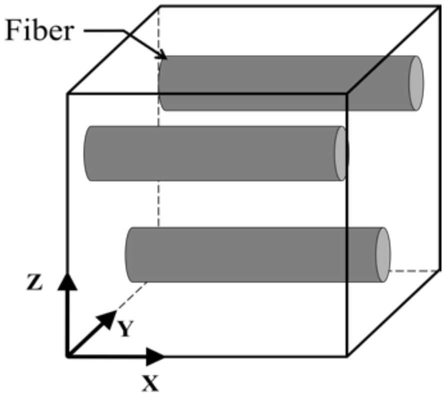


Figure 18 A Representative body of a perfectly-aligned fiber composites

Several micromechanics models for the stiffness of aligned composites have been proposed such as Halpin-Tsai, Nielsen, Mori-Tanaka, Lielens, self-consistent, and shear lag [16]. These models have been evaluated, specifically for short-fiber composites, by comparison to finite element calculation [8]. For longitudinal modulus E_{11} and transverse modulus E_{22}, the Halpin-Tsai model predicts the most accurate result for fibers having a relatively small aspect ratio. The predictions for the major Poisson's ratio, ν_{12}, and minor

Poisson's ratio, ν_{23} are acceptable at the lower aspect ratio as well. The Halpin-Tsai equations are as follows [7]:

$$\frac{P}{P_m} = \frac{1+\xi\eta\upsilon_f}{1-\eta\upsilon_f} \tag{1}$$

where
$$\eta = \frac{(P_f/P_m)-1}{(P_f/P_m)+1} \tag{2}$$

The ξ parameter, which is used to describe the influence of fiber geometry, for each property is given in **Table 3**. The *P* parameter can be expressed in terms corresponding to the material property of interest, as listed in **Table 3**. A subscript *f* indicates a quantity associated with the fibers and *m* denotes a matrix quantity. Hence, the fibers have modulus of elasticity E_f and Poisson's ratio ν_f, while the counterparts of the matrix are E_m and ν_m. υ_f is fiber volume fraction.

Table 3 Halpin-Tsai parameters for short-fiber composites [8]

Property	***P***	$\boldsymbol{P_f}$	$\boldsymbol{P_m}$	$\boldsymbol{\xi}$
Longitudinal modulus	E_{11}	E_f	E_m	2(*l*/*d*)
Transverse modulus	E_{22}	E_f	E_m	2

The in-plane Poisson's ratios are predicted by the rule of mixtures as follows:

$$\nu_{12} = \nu_f\upsilon_f + \nu_m\upsilon_m \tag{3}$$

$$\nu_{23} = \nu_f\upsilon_f + \nu_m\upsilon_m\,[1+\nu_m - \nu_{12}(E_m/E_{11})]/[1-\nu_m^2 + \nu_m\nu_{12}(E_m/E_{11})] \tag{4}$$

However, it is valuable to note that a perfectly-aligned fiber structure is normally difficult to fabricate. Typically, the fiber orientation distribution functions are generated analytically to describe the characteristics of composite systems with partially aligned short fibers [16]. However, the distribution functions of the composite systems that are fabricated under the influence of dielectrophoresis have not been thoroughly studied.

3.4.2. Randomly Aligned Short Fiber Composites

Despite the anisotropic properties of the fibers, composite systems with randomly aligned fibers have the macroscopic behavior of an isotropic material. Manera proposed a model to predict the effective elastic properties of randomly oriented short fiber composites and observed good correlation between experimental values and theoretical values [17]. The model used the classical laminate analogy in conjunction with invariants defined by Tsai and Pagano [18]. Puck's micromechanics model was then used to predict the modulus of elasticity and Poisson's ratios [17]. The study derived the approximate equations as follows:

$$\bar{E} = v_f \left(\frac{16}{45} E_f + 2E_m \right) + \frac{8}{9} E_m \tag{5}$$

$$\bar{\nu} = \frac{1}{3} \tag{6}$$

$\bar{E}$ and $\bar{\nu}$ represent the modulus of elasticity and Poisson's ratio of randomly aligned short fiber composites respectively. These equations are applicable only for short fiber composites with a volume fraction in the range of 0.1 to 0.4 and for matrix materials with a modulus of elasticity in the range of 2.0 to 4.0 GPa.

3.4.3. Stiffness Prediction

To simulate the performance of the composite components, glass fibers are used as the reinforcing inclusion. The FAiMTa technique has successfully been used to manipulate glass fibers of cylindrical shape having aspect ratios from 10 to 50 and average diameters 25 µm with 0.1 volume fraction in epoxy resin [10]. This study uses the stiffness of typical glass fibers with diameter of 10 µm. The matrix material for this model is epoxy which is the most widely used as the matrix material. The required material properties are listed in **Table 4**.

Table 4 Material properties used in stiffness predictions [19]

Property		
Modulus of elasticity (GPa)	f	7 2.4
	m	2 .75
Poisson's ratio	ν_f	0 .2
	ν_m	0 .33
Volume fraction	v_f	0 .1

Substituting the parameter values from Table 3 into Equations (1)-(6), the stiffness of the composite systems is predicted as shown in **Table 5**. Subscripts l and t stand for longitudinal and transverse properties respectively. As expected, the unidirectionally aligned fiber composite has substantially greater modulus of elasticity in the longitudinal direction compared to the transverse direction strength. The modulus of the randomly-aligned fiber composite falls in the middle, also as expected.

Table 5 Stiffness prediction

Fiber alignment	Property	
Unidirectional	E_l (GPa)	28.618
	E_t (GPa)	5.356
	ν_l	0.317
	ν_t	0.441
Random	$\bar{E}$ (GPa)	7.038
	$\bar{\nu}$	0.333

3.5. FEA and the Effect of Fiber Alignment

For comparison with analytical models, FEA was performed for both the randomly aligned fiber composite condition and the preferentially aligned fiber composite conditions. The steps followed are described as follows:

i. Perform FEA using isotropic material properties for the randomly-aligned fiber composite. Compute and record the magnitude of strain for each FEA element.

ii. Implement the geometric algorithms described in Chapter 2 to identify the critical FEA elements and their directions of preferred fiber alignment.

iii. Divide the FEA shell mesh into sections that represent critical FEA elements as shown in **Figure 19**. Assign orthotropic properties with the longitudinal axis defined by the preferred fiber direction for each element.

iv. Assign isotropic properties of randomly-aligned fiber composite for other parts of the component that are not covered by the electrode networks.

v. Perform FEA a second time using the new locally tailored material properties. Compute and store the magnitude of strain for each FEA element.

vi. Generate histograms of strains to compare the results between two models

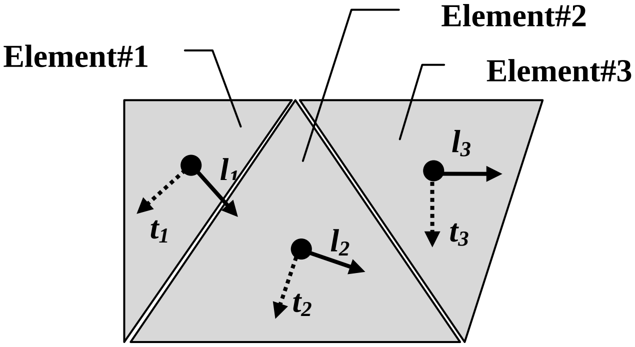
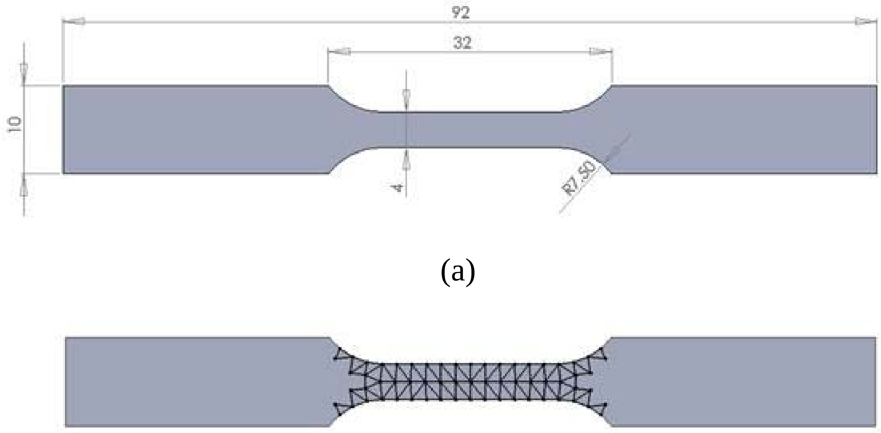


Figure 19 Sections of surface that represent separated FEA elements. Solid arrows ***l*** represent the preferred fiber longitudinal directions and dotted arrows ***t*** represent the transverse directions.

The steps described above were carried out for three different geometries. First, a tensile specimen was analyzed under an axial tension to demonstrate a simple case. In the second case, a corner bracket was modeled as an example of a structural component that could be fabricated as a composite. Finally, a customized bone plate was used to demonstrate the capability of the procedure on curved surfaces and holes.

3.5.1. Case I: Tensile Specimen

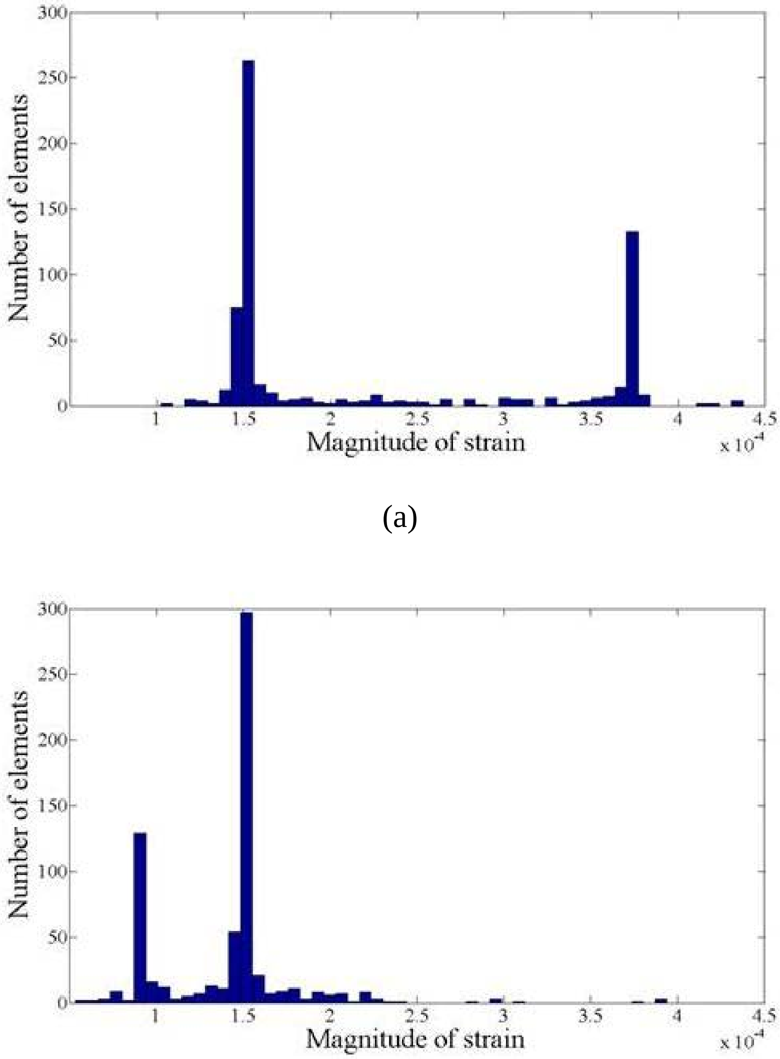
A 2 mm-thick tensile specimen was modeled in SolidWorks® as shown in **Figure 20(a)**. One end of the specimen was fixed and the other was placed under a uniform axial tension of 1.0 MPa. The FEA was performed for shell meshing with the stiffness of randomly-aligned fiber composite shown in **Table 5**. After the implementation of the design algorithms, the critical FEA elements requiring higher stiffness through fiber alignment are identified, as shown in **Figure 20(b)**. The histograms are shown in **Figure 21**.



(a)

(b)

Figure 20 CAD model of a tensile specimen with (a) dimensions in mm and (b) sketches of the critical elements



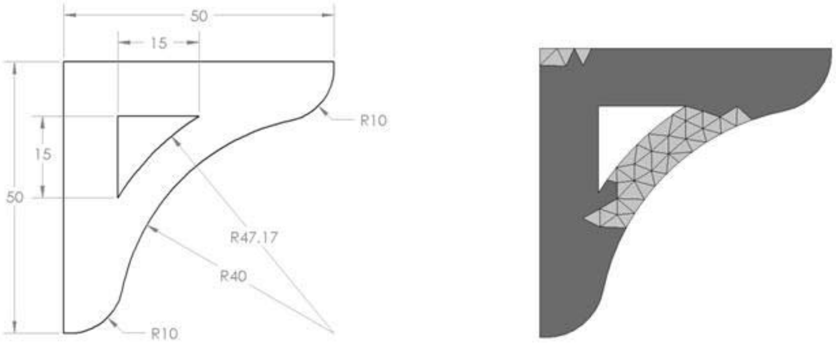
(a)

(b)

Figure 21 Histograms of strains in the specimen with (a) randomly-aligned fibers and (b) preferentially-aligned fiber

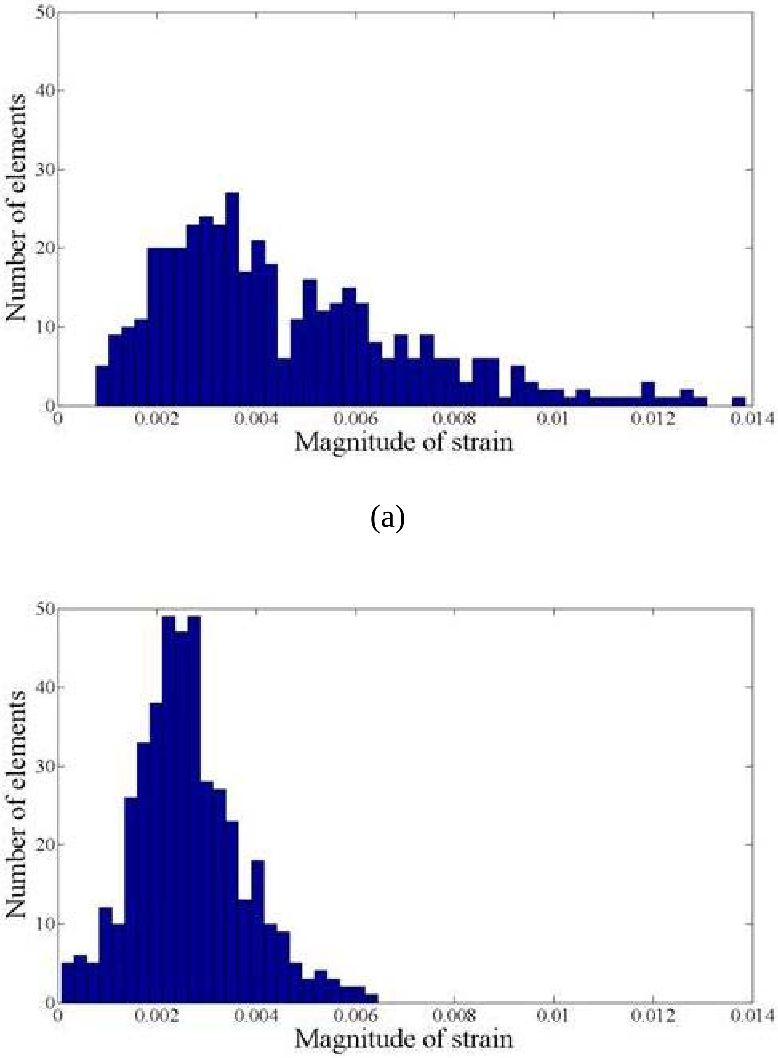
3.5.2. Case II: Corner Bracket

Stiffness improvement of structural components is critical. A 5 mm-thick corner bracket was modeled to represent a small-scale example of a load bearing structure. **Figure 22(a)** illustrates the dimensions of the bracket which were modeled in FEA to be fixed on the upright edge. A uniform pressure of 5.0 MPa was applied on the top surface. Based on the geometric algorithms, the critical elements are determined as shown in the lightly shaded triangles in **Figure 22(b)**. The histograms are shown in **Figure 23**.



(a) (b)

Figure 22 CAD model of a bracket with (a) dimensions in mm and (b) sketches of the critical elements



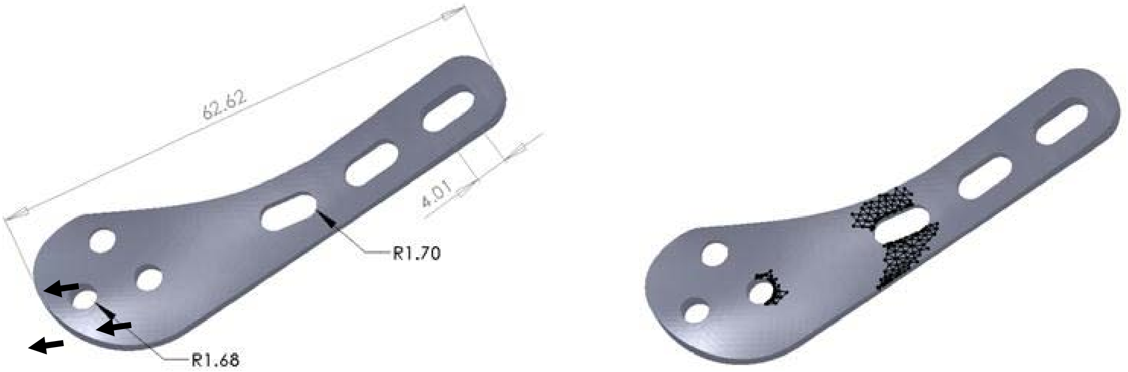
(a)

(b)

Figure 23 Histograms of strains in the bracket with (a) randomly-aligned fibers and (b) preferentially-aligned fiber

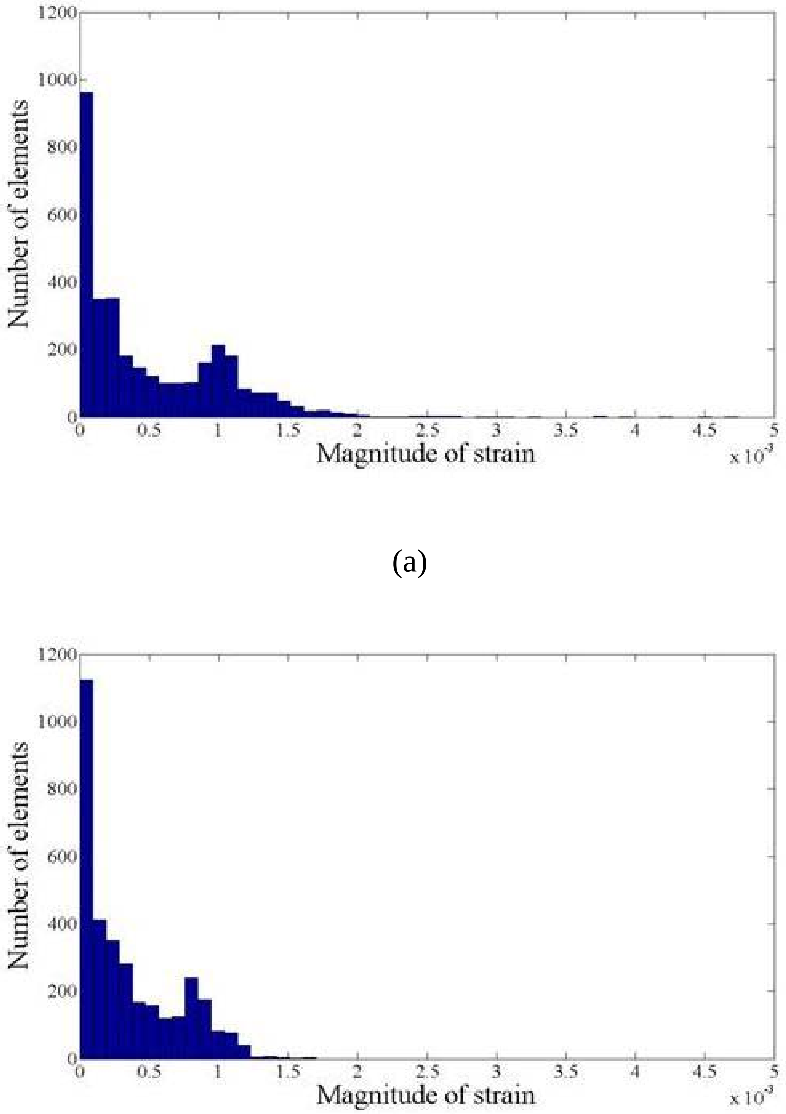
3.5.3. Case III: Bone Plate

A custom designed bone plate for a distal femur [20] was modeled in SolidWorks$^{®}$ as shown in **Figure 24(a)**. The bone plate has an overall length of 62.76mm and a uniform thickness of 4 mm. The component was modeled under fixed restraints on three of the oblong holes. A pressure of 1.0 MPa was applied on three of the round holes along the direction of the arrows. The sketches of triangles in **Figure 24(b)** are the FEA elements that are subjected to high strains. The histograms are shown in **Figure 25**.



(a) (b)

Figure 24 CAD model of a bone plate with (a) dimensions in mm and (b) sketches of the critical elements



(a)

(b)

Figure 25 Histograms of strains in the bone plate with (a) randomly-aligned fibers and (b) preferentially-aligned fiber

3.5.4. Discussion

The results of each case study show significant strain redistributions as the aligned fiber regions were subject to much lower strains. Peak strain concentrations were also considerably reduced. Histograms quantitatively represent the effect of fiber orientation on strain redistribution. With random fibers, the tensile specimen has a great number of FEA elements subject to high strains, as shown in **Figure 21(a)**. Specifically, more than 20% of the elements exhibit strain values in the range of $3.74\text{-}3.81\times10^{-4}$. However, those high strain values are not present in **Figure 21(b)** where the fibers are locally aligned in the model. The implementation on a corner bracket shows the effect of fiber alignment as well. **Figure 23(b)** also illustrates that none of the elements has a strain magnitude higher than 6.5×10^{-3} while approximately 20% of the elements in **Figure 23(a)** are subjected to strain higher than 6.5×10^{-3}. **Figure 25(b)** also shows that fewer elements are subjected to high strain compared to **Figure 25(a)**. The highest strain magnitude in the strengthened bone plate is 0.0017 while the randomly-aligned model has a peak strain of 0.0047.

3.6. Summary and Conclusions

Through FEA, models were analyzed under working conditions to identify critical areas subject to high strain concentrations. These areas were modeled to be reinforced by fiber alignment thus locally enhancing stiffness of the material. The orthotropic stiffness was predicted using micromechanics models, and the modulus of elasticity and Poisson's ratios were calculated. The FEA results showed an advantageous effect of fiber alignment after the critical areas were modeled with orthotropic stiffness according to the preferential fiber alignment. The effect of local fiber alignment was studied using histograms which showed that the components with aligned fibers are better able to resist the loading conditions.

The results of this study support the hypothesis that the design of composites with locally fiber alignment can influence the strain distribution pattern. However, the FEA models used in this study implied several assumptions regarding the simulated microstructures. The fibers in the model were assumed to be perfectly aligned within the areas that are influenced by the electrode fields. Future research and experiments can be conducted to experimentally investigate the actual fiber alignment distribution. This more accurate distribution and resulting material properties, in turn, can be used in the micromechanics models as part of a structural design optimization routine to improve specific strength and/or stiffness values in weight critical applications.

References

1. Chung DDL. Carbon fiber composites: Butterworth-Heinemann, 1994.

2. Pipes RB, McCullough RL, Taggart DG. Behavior of discontinuous fiber composites: Fiber orientation. Polymer Composites 1982;3(1):34-39.

3. De SK, White JR. Short fibre-polymer composites. Cambridge, England: Woodhead Pub., 1996.

4. Matthews FL, Rawlings RD. Composite materials: engineering and science: Woodhead Publishing, 1999.

5. Kacir L, Ishai O, Narkis M. Oriented short glass-fiber composites. IV. Dependence of mechanical properties on the distribution of fiber orientations. Polymer Engineering & Science 1978;18(1):45-52.

6. Chin WK, Liu HT, Lee YD. Effects of fiber length and orientation distribution on the elastic modulus of short fiber reinforced thermoplastics. Polymer Composites 1988;9(1):27-35.

7. Halpin JC, Finlayson KM, Ashton JE. Primer on composite materials analysis. Boca Raton, Fla.: CRC Press, Taylor & Francis Group, 1992.

8. Tucker III CL, Liang E. Stiffness predictions for unidirectional short-fiber composites: Review and evaluation. Urbana 1998;51:61801.

9. Fu SY, Lauke B. The elastic modulus of misaligned short-fiber-reinforced polymers. Composites Sci Technol 1998;58(3-4):389-400.

10. Kim GH, Shkel YM. Polymeric composites tailored by electric field. J Mater Res 2004;19(4):1164-1174.

11. Wetherhold RC, Seelman S, Wang J. The use of functionally graded materials to eliminate or control thermal deformation. Composites Sci Technol 1996;56(9):1099-1104.

12. Crothers P, Drechsler K, Feltin D, Herszberg I, Kruckenberg T. Tailored fibre placement to minimise stress concentrations. Composites Part A: Applied Science and Manufacturing 1997;28(7):619-625.

13. Suresh S. Graded materials for resistance to contact deformation and damage. Science 2001;292(5526):2447.

14. Pelletier JL, Vel SS. Multi-objective optimization of fiber reinforced composite laminates for strength, stiffness and minimal mass. Comput Struct 2006;84(29-30):2065-2080.

15. Spyrakos C. Finite element modeling in engineering practice. 1995.

16. Chou TW. Microstructural design of fiber composites: Cambridge Univ Pr, 1992.

17. Manera M. Elastic properties of randomly oriented short fiber-glass composites. J Composite Mater 1977;11(2):235.

18. Tsai SW, Pagano NJ. Invariant properties of composite materials: Defense Technical Information Center, 1968.

19. Mallick P. Fiber-reinforced composites: materials, manufacturing, and design: CRC Press, 1993.

20. Lee ES. Design and in vitro evaluation of five epiphyseal plate fracture stabilization methods in canines. 2008:121.

Chapter 4 Rapid Tooling for Preferentially Aligned Short Fiber Composite Components Fabricated Via Dielectrophoresis

Abstract

Rapid tooling (RT) is an alternative to conventional tool making. RT can significantly reduce lead times, and can also be used to produce parts with complex geometries via assembled tooling. This paper explores the use of RT to produce short fiber reinforced composite components with preferentially aligned fiber orientation. The field-aided microtailoring (FAiMTa) process has been shown to be capable of manipulating inclusions in uncured epoxy under a non-uniform electric field. Customized electrode networks are designed by algorithms developed to locally align fibers parallel to the direction of strain under expected loading conditions. In this study, the objective is to employ three direct-write techniques in order to print conformal electrode-integrated networks onto RT molds. Those techniques include nScrypt micro dispensing process, Optomec Aerosol Jet process, and nano silver ink dispensing platform.

4.1. Introduction

The unidirectional mechanical properties of continuous carbon fiber/epoxy composites, including tensile strength and stiffness, greatly exceed those of 7075-T6 aluminum and Ti-6Al-4V [1]. However, the fabrication processes for such materials are complex and challenging for mass production due to the high labor content and long curing cycles. Having short or discontinuous fibers rather than continuous ones can be very beneficial in terms of production cost and time when components must be mass produced. To improve upon the relatively low strength of polymer matrix composites with randomly oriented short carbon fibers, it is preferred to arrange the reinforcing fibers parallel to the direction of stress in a component for optimal performance. One technique for doing this is through the field-aided microtailoring (FAiMTa) process which uses principles of dielectrophoresis (DEP) to locally control the position and orientation of micro/nano scale inclusions in a PMC [2,3]. Studies have been conducted demonstrating the feasibility of using dielectrophoresis to manipulate different types of inclusions including multi-wall carbon nanotubes (MWNT) [4,5] and single-wall carbon nanotubes (SWNT) [6].

It is generally preferred to arrange reinforcing fibers to reduce local strain concentrations and to enhance directional strength without increasing the fiber volume fraction. As a result, the preferential alignment of the reinforcing phase varies with part geometry. The DEP electrode configurations must therefore be customized in response to the applied loads. When a voltage differential is applied between a pair of interdigitated electrodes, as illustrated in **Figure 26**, an electric field is generated between each pair of fingers. If the electrode network is embedded just beneath the surface of a mold used in the casting of fiber reinforced resin, then they can be used to preferentially orient fibers in the resin prior to curing. This is done by alternating the voltage polarity at low frequency such that the direction of the electric field lines between opposing electrode fingers also alternates. Fibers in the liquid matrix will rotate such that they become aligned with the electric field lines. Through careful design of the electrode geometry, it is therefore possible to generate customized orthotropic material properties stemming from preferential fiber alignment throughout the part.

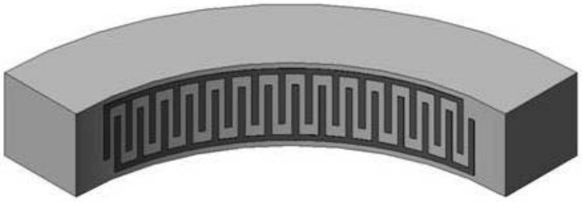


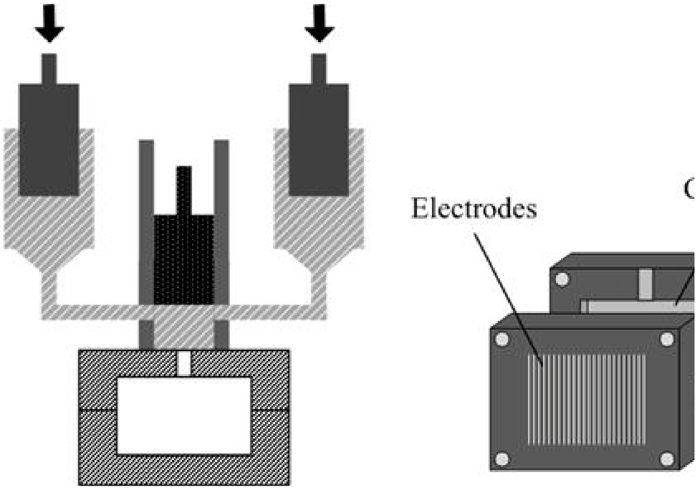
Figure 26 Planar interdigitated electrode network

Currently, there is no commercially available manufacturing method suitable for mass producing composite components with freeform preferential alignment of the short fiber reinforcing phase. The FAiMTa technique opens up the possibility of mass production of high performance composites through integration of electrode networks with the mold halves. However, fabrication of the tool and electrode networks is exceedingly complex. In this study, a methodology is proposed and implemented to manufacture the tools via a combination of direct-write electrode printing and rapid tooling (RT). The mold halves are first fabricated via a suitable additive manufacturing process. Then, toolpaths to direct-write print the conducting electrode geometries are generated. To print the integrated electrode networks onto the mold surfaces, a suitable direct-write process such as Optomec's Aerosol Jet or nScrypt's microdispensing technique is used with conducting ink.

4.2. Rapid Tooling for FAiMTa

4.2.1. Reinforced Reaction Injection Molding

Fiber-reinforced epoxy parts can be fabricated by the reinforced reaction injection molding (RRIM) process, as shown in **Figure 27(a)**. This technique utilizes fibers blended with two-component resins that crosslink upon mixing [7]. It is advantageous that the process can be adapted to preferentially align the reinforcing fibers using a mold with a series of interdigitated electrodes arranged based on the predefined fiber direction as shown in **Figure 27 (b)**. Specifically, the FAiMTa process opens up the possibility of automated production of high performance composites through integration of electrode networks just beneath the mold surface. However, the fabrication of mold halves can be challenging when it involves complex geometries and/or curved surfaces. The molds not only have to be as complex as the part geometries, but a method of arranging the electrodes must be devised. Nevertheless, additive manufacturing processes simplify this step with the ability to produce virtually any geometric shape including the conformal interdigitated electrode-integrated molds.



(a) (b)

Figure 27 (a) Schematic of RRIM, (b) Mold halves with integrated electrodes

4.2.2. Rapid Tooling for Fiber/epoxy Composite

In order to produce tooling for the RRIM process, a number of rapid tooling (RT) techniques reported in the literature are suitable. For instance, injection molding RT can be made of plastics, metals, hard alloys, or other special materials [8]. Some of the RT techniques are based on the stereolithography (SLA) process. SLA parts are built using a liquid UV-curable photopolymer which is solidified as a laser scans over the surface of the liquid resin. Parts made by SLA feature high precision and outstanding surface properties which are important characteristics in mold making [9]. Even though the toughness and wear resistance of the mold material are not sufficient for high volume production, epoxy molds suitable for production of hundreds of parts have been built via the SLA process [10]. These molds underwent the injection pressure and temperature for more than 500 injection cycles without tool failure. Moreover, several studies have been conducted to improve SLA mold performance involving thermal post-cure [11], cooling channel geometries [12], and the reduction in applied pressure [13].

4.2.3. Printing of Conformal Interdigitated Electrodes

The combination of RT processes with direct-write material printing processes is proposed as the means by which conformal interdigitated electrodes are produced in the molds. The feasibility of freeform fiber alignment via RT with manually embedded conformal electrode networks has been demonstrated [14]. The study involved fabrication of 1.2-2.5 wt% carbon fiber/epoxy composite samples using SLA molds with integrated interdigitated electrodes. Micrographs and preliminary mechanical tests were performed to verify the fiber alignment. However, only simple part geometries were studied, and the mold was manually designed and constructed to produce curved cylindrical parts whose fibers would be aligned with the curvature of the cylinder’s axis. The interdigitated electrodes were also manually constructed either via magnet wire hand laid into grooves in the mold surface or via manually printed silver ink traces. The electrodes were designed to produce electric field lines along the preferred direction of fiber alignment. Although the rough design of interdigitated electrode configuration may be accurate and straightforward in simple parts, the process can be very difficult for complex geometries. Prior to the RT fabrication of the molds, the part must be computationally analyzed based on the geometry and functional requirements. Specifically, the material strength and mechanical requirements must be taken into account in order to obtain the solution method identifying the preferred locations and directions of reinforcing fibers.

4.2.4. Local Fiber Alignment

Recently, several methods have been developed to optimize the performance of fiber reinforced composites [15-18]. The majority of the studies conducted are targeted towards laminated composites fabricated via a continuous fiber layup process. These methods involve the use of fiber tow (or prepreg) and offer the capability of spatially varying the fiber orientation within a layer. The tow-placement can be automated using a Computer Numerical Control (CNC) multi-axis machine. However, the process is limited by the minimum radius of curvature of the fiber path. This depends largely on the tow width and manufacturing machine used [19]. The deposition path must be adjusted to accommodate process limitations, hence it is not always possible to align fibers in the preferred fiber direction.

The FAiMTa process offers somewhat unique possibilities for locally manipulating the reinforcing phase. Moeller et al. [20] developed an algorithm to determine the orientation of the particulates in a geometrically favorable manner to reduce a stress concentration. Finite element analysis (FEA) was used in the iterations to define the optimal fiber direction aiming to reduce the stress/strain concentration around the geometric discontinuities.

However, the study only focused on using the algorithm to determine the optimal fiber directions. The authors did not describe methods to transfer the preferred fiber pattern into the fabrication of physical mold halves. Previous work in this dissertation successfully developed geometric algorithms that semi-automate the process of electrode design for preferential fiber alignment in composite components. Subsequently, RT techniques can be used to produce molds with interdigitated electrodes just beneath the mold surface as shown in **Figure 28**.

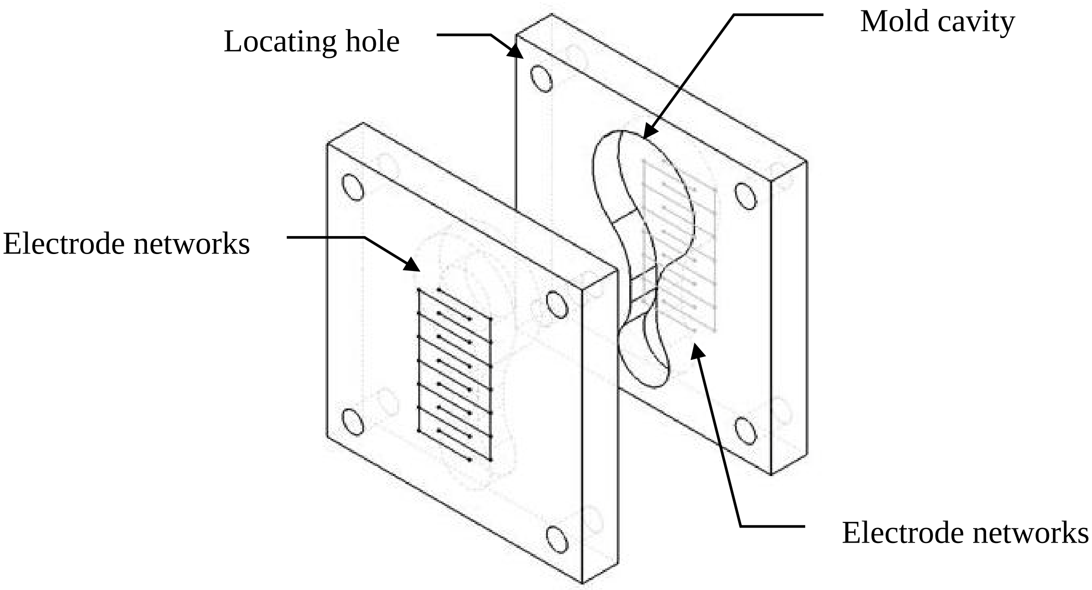


Figure 28 Electrode network-integrated mold halves

4.2.5. Electrode Printing

Given the locations of the electrodes designed by the algorithms, a suitable direct-write material printing process can be used to print the interdigitated electrode networks onto the mold. According to nScrypt, the microdispensing process is capable of printing inks with up to ±12µm accuracy at the maximum travel speed of 300 mm/s. In the process of electrode design described in the previous chapter, the XYZ coordinates of vertices that define each electrode are recorded in a spreadsheet. The direct-write system prints a trace of conducting ink from one vertex to the next in order to produce the electrode networks. For each area of strain concentration, there is a pair of electrodes as illustrated in **Figure 29**. The opposing electrodes are wired to the two opposing poles of a high voltage power supply while the fiber reinforced composite is cast. During the printing process, the main lines of the networks are fabricated first, and then the electrode fingers are branched out from the main line.

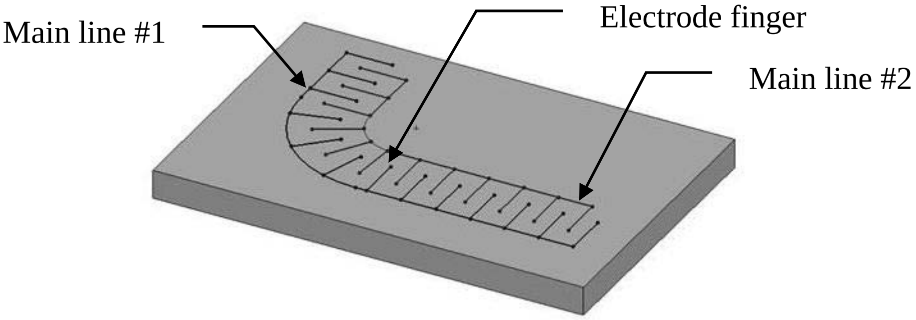


Figure 29 Two sets of sketches representing an electrode network

In the case of the nScrypt process, a text file must be created to control the path of the dispensing head to print the electrodes. Each motion command includes the incremental distance that the print head must move in the X, Y, and Z axes respectively. The flow of ink in the nScrypt process is controlled via what is effectively a miniature needle valve. Hence the machine can also be programmed to advance or retract the needle in order to start or stop printing. In this research, Visual Basic for Applications (VBA) was used to generate commands for the nScrypt motion controller based on the electrode vertex information stored in Microsoft Excel. At the beginning of the printing process, the dispensing tip is located at the origin of the XY plane. The dispensing tip is then moved to the starting point of a main electrode line and moved down in the Z axis direction so that it is approximately 200 µm above the substrate. The needle valve is then opened such that conducting ink begins to flow. As the conducting ink flows, the nozzle is moved along the desired path to produce the main electrode lines. At the end of each main line, the needle valve is closed to stop the flow of ink and the nozzle is moved up to 2 mm above the substrate. It is then moved back to the origin. This printing sequence is programmed to continue until every main electrode line is completed. To automate the interaction between Microsoft Excel and Microsoft Word, VBA retrieves the data from the spreadsheets and generates a text file that controls the steps described. The basic procedure is as follows:

nScrypt Tool Path Generation for Main Lines

```
For each critical region
    For each main line
        Write coordinates of the starting point and insert end-of-line (EOL)
        Write “0 0 -2” and insert EOL
        Write “PRINT” and insert EOL
        For each point on the main line
            Retrieve coordinates of the previous point
            Retrieve coordinates of the current point
            Calculate the incremental travel distance for each axis between the two points
            Write the travel distance in each axis and insert EOL
        Next point on the main line
        Write “STOP PRINT” and insert EOL
        Write “0 0 2” and insert EOL
        Write negative values of coordinates of the last point
    Next main line
Next critical region
```

After the tool paths needed to print the main lines have been generated, the VBA routine continues with a process to create tool paths for electrode finger printing. The process is similar to that used for the main electrode lines except that printing starts from the end of the finger and then extends towards the main line (as opposed to starting from the main line and branching outward). This is done simply to avoid starting a printing operation on top of an existing line. Once each electrode trunk is done, the machine is programmed to stop printing and to move the nozzle up and over to the origin. The basic procedure is as follows:

nScrypt Tool Path Generation for Electrode Trunks

```
For each critical region
    For each set of fingers
        For each trunk
            Write coordinates of the starting point and insert end-of-line (EOL)
            Write “0 0 -2” and insert EOL
            Write “PRINT” and insert EOL
            For each point on the trunk
                Retrieve coordinates of the previous point
                Retrieve coordinates of the current point
                Calculate the incremental travel distance  for each axis between the two points
                Write the travel distance in each axis and insert EOL
            Next point on the trunk
            Write “STOP PRINT” and insert EOL
            Write “0 0 2” and insert EOL
            Write negative values of coordinates of the last point
        Next trunk
    Next set of fingers
Next critical region
```

At the end of these procedures, a text file is generated and transferred to the nScrypt tabletop dispensing machine so that the interdigitated electrode networks can be accurately printed on the back of mold halves. The steps for generating printing paths for the Optomec’s Aerosol Jet and the silver ink dispensing process are very similar, although the process parameters are obviously different. The commands are generated accordingly under the same fundamental algorithms.

4.3. Implementation and Discussion

To check the validity of the process, an illustrative example was created to demonstrate the execution of the steps, including electrode design and mold fabrication. A commercial software package called COSMOSWorks®, a design analysis system fully integrated with SolidWorks®, was used for FEA.

4.3.1. Electrode Design

A 2 mm-thick bracket-shaped sample part with a size of 50 mm by 50 mm, as shown in **Figure 30**, was analyzed. Assuming that the part is fixed along the upright edge which sets the translational and rotational degrees of freedom to zero, it undergoes a uniform pressure of 1 MPa on the top surface. The material properties used in the analysis are similar to those of epoxy resin, i.e. modulus of elasticity was specified as 6.0 GPa and Poisson’s ratio was specified as 0.33. Regarding the capability of nScrypt's machine, a mesh size of 4 mm was used to accommodate electrode finger spacing of 2 mm. An equivalent strain plot from the finite element analysis as shown in **Figure 31** was generated illustrating two regions of high strain concentration as circled.

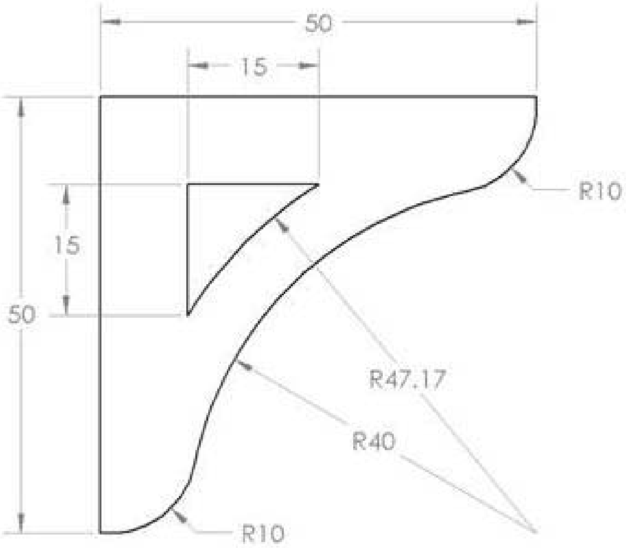


Figure 30 Sample part geometry

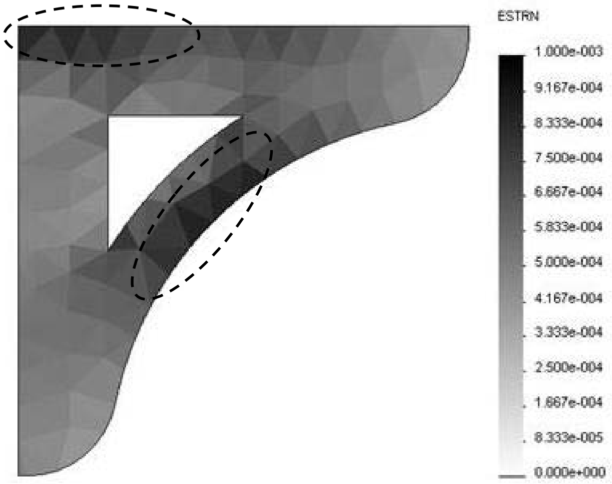


Figure 31 Equivalent strain plot on COSMOSWorks

With the use of Solidworks® and the Solidworks API, the series of algorithms were executed to design the interdigitated electrode based on the processed information. The design of the electrode networks was generated in a list of the XYZ coordinates in spreadsheets as well as a series of sketches in SolidWorks®, illustrated in **Figure 32**.

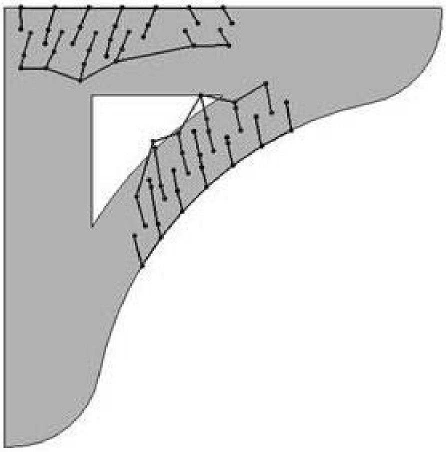


Figure 32 Interdigitated electrode network with 2 mm spacing

Depending on the type of AM process used, the CAD models are prepared differently prior to the fabrication process. Regardless of the manufacturers of the machine, the 3D solid model can be used and processed. In this research, epoxy mold halves were built via the SLA process and were used in the following direct-write electrode printing techniques.

4.3.2. nScrypt Microdispensing System

The procedures for text file generation, as previously described, were applied to this sample problem. Subsequently, the text file was transferred to an nScrypt microdispensing system for electrode network fabrication. An illustration of the electrode networks generated for the test part is shown in **Figure 33**.

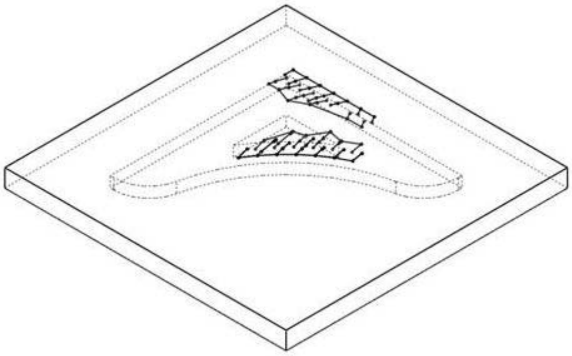


Figure 33 Mold half with a bracket-shaped cavity and printed electrodes

Conducting silver ink from Stan Rubenstein Associates (EP403920HV-50) was loaded into the nScrypt machine shown in **Figure 34**. Many printing trials with the silver ink were conducted in order to obtain process parameters suitable for this material. In the end, the smallest diameter nozzle that could be used with this highly viscous ink was 75 µm in diameter. Many attempts were then made to print the electrode network on a mold backing. The routine described in this chapter to generate the nScrypt printing tool paths worked perfectly, as shown in **Figure 35**.

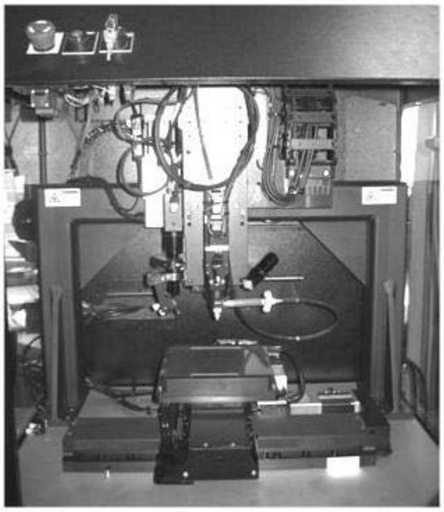


Figure 34 nScrypt Microdispensing machine

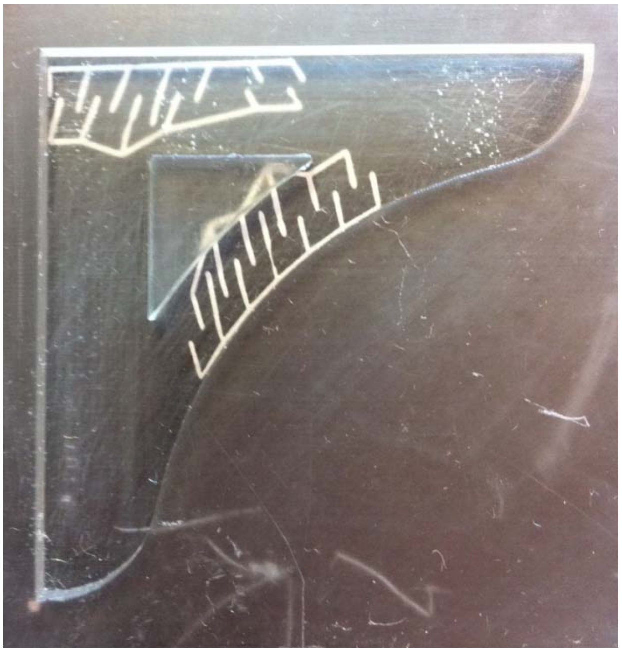


Figure 35 Electrode network printed by nScrypt Microdispensing machine

4.3.3. Nano Silver Ink Dispensing Platform

An alternative technique for direct-write electrode printing is to print a nano silver particle ink through a syringe that is numerically controlled in Z direction. The XY movement of the platform is also controlled by a G-code based program. **Figure 36** illustrates an electrode design on the same part geometry. The strain threshold was set at 60% of the maximum magnitude, thus resulting in a smaller critical region. The FEA mesh size was 2 mm, therefore the electrode trunks have 1 mm spacing. **Figure 37** and **Figure 38** show the electrode network that was fabricated by dispensing the nano silver ink. In order to guarantee a continuity of the conductive nano silver particle, the printing process was done three times over the same path. As a result, ink overflow caused two adjacent electrodes to bridge together as shown in **Figure 38**(b).

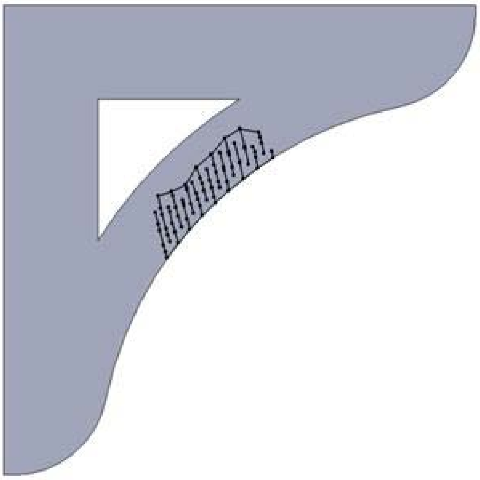


Figure 36 Interdigitated electrode network with 1 mm spacing

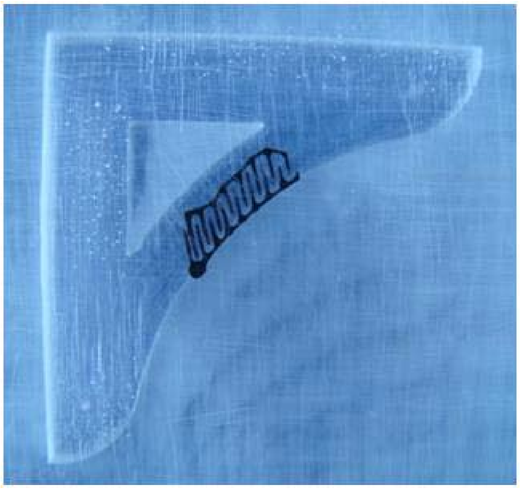
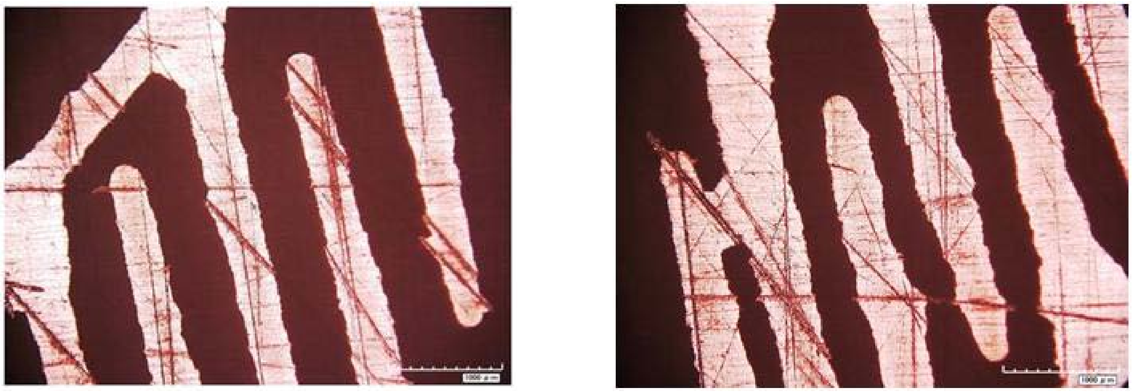


Figure 37 Electrode network printed with nano silver ink



(a) (b)

Figure 38 Micrograph of the electrodes printed with nano silver ink

However, the printing process must be sufficiently precise to ensure that adjacent electrode fingers do not bridge together and electrically arc during fiber alignment. For this reason, the next attempt at printing a functioning mold with interdigitated electrode fingers will be done using the Optomec Aerosol Jet process which is capable of printing conducting ink traces as narrow as 15 microns and with a thickness of 1micron or less. This is therefore an ideal process for printing DEP electrode fingers.

4.3.4. Optomec Aerosol Jet

The Optomec Aerosol Jet machine, utilizing aerodynamic focusing to precisely deliver fluid, generally serves the purpose of low-volume fabrication of electronic circuits. Since the printing process is capable of the smaller feature size, it is advantageous to design an electrode network at smaller spacing. The computational design algorithms was executed on the same part geometry using a mesh size of 400 microns to accommodate an electrode finger spacing of 200 microns as shown in **Figure 39**. A CAD drawing of the designed network was successfully transferred to the Optomec Aerosol Jet. The interdigitated electrodes were fabricated accordingly as shown in **Figure 40**.

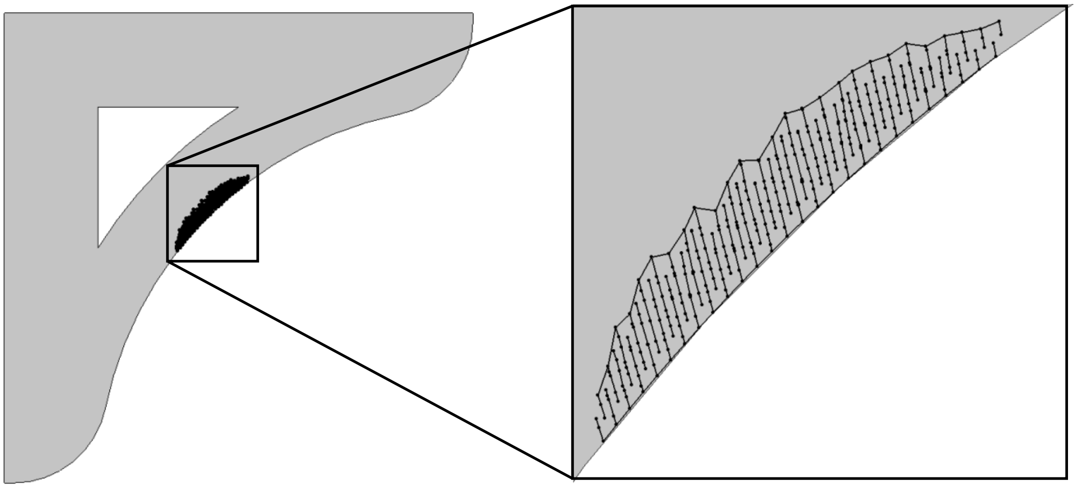


Figure 39 Interdigitated electrode network with 200 micron spacing

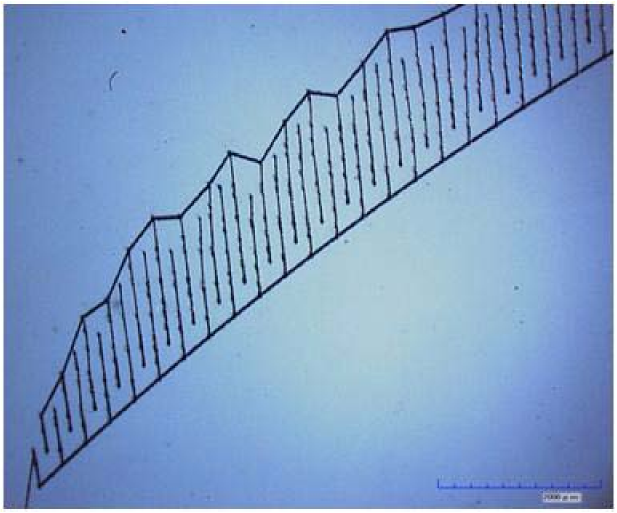


Figure 40 Micrograph of the electrodes printed by Optomec Aerosol Jet

4.4. Summary and Conclusions

Although the nScrypt machine and nano silver ink dispensing process were capable of following the tool paths generated using the algorithms described in this chapter, the high viscosity of the silver ink made it difficult to print electrode traces sufficiently narrow for DEP. It is important to point out that the DEP process is being done at a voltage of several thousand volts. No current flows between the interdigitated electrode fingers due to the fact that they are not connected. Therefore, these direct-write processes are only capable of the electrode network with spacing in the order of millimeters.

The finer interdigitated networks are always desirable for high performance composite parts because a finer FEA meshing provides better local fiber preferred orientation in each FEA element. Optomec Aerosol Jet system offers an exceptional method to print the electrodes as small as 15 microns width, hence the electrode design can be done with the minimum FEA mesh size of 60 microns. However, this research implemented the design algorithms at a larger mesh size to ensure that the electrodes are cleanly separated.

Once the electrode network is successfully printed using the direct-write process, the mold will then be attached to a Trek high voltage amplifier connected to a signal generator that provides a square wave reference signal. This will produce a high voltage input to the electrode network that is alternated at low frequencies in the 10-50 Hz range for fiber alignment.

References

1. Chung DDL. Carbon fiber composites. Boston: Butterworth-Heinemann, 1994.

2. Kim GH, Moeller D, Shkel YM. Orthotropic Polymeric Composites with Microstructure Tailored by Electric Field. J Composite Mater 2004;38(21):1895.

3. Kim GH, Shkel YM. Polymeric composites tailored by electric field. J Mater Res 2004;19(4):1164-1174.

4. Martin C, Sandler J, Windle A, Schwarz MK, Bauhofer W, Schulte K, Shaffer M. Electric field-induced aligned multi-wall carbon nanotube networks in epoxy composites. Polymer 2005;46(3):877-886.

5. Wang MW, Hsu TC, Weng CH. Alignment of MWCNTs in polymer composites by dielectrophoresis. The European Physical Journal Applied Physics 2008;42(3):241-246.

6. Park C, Wilkinson J, Banda S, Ounaies Z, Wise KE, Sauti G, Lillehei PT, Harrison JS. Aligned single-wall carbon nanotube polymer composites using an electric field. Journal of Polymer Science Part B: Polymer Physics 2006;44(12):1751-1762.

7. Astrom BT. Manufacturing of polymer composites: CRC, 1997.

8. Rosato DV, Rosato DV, Rosato MG, Rosato DV. Injection molding handbook. Boston: Kluwer Academic Publishers, 2000.

9. Menges G, Michaeli W, Mohren P. How to make injection molds: Hanser Gardner Pubns, 2001.

10. Rahmati S, Dickens P. Stereolithography for injection mould tooling. Rapid Prototyping Journal 1997;3(2):53-60.

11. Colton J, Blair B. Experimental study of post-build cure of stereolithography polymers for injection molds. Rapid Prototyping Journal 1999;5(2):72-81.

12. Xu X, Sachs E, Allen S. The design of conformal cooling channels in injection molding tooling. Polymer Engineering & Science 2001;41(7):1265-1279.

13. Nambiar R, Lee K, Nagarajan D. Stereolithography mold life extension using gas-assisted injection. Rapid Prototyping Journal 2007;13(2):92-98.

14. Modak S. Freeform fiber alignment in a polymer matrix using conformal electrode field networks. 2008:71.

15. Crothers P, Drechsler K, Feltin D, Herszberg I, Kruckenberg T. Tailored fibre placement to minimise stress concentrations. Composites Part A: Applied Science and Manufacturing 1997;28(7):619-625.

16. Wetherhold RC, Seelman S, Wang J. The use of functionally graded materials to eliminate or control thermal deformation. Composites Sci Technol 1996;56(9):1099-1104.

17. Suresh S. Graded materials for resistance to contact deformation and damage. Science 2001;292(5526):2447.

18. Pelletier JL, Vel SS. Multi-objective optimization of fiber reinforced composite laminates for strength, stiffness and minimal mass. Comput Struct 2006;84(29-30):2065-2080.

19. Setoodeh S, Abdalla M, Gürdal Z. Combined topology and fiber path design of composite layers using cellular automata. Structural and Multidisciplinary Optimization 2005;30(6):413-421.

20. Moeller DK, Cho HK, Derenne KL, Shkel YM. Micro-tailoring micro-and nano-composites: towards variable orthotropy for bio-mimicking materials. Proceedings of SPIE, 2005. p. 50.

Chapter 5 Conclusions

In this dissertation, algorithms for computational design of mold halves for aligned discontinuous fiber composite components via the FAiMTa process are presented. A technique for rapid tooling was also studied and implemented. To validate the material improvement, FEA models were created to study the effect of preferentially fiber alignment.

Based on the dielectrophoresis effect, the FAiMTa process introduces an effective method to manipulate the carbon fibers in curing epoxy. In Chapter 2, geometric algorithms were developed to design interdigitated electrode integrated mold halves. The procedures include (1) the FEA to obtain the necessary information, (2) the calculation of the principal strains and direction in each FEA element, (3) the determination of the critical areas in which the fibers should be aligned along the strain directions, (4) the generation of coordinates of interdigitated electrode trunks that are normal to the preferred fiber directions, and (5) the generation of coordinates of the main lines of the electrodes.

The proposed algorithms were implemented in the Solidworks® CAD system using the Solidworks® API with the VBA programming environment. Using Solidworks® for the implementation of the proposed algorithms, various geometrical components were used to demonstrate the procedures. Specifically, FEA was conducted on the components using the working conditions and material properties of a randomly aligned fiber composite. From the FEA results, the Solidworks® implementation was used to create sketches representing the electrode networks, which create electric field lines parallel to the desired fiber alignment direction. The implementation successfully produced electrode networks whose resulting electric fields are perpendicular to the direction of the preferred fiber alignment and it can be concluded that the proposed algorithms do, in fact, perform as intended.

In Chapter 3, the effect of fiber alignment was studied to validate that the preferential fiber alignment improves the overall performance of the components. Halpin-Tsai micromechanics model was used to predict the stiffness; including modulus of elasticity and Poisson's ratios, for unidirectionally-aligned fiber composites and a mixed model was used for randomly-aligned fiber composites. The original components were split into sections that corresponded to the FEA critical elements. Those sections were modeled with the orthotropic material properties of unidirectionally aligned fiber composite while the remainders of the component sections were assigned with those of randomly aligned fiber composite. Three different components were demonstrated to illustrate the procedures and the effects of fiber alignment. Histograms were effectively used to illustrate the distribution of strains and the reduction in strain magnitude. The results verify the hypothesis that, with the fibers preferentially aligned, the components have a smaller number of elements that are subjected to high strain and the high strain concentration regions are not present.

In Chapter 4, rapid tooling techniques were presented to demonstrate a fabrication process of mold halves with integrated interdigitated electrode networks. Additive Manufacturing (AM) is a crucial solution to fabrication of molds with high geometry complexity. In the previous version, the customized electrode networks were designed and the coordinates of the branches were kept in a tabular form. Automated procedures were developed to convert the electrode node locations into a text file which was transferred to different machines for direct-write printing techniques. The text file controls the movement of the dispensing head to fabricate the electrode network on the surface of the mold halves. nScrypt microelectronics dispensing machine and nano silver ink dispensing platform successfully fabricated the networks at the order of millimeters. Optomec Aerosol Jet effectively printed the finer electrodes in order to achieve an electrode network with small spacing.

In conclusion, this dissertation offers an opportunity for a mass production of preferentially aligned fiber composite components. The algorithms are able to design the customized electrode networks which are arranged based on the predefined fiber direction. The mold halves integrated with series of electrodes can readily be produced as in the procedures described previously. Subsequently, in reaction injection molding, composite parts with preferentially aligned fiber can be simply fabricated by connecting the electrode main lines to a high voltage supply. Since the data flow is semi-automated starting from the CAD model of the part to the RT fabrication of the mold halves, the time-to-market will be minimal for the fabrication of the aligned fiber composite at a high throughput rate.

However, there are some limitations of the technology. Since the fiber alignment is solely based on the effect of dielectrophoresis, the strength of electric field plays an important role in determining the magnitude of rotational torque. As the strength of electric field rapidly decreases as the distance from the electrode network increases, the fibers can only be manipulated to a thickness of a few millimeters. Thus, this technology and the geometric algorithms merely benefit the fabrication of the components that are practically thin and plate-like. It would be advantageous to develop the process so that it can arrange the fibers at a larger distance away from the electrodes and, possibly, in a three-dimensional space. Promising approaches include increasing the potential of the voltage supply and changing the frequency of the electric field.

For future research, the current algorithms can be extended to include an iteration of the microstructure improvement. Following the determination of critical regions and the modeling of the selectively orthotropic sections of the component, algorithms can be developed to continuously improve the fiber arrangement of the component and the additional collection of electrode networks are to be incorporated into the current ones. The components with fibers aligned in selective regions are analyzed to indicate a possible way to rearrange the fiber alignment for an improved performance. Criteria for the improvement include, but are not limited to, maximum load bearing capability, maximum Tsai-Wu stress, and minimum deflection of the components.

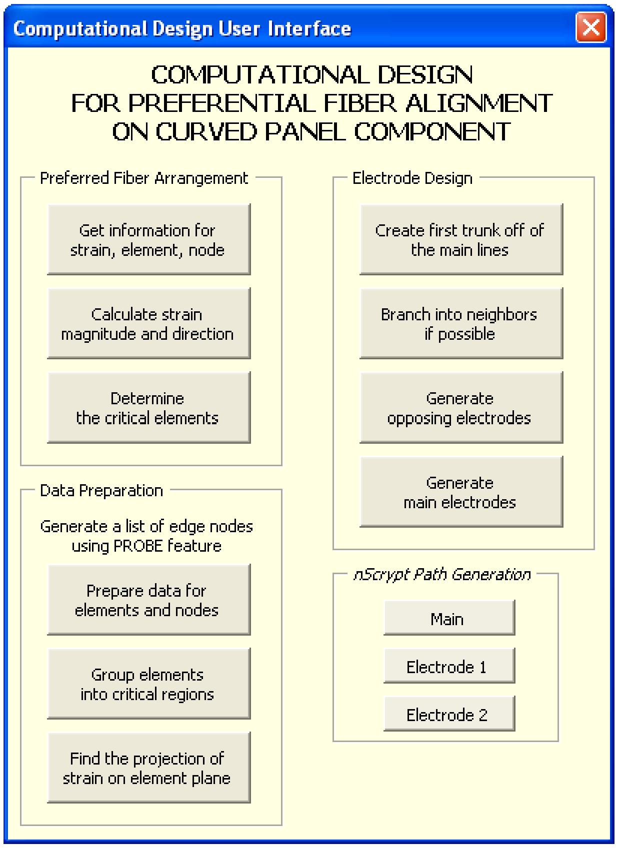
In addition, there is an opportunity to study the distribution of fiber orientation within the non-uniform electric field. Practically, not every fiber is aligned perfectly along the electric field and the distribution function of the deviation should be studied. The distribution function can be used in the FEA of the selectively aligned fiber composite to evaluate the effect of the fiber alignment via FAiMTa. In dielectrophoresis effect, the force exerted on a suspended inclusion depends strongly on the strength of the electric field and the difference between relative dielectric constant of the constituents. Moreover, the torque applied on the particles is affected by the shape and size of the particles and the viscosity of the matrix. As a result, the distribution functions of the fiber orientation are different as the parameters are varied.

Another possible extension is a physical component fabrication to evaluate the overall strength improvement. As the mold halves with integrated electrode networks are fabricated through the freeform electrode printing, experiments can be conducted to injection mold the composite components. Subsequently, the mechanical tests can be done as an alternative method to verify the favorable effect of fiber orientation and the efficiency of the process. Moreover, the results can be compared with those from FEA in order to continuously improve the prediction of the distribution function of the fiber orientation.

Appendix

Appendix A

The algorithms are implemented in Visual Basic for Applications (VBA) as follows:



COMPUTATIONAL DESIGN
FOR PREFERENTIAL FIBER ALIGNMENT
ON CURVED PANEL COMPONENT

Generate a list of edge nodes using PROBE feature

Figure 41 User interface on VBA

Get information for strain, element, node

```
'CONNECT_TO_SW
Set SwApp = Application.SldWorks
'Get Active document
Set ActDoc = COSMOSWORKS.ActiveDoc()
'Get Active study
Set StudyMngr = ActDoc.StudyManager
StudyIndex = StudyMngr.ActiveStudy
Set ActStudy = StudyMngr.GetStudy(StudyIndex)
Set ActMesh = ActStudy.Mesh
'Create the Excel application.
```

```
Set ExcelApp = GetObject(, "Excel.Application")
'Create a new spreadsheet
ExcelApp.ActiveWorkbook.Sheets.Add
ExcelApp.ActiveSheet.Name = "Element_List"
With ExcelApp
    'Label table head in Excel
    .Range("A1").Value = "element no"
    .Range("B1").Value = "X coordinate"
    .Range("C1").Value = "Y coordinate"
    .Range("D1").Value = "Z coordinate"
    .Range("E1").Value = "associated nodes"
End With
ElementArray = ActMesh.GetElements()
NodeArray = ActMesh.GetNodes()
'Start with first element in array
n = 0
Do
    With ExcelApp
    'Write element number
    .Range("A" & Format$(ElementArray(n) + 1)).Select
    .ActiveCell.FormulaR1C1 = Format$(ElementArray(n))
    'Write XYZ coordinate
    .Range("B" & Format$(ElementArray(n) + 1)).Select
    .ActiveCell.FormulaR1C1 = Format$(ElementArray(n + 11) * 1000)
    .Range("C" & Format$(ElementArray(n) + 1)).Select
    .ActiveCell.FormulaR1C1 = Format$(ElementArray(n + 12) * 1000)
    .Range("D" & Format$(ElementArray(n) + 1)).Select
    .ActiveCell.FormulaR1C1 = Format$(ElementArray(n + 13) * 1000)
    'Write associated nodes
    .Range("E" & Format$(ElementArray(n) + 1)).Select
    .ActiveCell.FormulaR1C1 = Format$(ElementArray(n + 1))
    .Range("F" & Format$(ElementArray(n) + 1)).Select
    .ActiveCell.FormulaR1C1 = Format$(ElementArray(n + 2))
    .Range("G" & Format$(ElementArray(n) + 1)).Select
    .ActiveCell.FormulaR1C1 = Format$(ElementArray(n + 3))
    End With
    n = n + 16
'Loop until the last element
Loop While n < UBound(ElementArray)
'Create a new spreadsheet
ExcelApp.ActiveWorkbook.Sheets.Add
ExcelApp.ActiveSheet.Name = "Node_List"
With ExcelApp
    'Label table head in Excel
```

```
    .Range("A1").Value = "node no"
    .Range("B1").Value = "X coordinate"
    .Range("C1").Value = "Y coordinate"
    .Range("D1").Value = "Z coordinate"
End With
NodeArray = ActMesh.GetNodes()
'Start with first node in array
n = 0
Do
    With ExcelApp
        'Write element number
        .Range("A" & Format$(NodeArray(n) + 1)).Value = Format$(NodeArray(n))
        'get total #of nodes
        .Range("E1").Value = Format$(NodeArray(n))
        'Write XYZ coordinate
        .Range("B" & Format$(NodeArray(n) + 1)).Value = Format$(NodeArray(n + 1) * 1000)
        .Range("C" & Format$(NodeArray(n) + 1)).Value = Format$(NodeArray(n + 2) * 1000)
        .Range("D" & Format$(NodeArray(n) + 1)).Value = Format$(NodeArray(n + 3) *
1000)
    End With
    n = n + 4
'Loop until the last element
Loop While n < UBound(NodeArray)
```

Calculate strain magnitude and direction

```
'Create the Excel application.
Set ExcelApp = GetObject(, "Excel.Application")
ExcelApp.Visible = True
Set MatLab = CreateObject("Matlab.Application")
ExcelApp.Worksheets("XYZStrain_List").Select
With ExcelApp
    'Label table head in Excel
    .Range("H5").Value = "Total E"
    .Range("I5").Value = "strain i"
    .Range("J5").Value = "strain j"
    .Range("K5").Value = "strain k"
End With
i = 6
Do
    strain(0, 0) = ExcelApp.Range("B" & Format$(i)).Value
    strain(0, 1) = ExcelApp.Range("E" & Format$(i)).Value
    strain(0, 2) = ExcelApp.Range("F" & Format$(i)).Value
```

```
   strain(1, 0) = ExcelApp.Range("E" & Format$(i)).Value
   strain(1, 1) = ExcelApp.Range("C" & Format$(i)).Value
   strain(1, 2) = ExcelApp.Range("G" & Format$(i)).Value
   strain(2, 0) = ExcelApp.Range("F" & Format$(i)).Value
   strain(2, 1) = ExcelApp.Range("G" & Format$(i)).Value
   strain(2, 2) = ExcelApp.Range("D" & Format$(i)).Value
   Call MatLab.PutWorkspaceData("a", "base", strain)
   Result = MatLab.Execute("[v,d] = eig(a)")
   Call MatLab.GetWorkspaceData("v", "base", evector)
   Call MatLab.GetWorkspaceData("d", "base", evalue)
   'get total strain magnitude
   Result = MatLab.Execute("x = [d(1,1) d(2,2) d(3,3)]")
   Result = MatLab.Execute("mag = norm(x)")
   Call MatLab.GetWorkspaceData("mag", "base", magnitude)
   ExcelApp.Range("H" & Format$(i)).Value = magnitude
   'get strain i j k compenent
   ExcelApp.Range("I" & Format$(i)).Value = evector(0, 0) * evalue(0, 0) + evector(1, 0) *
evalue(1, 1) + evector(2, 0) * evalue(2, 2)
   ExcelApp.Range("J" & Format$(i)).Value = evector(0, 1) * evalue(0, 0) + evector(1, 1) *
evalue(1, 1) + evector(2, 1) * evalue(2, 2)
   ExcelApp.Range("K" & Format$(i)).Value = evector(0, 2) * evalue(0, 0) + evector(1, 2) *
evalue(1, 1) + evector(2, 2) * evalue(2, 2)
   i = i + 1
Loop While IsEmpty(ExcelApp.Range("A" & Format$(i)).Value) = False
```

Determine the critical elements

```
'Create the Excel application.
Set ExcelApp = GetObject(, "Excel.Application")
ExcelApp.Visible = True
With ExcelApp
   .Worksheets("XYZStrain_List").Select
   'Set a threshold at a certain percentage of the maximum stress
   .Range("I1").Value = "=MAX(C[-1])*0.2"
End With
i = 6
Do
   If ExcelApp.Range("H" & Format$(i)).Value > ExcelApp.Range("I1").Value Then
      ExcelApp.Worksheets("Element_List").Range("H" & Format$(i - 4)).Value = "Critical"
   End If
   i = i + 1
Loop While IsEmpty(ExcelApp.Worksheets("XYZStrain_List").Range("A" &
Format$(i)).Value) = False
```

```
'Create a new spreadsheet
ExcelApp.ActiveWorkbook.Sheets.Add
ExcelApp.ActiveSheet.Name = "Critical_List"
    With ExcelApp
        'Label table head in Excel
        .Range("A1").Value = "element no"
        .Range("B1").Value = "vertex1"
        .Range("C1").Value = "vertex2"
        .Range("D1").Value = "vertex3"
        .Range("E1").Value = "strain i"
        .Range("F1").Value = "strain j"
        .Range("G1").Value = "strain k"
        .Range("H1").Value = "elec i"
        .Range("I1").Value = "elec j"
        .Range("J1").Value = "elec k"
    End With
i = 6
k = 2
Do
    'Find elements in critical areas
    If ExcelApp.Worksheets("Element_List").Range("H" & Format$(i)).Value = "Critical"
Then
        With ExcelApp
            'Get info for elements in critical areas
            .Worksheets("Critical_List").Range("A" & Format(k)).Value = i - 1
            .Worksheets("Critical_List").Range("B" & Format(k)).Value = "=VLOOKUP(" &
Format$(i - 1) & ",Element_List!A:G,5,FALSE)"
            .Worksheets("Critical_List").Range("C" & Format(k)).Value = "=VLOOKUP(" &
Format$(i - 1) & ",Element_List!A:G,6,FALSE)"
            .Worksheets("Critical_List").Range("D" & Format(k)).Value = "=VLOOKUP(" &
Format$(i - 1) & ",Element_List!A:G,7,FALSE)"
            .Worksheets("Critical_List").Range("E" & Format(k)).Value = "=VLOOKUP(" &
Format$(i - 1) & ",XYZStrain_List!A:K,9,FALSE)"
            .Worksheets("Critical_List").Range("F" & Format(k)).Value = "=VLOOKUP(" &
Format$(i - 1) & ",XYZStrain_List!A:K,10,FALSE)"
            .Worksheets("Critical_List").Range("G" & Format(k)).Value = "=VLOOKUP(" &
Format$(i - 1) & ",XYZStrain_List!A:K,11,FALSE)"
            .Worksheets("Element_List").Range("I" & Format(i)).Value = k
        End With
    k = k + 1
    End If
    i = i + 1
Loop While IsEmpty(ExcelApp.Worksheets("XYZStrain_List").Range("A" &
Format$(i)).Value) = False
```

Prepare data for elements and nodes

```
'Create the Excel application.
Set ExcelApp = GetObject(, "Excel.Application")
ExcelApp.Visible = True
With ExcelApp
   .Worksheets("Node_list").Select
   .Range("F1").Value = "No. of critical elements"
   .Range("G1").Value = "No. of elements matched"
   .Range("H1").Value = "Associated element index"
End With
i = 2
Do
   'Count number of repeated nodes in the element list
   With ExcelApp
      .Worksheets("Node_list").Select
      .Range("E" & Format$(i)).Value = "=IFERROR(VLOOKUP(A" & Format$(i) &
",Edge_List!A:A,1,FALSE)>0," & Chr(34) & "nonedge" & Chr(34) & ")"
      .Range("F" & Format$(i)).Value = "=COUNTIF(Critical_List!C[-4]:C[-2],RC1)"
      .Range("G" & Format$(i)).Value = "=COUNT(RC[1]:RC[8])"
   End With
   i = i + 1
Loop While IsEmpty(ExcelApp.Worksheets("Node_List").Range("A" & Format$(i)).Value)
= False
i = 2
Do
   'Find composite element that has only 1 node that is in NonEdge List
   'each if statement returns value of 1 if it is NonEdge node
   ExcelApp.Worksheets("Critical_List").Range("K" & Format$(i)).Value =
"=IF(ISNA(MATCH(RC2,Edge_List!C[-
10],0)),0,1)+IF(ISNA(MATCH(RC3,Edge_List!C[-
10],0)),0,1)+IF(ISNA(MATCH(RC4,Edge_List!C[-10],0)),0,1)"
   'If an element has more than one node on edge then it is an element on edge
   If ExcelApp.Worksheets("Critical_List").Range("K" & Format$(i)).Value <= 1 Then
      ExcelApp.Worksheets("Critical_List").Range("L" & Format$(i)).Value = "NonEdge"
   Else
      ExcelApp.Worksheets("Critical_List").Range("L" & Format$(i)).Value = "Edge"
   End If
   i = i + 1
Loop While IsEmpty(ExcelApp.Worksheets("Critical_List").Range("A" &
Format$(i)).Value) = False
'Repeat for each node
i = 2
Do
```

```
    'If the node has any associated composite element
    If ExcelApp.Worksheets("Node_List").Range("F" & Format$(i)).Value Then
        k = 8
        'Check each critical element
        j = 2
        Do
            ExcelApp.Worksheets("Node_list").Select
            ExcelApp.Cells(i, k).Select
            ExcelApp.ActiveCell.Value = "=MATCH(RC1,Critical_List!R" & Format$(j) &
"C2:R" & Format$(j) & "C4,0)"
            a = ExcelApp.ActiveCell.Value
            'If the element is assocaited with the node in question
            If IsError(ExcelApp.Worksheets("Node_List").Cells(i, k).Value) = False Then
                'Put the element number in the node list
                ExcelApp.ActiveCell.Value = ExcelApp.Worksheets("Critical_List").Cells(j,
1).Value
                k = k + 1
            End If
            j = j + 1
        Loop While (ExcelApp.Worksheets("Node_List").Range("F" & Format$(i)) <>
ExcelApp.Worksheets("Node_List").Range("G" & Format$(i)))
    End If
    i = i + 1
Loop While IsEmpty(ExcelApp.Worksheets("Node_List").Range("A" & Format$(i)).Value)
= False
```

Group elements into critical regions

```
'Create the Excel application.
Set ExcelApp = GetObject(, "Excel.Application")
ExcelApp.Visible = True
'Create a new spreadsheet
With ExcelApp
    .ActiveWorkbook.Sheets.Add
    .ActiveSheet.Name = "Groups_of_elements"
    .Range("A1").Value = "Group number"
    .Range("B1").Value = "Number of elements"
    .Range("D1").Value = "Name of electrodes"
End With
'i is the critical element
i = 2
'j is the group number
j = 0
```

```
Do
    ExcelApp.Sheets("Critical_list").Select
    'If the element is not already in another group
    If ExcelApp.Range("M" & Format$(i)).Value <> "Grouped" Then
        'Create another group
        j = j + 1
        With ExcelApp
            'List group number
            .Worksheets("Groups_of_elements").Select
            .Range("A" & Format$(j + 1)).Value = j
            'Count number of associated element in the group
            .Range("B" & Format$(j + 1)).Value =
"=COUNTIF(Critical_List!C[12],""=""&RC1)"
            'mark the element as grouped
            .Worksheets("Critical_List").Range("M" & Format$(i)).Value = "Grouped"
            'Assign group number to the  element
            .Worksheets("Critical_List").Range("N" & Format$(i)).Value = j
        End With
        'Call sub procedure
        Call FindNeighbor(i, j)
    'If the element is already in another group
    Else
        GroupIndex = ExcelApp.Worksheets("Critical_List").Range("N" & Format$(i)).Value
        Call FindNeighbor(i, GroupIndex)
    End If
i = i + 1
Loop While IsEmpty(ExcelApp.Worksheets("Critical_List").Range("A" &
Format$(i)).Value) = False
'LIST NODES ON MAIN ELECTRODES
'Create a new spreadsheet
ExcelApp.ActiveWorkbook.Sheets.Add
ExcelApp.ActiveSheet.Name = "Main_Electrode"
'For each group
c = 1
Do
'For each composite element
i = 2
Do
    'if the composite element is on the edge and it is in the group
    If ExcelApp.Worksheets("Critical_List").Range("L" & Format$(i)).Value = "Edge" And
ExcelApp.Worksheets("Critical_List").Range("N" & Format$(i)).Value = c Then
        'For all three nodes
        For a = 2 To 4
            'Get node number
```

```
      j = ExcelApp.Worksheets("Critical_List").Cells(Format$(i), a).Value
      'get the number of edge node in the group so far
      k = ExcelApp.Worksheets("Groups_of_elements").Range("C" & Format$(c +
1)).Value
      'Check if the node is on edge
      ExcelApp.Worksheets("Main_Electrode").Cells(c, (k + 1)).Value = "=MATCH(" &
Format$(j) & ",Edge_List!A:A,0)"
      'if the number matches edge node list
      If IsError(ExcelApp.Worksheets("Main_Electrode").Cells(c, (k + 1)).Value) = False
Then
        'Check if the node is already in the electrode
        If ExcelApp.Worksheets("Node_List").Range("P" & Format$(j + 1)).Value <>
"checked" Then
          'Write node number
          ExcelApp.Worksheets("Main_Electrode").Cells(c, (k + 1)).Value = j
          'record number of node so far
          ExcelApp.Worksheets("Groups_of_elements").Range("C" & Format$(c +
1)).Value = k + 1
          ExcelApp.Worksheets("Node_List").Range("P" & Format$(j + 1)).Value =
"checked"
        Else
          'if node is already checked then clear the cell
          ExcelApp.Worksheets("Main_Electrode").Cells(c, (k + 1)).Select
          ExcelApp.Selection.ClearContents
        End If
      Else
        'if node is not on the edge then clear the cell
        ExcelApp.Worksheets("Main_Electrode").Cells(c, (k + 1)).Select
        ExcelApp.Selection.ClearContents
      End If
    Next
  End If
i = i + 1
Loop While IsEmpty(ExcelApp.Worksheets("Critical_List").Range("A" &
Format$(i)).Value) = False
c = c + 1
Loop While IsEmpty(ExcelApp.Worksheets("Groups_of_elements").Range("A" &
Format$(c)).Value) = False
```

Sub procedure FindNeighbor

```
'Find the neighbor of the element and put in the table
'i is critical element index and j is group
```

```
For a = 2 To 4
   m = ExcelApp.Worksheets("Critical_List").Cells(i, a).Value
   'm is the node we are looking at
   n = ExcelApp.Worksheets("Node_List").Cells((m + 1), 6).Value
   'n is number of associated critical elements
   For b = 1 To n
      l = ExcelApp.Worksheets("Node_List").Cells((m + 1), (b + 7)).Value
      'l is composite element
      ExcelApp.Worksheets("Node_List").Range("J1").Value = "=MATCH(" & Format$(l)
& ",Critical_List!A:A,0)"
      CompositeIndex = ExcelApp.Worksheets("Node_List").Range("J1").Value
      ExcelApp.Worksheets("Node_List").Range("J1").Value = ""
      If ExcelApp.Worksheets("Critical_List").Range("M" & Format$(CompositeIndex)) <>
"Grouped" Then
         ExcelApp.Worksheets("Critical_List").Range("M" & Format$(CompositeIndex)) =
"Grouped"
         ExcelApp.Worksheets("Critical_List").Range("N" & Format$(CompositeIndex)) = j
      End If
   Next b
Next a
```

Find the projection of strain on element plane

```
'Create the Excel application.
Set ExcelApp = GetObject(, "Excel.Application")
ExcelApp.Visible = True
Set MatLab = CreateObject("Matlab.Application")
ExcelApp.Worksheets("Critical_List").Select
i = 2
Do
   'get node numbers
   vertex(0, 0) = ExcelApp.Range("B" & Format$(i)).Value
   vertex(1, 0) = ExcelApp.Range("C" & Format$(i)).Value
   vertex(2, 0) = ExcelApp.Range("D" & Format$(i)).Value
   'get node XYZ coordinate
   For a = 0 To 2
      vertex(a, 1) = ExcelApp.Worksheets("Node_List").Range("B" & Format$(vertex(a, 0) +
1)).Value
      vertex(a, 2) = ExcelApp.Worksheets("Node_List").Range("C" & Format$(vertex(a, 0) +
1)).Value
      vertex(a, 3) = ExcelApp.Worksheets("Node_List").Range("D" & Format$(vertex(a, 0) +
1)).Value
   Next a
```

```
    For b = 0 To 2
        strain(0, b) = ExcelApp.Worksheets("Critical_List").Cells(i, b + 5).Value
    Next b
    Call MatLab.PutWorkspaceData("x", "base", vertex)
    Call MatLab.PutWorkspaceData("strain", "base", strain)
    'find the normal vector of the plane from 2 vectors
    Result = MatLab.Execute("ab = x(2,2:4) - x(1,2:4)")
    Result = MatLab.Execute("ac = x(3,2:4) - x(1,2:4)")
    Result = MatLab.Execute("plane = cross(ab, ac)")
    Result = MatLab.Execute("n = plane/ norm(plane)")
    'find the projection of the strain vector on the plane
    Result = MatLab.Execute("v = strain(1,1:3)")
    Result = MatLab.Execute("y = (v - dot(v, n) * n)/ norm(v)")
    Result = MatLab.Execute("z = cross(n, y)")
    'output the projection of the strain to excel
    Call MatLab.GetWorkspaceData("z", "base", elec)
    ExcelApp.Worksheets("Critical_List").Range("H" & Format$(i)).Value = elec(0, 0)
    ExcelApp.Worksheets("Critical_List").Range("I" & Format$(i)).Value = elec(0, 1)
    ExcelApp.Worksheets("Critical_List").Range("J" & Format$(i)).Value = elec(0, 2)
    i = i + 1
Loop While IsEmpty(ExcelApp.Range("A" & Format$(i)).Value) = False
```

Create first trunk off of the main lines

```
'Create the Excel application.
Set ExcelApp = GetObject(, "Excel.Application")
n = 1
'for each group
Do
    'Create a new spreadsheet
    ExcelApp.ActiveWorkbook.Sheets.Add
    ExcelApp.ActiveSheet.Name = "Electrode1_" & Format$(n)
i = 2
b = 1
Do
    If ExcelApp.Worksheets("Critical_List").Range("N" & Format$(i)).Value = n Then
    If ExcelApp.Worksheets("Critical_List").Range("L" & Format$(i)).Value = "Edge" Then
    'CompositeIndex is index of ElemX in Critical_List
    CompositeIndex = i
    ElemX = ExcelApp.Worksheets("Critical_List").Range("A" & Format$(i)).Value
    'Find nodes on edge
    NodeX = ExcelApp.Worksheets("Critical_List").Cells(i, 2).Value
```

```
   ExcelApp.Worksheets("Critical_List").Range("K" & Format$(i)).Value =
"=IF(ISNA(VLOOKUP(" & Format$(NodeX) & ",Edge_List!A:A,1,FALSE)),1,0)"
   'if 1 then nodeX is not on edge
      If ExcelApp.Worksheets("Critical_List").Range("K" & Format$(i)).Value = 1 Then
         nodeC = NodeX
         'the other two nodes are on edge
         nodeA = ExcelApp.Worksheets("Critical_List").Cells(i, 3).Value
         nodeB = ExcelApp.Worksheets("Critical_List").Cells(i, 4).Value
      Else
         'it is on the edge
         nodeA = NodeX
         'find another one on edge
         NodeX = ExcelApp.Worksheets("Critical_List").Cells(i, 3).Value
         ExcelApp.Worksheets("Critical_List").Range("K" & Format$(i)).Value =
"=IF(ISNA(VLOOKUP(" & Format$(NodeX) & ",Edge_List!A:A,1,FALSE)),1,0)"
         'if 1 then nodeX is not on edge
         If ExcelApp.Worksheets("Critical_List").Range("K" & Format$(i)).Value = 1 Then
            nodeC = NodeX
            nodeB = ExcelApp.Worksheets("Critical_List").Cells(i, 4).Value
         'else nodeX is on edge
         Else
            nodeB = NodeX
            nodeC = ExcelApp.Worksheets("Critical_List").Cells(i, 4).Value
         End If
      End If
      If ExcelApp.Worksheets("Critical_List").Range("M" &
Format$(CompositeIndex)).Value <> "electrode" Then
         'Mark the element
         ExcelApp.Worksheets("Critical_List").Range("M" & Format$(CompositeIndex)) =
"electrode"
         ExcelApp.Worksheets("Critical_List").Range("P" & Format$(CompositeIndex)) =
nodeC
         ExcelApp.Worksheets("Critical_List").Range("U" & Format$(CompositeIndex)) = b
         'generate electrode
         Call SketchElectrode(CompositeIndex, ElemX, nodeA, nodeB, nodeC)
         b = b + 1
      End If
   End If
   End If
   i = i + 1
Loop While IsEmpty(ExcelApp.Worksheets("Critical_List").Range("A" &
Format$(i)).Value) = False
n = n + 1
```

```
Loop While IsEmpty(ExcelApp.Worksheets("Main_Electrode").Range("A" &
Format$(n)).Value) = False
```

Subprocedure SketchElectrode

```
Set SwApp = CreateObject("SldWorks.Application")
Set swModel = SwApp.ActiveDoc
Set swSelMgr = swModel.SelectionManager
Set swSketchMgr = swModel.SketchManager
Set swModeler = SwApp.GetModeler
Set swFeatMgr = swModel.FeatureManager
Set swPart = swModel
'Create the Excel application.
Set ExcelApp = GetObject(, "Excel.Application")
Set MatLab = CreateObject("Matlab.Application")
'Get node location from Excel
For a = 1 To 3
   vertex(0, a) = ExcelApp.Worksheets("Node_List").Cells(Format$(nodeA + 1), a +
1).Value
   vertex(1, a) = ExcelApp.Worksheets("Node_List").Cells(Format$(nodeB + 1), a +
1).Value
   vertex(2, a) = ExcelApp.Worksheets("Node_List").Cells(Format$(nodeC + 1), a +
1).Value
Next a
Set swModelDocExt = swModel.Extension
'Open a sketch
swSketchMgr.Insert3DSketch True
swModel.ClearSelection2 True
'Get the mid point between nodeA and nodeB as a starting point
vertex(3, 1) = (vertex(0, 1) + vertex(1, 1)) / 2
vertex(3, 2) = (vertex(0, 2) + vertex(1, 2)) / 2
vertex(3, 3) = (vertex(0, 3) + vertex(1, 3)) / 2
'Calculate the end point using i j k of the strain projection
strain(0, 0) = ExcelApp.Worksheets("Critical_List").Range("H" &
Format$(CompositeIndex))
strain(0, 1) = ExcelApp.Worksheets("Critical_List").Range("I" &
Format$(CompositeIndex))
strain(0, 2) = ExcelApp.Worksheets("Critical_List").Range("J" &
Format$(CompositeIndex))
vertex(4, 1) = vertex(3, 1) + strain(0, 0)
vertex(4, 2) = vertex(3, 2) + strain(0, 1)
vertex(4, 3) = vertex(3, 3) + strain(0, 2)
Call MatLab.PutWorkspaceData("x", "base", vertex)
```

```
'get initial electrode line
Result = MatLab.Execute("a1 = x(4,2:4)")
Result = MatLab.Execute("a2 = x(5,2:4)")
'get line connecting nodeA and nodeC
Result = MatLab.Execute("b1 = x(1,2:4)")
Result = MatLab.Execute("b2 = x(3,2:4)")
'find the intersection point
Result = MatLab.Execute("ts = [a2(:)-a1(:), -(b2(:)-b1(:))] \ (b1(:) - a1(:))")
Result = MatLab.Execute("c1 = a1 + (a2-a1) * ts(1)")
'get line connecting nodeB and nodeC
Result = MatLab.Execute("b1 = x(2,2:4)")
Result = MatLab.Execute("b2 = x(3,2:4)")
'find the intersection point
Result = MatLab.Execute("ts = [a2(:)-a1(:), -(b2(:)-b1(:))] \ (b1(:) - a1(:))")
Result = MatLab.Execute("c2 = a1 + (a2-a1) * ts(1)")
'calculate the length of each line
Result = MatLab.Execute("l1 = norm(a1 - c1)")
Call MatLab.GetWorkspaceData("l1", "base", DistX)
Result = MatLab.Execute("l2 = norm(a1 - c2)")
Call MatLab.GetWorkspaceData("l2", "base", DistY)
If DistX < DistY Then
   Call MatLab.GetWorkspaceData("c1", "base", endpt)
   ExcelApp.Worksheets("Critical_List").Range("O" & Format$(CompositeIndex)).Value =
nodeA
Else
   Call MatLab.GetWorkspaceData("c2", "base", endpt)
   ExcelApp.Worksheets("Critical_List").Range("O" & Format$(CompositeIndex)).Value =
nodeB
End If
Set swSketchSeg(0) = swSketchMgr.CreateLine(vertex(3, 1) / 1000, vertex(3, 2) / 1000,
vertex(3, 3) / 1000, endpt(0, 0) / 1000, endpt(0, 1) / 1000, endpt(0, 2) / 1000)
'Change the name of the open sketch to element number
Set swSketch = swModel.GetActiveSketch2
Set swFeat = swSketch
swFeat.Name = "e" & ElemX
swModel.ClearSelection2 True
'Get group number
n = ExcelApp.Worksheets("Critical_List").Range("N" & Format$(CompositeIndex))
'record the end point in excel for the neighbor
With ExcelApp
   c = ExcelApp.Worksheets("Critical_List").Range("U" & Format$(CompositeIndex))
   .Worksheets("Critical_List").Range("Q" & Format$(CompositeIndex)).Value = endpt(0,
0)
```

```
   .Worksheets("Critical_List").Range("R" & Format$(CompositeIndex)).Value = endpt(0, 1)
   .Worksheets("Critical_List").Range("S" & Format$(CompositeIndex)).Value = endpt(0, 2)

   .Worksheets("Electrode1_" & Format$(n)).Range("A" & Format$(c)).Value = vertex(3, 1)
   .Worksheets("Electrode1_" & Format$(n)).Range("B" & Format$(c)).Value = vertex(3, 2)
   .Worksheets("Electrode1_" & Format$(n)).Range("C" & Format$(c)).Value = vertex(3, 3)
   .Worksheets("Electrode1_" & Format$(n)).Range("D" & Format$(c)).Value = endpt(0, 0)
   .Worksheets("Electrode1_" & Format$(n)).Range("E" & Format$(c)).Value = endpt(0, 1)
   .Worksheets("Electrode1_" & Format$(n)).Range("F" & Format$(c)).Value = endpt(0, 2)
End With
'Exit sketch
swSketchMgr.Insert3DSketch False
```

Branch into neighbors if possible

```
'Create the Excel application.
Set ExcelApp = GetObject(, "Excel.Application")
ExcelApp.Worksheets("Critical_List").Range("N1").Value = "=COUNTIF(M:M," &
Chr(34) & "Electrode" & Chr(34) & ")"
i = 2
Do
   If ExcelApp.Worksheets("Critical_List").Range("M" & Format$(i)).Value = "electrode"
Then
      GroupIndex = ExcelApp.Worksheets("Critical_List").Range("N" & Format$(i)).Value
      nodeA = ExcelApp.Worksheets("Critical_List").Range("O" & Format$(i)).Value
      nodeB = ExcelApp.Worksheets("Critical_List").Range("P" & Format$(i)).Value
      ExcelApp.Worksheets("Critical_List").Range("M" & Format$(i)).Value = "end"
      'Find a critical element that shares nodeC and m
      a = 2
      Do
         element = 0
         'Only look for element with same group number
         If ExcelApp.Worksheets("Critical_List").Range("N" & Format$(a)).Value =
GroupIndex Then
         If a <> CompositeIndex And ExcelApp.Worksheets("Critical_List").Range("M" &
Format$(a)).Value = "Grouped" Then
            For b = 2 To 4
               l = ExcelApp.Worksheets("Critical_List").Cells(a, b).Value
               If l = nodeA Or l = nodeB Then
                  element = element + 1
               End If
            Next b
            If element = 2 Then
```

```
               ExcelApp.Worksheets("Critical_List").Range("T" & Format$(a)).Value = i
               ExcelApp.Worksheets("Critical_List").Range("U" & Format$(a)).Value = ExcelApp.Worksheets("Critical_List").Range("U" & Format$(i)).Value
               ExcelApp.Worksheets("Critical_List").Range("M" & Format$(a)).Value = "neighbor"
               ExcelApp.Worksheets("Critical_List").Range("M" & Format$(i)).Value = "done"
            End If
         End If
         End If
         a = a + 1
      Loop While IsEmpty(ExcelApp.Worksheets("Critical_List").Range("A" & Format$(a)).Value) = False
   End If
   i = i + 1
Loop While IsEmpty(ExcelApp.Worksheets("Critical_List").Range("A" & Format$(i)).Value) = False
'sketch the neighboring electrodes
i = 2
Do
   If ExcelApp.Worksheets("Critical_List").Range("M" & Format$(i)).Value = "neighbor" Then
      Call SketchNeighbor(i)
      ExcelApp.Worksheets("Critical_List").Range("M" & Format$(i)).Value = "electrode"
   End If
i = i + 1
Loop While IsEmpty(ExcelApp.Worksheets("Critical_List").Range("A" & Format$(i)).Value) = False

'show if more electrodes can be created
If ExcelApp.Worksheets("Critical_List").Range("N1").Value = 0 Then
   MessageBox = MsgBox("Process completed with NO more electrode", 0)
Else
   MessageBox = MsgBox("Process completed with more electrode ", 0)
End If
```

Subprocedure SketchNeighbor

```
Set SwApp = CreateObject("SldWorks.Application")
Set swModel = SwApp.ActiveDoc
Set swSelMgr = swModel.SelectionManager
Set swSketchMgr = swModel.SketchManager
Set swModeler = SwApp.GetModeler
```

```
Set swFeatMgr = swModel.FeatureManager
Set swPart = swModel
'Create the Excel application.
Set ExcelApp = GetObject(, "Excel.Application")
Set MatLab = CreateObject("Matlab.Application")
'get the original element to branch from
CompositeIndex = ExcelApp.Worksheets("Critical_List").Range("T" & Format$(i)).Value
nodeA = ExcelApp.Worksheets("Critical_List").Range("O" &
Format$(CompositeIndex)).Value
nodeB = ExcelApp.Worksheets("Critical_List").Range("P" &
Format$(CompositeIndex)).Value
'Find the 3rd node of i
For a = 2 To 4
If ExcelApp.Worksheets("Critical_List").Cells(i, a).Value <> nodeA And
ExcelApp.Worksheets("Critical_List").Cells(i, a).Value <> nodeB Then
   nodeC = ExcelApp.Worksheets("Critical_List").Cells(i, a).Value
   ExcelApp.Worksheets("Critical_List").Range("P" & Format$(i)).Value = nodeC
End If
Next a
'Get node location from Excel
For j = 1 To 3
   vertex(0, j) = ExcelApp.Worksheets("Node_List").Cells(Format$(nodeA + 1), j + 1).Value
   vertex(1, j) = ExcelApp.Worksheets("Node_List").Cells(Format$(nodeB + 1), j + 1).Value
   vertex(2, j) = ExcelApp.Worksheets("Node_List").Cells(Format$(nodeC + 1), j + 1).Value
Next j
Set swModelDocExt = swModel.Extension
'Open a sketch
swSketchMgr.Insert3DSketch True
swModel.ClearSelection2 True
'Get the end point of origin element as a starting point
vertex(3, 1) = ExcelApp.Worksheets("Critical_List").Range("Q" &
Format$(CompositeIndex)).Value
vertex(3, 2) = ExcelApp.Worksheets("Critical_List").Range("R" &
Format$(CompositeIndex)).Value
vertex(3, 3) = ExcelApp.Worksheets("Critical_List").Range("S" &
Format$(CompositeIndex)).Value
'Calculate the end point using i j k of the strain projection
strain(0, 0) = ExcelApp.Worksheets("Critical_List").Range("H" & Format$(i))
strain(0, 1) = ExcelApp.Worksheets("Critical_List").Range("I" & Format$(i))
strain(0, 2) = ExcelApp.Worksheets("Critical_List").Range("J" & Format$(i))
vertex(4, 1) = vertex(3, 1) + strain(0, 0)
vertex(4, 2) = vertex(3, 2) + strain(0, 1)
vertex(4, 3) = vertex(3, 3) + strain(0, 2)
Call MatLab.PutWorkspaceData("x", "base", vertex)
```

```
'get initial electrode line
Result = MatLab.Execute("a1 = x(4,2:4)")
Result = MatLab.Execute("a2 = x(5,2:4)")
'get line connecting nodeA and nodeC
Result = MatLab.Execute("b1 = x(1,2:4)")
Result = MatLab.Execute("b2 = x(3,2:4)")
'find the intersection point
Result = MatLab.Execute("ts = [a2(:)-a1(:), -(b2(:)-b1(:))] \ (b1(:) - a1(:))")
Result = MatLab.Execute("c1 = a1 + (a2-a1) * ts(1)")
'get line connecting nodeB and nodeC
Result = MatLab.Execute("b1 = x(2,2:4)")
Result = MatLab.Execute("b2 = x(3,2:4)")
'find the intersection point
Result = MatLab.Execute("ts = [a2(:)-a1(:), -(b2(:)-b1(:))] \ (b1(:) - a1(:))")
Result = MatLab.Execute("c2 = a1 + (a2-a1) * ts(1)")
'calculate the length of each line
Result = MatLab.Execute("l1 = norm(a1 - c1)")
Call MatLab.GetWorkspaceData("l1", "base", DistX)
Result = MatLab.Execute("l2 = norm(a1 - c2)")
Call MatLab.GetWorkspaceData("l2", "base", DistY)
If DistX < DistY Then
   Call MatLab.GetWorkspaceData("c1", "base", endpt)
   ExcelApp.Worksheets("Critical_List").Range("O" & Format$(i)).Value = nodeA
Else
   Call MatLab.GetWorkspaceData("c2", "base", endpt)
   ExcelApp.Worksheets("Critical_List").Range("O" & Format$(i)).Value = nodeB
End If
Set swSketchSeg(0) = swSketchMgr.CreateLine(vertex(3, 1) / 1000, vertex(3, 2) / 1000,
vertex(3, 3) / 1000, endpt(0, 0) / 1000, endpt(0, 1) / 1000, endpt(0, 2) / 1000)
'Change the name of the open sketch to element number
ElemX = ExcelApp.Worksheets("Critical_List").Range("A" & Format$(i)).Value
Set swSketch = swModel.GetActiveSketch2
Set swFeat = swSketch
swFeat.Name = "e" & ElemX
swModel.ClearSelection2 True
'Get group number
n = ExcelApp.Worksheets("Critical_List").Range("N" & Format$(i))
'Get the row number
c = ExcelApp.Worksheets("Critical_List").Range("U" & Format$(i))
'Get the empty cell in the row
b = 1
m = 0
Do
```

```
   If IsEmpty(ExcelApp.Worksheets("Electrode1_" & Format$(n)).Cells(c, b).Value) = True
Then
      m = 1
   End If
   b = b + 1
Loop While m <> 1
'record the end point in excel for the neighbor
With ExcelApp
   .Worksheets("Critical_List").Range("Q" & Format$(i)).Value = endpt(0, 0)
   .Worksheets("Critical_List").Range("R" & Format$(i)).Value = endpt(0, 1)
   .Worksheets("Critical_List").Range("S" & Format$(i)).Value = endpt(0, 2)
   .Worksheets("Electrode1_" & Format$(n)).Cells(c, b - 1).Value = endpt(0, 0)
   .Worksheets("Electrode1_" & Format$(n)).Cells(c, b).Value = endpt(0, 1)
   .Worksheets("Electrode1_" & Format$(n)).Cells(c, b + 1).Value = endpt(0, 2)
End With
'Exit sketch
swSketchMgr.Insert3DSketch False
```

Generate opposing electrodes

```
Private Sub CommandButton12_Click()
'CREATE ANOTHER SET OF ELECTRODES
Set SwApp = CreateObject("SldWorks.Application")
Set swModel = SwApp.ActiveDoc
Set swSelMgr = swModel.SelectionManager
Set swSketchMgr = swModel.SketchManager
Set swModeler = SwApp.GetModeler
Set swFeatMgr = swModel.FeatureManager
Set swPart = swModel
'Create the Excel application.
Set ExcelApp = GetObject(, "Excel.Application")
'Create Matlab application
Set MatLab = CreateObject("Matlab.Application")
Dim distance As Variant
n = 1
'for each group
Do
   'Create a new spreadsheet
   ExcelApp.ActiveWorkbook.Sheets.Add
   ExcelApp.ActiveSheet.Name = "Electrode2_" & Format$(n)
i = 1
Do
   'get start pt of the 1st line
```

```
    vertex(0, 1) = ExcelApp.Worksheets("Electrode1_" & Format$(n)).Range("A" &
Format$(i)).Value
    vertex(0, 2) = ExcelApp.Worksheets("Electrode1_" & Format$(n)).Range("B" &
Format$(i)).Value
    vertex(0, 3) = ExcelApp.Worksheets("Electrode1_" & Format$(n)).Range("C" &
Format$(i)).Value
    'get start pt of the 2nd line
    vertex(1, 1) = ExcelApp.Worksheets("Electrode1_" & Format$(n)).Range("A" &
Format$(i + 1)).Value
    vertex(1, 2) = ExcelApp.Worksheets("Electrode1_" & Format$(n)).Range("B" &
Format$(i + 1)).Value
    vertex(1, 3) = ExcelApp.Worksheets("Electrode1_" & Format$(n)).Range("C" &
Format$(i + 1)).Value
    'get mid pt between 2 start pts
    vertex(2, 1) = (vertex(0, 1) + vertex(1, 1)) / 2
    vertex(2, 2) = (vertex(0, 2) + vertex(1, 2)) / 2
    vertex(2, 3) = (vertex(0, 3) + vertex(1, 3)) / 2
    'get the end pt of the 1st line
    vertex(3, 1) = ExcelApp.Worksheets("Electrode1_" & Format$(n)).Range("D" &
Format$(i)).Value
    vertex(3, 2) = ExcelApp.Worksheets("Electrode1_" & Format$(n)).Range("E" &
Format$(i)).Value
    vertex(3, 3) = ExcelApp.Worksheets("Electrode1_" & Format$(n)).Range("F" &
Format$(i)).Value
    'get distance of the 1st line
    XVal = vertex(3, 1) - vertex(0, 1)
    YVal = vertex(3, 2) - vertex(0, 2)
    ZVal = vertex(3, 3) - vertex(0, 3)
    'export data to matlab
    Call MatLab.PutWorkspaceData("x", "base", XVal)
    Call MatLab.PutWorkspaceData("y", "base", YVal)
    Call MatLab.PutWorkspaceData("z", "base", ZVal)
    'find unit vector // to electrode
    Result = MatLab.Execute("n = [x y z]/norm([x y z])")
    Call MatLab.GetWorkspaceData("n", "base", distance)
    'start pt = mid pt + unit vector * half of element size
    StartX = vertex(2, 1) + distance(0, 0) * 2
    StartY = vertex(2, 2) + distance(0, 1) * 2
    StartZ = vertex(2, 3) + distance(0, 2) * 2
    'end pt increments
    EndX = distance(0, 0) * 2 + vertex(2, 1) - vertex(0, 1)
    EndY = distance(0, 1) * 2 + vertex(2, 2) - vertex(0, 2)
    EndZ = distance(0, 2) * 2 + vertex(2, 3) - vertex(0, 3)
    'record start pt in excel
```

```
   With ExcelApp
      .Worksheets("Electrode2_" & Format$(n)).Range("A" & Format$(i)).Value = StartX
      .Worksheets("Electrode2_" & Format$(n)).Range("B" & Format$(i)).Value = StartY
      .Worksheets("Electrode2_" & Format$(n)).Range("C" & Format$(i)).Value = StartZ
   End With
   'record end pt in excel
   j = 4
   Do
      ExcelApp.Worksheets("Electrode2_" & Format$(n)).Cells(i, j).Value =
ExcelApp.Worksheets("Electrode1_" & Format$(n)).Cells(i, j).Value + EndX
      ExcelApp.Worksheets("Electrode2_" & Format$(n)).Cells(i, j + 1).Value =
ExcelApp.Worksheets("Electrode1_" & Format$(n)).Cells(i, j + 1).Value + EndY
      ExcelApp.Worksheets("Electrode2_" & Format$(n)).Cells(i, j + 2).Value =
ExcelApp.Worksheets("Electrode1_" & Format$(n)).Cells(i, j + 2).Value + EndZ
   j = j + 3
   Loop While IsEmpty(ExcelApp.Worksheets("Electrode1_" & Format$(n)).Cells(i,
j).Value) = False
i = i + 1
Loop While IsEmpty(ExcelApp.Worksheets("Electrode1_" & Format$(n)).Range("A" &
Format$(i)).Value) = False
   i = 1
   Do
      j = 1
      Do
         'get start and end pt
         StartX = ExcelApp.Worksheets("Electrode2_" & Format$(n)).Cells(i, j).Value
         StartY = ExcelApp.Worksheets("Electrode2_" & Format$(n)).Cells(i, j + 1).Value
         StartZ = ExcelApp.Worksheets("Electrode2_" & Format$(n)).Cells(i, j + 2).Value
         EndX = ExcelApp.Worksheets("Electrode2_" & Format$(n)).Cells(i, j + 3).Value
         EndY = ExcelApp.Worksheets("Electrode2_" & Format$(n)).Cells(i, j + 4).Value
         EndZ = ExcelApp.Worksheets("Electrode2_" & Format$(n)).Cells(i, j + 5).Value
         Set swModelDocExt = swModel.Extension
         'Open a sketch
         swSketchMgr.Insert3DSketch True
         swModel.ClearSelection2 True
         Set swSketchSeg(0) = swSketchMgr.CreateLine(StartX / 1000, StartY / 1000, StartZ /
1000, EndX / 1000, EndY / 1000, EndZ / 1000)
         swModel.ClearSelection2 True
         'Exit sketch
         swSketchMgr.Insert3DSketch False
      j = j + 3
      Loop While IsEmpty(ExcelApp.Worksheets("Electrode2_" & Format$(n)).Cells(i, j +
3).Value) = False
   i = i + 1
```

```
    Loop While IsEmpty(ExcelApp.Worksheets("Electrode2_" & Format$(n)).Range("A" & Format$(i + 1)).Value) = False
n = n + 1
Loop While IsEmpty(ExcelApp.Worksheets("Main_Electrode").Range("A" & Format$(n)).Value) = False
```

Generate main electrodes

```
Set SwApp = CreateObject("SldWorks.Application")
Set swModel = SwApp.ActiveDoc
Set swSelMgr = swModel.SelectionManager
Set swSketchMgr = swModel.SketchManager
Set swModeler = SwApp.GetModeler
Set swFeatMgr = swModel.FeatureManager
Set swPart = swModel
'Create the Excel application.
Set ExcelApp = GetObject(, "Excel.Application")
n = 1
'for each group
Do
    'Create new spreadsheets
    ExcelApp.ActiveWorkbook.Sheets.Add
    ExcelApp.ActiveSheet.Name = "Main1_" & Format$(n)
    ExcelApp.ActiveWorkbook.Sheets.Add
    ExcelApp.ActiveSheet.Name = "Main2_" & Format$(n)
    i = 1
    'for each row
    Do
        'For main1
        With ExcelApp.Worksheets("Main1_" & Format$(n))
            .Range("A" & Format$(i)).Value = ExcelApp.Worksheets("Electrode1_" & Format$(n)).Range("A" & Format$(i)).Value
            .Range("B" & Format$(i)).Value = ExcelApp.Worksheets("Electrode1_" & Format$(n)).Range("B" & Format$(i)).Value
            .Range("C" & Format$(i)).Value = ExcelApp.Worksheets("Electrode1_" & Format$(n)).Range("C" & Format$(i)).Value
        End With
        'For main2
        'Get the empty cell in the row
        b = 1
        m = 0
        Do
```

```
         If IsEmpty(ExcelApp.Worksheets("Electrode2_" & Format$(n)).Cells(i, b).Value) =
True Then
            m = 1
         End If
         b = b + 1
      Loop While m <> 1
      With ExcelApp.Worksheets("Main2_" & Format$(n))
         .Range("A" & Format$(i)).Value = ExcelApp.Worksheets("Electrode2_" &
Format$(n)).Cells(i, b - 4).Value
         .Range("B" & Format$(i)).Value = ExcelApp.Worksheets("Electrode2_" &
Format$(n)).Cells(i, b - 3).Value
         .Range("C" & Format$(i)).Value = ExcelApp.Worksheets("Electrode2_" &
Format$(n)).Cells(i, b - 2).Value
      End With
   i = i + 1
   Loop While IsEmpty(ExcelApp.Worksheets("Electrode1_" & Format$(n)).Range("A" &
Format$(i)).Value) = False

n = n + 1
Loop While IsEmpty(ExcelApp.Worksheets("Main_Electrode").Range("A" &
Format$(n)).Value) = False
```

nScrypt command generation: main lines

```
'Create the Excel application.
Set ExcelApp = GetObject(, "Excel.Application")
'Create the Word application.
Set WordApp = GetObject(, "Word.Application")
WordApp.Visible = True
n = 1
'for each group
Do
sheet = "Main1_" & Format$(n)
DistX = ExcelApp.Worksheets(sheet).Range("A1").Value
DistY = ExcelApp.Worksheets(sheet).Range("B1").Value
DistZ = ExcelApp.Worksheets(sheet).Range("C1").Value
With WordApp.Selection
   .TypeText Text:=Format(DistX, "###0.00") & " " & Format(-DistZ, "###0.00") & " " &
Format(DistY, "###0.00")
   .TypeParagraph
   .TypeText Text:="0 0 -2"
   .TypeParagraph
   .TypeText Text:="PRINT"
```

```
   .TypeParagraph
End With
i = 2
Do
   StartX = ExcelApp.Worksheets(sheet).Range("A" & Format$(i - 1)).Value
   StartY = ExcelApp.Worksheets(sheet).Range("B" & Format$(i - 1)).Value
   StartZ = ExcelApp.Worksheets(sheet).Range("C" & Format$(i - 1)).Value
   EndX = ExcelApp.Worksheets(sheet).Range("A" & Format$(i)).Value
   EndY = ExcelApp.Worksheets(sheet).Range("B" & Format$(i)).Value
   EndZ = ExcelApp.Worksheets(sheet).Range("C" & Format$(i)).Value
   DistX = EndX - StartX
   DistY = EndY - StartY
   DistZ = EndZ - StartZ
   WordApp.Selection.TypeText Text:=Format(DistX, "###0.00") & " " & Format(-DistZ,
"###0.00") & " " & Format(DistY, "###0.00")
   WordApp.Selection.TypeParagraph
i = i + 1
Loop While IsEmpty(ExcelApp.Worksheets(sheet).Range("A" & Format$(i)).Value) = False
WordApp.Selection.TypeText Text:="STOP PRINT"
WordApp.Selection.TypeParagraph
WordApp.Selection.TypeText Text:="0 0 2"
WordApp.Selection.TypeParagraph
'move back to origin
WordApp.Selection.TypeText Text:=Format(-EndX, "###0.00") & " " & Format(EndZ,
"###0.00") & " " & Format(-EndY, "###0.00")
WordApp.Selection.TypeParagraph
n = n + 1
Loop While IsEmpty(ExcelApp.Worksheets("Main_Electrode").Range("A" &
Format$(n)).Value) = False
n = 1
'for each group
Do
sheet = "Main2_" & Format$(n)
DistX = ExcelApp.Worksheets(sheet).Range("A1").Value
DistY = ExcelApp.Worksheets(sheet).Range("B1").Value
DistZ = ExcelApp.Worksheets(sheet).Range("C1").Value
With WordApp.Selection
   .TypeText Text:=Format(DistX, "###0.00") & " " & Format(-DistZ, "###0.00") & " " &
Format(DistY, "###0.00")
   .TypeParagraph
   .TypeText Text:="0 0 -2"
   .TypeParagraph
   .TypeText Text:="PRINT"
   .TypeParagraph
```

```
End With
i = 2
Do
    StartX = ExcelApp.Worksheets(sheet).Range("A" & Format$(i - 1)).Value
    StartY = ExcelApp.Worksheets(sheet).Range("B" & Format$(i - 1)).Value
    StartZ = ExcelApp.Worksheets(sheet).Range("C" & Format$(i - 1)).Value
    EndX = ExcelApp.Worksheets(sheet).Range("A" & Format$(i)).Value
    EndY = ExcelApp.Worksheets(sheet).Range("B" & Format$(i)).Value
    EndZ = ExcelApp.Worksheets(sheet).Range("C" & Format$(i)).Value
    DistX = EndX - StartX
    DistY = EndY - StartY
    DistZ = EndZ - StartZ

    WordApp.Selection.TypeText Text:=Format(DistX, "###0.00") & " " & Format(-DistZ,
"###0.00") & " " & Format(DistY, "###0.00")
    WordApp.Selection.TypeParagraph
i = i + 1
Loop While IsEmpty(ExcelApp.Worksheets(sheet).Range("A" & Format$(i)).Value) = False
WordApp.Selection.TypeText Text:="STOP PRINT"
WordApp.Selection.TypeParagraph
WordApp.Selection.TypeText Text:="0 0 2"
WordApp.Selection.TypeParagraph
'move back to origin
WordApp.Selection.TypeText Text:=Format(-EndX, "###0.00") & " " & Format(EndZ,
"###0.00") & " " & Format(-EndY, "###0.00")
WordApp.Selection.TypeParagraph
n = n + 1
Loop While IsEmpty(ExcelApp.Worksheets("Main_Electrode").Range("A" &
Format$(n)).Value) = False
```

nScrypt command generation: electrode 1

```
'Create the Excel application.
Set ExcelApp = GetObject(, "Excel.Application")
'Create the Word application.
Set WordApp = GetObject(, "Word.Application")
WordApp.Visible = True
ExcelApp.Worksheets("Electrode1_1").Select
EndX = 0
EndY = 0
EndZ = 0
i = 1
Do
```

```
    j = 1
    Do
    j = j + 3
    Loop While IsEmpty(ExcelApp.Worksheets("Electrode1_1").Cells(i, j).Value) = False
    'distance to move from the origin
    DistX = ExcelApp.Cells(i, j - 3).Value - EndX
    DistY = ExcelApp.Cells(i, j - 2).Value - EndY
    DistZ = ExcelApp.Cells(i, j - 1).Value - EndZ
    'for 1st line
    With WordApp.Selection
        .TypeText Text:=Format(DistX, "###0.00") & " " & Format(-DistZ, "###0.00") & " " &
Format(DistY, "###0.00")
        .TypeParagraph
        .TypeText Text:="0 0 -2"
        .TypeParagraph
        .TypeText Text:="PRINT"
        .TypeParagraph
    End With
    j = j - 3
    'for the next lines
    For k = j To 4 Step -3
        StartX = ExcelApp.Worksheets("Electrode1_1").Cells(i, k).Value
        StartY = ExcelApp.Worksheets("Electrode1_1").Cells(i, k + 1).Value
        StartZ = ExcelApp.Worksheets("Electrode1_1").Cells(i, k + 2).Value
        EndX = ExcelApp.Worksheets("Electrode1_1").Cells(i, k - 3).Value
        EndY = ExcelApp.Worksheets("Electrode1_1").Cells(i, k - 2).Value
        EndZ = ExcelApp.Worksheets("Electrode1_1").Cells(i, k - 1).Value
        DistX = EndX - StartX
        DistY = EndY - StartY
        DistZ = EndZ - StartZ
        With WordApp.Selection
            .TypeText Text:=Format(DistX, "###0.00") & " " & Format(-DistZ, "###0.00") & " "
& Format(DistY, "###0.00")
            .TypeParagraph
        End With
    Next k
    With WordApp.Selection
        .TypeText Text:="STOP PRINT"
        .TypeParagraph
        .TypeText Text:="0 0 2"
        .TypeParagraph
    End With
i = i + 1
```

```
Loop While IsEmpty(ExcelApp.Worksheets("Electrode1_1").Range("A" &
Format$(i)).Value) = False
'move back to origin
WordApp.Selection.TypeText Text:=Format(-EndX, "###0.00") & " " & Format(EndZ,
"###0.00") & " " & Format(-EndY, "###0.00")
WordApp.Selection.TypeParagraph
```

nScrypt command generation: electrode 2

```
'Create the Excel application.
Set ExcelApp = GetObject(, "Excel.Application")
'Create the Word application.
Set WordApp = GetObject(, "Word.Application")
WordApp.Visible = True
ExcelApp.Worksheets("Electrode2_1").Select
EndX = 0
EndY = 0
EndZ = 0
i = 1
Do
    j = 1
    'distance to move from the origin
    DistX = ExcelApp.Cells(i, j).Value - EndX
    DistY = ExcelApp.Cells(i, j + 1).Value - EndY
    DistZ = ExcelApp.Cells(i, j + 2).Value - EndZ
    'for 1st line
    With WordApp.Selection
        .TypeText Text:=Format(DistX, "###0.00") & " " & Format(-DistZ, "###0.00") & " " &
Format(DistY, "###0.00")
        .TypeParagraph
        .TypeText Text:="0 0 -2"
        .TypeParagraph
        .TypeText Text:="PRINT"
        .TypeParagraph
    End With
    j = j + 3
    'for the next lines
    Do
        StartX = ExcelApp.Worksheets("Electrode2_1").Cells(i, j - 3).Value
        StartY = ExcelApp.Worksheets("Electrode2_1").Cells(i, j - 2).Value
        StartZ = ExcelApp.Worksheets("Electrode2_1").Cells(i, j - 1).Value
        EndX = ExcelApp.Worksheets("Electrode2_1").Cells(i, j).Value
        EndY = ExcelApp.Worksheets("Electrode2_1").Cells(i, j + 1).Value
```

```
    EndZ = ExcelApp.Worksheets("Electrode2_1").Cells(i, j + 2).Value
    DistX = EndX - StartX
    DistY = EndY - StartY
    DistZ = EndZ - StartZ
    With WordApp.Selection
      .TypeText Text:=Format(DistX, "###0.00") & " " & Format(-DistZ, "###0.00") & " "
& Format(DistY, "###0.00")
      .TypeParagraph
    End With
    j = j + 3
  Loop While IsEmpty(ExcelApp.Worksheets("Electrode2_1").Cells(i, j)) = False
  With WordApp.Selection
    .TypeText Text:="STOP PRINT"
    .TypeParagraph
    .TypeText Text:="0 0 2"
    .TypeParagraph
  End With
i = i + 1
Loop While IsEmpty(ExcelApp.Worksheets("Electrode2_1").Range("A" &
Format$(i)).Value) = False
'move back to origin
WordApp.Selection.TypeText Text:=Format(-EndX, "###0.00") & " " & Format(EndZ,
"###0.00") & " " & Format(-EndY, "###0.00")
WordApp.Selection.TypeParagraph
```