Finite Element Modeling of Arthropod Exoskeleton

by

William Clay Flannigan

Submitted in partial fulfillment of the requirements for the degree of
Master of Science

Thesis Advisor: Dr. Roger D. Quinn

Department of Mechanical and Aerospace Engineering
Case Western Reserve University
May 1998

Finite Element Modeling of Arthropod Exoskeleton

Abstract
by
William Clay Flannigan

Confocal microscopy methods have been employed to image the metathoracic trochanter, a small exoskeleton segment between the coxa and femur, of an adult cockroach, *Periplaneta americana*. The confocal images represent planar sections through the exoskeleton of the trochanter. Specialized software was implemented to convert the section images into a complete three-dimensional geometric model of the trochanter.

The raster image data of the confocal model was exported in a vector file format. Custom software was developed to resample the vector data and to automatically create a finite element mesh. The resulting mesh contained first-order, three-dimensional, triangular plate elements and also encoded data for element thicknesses.

The Algor finite element package was used to analyze the model under the conditions of several biologically pertinent experiments. Special emphasis was placed on the strain analysis of the four groups of campaniform sensilla on the trochanter. These sensory organs respond to strains in the exoskeleton and are instrumental in the control of natural walking in insects.

The results were compared to data from biological experiments which confirmed hypotheses and answered questions that have been difficult to determine experimentally on the animal. Functional differentiation both between groups and within groups of sensilla was observed. Further evidence that strain measurements are used in positive feedback control of the trochanteral extensor muscles was discovered. The structure of the trochanter was analyzed in the context of the function of the campaniform sensilla. Previously undescribed quantitative stress and strain data for insect cuticle was recorded.

This work is dedicated to my parents for being there when I needed you, for giving me space when I needed it, and for encouraging me the whole way.

Acknowledgments

I would first like to thank Dr. Mansour and Dr. Ritzmann for participating on my committee.

My graduate studies at CWRU have been an enjoyable experience, much of which can be attributed to my co-workers in the research group. Thanks Gabe Nelson, Rich Bachmann, and Matt Birch for the advice, encouragement, and diversions. Also included in this list are Greg Causey and Nick Barendt and the CAISR Lab for answering all my little questions and providing access to their computing facilities.

None of this work could have been possible without the tireless efforts of the Zill Laboratory of Marshall University School of Medicine. Dr. Zill and his research associates have not only provided the foundation for the modeling in this thesis, but also invaluable experience and advice. Thank you Dr. Zill for your continuous excitement for this research.

I have been fortunate to have an advisor who has been so involved in my endeavors—especially my backswing. Dr Quinn has been a great friend and has made my graduate work seem much less like work. I only hope I have reciprocated in some way, maybe by lowering his handicap.

Finally, this research has been graciously supported by the Timken Company and by the Office of Naval Research grant numbers N00014-96-1-0694 and N00014-96-10708.

Contents

Abstract ii

Acknowledgments v

List of Figures ix

List of Tables xi

Chapter I

Introduction 1
1.1 Purpose 1
1.2 Perspective 3
1.3 Group History 4

Chapter II

Biological Background 11
2.1 Cockroach Anatomy 11
2.2 Mechanoreceptors 14
2.2.1 Campaniform Sensilla Morphology 16
2.2.2 Campaniform Sensilla Physiology 18
2.3 Material Properties of Insect Exoskeleton 20
2.3.1 Material Parameters for Finite Element Analysis 23
2.4 Confocal Microscopy Model 26
2.4.1 Confocal Sectioning Method 27
2.4.2 Model Controls 30

Chapter III

Computer Modeling 32
3.1 The Finite Element Method 32
3.1.1 Finite Element Theory 32
3.1.2 Finite Element Application 37
3.2 The Algor FEA System 38
3.3 Confocal Microscopy Data 41

3.4 Mesh Generation . . . 43
3.4.1 Meshing Method . . . 43
3.4.2 Code Development . . . 45
3.4.3 Examples . . . 48
3.4.4 The Trochanter Model . . . 50

Chapter IV

Results . . . 54
4.1 Convergence Study . . . 54
4.2 Material Studies . . . 56
4.3 Campaniform Sensilla Location and Orientation . . . 58
4.4 Peg Leg Experiments . . . 60
4.4.1 Campaniform Sensilla Group Contribution . . . 61
4.4.2 Biological Experiments . . . 62
4.4.2.1 Forced Flexion . . . 64
4.4.3 Finite Element Model Experiments . . . 65
4.4.3.1 Forced Flexion . . . 66
4.4.3.2 Forced Extension . . . 75
4.4.3.3 Extensor Muscles . . . 76
4.4.3.4 Flexor Muscles . . . 80
4.4.3.5 Muscle Lines of Action . . . 84
4.4.3.6 Summary . . . 85
4.5 Smooth Model . . . 87

Chapter V

Conclusions . . . 90
5.1 Methods . . . 90
5.2 Discussion . . . 92
5.3 Future Work . . . 94

References . . . 95

Appendix A

Mesh Generator Code . . . 102
A.1 File Poly5.h . . . 102
A.2 File Poly5.c . . . 103

List of Figures

Figure 1.1 Robot I 5
Figure 1.2 Robot II 6
Figure 1.3 Robot III 7
Figure 1.4 K2T Demining Robot 8
Figure 2.1 Cockroach Leg Segments 13
Figure 2.2 Trochanter Photomicrograph 13
Figure 2.3 Trochanteral Muscle Definition 14
Figure 2.4 SEM of the Anterior Half of a Trochanter 17
Figure 2.5 Campaniform Sensilla Morphology 18
Figure 2.6 Anterior Half of Rendered Confocal Model, Interior View 29
Figure 2.7 Posterior Half of Rendered Confocal Model, Interior View 30
Figure 3.1 Truss Model 33
Figure 3.2 Three Element Truss Model 33
Figure 3.3 Sample Polygon File 41
Figure 3.4 Isoparametric Element Mapping 42
Figure 3.5 Sampling Vertex Data 45
Figure 3.6 Cylinder Test Model 49
Figure 3.7 Sphere Test Model 50
Figure 3.8 Trochanter Model, Anterior 51
Figure 3.9 Trochanter Model, Posterior 51
Figure 3.10 Finite Element Mesh Over Confocal Model 52
Figure 3.11 Finite Element Mesh Over Original Trochanter 53
Figure 4.1 Poisson’s Ratio Analysis 58
Figure 4.2 Peg Leg Forced Flexion Experiment 63
Figure 4.3 Peg Leg Flex ENG Response 65
Figure 4.4 Peg Leg Flex, Anterior, Von Mises Strain 67
Figure 4.5 Peg Leg Flex, Posterior, Von Mises Strain 67
Figure 4.6 Peg Leg Flex, Short-Axis Strain 70
Figure 4.7 Group 4 Sensilla Showing Single Separate Sensilla (L1) 73
Figure 4.8 Mohr’s Circle for Group 4 74
Figure 4.9 Peg Leg Extend, Short-Axis Strain 76
Figure 4.10 Peg Leg 177, Anterior, Von Mises Strain 78
Figure 4.11 Peg Leg 177, Posterior, Von Mises Strain 79
Figure 4.12 Peg Leg 177 Short-Axis Strain 80

Figure 4.13 Peg Leg 181, Anterior, Von Mises Strain 81
Figure 4.14 Peg Leg 181, Posterior, Von Mises Strain 82
Figure 4.15 Peg Leg 181, Short-Axis Strain 83
Figure 4.16 Smooth Model, Peg Leg Flex, Anterior, Von Mises Strain 88

List of Tables

Table 2.1 Cuticle Properties 22
Table 4.1 Campaniform Sensilla Orientation 59
Table 4.2 Strain Values, Peg Leg Flex 71
Table 4.3 Strain for Various Muscle Lines of Action 84
Table 4.4 Summary of Group Responses 85

Chapter I

Introduction

1.1 Purpose

Biological organisms have been evolving for millions of years to overcome the challenges of life in an uncertain and changing environment. Insects are unique in the animal kingdom in their number and diversity. They have adapted to almost every environmental niche and many are extremely adept locomotors. This observation has led many investigators to study the behavior and physiology of insects and arthropods in general.

The American Cockroach, *Periplaneta americana,* is one the champions in locomotion ability, being able to traverse walls, ceilings, and rough terrain at up to 24 steps per second [Delcomyn, 1971]. Its walking behavior and associated neural circuits are some of the most widely investigated in the invertebrate world, yet many questions remained unanswered as to how individual influences from sense organs affect leg coordination.

Answering these questions will lead to a better understanding of arthropod locomotion and nervous systems in general, but from an engineer's viewpoint there are other benefits to these investigations. Roboticists have recently begun looking towards biological research to gain insight for the design and control of mobile robots. Insects such as cockroaches have remarkable locomotion capabilities when compared with the most advanced walking robots. By extracting control and biomechanical principles from insects and applying them to solve locomotion problems, more agile walking robots are being designed to traverse hazardous and even extraterrestrial environments.

This work has goals founded in both of these fields—to further our understanding of insect physiology and to advance the development of walking vehicles. The sensory capabilities of insects are relatively coarse compared to typical robotic sensors, yet insects are capable of incredible navigation feats. This suggests that insects' means of incorporating relatively simple proprioceptive and exteroceptive (internal and external sensory) information into their behavior are successful and warrant investigation by biologists and roboticist alike.

The trochanteral campaniform sensilla are only a small part of an insect's vast sensory array, but it has been shown that campaniform sensilla play a major role in the motor activity of insects [Pearson, 1972; Zill and Moran, 1981b]. Additionally, in the trochanter, the largest muscles in the leg have their insertions [Alsop, 1978], and four groups of campaniform sensilla are present [Pringle, 1938b]. Because of this, researchers believe this to be a high stress area with strong influences on locomotion. The complex geometry and loading conditions of the trochanter have thus far prevented a comprehensive study of the trochanteral campaniform sensilla [Zill, 1990]. The use of computer modeling and finite element analysis in this work have proven to be a powerful method for describing the physiology of the trochanteral campaniform sensilla.

1.2 Perspective

This work was performed in R. Quinn's Biorobotics Lab in the Department of Mechanical and Aerospace Engineering. The larger Biologically Inspired Research Program at CWRU also involves R. Beer's Lab in the Department of Computer Science, and R. Ritzmann's and H. Chiel's Labs in the Department of Biology and Neuroscience. The group has also had the opportunity to work closely with S. Zill's Laboratory in the Department of Anatomy, Cell, and Neurobiology, at the Marshall University School of Medicine. One of the ongoing goals of this project and the primary goal of the Biorobotics Lab is to build better robots through understanding and application of biological principles.

This goal encompasses a variety of research in many fields. Basic scientific research is performed in the Ritzmann Lab at CWRU and the Zill Lab at Marshall University. This work includes behavioral and physiological studies on a variety of animals including the *Periplaneta americana.* The engineering aspects include the use of computer modeling and simulation of animals and robots. Using these tools, robot locomotion controllers have been developed and tested. Ultimately, the controllers are applied to biologically-based legged robots. The work described in this thesis falls in the transition between the biological research and the engineering design and analysis.

An attempt has been made to keep the discussions in this thesis at a level where both biologists and engineers with only a minimal background in the converse field can follow the reasoning. Hopefully, this has been accomplished without eliminating important details or creating a document that is less useful to either fields.

1.3 Group History

At CWRU, the biologically inspired robotics work began with the Artificial Insect Project [Beer, 1990] in which a simulated hexapod was controlled using a neural network. The controller was based on experimental studies of the cockroach nervous system [Pearson et al., 1973]. It had the ability to wander, follow edges, look for food, and eat. The locomotion controller was particularly interesting and its success led to the formation of the Biologically Inspired Robotics Group and the Biorobotics Lab.

The Biorobotics Lab's first robot, Robot I, shown in figure 1.1, was a hexapod with two degrees of freedom (DOF) per leg. It was capable of flat terrain, straight line motion in a continuum of insect-like gaits with the implementation of the neural network controller developed in the Artificial Insect Project [Quinn and Espenschied, 1993; Beer et al., 1992]. Studies showed that the controller was robust to lesions [Chiel et al., 1992].

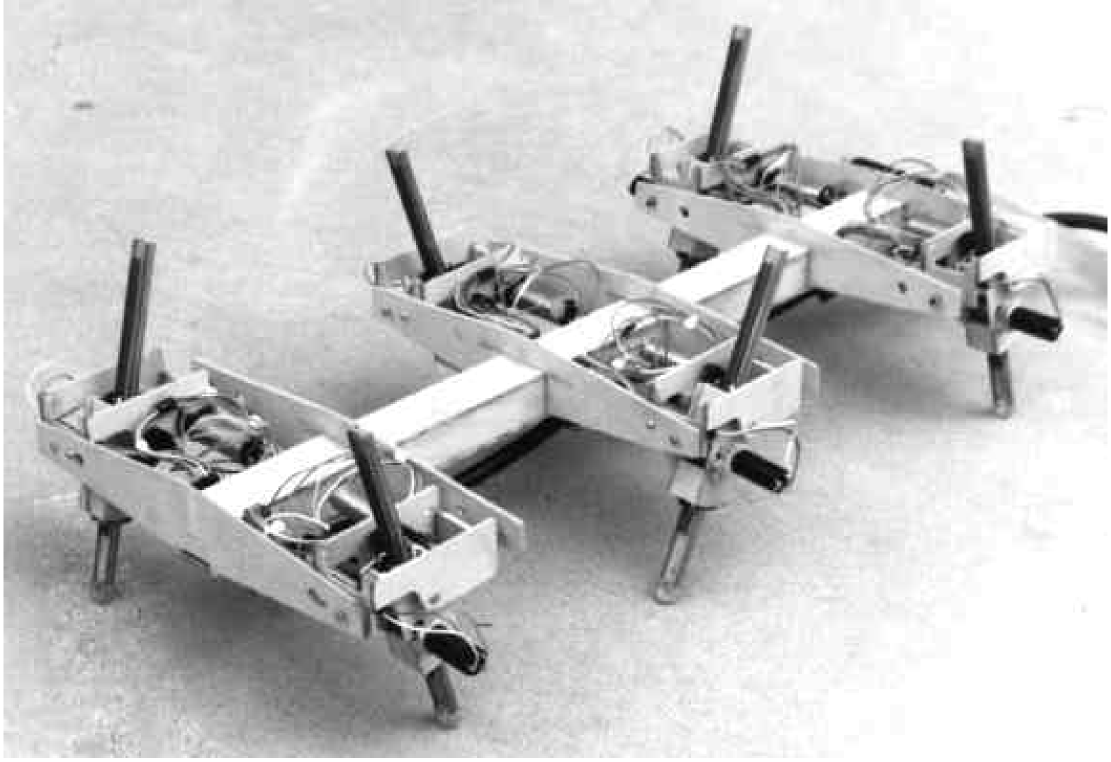

Figure 1.1 Robot I

The lab's next hexapod, Robot II, figure 1.2, addressed several of the short comings of Robot I. Each leg of the robot had three rotational DOF which permitted turning and rough terrain locomotion. The gait controller for this robot [Espenschied et al., 1993] was not a neural network, but instead was based on inter-leg influences that are thought to be responsible for the leg coordination of the stick insect [Cruse, 1990]. In addition, localized leg reflexes were added that permitted navigation over uneven, unstable, and partially missing terrain [Espenschied et al., 1996].

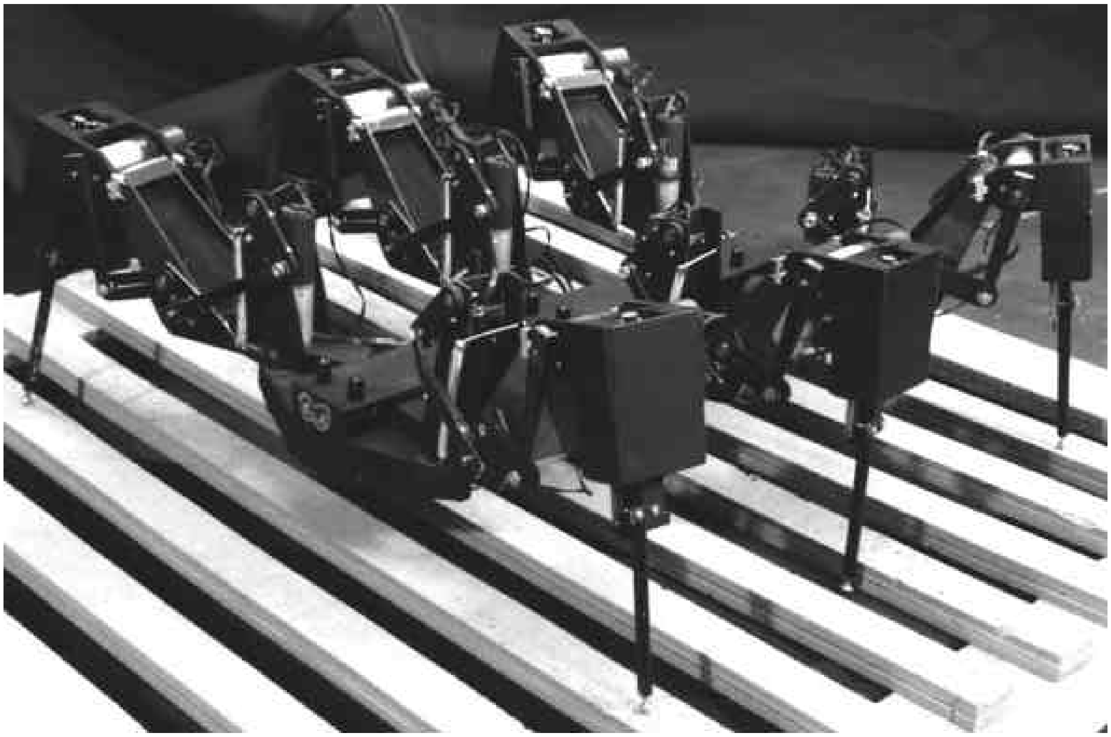

Figure 1.2 Robot II

While Robot II displayed exceptional rough terrain performance, it suffered from the curse of most electric motor actuated mobile robots: slowness. One answer to this problem is the current hexapod robot in development, Robot III, shown in figure 1.3. It is believed that by tieing the mechanical design and control strategy very closely to that of the animal, great advances in performance can be realized. Robot III's mechanical design is based directly on data gathered in the Ritzmann Lab on the Death-Head Cockroach, *Blaberous discoidalis* [Watson and Ritzmann, 1998a; 1998b]. Muscle-like pneumatic actuators are used to actuate the 24 DOF hexapod [Nelson et al., 1997]. Initial posture control studies have shown great promise for this approach [Nelson and Quinn, 1998].

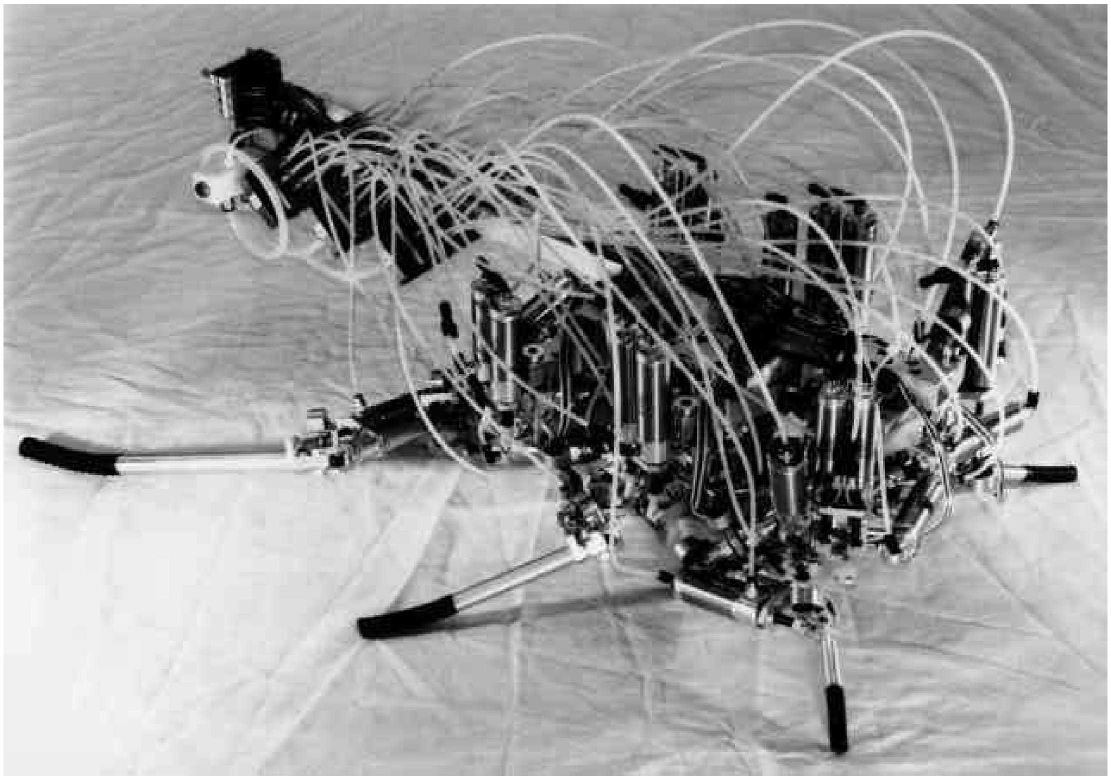

Figure 1.3 Robot III

The Biorobotics Lab has also had the opportunity to work with K^2T inc. of Duquesne, PA on legged robots. K^2T developed an autonomous crab-like robot, figure 1.4, for the purpose of removing and disposing military mines and unexploded munitions [Flannigan et al., 1998]. The robot has eight two DOF legs which are attached to two vertically mounted frames. The robot turns by rotating the two frames with respect to each other. It’s biologically-based locomotion controller was developed at CWRU using simulation tools and is based partly on Robot II's locomotion controller.

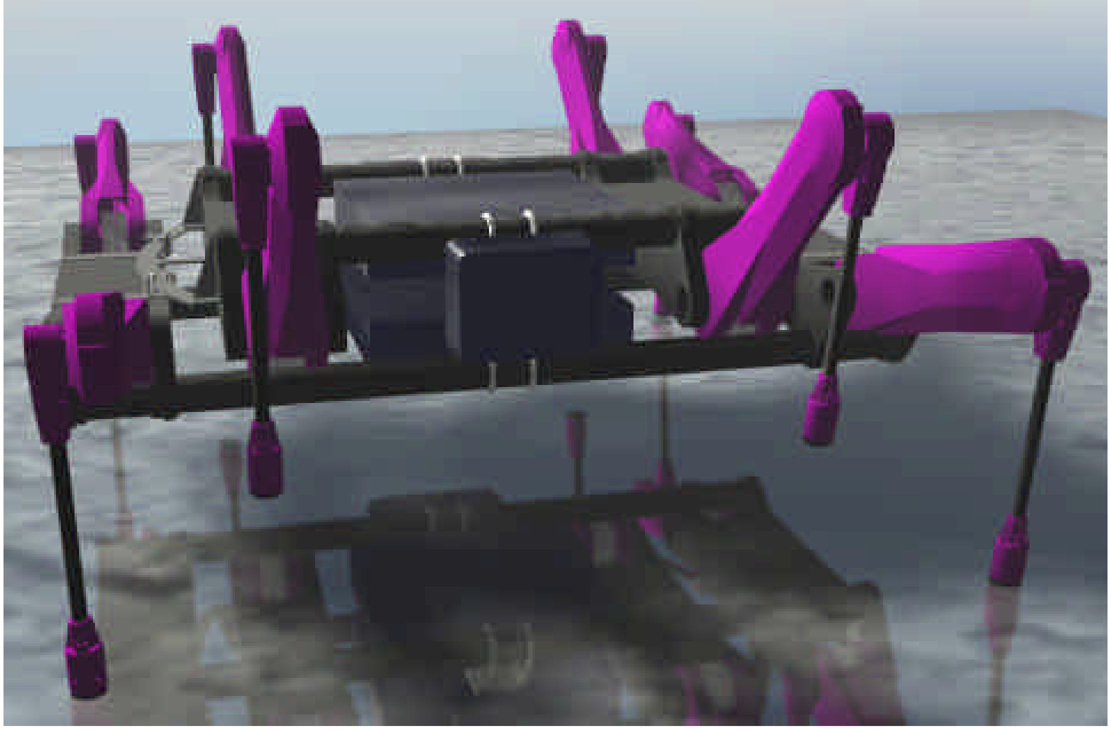

Figure 1.4 K^{2}T Demining Robot

The Ritzmann Laboratory in the Biology Department of CWRU has worked closely with the Biorobotics Laboratory from its inception. Dr. Ritzmann's research interests include neural integration of sensory data in the control of insect locomotion. He has extensively studied the neural organization of the cockroach escape response [Ritzmann, 1993]. This system is an excellent example of how arthropod nervous systems integrate complex sensory data and incorporate that information into locomotive behavior.

In addition, the Ritzmann Laboratory has performed studies on neurological control of insect walking, and running [Watson and Ritzmann, 1998a; 1998b]. Through the use of high speed video to extract joint kinematic data and concurrent electrophysiological recordings, they have been able to gain an understanding of how neural signals direct muscle contractions and movement during real locomotion. This data has been invaluable in the mechanical design and controller development for Robot III.

The Biologically Inspired Robotics Group has also had the pleasure of working with the Zill Laboratory in the Department of Anatomy, Cell and Neurobiology, at the Marshall University School of Medicine. Dr. Zill has been at the forefront of arthropod mechanoreceptor research for the past two decades. His interests include campaniform sensilla research which is highlighted by the influential work on tibial campaniform sensilla and their effect on cockroach walking [Zill and Moran, 1981a; Zill et al., 1981; Zill and Moran, 1981b]. Zill has experience with most of the sense organs found in insect legs and the neural circuitry innervating them. He has performed experiments and documented mechanoreceptor's effects on reflexes [Zill, 1990], posture [Zill 1982], and walking [Zill and Seyfarth, 1996] of arthropods.

The Zill Laboratory has been involved in all of the biological studies for this work. This includes the confocal imaging and subsequent model constructions. In addition, many anatomical studies were performed to gather data for model details and to confirm the model's accuracy. The Zill Laboratory also performed several physiological experiments [DiCaprio et al., 1997] which are used for direct correlation between these results and the insect.

Chapter II

Biological Background

2.1 Cockroach Anatomy

The insect has three distinct body regions—head, thorax, and abdomen. It is postulated that all the segments once had walking appendages, but after years of evolution, there only remain six walking legs on the thorax [Alsop, 1978]. These legs are designated by the three segments of the thorax—pro (front), meso (middle), and meta (rear).

The three pairs of insect legs may be similar in form and function as in the beetle [Delcomyn et al., 1996] or they may be specialized as is the case with the cockroach. The prothoracic legs of the cockroach are small and highly mobile and are used primarily as sensing appendages to find foot holds and monitor the environment. The mesothoracic legs support much of the body weight and are important for controlling the pitch of the body for climbing. The metathoracic legs are the largest and have the least flexibility of the three pairs. They provide much of the forward propulsive force for walking and running [Snodgrass, 1965; Full et al., 1991]. This study concentrates on the metathoracic leg.

The cockroach leg, shown in figure 2.1, has five major divisions. Working from the thorax distally they are: the coxa, the trochanter, the femur, the tibia, and the tarsus. Most of the joint articulations are about two hinge joints (condyles). The coxa attaches to the trochanter with one condyle on each side (anterior and posterior) of the trochanter, slightly askew with the plane of the coxa [Snodgrass, 1965], figure 2.2. The femoral-trochanteral joint is often considered fused [Delcomyn et al., 1996], but in fact, small movements due occur along a hinge-like joint supported by two condyles on the anterior side of the trochanter [Alsop, 1978; Snodgrass, 1965]. This joint is also the location of autotomy where the animal will separate its leg if trapped or damaged [Pringle, 1938b]. It is interesting to note that the line of action of the trochanteral-femoral joint is oblique to that of the coxal-trochanteral joint which allows the large moments created by the muscles in the coxa to be transmitted to the femur by using the strength of the exoskeleton to transmit the force rather than musculature. The femoral-tibial joint is a one DOF hinge joint, and the tibial-tarsal joint is flexible in three rotations [Full and Ahn, 1995]. The tarsus itself has flexibility between its segments due to elastic members. It's segments also have one active DOF that enables the claw to be retracted.

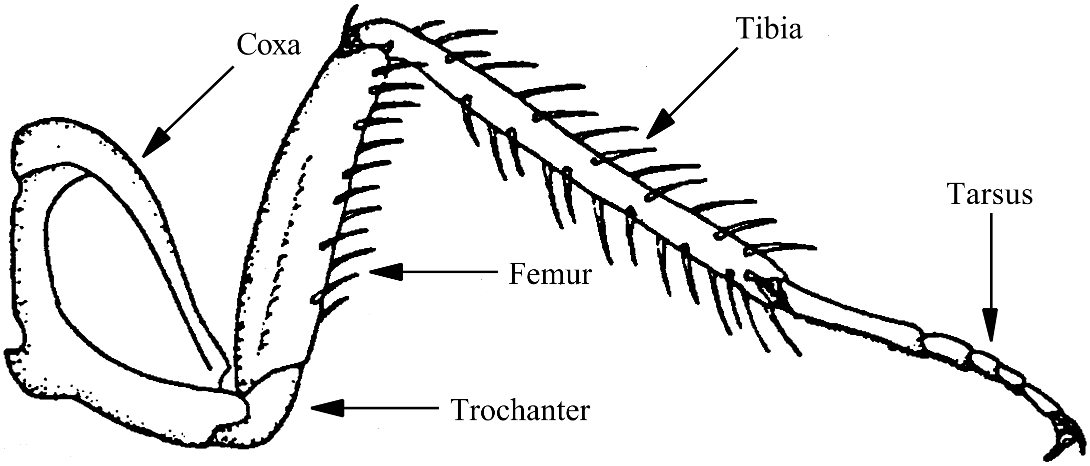

Figure 2.1 Cockroach Leg Segments

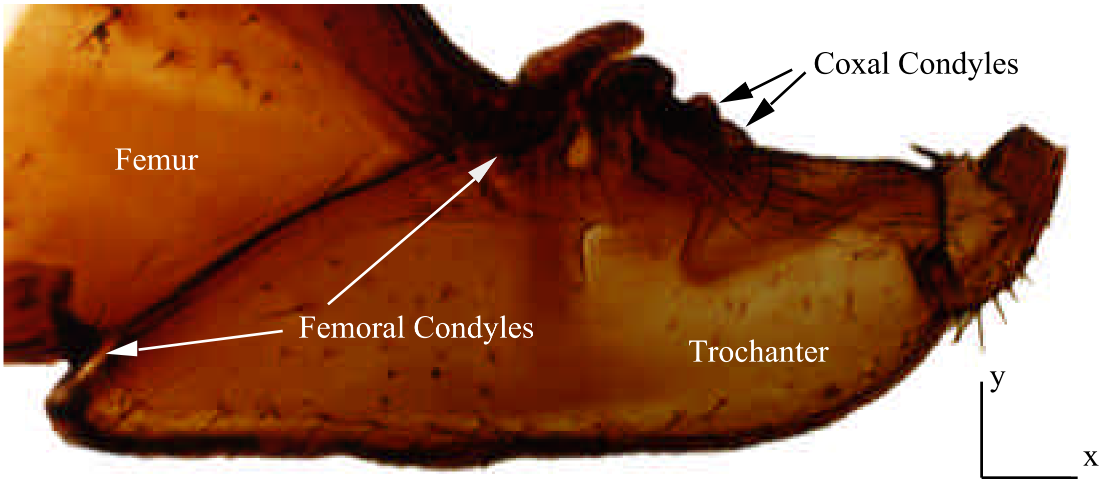

Figure 2.2 Trochanter Photomicrograph

Biologists are in some disagreement on the musculature of the trochanter in terms of its functional differentiation. Alsop [1978] chose a narrow morphological view and identified 26 different muscles inserting on the metathoracic trochanter. For this work, the description by Snodgrass [1965] will be used and is shown in his drawing in figure 2.3. The six numbered elements are actually the muscle apodemes (tendons) that the muscle groups insert onto. In fact, these apodemes do not insert directly on the exoskeleton but instead on elastic membranes that distribute the forces to the exoskeleton [Snodgrass, 1965]. Muscle groups 177, 178, and 179 are the trochanteral extensors, the large muscles of the coxa that support the body and drive it forward. Muscles 180, 181, and 182 are the trochanteral flexors which lift the leg. There is also one muscle, the reductor femoris, that originates in the trochanter and inserts on the posterior side of the femur which accounts for the small movements of the trochanteral-femoral joint [Snodgrass, 1965].

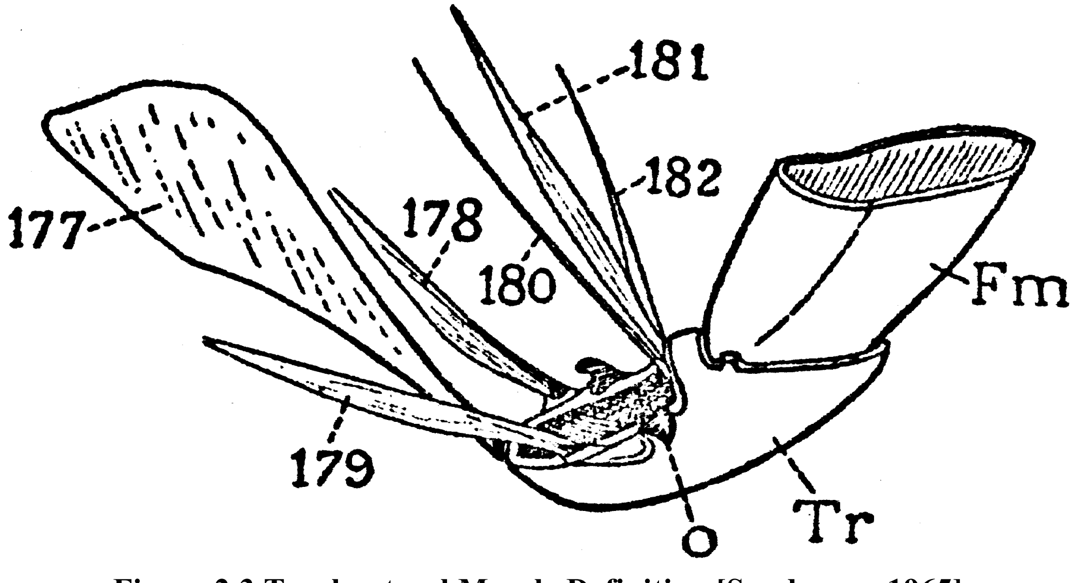

Figure 2.3 Trochanteral Muscle Definition [Snodgrass, 1965]

2.2 Mechanoreceptors

An insect requires the ability to physically sense its surroundings to locomote, find food, and escape from dangers. These sensory activities require the measurement of force which is provided by a broad class of sensory neurons know as mechanoreceptors. Mechanoreceptors come in many different forms with various means of stimulation such as touch, muscle tension, and joint angle [Zill, 1990]. These sensors have been further differentiated by the source of the stimulation; Those that respond to forces from the environment are exteroceptors and those that are stimulated by forces generated by the animal are proprioceptors [Sherrington, 1906]. It will be seen that some sensory organs such as campaniform sensilla can fall into both of these classes.

In the legs of cockroaches, there are five major groups of mechanoreceptors: isolated hair sensilla, hair plates, chordotonal organs, multipolar receptors, and campaniform sensilla. Isolated hair sensilla are single hairs that are found over most of the cockroach's legs. They are innervated (connected by nerves) by a single neuron and have a phasic response; that is, they detect changes in position of the hair not actual position. They are, therefore, probably used as transitional tactile sensors. Hair plates are similar to hair sensilla in that they are groups of hairs, but they are located near joints and are stimulated by joint movement. This classifies hair plates as proprioceptors as opposed to hair sensilla. Hair plates also show a tonic response to joint position meaning that they measure joint positions not velocities. Chordotonal organs are also proprioceptors that measure joint angle position and rate of change. They span joints and encode signals by means of a structure called a scolopale. The multipolar receptors are less well understood, but are believed to measure joint angle tonically (actual position) and joint velocity phasically [Zill, 1990].

2.2.1 Campaniform Sensilla Morphology

The campaniform sensilla are a type of mechanoreceptor that respond to strains in the exoskeleton, first described by Pringle [1938a] and elegantly demonstrated by Spinola and Chapman [1975]. On the cockroach leg, there are six groups of campaniform sensilla. Four are located on the trochanter (three on the anterior, one on the posterior), see figure 2.4; 2.6; 2.7, one on the proximal femur and one on the proximal tibia. A comparable arrangement of sensilla is found on a variety of insects. The sensilla are comprised of a thin cuticular cap in the exoskeleton which is innervated by a single neuron, figure 2.5. The dendrite of the neuron is connected to the cap which, when deflected by cuticular distortions, stimulates the neuron [Moran et al. 1971]. The caps are ellipsoidal and are from 6 to 24 μm in length. They have been shown in the tibial group to be responsive to compressive strains in the direction perpendicular to the long-axis of the sensilla (short-axis strains) [Zill and Moran, 1981a]. Within the groups of sensilla, the long-axis of the cuticular caps are oriented in approximately the same direction which further emphasizes their directional sensitivity.

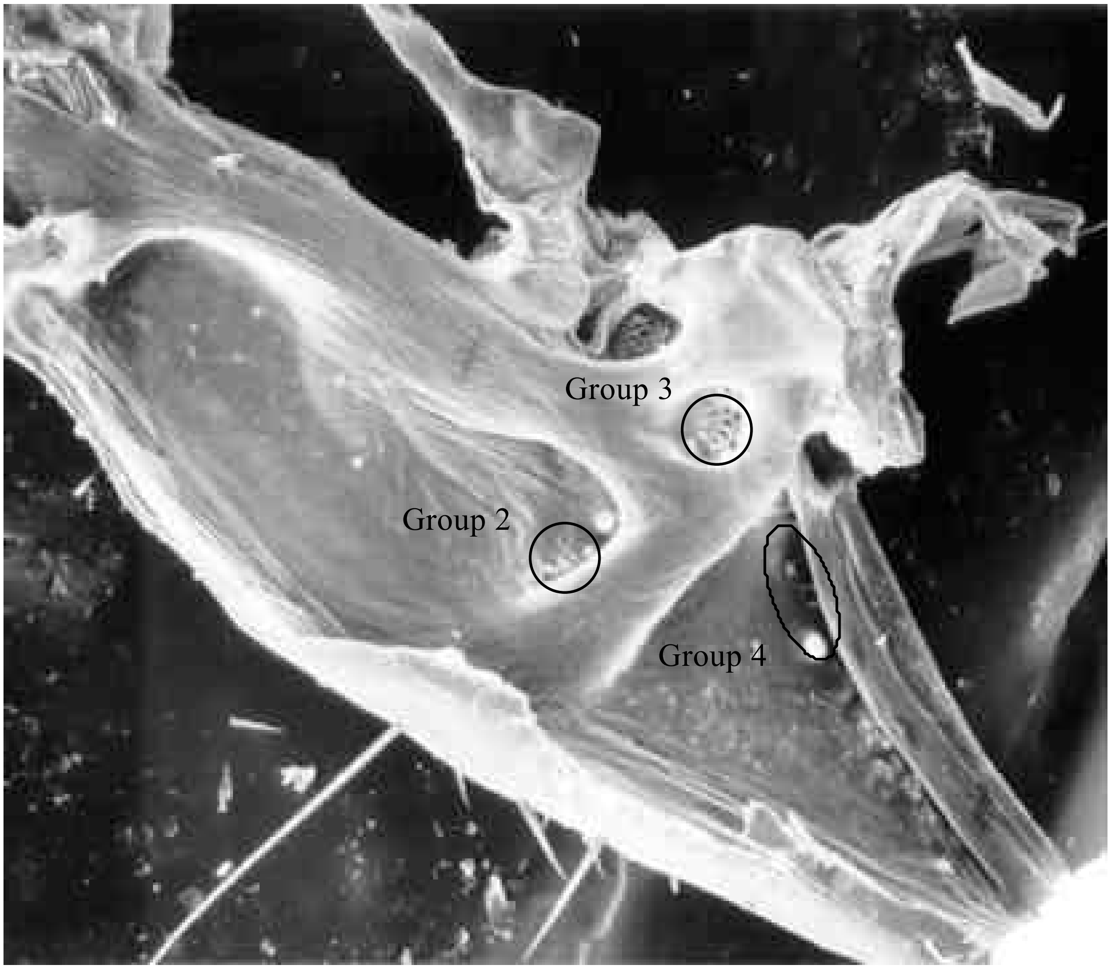

Figure 2.4 SEM of the Anterior Half of a Trochanter

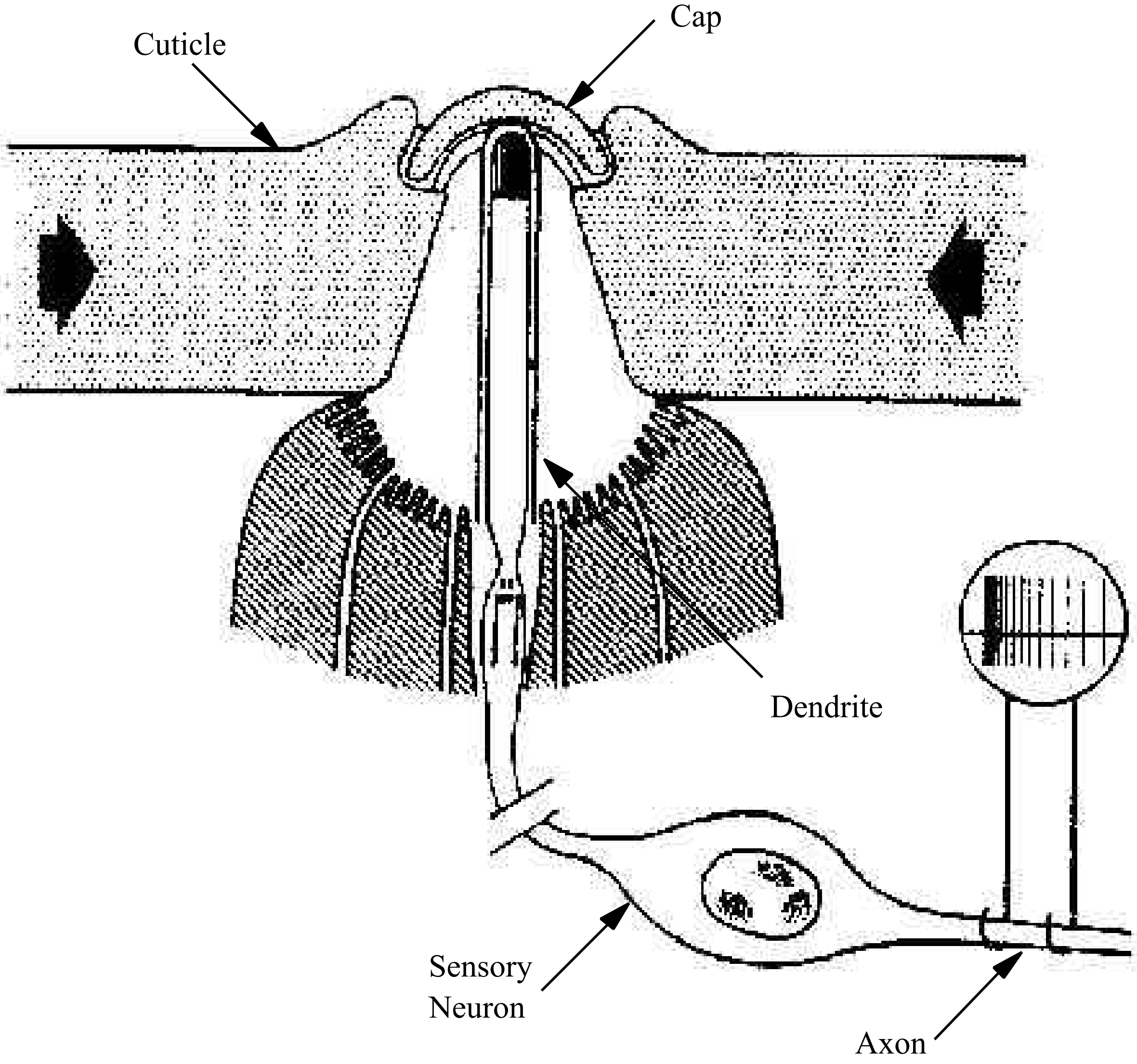

Figure 2.5 Campaniform Sensilla Morphology

2.2.2 Campaniform Sensilla Physiology

Spinola and Chapman [1975] were the first to demonstrate the directional sensitivity of campaniform sensilla. Zill and Moran [1981a] have characterized the response of the tibial (group 6) campaniform sensilla of the *Periplaneta americana*. In their study, they showed that there are two subgroups of sensilla: one with the long-axis orientation perpendicular to the axis of the tibia and one with the long-axis orientation parallel to the axis of the tibia. By applying bending moments, axial forces, and torques to the tibia, they were able to show that the two subgroups responded to only those stimulations that caused compressive strains perpendicular to the long-axis of the cuticular caps. Based on this evidence, campaniform sensilla are exteroceptors; they respond to strains created through interaction with the insect's environment. Zill and Moran went on to show that the sensilla are also stimulated by the activation of the tibial extensor and flexor muscles. Again, each subgroup was selectively sensitive to either the extensor or the flexor muscles. The sensilla may in this case be used to monitor the insect's muscle activities and could, in that facility, be classified as proprioceptors.

Further, Zill et al. [1981] have postulated that the tibial campaniform sensilla act as part of a negative feedback system. The sensilla tend to reflexively excite motoneurons that in turn lessen the stimulation of the sensilla. For instance, the proximal sensilla on the tibia are excited by forces that act in the direction of leg extension. They have shown that the same sensilla tend to excite motoneurons to the tibial extensors while inhibiting the flexor muscles. They suggest that this system may control load compensation and limit muscle tension. As more load (from walking for example) is applied to a leg, the sensilla fire with greater frequency which causes the extensor muscles to fire reducing the load. Similarly, as the muscle tension increases, sensilla fire which reduces the muscle exertion and protects the structural integrity of the tibia.

The response of campaniform sensilla to cuticular strains depends on the anatomy and orientation of the sensilla and the nature of the stimulus. It has been shown in the slit sensilla of spiders, which are the arachnid analog of the campaniform sensilla, that the size of the sensilla reflect the magnitude and the sensitivity of the response. The gradations in size of the sensilla correspond to variable sensitivity to strain. This effect is called range fractionation and may allow for different responses depending on the level of force application [Bohnenberger, 1981].

Additionally, it has been shown that within group 3, individual sensilla can have low threshold levels and high sensitivity to the rate of force application. This suggests that these sensilla might function as touch sensors and detect the stability of the leg placement during walking. Conversely, some of the sensilla have high threshold levels and are less sensitive to the rate of force application. These sensilla may report when adequate force has been applied during leg placement to begin propulsion and possibly influence inter-leg coordination [DiCaprio et al., 1997].

2.3 Material Properties of Insect Exoskeleton

Insect cuticle is not a typical homogeneous engineering material. It is in general formed of three components: fibrous chitin, protein matrix, and water. It is a composite with the protein matrix supporting the chitin fibers. In addition, water acts as a plasticizing agent affecting the material properties of the cuticle [Hillerton, 1984]. The three components are found in varying ratios in different cuticle which creates a huge range of material properties. Soft, pliant cuticles such as those found in interjoint membranes have a high chitin content while hard structural members such as the trochanter tend to have a larger quantity of the relatively stiff protein matrix [Hillerton, 1984].

To further complicate the issue, insect exoskeleton is a layered structure with a shell-like epicuticle, tough exocuticle, and pliant endocuticle [Hepburn and Joffe, 1976]. The epicuticle is a very thin (1 - 5 μm [Hillerton, 1984]) and stiff shell that only plays a small role in the bulk mechanical behavior of cuticle [Hepurn and Joffe, 1976]. The exocuticle and endocuticle can, in the case of pliant cuticles, have similar mechanical properties [Jensen and Weis-Fogh, 1962], but Ker [1977] has shown the exocuticle in some cases to be twice as stiff as the endocuticle. There is limited information describing the properties of the individual layers of cuticle due in part to the many variations of cuticle and the extreme difficulty in cleanly separating the layers for testing. As a result, most of the reported mechanical properties are for whole cuticle samples.

Most cuticle is tested for mechanical properties with typical material tensile tests. These tests measure in-plane tensile elasticities and strengths. Jensen and Weis-Fogh [1962] have shown through tensile and compressive buckling tests with locust tibias that cuticle performs similarly in tension and compression for small loads. In addition, their results show a nearly linear stress-strain relationship for loads less than half the ultimate tensile strength. It will be shown that normal operating stresses are well below this level in the cockroach trochanter. Several published material properties of stiff cuticles such as those found in the trochanter are shown in table 2.1.

Species	Specimen	Elastic Modulus (N/mm^2)	Ultimate Tensile Strength (N/mm^2)	Reference
Leucophaea maderae (Madeira Cockroach	Hind Femur	3160	158	[Hepburn and Joffe, 1976]
Schistocerca gregaria (Desert Locust)	Tibia	1800		[Ker, 1977]
Schistocerca gregaria (Desert Locust)	Hind Femur	1350	81	[Hepburn and Roberts, 1975]
Schistocerca gregaria (Desert Locust)	Hind Tibia	4600		[Katz and Gosline, 1994]
Schistocerca gregaria (Desert Locust)	Hind Tibia	9400	94	[Jensen and Weis-Fogh, 1962]

Table 2.1 Cuticle Properties

Cuticle can have a huge range of moduli demonstrated in the fact that intersegmental membranes have been shown to have elastic moduli 10^8 times more pliant than those listed in table 2.1 [Hepburn and Chandler, 1976]. Much of this variation can be attributed to the chitin-protein ratio, but the effect of sclerotization or tanning is also important. After molting, cockroaches have a soft, white, flexible exoskeleton. In the days that follow, sclerotization takes place as the cuticle browns and the chitin-protein matrix stabilizes through a process of cross-linking [Hepburn and Joffe, 1976]. The elastic modulus can change by orders of magnitude during this process. Tanning can occur to various extent in different locations. For instance, joint condyles tend to be tanned to a large extent suggesting a stiffer structure.

A final important characteristic of cuticle is its viscoelastic property. Viscoelasticity occurs in many polymeric materials and has characteristics of both fluids and elastic solids. Plastic deformation in viscoelastic solids is rate dependent [Reitveld, 1995]. Under small loads and small rates of strain, cuticle has nearly elastic behavior, but its viscous properties must be accounted for when performing material testing [Joffe and Hepburn, 1973]. The viscoelastic effects also explain some of the variation between measured values since the testing method can affect the results.

2.3.1 Material Parameters for Finite Element Analysis

Finite element analysis (FEA) requires definition of the material properties for the model. For an isotropic, homogeneous material undergoing linear elastic analysis the properties in consideration are the modulus of elasticity (E), Poisson's ratio (v), and the shear modulus of elasticity (G). The modulus of elasticity or Young's Modulus is the ratio of the stress versus strain for an elastic material and is the slope of the stress-strain curve in the elastic region. The shear modulus is the similar relationship between shear stress and shear strain. Poisson's ratio is a ratio of the axial deflection to transverse deflection of a material subject to axial loading or, in other words, how much a material "squashes" out the side when compressed [Popov, 1990]. The three values are related by the equation

$$G = \frac{E}{2(1+\nu)} \tag{2.1}$$

[Hibbler, 1993]. Therefore, only two of the three empirical values are required for the analysis.

The values in table 2.1 demonstrate some of the variation evident in the determination of the elastic modulus. Without specimen-specific material tests, probably the best course of action is to examine the analysis over a range of values and quantify the effects this parameter has on the analysis. This approach is discussed in Section 4.2.

Poisson's ratio may be even more nebulous than the elastic modulus. No major studies have examined this parameter for insect cuticle. This may be in part due to the difficulty in obtaining accurate data with a material as complex as cuticle, but may also be due to the fact that finite element analysis has not been performed before on insect exoskeleton. For most materials, Poisson's ratio falls within a narrow band from about 0.25 to 0.33 [Hibbler, 1993]. Using this range and values from other more studied fibrous materials such as wood, an estimate can be made of 0.30 for Poisson's ratio for cuticle. Again the approach here will be to test a range of values and quantify the results.

In this thesis, linear elastic analysis was performed on the trochanter model. This approach ignores the possible viscoelastic effects and non-linearity of the material. It will be shown that these assumptions are justified when analyzing the normal operating conditions of the cockroach. Normal locomotion activities do not cause significant plastic deformation of the exoskeleton. In fact, it can be argued that the campaniform sensilla prevent such deformation through modulation of muscle tension [Zill and Moran, 1981a]. Further, under small loads the material response is relatively linear and can be considered elastic [Jensen and Weis-Fogh, 1962; Hepburn and Joffe, 1976].

The assumption of an isotropic material may be more problematic in the analysis. Composite theory states that strength benefits can be obtained by using materials of varying properties to create a layered structure [Callister, 1991]. Ker [1977] has shown this to be the case in terms of the exocuticle-endocuticle layering.

Fortunately, isotropic mechanical properties can be expected in the plane of the cuticle [Barth, 1973]. The loading of the trochanter tends to involve inplane bending as this is where the strength of shell-like structures is realized. Also important to this work is the analysis of inplane normal stresses and their effects on the campaniform sensilla. Consequently, isotropic material properties are a reasonable assumption when considering inplane loading. The composite structure of cuticle and its benefits are certainly an area for further research since relatively little is known about the individual properties of the cuticle components.

2.4 Confocal Microscopy Model

In order to create a finite element model, an accurate, high resolution geometrical description of the trochanter is required. For the experiments, the metathoracic (rear) leg of the *Periplaneta americana* was used. Although the rear leg is the largest on the insect, the trochanter of an adult is only about 2.5 mm in length. Thus in the Zill and DiCaprio Labs, specialized microscopy and optical reconstruction methods were used to create the three-dimensional model.

The confocal microscopy method uses ultraviolet light to irradiate the specimen. Fluorescent qualities of many biological tissues allow the light to be adsorbed and reemitted at a lower frequency. The microscope is able to selectively filter the reflected light so that only the desired frequency is retained. Additionally, by using a pin-hole iris diaphragm a single focus plane can be captured. The result is that a three-dimensional object can be optically sectioned without performing tedious physical sections by focusing on separate layers through the specimen. The optical sections can be assembled using specialized software to create a three-dimensional model of the specimen.

2.4.1 Confocal Sectioning Method (Zill and DiCaprio Labs)

Adult *Periplaneta americana* cockroaches obtained from commercial suppliers were anesthetized with carbon dioxide and the left metathoracic leg was removed. The trochanteral segment of the leg was isolated by making transverse cuts through the distal coxa and proximal femur and then immersed in 1M NaOH for 1 hour to facilitate removal of soft tissues. Using 4% phosphate buffered formaldehyde, the trochanter was fixed and then bisected along its long axis into anterior and posterior halves. The two halves were dehydrated in an alcohol series and mounted in Permount on aluminum slides with holes drilled in the centers. The tissue was mounted between coverslips attached to each side of the slide which permitted viewing of the outer or inner surfaces of the exoskeleton.

The trochanter was visualized and reconstructed by employing its endogenous fluorescence under ultraviolet illumination. Although the fluorescent signal was stronger when the samples were viewed from the inner endocuticular surface, images could be obtained from either side of the specimens. Serial optical sections were taken with a Molecular Dynamics (DiCaprio Lab) or Biorad (Zill Lab) confocal microscope through the entire width of the cuticle using a 10X objective and stored as 512 x 512 pixel images with a 1.597 micron/pixel resolution.

In order to permit subsequent reconstruction of three-dimensional montages of each half of the trochanter, the image series were started from the same position in the z-axis in a plane above the specimen. The microscope stage was moved laterally between the series so that the series overlapped and included the full field of view of the trochanter. To construct montages, projection images were created and the exact extent of overlap was determined by using landmarks such as cuticular hairs and surface features. Individual images from each set, with corresponding z-axis values, were then placed in the determined positions in a 2048 x 1024 pixel template and saved as a series of bitmap files. These images were then reduced to 512 x 256 pixels, which corresponds to a 6.388 micron/pixel resolution, using resampling algorithms in Voxblast software (Vaytek Inc., University of Iowa). The montage images were loaded as data sets in Voxblast and resampled as 2-bit images by establishing gray-scale look-up values for tissue differentiation.

The montage series of the front and back halves were then combined and resectioned as a single data set for reconstruction of the entire trochanter. Alignment of the halves was performed by rotations and translations of the anterior half to match the posterior. The final resectioning created the unified model and was saved as TIFF bitmap images. These images were individually loaded into a graphics software program (Corel Photo Paint 4.0) to eliminate tissues which remained attached to the trochanter and to correct inadvertent breaking which occurred during bisection. This process was guided by measurements from images taken of a number of other hind leg trochanters.

Light rendered views of the two halves of the model in Voxblast are shown in figures 2.6 and 2.7. Included are the locations of the joint condyles and the locations of the four groups of campaniform sensilla on the trochanter. Notice also the large topology changes of the inner surface evident in the strut-like ridges. In contrast, the outer surface is relatively smooth.

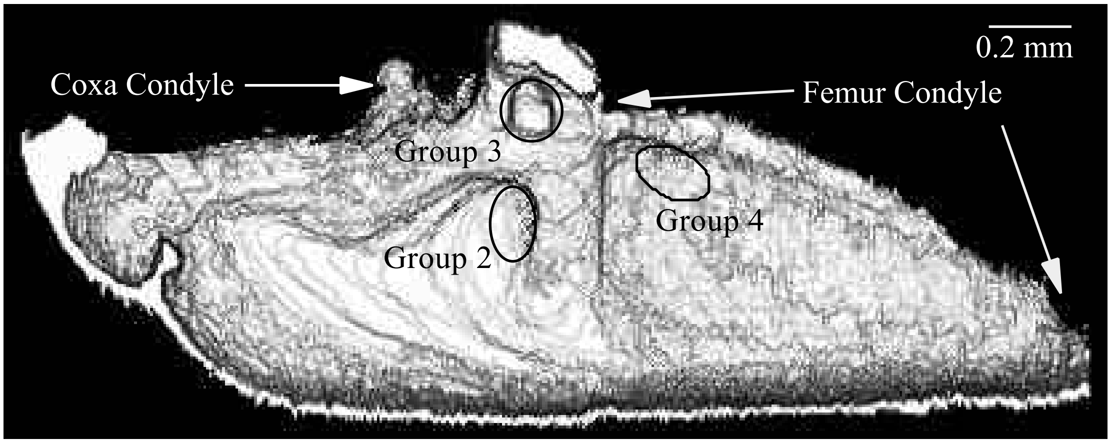

Figure 2.6 Anterior Half of Rendered Confocal Model, Interior View

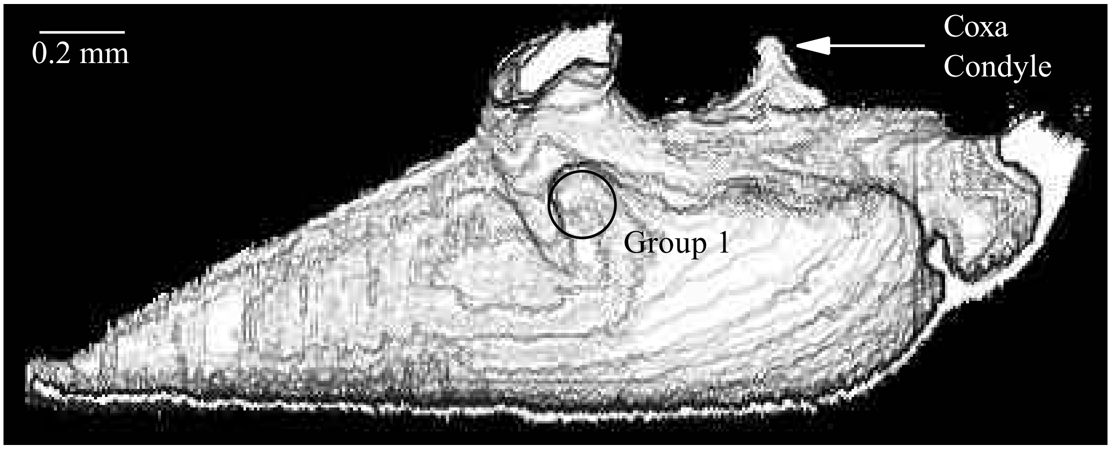

Figure 2.7 Posterior Half of Rendered Confocal Model, Interior View

2.4.2 Model Controls (Zill Lab)

To confirm the accuracy of the three dimensional reconstructions, leg segments were examined by plastic sectioning and conventional light microscopy. Specimens containing the trochanteral, distal coxal and proximal femoral segments were fixed in Karnovsky's fixative, dehydrated in graded alcohol series and embedded in Spurr's resin. These specimens were sectioned at 1 micron increments through the transverse axis of the trochanter and stained with toluidine blue. Sections were viewed using a Kohu video camera mounted to a conventional light microscope and digitized images obtained through a Matrox Millenium video capture board on a PC. The measurements were also confirmed by digitizing images of unfixed trochanteral segments that were isolated and viewed in a dissecting microscope.

At this point the high resolution confocal microscope model was ready for finite element meshing as described in Chapter III.

Chapter III

Computer Modeling

3.1 The Finite Element Method

The finite element method is a relatively recently developed (mid 1960's) analysis tool that can be used generally to solve a wide variety of analysis problems. It is well documented in many sources [Zienkiewicz and Taylor (1994); Fagan (1992); Spyrakos (1994)] and will not be reiterated here except in brief detail. Traditional closed form solutions work well for simple geometries or in cases where approximations can be made that allow the problem to be modeled so that it fits the theory. As geometries get more complicated and accuracy requirements get more stringent, traditional methods become cumbersome and computational methods are required.

3.1.1 Finite Element Theory

The finite element method surmounts these problems by breaking the analysis into parts or elements that are well described by the physics of the situation. Many smaller problems can then be solved and combined to provide the overall solution to the problem. The use of computers automates much of the process of setting up, solving, and examining the analysis. A very simple example of the method is a one-dimensional truss such as that shown in figure 3.1 with an axial compressive force. Although elastic theory easily describes this problem, we could break the truss into three elements, and each of these elements could be modeled as a spring-mass system shown in figure 3.2.

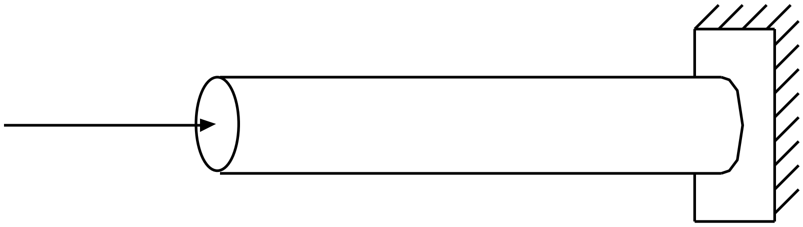

Figure 3.1 Truss Model

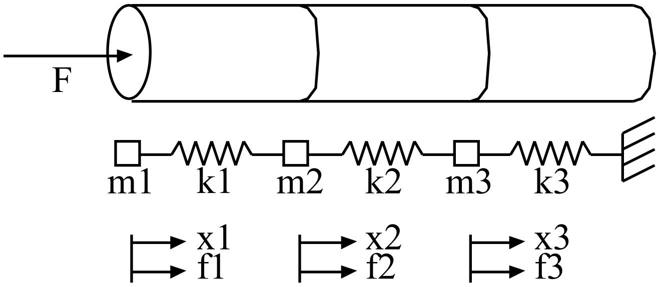

Figure 3.2 Three Element Truss Model

The equations of motion for the spring-mass system are

$$\begin{aligned} F &= m_1\ddot{x}_1 + k_1(x_1 - x_2) \\ 0 &= m_2\ddot{x}_2 + k_1(x_2 - x_1) + k_2(x_2 - x_3) \\ 0 &= m_3\ddot{x}_3 + k_2(x_3 - x_2) + k_3x_3 \end{aligned} \qquad 3.1$$

which can be written in matrix form as

$$\begin{bmatrix} F \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} m_1 & 0 & 0 \\ 0 & m_2 & 0 \\ 0 & 0 & m_3 \end{bmatrix} \begin{bmatrix} \ddot{x}_1 \\ \ddot{x}_2 \\ \ddot{x}_3 \end{bmatrix} + \begin{bmatrix} k_1 & -k_1 & 0 \\ -k_1 & k_1+k_2 & -k_2 \\ 0 & -k_2 & k_2+k_3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \qquad 3.2$$

$$\boldsymbol{F} = \boldsymbol{M}\ddot{\boldsymbol{X}} + \boldsymbol{K}\boldsymbol{X}$$

where **F**, **M**, and **K** are the load, mass, and stiffness matrices respectively. If static equilibrium is assumed, the accelerations are zero and

$$\boldsymbol{F} = \boldsymbol{K}\boldsymbol{X} \qquad 3.3$$

which is the familiar linear elastic spring equation in matrix form [Rao, 1995]. Once the problem is formulated as above, equation 3.3 is solved computationally for the displacements by effectively inverting the stiffness matrix, **K**

$$\boldsymbol{X} = \boldsymbol{K}^{-1}\boldsymbol{F} \qquad 3.4$$

This is the most difficult and time consuming part of the analysis since in a large problem the stiffness matrix can have hundreds of thousands of elements. After the displacements are known, it is a relatively simple matter to solve for the stress and strain tensors based on the geometry of the element [Porter, 1996].

For a general problem, the global stiffness matrix, **K**, of equation 3.3 is created by filling the matrix with the element stiffness matrices, $\mathbf{K}^{\mathbf{e}}$, based on the connectivity between elements. In order to form the element stiffness matrices, the concept of a shape function is introduced. Shape functions are typically polynomial approximations to the displacements within an element. They are chosen such that they yield the exact displacements at the nodes (vertices). Thus, the actual displacements within an element, **u**, can be approximated by the matrix of shape functions, **N**, times the nodal displacements, **a**

$$\boldsymbol{u} = \boldsymbol{Na} \qquad 3.5$$

Elastic theory provides the relationship between stress and displacement as

$$\varepsilon = \boldsymbol{Su} \qquad 3.6$$

where **S** is a linear operator. For example, in the case of two-dimensional plane stress

$$\varepsilon = \begin{bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{bmatrix} = \begin{bmatrix} \frac{\partial}{\partial x} & 0 \\ 0 & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial x} & \frac{\partial}{\partial y} \end{bmatrix} \boldsymbol{u} \qquad 3.7$$

which can be approximated with the matrix of shape functions by

$$\varepsilon = \boldsymbol{SNa} = \boldsymbol{Ba} \qquad 3.8$$

The constitutive relation between stress and strain can also be defined in matrix notation as

$$\sigma = \boldsymbol{D}\varepsilon \qquad 3.9$$

where, again for isotropic plane stress, **D** is

$$\boldsymbol{D} = \frac{E}{1 - \nu^2} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix} \qquad 3.10$$

The element stiffness matrix can now be derived using virtual work principles as

$$\boldsymbol{K}^e = \int_{vol} \boldsymbol{B}^T \boldsymbol{DB}\, d(vol) \qquad 3.11$$

In the one-dimensional example from above the shape functions are

$$\boldsymbol{N} = \begin{bmatrix} -x + 1/2 \\ x + 1/2 \end{bmatrix} \qquad 3.12$$

and the relationship between strain and displacement, **S**, is a derivative of x so **B** is

$$\boldsymbol{B} = \begin{bmatrix} -1 \\ 1 \end{bmatrix} \qquad 3.13$$

D is simply the modulus of elasticity or in the notation of above k_i. So the element stiffness matrix is

$$\boldsymbol{K}_i^e = \int_0^1 \begin{bmatrix} -1 & 1 \end{bmatrix} \begin{bmatrix} k_i \end{bmatrix} \begin{bmatrix} -1 \\ 1 \end{bmatrix} dx = k_i \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \qquad 3.14$$

Finally, the global stiffness matrix is assembled by placing the element stiffnesses in the global matrix based on the element connectivity

$$\boldsymbol{K} = \begin{bmatrix} k_1 & -k_1 & 0 \\ -k_1 & k_1 + k_2 & -k_2 \\ 0 & -k_2 & k_2 + k_3 \end{bmatrix} \qquad 3.15$$

which is what we obtained directly above. Once the displacements are solved for in equation 3.4 it is a simple matter to find the strains using equation 3.8 and the stresses by equation 3.9.

3.1.2 Finite Element Application

Finite element analysis (FEA) software has matured such that the solution of the analysis problem is automated, but development of the model can still be a time consuming and difficult process. Often computer aided design (CAD) tools are used to create a geometrical description in the form of a solid model. This model contains the complete three-dimensional description of the part.

The solid model then needs to be partitioned into elements that comprise the finite element model. This process is called meshing or creating a mesh for the part. The meshing activity presents many options. The first decision is to choose the type of element for the analysis. One, two, and three-dimensional elements are available in many different configurations. Each element type has specific qualities that effect the speed of the analysis, the accuracy of the solution, and the appropriateness for a specific problem. For example, the one-dimensional approach shown above might be useful for modeling the trusses of a bridge. Two-dimensional plate models work well for thin-walled pressure vessels, whereas the complex geometry of an automobile suspension component requires the used of three-dimensional elements [Zienkiewicz and Taylor, 1994].

A major consideration is the number of elements to approximate the model, also termed the mesh density. In general, the more elements used, the more accurate the solution. There is a trade off between accuracy and computation time, though.

More degrees of freedom through higher order elements and higher mesh densities require more solution time. One way to circumvent this is to use larger, less complicated elements in low stress areas. This allows the forces and displacements to be carried to the higher stress areas where a denser mesh is used and more accurate solutions are required.

A final important modeling consideration is how the model will interact with its environment. This entails applying forces and boundary conditions to the model. The boundary conditions can include constraints on the degrees of freedom (DOF) of the nodes of individual elements. Depending on the type of element, a node can have up to six DOF (three translations and three rotations). For example, a cantilevered beam would have all the DOF fixed of the nodes where the beam attached to its support. Field effects such as pressure and temperature can also be applied to elements.

3.2 The Algor FEA System

The Algor analysis software (Pittsburgh, PA) includes a complete set of tools for finite element analysis. Model development occurs in the CAD program SuperDraw III. SuperDraw has facilities for three-dimensional feature line construction and advanced surface and solid mesh generation. These mesh generators are extremely important in modern FEA packages since they greatly simplify the process of converting a solid CAD model to a FEA model. SuperDraw can also import models from a wide variety of other CAD programs such as Autodesk's AutoCAD. All of the models in this work were imported in this manner through the AutoCAD drawing exchange format, dxf [AutoCAD Users Guide, 1994]. The forces and boundary conditions are defined and applied to individual nodes once the mesh has been created in SuperDraw.

After the model has been drawn or imported and the mesh created, the properties of the elements, the materials, and the body forces are applied in the Decoder [Porter, 1996]. The Decoder is a text based menu system that acts as a database for the parameters of the model. Here, the type of element such as plate or brick elements are defined and any element specifics such as plate thickness are entered. The important material properties are also entered which for static linear stress analysis ignoring gravitational and temperature effects are Young's Modulus, the shear modulus, and Poisson's Ratio. The three values are related by equation 2.1. Finally, the decoder accepts information about body forces such as pressure, temperature, and gravitational effects. All of this information is applied when the Decoder is "run", and the finite element model is complete at this point.

The real work of the finite element software occurs next in the Processor. The Processor reads the input files, creates the stiffness matrix, inverts the matrix, and multiplies it with the load matrix to solve for the displacements [Porter, 1996]. There are only a few options, which mostly control the output files, that can be set in the Processor. The output files can be used as tabular results or for debugging purposes if there is a problem with the model. The output of the Processor is the set of nodal displacements which are immediately converted by another process into the stresses and strains for the model.

As a note, the plate elements used in this thesis have only five DOF at each node [Porter, 1994]. There is no rotational DOF about the z-axis normal to the element face. Because of this, the boundary conditions must be carefully applied to ensure that this DOF is fixed. Otherwise, the processor will apply a small stiffness to the DOF that allows the solution to continue, but yields incorrect results. This effect can be tested by creating a beam of plate elements in the x-y plane and applying a load also in the x-y plane without any boundary conditions. An incorrect solution will be returned.

The final step in the process is to view and analyze the results. Algor has a program called SuperView which is designed specifically for that purpose. SuperView allows many different stress calculations to be viewed as color-coded graphics. In addition, strains or displacements can be viewed and the values for any node can be obtained. SuperView has a wide variety of options for creating and viewing the graphical output.

3.3 Confocal Microscopy Data

The confocal microscopy model data as described in Section 2.4 is output in a proprietary file format called a polygon file by the Voxblast imaging software. The polygon file is similar to a finite element mesh in that the surfaces of the model are represented by a large number of polygon facets. The actual file format contains a section for three-dimensional node locations and a section for connectivity data between the nodes. The connectivity data can be represented as vectors that create the polygons of the surface model. A graphical representation of a polygon file is shown in figure 3.3 with an inset showing the element shapes.

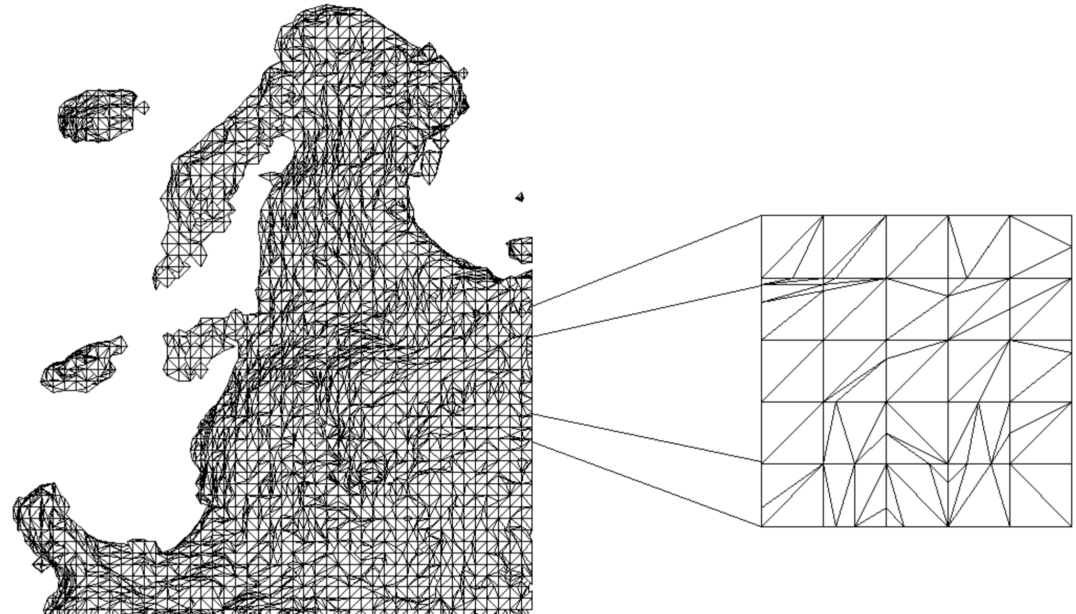

Figure 3.3 Sample Polygon File

The polygon file format is inappropriate for finite element analysis for several reasons. One obvious problem is the shape of elements in the polygon file. In the finite element method, there is a necessity to represent any possible element shape or size by a standard model. This is accomplished by a process of mapping the actual element to the theoretical standard element. In an isoparametric mapping, there is a one-to-one correspondence between the actual and the theoretical geometry shown in figure 3.4.

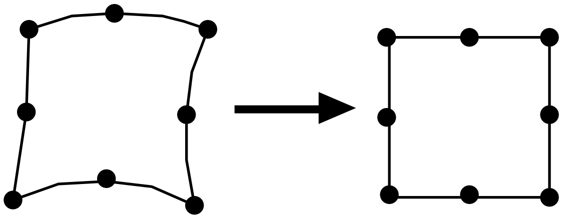

Figure 3.4 Isoparametric Element Mapping

If the actual element is extremely distorted with respect to the theoretical element, inaccuracies or even singularities can occur that can cause computational singularities in the solution. In the case of the triangular elements in the polygon file, equilateral triangles are optimal, and any angles less than about 30 degrees and more than 90 degrees should be avoided. As can be seen in the inset of figure 3.3, there are many elements that violate this criteria.

Another problem with the polygon files is the lack of control of the mesh density. Mesh density is an important parameter that controls the accuracy and speed of the solution. Often the analyzer would like to be able to use a smaller mesh density in critical areas which is not possible with the polygon files.

Finally, the polygon files of the trochanter are unacceptable for FEA because of the manner that the surfaces are represented. The shell-like exoskeleton of the trochanter is mapped by triangular plate elements on both the inner and outer surfaces in the polygon file. A three-dimensional plate element model for FEA must represent the solid shell by a single surface. The surface can be visualized as the center of the shell or average of the outer and the inner surfaces. The thickness of the single surface is then input as a parameter in the Decoder.

3.4 Mesh Generation

The mesh generation process begins with the data exported from the confocal microscope model in the form of a polygon file. For the above reasons, this format is unfit for FEA, so a usable mesh must be created from this data. A mesh created by hand would be extremely time consuming and error prone. Commercially available meshing tools were also not a feasible solution based on the unique data format and problems with the inner and outer surface representation. As a result, a custom meshing tool was created.

3.4.1 Meshing Method

It was decided early in the code development to use three-dimensional plate elements. These elements have four nodes, and if they are triangular, two of the nodes lie at the same position. The third dimension is a thickness parameter that is assigned in the Decoder. Plate elements are appropriate for thin shell-like models. A rule of thumb is that the maximum dimension of the model must be at least ten times the thickness in order to use plate elements [Porter, 1996]. In the case of a fully developed trochanter, the length is approximately 3 mm and the average thickness is about 0.1 mm—well within the requirements.

The meshing problem is to take the vertex and connectivity data of the polygon file and create appropriate plate elements. The method that was adopted after trying several different approaches is as follows. The connectivity data is discarded and only the vertex data is examined. This data set is a field of approximately half a million three-dimensional points that represent the inner and outer surfaces of the trochanter. A sampling method is employed where only a small geometric volume of the model is examined at a time as depicted in figure 3.5. The points in this volume are sorted according to the coordinate normal to the surfaces. This normal coordinate is then examined to determine the location of the largest difference between successive points. This gap represents the differentiation between the upper and lower surfaces in the data, and any point above it is considered on the upper surface and vice-versa. The normal coordinates of the upper and lower surfaces can then be averaged. The difference between these two values is the average thickness of the model over the sampled volume. Finally, a single four-dimensional point can be distilled from the data, by using the three coordinates of the center of the sampled volume and the thickness value.

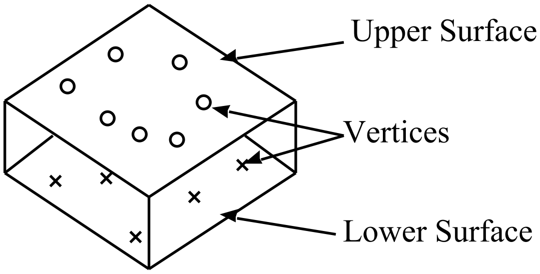

Figure 3.5 Sampling Vertex Data

The above process is repeated over the entire domain of the data yielding points that represent the center of the exoskeleton shell and the thickness at that point. The points are then used to create triangular elements. The final mesh is output as an AutoCAD dxf file with the elements described by the AutoCAD 3Dface object [AutoCAD Users Guide, 1994]. The thickness values are sorted into categories and each category is assigned a color. When the elements are created, the thickness is encoded into each by assigning a color. The colors and their corresponding thicknesses are output into a separate file which is later input into the Algor Decoder.

3.4.2 Code Development

The mesh generator code is included in Appendix A. The first step of the process is to parse the polygon input file for the vertex data and read it into memory.

Because of the large number of data points (about 500,000 for a full model), and because each node in the finite element model requires a search of the entire data set, an efficient data structure is a necessity. A binary tree was chosen because its worst case search takes $\log_2 n$ comparisons where n is the number of data points[Schildt, 1990]. The data is stored in the binary tree according to the x-coordinate (see figure 2.2 for coordinate system). Also at this time, the cartesian coordinates are transformed into polar coordinates.

Sampling of the data occurs next. The algorithm is as follows:

```
1. Determine the extent of the sample volume in the x-direction
2. Search the tree recursively to find all points in the x-range of the sample volume
3. If points are found in the x-range,
    4. Determine the extent of the sample volume in the theta-direction
    5. Search the candidate points for values within the theta-range
    6. If too few points are found in the sample volume,
        7. Expand the sample volume by a set amount
        8. Repeat 2 - 7 until enough points are found or until the sample
           volume reaches its maximum size
```

At this point we have all the points in the data set that lie within the sample volume. The iteration in step 8 of the above algorithm guarantees that at least 20 points are found. Assuming an equal number of points on the inner and outer surface of the trochanter, there is a 1/2 probability that a point lies on the either the inner or the outer surface. With 20 points, the probability of all the points laying on either surface is

$$\frac{1}{2}^{19} = 1.9x10^{-6} \qquad 3.5$$

and since there are on the order of 5000 nodes in a model, the chances are small that either the inner or outer surface will not be represented over the full model.

With the high probability that both surfaces are represented, the thickness can be determined over the test volume by the following algorithm:

1. Use a quicksort algorithm to sort the points by the radial coordinate
2. Determine the largest difference between any two successive points
3. Average the radial coordinates of the points above and below the largest difference
4. The difference of the two averages is the average thickness over the test volume

The code allows several choices of how to best represent the points in the sample volume with a single node. These include using a single point in the volume, using the lower surface average, the upper surface average, or a central average. The central average is usually the best choice since in a plate element the thickness parameter is centered on the two dimensional element.

The information contained in each sample volume has been reduced to four pieces of data—three spacial coordinates and one thickness value. The elements are now created and written to a dxf file. The meshing is a simple task of connecting the nodes created in the sampling procedure. The dxf format was chosen since it allows the models to be viewed and modified in AutoCAD. The dxf files may be also directly imported into Algor's SuperDraw. Plate elements could be represented by either simple lines or by 3Dface elements. The advantage of using the 3Dface elements is that they are true solid model elements and as such can be rendered in AutoCAD for model verification purposes. The meshing code allows output in either format. The process of meshing converts a 55 Megabyte polygon file with half a million vertices to a 2 Megabyte dxf file with approximately 5000 nodes and 11000 elements.

3.4.3 Examples

Several tests were run on the meshing code before the trochanter model was attempted. These tests used files of approximately 100,000 vertices that covered the surfaces of simple shapes. The points were generated with a degree of randomness to simulate the distribution found in the polygon files. The first was a simple cylinder with points on the inner and outer surface, shown in figure 3.6. The bluish color shows the thickness of the cylinder wall which is constant along its length.

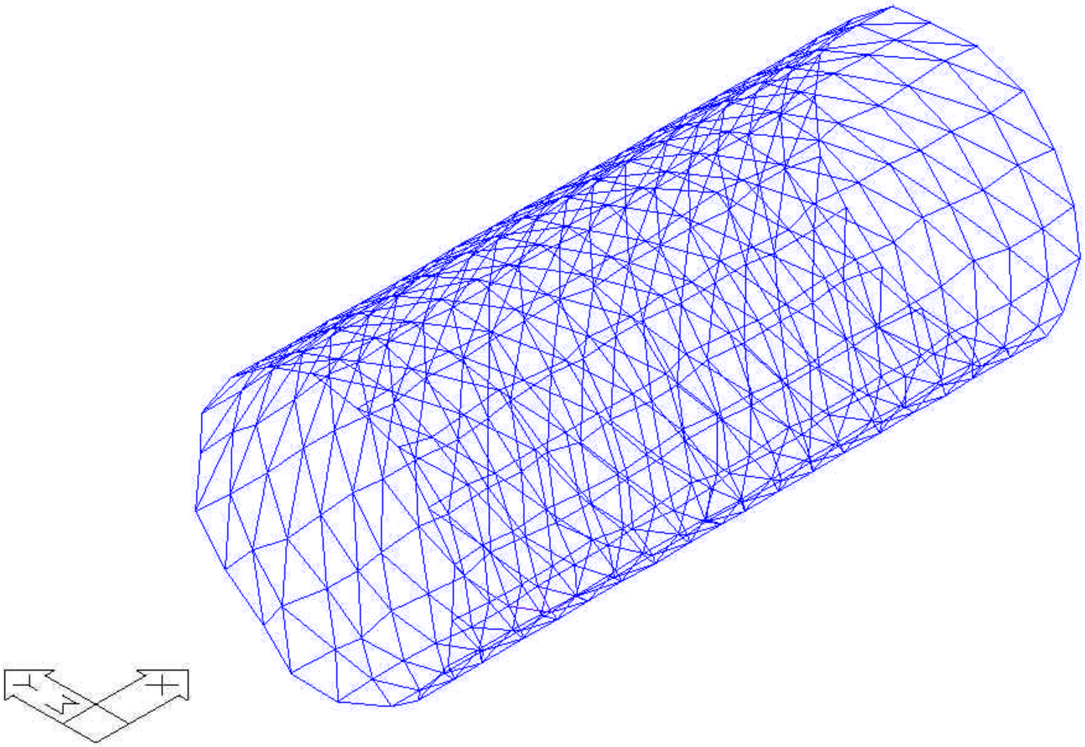

Figure 3.6 Cylinder Test Model

The second test case is a spherical shell model, shown in figure 3.7. Again, the colors show the thickness of the shell. This shell is also of constant thickness, but the model shows some distortion near the ends. This distortion is due to the cylindrical sampling method that was used. The model was sampled radially about the x-axis which causes the shell to appear thicker near the ends. This radial sampling was chosen since the trochanter model is well described by a cylinder thus reducing this type of distortion. Additionally, code was added to the meshing algorithm to close the ends of the model so that a continuous model could be created.

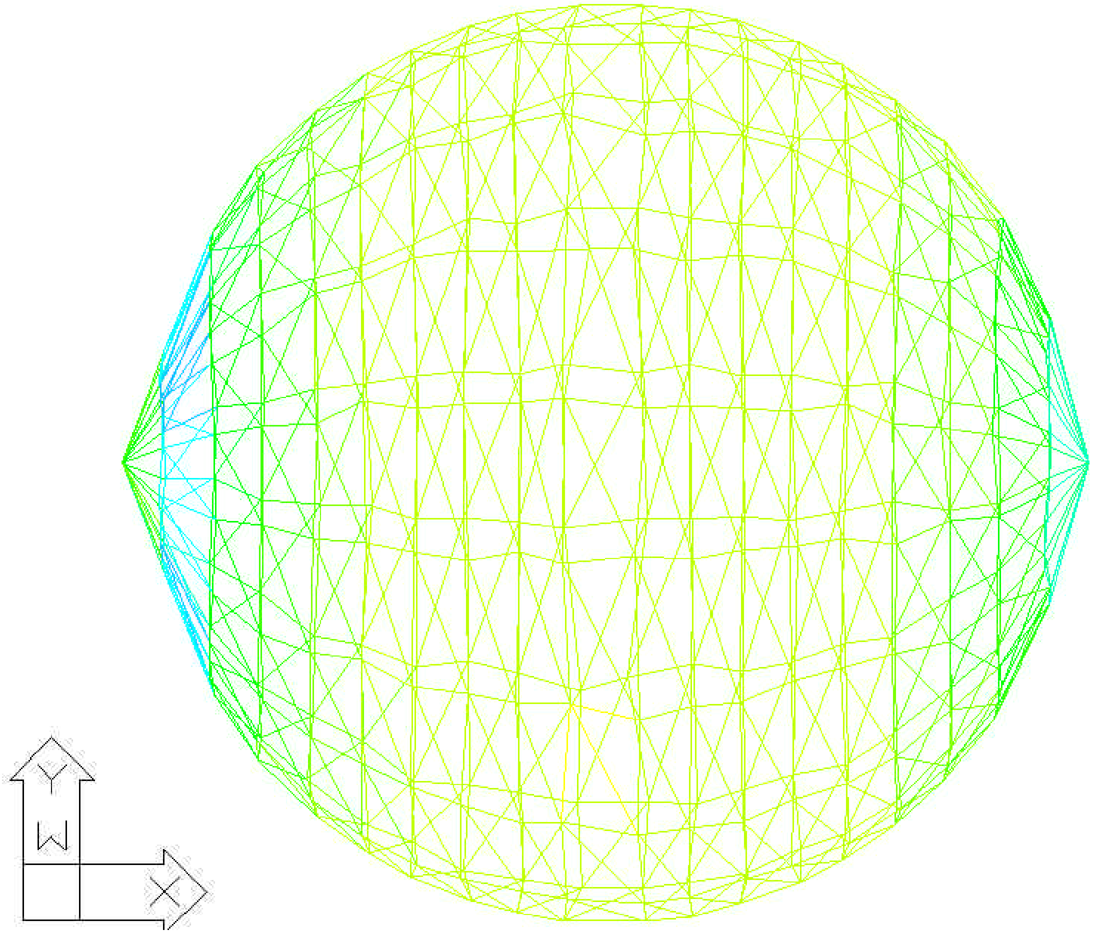

Figure 3.7 Sphere Test Model

3.4.4 The Trochanter Model

The meshing tool provides a simple and efficient means for generating first-order triangular plate element meshes. The mesh density is easily variable, and the performance is fast—a 410,000 vertex file can be meshed with 5500 elements in 70 seconds on a 500 MHz DEC Alpha Workstation. The result is a mesh such as that shown in figure 3.8 and 3.9. This mesh was used for the analyses unless noted otherwise. Note again that the colors represent discrete thickness values. The model incorporates 20 thickness values ranging from 6.5 μm shown as red to 265 μm which is blue. Notice the thick vertical ridge and the top central ring. Also indicated on these two figures are the locations of the sampling points for each group of campaniform sensilla, marked as small black circles. Three nodal locations for each group were selected for the data analysis.

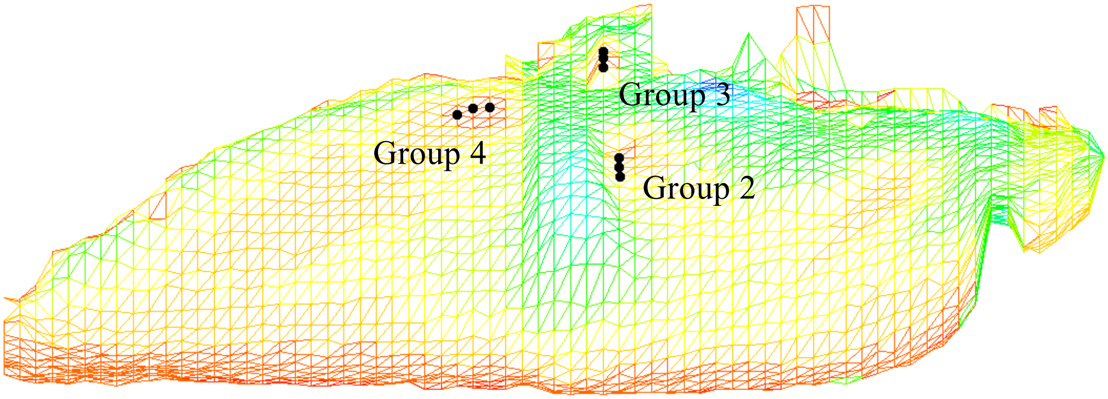

Figure 3.8 Trochanter Model, Anterior

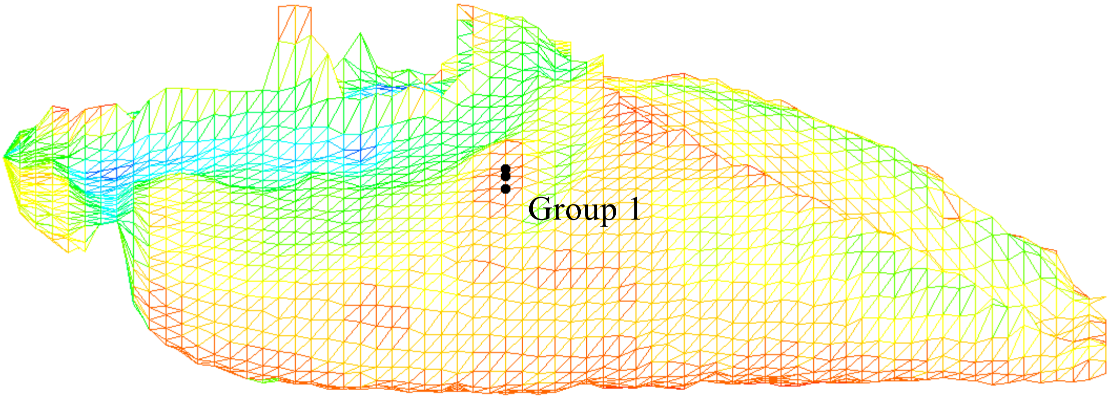

Figure 3.9 Trochanter Model, Posterior

Figure 3.10 shows a finite element mesh laid over the confocal microscope model from which the data was extracted. This image gives an indication of the close correlation between the two models. The proximal end of the trochanter on the right side of the figure shows a thick area that is not well modeled by plate elements. While this is the insertion of the large coxal muscle group 177, it not near any of the campaniform sensilla regions and therefore the stress distribution in that area is of less importance. The exaggerated thickness plate elements in that area will effectively transmit the forces and displacements to the rest of the model. [Porter, 1996]

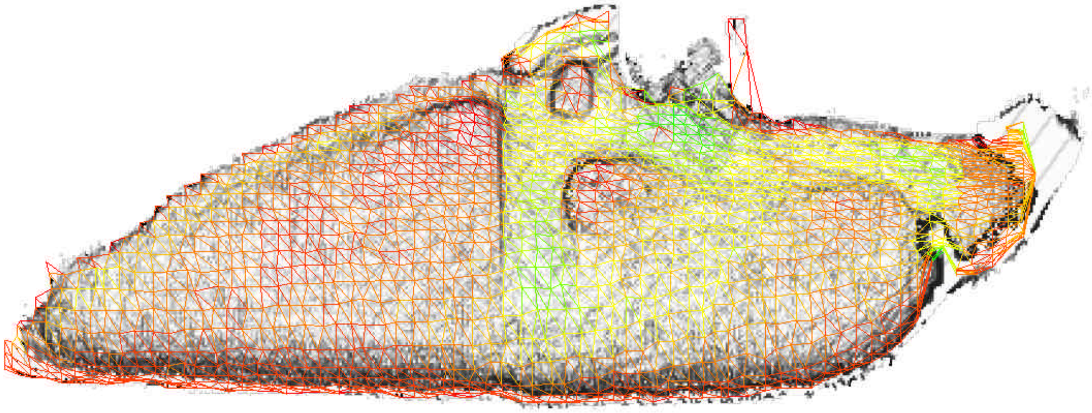

Figure 3.10 Finite Element Mesh Over Confocal Model

Figure 3.11 is the mesh over a photomicrograph of the original trochanter shown in a negative image for clarity. The orientation of the original exoskeleton and the finite element model are not exact, but a good correlation can be seen. The photo also has the femur still attached in the upper left corner.

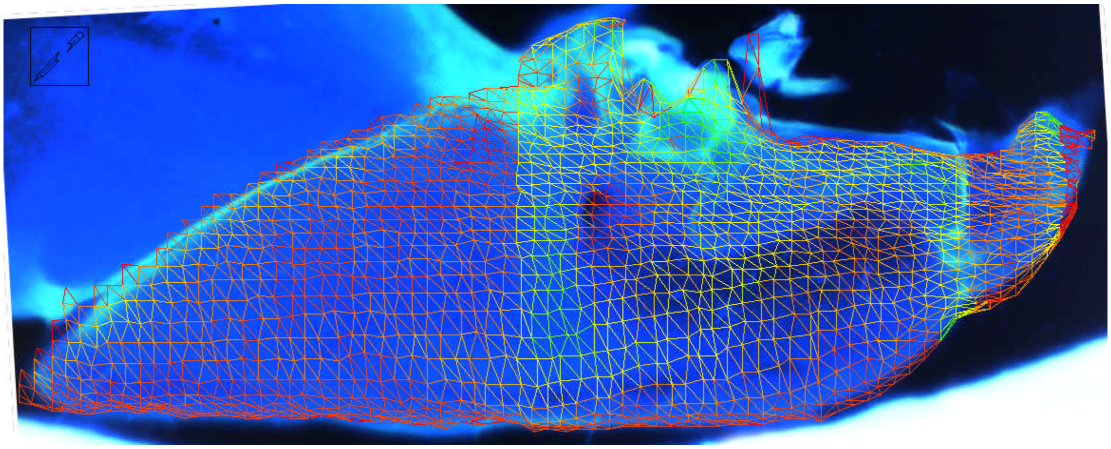

Figure 3.11 Finite Element Mesh Over Original Trochanter

Chapter IV

Results

4.1 Convergence Study

For first order elements such as those used in this model, the displacements within an element are approximated by linear shape functions. The strains (and stresses) which are of interest are first derivatives of the displacements and are therefore constant within an element. Loading conditions such as bending which are present in the trochanter model cause linear strain gradients which can only be approximated by the constant strain distribution within the element. As such, the finite element solution is only approximate, and the accuracy of the solution must be determined.

It can be shown [Zienkiewicz and Taylor, 1994] that the order of convergence to the exact solution for displacement is given by

$$O(h^{p+1}) = O(h^2) \tag{4.1}$$

where h is the element size and p is the order of the polynomial approximation (p = 1 for linear elements). Thus, there is a quadratic convergence of displacement. The convergence of strain is given by

$$O(h^{p+1-m}) = O(h) \tag{4.2}$$

where m is the number of derivatives of the displacement. There is a linear convergence of stress and strain in the model; if the element size is halved we should expect the strain error to also be halved.

The peg leg flexion experiment (described in Section 4.4) was selected and run at varying mesh densities to calculate the error. The element size, h, was determined by the number of elements along the x-axis. The Von Mises stress representation at the group 2 location was chosen for the analysis. A mesh with 36 elements along the x-axis returned a value of 1.122 MPa and a mesh with one third the element size, 108 elements along the x-axis, returned 1.174 MPa. Based on linear convergence, the actual stress value can be extrapolated by

$$\frac{\sigma_1 - \sigma_a}{\sigma_2 - \sigma_a} = \frac{O(h_1)}{O(h_2)} \qquad 4.3$$

$$\frac{1.122 - \sigma_a}{1.174 - \sigma_a} = \frac{O(h)}{O(h/3)} = 3$$

$$\sigma_a = 1.201$$

This gives an estimated error of 6.5% with 36 elements along the x-axis and 2.1% with 108 elements along the x-axis. 6.5% is an acceptable level of error based on the assumptions in the model, but the fine geometry in the areas of the campaniform sensilla is not well represented at this level of accuracy. Consequently, a mesh density of 72 elements along the x-axis was chosen which represents the geometry well and has an estimated error in the strain values of 3.25%. The model at this resolution has 3251 nodes and 6320 triangular elements.

4.2 Material Studies

Based on the values of table 2.1, Joffe and Hepburn's [1976] measured elastic modulus of 3160 N/mm^2 for the hind femur of the cockroach was used for all analyses. Global variations in the elastic modulus can be accounted for as follows. The **D** matrix used in the formulation of the stiffness matrix of equation 3.11 for the three-dimensional isotropic elements used in this analysis is

$$\boldsymbol{D} = \frac{E(1-\nu)}{(1+\nu)(1-2\nu)} \begin{bmatrix} 1 & \frac{\nu}{1-\nu} & \frac{\nu}{1-\nu} & 0 & 0 & 0 \\ \frac{\nu}{1-\nu} & 1 & \frac{\nu}{1-\nu} & 0 & 0 & 0 \\ \frac{\nu}{1-\nu} & \frac{\nu}{1-\nu} & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1-2\nu}{2(1-\nu)} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1-2\nu}{2(1-\nu)} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1-2\nu}{2(1-\nu)} \end{bmatrix} \qquad 4.4$$

The calculation of the strains requires the inversion of the stiffness matrix, and because the inversion of the triple matrix multiplication of equation 3.11 is equal to the multiplication of the three matrix inverses, the strain is proportional to the inverse of **D**

$$\boldsymbol{D}^{-1} = \frac{1}{E}\begin{bmatrix} 1 & -\nu & -\nu & 0 & 0 & 0 \\ -\nu & 1 & -\nu & 0 & 0 & 0 \\ -\nu & -\nu & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2(\nu+1) & 0 & 0 \\ 0 & 0 & 0 & 0 & 2(\nu+1) & 0 \\ 0 & 0 & 0 & 0 & 0 & 2(\nu+1) \end{bmatrix} \qquad 4.5$$

It follows that the strain is inversely proportional to modulus of elasticity, E. Thus, E is simply a linear scaling factor for the strain values. In calculating the stresses of equation 3.9, E is eliminated and does not affect the stress values.

Equation 4.5 shows that there is a proportional relationship between Poisson's ratio (ν) and the off-diagonal strain elements. Thus as ν gets larger, the strains increase, but the magnitude depends on the orientation of the strain tensor. To quantify the effect, the peg leg flexion experiment (described below in Section 4.4) was run using values for ν of 0.25, 0.30, and 0.33 which covers a wide range of materials [Hibbler, 1991]. The strains were measured at the campaniform sensilla locations normal to the average angle of the long-axis of the sensilla (short-axis strains). The results are shown in figure 4.1.

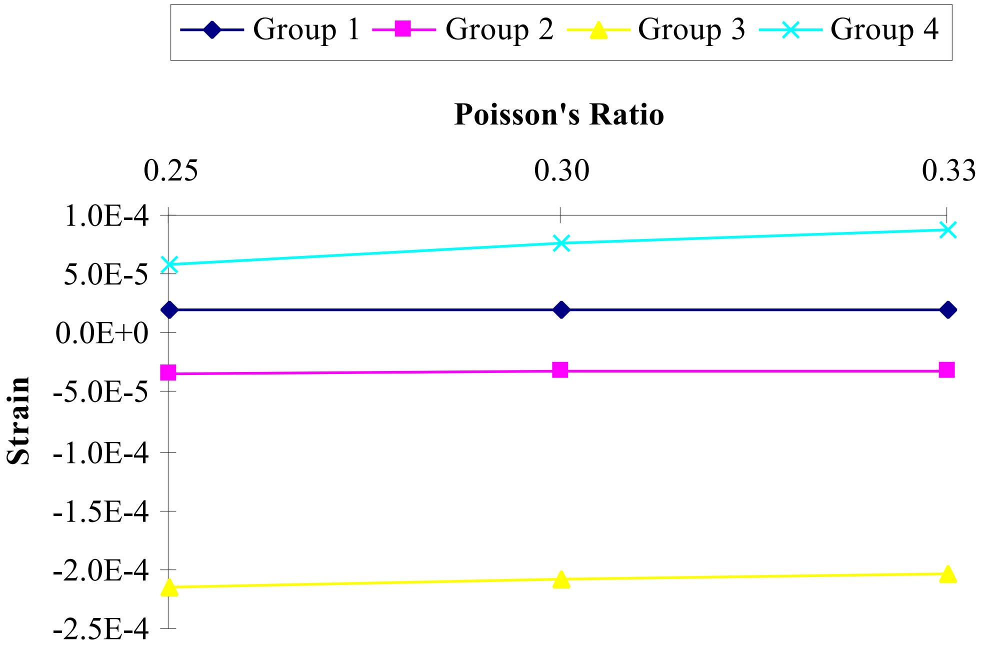

Figure 4.1 Poisson's Ratio Analysis

The effect of Poisson's ratio depends on location and orientation. Larger values of ν in all cases led to greater strain. Going from 0.25 to 0.33 had an average strain increase of 6.1% for groups 1 - 3, but group 4 showed a change of 46.6%. The variation in group 4 will be discussed below. A value of 0.30 was used as an approximation of Poisson's ratio for all of the experiments, and each case needs to be considered individually as to the effect of poisson's ratio.

4.3 Campaniform Sensilla Location and Orientation

It has been shown that campaniform sensilla are most sensitive to strains perpendicular to the long axis of the cuticle [Zill and Moran, 1981a]. Consequently, it was necessary to locate the sensilla groups on the model and orient their axes. The locations are shown in figures 3.8 and 3.9 and are relative to an arbitrary coordinate system. The orientations, on the other hand, are reproducible. Direct measurements were made on the trochanter used for the confocal imaging relative to the coordinate system shown in figure 2.2. The x-axis is oriented along the flat ventral side of the trochanter, and the angles are measured counterclockwise from the positive x-axis. The results are summarized in table 4.1.

Group	Number of Sensilla Measured	Average Long Axis Orientation (degrees)	Standard Deviation
1	7	86.9	2.4
2	15	85.7	2.3
3	12	67.7	2.2
4	14	15.5	9.2

Table 4.1 Campaniform Sensilla Orientation

It is interesting to note that within groups 1, 2, and 3 the sensilla are closely oriented to each other, but group 4 has more than four times the variation. It should be stated that because of group 4's location on a curved section of cuticle, the orientation measurements were the most difficult to make accurately. Group 4 was also by far the most affected by the Poisson's ratio study (46.6% versus 6.1% average for groups 1-3). This fact suggests that there are significant strains in other orientations than along the short-axis of the sensilla that are causing the off diagonal v terms in equation 4.5 to be prominent. The variation in the orientation of the group 4 sensilla may be structured to detect these strains.

4.4 Peg Leg Experiments

The peg leg experiments get their names from a series of classic experiments performed by von Buddenbrock [1921; 1953] and Wendler [1964; 1966] that studied the effects of leg ablations on walking patterns of the stick insect. In these experiments, the middle legs were amputated at the proximal femur leaving just the trochanter and coxa which could not touch the ground during normal walking. The amputations produced striking changes in the insects walking patterns. During normal walking, a tripod gait is produced where the front and rear legs on a side are in phase with the middle leg of the opposite side. The ablated specimens, however, immediately adapted to a diagonal gait where the front and rear legs on a side were 180 degrees out of phase, similar to that seen in quadrupeds [Hughes, 1952; 1957].

To determine which sensory inputs caused the gait adaptation, prosthetic limbs (peg legs) were attached to the partially ablated legs allowing normal contact with the walking surface. The tripod gait once again was observed. Wendler [1966] concluded that joint angle receptors were not responsible for the changes based on his observation that ablation of the receptors had no effect on walking patterns and that the partially ablated legs continued to move through significant joint angles even without the prosthetic limbs. The conclusion was that force sensory information in the proximal leg must be contributing to the gait coordination.

Several researchers have performed studies to determine how force information is integrated into the control of walking patterns. Pringle [1940] was the first to show that deformation of the trochanter caused a response from the trochanteral campaniform sensilla and a strong reflex excitation in the slow motoneuron to the trochanteral extensor. Pearson [1972] repeated this experiment and postulated that this reflex is part of a positive feedback system; as the trochanter is strained, the extensor motoneuron is excited more rapidly which causes increased strain in the trochanter. Zill has recently repeated the peg leg experiments while monitoring tibial extensor motoneuron activity and was able to show that in the ablated leg, the increase in activity did not occur, but if the partially amputated leg made contact with the ground through a prosthesis, the characteristic increase in activity was observed.

4.4.1 Campaniform Sensilla Group Contribution

The contribution of the individual groups of sensilla on the trochanter to the extensor reflex response has not previously been described in detail. DiCaprio et al. [1997] have performed experiments stimulating individual sensilla which support the above conclusions. One of the main goals of this work has been to characterize the placement and response of the four groups of sensilla on the trochanter. In terms of the peg leg experiments, it is of interest to determine which groups are stimulated by external forces applied to a simple distal leg (peg-leg) and which are stimulated by internal muscle generated forces.

To this end, analogous experiments were performed on the insect and within the finite element model. The parallel experiments were used to help strengthen the results and act as a control for the model. The finite element experiments used a simple planar model of the femur attached to the high resolution trochanter model. External forces were applied to the femur model which were accurately transmitted to the trochanter. The biological experiments used a preparation where the trochanter was fixed at the proximal end and forces were applied to the distal femur [DiCaprio et al., 1997].

4.4.2 Biological Experiments

The procedure was as follows. An animal was restrained ventral side up and all nerves innervating a hind leg were crushed close to the associated thoracic ganglion. One pair of 30 μm wires, insulated to their tips, was inserted through holes made in the cuticle and placed near the main leg nerve (n5) in the coxa. Another pair of wires was placed next to n5 distal to the first pair. To limit the number of sense organs recorded in nerve 5, the leg was severed in the proximal tibia or distal femur and the branches of n5 and 5r8 were cut just distal to the trochanter. The trochanteral segment was then immobilized by securely gluing it on its ventral edge to a pin placed close to the coxal-trochanteral joint.

Bending forces were then applied to the distal femur using a blunt, stiff segment of thick tungsten wire attached to a piezo-electric crystal, shown in figure 4.2. In preliminary experiments, voltages have been delivered to the crystal as step functions generated by a stimulator. A second probe, of much finer etched tungsten wire, attached to another crystal was used to stimulate the cuticular caps of individual trochanteral sensilla to facilitate identification of discharges. After a series of test was applied, individual groups of sensilla or the caps of single receptors were selectively ablated using a sharp pin or etched tungsten wire.

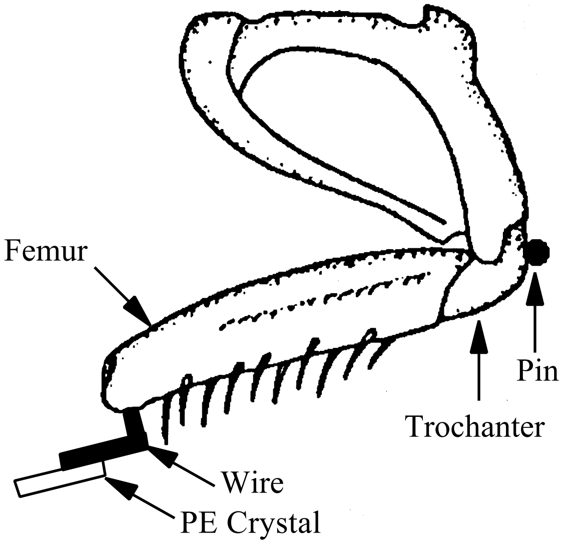

Figure 4.2 Peg Leg Forced Flexion Experiment

4.4.2.1 Forced Flexion

A ramp force was applied and held on the dorsal femur. The force acted perpendicular to the axis of the femur causing dorsal bending. This type of bending could occur as a result of forces exerted by the large extensor muscles of the coxa during normal walking. Although the true structural force vector acting at the femoral-tibial joint is certainly variable, the forced flexion component is significant. The electroneurogram readings of figure 4.3 show the combined activity of group 3 and 4 sensilla. Upon ablation of the group 3 sensilla, group 4 remains in the response as verified by stimulation of an individual sensilla. Qualitatively, groups 1 and 2 appear not to affect the response although definitive experiments have not yet been performed for determination.

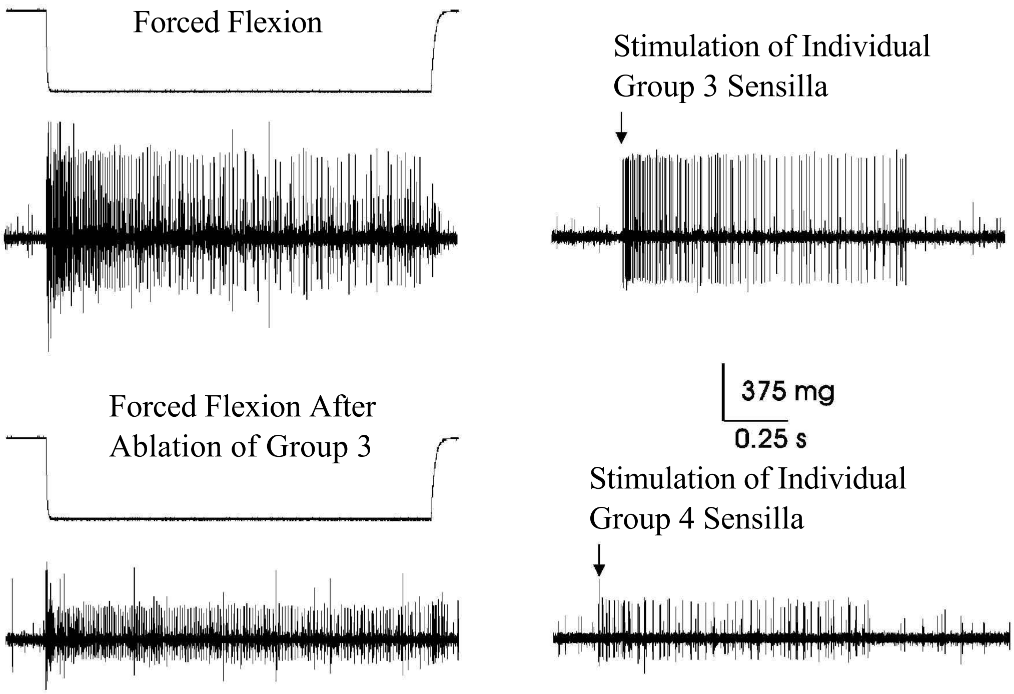

Figure 4.3 Peg Leg Flex ENG Response

4.4.3 Finite Element Model Experiments

The finite element experiments make use of the combined trochanter-femur model and all forces are applied to the distal end of the femur segment. The model has been rotated such that the two coxal condyles lie in an axis parallel to the z-axis. This orientation allows hinge-like boundary conditions to be applied to the condyles; they are fixed in all translations and in rotations about the x and y-axes. Thus, rotation about the condyle axis is permitted which simulates having a hinge joint between the coxa and trochanter. To fix the model for analysis, the proximal end of the trochanter on the ventral side is prevented from translating in the x and y-directions similar to the glued pin of the complementary experiment.

4.4.3.1 Forced Flexion

A force of 1.96 mN, which is approximately one quarter of the weight of the animal, was applied to the distal end of the femur as in the biological experiment. This level of force was above the threshold value that caused sensilla activity during the biological experiment. The force acted in the plane of the trochanteral-femoral joint perpendicular to the axis of the femur similarly to figure 4.2. The Von Mises strain representation of the anterior and posterior halves is shown in figure 4.4 and 4.5. The Von Mises strain is given by

$$\varepsilon_{VM}=\sqrt{\tfrac{1}{2}((\varepsilon_x-\varepsilon_y)^2+(\varepsilon_y-\varepsilon_z)^2+(\varepsilon_z-\varepsilon_x)^2)+3(\varepsilon_{xy}^2+\varepsilon_{yz}^2+\varepsilon_{zx}^2)} \qquad 4.6$$

and is derived from the maximum distortion energy failure theory [Popov, 1990]. It is often used in stress analysis to determine yield failure since the value can be compared directly to the tensile yield stress. For deformation analysis, it is a convenient scalar representation of the state of strain at a point.

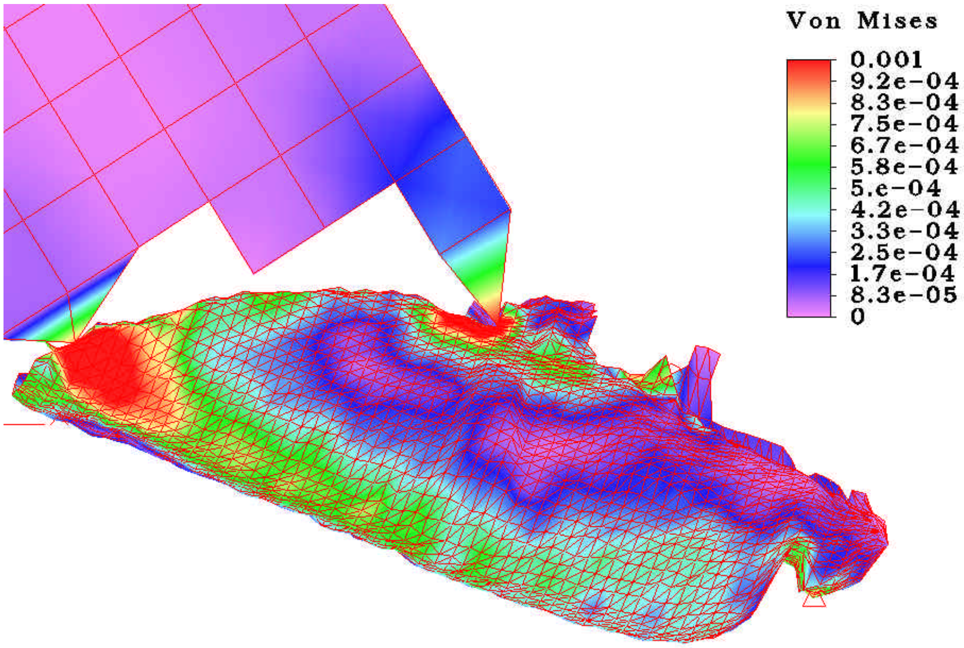

Figure 4.4 Peg Leg Flex, Anterior, Von Mises Strain

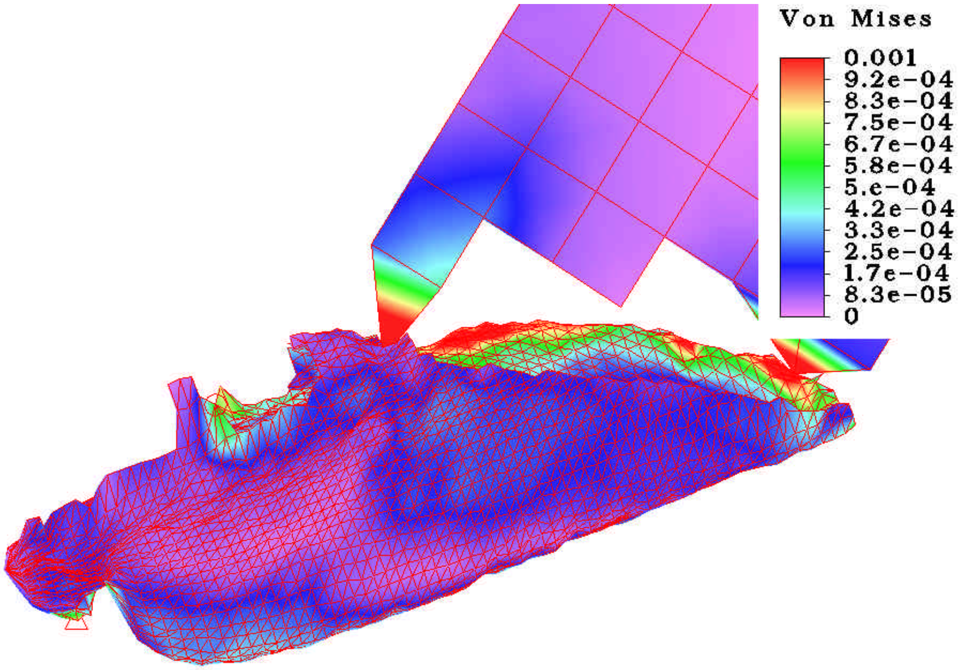

Figure 4.5 Peg Leg Flex, Posterior, Von Mises Strain

Figures 4.4 and 4.5 give an indication of the distribution of strain across the trochanter. Strain concentrations are evident at the point-contact joint locations. Saint-Venant's principle states that the manner of force application is relevant only in the localized area of force application [Popov, 1990]. Real materials have an averaging effect that quickly dissipates the strain concentrations caused by point applications of loads. In terms of the sensilla locations, only group 4 is close enough to a condyle to be effected by the loading conditions. The condyles are relatively small and hard (sclerotized) and are not entirely mis-modeled as point loads. It may be argued that the group 4 sensilla are located where they are in order to be receptive to the high strains in the area of the condyles.

Some general observations can be made looking at the qualitative data presented in figures 4.4 and 4.5. The strain present on the anterior half is larger in magnitude and creates steeper strain gradients than that of the posterior side. This is of course due to the trochanteral-femoral joint being affixed to the anterior half. The posterior half does have large fields of significant strain. These strains are transferred in part by the dorsal bridge-like structure connecting the anterior and posterior halves. The bridge is relatively thick and shows a high degree of sclerotization suggesting a hard structure. The bridge also forms into the vertical ridges on both sides of the trochanter which makes it part of the main force carrying structure.

The anterior half shows lines of high strain along the ventral and dorsal surfaces as would be expected in a simplified beam bending model. The ventral trochanter is quite thin and therefore is susceptible to large deformations. Conversely, the dorsal trochanter is much thicker yet still sees appreciable strains. It is carrying large loads transmitted from the coxal condyles and the upper femoral condyle that are in a line across the dorsal edge. From an engineering design point of view, the fact that the structures are sized such that strains are relatively constant over the exoskeleton is significant. This suggests an efficient design where the maximum strength is achieved for the minimum weight. Exoskeletons in general have long been described as excellent structural members because of their high moments of inertia [Currey, 1967], but even at this level of structural detail, the elegant designs are evident.

The vertical ridges are visible on both halves of the strain representation. Their effect is not clear from this analysis and will be discussed more in Section 4.5. Finally, it can be seen that the ring of group 3, while in a high strain area, is not in a significantly higher state of strain than the surrounding cuticle.

To quantify the results, the numerical strain values were collected at the location of each of the four groups of sensilla. Three nodes that represent the sensilla locations were chosen for each group and are shown in figures 3.8 and 3.9. Each recorded value was the normal strain perpendicular to the long-axis of the sensilla (short-axis strain) as listed in table 4.1. The strains are graphed in figure 4.6 with the red horizontal dashes representing the mean of the three points for each group. The normal convention is adhered to: positive strains are tensions and negative strains are compressions. Negative values will, therefore, tend to excite the sensilla.

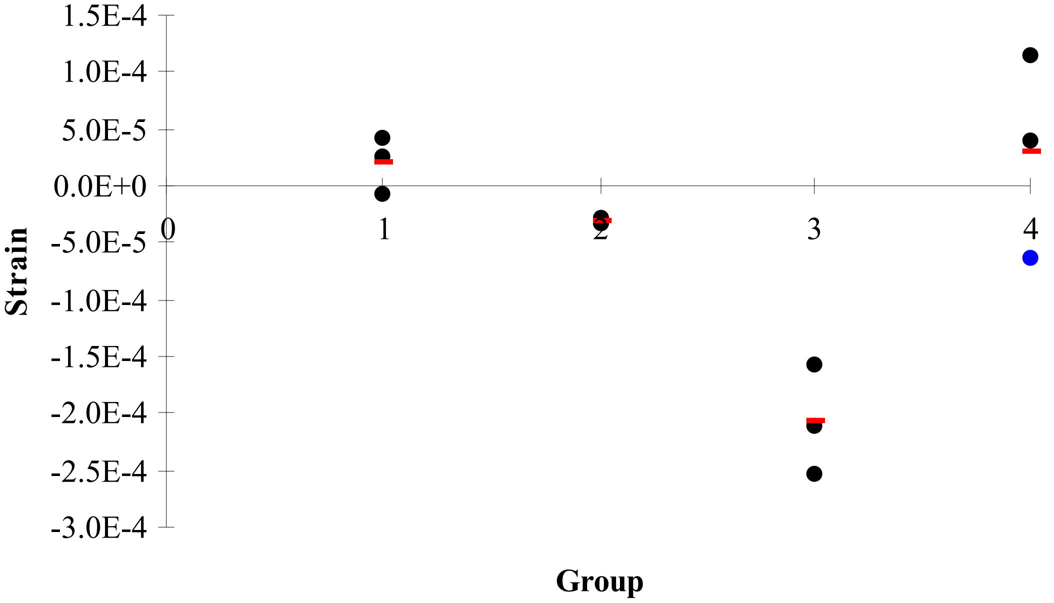

Figure 4.6 Peg Leg Flex, Short-Axis Strain

It is clear from the data that group 3 is showing a consistent high level of strain. This value is well above the threshold that would cause the sensilla to respond as will be discussed below. This strong response from group 3 is in agreement with the biological experiment. One interesting analysis of the short-axis strain is to compare it with the principle strains which can also be output from Algor. These values are displayed in table 4.2 for the four groups. The orientation of group 3 with respect to the minimum principle strain can now be calculated by the strain transformation

$$\theta = \frac{1}{2}\cos^{-1}\left(\pm\frac{2\varepsilon_{xx} - \varepsilon_{11} - \varepsilon_{22}}{\varepsilon_{11} - \varepsilon_{22}}\right) = 32.6^{\circ} \qquad 4.7$$

where the minus sign is used for the minimum principle strain. Thus, group 3 is oriented within 33° of the maximum compressive strain and is well suited for detecting the forces applied during flexion.

Group	Mean Short-Axis Strain, ε_{xx} (με)	Maximum Principle Strain, ε_{11} (με)	Minimum Principle Strain, ε_{22} (με)
1	20	72	-60
2	-31	65	-99
3	-207	100	-334
4	31	246	-484

Table 4.2 Strain Values, Peg Leg Flex

Group 1 has a negligible compressive strain component and a statistical significance test for a normal distribution of a random variable states that the mean is zero with 85% confidence. Group 2 has an average compressive strain of -31 με and a significance test shows only 25% probability that the mean is zero. This raises the question of whether the sensilla of group 2 will be stimulated by this level of strain. The only researchers who have attempted to make direct sensitivity measurements of cuticle strain detectors have been Barth and Blickhan [1984]. They were able to attach strain gages directly to the locations of slit sensilla of spiders during normal walking. They found, for slow walking, strains between 10 and 20 με and maximum values during fast walking of 120 $\mu\varepsilon$. While these values may suggest the validity of the finite element results, comparison between the slit sensilla and campaniform sensilla is not a direct correlation because of differences in morphology.

Another way to approach the sensitivity question is to use the threshold values from the biological experiments. Measurements of group 3 during forced flexion show no response until approximately 50 mN has been applied to the distal femur. The same force applied to the finite element model yields an average stress of 52 $\mu\varepsilon$ at group 3. This is the corresponding threshold strain for the model. Although there is evidence of variations in sensitivity between sensilla, it is unlikely that the 30 $\mu\varepsilon$ at group 2 will be a significant factor in the response compared to group 3. However, some response from group 2 cannot be ruled out.

Group 4 shows a wide variation in strain among the three points. This is in part due to the proximity of the sensilla to the femoral condyle. As a result, there is a large strain gradient across the region. In addition, group 4 tends to be more dispersed relative to group 3, for example, which is tightly packed in the ring. In fact, there is one large sensilla, shown in figure 4.7 as L1, found on all *Periplaneta americana* that is separate from the group. This sensilla is in the approximate location of the first sampling point (marked blue) which is the only one that shows compression for the forced flexion. It is conceivable that only a fraction of the group is recruited during specific types of loading such as the peg leg experiments.

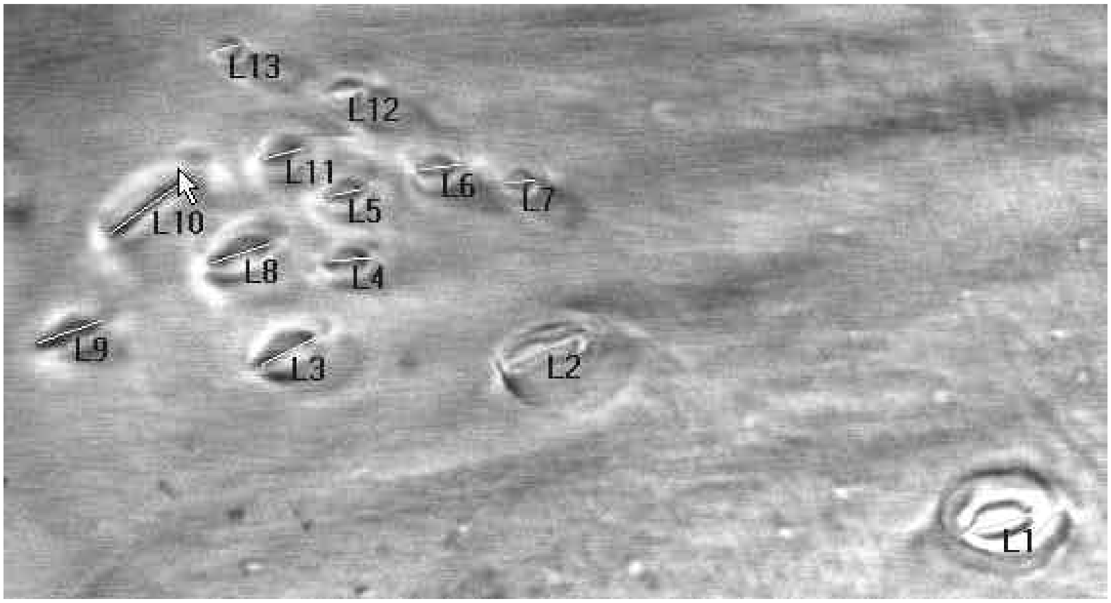

Figure 4.7 Group 4 Sensilla Showing Single Separate Sensilla (L1)

It is again insightful to examine the average orientation of the sensilla relative to the principle strains. The strain transformation can be easily visualized using a Mohr's circle representation [Popov, 1990], shown in figure 4.7. The maximum and minimum principle strains are ε_{11} and ε_{22} respectively. The point $(\varepsilon_{xx},\gamma_{xy})$ is the two dimensional state of strain for group 4. The angle 2θ, where θ is given by equation 4.7, shows the current orientation with respect to the principle strains. Also plotted are two points, ε_a and ε_b which are the strain transformations at +/- one standard deviation away from the mean sensilla orientation. Graphically, it can be seen that the variation is significant. Within one standard deviation, the normal strain can vary from approximately -80 μ to 126 μ. The variation within group 4 has not reconfirmed yet, but this is another indication that the individual sensilla may be responding differently to the same force application.

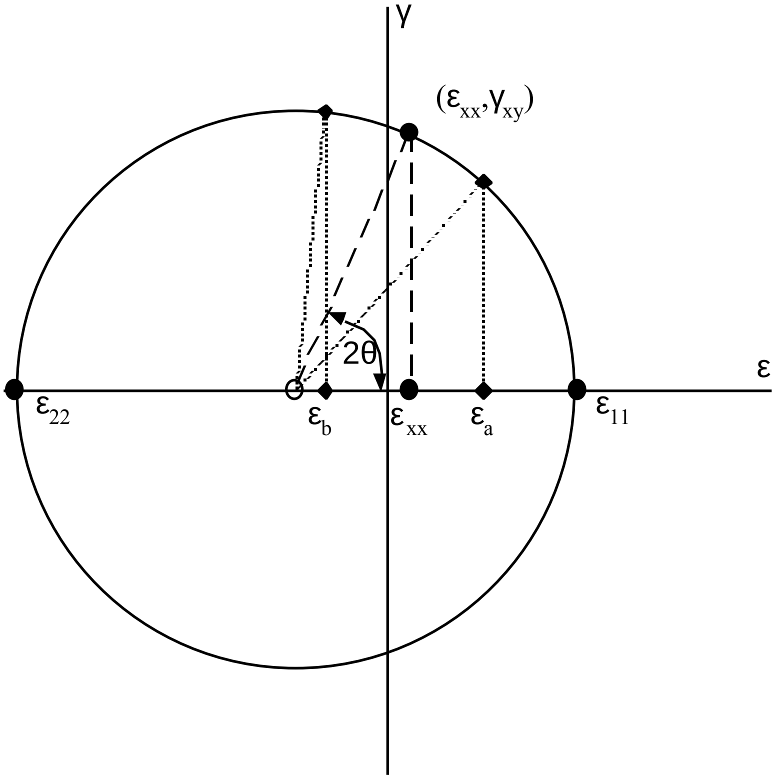

Figure 4.8 Mohr's Circle for Group 4

It has been shown that for the forced flexion experiment, the finite element model agrees with the physiological experiment. Group 3 has strong short-axis strains which will yield sensilla responses. Groups 1 and 2 have negligible strains and will not contribute significantly to the response. Group 4 has shown variability on several accounts: its orientation, its sensitivity to poisson's ratio, its spacial arrangement, and its response to forced flexion. It has shown evidence of strain values that would stimulate sensilla in accordance with the biological experiment.

4.4.3.2 Forced Extension

The forced extension experiment was performed in exactly the same manner as the forced flexion experiment except that the direction of the force was opposite. Not surprisingly, the results are the opposite of the flexion experiment. Figure 4.9 has the plot of the short-axis strain values. Groups 2 and 3 are clearly in tension and therefore not contributing to the response. Group 1 has an average compressive strain, although the value is statistically zero. Group 4 is again interesting in that the two points closest to the femoral condyle show compression, while the point representing the large separate sensilla (marked in blue) is in tension.

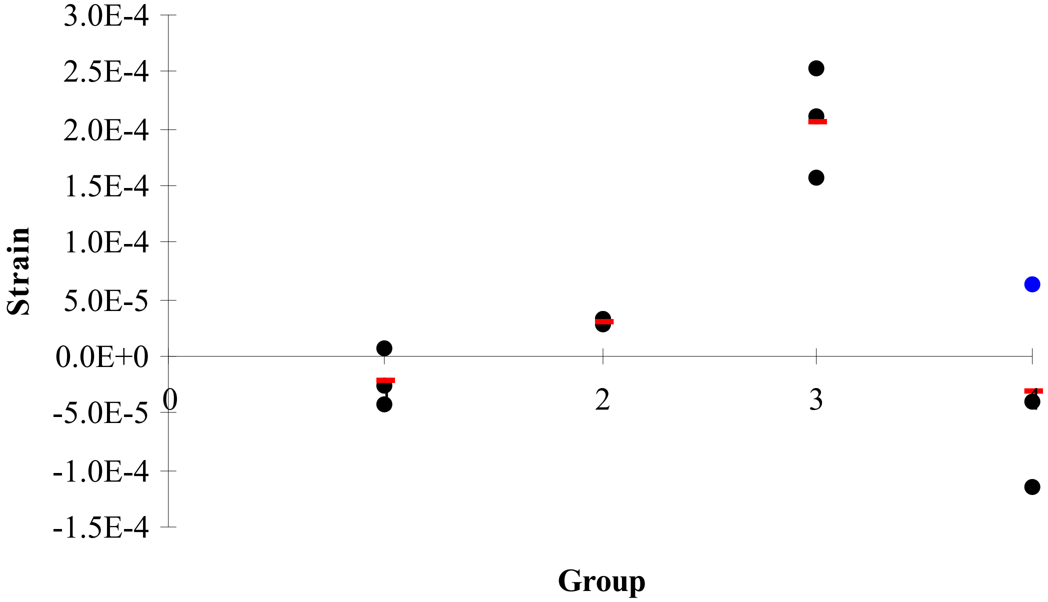

Figure 4.9 Peg Leg Extend, Short-Axis Strain

4.4.3.3 Extensor Muscles

Another way to examine the peg leg experiments is to take the opposite approach to applying forces. Instead of having the leg withstand an externally applied load as with the forced flexion experiment, the forces can be internally generated by contractions of muscle groups. Using this approach, the two major muscle groups of the coxa, the flexors and the extensors, were examined.

Pearson and Iles [1970] performed detailed studies of the coxal musculature while performing research on inhibitory neurons. They were able to stimulate muscles of group 177 (d, e, e'), see figure 2.3, in a simulated walking motion. The tension of the muscle group was recorded simultaneously. With inhibition, the muscle tension varied between approximately 0.0 and 2.0 gmf (19.6 mN). Group 177 is innervated by a slow motoneuron and is the major contributor to trochanteral extension during walking.

The finite element experiment was performed using a force of 19.6 mN applied at the proximal end on the trochanter centerline. The actual 177 apodeme attaches to a membrane that spans the posterior end. The same argument is made here for point loading as was described for the point-connection joint condyles; the inaccuracies introduced for point loading are local and will not have an effect at the sensilla locations. The orientation of the force was placed from observations of the muscle group and was approximately 75° from the x-axis along the center line of the trochanter. Hinge-like boundary conditions were applied at the coxal condyles as described in the forced flexion experiment, and the distal femur was completely fixed simulating obstructed movement.

The Von Mises strain distribution is shown in figures 4.10 and 4.11. The distributions are similar to those seen in the forced flexion with some differences. In fact, this is a similar experiment to forced flexion except that the location of the forces and the fixed nodes are reversed. A simple static analysis shows that 19.6 N applied to the proximal trochanter is equivalent to 2.18 mN at the distal femur which is close to the 1.96 mN applied during forced flexion. The posterior end is under relatively more stress which is not surprising since that is where the muscle force is applied. There are again high strain areas at the condyles, but there is a vertical band of lower strain which extends through group 3 and down the vertical ridge.

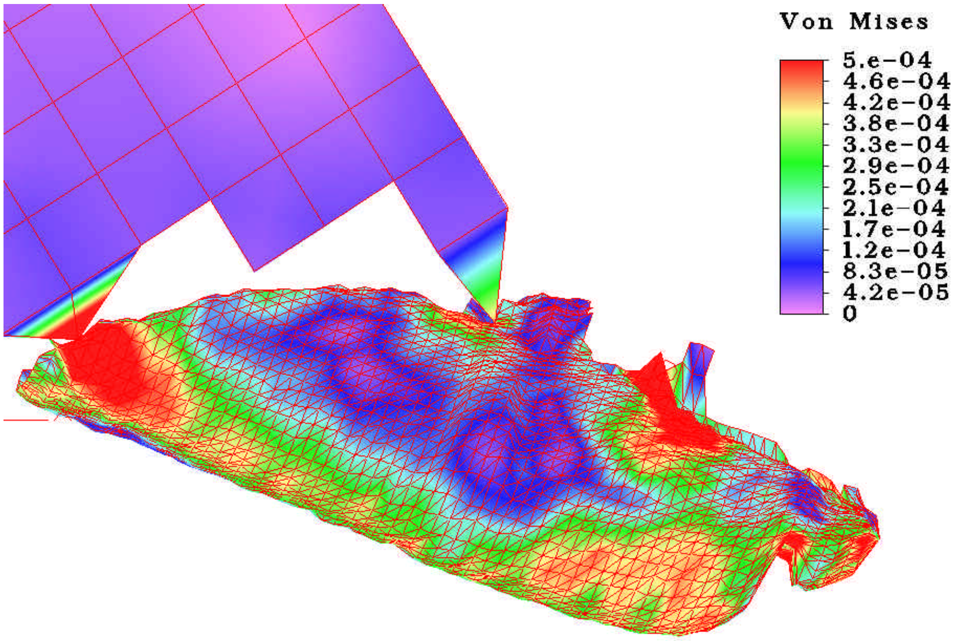

Figure 4.10 Peg Leg 177, Anterior, Von Mises Strain

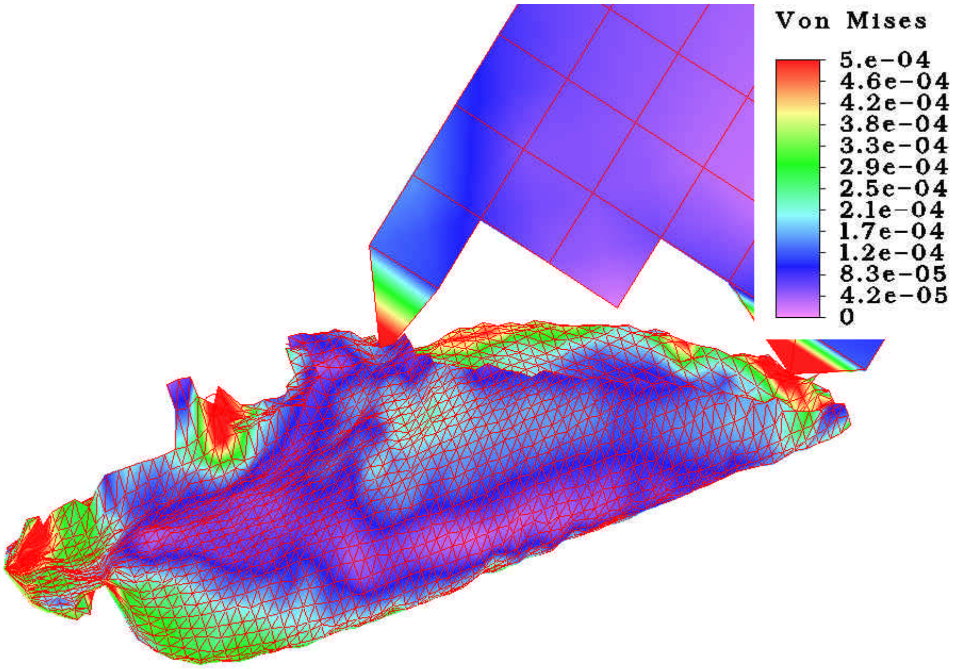

Figure 4.11 Peg Leg 177, Posterior, Von Mises Strain

The numerical results are graphed in figure 4.12. There are consistent compressive strains for group 3. This is unexpected based on the Von Mises strain of figure 4.10 where there is a region of low strain around the ring of group 3. Closer inspection shows that the minimum principle strain is -68.9 με and the average short-axis strain is -64.5 με. Using equation 4.7 it can be determined that the short-axis is only 14.5° from the principle axis. Thus, group 3 is well oriented to detect strains from the contraction of muscle group 177.

Group 4 is more consistently under compression compared to the forced flexion experiment, although there is still significant variance. Again the point at the location of the single separate sensilla stands out. It is likely that at least this sensilla will show a response to application of muscle group 177. Groups 1 and 2 are reminiscent of the forced flexion experiment with the strain magnitude less than that of groups 3 and 4. The groups will probably not contribute to a response.

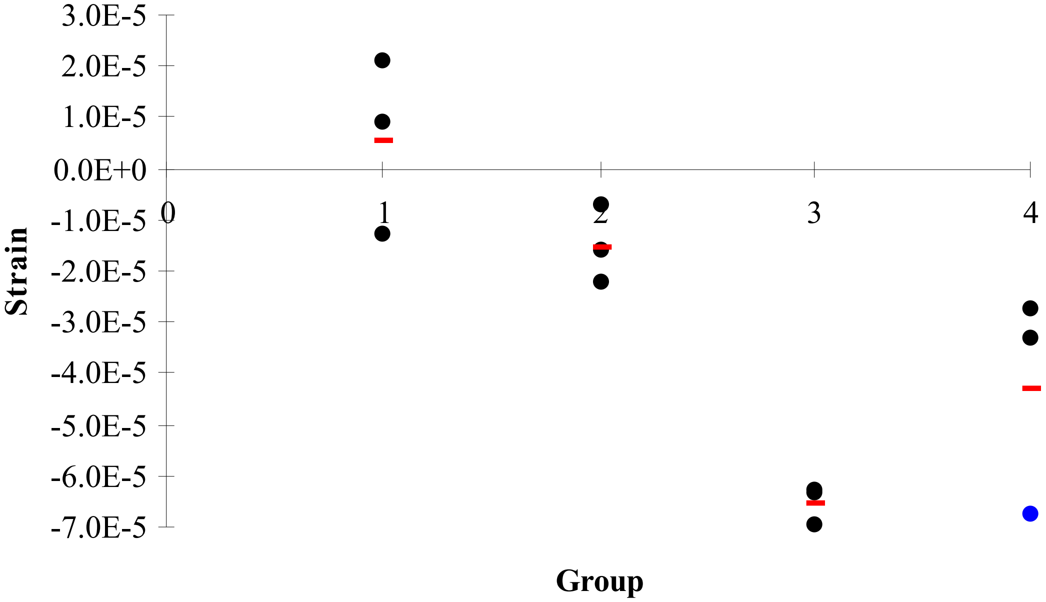

Figure 4.12 Peg Leg 177 Short-Axis Strain

4.4.3.4 Flexor Muscles

The complementary muscles to the 177 extensor group are the trochanteral flexors whose major slow-innervated, tension-creating muscle group is 181. Similarly to the group 177 experiment, it was desired to examine the effect of flexor contraction against a restrained distal leg. The boundary conditions were the same as the extensor experiment—coxal condyles acting as a hinge joint and the distal femur completely fixed. Pearson and Bergman [1969] in their study on inhibitory motoneurons recorded a force of 12.7 mN from muscle group 182. Although the value was not directly recorded from group 181, it is a good estimate for tension of group 181, and since the analysis is linear, all values can be scaled based on the magnitude of the muscle force. 12.7 mN was applied to the model on the dorsal bridge-like structure connecting the anterior and posterior halves. The force acted on the center line of the trochanter at an angle of 75° with the x-axis. Figures 4.13 and 4.14 have the Von Mises strain distribution.

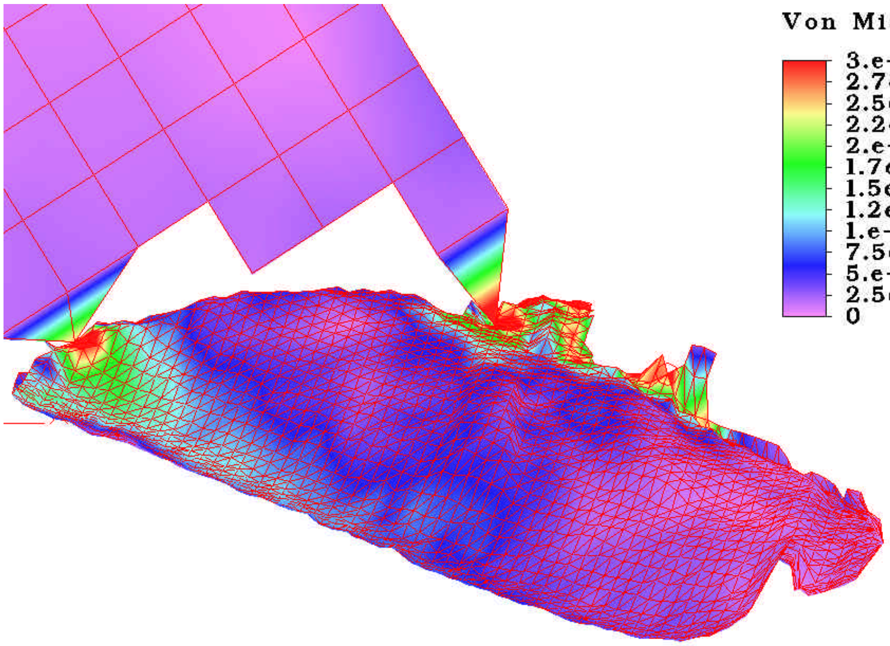

Figure 4.13 Peg Leg 181, Anterior, Von Mises Strain

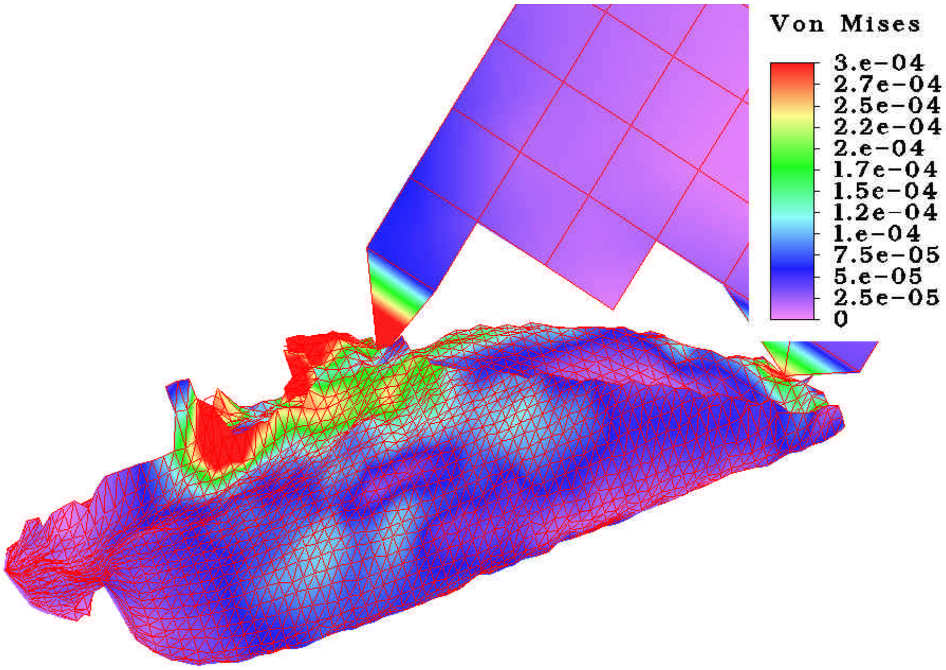

Figure 4.14 Peg Leg 181, Posterior, Von Mises Strain

The strains overall are lower than those seen in the previous experiments. This is due to the smaller force exerted by the flexor muscles and the smaller moment arm between the condyles and the muscle insertion. It is worth noting that the posterior side is seeing significant strains compared to the anterior which in general has not been true of the previous experiments. This is likely a testament to the load transmitting capacity of the bridge structure. The numerical data is charted in figure 4.15.

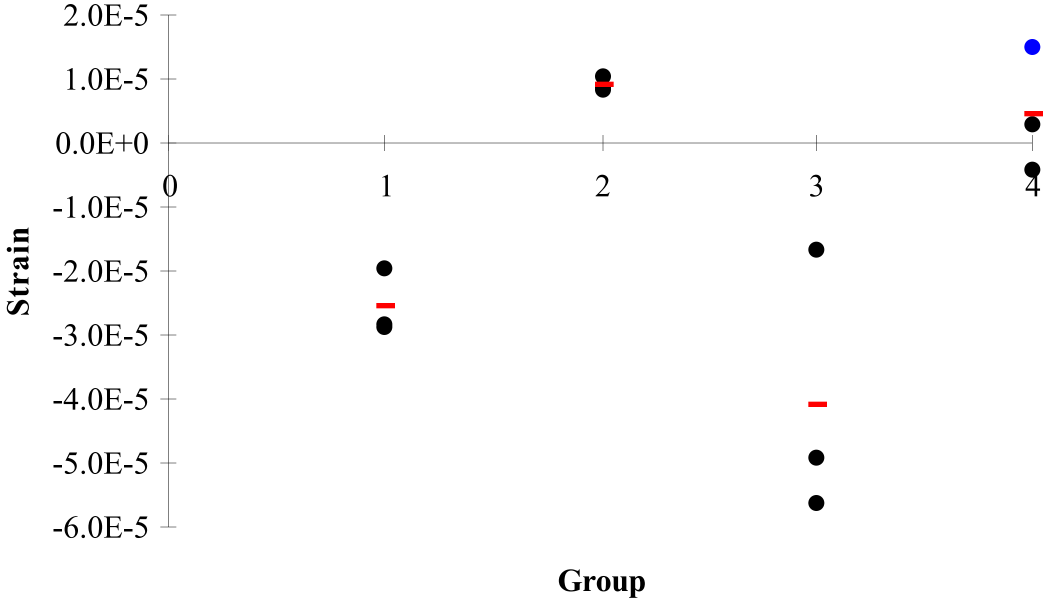

Figure 4.15 Peg Leg 181, Short-Axis Strain

Group 1 is prominent as was predicted by the Von Mises distribution. The minimum principle strain there is -34.6 με and the average short-axis strain is -25.6 με which corresponds to an orientation of 17.3° off the principle axis. Group 1's orientation is much closer to the principle axis here than in any of the previous experiments suggesting that group 1 may have a proprioceptive functionality for the flexor muscles. Delcomyn [1991] has shown in the stick insect that by pulling on the flexor apodeme strong responses from group 1 can be elicited.

Group 3 is again showing strong short-axis compressions. It's location between the two condyles and the insertion for the flexor muscles is proving to be beneficial for strain measurement. Groups 2 and 4 in this case are showing small tensions.

4.4.3.5 Muscle Lines of Action

The line of action of muscle groups 177 and 181 can vary greatly during normal walking depending on the coxal-trochanteral joint angle. The lines of action in the above tests were near the flexion limit (75° relative to the positive x-axis). During the stroke phase, these angles decrease until the leg is fully extended. To examine this effect, the flexion and extension muscle experiments were run with the muscle forces at angles of 40° and 20° relative to the positive x-axis. Data for group 3 for the 177 muscle experiment and group 1 for the 181 muscle experiment was collected and is shown in table 4.3.

Experiment	Average Strain at 75° (με)	Average Strain at 40° (με)	Average Strain at 20° (με)
177, Group 3	-64.5	-3.6	-1.4
181, Group 1	-25.6	-0.4	0.7

Table 4.3 Strain for Various Muscle Lines of Action

The strains drop off quickly as the angle decreases. The effect can be visualized as a shortening of the moment arm between the muscle insertion and the coxal condyle as the angle decreases. At 20° the muscles are performing mostly axial loading instead of the large bending moments that are created at larger angles.

The lessening of the strain seen by group 3 during extension must be taken in context of the experiment. In a walking animal, as the metathoracic leg extends, the bending moment created on the trochanter by the ground reaction forces will increase. This external stimulus will counter the weakened stimulus provided by the extensor muscles. In fact, Pearson [1972] has shown that the sensilla reflexively excite the extensor muscles and show an increase in activity as the leg extends. Thus, it is likely that the sensilla are more highly excited at the end of the stroke due to the increased moment arm on which the ground reaction forces act.

4.4.3.6 Summary

A summary of the responses for the peg leg experiments is shown in table 4.4. Full responses are shown as X and partial are shown as x. The orientation of the sensilla of group 1 (17.3° off the minimum principle axis) suggest that they are tuned to detect forces from the flexor muscles, and indeed, the strains transmitted to the posterior side during flexion prove this to be an appropriate location.

Group	Flexion	Extension	177	181
1				X
2	x			
3	X		X	X
4	x	x	X	

Table 4.4 Summary of Group Responses

Group 2 was not strongly strained by any of the experiments shown here. Schmitz [1993] performed a series of experiments on the stick insect where he studied the reflex effects of campaniform sensilla. He found that, by performing experiments similar to the forced flexion experiment except with posterior and anterior bending of the femur, retractor and protractor reflexes would result. Although he was not able to conclusively determine which sensilla groups caused which reflex, he did show that the reflexes were modulated by the anterior and posterior sensilla groups, analogous to groups 1 and 2. In addition, Delcomyn [1991] showed that posterior bending of the stick insect trochanter produces responses from group 1. Similar anterior/posterior bending experiments were performed on this model, but the results were inconclusive. The difficulty lies with the modeling of the trochanteral-femoral joint which allows the femur a small amount of rotation posteriorly. The movement of this joint is not yet well understood and therefore difficult to model accurately. The experiments presented here cause moments transverse to the axis of this joint effectively removing that degree of freedom from the model.

Group 3 has shown to be quite responsive to various loading conditions. Its location, near the coxal and femoral condyles and the insertion of muscle group 181, place it in an excellent position to detect strains from a variety of sources. It may be a “catch-all” strain detector that with input from the other sensilla groups help differentiate its signal. Delcomyn [1991] has hypothesized that an even higher level of integration, for example with motoneurons, may be occurring within the nervous system to identify the stimulation of the sensilla. Group 3's orientation to the minimum principle strain for the 177 muscle experiment was the closest measured for any group (14.5°) under these experiments. This orientation and the orientation of group 1 during flexor contraction make a strong argument for proprioceptive control of the coxal musculature modulated by the responses of groups 1 and 3.

Group 4's functionality is difficult to describe because of it's variable results. It may be more accurate to consider group 4 as two subgroups—one consisting of the single large sensilla and the rest of the sensilla in the other. The resolution of this model is not such that detailed analysis of individual sensilla can be made, but there is clear indication that strains are variable across the group. The measurement point at the location of the single sensilla consistently showed a disparate response to the other two points, even showing compression while the others showed tension.

4.5 Smooth Model

In an attempt to examine the effects of the structure of the trochanter on the strains, another model was created that essentially removes the thickness variations present on the interior surface. In this model, the outer surface, which is relatively smooth, was sampled and a constant thickness value was assigned to the elements. The result is the shell of a trochanter without the ridges and thickness variations of the inner surface.

The peg leg flexion experiment was performed on the model in exactly the same manner as the real trochanter model. The Von Mises strain distribution is shown in figure 4.16. A direct comparison between the strain distribution but not actual values can be made between figures 4.16 and 4.4. A striking difference can be seen in the area of the central vertical ridge. The smooth model has a vertical region of high stress where the ridge would be in the normal model. Figure 4.5 has no such strain concentration and has a remarkably smooth distribution across the dorsal edge. It is clear that internal structures are designed to help distribute forces and keep localized stress concentrations to a minimum.

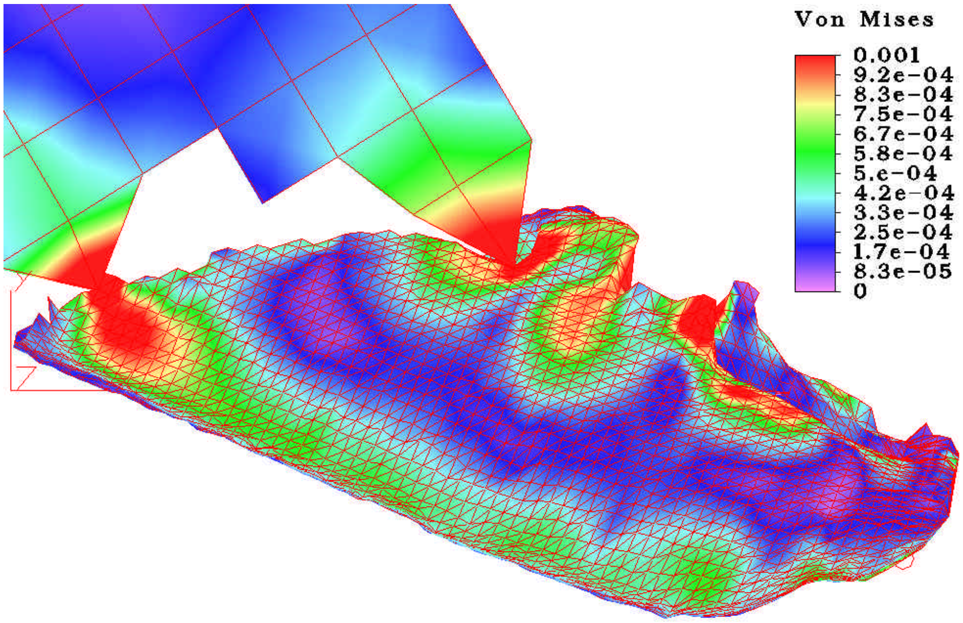

Figure 4.16 Smooth Model, Peg Leg Flex, Anterior, Von Mises Strain

4.6 Stress Distribution

It is of some interest to look at cuticle stresses from a design point of view. Many of the cuticle material studies give ultimate tensile stress values that can be used as a strength criteria. The stress distributions are the same as those shown for strain since the constitutive relation is assumed linear. The peak stress value for the peg leg flexion experiment was 5.7 MPa and for the peg leg 177 experiment was 4.9 MPa. Jensen and Weis-Fogh [1962] found an ultimate tensile strength of 94 MPa for the hind tibia of the locust, but this stress occurs at catastrophic failure which is an uncommon occurrence in biological materials [Alexander, 1981]. Their stress versus strain plots show nearly linear behavior below about 30 MPa. The experiments in this work have not been worst case situations. For example, Full and Tu [1991] have found peak vertical reaction forces for running *Periplaneta americana* of 1.9 times the weight of the cockroach or about 15.5 mN—nearly 8 times what was applied during the forced flexion experiment. This force applied to the peg leg flexion experiment would cause maximum stresses of about 45 MPa. Based on this evidence, it is likely that the animal normally operates in the linear range of its cuticle properties—well below the material's ultimate strength. The campaniform sensilla are the means for maintaining this limit [Zill et al., 1981].

Chapter V

Conclusions

5.1 Methods

The confocal microscopy method has developed to be more troublesome than first envisioned. One issue is that multiple image series were required to cover the entire area of the trochanter. These series then required alignment by hand. A similar issue was that the trochanter had to be bisected to obtain accurate images through the entire specimen. Focusing through the upper shell to get to the lower half caused shadows and general degradation of the images. Bisecting the trochanter requires that it be reassembled later in software—again by hand. In addition, the reassembling process, done in Voxblast, requires resampling the dataset which tends to have an averaging or smoothing effect. In the future, concatenation code created to reassemble the polygon files after being split for ease of handling will be used to reassemble the halves of the trochanter thereby eliminating the resampling step, but the halves will still have to be manually aligned. A final issue of tedium with the model formulation is that the individual images required editing to remove extraneous material such as individual hairs or muscle tissues.

The end product of the confocal model is the polygon file which itself is far from perfect for analysis. In retrospect, it may have been easier to start with the raw images and create code to formulate the finite element model directly from the raster data. Although this would require a significant increase in programming effort, it would be a more straight-forward approach. The current meshing code is in its fifth version mostly due to complexities of dealing with the polygon files.

The resulting finite element model is geometrically very accurate due to a great extent to the care taken in formulating the confocal model and use of control criteria at every step. The mesh accuracy is well within the assumptions of the model. The complexity of accurately modeling cuticle is still a difficult issue. The elastic modulus is relatively well researched, but there is evidence that there are variations even within the trochanter. Modifying the model to account for the fluctuations is a trivial matter, but accurately describing the biology is not. Poisson's ratio has not been accurately determined and at this point must be considered on a case-by-case basis. The results have shown that many of the nonlinearities reported for cuticle properties can be neglected because they are beyond the normal operation range of the material; the linear analysis used here is an accurate approach.

The model has provided a platform by which numerous experiments and hypotheses can be tested quickly and easily. As with any finite element model, the application of boundary and loading conditions needs to be carefully considered. It has proved to be a less than trivial matter to transfer biological observations to the mathematics of finite element modeling. The use of parametric studies has been valuable in the assessment of the variations present in biological systems.

5.2 Discussion

This work disproved the hypothesis that the structure of the trochanter and the placement of the sensilla were “designed” to create stress concentrations to increase the sensitivity of the sensilla. The opposite has turned out to be true. It is not a good design practice to create weak links in a structure by introducing stress concentrators. A design should take into account high stress areas by reinforcing the structure in these areas through the use of material or form enhancements. The trochanter does both. Force bearing structures tend to be highly sclerotized (hard) and thick truss-like members help distribute stresses. The strain distributions show consistent levels across the dorsal and ventral edges despite irregular geometries and thickness variations. This is directly evident in comparison of the full-featured model with the smooth model.

In addition, the stress analysis has shown that under normal conditions such as the experiments in this thesis, cockroach cuticle operates well below the material’s strength limits. Most structural cuticles are linearly elastic for small deformations [Hepburn and Joffe, 1976], and *Periplaneta americana* typically functions within this range.

The position and orientation of the sensilla are more likely optimized for differentiation of strains from various sources than for simply maximizing the strain values. Groups 1 and 3 provide evidence of this. While group 3 is located such that it detects a variety of sources of strain, based on the experiments here, it is well oriented to pick up the forces caused by the trochanteral extensors. Conversely, group 1 is relatively unresponsive to most force applications, but detects the flexor contractions well. These two groups form a complementary pair that sense the activities of the antagonistic muscle pair that is most important to cockroach walking.

It has also been evident that the response of a single group of sensilla alone is not enough to characterize the source of the stimulus. Group 3 responds to a variety of loading conditions, but to make use of this information the insect's nervous system must integrate information from other sources to differentiate the response. Table 4.4 summarizes the responses to the experiments, and in each case, the total response of the four groups is different. The combined reaction is enough to identify the source of cuticular strain for the experiments shown, but as Delcomyn [1991] suggests the nervous system may also interpret other information such as motoneuron activity.

Comparison of the finite element model with complementary biological experiments has been a fruitful approach. It has provided a check for the model as in the case of Zill's forced flexion experiment, but it has also added definition to existing biological theories and raised new questions. The activity of group 3 during contraction of the extensor muscles agrees nicely with Pearson's [1972] model for positive feedback control of this muscle group. The identification of group 1 as a complementary receptor for the flexors proposes that this group may perform a similar function. Schmitz's [1993] and Delcomyn's [1991] work demonstrate some of the characteristics of the system in the stick insect, but the issue is unsettled.

5.3 Future Work

The finite element method has not previously been used in the study of arthropod exoskeletons. It has offered a level of detail and accuracy that has previously been impossible with other structural models. Because it is a new approach, the value of the method for arthropod research remains to be seen, but several applications relative to this work are apparent. For instance, it would be a relatively simple step to increase the resolution of the method and examine the detailed structure of a single sensillum. Several questions might be answered about the fine structure of campaniform sensillum such as how the cap actually deforms and interacts with the dendrite and how the sensilla respond to different types of strain. The approach could also be used on a larger scale to create an accurate model of the entire leg. The model could be supplied with ground reaction and joint data from simulation [Nelson, 1994] or experiments [Full, 1995] and would be able to predict strains actually seen during normal insect behavior.

References

Alexander, R., (1981), "Factors of Safety in the Structure of Animals," *Sci. Prog.*, vol. 67, pp. 109-130.

Alsop, D. W., (1978), "Comparative Analysis of the Intrinsic Leg Musculature of the American Cockroach, *Periplaneta americana,*" *Journal of Morphology*, vol. 158, pp. 199-242.

Barth, F. G., (1973), "Microfiber Reinforcement of an Arthropod Cuticle," *Z. Zellforsch. mikrosk. Anat.,* vol. 144, pp. 409-433.

Barth, F. G., and Blickhan, R., (1984), "Mechanoreception," In *Biology of the Integument Vol 1*, Springer-Verlag: Berlin, pp. 554-582.

Beer, R. D., (1990), "Intelligence as Adaptive Behavior: An Experiment in Computational Neuroethology," Cambridge, MA: Academic Press.

Beer, R. D., Chiel, H. J., Quinn, R. D., Espenschied, K. S., and Larsson, P., (1992) "A distributed Neural Network Architecture for Hexapod Robot Locomotion," *Neural Computation*, vol. 4, pp. 356-365.

Bohnenberger, J., (1981), "Matched Transfer Characteristics of Single Units in a Compound Slit Sense Organ," *Journal of Comparative Physiology*, vol. 142, pp. 391-402.

Buddenbrock, W. von, (1921), "Der Rhythmus der Schreitbewegungen der Stabheuschrecke Dyxippus," *Biol. Zentralbl*, vol. 41, pp. 41-48.

Buddenbrock, W. von, (1953), *Vergleichende Physiologie. Band II Nervenphysiologie*, Basel: Verlag Birkhauser.

Callister, W. D. Jr., (1991), *Material Science and Engineering an Introduction,* New York: John Wiley and Sons, Inc.

Chiel, H. J., Beer, R. D., Quinn, R. D., and Espenschied, K. S., (1992), "Robustness of a distributed Neural Network Controller for Locomotion in a Hexapod Robot," *IEEE Transactions on Robotics and Automation*, vol. 8, pp. 293-303.

Cruse, H., (1990), "What Mechanisms Coordinate Leg Movement in Walking Arthropods?" *Trends in Neural Science*, vol. 13, pp. 15-21.

Currey, J. D., (1967), "The Failure of Exoskeletons and Endoskeletons," *Journal of Morphology*, vol. 123, pp. 1-16.

Delcomyn, F., (1971), "The Locomotion of the Cockroach *Periplaneta americana,*" *Journal of Experimental Biology,* vol. 54, pp. 443-452.

Delcomyn, F., (1991), "Activity and Directional Sensitivity of Leg Campaniform Sensilla in a Stick Insect," *Journal of Comparative Physiology*, vol. 168, pp. 113-119.

Delcomyn, F., Nelson, M. E., and Cocatre-Zilgien, J. H., (1996), "Sense Organs of Insect Legs and the Selection of Sensors for Agile Walking Robots," *International Journal of Robotics Research*, vol. 15, pp. 113-127.

DiCaprio, R. A., Ridgel, A. L., Frazier, S. F., and Zill, S. N., (1997), "Measuring forces in an insect leg," *Society of Neuroscience Abstracts*, vol. 23, part 1, p. 480.

Espenschied, K. S., Quinn, R. D., Beer, R. D., and Chiel, H. J., (1996), "Biologically Based Distributed Control and Local Reflexes Improve Rough Terrain Locomotion in a Hexapod Robot," *Robotics and Automation*, vol. 18, pp. 59-64.

Espenschied, K. S., Quinn, R. D., Chiel, H. J., and Beer, R. D., (1993), "Leg Coordination Mechanisms in the Stick Insect Applied to Hexapod Robot Locomotion," *Adaptive Behavior*, vol. 1, no. 4, pp. 455-468.

Fagan, M. J., (1992), *Finite Element Analysis, Theory and Practice*, Essex, UK: Longman Scientific and Technical.

Flannigan, W. C., Nelson, G. M., and Quinn, R. D., (1998), "Locomotion Controller for a Crab-like Robot," *Proceedings of the 1997 Conference on Robotics and*

Automation.

Full, R. J., and Ahn, A. N., (1995), "Static Forces and Moments Generated in the Insect Leg: Comparison of a Three-dimensional Musculo-skeletal Computer Model with Experimental Measurements," *Journal of Experimental Biology*, vol. 198, pp. 1285-1298.

Full, R. J., Blickhan, R., and Ting, L. H., (1991), "Leg Design in Hexapedal Runners," *Journal of Experimental Biology*, vol. 158, pp. 369-390.

Full, R. J., and Tu, M. S., (1991), "Mechanics of a Rapid Running Insect: Two-, Four-, and Six-Legged Locomotion," *Journal of Experimental Biology*, vol. 156, pp. 215-231.

Hepburn, H. R., and Chandler, H. D., (1976), "Mechanical Hysteresis of Insect Cuticles," *Journal of Insect Physiology*, vol. 22, pp. 221-226.

Hepburn, H. R., and Joffe, I., (1976), "On the Material Properties of Insect Exoskeleton," In *The Insect Integument*, Amsterdam: Elsevier, pp. 207-235.

Hepburn, H. R., and Roberts, D. C., (1975), "Stiffness and Tanning of Sclerites," *Journal of Insect Physiology*, vol. 21, pp. 1741-1746.

Hibbler, R. C., (1993), *Statics and Mechanics of Material*, New York: Macmillan.

Hillerton, J. E., (1984), "Cuticle: Mechanical Properties," In *Biology of the Integument*, Berlin: Springer-Verlag, pp. 626-637.

Hughes, G. M., (1952), "The Coordination of Insect Movements. II. The Effect of Limb Amputation and the Cutting of Commissures in the Cockroach (*Blatta orientalis)*," *Journal of Experimental Biology*, vol. 29, pp. 267-284.

Hughes, G. M., (1957), "The Coordination of Insect Movements. I. The Walking Movements of Insects," *Journal of Experimental Biology*, vol. 34, pp. 306-333.

Jensen, M., and Weis-Fogh, T., (1962), "Biology and Physics of Locust Flight V. Strength and Elasticity of Locust Cuticle," *Philosophical Transactions of the Royal Society of London*, no. 721, vol. 245, pp. 137-169.

Joffe, I., and Hepburn, H. R., (1973), “Observations on Regenerated Chitin Films,” *Journal of Material Science*, vol. 8, pp. 1751-1754.

Katz, S. L., and Gosline, J. M., (1994), “Scaling Modulus as a Degree of Freedom in the Design of Locust Legs,” *Journal of Experimental Biology*, vol. 187, pp. 207-223.

Ker, R. F., (1977), “Some Structural and Mechanical Properties of Locust and Beetle Cuticle,” Thesis, Oxford University, UK.

Moran, D. T., Chapman, K. M., and Ellis, R. A., (1971). “The Fine Structure of Cockroach Campaniform Sensilla,” *Journal of Cellular Biology*, vol. 48, pp. 155-173.

Nelson, G. M., (1994), “Dynamic Simulation of an Insect-like Hexapod,” M. S. Thesis, Case Western Reserve University.

Nelson, G. M., Quinn, R. D., Bachmann, R. J., Flannigan, W. C., Ritzmann, R. E., and Watson, J. T., (1997), “Design and Simulation of a Cockroach-Like Hexapod Robot,” *Proceedings of the 1997 IEEE International Conference on Robotics and Automation.*

Nelson, G. M., and Quinn, R. D., (1998), “Posture Control of a Cockroach-like Robot,” *Proceedings of the 1998 IEEE International Conference on Robotics and Automation.*

Pearson, K. G., (1972), “Central Programming and Reflex Control of Walking in the Cockroach,” *Journal of Experimental Biology*, vol. 56, pp. 173-193.

Pearson, K. G., and Bergman, S. J., (1969) “Common Inhibitory Motoneurons in Insects,” *Journal of Experimental Biology*, vol. 50, pp. 445-471.

Pearson, K. G., Fourtner, C. R., and Wong, R. K., (1973), “Nervous Control of Walking in the Cockroach,” In R. B. Stein, K. G. Pearson, R. S. Smith, and J. B. Redford (eds.): *Control of Posture and Locomotion*, New York: Plenum Press, pp. 495-514.

Pearson, K. G., and Iles, J. F., (1970), "Discharge Patterns of Coxal Levator and Depressor Motoneurons of the Cockroach, *Periplaneta americana*," *Journal of Experimental Biology*, vol. 52, pp. 139-165.

Popov, E. P., (1990), *Engineering Mechanics of Solids*, Englewood Clifs, NJ: Prentice-Hall, Inc.

Porter, M. A., (1994), *FEA Step by Step with Algor*, Leawood, KS: Dynamic Analysis.

Pringle, J. W. S., (1938a), "Proprioception in Insects. II. A New Type of Mechanoreceptor from the Palps of the Cockroach," *Journal of Experimental Biology*, vol. 15, pp. 101-113.

Pringle, J. W. S., (1938b), "Proprioception in Insects. II. The Action of the Campaniform Sensilla of the Legs," *Journal of Experimental Biology*, vol. 15, pp. 114-131.

Pringle, J. W. S., (1940), "The Reflex Mechanism of the Insect Leg," *Journal of Experimental Biology*, vol. 17, pp. 8-17.

Quinn, R. D., and Espenschied, K. S., (1993) "Control of a Hexapod Robot Using a Biologically-Inspired Neural Network," In R. D. Beer, R. E. Ritzmann, and T. McKenna (eds.), *Biological Neural Networks in Invertebrate Neuroethology and Robotics*, New York: Academic Press, pp. 365-381.

Rao, S. S., (1995), *Mechanical Vibrations*, Reading, MA: Addison-Wesley.

Reitveld, J., (1995), "Viscoelasticity", In C. A. Dostal et al. (Eds.), *Engineered Materials Handbook, Engineering Plastics*, Metals Park: ASM International, vol. 2, pp. 412-422.

Ritzmann, R. E., (1993), "The Neural Organization of the Cockroach Escape and its role in Context Dependent Orientation," In R. D. Beer, R. E. Ritzmann, and T. McKenna (eds.), *Biological Neural Networks in Invertebrate Neuroethology and Robotics*, New York: Academic Press.

Schildt, H., (1990), *C The Complete Reference*, Berkeley, CA: McGraw-Hill.

Schmitz, J., (1993), “Load-Compensating Reactions in the Proximal Leg Joints of Stick Insects During Standing and Walking,” *Journal of Experimental Biology*, vol. 183, pp. 15-33.

Sherrington, C. S., (1906), *The integrative Action of the Nervous System,* New York: Charles Scribner’s Sons.

Snodgrass, R. E., (1965), *A Textbook of Arthropod Anatomy*, New York: Hafner Pub. Co.

Spinola, S. M., and Chapman, K. M., (1975), “Proprioceptive Indentation of the Campaniform Sensilla of Cockroach Legs,” *Journal of Comparative Physiology*, vol. 96, pp. 257-272.

Spyrakos, C. C., (1994), *Finite Element Modeling in Engineering Practice*, West Virginia University Press.

Watson, J. T., and Ritzmann, R. E., (1994), “The Escape Response versus the Quiescent Response of the American Cockroach: Behavioural Choice Mediated by Physiological State,” *Animal Behavior*, vol. 48, pp. 476-478.

Watson, J. T., and Ritzmann, R. E., (1998a), “Leg Kinematics and Muscle Activity During Treadmill Running in the Cockroach, *Blaberus discoidalis*: I. Slow Running,” *Journal of Comparative Physiology*, vol. 182, pp. 11-22.

Watson, J. T., and Ritzmann, R. E., (1998b), “Leg Kinematics and Muscle Activity During Treadmill Running in the Cockroach, *Blaberus discoidalis*: II. Fast Running,” *Journal of Comparative Physiology*, vol. 182, pp. 23-33.

Wendler, G., (1964), “Laufen and stehen der stabheuschrecke Carausius morosus: Sinnesborstenfelder in den Beingelenken als Glieder von Regelkreisen,” *Z. Vergl. Physiol.*, vol. 48, pp. 198-250.

Wendler, G., (1966), “The Coordination of Walking Movements in Arthropods,” In *Nervous and Hormonal Mechanisms of Integration,* Symp. Soc. Exp. Biol., vol. 20, pp. 229-249.

Zienkiewicz, O. C., and Taylor, R. L., (1994), *The Finite Element Method, Volume 1*

Basic Formulation and Linear Problems, Berkshire, UK: McGraw-Hill.

Zill, S. N., (1982), “Suppression of Reflex Postural Tonus: A Role of Peripheral Inhibition in Insects,” *Science*, vol. 216, pp. 751-753.

Zill, S. N., (1990), “Mechanoreceptors: Exteroceptors and Proprioceptors,” In Huber, I., Masler, E. P., and Rao, B. R. (eds.): *Cockroaches as Models for Neurobiology: Applications in Biomedical Research*, vol. II, Boca Raton, FL: CRC Press, pp. 247-267.

Zill, S. N., and Moran, D. T., (1981a), “The Exoskeleton and Insect Proprioception. I. Responses of Tibial Campaniform Sensilla to External and Muscle-Generated Forces in the American Cockroach, *Periplaneta americana*,” *Journal of Experimental Biology,* vol. 91, pp. 1-24.

Zill, S. N., and Moran, D. T., (1981b), “The Exoskeleton and Insect Proprioception. III. Activity of Tibial Campaniform Sensilla During Walking in the American Cockroach, *Periplaneta americana*,” *Journal of Experimental Biology,* vol. 94, pp. 57-75.

Zill, S. N., Moran, D. T., and Varela, F. G., (1981), “The Exoskeleton and Insect Proprioception. II. Reflex effects of Tibial Campaniform Sensilla in the American Cockroach, *Periplaneta americana*,” *Journal of Experimental Biology,* vol. 94, pp. 43-55.

Zill, S. N., and Seyfarth, E. A., (1996), “Exoskeletal Sensors for Walking,” *Scientific American*, July, pp. 71-74.

Appendix A

Mesh Generator Code

A.1 File Poly5.h

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <conio.h>
#include <math.h>

#define inFile      "..\\concatenate\\concfile.pol"//input file
#define dxfFile     "d:\\Program Files (x86)\\AutoCAD R14\\out.dxf"//dxf file
#define outFile     "out.txt"  //debug out file
#define colorFile   "color.txt"//color thickness value file
#define numTheta    72//number of elements along x-axis
#define numX        72//number of elements around theta direction
#define maxGood     20000  //maximum number of points selected in a sampling range
#define xRange      0.015  //inital sample range size - x coordinate
#define PI          3.14159265359//PI
#define minNumPoints20//minimum number of sample points
#define maxNumIter  8 //maximum number of sample interations
#define minRadius   0.05 //below this radius elements degenerate to point on x-axis
#define maxColor    200  //highest AutoCAD color number to be used (maxColor/10 colors)
#define scaleFactor 0.006388 //mm per pixel scaling factor

#define  centerSampling  //which surface to sample
#define  faces  //what to write to dxf file

struct vertex
{
  float  x;
  float  y;
  float  z;
  float  r;
  float  theta;
  float  thickness;
};

struct tree
{
  struct vertexcoord;
  struct tree*xLeft;
  struct tree*xRight;
};

typedef struct vertex  VERTEX;
typedef struct tree    TREE;

TREE *parse(FILE *ifp, long numObj, TREE *root, float *minX, float *maxX);
TREE *addToTree(VERTEX point, TREE *r, TREE *root);
void printTree(TREE *root);
void addThetas(TREE *r, TREE *root);
void addTheta(float theta, TREE *r, TREE *root, TREE *R);
TREE *searchTree(TREE *root, VERTEX point);
void findPoints(TREE *root, VERTEX *goodPoint, int *numGood, VERTEX max, VERTEX min);
void sample(TREE *root, float minX, float maxX, VERTEX ***realPoint, float *maxThickness);
void printPoints(VERTEX *goodPoint, int numGood, FILE *ofp);
void removeBadTheta(VERTEX *goodPoint, int *numGoodX, int numGood, VERTEX max, VERTEX min);
```

```
VERTEX findThickness(VERTEX *goodPoint, int numGood, float *maxThickness);
int compare(const VERTEX *first, const VERTEX *second);
void dxf(VERTEX ***realPoint, float maxThickness, float minX, float maxX);
void findMaxMin(TREE *root, float *minX, float *maxX);
void rotate3D (VERTEX *point, double x, double y, double z);
```

A.2 File Poly5.c

```
#include "poly5.h"

/******************************************************************************
*main control
******************************************************************************/
int main(void)
{
  FILE   *ifp, *color;
  long   numVert;
  TREE   *root = NULL;
  float  minX = (float)1000.0, maxX = (float)-1000.0;
  VERTEX ***realPoint;
  float  maxThickness = (float)0.0;
  int    i;


  root = parse(ifp, numVert, root, &minX, &maxX);

  realPoint = malloc((numX + 1) * sizeof(VERTEX*));

  sample(root, minX, maxX, realPoint, &maxThickness);

  dxf(realPoint, maxThickness, minX, maxX);

  color = fopen(colorFile, "w");
  for(i = 1; i <= maxColor / 10 + 1; i++)
    fprintf(color, "color: %d\tthickness: %f\n", i * 10, (i - .5) * maxThickness / (maxColor / 10));
  fclose(color);

  free(root);
  free(realPoint);

  return 0;
}

/******************************************************************************
*opens poly file
*reads in vertex
*adds data to binary tree
******************************************************************************/
TREE *parse(FILE *ifp, long numVert, TREE *root, float *minX, float *maxX)
{
  char   line[80], junk[80];
  int    i;
  VERTEX point;

  if((ifp = fopen(inFile, "r")) == NULL)
  {
    printf("Can't open input file\n");
    exit(EXIT_FAILURE);
  }

  while(fgets(line, 7, ifp) != NULL)
  {
    if(strcmp(line, "VERTEX") == 0)
      break;
    fgets(junk, 80, ifp);
  }
  if(line == NULL)
  {
    printf("Unable to find vertex data\n");
    exit(EXIT_FAILURE);
  }

  fgetc(ifp);
  fscanf(ifp, "%ld\n", &numVert);

  printf("%d verticies found\n", numVert);
  printf("adding x values\n");

  for(i = 0; i < numVert; i++)
  {
    fscanf(ifp, "%f %f %f", &point.y, &point.x, &point.z);  //swap x and y
    fgets(junk, 80, ifp);

    point.x *= (float)scaleFactor;
    point.y *= (float)scaleFactor;
    point.z *= (float)scaleFactor;

//  rotate3D(&point, 0.0, PI / 2., 0.0);

    point.r = (float)sqrt(point.z * point.z + point.y * point.y);
    point.theta = (float)atan2(point.y, point.z);

    if(root != NULL)
      addToTree(point, root, root);
    else
      root = addToTree(point, root, root);

    if(i % 10000 == 0)
      printf("%d\n", i);
  }

  findMaxMin(root, minX, maxX);

  fclose(ifp);

  return root;
}

/******************************************************************************
*controls sampling process
*chooses sampling range
*selects point to represent sampling area
******************************************************************************/
void sample(TREE *root, float minX, float maxX, VERTEX ***realPoint, float *maxThickness)
{
  VERTEX   goodPoint[maxGood];
  VERTEX   low, high;
  int      numGoodX = 0, numGood = 0;
  int      i, j;
  float    testTheta, testX;
  int      iterations;
  FILE     *ofp;
  float    rSum;
  float    thetaRange;

  printf("sampling\n");

  if((ofp = fopen(outFile, "w")) == NULL)
  {
    printf("Can't open output file\n");
    exit(EXIT_FAILURE);
  }

  for(i = 0; i <= numX; i++)
  {

    testX = (float)(i * (maxX - minX) / numX + minX + xRange);
    low.x = (float)(testX - xRange / 2.);
    high.x = (float)(testX + xRange / 2.);

    numGoodX = 0;

    findPoints(root, goodPoint, &numGoodX, high, low);

    if(numGoodX != 0)
    {
      rSum = (float)0.0;
      for(j = 0; j < numGoodX; j++)
        rSum += goodPoint[j].r;

      thetaRange = (float)xRange / (rSum / (float)numGoodX);

      printf("numGoodX: %d r avg: %.2f\n", numGoodX, rSum / numGoodX);

      realPoint[i] = malloc((numTheta + 1) * sizeof(VERTEX*));

      for(j = 0; j < numTheta; j++)
      {
        testTheta = (float)(j * 2. * PI / numTheta - PI);

        low.theta = (float)(testTheta - thetaRange / 2.);
        high.theta = (float)(testTheta + thetaRange / 2.);
        low.x = (float)(testX - xRange / 2.);
        high.x = (float)(testX + xRange / 2.);

        numGood = 0;

        removeBadTheta(goodPoint, &numGood, numGoodX, high, low);

        if(numGood > 0)
        {
          iterations = 0;

          while(numGood < minNumPoints && iterations < maxNumIter)
          {
            iterations++;

            numGood = 0;
            numGoodX = 0;

            low.x -= (float)(xRange / 10.);
            high.x += (float)(xRange / 10.);

            findPoints(root, goodPoint, &numGoodX, high, low);

            low.theta -= (float)(thetaRange / 10.);
            high.theta += (float)(thetaRange / 10.);

            removeBadTheta(goodPoint, &numGood, numGoodX, high, low);
          }

          if(numGood > 1)
          {
            realPoint[i][j] = malloc(sizeof(VERTEX));

            *realPoint[i][j] = findThickness(goodPoint, numGood, maxThickness);

  #ifdef innerSampling
            realPoint[i][j]->x = goodPoint[0].x;
            realPoint[i][j]->y = goodPoint[0].y;
            realPoint[i][j]->z = goodPoint[0].z;
            realPoint[i][j]->r = goodPoint[0].r;
            realPoint[i][j]->theta = goodPoint[0].theta;
  #endif

  #ifdef centerSampling
//          realPoint[i][j]->x = goodPoint[0].x;
//          realPoint[i][j]->y = goodPoint[0].y;
//          realPoint[i][j]->z = goodPoint[0].z;
```

```
            realPoint[i][j]->r = goodPoint[0].r;
            realPoint[i][j]->theta = goodPoint[0].theta;

            realPoint[i][j]->x = testX;
            realPoint[i][j]->y = (float)(realPoint[i][j]->r * sin(realPoint[i][j]->theta));
            realPoint[i][j]->z = (float)(realPoint[i][j]->r * cos(realPoint[i][j]->theta));

            realPoint[i][j]->y += (float)(realPoint[i][j]->thickness
              / 2.0 * cos(realPoint[i][j]->theta));
            realPoint[i][j]->z += (float)(realPoint[i][j]->thickness
              / 2.0 * sin(realPoint[i][j]->theta));
  #endif

  #ifdef outerSampling
//          realPoint[i][j]->x = goodPoint[numGood - 1].x;
//          realPoint[i][j]->y = goodPoint[numGood - 1].y;
//          realPoint[i][j]->z = goodPoint[numGood - 1].z;
            realPoint[i][j]->r = goodPoint[numGood - 1].r;
            realPoint[i][j]->theta = goodPoint[numGood - 1].theta;

            realPoint[i][j]->x = testX;
            realPoint[i][j]->y = (float)(realPoint[i][j]->r * sin(realPoint[i][j]->theta));
            realPoint[i][j]->z = (float)(realPoint[i][j]->r * cos(realPoint[i][j]->theta));

  #endif

            printf("numGood: %d  iter: %d\n", numGood, iterations);
          }

          else
            realPoint[i][j] = NULL;
        }

        else
          realPoint[i][j] = NULL;
      }
    }

    else
    {
      realPoint[i] = malloc(numTheta * sizeof(VERTEX*));

      for(j = 0; j < numTheta; j++)
        realPoint[i][j] = NULL;
    }

    realPoint[i][numTheta] = realPoint[i][0];
  }

  fclose(ofp);
}

/*********************************************************************
*adds vertex data to binary tree recusively
*********************************************************************/
TREE *addToTree(VERTEX point, TREE *r, TREE *root)
{
  if(r == NULL)
  {
    r = (TREE *)malloc(sizeof(TREE));

    r->coord.x = point.x;
    r->coord.y = point.y;
    r->coord.z = point.z;
    r->coord.r = point.r;
    r->coord.theta = point.theta;

    r->xLeft = NULL;
    r->xRight = NULL;

    if(root == NULL)
      return r;
    if(point.x < root->coord.x)
      root->xLeft = r;

    else
      root->xRight = r;

    return r;
  }

  if(point.x < r->coord.x)
    addToTree(point, r->xLeft, r);

  else
    addToTree(point, r->xRight, r);
}

/******************************************************************************
*calculates min and max values of tree
******************************************************************************/
void findMaxMin(TREE *root, float *minX, float *maxX)
{
  TREE *currentNode;

  currentNode = root;

  while(currentNode->xRight != NULL)
    currentNode = currentNode->xRight;

  *maxX = currentNode->coord.x;

  currentNode = root;

  while(currentNode->xLeft != NULL)
    currentNode = currentNode->xLeft;

  *minX = currentNode->coord.x;
}

/******************************************************************************
*prints out data tree recusively
******************************************************************************/
void printTree(TREE *root)
{
  if(root == NULL)
    return;

  printTree(root->xLeft);

  printf("x=%f y=%f z=%f\n", root->coord.x, root->coord.y, root->coord.z);

  printTree(root->xRight);
}

/******************************************************************************
*finds individual struct in tree
******************************************************************************/
TREE *searchTree(TREE *root, VERTEX point)
{
  while(root->coord.x != point.x)
  {
    if(point.x < root->coord.x)
      root = root->xLeft;

    else
      root = root->xRight;

    if(root == NULL)
      break;
  }

  return root;
```

```
}

/*******************************************************************************
*searches through tree to find all data in sampling range
*******************************************************************************/
void findPoints(TREE *root, VERTEX *goodPoint, int *numGood, VERTEX max, VERTEX min)
{
  if(root == NULL)
    return;

  if(root->coord.x <= max.x && root->coord.x >= min.x)
  {
    goodPoint[*numGood].x = root->coord.x;
    goodPoint[*numGood].y = root->coord.y;
    goodPoint[*numGood].z = root->coord.z;
    goodPoint[*numGood].r = root->coord.r;
    goodPoint[*numGood].theta = root->coord.theta;

    if(*numGood >= maxGood - 1)
    {
      printf("\ntoo many good points in range...quitting\n");
      return;
    }

    (*numGood)++;

    findPoints(root->xLeft, goodPoint, numGood, max, min);
    findPoints(root->xRight, goodPoint, numGood, max, min);
  }

  if(root->coord.x > max.x)
    findPoints(root->xLeft, goodPoint, numGood, max, min);

  else if(root->coord.x < min.x)
    findPoints(root->xRight, goodPoint, numGood, max, min);
}

/*******************************************************************************
*outputs points found in sampling range
*******************************************************************************/
void printPoints(VERTEX *goodPoint, int numGood, FILE *ofp)
{
  int i;

  for(i = 0; i < numGood; i++)
  {
    fprintf(ofp, "%f\t%f\t%f\n", goodPoint[i].x, goodPoint[i].y, goodPoint[i].z);
  }

  fprintf(ofp, "\n");

}

/*******************************************************************************
*removes sample points that are not with the theta test values
*******************************************************************************/
void removeBadTheta(VERTEX *goodPoint, int *numGood, int numGoodX, VERTEX max, VERTEX min)
{
  int i;

  for(i = 0; i < numGoodX; i++)
  {
    if(goodPoint[i].theta >= min.theta && goodPoint[i].theta <= max.theta)
    {
      goodPoint[*numGood] = goodPoint[i];

      (*numGood)++;
    }
  }
}

/*******************************************************************************
```

```
*sorts points in sample range
*finds largest difference between consecutive points
*calculates thickness based on average of inner and outer points
******************************************************************************/
VERTEX findThickness(VERTEX *goodPoint, int numGood, float *maxThickness)
{
  VERTEX   realPoint;
  int      i;
  float    maxDiff = (float)-1.0;
  float    diff;
  float    perc;
  int      diffLocation;
  float    lowSum = (float)0.0, highSum = (float)0.0;
  float    lowAvg, highAvg, thickness;

  qsort(goodPoint, numGood, sizeof(VERTEX), compare);

  for(i = 0; i < numGood - 1; i++)
  {
    diff = goodPoint[i + 1].r - goodPoint[i].r;

    if(diff >= maxDiff)
    {
      maxDiff = diff;
      diffLocation = i;
      perc = (float)i / numGood;
    }
  }

  for(i = 0; i <= diffLocation; i++)
    lowSum += goodPoint[i].r;

  for(i = diffLocation + 1; i < numGood; i++)
    highSum += goodPoint[i].r;

  lowAvg = lowSum / (diffLocation + 1);
  highAvg = highSum / (numGood - (diffLocation + 1));

  thickness = highAvg - lowAvg;

//printf("lowAvg: %.2f\thighAvg: %.2f\tthickness: %.2f\n", lowAvg, highAvg, thickness);
//printf("maxDiff: %.2f\tperc: %.2f\tnumGood: %d\n", maxDiff, perc, numGood);

  realPoint.thickness = thickness;

  if(thickness > *maxThickness)
    *maxThickness = thickness;

  return realPoint;
}

/******************************************************************************
*comparison function for qsort
******************************************************************************/
int compare(const VERTEX *first, const VERTEX *second)
{
  if(first->r < second->r)
    return -1;
  else if(first->r > second->r)
    return 1;
  else
    return 0;
}

/******************************************************************************
*creates elements
*writes dxf file
******************************************************************************/
void dxf(VERTEX ***realPoint, float maxThickness, float minX, float maxX)
{
  FILE     *dxf;
  long     handle;
  int         color;
  float       avgThickness;
  int         i, j;
  unsigned short  thisRadiusIsSmall, nextRadiusIsSmall;
  float       thisTestX, nextTestX;

  printf("creating dxf file\n");

  if((dxf = fopen(dxfFile, "w")) == NULL)
  {
    printf("Can't open dxf file\n");
    exit(EXIT_FAILURE);
  }

  fprintf(dxf, "  0\nSECTION\n  2\nENTITIES\n");

  for(i = 0; i < numX; i++)
  {
    thisRadiusIsSmall = 0;
    nextRadiusIsSmall = 0;

    for(j = 0; j < numTheta; j++)
    {
      if(realPoint[i][j] != NULL)
        if(realPoint[i][j]->r < minRadius)
        {
          thisRadiusIsSmall = 1;
          thisTestX = (float)(i * (maxX - minX) / numX + minX + xRange);
          break;
        }
    }

    for(j = 0; j < numTheta; j++)
    {
      if(realPoint[i + 1][j] != NULL)
        if(realPoint[i + 1][j]->r < minRadius)
        {
          nextRadiusIsSmall = 1;
          nextTestX = (float)((i + 1) * (maxX - minX) / numX + minX + xRange);
          break;
        }
    }

    for(j = 0; j < numTheta; j++)
    {

#ifdef points
      if(realPoint[i][j] != NULL)
      {
        color = ((int)((realPoint[i][j]->thickness / maxThickness) * maxColor) / 10 + 1) * 10;

        fprintf(dxf, "  0\nPOINT\n  5\n%i\n100\nAcDbEntity\n  8\n0\n 62\n%i\n100\nAcDbPoint\n",
handle++, color);
        fprintf(dxf, " 10\n%f\n 20\n%f\n 30\n%f\n", realPoint[i][j]->x, realPoint[i][j]->y,
realPoint[i][j]->z);
      }
#endif

#ifdef lines
      if(thisRadiusIsSmall && !nextRadiusIsSmall)
      {
        if(realPoint[i][j] != NULL)
        {
          if(realPoint[i + 1][j] != NULL)
          {
            color = ((int)((realPoint[i][j]->thickness / maxThickness) * maxColor) / 10 + 1) * 10;

            fprintf(dxf, "  0\nLINE\n  5\n%i\n100\nAcDbEntity\n  8\n0\n 62\n%i\n100\nAcDbLine\n",
handle++, color);
            fprintf(dxf, " 10\n%f\n 20\n%f\n 30\n%f\n", thisTestX, 0.0, 0.0);
            fprintf(dxf, " 11\n%f\n 21\n%f\n 31\n%f\n", realPoint[i + 1][j]->x, realPoint[i + 1][j]-
>y, realPoint[i + 1][j]->z);
```

```
          }
        }
      }

      else if(!thisRadiusIsSmall && nextRadiusIsSmall)
      {
        if(realPoint[i][j] != NULL)
        {
          if(realPoint[i][j + 1] != NULL)
          {
            avgThickness = (float)((realPoint[i][j]->thickness + realPoint[i][j + 1]->thickness) /
2.0);
            color = ((int)((avgThickness / maxThickness) * maxColor) / 10 + 1) * 10;

            fprintf(dxf, "  0\nLINE\n  5\n%i\n100\nAcDbEntity\n  8\n0\n 62\n%i\n100\nAcDbLine\n",
handle++, color);
            fprintf(dxf, " 10\n%f\n 20\n%f\n 30\n%f\n", realPoint[i][j]->x, realPoint[i][j]->y,
realPoint[i][j]->z);
            fprintf(dxf, " 11\n%f\n 21\n%f\n 31\n%f\n", realPoint[i][j + 1]->x, realPoint[i][j + 1]-
>y, realPoint[i][j + 1]->z);
          }
          if(realPoint[i + 1][j] != NULL)
          {
            color = ((int)((realPoint[i][j]->thickness / maxThickness) * maxColor) / 10 + 1) * 10;

            fprintf(dxf, "  0\nLINE\n  5\n%i\n100\nAcDbEntity\n  8\n0\n 62\n%i\n100\nAcDbLine\n",
handle++, color);
            fprintf(dxf, " 10\n%f\n 20\n%f\n 30\n%f\n", realPoint[i][j]->x, realPoint[i][j]->y,
realPoint[i][j]->z);
            fprintf(dxf, " 11\n%f\n 21\n%f\n 31\n%f\n", nextTestX, 0.0, 0.0);
          }
        }
      }

      else if(thisRadiusIsSmall && nextRadiusIsSmall)
      {
        if(realPoint[i][j] != NULL)
        {
          if(realPoint[i][j + 1] != NULL)
          {
            color = ((int)((realPoint[i][j]->thickness / maxThickness) * maxColor) / 10 + 1) * 10;

            fprintf(dxf, "  0\nLINE\n  5\n%i\n100\nAcDbEntity\n  8\n0\n 62\n%i\n100\nAcDbLine\n",
handle++, color);
            fprintf(dxf, " 10\n%f\n 20\n%f\n 30\n%f\n", thisTestX, 0.0, 0.0);
            fprintf(dxf, " 11\n%f\n 21\n%f\n 31\n%f\n", nextTestX, 0.0, 0.0);
          }
        }
      }

      else
      {
        if(realPoint[i][j] != NULL)
        {
          if(realPoint[i][j + 1] != NULL)
          {
            avgThickness = (float)((realPoint[i][j]->thickness + realPoint[i][j + 1]->thickness) /
2.0);
            color = ((int)((avgThickness / maxThickness) * maxColor) / 10 + 1) * 10;

            fprintf(dxf, "  0\nLINE\n  5\n%i\n100\nAcDbEntity\n  8\n0\n 62\n%i\n100\nAcDbLine\n",
handle++, color);
            fprintf(dxf, " 10\n%f\n 20\n%f\n 30\n%f\n", realPoint[i][j]->x, realPoint[i][j]->y,
realPoint[i][j]->z);
            fprintf(dxf, " 11\n%f\n 21\n%f\n 31\n%f\n", realPoint[i][j + 1]->x, realPoint[i][j + 1]-
>y, realPoint[i][j + 1]->z);
          }
          if(realPoint[i + 1][j] != NULL)
          {
            avgThickness = (float)((realPoint[i][j]->thickness + realPoint[i + 1][j]->thickness) /
2.0);
            color = ((int)((avgThickness / maxThickness) * maxColor) / 10 + 1) * 10;
```

```
             fprintf(dxf, "  0\nLINE\n  5\n%i\n100\nAcDbEntity\n  8\n0\n 62\n%i\n100\nAcDbLine\n",
handle++, color);
             fprintf(dxf, " 10\n%f\n 20\n%f\n 30\n%f\n", realPoint[i][j]->x, realPoint[i][j]->y,
realPoint[i][j]->z);
             fprintf(dxf, " 11\n%f\n 21\n%f\n 31\n%f\n", realPoint[i + 1][j]->x, realPoint[i + 1][j]-
>y, realPoint[i + 1][j]->z);
           }
           if(realPoint[i + 1][j + 1] != NULL)
           {
             avgThickness = (float)((realPoint[i][j]->thickness + realPoint[i + 1][j + 1]->thickness) /
2.0);
             color = ((int)((avgThickness / maxThickness) * maxColor) / 10 + 1) * 10;

             fprintf(dxf, "  0\nLINE\n  5\n%i\n100\nAcDbEntity\n  8\n0\n 62\n%i\n100\nAcDbLine\n",
handle++, color);
             fprintf(dxf, " 10\n%f\n 20\n%f\n 30\n%f\n", realPoint[i][j]->x, realPoint[i][j]->y,
realPoint[i][j]->z);
             fprintf(dxf, " 11\n%f\n 21\n%f\n 31\n%f\n", realPoint[i + 1][j + 1]->x, realPoint[i + 1][j
+ 1]->y, realPoint[i + 1][j + 1]->z);
           }
         }
       }
#endif

#ifdef faces
       if(realPoint[i][j] != NULL && realPoint[i + 1][j + 1] != NULL && realPoint[i + 1][j] != NULL)
       {
         avgThickness = (float)((realPoint[i][j]->thickness + realPoint[i + 1][j + 1]->thickness +
realPoint[i + 1][j]->thickness) / 3.0);
         color = ((int)((avgThickness / maxThickness) * maxColor) / 10 + 1) * 10;

         fprintf(dxf, "  0\n3DFACE\n  5\n%i\n100\nAcDbEntity\n  8\n0\n 62\n%i\n100\nAcDbFace\n",
handle++, color);
         if(thisRadiusIsSmall)
           fprintf(dxf, " 10\n%f\n 20\n%f\n 30\n%f\n", thisTestX, 0.0, 0.0);
         else
           fprintf(dxf, " 10\n%f\n 20\n%f\n 30\n%f\n", realPoint[i][j]->x, realPoint[i][j]->y,
realPoint[i][j]->z);
         if(nextRadiusIsSmall)
         {
           fprintf(dxf, " 11\n%f\n 21\n%f\n 31\n%f\n", nextTestX, 0.0, 0.0);
           fprintf(dxf, " 12\n%f\n 22\n%f\n 32\n%f\n", nextTestX, 0.0, 0.0);
           fprintf(dxf, " 13\n%f\n 23\n%f\n 33\n%f\n", nextTestX, 0.0, 0.0);
         }
         else
         {
           fprintf(dxf, " 11\n%f\n 21\n%f\n 31\n%f\n", realPoint[i + 1][j]->x, realPoint[i + 1][j]->y,
realPoint[i + 1][j]->z);
           fprintf(dxf, " 12\n%f\n 22\n%f\n 32\n%f\n", realPoint[i + 1][j + 1]->x, realPoint[i + 1][j +
1]->y, realPoint[i + 1][j + 1]->z);
           fprintf(dxf, " 13\n%f\n 23\n%f\n 33\n%f\n", realPoint[i + 1][j + 1]->x, realPoint[i + 1][j +
1]->y, realPoint[i + 1][j + 1]->z);
         }
       }
       if(realPoint[i][j] != NULL && realPoint[i + 1][j + 1] != NULL && realPoint[i][j + 1] != NULL)
       {
         avgThickness = (float)((realPoint[i][j]->thickness + realPoint[i + 1][j + 1]->thickness +
realPoint[i][j + 1]->thickness) / 3.0);
         color = ((int)((avgThickness / maxThickness) * maxColor) / 10 + 1) * 10;

         fprintf(dxf, "  0\n3DFACE\n  5\n%i\n100\nAcDbEntity\n  8\n0\n 62\n%i\n100\nAcDbFace\n",
handle++, color);
         if(thisRadiusIsSmall)
         {
           fprintf(dxf, " 10\n%f\n 20\n%f\n 30\n%f\n", thisTestX, 0.0, 0.0);
           fprintf(dxf, " 11\n%f\n 21\n%f\n 31\n%f\n", thisTestX, 0.0, 0.0);
         }
         else
         {
           fprintf(dxf, " 10\n%f\n 20\n%f\n 30\n%f\n", realPoint[i][j]->x, realPoint[i][j]->y,
realPoint[i][j]->z);
```

```
     fprintf(dxf, " 11\n%f\n 21\n%f\n 31\n%f\n", realPoint[i][j + 1]->x, realPoint[i][j + 1]->y,
realPoint[i][j + 1]->z);
    }
    if(nextRadiusIsSmall)
    {
     fprintf(dxf, " 12\n%f\n 22\n%f\n 32\n%f\n", nextTestX, 0.0, 0.0);
     fprintf(dxf, " 13\n%f\n 23\n%f\n 33\n%f\n", nextTestX, 0.0, 0.0);
    }
    else
    {
     fprintf(dxf, " 12\n%f\n 22\n%f\n 32\n%f\n", realPoint[i + 1][j + 1]->x, realPoint[i + 1][j +
1]->y, realPoint[i + 1][j + 1]->z);
     fprintf(dxf, " 13\n%f\n 23\n%f\n 33\n%f\n", realPoint[i + 1][j + 1]->x, realPoint[i + 1][j +
1]->y, realPoint[i + 1][j + 1]->z);
    }
   }
   if(realPoint[i][j] == NULL && realPoint[i + 1][j + 1] != NULL && realPoint[i + 1][j] != NULL &&
realPoint[i][j + 1] != NULL)
   {
    avgThickness = (float)((realPoint[i + 1][j + 1]->thickness + realPoint[i + 1][j]->thickness +
realPoint[i][j + 1]->thickness) / 3.0);
    color = ((int)((avgThickness / maxThickness) * maxColor) / 10 + 1) * 10;

    fprintf(dxf, "  0\n3DFACE\n  5\n%i\n100\nAcDbEntity\n  8\n0\n 62\n%i\n100\nAcDbFace\n",
handle++, color);
    if(thisRadiusIsSmall)
     fprintf(dxf, " 10\n%f\n 20\n%f\n 30\n%f\n", thisTestX, 0.0, 0.0);
    else
     fprintf(dxf, " 10\n%f\n 20\n%f\n 30\n%f\n", realPoint[i][j + 1]->x, realPoint[i][j + 1]->y,
realPoint[i][j + 1]->z);
    if(nextRadiusIsSmall)
    {
     fprintf(dxf, " 11\n%f\n 21\n%f\n 31\n%f\n", nextTestX, 0.0, 0.0);
     fprintf(dxf, " 12\n%f\n 22\n%f\n 32\n%f\n", nextTestX, 0.0, 0.0);
     fprintf(dxf, " 13\n%f\n 23\n%f\n 33\n%f\n", nextTestX, 0.0, 0.0);
    }
    else
    {
     fprintf(dxf, " 11\n%f\n 21\n%f\n 31\n%f\n", realPoint[i + 1][j]->x, realPoint[i + 1][j]->y,
realPoint[i + 1][j]->z);
     fprintf(dxf, " 12\n%f\n 22\n%f\n 32\n%f\n", realPoint[i + 1][j + 1]->x, realPoint[i + 1][j +
1]->y, realPoint[i + 1][j + 1]->z);
     fprintf(dxf, " 13\n%f\n 23\n%f\n 33\n%f\n", realPoint[i + 1][j + 1]->x, realPoint[i + 1][j +
1]->y, realPoint[i + 1][j + 1]->z);
    }
   }
   if(realPoint [i + 1][j + 1] == NULL && realPoint[i][j] != NULL && realPoint[i + 1][j] != NULL &&
realPoint[i][j + 1] != NULL)
   {
    avgThickness = (float)((realPoint[i][j]->thickness + realPoint[i][j + 1]->thickness +
realPoint[i + 1][j]->thickness) / 3.0);
    color = ((int)((avgThickness / maxThickness) * maxColor) / 10 + 1) * 10;

    fprintf(dxf, "  0\n3DFACE\n  5\n%i\n100\nAcDbEntity\n  8\n0\n 62\n%i\n100\nAcDbFace\n",
handle++, color);
    if(thisRadiusIsSmall)
    {
     fprintf(dxf, " 10\n%f\n 20\n%f\n 30\n%f\n", thisTestX, 0.0, 0.0);
     fprintf(dxf, " 11\n%f\n 21\n%f\n 31\n%f\n", thisTestX, 0.0, 0.0);
    }
    else
    {
     fprintf(dxf, " 10\n%f\n 20\n%f\n 30\n%f\n", realPoint[i][j]->x, realPoint[i][j]->y,
realPoint[i][j]->z);
     fprintf(dxf, " 11\n%f\n 21\n%f\n 31\n%f\n", realPoint[i][j + 1]->x, realPoint[i][j + 1]->y,
realPoint[i][j + 1]->z);
    }
    if(nextRadiusIsSmall)
    {
     fprintf(dxf, " 12\n%f\n 22\n%f\n 32\n%f\n", nextTestX, 0.0, 0.0);
     fprintf(dxf, " 13\n%f\n 23\n%f\n 33\n%f\n", nextTestX, 0.0, 0.0);
    }
```

```
        else
        {
          fprintf(dxf, " 12\n%f\n 22\n%f\n 32\n%f\n", realPoint[i + 1][j]->x, realPoint[i + 1][j]->y,
realPoint[i + 1][j]->z);
          fprintf(dxf, " 13\n%f\n 23\n%f\n 33\n%f\n", realPoint[i + 1][j]->x, realPoint[i + 1][j]->y,
realPoint[i + 1][j]->z);
        }
      }
#endif
    }
  }

  fprintf(dxf, "  0\nENDSEC\n  0\nEOF");

  fclose(dxf);
}

/******************************************************************************
*performs rotation transformations on data
******************************************************************************/
void rotate3D (VERTEX *point, double x, double y, double z)
{
  float intermediate;

  if(x != 0.0)
  {
    intermediate = (float)(cos(x) * point->y - sin(x) * point->z);
    point->z = (float)(sin(x) * point->y + cos(x) * point->z);
    point->y = intermediate;
  }

  if(y != 0.0)
  {
    intermediate = (float)(cos(y) * point->x + sin(y) * point->z);
    point->z = (float)(-sin(y) * point->x + cos(y) * point->z);
    point->x = intermediate;
  }

  if(z != 0.0)
  {
    intermediate = (float)(cos(z) * point->x - sin(z) * point->y);
    point->y = (float)(sin(z) * point->x + cos(z) * point->y);
    point->x = intermediate;
  }
}
```

Index

Algor analysis software 38
Apodemes 14
Artificial insect project 4
AutoCAD 39
Binary tree 46
Biologicallly Inspired Research Program 3
Biorobotics Lab 3
Boundary conditions 38
Campaniform sensilla 16
Central ring 51
Chitin 21
Chordotonal organs 15
Condyle 12
Confocal microscopy 26
Convergence 54
Coxa 12
Diagonal gait 60
Endocuticle 21
Epicuticle 21
Exocuticle 21
Exteroceptors 15
Femur 12
Finite element method 32
Hair plates 15
Hair sensilla 15
Image series 28
K2T inc. 7
Load compensation 19
Mapping 42
Material properties 23
Mechanoreceptors 14
Mesh density 37
Meshing 37
Meshing tool 43

Mesothoracic 11
Metathoracic 11
Modulus of elasticity 24
Mohr’s circle 73
Montages 28
Multipolar receptors 15
Negative feedback 19
Peg leg experiments 60
Periplaneta americana 1
Poisson’s ratio 24
Polygon file 41
Proprioceptors 15
Protein matrix 21
Prothoracic 11
Range fractionation 20
Ritzmann Lab 3
Robot I 4
Robot II 5
Robot III 6
Saint-Venant’s principle 68
Sclerotization 23
Shape function 34
Shear modulus of elasticity 24
Slit sensilla 71
Solid model 37
Stiffness matrix 34
Strain transformation 70
Tanning 23
Tarsus 12
Tensile tests 22
Thorax 11
Threshold strain 72
Tibia 12
Tripod gait 60
Trochanter 12
Trochanteral extensors 14
Trochanteral flexors 14
Ultimate tensile stress 89

Vertical ridge 51
Viscoelasticity 23
Von Mises strain 66
Voxblast software 28
Zill Lab 3