



Modeling the non-linear response of mixed culture biofilm structures to turbulent flow  
by Brett William Towler

A dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of  
Philosophy in Engineering  
Montana State University  
© Copyright by Brett William Towler (2004)

Abstract:

Microbial biofouling of wetted interfaces can negatively impact the hydrodynamic performance of pressurized conduits. These impacts are due, in part, to the material properties of biofilm, yet few studies have examined this polymeric substance in the context of a constitutive relation. The goal of this research was two-fold; 1) to determine a suitable constitutive model for a mixed-culture biofilm and 2) use this material model in a numerical simulation to evaluate biofilm mechanical behavior in response to varying hydrodynamic conditions.

Creep tests revealed that these biofilms may be classified as viscoelastic fluids. Furthermore, results indicated the presence of viscous, time-dependent and instantaneous components to the biofilm compliance functions. A regression analysis ( $r^2 = 0.8819$ ) supported the treatment of these samples as linear viscoelastic fluids within the stress range of 0.1 Pa to 0.5 Pa. A specific linear viscoelastic constitutive equation was then determined by fitting experimental results to analytical solutions using an optimization algorithm. It was found that the Burger material model closely approximated the behavior of all samples. A numerical fluid-structure interface model was then developed and employed in a parametric study to investigate biofilm behavior. The effect of the Burger material parameters, mean flow velocity and biofilm size were examined. Simulations showed that weaker or softer biofilms (characterized by lower elastic moduli) were highly susceptible to lift forces. Additionally, polar diagrams were generated by plotting the coefficients of drag versus lift. The plots suggested that in the first few seconds after loading, the deformation paths taken by hemispherical biofilms are largely insensitive to specific material coefficients. Moreover, the diagrams illustrated that the effects of biofilm strength, size and channel velocity on displacement were predictable. These relationships may lead to the development of a simple, yet accurate method for predicting the hydrodynamic forces acting on an attached biofilm.

MODELING THE NON-LINEAR RESPONSE OF MIXED CULTURE BIOFILM  
STRUCTURES TO TURBULENT FLOW

By

Brett William Towler

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

of

Doctor of Philosophy

in

Engineering

MONTANA STATE UNIVERSITY  
Bozeman, Montana

January 2004

© COPYRIGHT

by

Brett William Towler

2004

All Rights Reserved

0378  
T659

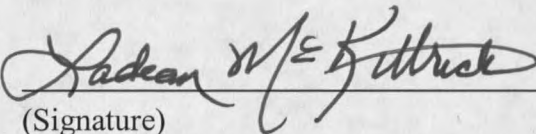
## APPROVAL

of a dissertation submitted by

Brett William Towler

This dissertation has been read by each member of the dissertation committee and has been found to be satisfactory regarding content, English usage, format, citations, bibliographic style, and consistency, and is ready for submission to the College of Graduate Studies.

Ladean McKittrick, Ph.D.

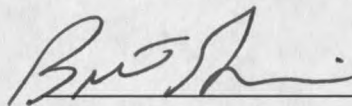
  
(Signature)

9 Jan 2004

(Date)

Approval for the Department of Civil Engineering

Brett Gunnink, Ph.D.

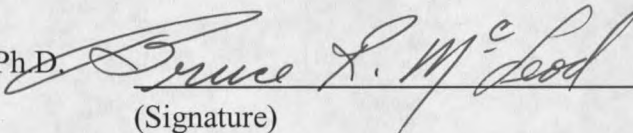
  
(Signature)

12 Jan 2004

(Date)

Approval for the College of Graduate Studies

Bruce McLeod, Ph.D.

  
(Signature)

1-14-04

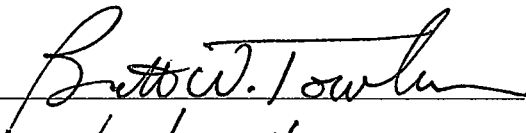
(Date)



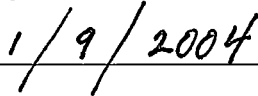
## STATEMENT OF PERMISSION TO USE

In presenting this dissertation in partial fulfillment of the requirements for a doctoral degree at Montana State University, I agree that the Library shall make it available to borrowers under rules of the Library. I further agree that copying of this dissertation is allowable only for scholarly purposes, consistent with "fair use" as prescribed in the U.S. Copyright Law. Requests for extensive copying or reproduction of this dissertation should be referred to Bell & Howell Information and Learning, 300 North Zeeb Road, Ann Arbor, Michigan 48106, to whom I have granted "the exclusive right to reproduce and distribute my dissertation in and from microform along with the non-exclusive right to reproduce and distribute my abstract in any format in whole or in part."

Signature



Date



*To  
Jennifer  
for her faith and love*

## TABLE OF CONTENTS

|                                                      |    |
|------------------------------------------------------|----|
| 1. INTRODUCTION.....                                 | 1  |
| RESEARCH GOAL.....                                   | 2  |
| 2. BIOFILM PHENOMENA.....                            | 3  |
| BIOFILM STRUCTURE.....                               | 4  |
| SIGNIFICANCE OF THE EPS .....                        | 5  |
| MORPHOLOGY AND MATERIAL PROPERTIES.....              | 7  |
| RELEVANCE OF BIOFILM: BIOFOULING.....                | 8  |
| 3. MECHANICS OF A VISCOELASTIC FLUID.....            | 12 |
| MATERIAL BEHAVIOR.....                               | 13 |
| ELASTICITY.....                                      | 14 |
| Isotropy.....                                        | 15 |
| INELASTIC FLUIDS.....                                | 16 |
| LINEAR VISCOELASTICITY.....                          | 18 |
| Maxwell Model.....                                   | 18 |
| Kelvin Model.....                                    | 19 |
| Burger Model.....                                    | 21 |
| CREEP AND RELAXATION BEHAVIOR.....                   | 22 |
| 4. RHEOLOGY OF A MIXED CULTURE BIOFILM.....          | 28 |
| SHEAR FLOW.....                                      | 28 |
| SHEAR CREEP.....                                     | 30 |
| MATERIAL FUNCTIONS.....                              | 36 |
| CREEP TESTING.....                                   | 38 |
| EXPERIMENTAL DESIGN.....                             | 39 |
| Biofilm Growth Reactor.....                          | 40 |
| Removal, Transport and Installation of Samples ..... | 44 |
| SHEAR CREEP TESTS.....                               | 47 |
| DISCUSSION OF RESULTS.....                           | 52 |
| SUMMARY.....                                         | 62 |
| 5. BIOFILM CONSTITUTIVE LAW.....                     | 64 |
| LINEARITY OF THE BIOFILM RESPONSE .....              | 66 |
| IMPLICATIONS OF THE LINEARITY COEFFICIENT.....       | 71 |
| SIGNIFICANCE OF THE CREEP COMPLIANCE.....            | 74 |

## TABLE OF CONTENTS - CONTINUED

|                                                                  |     |
|------------------------------------------------------------------|-----|
| OPTIMIZING PARAMETERS IN THE BURGER CONSTITUTIVE EQUATION.....   | 77  |
| PARAMETER OPTIMIZATION RESULTS .....                             | 84  |
| 6. DEVELOPMENT OF A COMPUTATIONAL MODEL.....                     | 95  |
| PHYSICAL PROBLEM DESCRIPTION.....                                | 95  |
| GOAL OF THE COMPUTATION MODEL .....                              | 96  |
| Finite Element Platform.....                                     | 97  |
| SOLID MECHANICS IN ANSYS.....                                    | 98  |
| Biofilm Constitutive Law in ANSYS.....                           | 99  |
| ANSYS Representation of the Burger Model.....                    | 99  |
| Numerical Validation of the Burger Model.....                    | 103 |
| Description of the Burger Model Assessment.....                  | 104 |
| Results of the Burger Model Assessment.....                      | 105 |
| COMPUTATIONAL FLUID DYNAMICS IN FLOTRAN.....                     | 109 |
| Fluid Boundary Conditions.....                                   | 109 |
| FLUID-STRUCTURE INTERACTION.....                                 | 109 |
| TEST CASE: SIMPLE SHEAR CUBE PROBLEM.....                        | 110 |
| Problem Description.....                                         | 112 |
| Solution and Convergence.....                                    | 112 |
| Discussion of Results.....                                       | 114 |
| Effective Shear Angle.....                                       | 117 |
| Comments on Material Behavior.....                               | 117 |
| SUMMARY.....                                                     | 118 |
| 7. APPLICATION OF THE COMPUTATIONAL MODEL.....                   | 120 |
| ANALYSIS OF ROLLING BIOFILM PHENOMENA.....                       | 120 |
| Problem Description.....                                         | 121 |
| Flow Development.....                                            | 121 |
| Centroid of an Arbitrarily Oriented Quadrangle.....              | 123 |
| Parametric Study of FSI Relationship to Burger Coefficients..... | 126 |
| Series 0: Maxwell Elastic Modulus Varied.....                    | 127 |
| Series 1: Kelvin Elastic Modulus Varied.....                     | 137 |
| Series 2: Maxwell Elastic Modulus Varied.....                    | 142 |
| Series 3: Kelvin Elastic Modulus Varied.....                     | 143 |
| Series 4 and 5: Mean Channel Velocity Varied.....                | 145 |
| Series 6 and 7: Biofilm Hemisphere Radius Varied.....            | 147 |
| Series 8: Pure Elastic Response Examined.....                    | 147 |
| Drag and Lift.....                                               | 148 |
| Polar Diagrams.....                                              | 148 |

## TABLE OF CONTENTS - CONTINUED

|                                                                   |     |
|-------------------------------------------------------------------|-----|
| 8. SUMMARY.....                                                   | 157 |
| REFERENCES CITED.....                                             | 160 |
| APPENDICES.....                                                   | 165 |
| APPENDIX A: BURGER COMPLIANCE CONVERSION VIA LAPLACE TRANSFORM... | 166 |
| APPENDIX B: MATLAB CODE FOR SLIDE & STAGGER ALGORITHM.....        | 172 |
| LVE_i subroutine.....                                             | 173 |
| LVE_r subroutine.....                                             | 176 |
| LVE_s subroutine.....                                             | 178 |
| LVEfit driver.....                                                | 181 |
| APPENDIX C: ANSYS INPUT FOR BURGER MATERIAL MODEL.....            | 191 |
| APPENDIX D: RESULTS OF BURGER MATERIAL MODEL VALIDATION.....      | 197 |
| APPENDIX E: ANSYS INPUT FOR SHEAR CUBE PROBLEM.....               | 202 |
| APPENDIX F: EFFECTIVE SHEAR ANGLE & MODULUS CALCULATIONS.....     | 220 |
| APPENDIX G: SAMPLE ANSYS INPUT FOR ROLLING BIOFILM PROBLEM.....   | 222 |
| APPENDIX H: SAMPLE CALCULATION OF A QUADRANGLE CENTROID.....      | 246 |

## LIST OF TABLES

| Table                                                                            | Page |
|----------------------------------------------------------------------------------|------|
| 1. Steady-state Viscosity and Recovery Capacity of Biofilm Samples.....          | 53   |
| 2. Steady-state Compliances of Biofilm Samples.....                              | 57   |
| 3. Zero-shear Viscosities of Biofilm Samples.....                                | 59   |
| 4. Recoverable Compliances of Biofilm Samples.....                               | 60   |
| 5. Descriptive Statistics for Biofilm Sample 3a, 3b Comparisons.....             | 62   |
| 6. Optimized Burger Model Parameters and Weighted Residuals.....                 | 89   |
| 7. Burger Parameters Used in Validation of<br>ANSYS/Prony Series Conversion..... | 105  |
| 8. Convergence Criteria after Final FSI Iteration.....                           | 113  |
| 9. Series 0 Parametric Study.....                                                | 127  |
| 10. Series 1 Parametric Study.....                                               | 137  |
| 11. Series 2 Parametric Study.....                                               | 142  |
| 12. Series 3 Parametric Study.....                                               | 143  |

## LIST OF FIGURES

| Figure                                                                                                                                           | Page |
|--------------------------------------------------------------------------------------------------------------------------------------------------|------|
| 1. Conceptual Drawing of Biofilm Formation.....                                                                                                  | 2    |
| 2. Linear Viscoelastic Mechanical Analogs.....                                                                                                   | 20   |
| 3. Simple Shear Flow.....                                                                                                                        | 29   |
| 4. Physical Interpretation of Shear Strain in Shear Flow.....                                                                                    | 31   |
| 5. Schematic of the Biofilm Growth Chamber, Drive Shafts,<br>Pulleys and Rheometer Disks.....                                                    | 41   |
| 6. Side View of the Biofilm Growth Chamber.....                                                                                                  | 42   |
| 7. View of the Biofilm Growth Reactor, as Built.....                                                                                             | 43   |
| 8. Side View of the AR1000 Rheometer Including Water Dam.....                                                                                    | 46   |
| 9. Epi-fluorescent Image of the Top of the Mixed Species Biofilm<br>Grown on a Rheometer Disk Showing Bacterial,<br>Rods, Cocci, and Chains..... | 47   |
| 10. Shear Creep Strain Response and Compliance Function for Sample 1.....                                                                        | 48   |
| 11. Shear Creep Strain Response and Compliance Function for Sample 2.....                                                                        | 48   |
| 12. Shear Creep Strain Response and Compliance Function for Sample 3a.....                                                                       | 49   |
| 13. Shear Creep Strain Response and Compliance Function for Sample 3b.....                                                                       | 49   |
| 14. Shear Creep Strain Response and Compliance Function for Sample 4.....                                                                        | 50   |
| 15. Shear Creep Strain Response and Compliance Function for Sample 5.....                                                                        | 50   |
| 16. Shear Creep Strain Response and Compliance Function for Sample 6.....                                                                        | 51   |
| 17. Shear Creep Strain Response and Compliance Function for Sample 7.....                                                                        | 51   |
| 18. Combination Graph of Biofilm Compliance Curves.....                                                                                          | 56   |
| 19. Compliance Functions for Repeated Tests on Sample 3.....                                                                                     | 61   |

## LIST OF FIGURES - CONTINUED

|                                                                                                                                                                                          |     |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| 20. Strain Response of Samples 3a and 3b.....                                                                                                                                            | 67  |
| 21. Linear Proportionality Coefficient.....                                                                                                                                              | 69  |
| 22. Residual Plot of the Linear Proportionality Coefficient<br>Versus the Observed Strain Ratios.....                                                                                    | 71  |
| 23. Creep Test Strain and Optimized Burger Model Function<br>for Sample 1.....                                                                                                           | 85  |
| 24. Creep Test Strain and Optimized Burger Model Function<br>for Sample 2.....                                                                                                           | 85  |
| 25. Creep Test Strain and Optimized Burger Model Function<br>for Sample 3a.....                                                                                                          | 86  |
| 26. Creep Test Strain and Optimized Burger Model Function<br>for Sample 4.....                                                                                                           | 86  |
| 27. Creep Test Strain and Optimized Burger Model Function<br>for Sample 5.....                                                                                                           | 87  |
| 28. Creep Test Strain and Optimized Burger Model Function<br>for Sample 6.....                                                                                                           | 87  |
| 29. Creep Test Strain and Optimized Burger Model Function<br>for Sample 7.....                                                                                                           | 88  |
| 30. Relaxation Test Simulation.....                                                                                                                                                      | 105 |
| 31. Comparison of the Analytically Calculated Time-dependent Relaxation<br>Modulus, $G_A(t)$ and the ANSYS/numerically-approximated<br>Time-dependent Relaxation Modulus, $G_S(t)$ ..... | 106 |
| 32. Absolute and Cumulative Errors Between the Analytical Relaxation<br>Modulus, $G_A(t)$ and the ANSYS Relaxation Modulus, $G_S(t)$ .....                                               | 108 |
| 33. Weakly-Coupled FSI Algorithm.....                                                                                                                                                    | 111 |
| 34. Shear Cube Model.....                                                                                                                                                                | 113 |



## LIST OF FIGURES - CONTINUED

|                                                                                            |     |
|--------------------------------------------------------------------------------------------|-----|
| 35. ANSYS-Generated Contour Plot of Elemental von Mises<br>Stresses in the Shear Cube..... | 115 |
| 36. ANSYS-Generated Contour Plot of Nodal Velocity<br>Norms in the Bulk Fluid.....         | 116 |
| 37. Rolling Biofilm Model.....                                                             | 122 |
| 38. Centroid of an Arbitrarily Oriented Quadrangle.....                                    | 124 |
| 39. Initial Geometry and Stress State of Simulation 05.....                                | 129 |
| 40. Final Geometry and Stress State of Simulation 05.....                                  | 130 |
| 41. Initial (t = 12 seconds) Velocity Field for Simulation 05.....                         | 131 |
| 42. Final (t = 42 seconds) Velocity Field for Simulation 05.....                           | 132 |
| 43. Change in x Component of Centroid with Time, Series 0.....                             | 133 |
| 44. Change in y Component of Centroid with Time, Series 0.....                             | 133 |
| 45. Change in the x Component of Fluid Forces, $R_x$ ,<br>Through Time, Series 0.....      | 134 |
| 46. Change in the y Component of Fluid Forces, $R_y$ ,<br>Through Time, Series 0.....      | 135 |
| 47. Change in the Resultant of Fluid Forces, $R$ , Through Time, Series 0.....             | 136 |
| 48. Change in x Component of Centroid with Time, Series 1.....                             | 138 |
| 49. Change in y Component of Centroid with Time, Series 1.....                             | 139 |
| 50. Change in the x Component of Fluid Forces, $R_x$ ,<br>Through Time, Series 1.....      | 140 |
| 51. Change in the y Component of Fluid Forces, $R_y$ ,<br>Through Time, Series 1.....      | 141 |
| 52. Change in the Resultant of Fluid Forces, $R$ , Through Time, Series 1.....             | 141 |

## LIST OF FIGURES - CONTINUED

|                                                                                       |     |
|---------------------------------------------------------------------------------------|-----|
| 53. Change in the Resultant of Fluid Forces, $R$ , Through Time, Series 2.....        | 143 |
| 54. Change in the Resultant of Fluid Forces, $R$ , Through Time, Series 3.....        | 144 |
| 55. Change $R$ , Through Time, for 2 Different<br>Channel Velocities, Series 4.....   | 145 |
| 56. Change $R$ , Through Time, for 2 Different<br>Channel Velocities, Series 5.....   | 146 |
| 57. Polar Diagram of Drag ( $C_D$ ) Versus Lift Coefficients ( $C_L$ ), Series 0..... | 150 |
| 58. Polar Diagram of Drag ( $C_D$ ) Versus Lift Coefficients ( $C_L$ ), Series 1..... | 152 |
| 59. Polar Diagram of Drag ( $C_D$ ) Versus Lift Coefficients ( $C_L$ ), Series 2..... | 152 |
| 60. Polar Diagram of Drag ( $C_D$ ) Versus Lift Coefficients ( $C_L$ ), Series 3..... | 153 |
| 61. Composite Polar Diagram for Series 0 Through 3.....                               | 153 |
| 62. Effect of Channel Velocity, $RE$ , on $C_D$ and $C_L$ . Series 0, 4, 5.....       | 154 |
| 63. Effect of Biofilm Radius on $C_D$ and $C_L$ . Series 0, 6, 7.....                 | 155 |

### ABSTRACT

Microbial biofouling of wetted interfaces can negatively impact the hydrodynamic performance of pressurized conduits. These impacts are due, in part, to the material properties of biofilm, yet few studies have examined this polymeric substance in the context of a constitutive relation. The goal of this research was two-fold; 1) to determine a suitable constitutive model for a mixed-culture biofilm and 2) use this material model in a numerical simulation to evaluate biofilm mechanical behavior in response to varying hydrodynamic conditions.

Creep tests revealed that these biofilms may be classified as viscoelastic *fluids*. Furthermore, results indicated the presence of viscous, time-dependent and instantaneous components to the biofilm compliance functions. A regression analysis ( $r^2 = 0.8819$ ) supported the treatment of these samples as *linear* viscoelastic fluids within the stress range of 0.1 Pa to 0.5 Pa. A specific linear viscoelastic constitutive equation was then determined by fitting experimental results to analytical solutions using an optimization algorithm. It was found that the *Burger* material model closely approximated the behavior of all samples. A numerical fluid-structure interface model was then developed and employed in a parametric study to investigate biofilm behavior. The effect of the Burger material parameters, mean flow velocity and biofilm size were examined. Simulations showed that weaker or softer biofilms (characterized by lower elastic moduli) were highly susceptible to lift forces. Additionally, polar diagrams were generated by plotting the coefficients of drag versus lift. The plots suggested that in the first few seconds after loading, the deformation paths taken by hemispherical biofilms are largely insensitive to specific material coefficients. Moreover, the diagrams illustrated that the effects of biofilm strength, size and channel velocity on displacement were predictable. These relationships may lead to the development of a simple, yet accurate method for predicting the hydrodynamic forces acting on an attached biofilm.

## INTRODUCTION

Bacterial aggregations and their associated activities on wetted interfaces are collectively referred to as 'biofilms'. Though the term was coined much later, Zobell (1943) is generally credited as the first true study of biofilms (Characklis and Marshall, 1990). The pace of progress in this area was initially slow, but accelerated in the 1970s when the ubiquity of attached microbial communities and their relevance to the fields of engineering and medicine became apparent. Biofilm research is now a critical component to a number of sciences including biomechanics, environmental engineering, naval science, agricultural and industrial manufacturing. Moreover, the impacts of attached biofilms are so varied in nature that collaboration among scientists, medical doctors, engineers and microbiologists is commonplace. Indeed, this study of biofilms is the very quintessence of interdisciplinary research.

There are two impelling forces behind much of the activity in this relatively young technical field. For one, there is great promise in the potential for harnessing the beneficial attributes of biofilms. For example, biofilm has been used as an oxygen barrier for abating the effects of acid rock drainage and as a novel subsurface remediation technology capable of simultaneously degrading trichloroethylene and reducing porous media permeability (Cunningham, et al., 1997). Secondly, there are significant costs associated with the adverse effects of microbial activity at interfaces and thus opportunities to reduce these effects are numerous. This is especially true in the disciplines of water resources and environmental engineering. For example, biofilm accumulation on water distribution mains contribute to the contamination of drinking

water due to continuous microbial growth and shear-induced detachment processes (Geesey and Bryers, 2000). Also, biofilm accumulation has been shown to significantly affect hydrodynamic performance of piping systems (Zelver, 1979). Given the pervasiveness of microbial organisms on wetted surfaces, biofilm may be the single most underrepresented hydrodynamic boundary condition. Understanding the physical properties of biofilm may lead to solutions that will reduce costs due to bacterial accumulation and possibly elucidate additional beneficial functions.

### Research Goal

This thesis represents an investigation of the material properties of a mixed-culture microbial biofilm. The work is comprised of three phases; experimentation, material analysis, and simulation. In the first phase, a mixed-culture biofilm was grown and mechanical tests were performed on the samples. Next, the experimental data was analyzed to determine an appropriate material model for these bacterial films. In the final step, the aforementioned material model was incorporated into a numerical simulation of biofilm growing on a wetted surface. The goal of this research was two-fold; 1) to determine a suitable constitutive model for a mixed-culture biofilm and 2) use this material model in a numerical simulation to evaluate biofilm mechanical behavior in response to varying hydrodynamic conditions.

## BIOFILM PHENOMENA

In an aqueous environment, microbial cells, transported downstream by a bulk fluid (i.e. water), can attach to virtually any wetted surface such as pipe walls or porous media. Once anchored to a substratum, these cells grow and reproduce. The immobilized bacteria produce an extra-cellular polymeric slime or EPS that serves to enhance the survival of these cells. This organic polymer matrix, together with the imbedded microbial cells, is referred to as biofilm. This sequence of transport, attachment and EPS development is conceptualized in Figure 1.

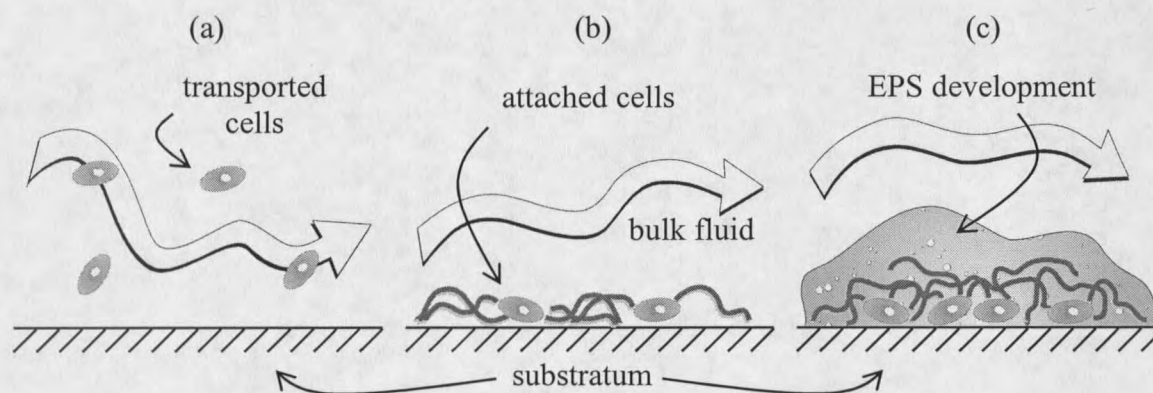


Figure 1. Conceptual Drawing of Biofilm Formation. (a) Microbial cells can be carried by a fluid until; (b) they attach to a wetted substratum. (c) Thereafter, many species of bacteria begin forming a polymeric slime known as EPS.

Biofilm Structure

On the microscale, this composite matrix of cells and polymer fibers is decidedly heterogeneous. In fact, this heterogeneity is the clear distinction between biofilm and other microbial communities (Characklis and Marshall, 1990). Furthermore, biofilms

tend to be porous, consisting mostly of water. Because of this porosity and the associated capacity for interstitial transport, biofilms also tend to adsorb and entrap solutes and inorganic particles such as clay and silt. Nevertheless, it is the biotic matrix that binds the biofilm into a coherent structure.

On a larger scale, biofilm can grow in discrete patches or cover surfaces uniformly. The thickness of a biofilm can vary greatly. Thickness can differ very little over the width of a rectangular duct (Bakke, 1986) or vary locally. Biofilms can be as thin as a single layer of cells attached to the wetted surface or form algal mats, which represent an upper extreme of depth (mats as thick as 400 mm have been observed). In general, biofilm thickness typically ranges from 10  $\mu\text{m}$  to 1000  $\mu\text{m}$  (Christensen and Characklis, 1990). Biofilm mass density can also vary greatly; values as high as 105  $\text{kg/m}^3$  (Hoehn and Ray, 1973) and as low as 10  $\text{kg/m}^3$  (Characklis, 1980) have been recorded.

A variety of factors, including bacterial species, nutrient supply and hydrodynamic conditions, can influence biofilm size, shape and density. It appears that biofilm thickness is influenced not only by species but by species diversity. Trulear (1983) found that film thicknesses rarely exceed 50  $\mu\text{m}$  in pure cultures of *Pseudomonas aeruginosa* compared with work by Characklis (1980) in which thicknesses in excess of 120  $\mu\text{m}$  were measured in mixed culture biofilms. Moreover, Characklis (1980) reported a link between biofilm density and fluid-induced shear stress. In these experiments, an increase in shear stress resulted in a proportional increase in biofilm density. Contrary to these findings, Kornegay and Andrews (1967) found no appreciable increase in density

due to shear stress increases. However, the latter experiments were conducted in a RotoTorque reactor, while the former investigation was a flow-through tubular reactor experiment. Other factors may affect density as well. Turakhia (1986) reported that increasing calcium concentration lead to an increase in biofilm density. Evidence also suggests that biofilm density increases with age (Hoehn and Ray, 1973, Trulear, 1983).

### Significance of the EPS

Most bacteria excrete organic polymer substances. Specifically, biofilms excrete a viscous, soluble slime comprised almost entirely of polysaccharides, a carbohydrate compound consisting of monosaccharides joined by glycosidic bonds. Thus, biofilm is often regarded as a biosynthetic polymer (Flemming et al. 2002). While the composition of biofilm can include any number of entrapped organic and non-organic materials, the physical properties of biofilm are largely dependent upon the specific microbial species and its form of extra-cellular polymeric slime. Accordingly, biofilm can be regarded as a collection of microorganisms trapped in an organic polymer gel. Indeed, the properties of this polymer gel are of great importance to the understanding of biofilm physical behavior (Christensen and Characklis, 1990).

In general, polymers are comprised of chains of organic molecules or mers (polymer building blocks formed from monomers) and a solvent (Shackelford, 1985). Cross-linking occurs when adjacent chains are chemically bonded. This bonding acts to resist vibration and rotation of the chains, thereby lending strength to the polymer (Callister, 1985). When partly hydrated, the molecular chains in a biopolymer become,



to some degree, cross-linked. The resulting tangled, three-dimensional web of polymer chains forms a biopolymer gel. This gel or EPS can account for as much as 90% of the biofilm organic carbon (Bakke, 1984). Furthermore, the EPS may contain several different polymer types. Christensen et al. (1985) observed that *Pseudomonas sp.* NCMB 2021 produced two different polysaccharides; one produced during exponential growth and another during the decay and stationary phases of cell growth. In an engineering context, the EPS is significant for two reasons. First, it is the polymeric nature of the EPS that determines the interaction between the biofilm and the substratum. Clearly, this interaction plays a role in biofilm adhesive strength. Secondly, a biofilm's integrity is derived from the physical properties of its biopolymer. In fact, it has been posited that microbial EPS is responsible for the mechanical properties of biofilm (Wingender & Flemming 1999). The solvent, typically water, in which these polymers are dispersed, plays a role as well. The water content of any biofilm is high. Values from 87% to 99% have been reported (Kornegay and Andrews, 1967, Characklis et al., 1981, Zilver, 1979). The solvent can affect chemical reactions that in turn affect biofilm adhesion, stability and rheology. Also significant is that the solvent (i.e. water) will dictate the nature of biofilms hydrostatic response to stress.

### Morphology and Material Properties

Typically, the life cycle of biofilms has been described in terms of three distinct processes: attachment, accumulation and detachment (Characklis, 1981). Attachment is the process wherein planktonic cells, transported by a bulk fluid, move downstream

before finally attaching to a wetted surface (i.e. substratum). During accumulation, cells reproduce and create an EPS matrix. Detachment refers to two mechanisms: erosion of individual cells and larger scale detachment or sloughing of biofilm. Excluding detachment, attached biofilms were historically modeled as stationary microbial colonies.

More recent investigations have identified different transient forms and structural features in biofilms including ripples, dunes and finger-like projections called 'streamers' (Gjaltema et al., 1994, Stoodley et al., 1997, Dockery and Klapper, 2001). Furthermore, the relationship between these forms and bulk fluid hydrodynamics has been studied. Stoodley et al. (1999a) grew mixed culture biofilms consisting of *Klebsiella pneumoniae*, *Pseudomonas aeruginosa*, *Pseudomonas fluorescens*, and *Stenotrophomonas maltophilia* in glass flow cells under different hydrodynamic conditions. Short-term changes in mean velocity resulted in migration and elongation of the biofilm structures. This demonstrates interplay between biofilm morphology and local hydrodynamic conditions. However, this was essentially a study of biofilm kinematics; no account of fluid-induced force or biofilm material properties was made.

Biofilms respond to stresses induced by a flowing fluid. Clearly, the nature of that response, and more generally biofilm morphology, is due to intrinsic material properties. Nevertheless, many models do not account for these properties (Gujer and Wanner, 1990; Hermanowicz, 1998; Dockery and Klapper, 2001). Those that have recognized the relevance of material properties are ultimately limited in applicability. Stoodley et al. (1999b) measured streamer elongation in glass flow tube experiments. The methods developed in this study for in-situ measurement of structural deformation

were highly novel and the experimental results were equally informative. However, the authors' calculation of an 'effective' shear modulus was less valuable. By approximating the fluid induced forces by a constant wall shear stress, the applied load was drastically oversimplified. Additionally, the authors' statements that the mixed biofilm culture behaved as a viscoelastic fluid contradicts the elastic model implicit in reporting a one-dimensional shear modulus. Picioreanu et al. (2001) developed a two-dimensional model of biofilm detachment that includes an elastic response to fluid stress. While this multi-dimensional model is more realistic, the elastic constitutive relation does not agree with the viscoelastic responses observed by other investigators (Ohashi and Harada, 1994). An accurate model of biofilm streamers must resolve the shear and pressure forces due to a complex localized flow field and couple that load distribution with a realistic viscoelastic stress-strain relation.

#### Relevance of Biofilm: Biofouling

Biofilm fouling or simply 'biofouling' refers to the undesirable accumulation of microbial biofilm deposits on wetted surfaces. In the preceding discussion, biofilms and their properties were described in terms of well-defined experimental systems. While mention was made of entrapped particulate matter, the focus of the discussion was on overall geometry and the attributes of a microbial biopolymer called EPS. Heretofore, the term biofilm referred only to the attached microbial cells and the EPS which they create. In real settings however, biofilm is a more complex aggregation of microorganisms, their products and accumulated inorganic material. The experimental

conditions under which many biofouling studies are performed are not well defined. Therefore, it is difficult, if not impossible, to separate the effect of individual processes (Characklis and Marshall, 1990). Consequently, the effects of internal heterogeneity cannot be appreciated in most of these field studies.

Microbial fouling can occur in a number of different settings. Biofilm can accumulate in cooling water systems, storage tanks and wastewater treatment systems (Characklis, 1990). This accumulation can inhibit the hydraulic performance in water distribution systems and piping networks (Picaloglou et al. 1980; Zilver 1979; Characklis 1973). It has been shown that biofilm accumulation can negatively impact other industrial processes such as ion exchangers (Flemming, 1987) and reverse osmosis membranes (Ridgway et al., 1984). Biofilm accumulation in porous media can impact oil recovery operations (Cunningham et al., 1990). Interfacial bacterial films can also thrive in marine environments (Cooksey and Wigglesworth-Cooksey, 1995) and can cause an increase in hydrodynamic drag on ship hulls (Lewthwaite, 1985). Fouling is also relevant in the medical field where the growth of biofilm on medical devices can result in persistent infections (Costerton et al. 1999) and the detachment of dental biofilms has been linked to a range of systemic diseases (Newman 1998). Furthermore, there are significant economic impacts of biofouling. Pritchard (1981) estimates the costs of biofouling in Britain at more than \$600 million per year. Energy losses due to hydrodynamic drag, reductions in equipment capacity, replacement of corroded equipment and quality control problems all have associated costs.

The hydrodynamic impact of biofouling or more specifically, the increased resistance due to biofilm accumulation is well documented (Bryers and Characklis, 1981). In general, hydrodynamic resistance is a function of both surface drag and form drag (Currie, 1974; Rouse, 1946). Surface drag is due to the frictional resistance of a fluid boundary. Form drag is an expression of pressure differentials due to boundary layer separation. In the context of biofouling, this suggests there are two causes of hydrodynamic drag: biofilm 'skin' friction and form drag due to the aforementioned biofilm structures (e.g. ripples, dunes, streamers). Several investigators have posited that pressure drop due to increased frictional resistance occurs when biofilm growth exceeds the thickness of the laminar sublayer (Characklis, 1979; McCoy and Costeron, 1982). Since uniform biofilm accumulation would effectively redefine the viscous sublayer, these conclusions imply that form drag is the operative cause of biofilm-induced pressure drop. This implication was supported by Stoodley, et al(1998).

As mentioned previously, the filamentous biofilm structures called streamers commonly manifest under high flow conditions. If regarded simply as static drag-inducing forms in the flow field, they would contribute to the overall pressure drop observed in many internal and external flows. However, there is evidence to suggest their energy reducing effects are greater than can be accounted for by size and shape alone. Picologlou et al. (1980) found that biofilm caused a greater pressure drop in a circular flow cell than would be expected if the biofilms were rigid. They suggested this was due, in part, to the viscoelastic material properties of microbial biofilms. Viscoelastic materials have the potential to dissipate mechanic energy. The implication

of the Picologlou, et al. (1980) study is that velocity perturbations in turbulent flow fields incite oscillations in streamers. The streamers, in turn, dissipate energy in the flow thereby contributing to the overall pressure drop. The dependence of streamer morphology on material properties was previously identified as a catalyst for examining the biofilm stress-strain relation. Clearly, the pressure drop due to the viscoelastic nature of streamers provides further motivation for activity in this area.

## MECHANICS OF A VISCOELASTIC FLUID

Mechanics is the physical science that describes the motion of objects subjected to forces. It is often divided into three separate fields of study: the mechanics of rigid bodies, the mechanics of deformable bodies and fluid mechanics (Beer and Johnston, 1988). While many real applications of this applied science can be conveniently categorized as one of the three, problem complexity often leads to topical crossover. Regardless, all of these pursuits share a commonality; mechanists seek to relate two quantities: force and deformation.

Forces are defined as either surface forces or body forces. Body forces are those that act on all differential volume elements in an object. Examples include gravity and electromagnetism. Surface forces are those that are distributed across the bounding exterior of an object or element. Hydrostatic pressure and fluid shear are examples of surface forces. Stress is a measure of force intensity or force per unit area. This force intensity can be further subcategorized as normal stress (stress acting perpendicular to a surface) or shear stress (stress acting tangential to a surface). Moments, such as the torque applied to a cylindrical shaft or the bending moment of a steel beam, are products of a force and an arm, or distance. As such, they are also included in a discussion of forces.

Historically, the study of deformable solids was referred to as "strengths of materials". It is clear that the name arose from the very real need to understand the 'strengths' of construction materials (Lardner and Archer, 1994). However, as this investigation into biofilm as an engineering material demonstrates, modern application of

these principles is much broader. Thus, the field is now commonly referred to as “mechanics of materials”. In the context of mechanics of materials, deformation is defined as the alteration of shape due to pressure or stress. Deformation, however, can be presented in two ways: strain and displacement. Strain refers to the local deformation of an object, or element, typically due to an applied stress (Flügge, 1975). More precisely, strain is change of form without reference to an external coordinate system. Analogous to the categorization of stress in the preceding paragraphs, normal strain is the ratio of an element’s elongation to its original length and shear strain is a measure of a rectangular element’s angular deformation. Displacement is the translation or rotation of a point within a larger solid (or fluid mass) due to the application of stress or pressure. Unlike strain, deformation is described with respect to a fixed coordinate system. Relating deformation to applied forces or stresses requires an understanding of material behavior.

### Material Behavior

Obviously, how an object behaves under loading must be due to some essential, intrinsic, constitutive property of the material. In fact, the deformational response of a specific material to stress is specified by a so-called ‘constitutive law’. The development of a constitutive law or relation is a complex process because a real material’s strain response may vary with, among other things, temperature, strain-rate and the magnitude of applied loads (Mase & Mase 1999). An allusion to the viscoelastic behavior of microbial biofilm was made in the previous chapter. Viscoelastic materials are ones



which exhibit a characteristic called a 'memory effect'. Material memory implies that the strain at a given time is dependent not only to the stress at that instant, but, in fact, on all past states of stress. Viscoelastic materials have a memory for all past states of stress (Christensen, 1982). Strictly speaking, material memory is not in and of itself indicative of viscoelasticity. Other material theories, such as the incremental theory of plasticity, demonstrate this memory effect. Another fundamental distinction is necessary; viscoelastic materials are strain-rate dependent. Plasticity theory is independent of the time scale used in loading and unloading procedures (Christensen, 1982).

As the name implies, viscoelastic models combine the behaviors of both elastic solids and viscous fluids. Qualitatively, the capacity to store mechanical energy is indicative of elastic behavior. Conversely, viscous fluids readily dissipate mechanical energy. Viscoelastic material models possess the ability to both store and dissipate such energy. Thus, one may regard the elastic and viscous responses as end-points on a spectrum of viscoelastic behavior. It is appropriate then to first review elasticity and Newtonian viscosity before coupling these concepts into a viscoelastic stress-strain relation that may be used to describe microbial biofilm.

### Elasticity

Elastic models respond instantaneously to applied stress and manifest complete recovery once unloaded. The term linear elastic is used to identify models that exhibit a linear elastic relationship between stress and strain. More formally, linear theory is based on the assumption that displacement and displacement gradients are so small that no

distinction between Eulerian and LaGrangian coordinate systems is necessary (Lai, 1993). This stress-strain relationship takes the form:

$$\sigma_{ij} = C_{ijkl} \varepsilon_{kl} \quad (1)$$

where  $\sigma_{ij}$  is the stress tensor,  $\varepsilon_{kl}$  is the strain tensor and  $C_{ijkl}$  represents 36 distinct elastic coefficients. This equation is often called Hooke's law after Sir Robert Hooke who first recorded this type of behavior in 1678 (Nash, 1994). By neglecting thermal effects and employing the principle of conservation of energy, it can be shown that only 21 distinct coefficients exist.

### Isotropy

If the substructure of a material is sufficiently randomized, any part of the body will display the same material parameters in all directions. This condition is referred to as isotropy. Under these circumstances, an isotropic material's elastic coefficients are invariant under any rotation of coordinates. Applying this assumption to Hooke's Law (1), it can further be shown that there are only two distinct elastic coefficients. The relation then takes the form

$$\sigma_{ij} = \frac{E}{1+\nu} \left( \varepsilon_{ij} + \frac{\nu}{1-2\nu} \delta_{ij} \varepsilon_{kk} \right), \quad (2)$$

where  $E$  is the elastic or Young's modulus,  $\delta$  is the Kroneker delta and  $\nu$  is the Poisson's ratio. Two other terms are identified as

$$K = \frac{E}{3(1-2\nu)}, \quad (3)$$

and

$$G = \frac{E}{2(1+\nu)}, \quad (4)$$

where  $K$  and  $G$  represent the Bulk Modulus and Shear Modulus, respectively.

### Inelastic Fluids

Unlike an elastic solid, a fluid, by definition, will deform continuously under the action of an applied stress (Janna, 1993). This concept will be revisited in the next chapter when classifying viscoelastic substances. In general, fluids can be classified as viscous or invicid. Viscous fluids are modeled using a non-zero viscosity and allow for the development of shear stresses within the fluid. Shear stresses do not develop in invicid fluids giving them a zero effective viscosity. While all real fluids are viscous to some degree, the distinction is made to simplify some solutions where viscosity of a fluid plays no appreciable role (Anderson, 1995). Fluids can also be labeled as compressible

or incompressible. Since this investigation involved studying biofilm in a hydrodynamic environment, the discussion is limited to compressible fluids. This will be made clearer later.

The stress-deformation relationship for any fluid takes the form:

$$\tau_{ij} = f_{ij}(D_{pq}), \quad (5)$$

where  $\tau$  is the shear stress acting on the  $e_2$  plane in the  $e_1$  direction and  $f(D)$  is an undefined function of deformation rate,  $D$ . If this function,  $f$ , is linearly proportional to  $D$ , the resulting stress function is that of an inelastic fluid or

$$\tau_{ij} = K_{ijpq} D_{pq}, \quad (6)$$

where  $K$  is a fluid specific coefficient. This type of response is called Newtonian, named for Sir Isaac Newton who, in 1687, wrote of a fluid property he called the “lack of slipperiness” (Phan-Thien, 2002). This property is now referred to as viscosity. Newtonian viscosity is an important rheological concept; coupling equation (6) with Hooke’s Law (2) will lead to the development of a linear viscoelastic stress-strain relationship.

### Linear Viscoelasticity

Linear viscoelasticity is a special case of viscoelastic theory that makes use of the linear elastic and Newtonian theories presented in the previous section. Through the application of superposition principles, the stress-strain relation of viscoelastic materials may be conveniently modeled using combinations of the mechanical analogs for elastic and viscous behavior (i.e. a spring and a dashpot). By arranging these linear analogs in series or in parallel, one can develop a truly non-linear stress-strain response. Nevertheless, the theory is referred to a *linear* viscoelasticity because an increase in the applied stress results in a proportional increase in material strain (Findley et al, 1975).

#### Maxwell Model

The two-element Kelvin and Maxwell models are considered the building blocks of higher order linear viscoelastic analogs. The Maxwell element consists of a spring and dashpot in series (Fig 2). Conceptually, the dashpot may be regarded as a simple piston that contains a fluid which slowly passes through an orifice if a tensile stress is exerted on it. The stress-strain relationship for this analog is equation (6). A spring element is used to conceptually represent the linear Hookean response (2). The stress-strain relation is developed in one dimension and will be extrapolated into a 3 dimensional constitutive model in Chapter 5. Connected in a series, the elastic strain of the spring and the time-dependent strain of the dashpot are additive. Thus, the Maxwell stress-strain relation is

$$\frac{\dot{\sigma}}{G} + \frac{\sigma}{\eta} = \dot{\varepsilon}, \quad (7)$$

where  $\sigma$  is the stress across both elements,  $\varepsilon$  is the resulting strain,  $G$  is the elastic modulus of the spring and  $\eta$  is the viscosity of the dashpot. Note that the deformation rate  $D$  in (6) is equivalent to  $\partial\varepsilon/\partial t$ . The Maxwell model's behavior under any stress is an instantaneous elastic deformation followed by a strain rate dependent stress response. Because the Maxwell model is the "linear" sum of linear elastic and linear viscous components, it will deform continuously under a constant load and is often referred to as a "linear viscoelastic fluid".

#### Kelvin Model

The Kelvin model is the second of these basic forms or building blocks. It consists of a spring and dashpot in parallel (Fig. 2). The strain across these parallel elements is not additive; the stress, however, is (Gittus 1975). Accordingly, the stress-strain relation for a Kelvin model is

$$\sigma = G\varepsilon + \eta\dot{\varepsilon}. \quad (8)$$

The Kelvin response is drastically different from the Maxwell element's behavior. There is no instantaneous elastic response. When exerting a stress (tensile or compressive) on the Kelvin element, the dashpot acts in such a way as to dampen the behavior of the

elastic spring. Furthermore, when unloaded, the dashpot will act to dampen the recovery of the deformed spring. Unlike the Maxwell model, strain energy is not dissipated by the dashpot. Therefore, complete recovery will occur after an infinite amount of time. For this reason, the Kelvin model is referred to as a linear viscoelastic solid.

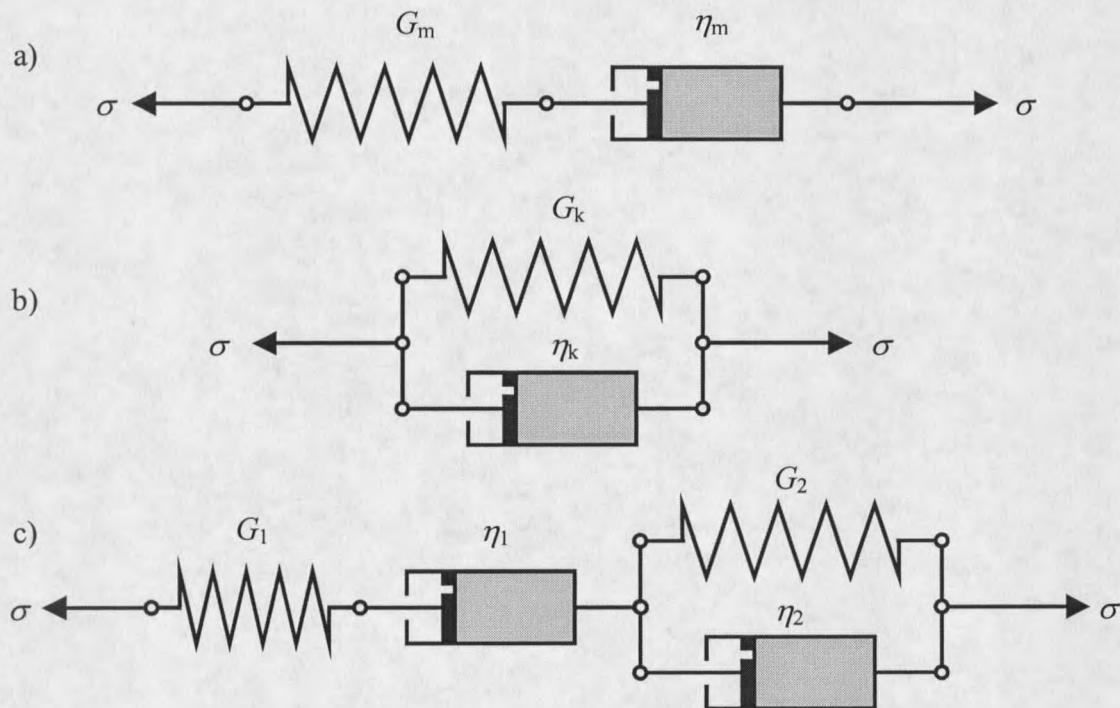


Figure 2. Linear Viscoelastic Mechanical Analogs: a) the Maxwell model; b) the Kelvin model and; c) the Burger model.  $\sigma$  denotes an applied compressive or tensile stress.  $G$  and  $\eta$  represent elastic and viscous coefficients, respectively.

Neither of these two linear viscoelastic models is appropriate for describing the stress-strain response of a real material. Their limitations can be mitigated by adding elemental analogs to increase the complexity of the behavior (and stress-strain equations). For example, the standard linear solid model is a serial combination of a Kelvin element

and a spring. Such a model will demonstrate time-dependent and instantaneous deformation and recovery. However, it has no capacity for permanent viscous loss observed in fluids. Similarly, a Jefferies fluid or standard linear fluid model is formed by a serial arrangement of a Kelvin element and a dashpot. It, too, is limited in that it makes no account for instantaneous elastic behavior.

### Burger Model

The 4-element Burger model (Fig. 2) is developed by arranging Kelvin and Maxwell elements in series. The Burger model is significant because it is the lowest order linear viscoelastic analog that can account for instantaneous response, time-dependent deformation and permanent strain (i.e. viscous loss). Its stress-strain equation is derived by considering the analog's strain response under stress. As stated previously, the strain of two models connected serially is additive. Thus, the total strain of a Burger model is the sum of the strain of both serial elements, Kelvin and Maxwell. Equations (7) and (8) can be combined to produce the stress-strain relation for this 4 element viscoelastic analog:

$$\tau + \left( \frac{\eta_1}{G_1} + \frac{\eta_1}{G_2} + \frac{\eta_2}{G_2} \right) \frac{\partial \tau}{\partial t} + \left( \frac{\eta_1 \eta_2}{G_1 G_2} \right) \frac{\partial^2 \tau}{\partial t^2} = \eta_1 \frac{\partial \gamma}{\partial t} + \left( \frac{\eta_1 \eta_2}{G_2} \right) \frac{\partial^2 \gamma}{\partial t^2}, \quad (9)$$

where the subscript 1 refers to the Maxwell subcomponents and the subscript 2 refers to the Kelvin subcomponents. Additional elements can be added to this analog to create additional degrees of freedom in the model's response.



### Creep and Relaxation Behavior

Creep and relaxation are generalized forms of time-dependent behavior as well as common experimental techniques designed to elucidate such behavior. While the details of how these experiments are carried out are presented later in Chapter 4, the descriptions of these two behaviors and the derivations of the associated strain functions are presented here. Creep refers to the shear or elongation strain of a material in response to an instantaneous, and subsequently constant, stress. Mathematically, this loading can be expressed using the unit step function.

$$\sigma_{12} = \sigma_0 U(t), \quad (10)$$

where  $\sigma_0$  is a constant shear, tensile or compressive stress.  $U(t)$  is the unit step function of the form:

$$U(t - t_1) = 1 \quad t > t_1, \quad U(t - t_1) = 0 \quad t \leq t_1, \quad (11)$$

where  $t_1$  is the time at which the constant load is applied. If the argument is restricted to shear (though the tensile response is analogous) and the load is applied at  $t_1 = 0$ , a shear creep strain equation is developed. Moreover, the specific shear strain equation for a Burger model is attained by substituting (10) into the stress-strain response for the Burger model (9) and integrating with respect to time. Then

$$\gamma(t) = \tau_o \left[ \frac{1}{G_1} + \frac{t}{\eta_1} + \frac{1}{G_2} \left( 1 - e^{-G_2 t / \eta_2} \right) \right], \quad (12)$$

where  $\gamma$  is the shear strain,  $t$  is the time in seconds,  $\tau_o$  is the constant shear stress,  $\eta_1$  is the viscosity coefficient in the Maxwell element,  $G_1$  is the (elastic) shear coefficient in the Maxwell element,  $\eta_2$  is the viscosity coefficient in the Kelvin element,  $G_2$  is the shear coefficient in the Kelvin element. If the stress,  $\tau_o$ , is removed at a later time  $t = t_s$  then the recovery behavior of the Burger model can be described by

$$\gamma(t) = \tau_o \left[ \frac{1}{G_1} + \frac{t}{\eta_1} + \frac{1}{G_2} \left( 1 - e^{-G_2 t / \eta_2} \right) \right] - \tau_o \left[ \frac{t_s}{\eta_1} + \frac{1}{G_2} \left( e^{-G_2 t_s / \eta_2} - 1 \right) e^{-G_2 t / \eta_2} \right]. \quad (13)$$

Collectively, the bracketed terms in equations (12) and (13) are referred to as the creep compliance.

Several important relationships may be drawn from (12) and (13). First, the instantaneous elastic response (at  $t = 0$ ) and recovery (at  $t = t_s$ ) are governed by the isolated spring in the Maxwell element. In theory then, the shear modulus of the Maxwell spring can be calculated directly from the ratio of the applied stress,  $\tau_o$ , to the instantaneous strain response,  $\gamma(t = 0)$ . Secondly, the unrecoverable strain or viscous loss is governed solely by the isolated dashpot also in the Maxwell element. Thus, the Maxwell's viscous coefficient can also be directly calculated as the product of  $t_s$  and the ratio of the applied stress to the ultimate strain response,  $\gamma(t \rightarrow \infty)$ . Finally, it can be

shown that the slopes of the strain curve at  $t = 0$  and  $t = t_s$  can be related to the shear modulus and viscous coefficients in the Kelvin component. Few, if any, real fluids demonstrate a perfect Burger creep response. Therefore, direct determination of these material parameters from discrete data is almost always impossible (Findley et al, 1976). However, these observations can be used as initial estimates for a numerical optimization scheme (see Chapter 5).

Relaxation is the counterpart behavior to creep. Specifically, relaxation connotes dissipation of internal stresses in a material under a constant strain condition. This is expressed as

$$\varepsilon_{12} = \varepsilon_0 U(t), \quad (14)$$

where  $\varepsilon_0$  is a constant shear, compressive or tensile strain. As with the development of the creep response, if the argument is restricted to shear (and again, the tensile response is similar) and the load is applied at  $t_1 = 0$ , a shear creep strain function can developed from equation (9).

Describing the constant shear strain condition in a fashion analogous to constant shear stress (10, 11) one can derive a strain history function and its first and second derivatives with respect to time, i.e.,

$$\varepsilon = \varepsilon_0 U(t); \quad \dot{\varepsilon} = \varepsilon_0 \delta(t); \quad \ddot{\varepsilon} = \varepsilon_0 \frac{\partial \delta(t)}{\partial t}, \quad (15)$$

where  $\varepsilon_0$  is a constant shear, tensile or compressive strain.  $\delta(t)$  is the Dirac delta or Unit Impulse function of the form:

$$\delta(t) = 0, \quad t \neq 0; \quad \int_{-\infty}^{+\infty} \delta(t) dt = 1. \quad (16)$$

Substituting equations (15, 16) into the stress-strain relation for a Burger model (9) results in

$$\tau + \left( \frac{\eta_1}{G_1} + \frac{\eta_1}{G_2} + \frac{\eta_2}{G_2} \right) \frac{\partial \tau}{\partial t} + \left( \frac{\eta_1 \eta_2}{G_1 G_2} \right) \frac{\partial^2 \tau}{\partial t^2} = \eta_1 \varepsilon_0 \delta(t) + \left( \frac{\eta_1 \eta_2}{G_2} \right) \varepsilon_0 \frac{\partial \delta(t)}{\partial t}. \quad (17)$$

Converting equation (17) into a stress function (and relaxation modulus) resembling the strain function (12) is not straightforward. This complexity is due to the fact that the stress function of the Burger model is not the sum of the stress functions of its serial components, the Kelvin and Maxwell models. This was the case with the stress functions (12, 13). Here, a Laplace transform of (17) must be taken. The details of this calculation are left for Appendix A. The resulting stress equation for a 4 element linear viscoelastic Burger model is :

$$\sigma(t) = \varepsilon_0 \frac{\left[ \left( \eta_1 - \frac{\eta_1 \eta_2}{G_2} T_1 \right) e^{-T_1 t} - \left( \eta_1 - \frac{\eta_1 \eta_2}{G_2} T_2 \right) e^{-T_2 t} \right]}{\sqrt{\left( \frac{\eta_1}{G_1} + \frac{\eta_1}{G_2} + \frac{\eta_2}{G_2} \right)^2 - 4 \frac{\eta_1 \eta_2}{G_1 G_2}}} \quad (18)$$

The constants  $T_1, T_2$  are defined as

$$T_1 = \frac{\frac{\eta_1}{G_1} + \frac{\eta_1}{G_2} + \frac{\eta_2}{G_2} - \sqrt{\left( \frac{\eta_1}{G_1} + \frac{\eta_1}{G_2} + \frac{\eta_2}{G_2} \right)^2 - 4 \frac{\eta_1 \eta_2}{G_1 G_2}}}{2 \frac{\eta_1 \eta_2}{G_1 G_2}}, \quad (19)$$

and

$$T_2 = \frac{\frac{\eta_1}{G_1} + \frac{\eta_1}{G_2} + \frac{\eta_2}{G_2} + \sqrt{\left( \frac{\eta_1}{G_1} + \frac{\eta_1}{G_2} + \frac{\eta_2}{G_2} \right)^2 - 4 \frac{\eta_1 \eta_2}{G_1 G_2}}}{2 \frac{\eta_1 \eta_2}{G_1 G_2}}. \quad (20)$$

Collectively, all the terms in equations (17-20) save  $\varepsilon_0$  are referred to as the shear stress relaxation modulus.

As outlined in Chapter 1, this body of work represents an examination of the material properties of a mixed culture microbial biofilm. As suggested by previous researchers (Ohashi and Harada, 1994) and validated in this investigation (See Chapter

5), biofilms respond to stress in a non-linear fashion. Indeed, the behavior is decidedly viscoelastic in nature. The preceding development of the equations governing this viscoelastic stress-strain relation served two purposes; a basis for empirical validation of the linear Burger formulae and as a general constitutive law to be used in computer-aided modeling. Specifically, the creep and relaxation equations (12, 18, 19, 20) will be used in conjunction with experimental data (see Chapter 4) and a numerical optimization procedure (see Chapter 5) to validate the use of a linear Burger model in describing biofilm behavior. The general Burger stress-strain relation (9) will be used in conjunction with the results of Chapter 5 to develop a coupled finite element fluid-structure model of biofilm streamers in a hydrodynamic environment using ANSYS 7.1 (see Chapter 6). This computational model will provide insight into the interaction between force and deformation in a realistic setting: the biofilm response to turbulent flow.

## RHEOLOGY OF A MIXED CULTURE BIOFILM

In general terms, rheology is the study of deformation and flow of matter, specifically fluids. In a more practical sense, however, this field is dominated by work with non-Newtonian fluids whereas the study of simpler fluids (e.g. water, air, oil) fall under the classification of fluid mechanics. Complex fluids follow neither Newtonian viscosity relationships nor Hooke's law of elasticity. The range of behavior between these extremes is quite varied and classifying non-Newtonian responses requires extensive testing or rheometry. Rheometry can be categorized as either mechanical, such as methods based on torsional plates, or non-mechanical such as birefringence techniques.

Biofilm is a polymeric substance (Flemming, et al., 2002) and as such its behavior is decidedly non-Newtonian. The rheology of a mixed biofilm culture was investigated through creep experiments using a research-grade rheometer. Quantitative and qualitative assessments of the biofilm response were made using the concepts of shear flow, creep and material functions.

### Shear Flow

Shear flow is perhaps the most commonly used type of flow in the field of rheology. Shear flows are produced in a variety of standard rheometers including rectilinear plates, torsional plates, cone and plate, and couette rheometers. The physics of

simple shear flow is idealized by sandwiching a fluid between two effectively infinite parallel plates (Fig. 3).

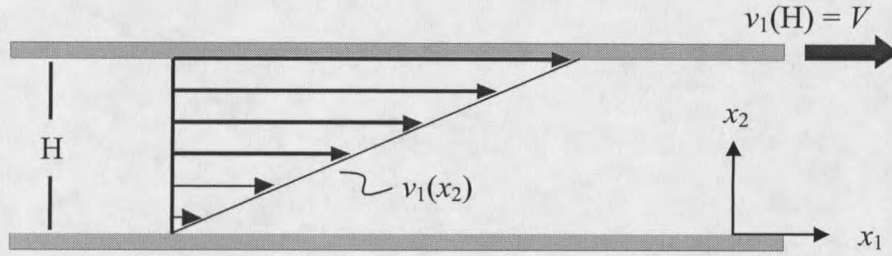


Figure 3. Simple Shear Flow. A velocity of  $V$  is imparted to the top plate which generates a shear driven velocity profile in the fluid.

A drag flow is induced when one of the plates is held stationary while the other is moved at a constant velocity parallel to the other plate. In this situation, shear stress gradients cause fluid laminae or layers to slide past each other without mixing. The velocity gradient varies with the distance between plates and is invariant in all other directions. As a result, particle path lines run parallel to the infinite plates. The velocity profile is related to the shear rate

$$\begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix} = \begin{pmatrix} \frac{\partial v_1}{\partial x_2} x_2 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} \dot{\gamma}_{21}(t) x_2 \\ 0 \\ 0 \end{pmatrix}, \quad (21)$$



where  $v_1, v_2, v_3$  refer to the 1, 2 and 3 directions of the velocity vector, respectively, and  $\gamma_{21}(t)$  refers to the shear strain on the  $e_2$  (or  $x_1$ - $x_2$ ) plane in the 1 direction.

### Shear Creep

Most rheological tests based on simple shear flow involve the application of a constant shear rate while a shear stress response is measured. These tests are convenient because a constant shear rate allows for simplification of equation (21):  $\partial v_1 / \partial x_2$  is a constant. Examples include steady-state capillary flow, cone and plate and parallel plate rheometry. An exception to the constant shear rate approach is to drive the flow through application of a constant applied force. In this situation, the time-dependent deformation of a fluid is measured while the shear stress is held constant. This unsteady shear response and the related experiment are referred to as creep. Mathematically, a shear stress function is prescribed:

$$\tau_{21}(t) = \begin{cases} 0 & t < 0 \\ \tau_0 = \text{constant} & t \geq 0 \end{cases} \quad (22)$$

This approach involves measuring the change in shape of the material over some interval as a result of a constant applied shear stress. Ultimately, this change in shape must be related to shear strain.

Shear strain,  $\gamma$ , is a measure of the change in shape of a fluid particle. More specifically, shear strain is denoted by  $\gamma_{21}(t, t_0)$  where the indices '21' refer to strain generated from shear flow in which  $e_2$  planes slide over each other in the  $e_1$  direction. As

stated previously, the strain response is unsteady and thus the two arguments to  $\gamma_{21}$  denote a function of time,  $t$ , and a reference time,  $t_0$ . For simplicity,  $t_0$  will be taken as  $t = 0$  thereby abbreviating  $\gamma_{21}(t, t_0)$  as merely  $\gamma_{21}(t)$ . Over short time periods, the shear strain is defined as

$$\gamma_{21}(t) \equiv \frac{\partial u_1}{\partial x_2}, \quad (23)$$

where  $u_1(t)$  is the displacement function in the  $x_1$  direction. The displacement function tracks the position of a fluid particle with respect to time.

A physical interpretation of  $\partial u_1 / \partial x_2$  can be made by considering the shear flow diagram in Figure 4.

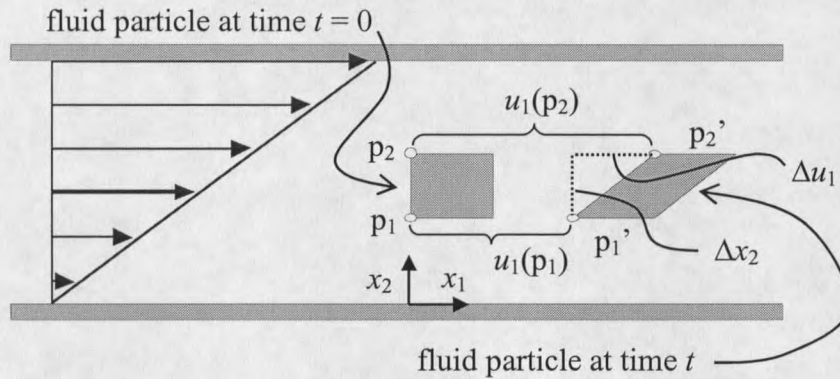


Figure 4. Physical Interpretation of Shear Strain in Shear Flow. As the fluid particle moves forward in time, the displacement of  $p_2$  outstrips that of  $p_1$  due to the linear velocity field in the fluid.

Assume that the two points  $p_1$  and  $p_2$  lie on the  $x_2$  axis as shown at the reference time  $t = 0$ . For shear flow, one can assume that  $u_2$  and  $u_3$  are zero and the displacement function reduces to the 1-component of  $u$ . At time  $t$ , both particles have moved downstream to the points denoted by  $p_1'$  and  $p_2'$ . Note that the displacement of  $p_2$  is greater than that of  $p_1$  because  $p_2$  lies in a plane of higher velocity. Recall equation (23) and apply the simplification that  $\partial u_1 = \Delta u_1$  and  $\partial x_2 = \Delta x_2$  then

$$\gamma_{21}(t_i, t_{i+1}) \approx \frac{\Delta u_1}{\Delta x_2} = \dot{\gamma}_{21} \Delta t. \quad (24)$$

This equation is valid for any small strain interval from  $t_i$  to  $t_{i+1}$ . If the flow is further limited to steady state conditions ( $\partial \gamma_{21} / \partial t = \text{constant}$ ) the strain over a larger interval from 0 to  $t$  can be represented by summing the individual small strains in equation (24), or

$$\gamma_{21}(0, t) = \sum_{i=0}^{N-1} \gamma_{21}(t_i, t_{i+1}) = N \Delta t \dot{\gamma}_0 = t \dot{\gamma}_0. \quad (25)$$

While equation (25) sufficiently describes shear strain for steady flow conditions, a creep test is clearly unsteady. As such, the derivative of shear with respect to time varies with time. Equation (25) must be modified to reflect this time dependence.

$$\gamma_{21}(t_1, t_2) = \sum_{i=0}^{N-1} \gamma_{21}(t_i, t_{i+1}) = \sum_{i=0}^{N-1} \Delta t \dot{\gamma}_{21}(t_{i+1}) \quad (26)$$

where  $\gamma_{21}(t_1, t_2)$  is the large strain between times  $t_1$  and  $t_2$ . Taking the limit of (26) as  $\Delta t$  approaches zero results in the integral of the time-dependent strain rate between  $t_1$  and  $t_2$ .

$$\gamma_{21}(t_1, t_2) = \int_{t_1}^{t_2} \dot{\gamma}_{21}(t') dt' \quad (27)$$

Equation (27) describes shear strain for unsteady flows such as creep.

It is important to note the implications of equation (27) on the creep experiments using torsional instruments such as the parallel plate rheometer. A comparison of equations (25) and (27) illustrate the practical differences between a creep test and a steady state shear rate test. The latter requires no complicated mathematics to determine the total shear strain during the test. The shear rate and test duration are simply prescribed while the stress response is measured directly. Conversely, the creep test requires that the shear rate is measured instantaneously as a function of time and then integrated over the duration of the test. Historically, this was a disadvantage of rheometer creep experiments (TA Instrument Manual, AR1000). More recently computers have alleviated these difficulties by allowing for the integration of the discrete data by first fitting the data to an interpolating curve. A commonly used interpolating approach involves fitting the observed strain rate points to cubic splines.

Cubic spline interpolation is one of a class of interpolation methods that includes Newton Divided-Difference polynomials, and Lagrange Interpolating polynomials.

While polynomial methods can be used to perfectly fit discrete data to curves, they are impractical for this application for two reasons: fitting  $n$  data points would necessitate deriving an  $n-1$  order interpolating polynomial; and oscillations between data points can be extreme and result in overshoot (Chapra and Canale, 1998). As an alternative, cubic splines are third order curves connecting each adjacent pair of data points. They often provide a smoother fit and prevent unwarranted, abrupt oscillations between points.

In the context of strain rate data points, the objective is to derive  $n$  third order polynomials for the interval between each adjacent pair of  $n+1$  ( $i = 0, 1, 2, \dots, n$ ) strain rate points. This interpolating polynomial takes the form

$$\Psi_i(\dot{\gamma}) = C_i^3 t^3 + C_i^2 t^2 + C_i^1 t + C_i^0, i = 1, 2, \dots, n, \quad (28)$$

where  $\Psi_i$  is the interpolating cubic function for  $\partial\gamma/\partial t$  over the  $i^{\text{th}}$  interval, and  $\{C_i\}$  is the set of 4 coefficients of the interpolating polynomial for the  $i^{\text{th}}$  interval. There are  $4n$  unknown constants (i.e. interpolating function coefficients) and accordingly,  $4n$  conditions are required to solve for the unknown coefficients. These are:

1.  $\Psi$  values must be equal at all interior strain rate data points. This results in  $2n-2$  conditions;
2. The strain rate interpolating function,  $\Psi$ , must pass through the end points ( $i = 0, n$ ) or  $\Psi_0(\gamma_0) = (\partial\gamma/\partial t)_0$  and  $\Psi_n(\gamma_n) = (\partial\gamma/\partial t)_n$ . This produces 2 additional conditions;

3.  $\partial\Psi_i/\partial t$  at all interior data points ( $i = 1, 2, \dots, n-1$ ) must be equal. This yields  $n-1$  conditions;
4.  $\partial^2\Psi_i/\partial t^2$  at all interior data points ( $i = 1, 2, \dots, n-1$ ) must be equal. This yields another  $n-1$  conditions. Finally;
5.  $\partial^2\Psi_i/\partial t^2$  at the end points ( $i = 0, n$ ) must be zero. These are the last 2 conditions.

With these  $4n$  unknown coefficients and  $4n$  conditions the systems of equations can be solved numerically resulting in a continuous expression for strain rate as a function of time. Combining (27) and (28) results in

$$\gamma_{21}(t_1, t_2) = \int_{t_1}^{t_2} \Psi(t') dt'. \quad (29)$$

Equation (29) is then used to calculate strain from unsteady flow, parallel plate creep experiment data over any time period defined by  $t_1$  and  $t_2$ .

Shear creep tests are performed by applying a low constant shearing stress,  $\tau_0$ , to a sample while recording the resulting strain rate. Strain rate is then numerically converted to a strain response using equation (29). The character of the shear strain response over time is distinctly different for three idealized behaviors; elasticity, Newtonian viscosity, and viscoelasticity. If a low stress is applied to an elastic sample, it will respond by deforming almost instantaneously to a new position, and then stopping. When the stress is removed, the sample will immediately recover its original dimensions. If the stress is

placed on a viscous liquid, it will deform continuously, the rate of deformation being inversely proportional to the viscosity of the liquid. When the stress is removed the sample will cease to deform, and will not demonstrate any recovery. Many materials exhibit both elastic and viscous properties. These viscoelastic materials demonstrate properties that are some way between these extreme forms of behavior. When a stress is applied to a viscoelastic material, the deformation will be retarded. When the stress is removed, the recovery will also be time-dependent.

### Material Functions

Material functions describe relationships between kinematic parameters and rheological behavior (Morrison, 2001). They also provide a basis for comparison among different samples and materials that exhibit a non-Newtonian flow response. In the context of shear creep, the material function is called the creep compliance,  $J(t, \tau_0)$ , which relates sample strain to prescribed shear stress,

$$J(t, \tau_0) \equiv \frac{\gamma_{21}(0, t)}{-\tau_0}, \quad (30)$$

where  $\tau_0$  is the constant prescribed shear stress and  $\gamma_{21}(0, t)$  is the shear strain calculated from observations of  $\partial\gamma/\partial t$  using equation (29). Several other material functions are related to (30).

As the creep experiment progresses during the loading stage, the strain will become linear. In other words, the slope of the compliance curve approaches the ratio of the steady-state shear rate (ie.  $\partial\gamma/\partial t = \text{constant}$ ) over the negative of the prescribed shearing stress,  $\tau_0$ . Note that conceptually this ratio is the inverse of the steady-state shear viscosity,  $\eta(\partial\gamma/\partial t_\infty)$ . A material function known as the steady-state compliance or  $J_s$  is the difference between the compliance function and the ratio of time over the steady-state shear viscosity.

$$J_s \equiv J(t, \tau_0) - \frac{t}{\eta(\dot{\gamma}_\infty)} \quad (31)$$

Typically,  $J_s$  is estimated by extrapolating the linear portion of the compliance function back to the  $J(t, \tau_0)$  axis at time  $t = 0$ .

At the start of the recovery stage of a creep test, the constant shear stress,  $\tau_0$ , is removed and the material begins to recoil or recover. The recovery compliance,  $J_r$ , also known as the recoil function, is another common used material function.  $J_r$  is an expression of the material's ability to spring back in the direction opposite to the initial flow. It is defined as follows:

$$J_r(t', \tau_0) \equiv \frac{\gamma_r(t')}{-\tau_0} \quad (32)$$



where  $t'$  is the transformed time variable measured as  $t - t_S$ ;  $t_S$  is the time at which the constant shear stress is removed; and  $\gamma_r$  is the time-dependent recoil strain. The ultimate recoverable compliance,  $J_\infty$ , is a material function that is an expression of the shear strain once the sample has completely come to rest. The following formula is derived by taking the limit of equation (32) as  $t'$  approaches infinity.

$$J_\infty = \lim_{t' \rightarrow \infty} J_r(t', \tau_0) = \frac{\gamma_\infty}{-\tau_0} \quad (33)$$

where  $\gamma_\infty$  is the ultimate shear strain recovered as  $t'$  approaches infinity.  $J_\infty$  can be approximated by extrapolating the compliance function back to  $t = 0$  once  $\partial J / \partial t$  asymptotically approaches zero.

Finally, the zero-shear viscosity,  $\eta_0$ , is another commonly measured material function. This value is a measure of the steady state effective viscosity as time approaches infinity.  $\eta_0$  is proportional to the applied shearing stress and inversely proportional to the shear rate attained at steady state.

$$\eta_0 = \frac{-\tau_0}{\dot{\gamma}_\infty} \quad (34)$$

### Creep Testing

As described in the previous sections, the processes of biofilm deformation and failure are largely influenced by the applied forces and the material properties of the bacterial biofilm. Rheometry provides a means of investigating this influence. As a preliminary step towards the development of a biofilm constitutive law, rheometry was used to make a number of qualitative assessments about biofilm behavior and provide a framework for comparison between samples by quantifying biofilm material functions. The goal was to grow viable mixed biofilm cultures for shear creep testing in a rotating disk rheometer. To determine the viscoelastic properties of the microbial biofilm samples, creep tests were performed on a research-grade rheometer (TA Instrument's AR1000) in the Montana State University Mechanical Engineering Department's Material Testing Laboratory.

### Experimental Design

Testing a biological material like biofilm in a rheometer was a novel application of this instrument, and as such, methods for this procedure were developed. There were two primary concerns in the development of these methods. Unlike the flow-through study conducted by Stoodley et al. (1999b), the use of an electronic rheometer precluded any in-situ testing. That is, there was no way biofilm could have been grown in the rheometer itself. Therefore, the samples had to be grown and tested separately. The first concern was ensuring that disturbance of the samples during transport and transference

from the growth reactor to the rheometer was minimized. Rheometer disks or 'geometries' are 40mm anodized aluminum disks that comprise the top plate in this torsional flat-plate instrument. These disks are tapped for a machine screw and can be quickly and easily removed and installed back into the rheometer. Therefore, a new growth chamber was constructed around the use of rheometer disks. In this way, samples were grown on the disk directly, removed from the growth reactor and re-installed into the rheometer without disturbing the biofilm itself. Additionally, this method allowed for the quick removal and re-installation which also minimized time of transport. The second concern was ensuring that the samples could be grown under varying stress conditions and that samples were loaded during testing in the same fashion as they were during growth. Samples were grown on disks in a flow-through growth chamber while subjected to a constant, fluid-induced, shear stress induced by the rotation of the disks. Shear creep tests were performed on the samples and loaded (rotated) in the same direction as during growth. By adhering to these guidelines, a degree of consistency with the flow-tube experiments was maintained.

### Biofilm Growth Reactor

The biofilm reactor consisted of a polycarbonate growth chamber (Fig. 5, 6), a 1-amp electric motor, a peristaltic pump, flexible tubing and two 5-gallon glass carboys. The growth chamber was approximately 43 cm long, 24 cm wide and 27 cm high. The top cover of the growth chamber was removable and contained a sealed bearing through

which the drive shaft was held. A bridge assembly, which housed two secondary drive pulleys, was anchored to the bottom of this top cover.

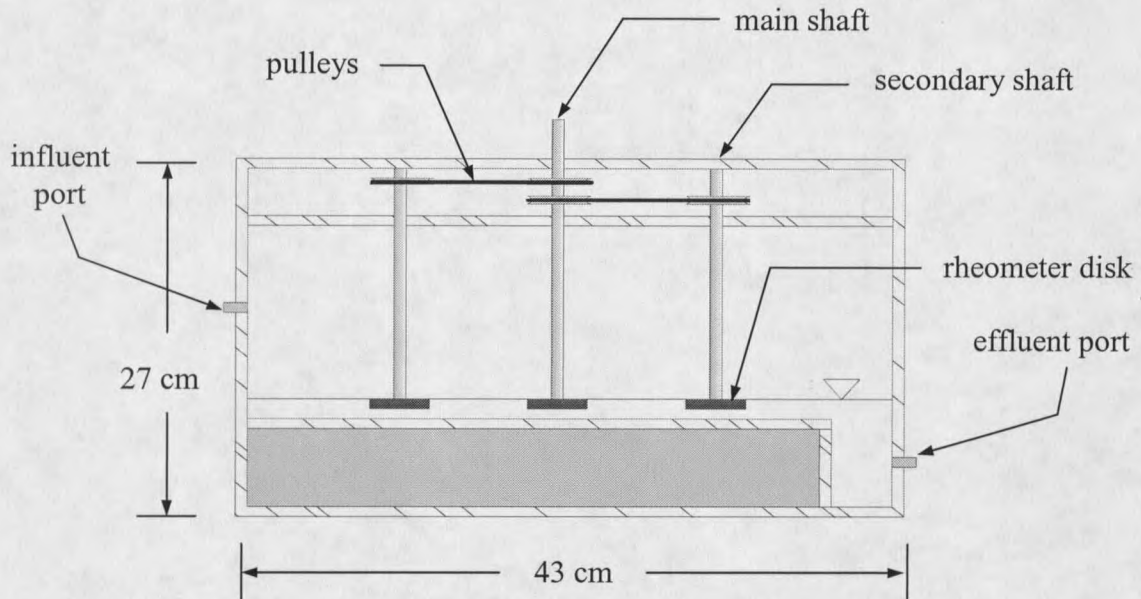


Figure 5. Schematic of the Biofilm Growth Chamber, Drive Shafts, Pulleys and Rheometer Disks.

The main drive shaft was connected to the electric motor and rotated the secondary shafts through the pulley system. Secured to the end of each of the three drive shafts was a 40 mm diameter anodized aluminum rheometer disk or 'geometry'. These disks were positioned 5mm above the chamber floor. The rotational speed of the spinning disks was controlled by the electric motor. These spinning rheometry disks



introduced a shear-driven rotational flow around each shaft. The angular velocity of the shaft/disk was held constant at 20 rpm throughout the experiments.

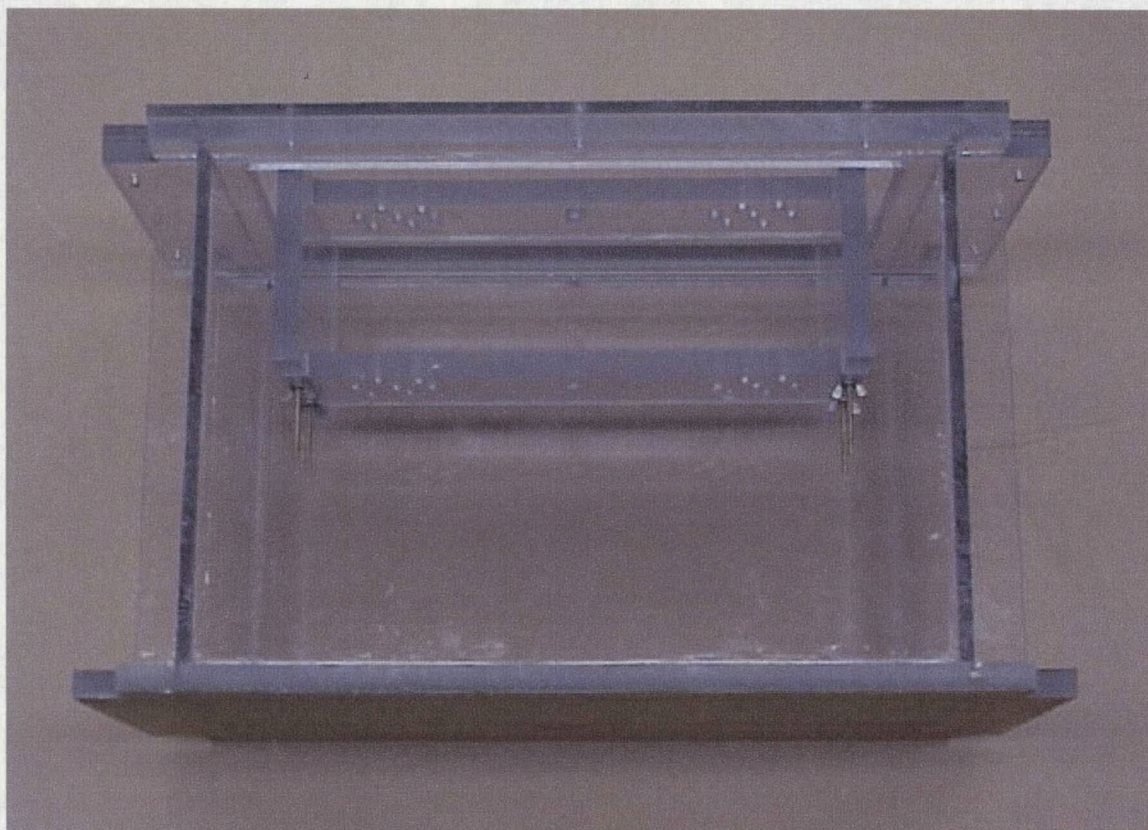


Figure 6. Side View of the Biofilm Growth Chamber.

A research grade nutrient broth (Luria-Bertani  $1/100$ , 2g/l, strength LB) was used to grow the biofilms. This nutrient supply was stored in one of the glass carboys and pumped into the front of the growth chamber by a peristaltic pump (model here). Throughout the experiment, a constant flow rate of 2.0 ml/min was maintained. This flow rate, which translated into a hydraulic residence time ( $\theta_H$ ) of approximately 7.8



hours, balanced a high turnover of the bulk fluid with a relatively low lateral flow velocity. This balance was designed to minimize nutrient consumption by planktonic cells, and consequently favor the growth of attached biofilms, without creating a substantial lateral velocity that would inhibit the growth of attached microorganisms. After passing each of the 3 rheometer disks, the flow collected in an effluent reservoir at the rear of the reactor. Gravity discharge from this reservoir maintained the nutrient depth at a constant level of approximately 5 mm above the growth surface. The reactor, as built, is shown in Figure 7.

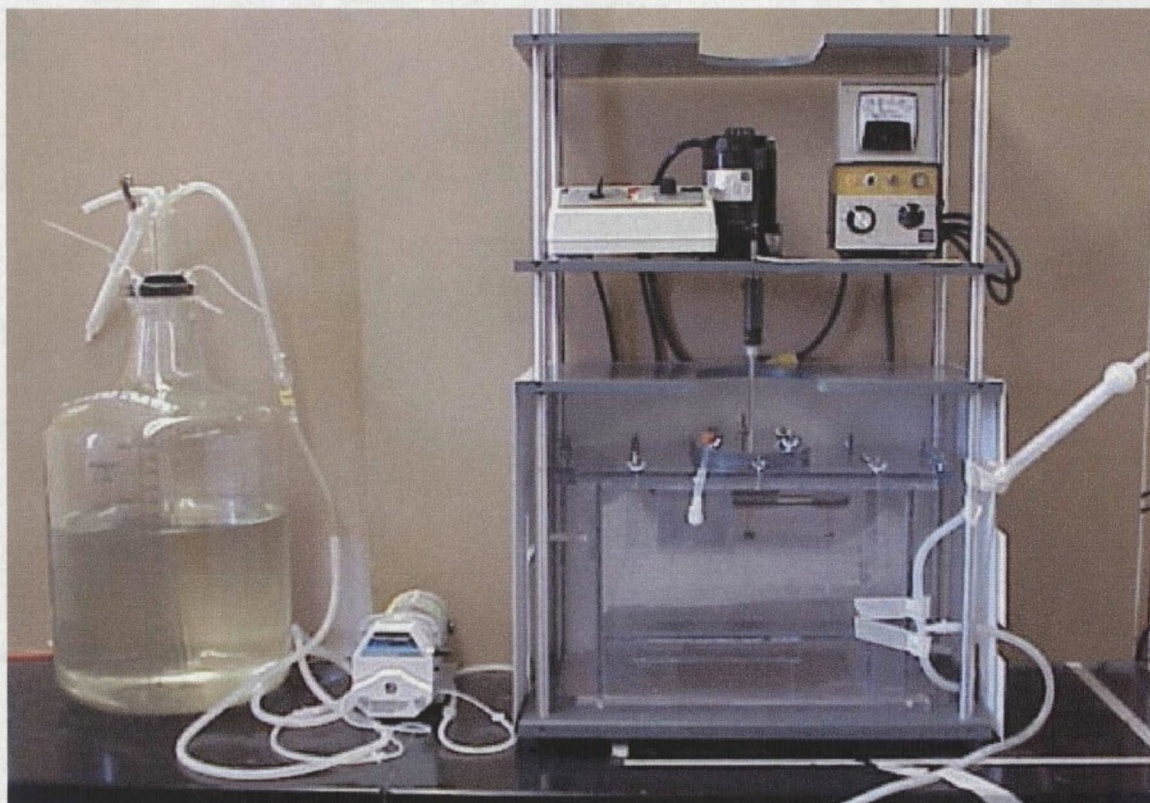


Figure 7. View of the Biofilm Growth Reactor, as Built.

The chamber was inoculated with a 5 ml sample of water from a local pond. The inoculum was injected into the reactor through a sealed port tapped into the front face of the chamber. In order to limit algal growth, the entire growth chamber was covered with opaque cardboard sheeting. The system was operated as a batch culture for the first 24-hour period (day 1), to allow for the attachment of cells to the disks. On day 2 the influent was turned on and the system operated as a continuous flow system. After 12 days of nutrient-fed growth, a thin microbial biofilm layer covered each spinning rheometer disk. Collectively, each disk and attached biofilm was designated as a single sample for testing in the rheometer.

#### Removal, Transport and Installation of Samples

Rheometer disks and attached biofilm samples were removed for testing from the growth chamber after 12 days of growth. A biofilm of irregular thickness grew on each of the rheometer geometries. Disks were carefully unscrewed from the drives shafts and removed from the growth chamber. Care was taken to move the disks gently so as not to compromise the structure of the biofilm growing on the disk surface. Once removed from the growth chamber, they were placed face-up and submerged in a beaker containing Ringers, a non-alkalizing salt solution used to stabilize the biofilms for transport to the material-testing laboratory. At the laboratory, disks with attached biofilms were installed into a TA Instruments, Inc., AR1000 Rheometer. The time in transport from growth chamber to material-testing laboratory was routinely less than 15 minutes. Installation and testing for each sample took less than 15 minutes. Since the

growth chamber produced 3 samples per growth period, the longest any one sample spent between removal and testing was less than 1 hour. This helped to minimize the effect that the Ringers solution or bacterial decay in the absence of a nutrient supply may have had on material properties of the cultures. Once installed, the samples were oriented such that the direction of rotation of the rheometer disk during growth and the direction of the applied stress for the creep test were consistent.

Again, in-situ testing was impossible, but every effort was made to mimic the growth conditions during testing. Because the samples were grown in an aqueous environment, testing the culture in the dry environment between the aluminum geometry and the steel instrument base plate would affect the results. Therefore, the rheometer was modified to ensure the samples were tested in a similarly wetted condition. A water barrier was formed on the base plate of the AR1000. This water dam (Fig. 8) was fashioned around the edges of the instrument base plate using modeling clay. The resulting reservoir was then filled with approximately 0.25 N of Ringers solution to prevent de-watering of the sample during testing. To minimize edge effects of Ringer's solution in contact with the disk, the fluid level was maintained high enough to submerge the attached biofilm without submerging the tops of the rheometer geometries.



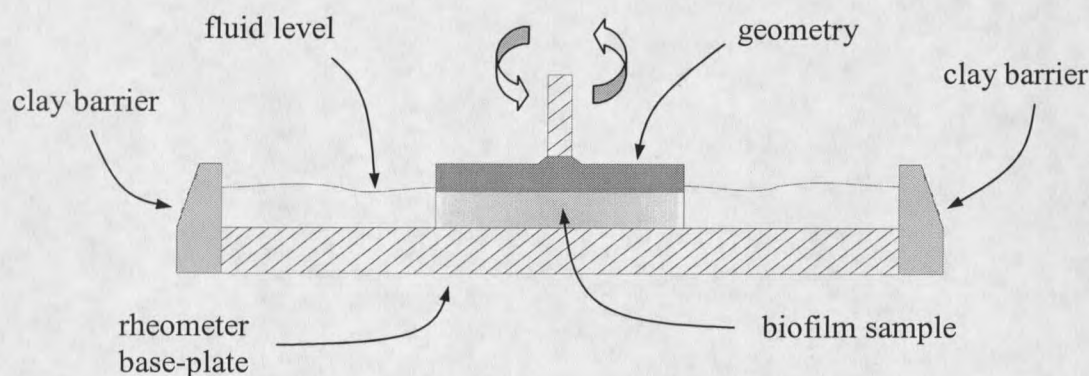


Figure 8. Side View of the AR1000 Rheometer Including Water Dam. Water dam is created by a clay barrier that encircles the lower base plate. The reservoir is filled with a Ringers solution to prevent dewatering of the biofilm samples.

Testing this biological material using samples that were grown on rheometer disks represented a highly novel application of the AR1000. As such, several rounds of pre-testing of biofilm samples were performed in the development of the following methods: In random order, each sample was subjected to shear creep (or retardation) and then recovery. Only samples that had biofilm covering the entire surface of the rheometer disks were selected for creep tests. The geometry gap, the distance between the aluminum disk and the instrument's base plate, was set to  $30\text{ }\mu\text{m}$ . This gap was based on approximations of average biofilm thickness made with a fluorescent microscope. One randomly selected sample was examined using an imaging microscope (Fig. 9).

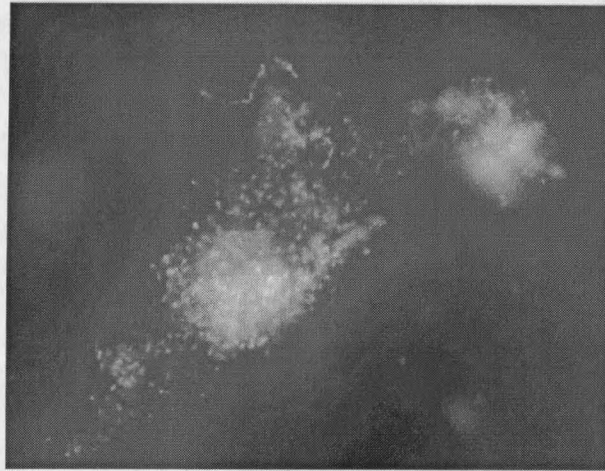


Figure 9. Epi-fluorescent Image of the Top of the Mixed Species Biofilm Grown on a Rheometer Disk Showing Bacterial, Rods, Cocci, and Chains. Fluorescent stain identifies concentrations of bacterial cells. Image width is approximately 150  $\mu\text{m}$ .

### Shear Creep Tests

The thickness of the biofilm varied from approximately 35 $\mu\text{m}$  to 50 $\mu\text{m}$ . A constant torsion yielding a shear stress of 0.1  $\text{N/m}^2$  at the disk's edge was applied for a 3-minute retardation (loading) period, followed by a 3-minute recovery (unloading) period ( $\tau_0=0$ ). Approximately 900 measurements of strain rate versus time were taken over each six-minute test. These values of strain rate were converted into shear strain using equation 29. Additionally, to determine if the material behaved in a linear viscoelastic manner, one sample was tested again at a load of 0.5  $\text{N/m}^2$  after an approximately 10 minute long recovery period. The creep responses and compliance functions of these samples are presented in Figures 10 through 17.

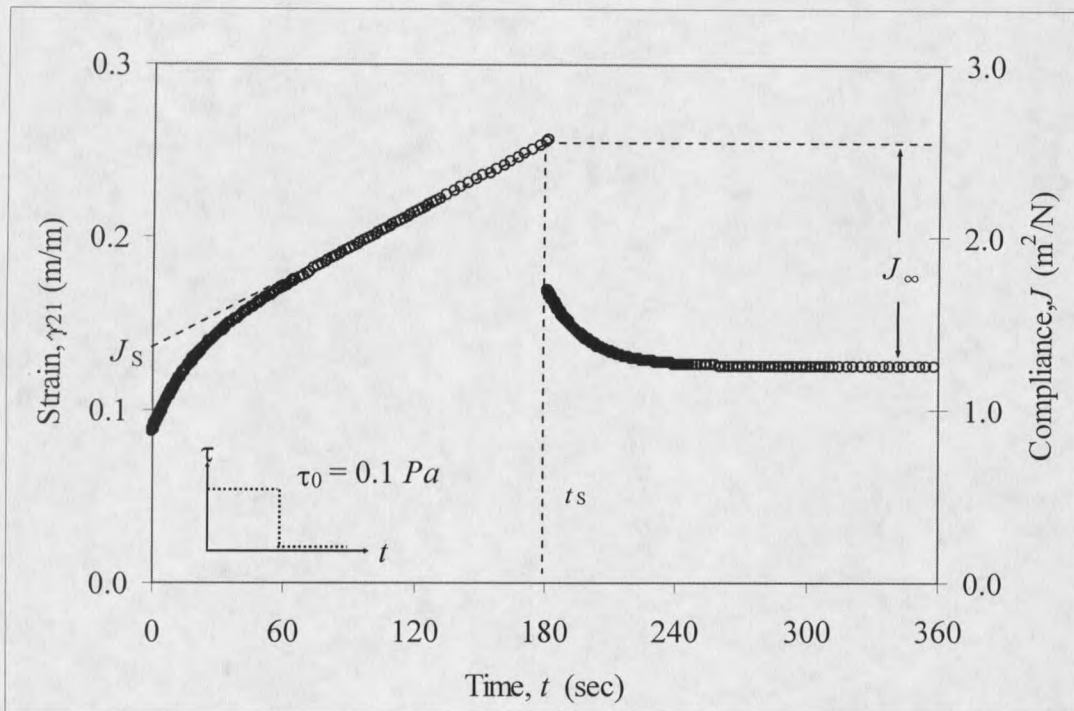


Figure 10. Shear Creep Strain Response and Compliance Function for Sample 1.

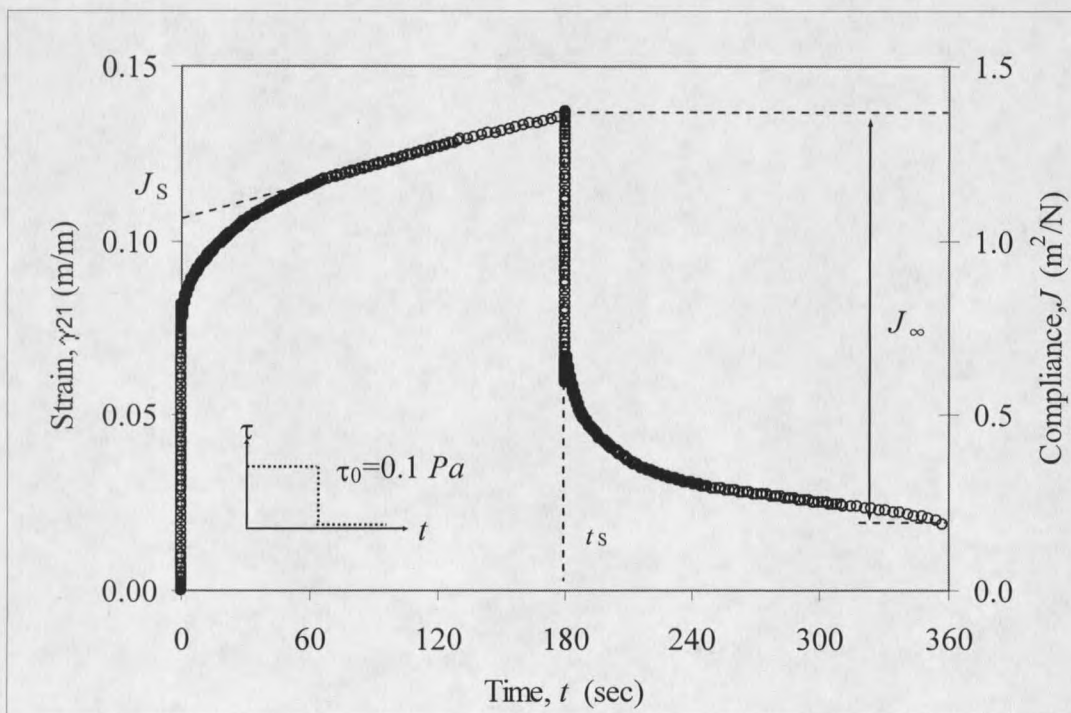


Figure 11. Shear Creep Strain Response and Compliance Function for Sample 2.

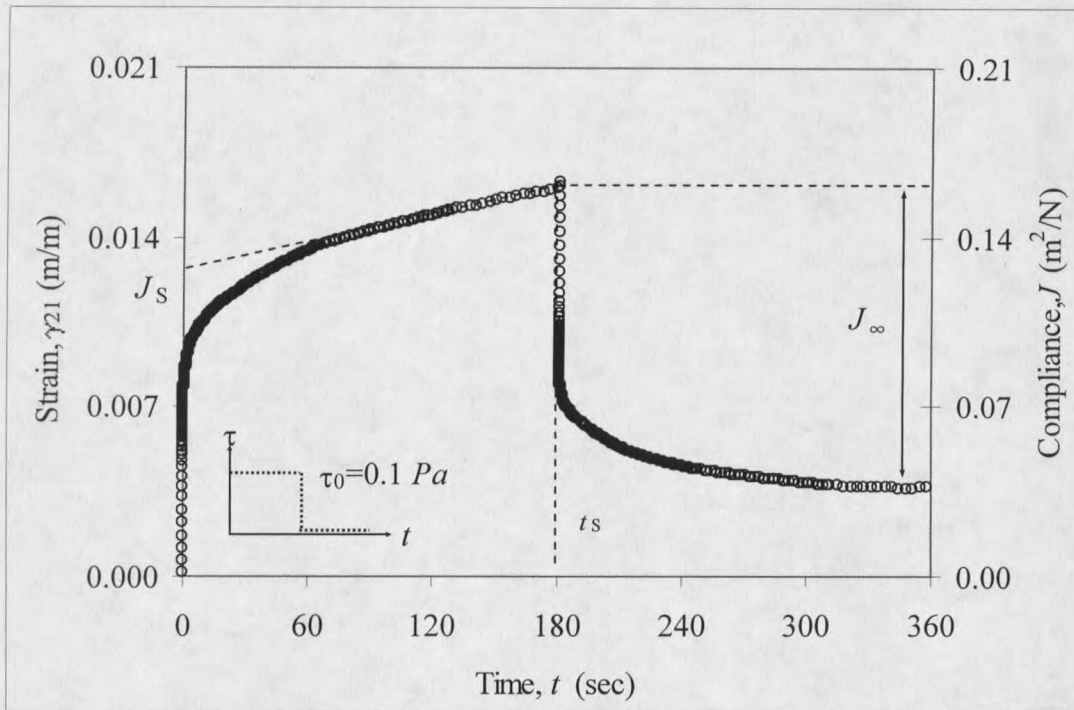


Figure 12. Shear Creep Strain Response and Compliance Function for Sample 3a.

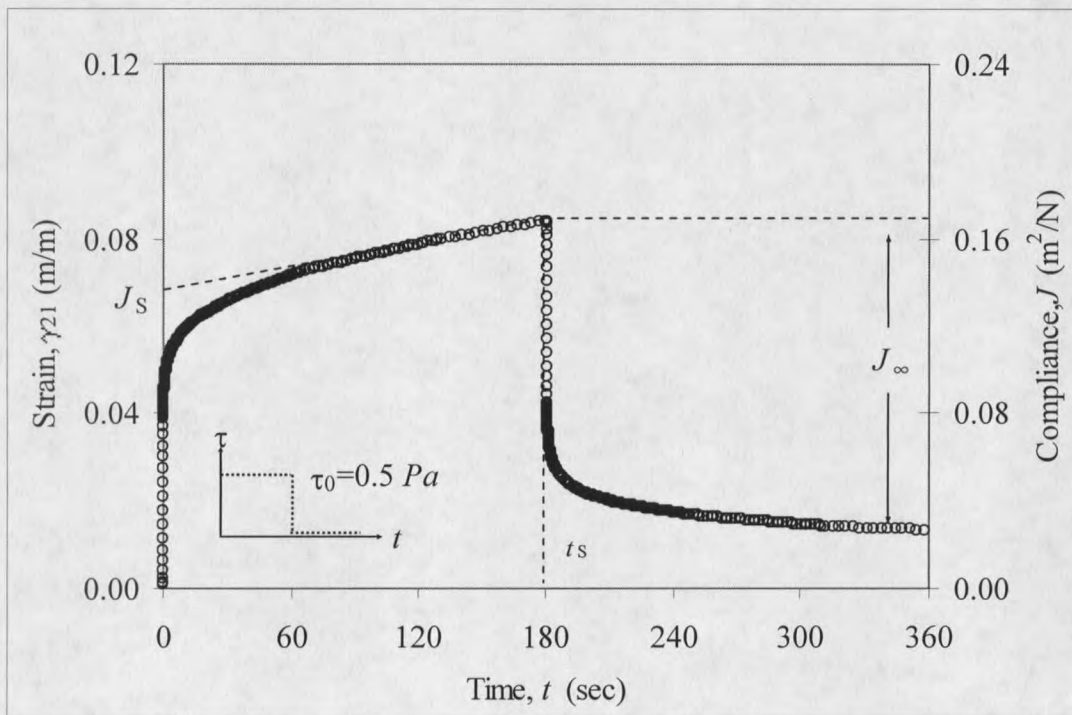


Figure 13. Shear Creep Strain Response and Compliance Function for Sample 3b.



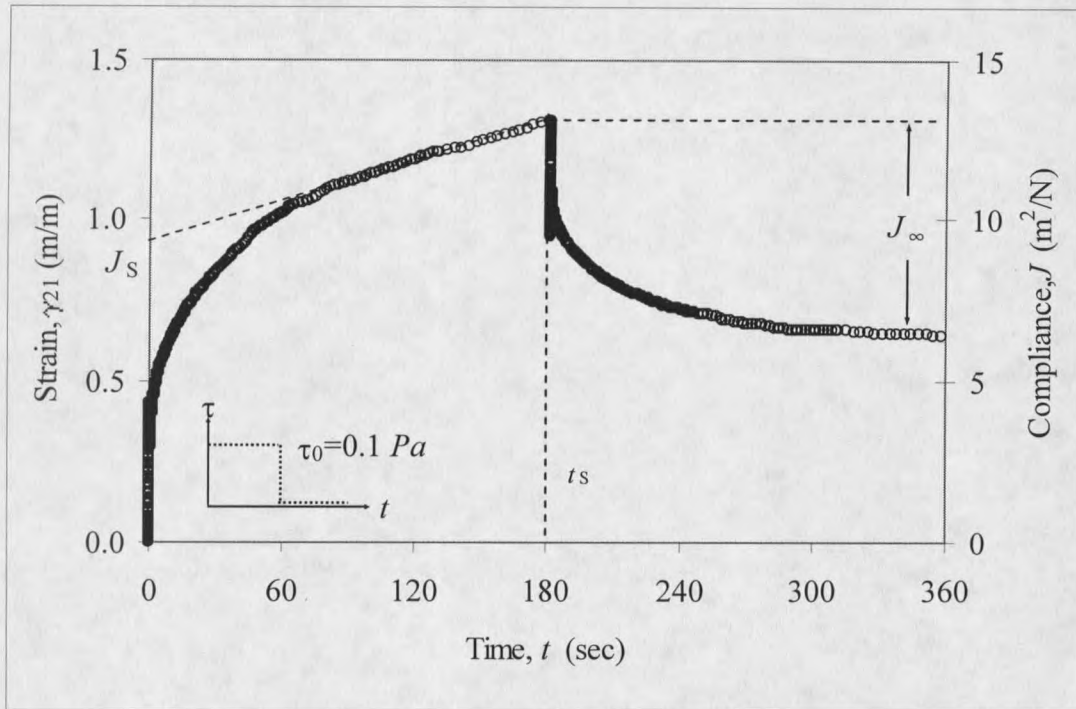


Figure 14. Shear Creep Strain Response and Compliance Function for Sample 4.

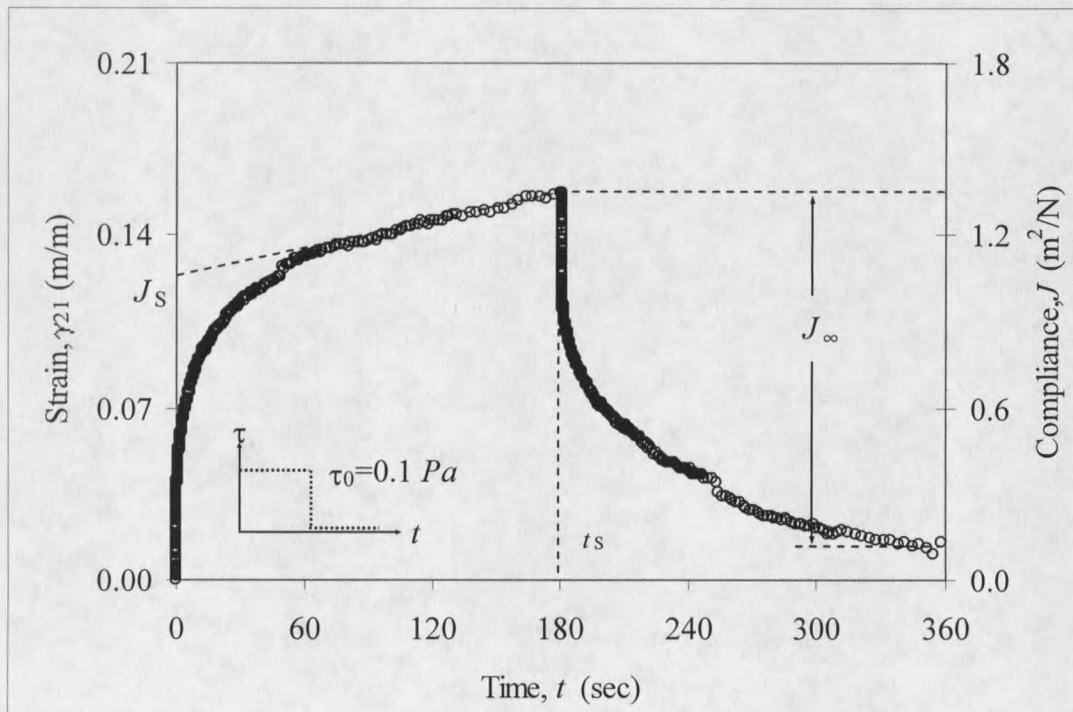


Figure 15. Shear Creep Strain Response and Compliance Function for Sample 5.

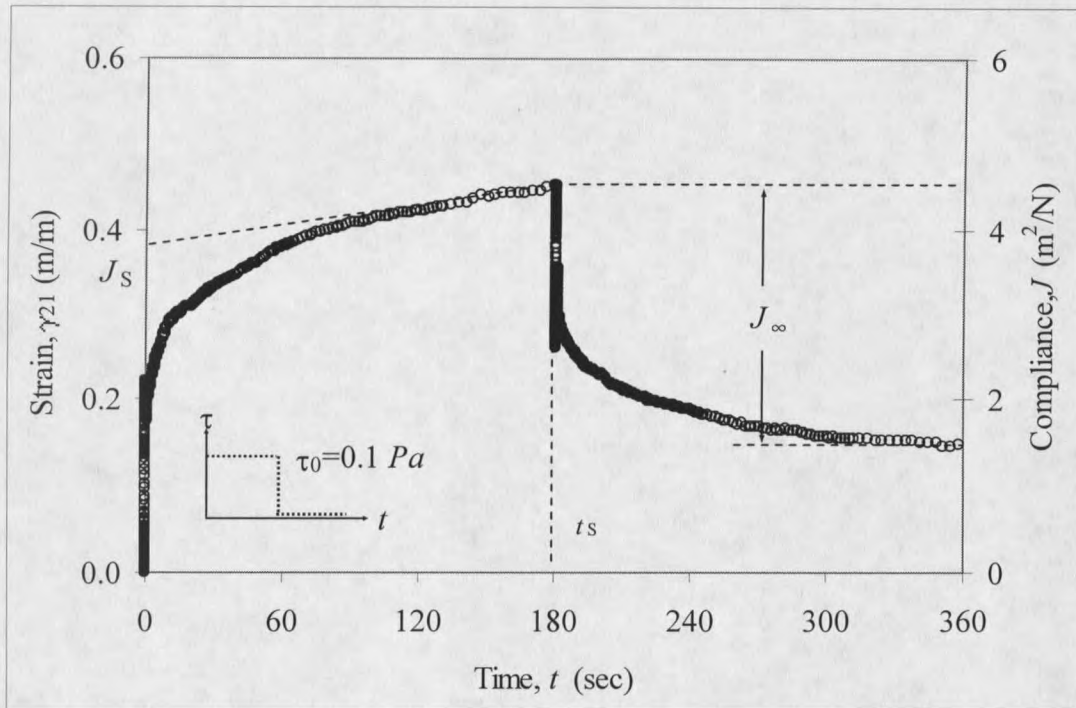


Figure 16. Shear Creep Strain Response and Compliance Function for Sample 6.

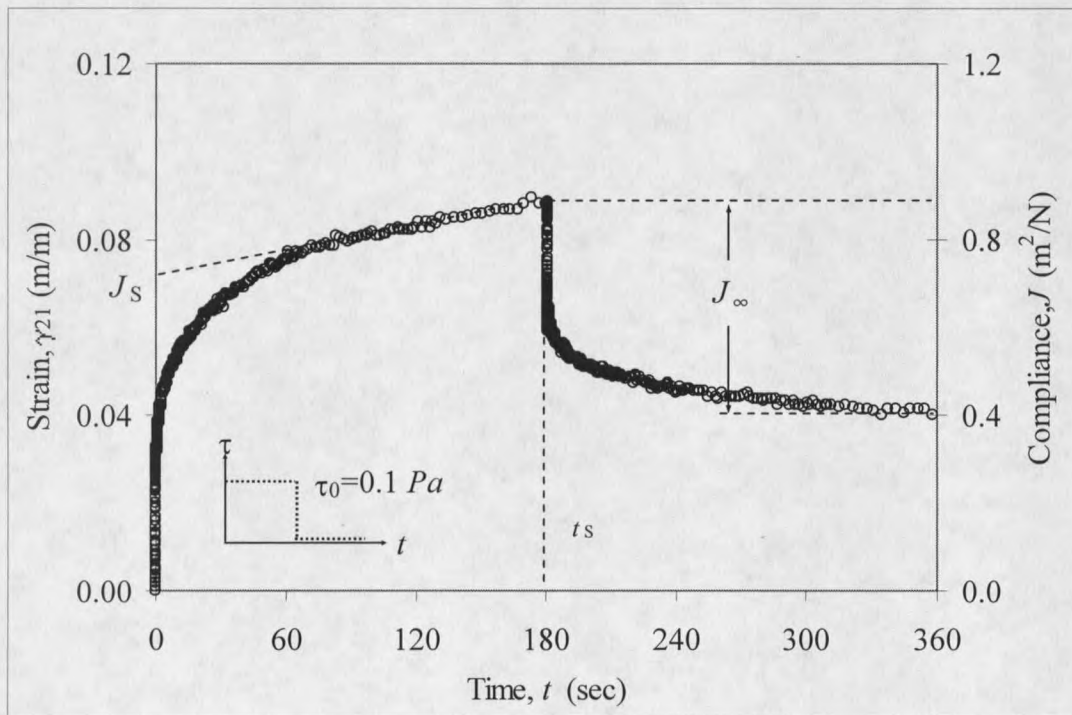


Figure 17. Shear Creep Strain Response and Compliance Function for Sample 7.

### Discussion of Results

The data from the creep tests demonstrated a number of interesting qualities. The creep response of all 7 samples was clearly non-Newtonian. Consider the loading or retardation stage of Figure 14. A Newtonian response to constant loading ( $\tau_0 = 0.1$  Pa) would have resulted linear strain response. In fact, the slope of strain-time curve in such a response would be proportional to the fluid viscosity (Findley et al, 1975).

$$\gamma(t) = \frac{\tau_0}{\eta} t \quad (35)$$

where  $\eta$  is also the effective dynamic viscosity,  $\mu_{\text{eff}}$ . Excluding the instantaneous strain response ( $\gamma \approx 0.4$  m/m) observed in the test on sample 4, the effective viscosity of the biofilm decreases from effectively infinite at  $t = 0$  to approximately  $\mu_{\text{eff}} \approx 48.84$  N-s/m<sup>2</sup> at  $t = t_s = 180$  seconds. This tendency of the fluid viscosity to decrease is referred to as shear-thinning and is one of the more commonly observed non-Newtonian effects (Morrison, 2001). Shear-thinning does not necessarily imply viscoelasticity, however. While Sample 4 demonstrated this phenomenon quite clearly, shear-thinning was observed in all 7 samples.

Another commonly observed non-Newtonian response is that of a yield stress. A yield stress is the magnitude of an applied load below which the material will not deform. Fluids that express this behavior are known as Bingham fluids. None of the 7 samples demonstrated a yield stress within the range of 0.1 to 0.5 Pa.

Newtonian fluids deform under stress and when the load is removed, the deformation ceases immediately. Another non-Newtonian effect is that of fluid memory. Fluids with memory deform, but unlike Newtonian fluids they do not dissipate internal stresses immediately. As a result, these materials will deform in the opposite direction once the applied load has been removed. Consider Figure 15 which illustrates this effect by Sample 5. After 3 minutes of loading, the recovery stage of the creep test begins when the shear stress is reduced from  $\tau_0 = 0.1$  Pa to zero. Immediately the material begins to recoil in the opposite direction. Initially, this strain rate is instantaneous, but quickly begins to decrease until it asymptotically approaches a constant value or zero-shear viscosity. Defining recovery capacity,  $R$ , as the ratio of maximum strain to recovered strain at  $t = \infty$ , sample 5 demonstrated the greatest recovery capacity on a percentage basis, though all samples exhibited recovery to some degree (Table 1).

Table 1. Steady-state Viscosity and Recovery Capacity of Biofilm Samples

| Sample | $\tau_0$<br>$N/m^2$ | $\eta_s$<br>$N \cdot s/m^2$ | $\gamma(t_s)$<br>$m/m$ | $\gamma(t=\infty)$<br>$m/m$ | $\gamma_\infty$<br>$m/m$ | $R$<br>% |
|--------|---------------------|-----------------------------|------------------------|-----------------------------|--------------------------|----------|
| 1      | 0.1                 | 144.9                       | 0.256                  | 0.125                       | 0.131                    | 51.3     |
| 2      | 0.1                 | 667.2                       | 0.137                  | 0.0182                      | 0.118                    | 86.7     |
| 3a     | 0.1                 | 5599                        | 0.0163                 | 0.00370                     | 0.0126                   | 77.5     |
| 3b     | 0.5                 | 5828                        | 0.0842                 | 0.0134                      | 0.0708                   | 84.1     |
| 4      | 0.1                 | 48.84                       | 1.31                   | 0.641                       | 0.668                    | 51.0     |
| 5      | 0.1                 | 473.9                       | 0.157                  | 0.0106                      | 0.147                    | 93.3     |
| 6      | 0.1                 | 230.2                       | 0.452                  | 0.149                       | 0.303                    | 67.0     |
| 7      | 0.1                 | 1011                        | 0.0885                 | 0.0396                      | 0.0489                   | 55.3     |



Finally, it is important to note that fluid memory is a decidedly viscoelastic effect. That is, the material has the capacity for elastic deformation and recovery as well as viscous flow.

These results support the conclusions by previous investigators (Characklis, 1980, Stoodley, 1999a) that biofilm is a viscoelastic material. Further distinction can be made, however. A viscoelastic material can be classified as either a viscoelastic fluid or a viscoelastic solid. A solid demonstrates complete recovery and a limited strain (i.e.  $\eta_s \rightarrow 0$ ) under a finite applied load. Conversely a viscoelastic fluid will, by definition, continue to flow (i.e. deform) under a finite stress. Furthermore, a solid given enough time will completely recover; a fluid will not. While the creep test durations were limited by the biofilm's tendency to degrade in the absence of a nutrient supply, the 6 minute long experiments were sufficiently long to classify this mixed biofilm culture as a viscoelastic fluid. Consider the retardation stage of each curve (Samples 1, 2, 3a, 3b, ... 7). All 8 tests show that the steady state viscosity is substantially greater than zero. In other words, this culture demonstrates a capacity for continuous, unlimited deformation at a constant applied stress of 0.1 Pa or 0.5 Pa. Evidence of the fluid nature of this material appears in the recovery stage of the creep curves. In all tests, save Sample 2, extrapolation of the recovery curves clearly indicates that the material will never completely recover to its original position. This capacity for a viscous loss is also indicative of a fluid.

Sample 2 (Figure 11) was the exception to this tendency. At approximately 338 seconds into the creep experiment, or more specifically 22 seconds remaining in the

recovery stage, the slope of the creep curve decreases significantly. This abrupt change in the shear rate is observed in no other sample. From  $t = 314.8$  seconds to 337.7 seconds a zero-shear viscosity of  $1274.2 \text{ N/m}^2\text{s}$  is observed. From  $t = 337.7$  seconds until the end of the test, the zero-shear viscosity decreases by half to  $\eta_0 = 556.2 \text{ N/m}^2\text{s}$ . While it is possible that this behavior is an expression of some other time-dependent material response, this was not observed in any other sample. Therefore, the most likely explanation of this change in behavior is wall slippage. In a creep test wall slippage occurs when the adhesion between the sample and the rheometer base-plate fails, in part or completely. Under these circumstances, internal stresses in the biofilm act on a now weakened biofilm/plate surface and force the biofilm to slip. This registers in the test as an increase in shear rate. This single occurrence was treated as an anomaly.

The compliance functions,  $J(t)$ , for Samples 1 through 7 were calculated using equation (30) and are presented in Figures 10 through 17, respectively. Compliance functions are independent of the applied stress and, accordingly, are useful in comparing creep responses from different samples. Ideally, all 8 tests should have produced 8 nearly identical compliance curves. Figure 18 is a combination graph showing all 8  $J(t)$  curves during the creep phase of the tests only.

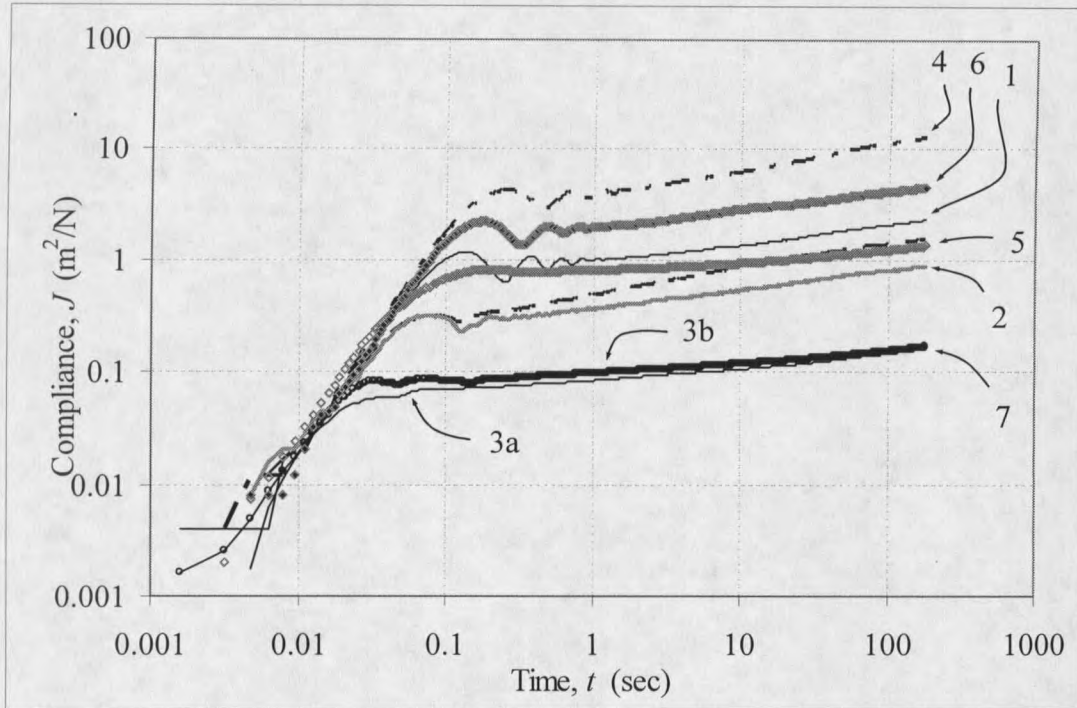


Figure 18. Combination Graph of Biofilm Compliance Curves. All eight samples (1, 2, 3a, 3b, 4, .. 8) are shown.

Clearly, there is considerable variation in the magnitude of the compliance functions. Recalling equation (30), the compliance can be regarded as a material capacity for deformation normalized against applied force. Sample 3a produced the smallest  $J(t)$  capacity for deformation while Sample 4 produced the greatest. Steady-state compliance values were similarly varied.  $J_s$  for each sample was calculated by extrapolating the linear portion of the retardation stage back to the intercept as per equation (31). These values are presented in Table 2.

Table 2. Steady-state Compliances of Biofilm Samples.  $J_s$  was calculated from the linear portion of  $J(t)$  defined by the two points  $[t, \gamma_{21}]_1$  and  $[t, \gamma_{21}]_2$ .

| Sample | $\tau_0$<br>N/m <sup>2</sup> | $[t]_1$<br>sec | $[\gamma_{21}]_1$<br>m/m | $[t]_2$<br>sec | $[\gamma_{21}]_2$<br>m/m | M<br>sec <sup>-1</sup> | b<br>m/m | $J_s$<br>m <sup>2</sup> /N |
|--------|------------------------------|----------------|--------------------------|----------------|--------------------------|------------------------|----------|----------------------------|
| 1      | 0.1                          | 181.0          | 0.256                    | 147.4          | 0.233                    | 0.0007                 | 0.131    | 1.314                      |
| 2      | 0.1                          | 176.9          | 0.135                    | 137.6          | 0.130                    | 0.0001                 | 0.109    | 1.09                       |
| 3a     | 0.1                          | 170.3          | 0.0158                   | 147.4          | 0.0154                   | 0.0000                 | 0.0128   | 0.128                      |
| 3b     | 0.5                          | 176.9          | 0.0838                   | 134.3          | 0.0801                   | 0.0001                 | 0.0686   | 0.137                      |
| 4      | 0.1                          | 176.9          | 1.30                     | 147.4          | 1.24                     | 0.002                  | 0.940    | 9.39                       |
| 5      | 0.1                          | 170.3          | 0.156                    | 137.6          | 0.149                    | 0.0002                 | 0.120    | 1.20                       |
| 6      | 0.1                          | 176.9          | 0.451                    | 150.7          | 0.440                    | 0.0004                 | 0.374    | 3.74                       |
| 7      | 0.1                          | 170.3          | 0.0881                   | 137.6          | 0.0849                   | 0.0001                 | 0.0713   | 0.713                      |

The placement of individual samples within the growth reactor was investigated as a cause of this variability in compliance. The previous section describes the experimental design of the reactor and specifically the growth chamber. Three rheometer disks were installed in the chamber at one time (i.e. one growth period). The disks were oriented in-line with respect to the direction of the flow induced by the nutrient supply. In light of the varying compliances, the nutrient supply, or lack thereof, could have affected the strength of the individual biofilm samples. However, this was not the case. Samples 1, 2, 3, and 6 were grown on the forward shaft, closest to the nutrient inflow port (see Figure 5). Variations in compliances among these samples were still considerable. Indeed, the mean difference in compliance between samples 3a and 1 was approximately an order of magnitude.

Biofilm material behavior, as described in Chapter 2, is dominated by the EPS. EPS is a polymeric substance and comparison to engineered polymers may shed light on the effect of temperature on biofilm behavior. Plazek (1965) performed shear creep tests on polystyrene at various temperatures. Temperature increases resulted in a clear vertical

shift in a log-log plot of  $J(t)$  versus  $t$ . The author was able to discern the compliance dependence on temperature and correct for it. While much care was taken to apply uniform growing and testing techniques to these samples, temperature was neither controlled nor recorded during either the growth period or during rheometry. Nevertheless, temperature likely played some role in biofilm strength. The vertical shift in  $J(t)$  observed in Figure 18 might be due to an increase in temperature during testing, growth or both. It is important to note that there are other likely causes of this variation in behavior including species differences and the inherent variability of biological materials.

Again, making the comparison between a polymeric biofilm and man-made polymers, the zero-shear viscosity measurements are related to a material's molecular weight. The zero-shear viscosity of each sample was estimated from their strain-time curves. A linear regression was performed on the tail end of each strain curve. The slopes,  $m$ , of the resulting linear equations were used with equation (34) to determine the zero-shear viscosities for all 8 mixed biofilm cultures. These values are presented in Table 3.

Table 3. Zero-shear Viscosities of Biofilm Samples. The shear rates,  $\partial\gamma/\partial t$ , were estimated as the slope,  $m$ , of the strain-time curves at  $t > 345$  seconds.

| Sample | $\tau_0$<br>N/m <sup>2</sup> | $m$<br>sec <sup>-1</sup> | $\eta_0$<br>N/m <sup>2</sup> s |
|--------|------------------------------|--------------------------|--------------------------------|
| 1      | 0.1                          | -3.86E-07                | 2.59E+05                       |
| 2      | 0.1                          | -1.92E-04                | 5.22E+02                       |
| 3a     | 0.1                          | -1.53E-06                | 6.53E+04                       |
| 3b     | 0.5                          | -1.85E-05                | 2.70E+04                       |
| 4      | 0.1                          | -1.75E-04                | 5.70E+02                       |
| 5      | 0.1                          | -1.35E-04                | 7.39E+02                       |
| 6      | 0.1                          | -1.22E-04                | 8.21E+02                       |
| 7      | 0.1                          | -1.87E-05                | 5.35E+03                       |

For a wide variety of polymeric substances, the zero-shear viscosity is proportional to a positive power of the material's molecular weight (Morrison, 2001). While the power parameter for biofilm remains unquantified, it is reasonable to assume that this relationship is still valid. This expression was developed for largely homogenous industrial polymers but considering the heterogeneous structure of biofilm, it may be more appropriate to correlate  $\eta_0$  with density. Measuring sample densities would have been destructive to the sample. However, the values reported in Table 3 indicate that there was significant variation in sample density. Density variations occurred despite uniform growth and transport conditions. Also, no correlation could be made between disk location in the growth chamber and  $\eta_0$ . This further supported the widely accepted supposition that biological materials, specifically mixed biofilm cultures, are inherently variable.

Finally, the recoverable compliances were also measured.  $J_{\infty}$  is an indicator of the materials ability to recover after loading. While not truly non-dimensional like the recovery capacities presented in Table 1,  $J_{\infty}$  is independent of the applied force. The values in Table 4 were calculated using equation (33) and measurements of maximum and final strain.

Table 4. Recoverable Compliances of Biofilm Samples.

|        | $\tau_0$         | $\gamma(t_s)$ | $\gamma(t=\infty)$ | $\gamma_{\infty}$ | $J_{\infty}$      |
|--------|------------------|---------------|--------------------|-------------------|-------------------|
| Sample | N/m <sup>2</sup> | m/m           | m/m                | m/m               | m <sup>2</sup> /N |
| 1      | 0.1              | 0.2562        | 0.1248             | 0.1314            | -1.3140           |
| 2      | 0.1              | 0.1366        | 0.0182             | 0.1184            | -1.1843           |
| 3a     | 0.1              | 0.0163        | 0.0037             | 0.0126            | -0.1261           |
| 3b     | 0.5              | 0.0842        | 0.0134             | 0.0708            | -0.1415           |
| 4      | 0.1              | 1.3088        | 0.6412             | 0.6676            | -6.6761           |
| 5      | 0.1              | 0.1571        | 0.0106             | 0.1465            | -1.4650           |
| 6      | 0.1              | 0.4526        | 0.1492             | 0.3033            | -3.0334           |
| 7      | 0.1              | 0.0885        | 0.0396             | 0.0489            | -0.4891           |

Given the variability observed in these responses, the most appropriate comparison was made between tests 3a and 3b. Sample 3 was subjected to two successive creep tests. Initially, 3a was performed using an applied stress of 0.1 N/m<sup>2</sup>. After the 6 minute long test, the same sample was tested again at a higher stress level ( $\tau_0 = 0.5$  N/m<sup>2</sup>). In between the two tests, the material was allowed to recover for approximately 10 minutes. Because these two tests used the same mixed biofilm culture, the issues of variability due to temperature, nutrient supply and biology, as discussed

previously, were not applicable. Figure 19 shows the compliance functions for samples 3a and 3b using a common ordinate.

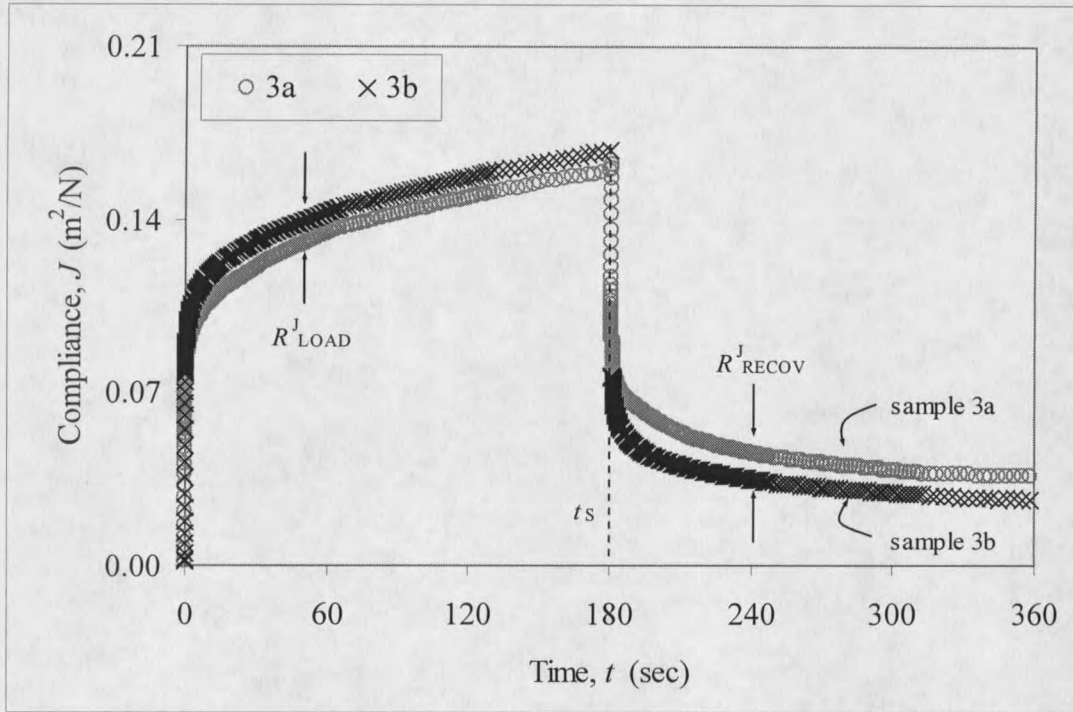


Figure 19. Compliance Functions for Repeated Tests on Sample 3. 3a represented the initial loading at  $\tau = 0.1$  Pa. 3b was subsequently loaded at  $\tau = 0.5$  Pa.

Recall that the compliance function is independent of the applied stress. Thus two identical materials, tested over the same length of time, should have produced identical compliance curves. Clearly, Figure 19 demonstrates that there was close agreement between the  $J(t)$  curves. In fact, there was a higher degree of agreement between 3a and 3b than between the other two compliance functions as can be seen in Figure 18. Despite this agreement, the differences between the two curves were significant as noted by the measurement of central tendencies reported in Table 5.



Table 5. Descriptive Statistics for Biofilm Sample 3a, 3b Comparisons.  $R^J = J(t)_{3a} - J(t)_{3b}$  is the residual compliance.  $J^* = J(t)_{3b} / J(t)_{3a}$  is the ratio of the compliances.

| measure<br>units | $R^J$<br>$m^2/N$ | $R^J_{LOAD}$<br>$m^2/N$ | $R^J_{RECOV}$<br>$m^2/N$ | $ R^J $<br>$m^2/N$      | $ R^J_{LOAD} $<br>$m^2/N$ | $ R^J_{RECOV} $<br>$m^2/N$ |
|------------------|------------------|-------------------------|--------------------------|-------------------------|---------------------------|----------------------------|
| Mean             | 0.00222          | -0.0114                 | 0.0158                   | 0.0138                  | 0.0117                    | 0.0158                     |
| Median           | 0.00873          | -0.0118                 | 0.0145                   | 0.0129                  | 0.0118                    | 0.0145                     |
| SD               | 0.0155           | 0.00546                 | 0.00897                  | 0.00741                 | 0.00458                   | 0.00897                    |
| measure<br>units | $J^*$<br>-       | $J^*_{LOAD}$<br>-       | $J^*_{RECOV}$<br>-       | $J^*_{RECOV}^{-1}$<br>- |                           |                            |
| Mean             | 0.952            | 1.13                    | 0.777                    | 1.29                    |                           |                            |
| Median           | 0.835            | 1.12                    | 0.767                    | 1.30                    |                           |                            |
| SD               | 0.211            | 0.160                   | 0.0508                   | 0.110                   |                           |                            |

### Summary

The time-dependent strain response evident in these creep curves clearly indicate that our mixed biofilm culture exhibited both elastic and viscous behavior. Additionally, all eight curves indicate the material had the capacity for an instantaneous elastic response and recovery. Also significant was the expression of an unrecoverable loss seen in these tests. This biofilm's capacity for unlimited deformation under a finite stress supports its classification as a viscoelastic fluid. This supports the conclusions drawn by previous investigators (Ohashi and Harada 1994, Stoodley et al. 1999, Korstgens et al. 2001) through the use of an accepted material testing technique (i.e. rotating disk rheometry). Despite the consistent viscoelastic character of the strain response, there was significant variability in the magnitudes of the observed strains for the applied stress of  $\tau_0 = 0.1 \text{ N/m}^2$ . This is not surprising given the natural variability inherent in many biofilms. Despite this variability in strengths, each sample clearly displayed elastic, time-

dependent and viscous responses. In the next chapter, these experimental results will be used to determine an appropriate viscoelastic constitutive relation for these microbial biofilms.

## BIOFILM CONSTITUTIVE LAW

In the study of continuum mechanics, material parameters are couched in a constitutive law, which relates stress to strain. Previous work has shown that laboratory-grown biofilm demonstrated a viscoelastic response to stress. (Klapper, et al., 2002; Korstgens, et al., 2001; Stoodley, et al., 1999b). Moreover, these studies suggested that biofilms behave like viscoelastic fluids (i.e. materials that exhibit continuous strain under a constant load). Stoodley, et al., (1999b) calculated a shear modulus and viscosity for a mixed biofilm culture. These properties were based on measurements of the elongation of biofilm streamers in glass flow tubes in response to varying flow conditions in which a three-dimensional deformation was approximated as a one-dimensional uni-axial strain. Such results cannot be interpreted as a valid constitutive law. However, these data were informative and provided a foundation for a formal investigation of the biofilm constitutive relationship. While the previous chapter concentrated on empirical observations made from the test data and qualitative assessments of those results, the following section concerns itself with mathematically predicting the behavior of this complex fluid.

To precisely describe the mechanics of biofilm behavior, indeed the behavior of any continuous medium, one must employ three types of relationships: field equations, equations of state and material-specific constitutive equations. The three categories of field equations are the continuity equation, the equations of motion, and the energy equation. If the investigation can be limited to isothermal conditions, one effectively

uncouples the conservation of energy principle from the other field equations. Then only the equations of motion and the continuity equation are mathematically necessary (Mase and Mase, 1999). Based upon the concept of conservation of mass, in Eulerian form, the scalar continuity equation in Einstein indicial notation is

$$\frac{\partial \rho}{\partial t} + (\rho v_k)_{,k} = 0, \quad (36)$$

where  $\rho$  is the material density, and  $v_k$  is the velocity in the  $k^{\text{th}}$  direction. The equations of motion are derived from the principle of conservation of momentum. As a vector equation, this is a set of three field equations representing motion in three independent directions ( $i, j$ , and  $k$ )

$$\sigma_{ji,j} + \rho b_i = \rho \dot{v}_i, \quad (37)$$

where  $\sigma$  is the stress tensor and  $b$  is the body force. In the absence of electro-magnetism, the sole body force is that of gravitational attraction.

The conservation laws that describe these four field equations are applicable to all continuous media. That is, they make no account of material properties. It is the constitutive equations that knit together these relationships and allow for the mathematical formulation of problems in engineering mechanics. Real materials, however, can exhibit a wide range of behavior when subjected to different thermal loads

and stresses. Consequently, the response of any material over the full range of possible loadings cannot be described by a solitary constitutive law. Instead, an investigator seeks to develop an idealized response that describes a material's behavior over a discrete range only. Thus, constitutive laws more appropriately describe a particular behavior rather than a specific material (Mase and Mase, 1999).

In that context, a constitutive law describing the behavior of the mixed biofilm culture presented in the previous chapter was developed. The only way to determine the validity of a constitutive law is through of comparison of measured results and response predicted by the constitutive equation (Morrison, 2001). This was accomplished by positing different constitutive laws, calculating the shear strain response based on the stress applied in the rheometer tests and then evaluating how well these formulas predicted the shear strain that was measured in the laboratory.

### Linearity of the Biofilm Response

As a first step in determining an applicable constitutive law, it is convenient to categorize the material as linear or non-linear viscoelastic. A great many materials produce a linear response over discrete ranges of stresses, strain, time and temperature (Boresi and Chong, 2000). The applicability of a linear viscoelastic model was examined for this microbial biofilm under the specified creep test parameters ( $\sigma_{12}=0.1 \text{ N/m}^2$ ). Two conditions must be satisfied to conclusively define a material response as linearly viscoelastic (Findley et al, 1976). They are linear proportionality and the Boltzmann superposition principles. Mathematically, these conditions can be stated as

$$\varepsilon_{12}[c\sigma_{12}(t)] = c\varepsilon_{12}[\sigma_{12}(t)] \quad (38)$$

and

$$\varepsilon_{12}[\sigma_{12}(t) + \sigma_{12}(t - t_1)] = \varepsilon_{12}[\sigma_{12}(t)] + \varepsilon_{12}[\sigma_{12}(t - t_1)] \quad (39)$$

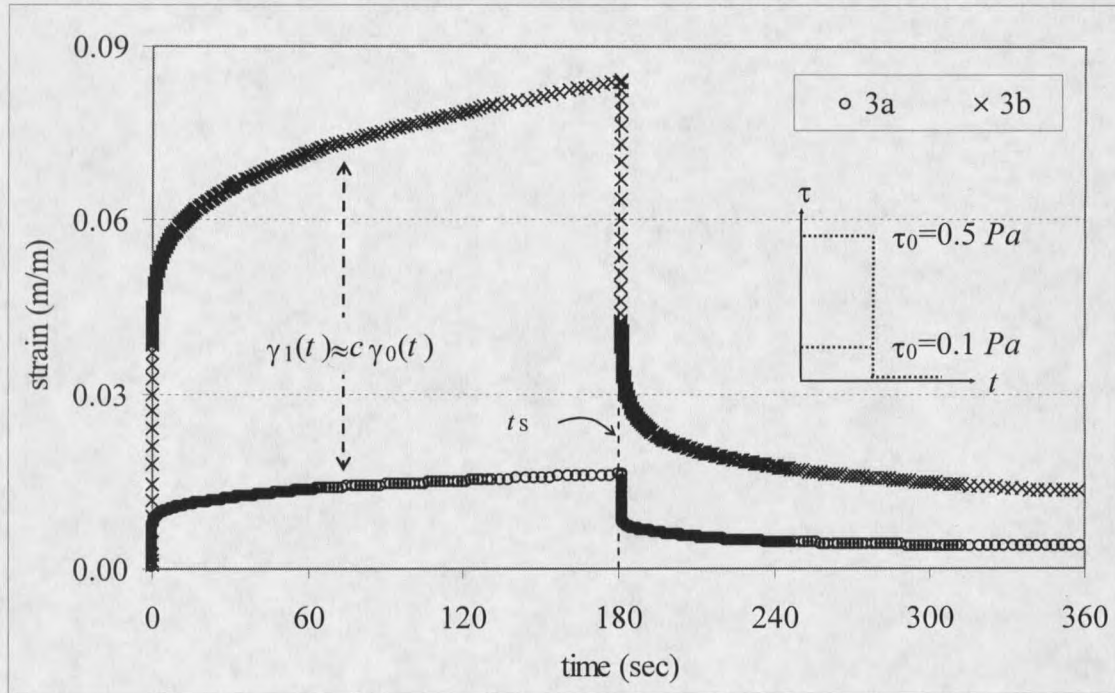


Figure 20. Strain Response of Samples 3a and 3b. A consistent ratio between  $\tau_1/\tau_0$  and  $\gamma_1/\gamma_0$  indicates linear viscoelasticity.

The first requirement (38) states that the strain,  $\varepsilon_{12}(\tau)$  due to a stress of  $c\sigma_{12}(t)$  is equal to the product of the constant  $c$  and the strain due to a stress of  $\sigma_{12}(t)$ . Equation (39) requires that the strain output due to the sum of any two stresses must be equal to the sum of the strains produced by those stresses individually.

As outlined in Chapter 4, producing viable mixed culture biofilm samples was time-intensive. Indeed, sampling and testing of the samples used in this analysis took over 4 months with an additional 6 weeks required for methods development at the outset. Because of these time constraints, an insufficient number of samples were produced to assess the applicability of the Boltzmann principle. However, the linear proportionality condition (38) was tested by performing two creep tests (under two different constant shear stress inputs,  $\sigma_{12} \rightarrow \tau_0, \tau_1$ ) on sample 3 (Fig. 12, 13). The strain responses of samples 3a and 3b are shown on a common plot in Figure 20. The first test (3a) was performed under a shear stress of  $\tau_0=0.1 \text{ N/m}^2$ . In the second test (3b), the applied stress was increased by a factor of five ( $\tau_1=0.5 \text{ N/m}^2$ ). The resulting strain-time curves were used to investigate the linearity of the biofilm responses.

To assess the linearity of the responses, the two data sets were plotted against each other. Though not taken at constant intervals, the times at which the two strain responses were measured were coincident. The strain outputs were plotted against each other, with  $\gamma_{3a}$  on the abscissa and  $\gamma_{3b}$  on the ordinate (Fig. 21). Assuming an ideal linear viscoelastic response, with no experimental variability, the slope of this plot should be equal to the ratio of the applied stresses,

$$c = \frac{\gamma[c\tau(t)]}{\gamma[\tau(t)]} = \frac{\gamma[5\tau(t)]}{\gamma[\tau(t)]} = \frac{\gamma[0.5 \text{ Pa}]}{\gamma[0.1 \text{ Pa}]} = 5.0. \quad (40)$$

Clearly, these two experimental data sets would not produce a constant proportionality ratio due to experimental variability and instrument accuracy. Thus, a least squares regression was used to approximate the proportionality constant. Through regression analysis, a 'best fit' line was determined by minimizing the sum of the squared residuals while fixing the intercept at the origin. The intercept was fixed at the origin to match the theoretical strain output of both responses in their initial, unloaded condition. The resulting analysis produced a proportionality constant,  $c$ , of 5.0843 with an  $R^2$  of 0.8819.

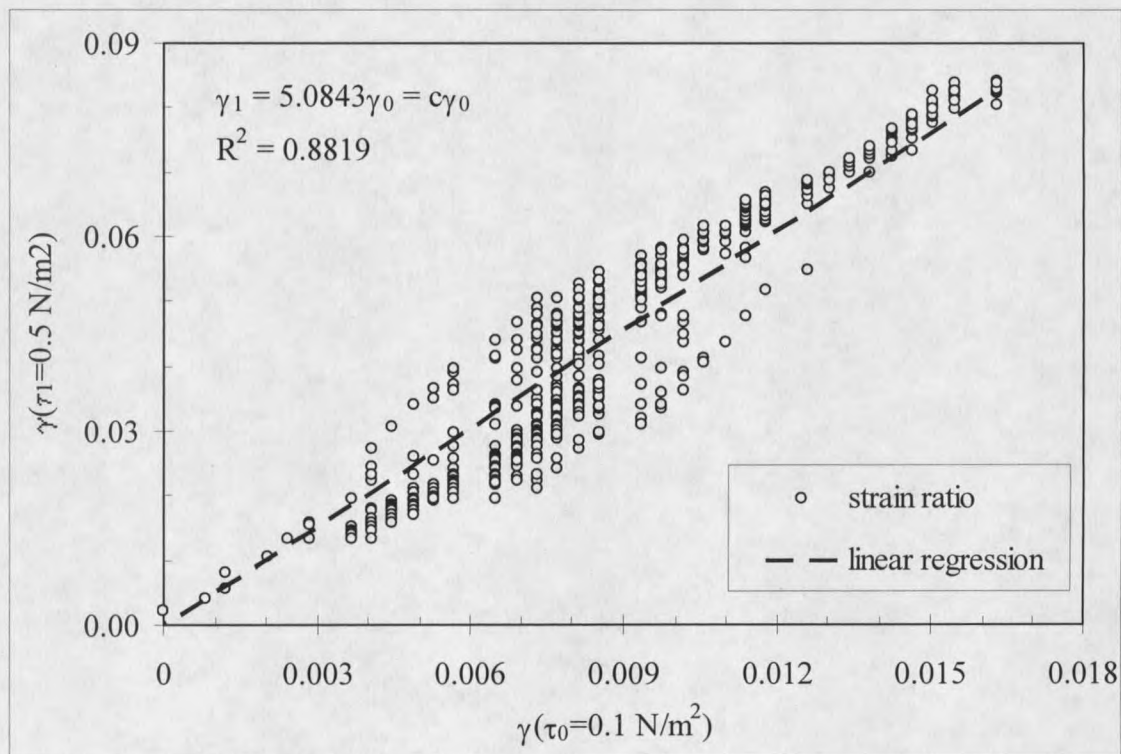


Figure 21. Linear Proportionality Coefficient. Coefficient determined through regression analysis. Point data are observations of shear strain ratios.



The high  $R^2$  value of 0.8819 suggested that the strain relationship over a shear stress range of 0.1 N/m<sup>2</sup> to 0.5 N/m<sup>2</sup> was approximately linear. In a regression analysis, a residual plot can elucidate trends in the data. The residual of the linear coefficient was calculated by

$$R_c(t) = \frac{\gamma[\tau_1(t)]}{\gamma[\tau_0(t)]} - c, \quad (41)$$

and a plot of this function (41) versus time for both the retardation and recovery stages is presented in Figure 22.

Overall, the close agreement between the calculated proportionality coefficient and the actual strain ratio (5.0) justified the use of a linear viscoelastic model for this mixed culture biofilm under these conditions ( $\tau_0 = 0.1$  N/m<sup>2</sup> to 0.5 N/m<sup>2</sup>), especially during the loading phase.

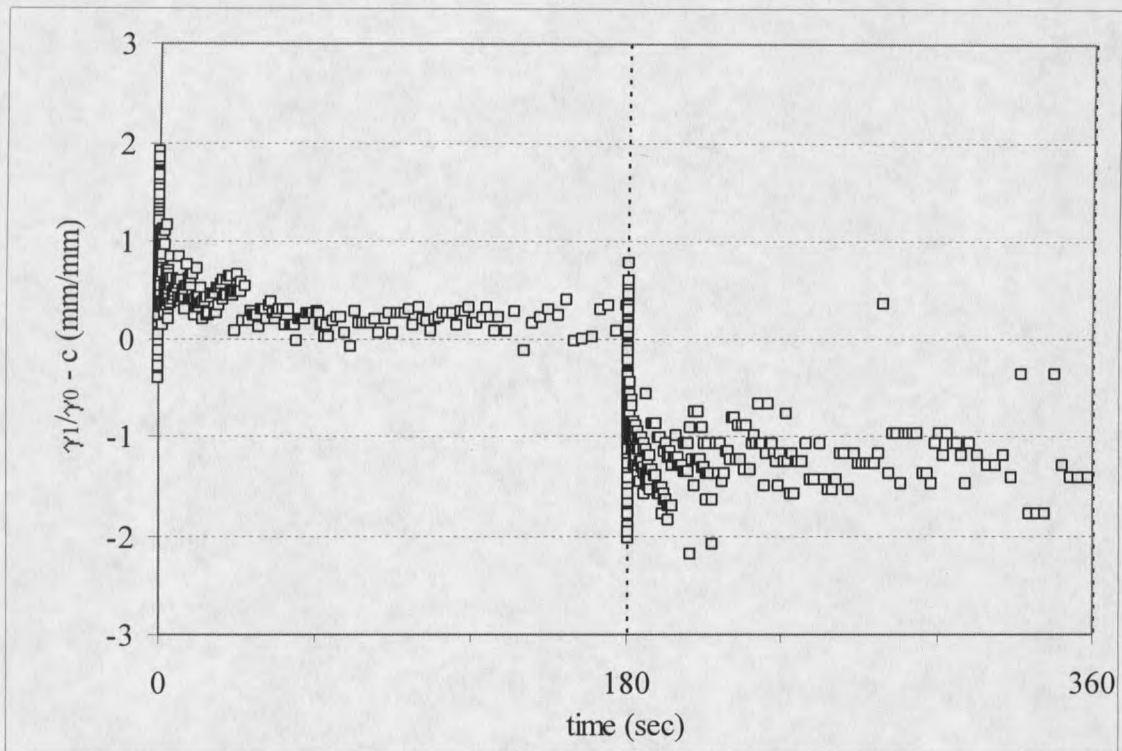


Figure 22. Residual Plot of the Linear Proportionality Coefficient Versus the Observed Strain Ratios. The dashed vertical line indicates the end of the retardation stage.

### Implications of the Linearity Coefficient

It is interesting to note that this coefficient was slightly greater than 5.0. This indicated that when the applied stress was increased by a factor of 5.0, the resulting strain response was slightly greater than this ratio. Two possible explanations may exist. The second stress level applied to the sample was  $\tau_0 = 0.5 \text{ N/m}^2$ . The applied stress in this test may have approached the biofilm's threshold of non-linearity. Alternatively, the high  $c$  value may have been due to partial adhesive failure between the biofilm and the rheometer disk.

During the methods development for these shear creep tests, several different applied stress values were examined. It was observed that shear stresses in excess of  $1.0 \text{ N/m}^2$  consistently resulted in a strain response that closely approximated a Newtonian fluid. That is, no appreciable elastic response was observed and the strain rate was nearly constant. Such a response could have several different causes. First, the internal biofilm structure may have failed under the applied stress of  $1.0 \text{ N/m}^2$ . Failure of the cross-linked polymer chains within the EPS robbed the material of its elastic attributes. In this case, the material characteristics of the biofilm might revert to those of the underlying bulk fluid: water. Alternatively, adhesive failure may be the cause. In such a situation, slippage may have been the cause of the Newtonian behavior. Adhesive failure between the sample and the bottom plate would have resulted in a rotating disk that was affected only by the viscosity of the bulk fluid between the sample and not the material properties of the biofilm growth. Determining the actual type of failure would have been difficult. There is no method for assessing adhesion between the sample and rheometer base beyond the empiricism of creep test results. Nevertheless, given the physiology of biofilms in general, it is likely that adhesive failure occurred. EPS, the outer covering of an attached microbial biofilm is essentially a slick, slime-like substance. Therefore, as a fluid/structure boundary condition, the EPS has a low coefficient of drag. In fact, the EPS likely enhances the microorganism's survival by reducing the potential for fluid-induced detachment in an aqueous environment. Consequently, this same slime matrix is present between the biofilm and the rheometer base. It is likely that this adhesive bond is considerably weaker than the internal polymeric structure of the biofilm.

There was no evidence that slippage or internal failure was the cause of the deviations from true linearity. Slippage or adhesive failure would have expressed itself at a sharp increase in strain over time. True 'near' instantaneous slippage would have appeared as a vertical jump strain on the creep curves. No such jump was observed. Therefore, it is more likely that the applied stress of  $1.0 \text{ N/m}^2$  was beyond the linear viscoelastic range and the slight departure from a constant proportionality ratio may have been due to the fact that this second stress value was nearing a point of transition from linear to non-linear viscoelasticity.

Another explanation for the increase in the proportionality coefficient may be permanent damage. Because of the inherent variability in any biological material, the linear proportionality test had to be performed on the same sample. The repeated testing may have weakened or damaged the structure of the EPS. Biofilm strength is, in part, a function of the shear stress under which it was grown (Stoodley, et al., 2001). As such, the material properties of our samples were based on the growth shear stress of approximately  $0.084 \text{ N/m}^2$ . The strength of this biological material may have been exceeded by a five-fold increase in applied load. Some degree of ductile failure, due in part to realignment of polymer chains, may be responsible for permanently altering, and indeed weakening, the material properties of this sample. This weakened biofilm would have expressed a greater strain resulting in a proportionality constant greater than the ratio of stress inputs.

The residual plot (Fig. 22) illustrates two temporal trends of interest. The first is a slight non-linearity in the strain data near the start of the test. This may suggest that the

higher shear value,  $\tau_1$ , is near the limit of linear behavior. This trend may also indicate that, in general, the delayed compliance,  $J_D$ , of this material is not perfectly linear under any shear loading condition. However, the overall departure from linearity throughout the retardation phase is slight and the strain response suggests that a suitable mechanical analog can be used to represent the behavior of this mixed culture biofilm. Of greater concern was the material response during the recovery ( $t > 180$  s) stage. As can be seen in the residual plot, the  $3a/3b$  strain ratios decrease markedly once the material is unloaded. Again, damage may have been the cause of this change in the ratio. However, damage to the EPS would have accrued incrementally. The deviation from linearity was, in fact, sudden. A more likely explanation may be that this change in ratio is actually evidence of anisotropy in the material. The biofilm was loaded during testing in the same direction as it was loaded during the growth period. The sharp deviation from recovery was observed during unloading when the material was recovering in the opposite direction. More insight into the range of linear viscoelasticity for this material may be gained from further experimentation.

#### Significance of the Creep Compliance

Once it was found that the biofilm creep results satisfactorily met a criterion for linear viscoelasticity according to equation (40), a specific linear viscoelastic constitutive model was determined. As outlined in Chapter 3, linear viscoelastic models are constructed by superimposing the effects of dashpot and spring elements arranged serially or in parallel. The process used to determine an appropriate model began with

identifying certain characteristics within the biofilm creep responses: an instantaneous elastic response, a time dependent strain-rate, and a permanent viscous loss. The presence or absence of these three characteristic behaviors was used to determine the presence or absence of three complimentary components of creep compliance in the linear viscoelastic constitutive equation. The combination of these individual creep compliances leads to an appropriate mechanical analog (e.g. Kelvin, Maxwell, Burger, etc.).

As discussed in the previous chapter, the compliance is a ratio of strain output to stress input. In the context of continuum mechanics, the compliance is the inverse of a model's modulus. The true utility of using creep compliances, as opposed to moduli, is in how the three aforementioned characteristic viscoelastic responses may be related to separable (i.e. subject to superposition) creep compliance components. The compliance of a complex linear viscoelastic model is the sum of the compliances of any simpler, series-connected viscoelastic models (Flugge, 1975). In terms of its three components, the compliance can be written as

$$J(t) = J_E + J_V(t) + J_D(t), \quad (42)$$

where  $J(t)$  is the combined creep compliance,  $J_E$  is the instantaneous elastic component of the compliance,  $J_V(t)$  is the time-dependent loss component of the compliance and  $J_D(t)$  is the time-dependent delayed component of the compliance.

Given this relationship, the following approach was used to identify the linear model. First, the presence of instantaneous elastic behavior was determined. The only linear analog that will produce this type of response is a spring element (Figure 2). Therefore, if present in the creep curve, the creep function will contain an elastic component,  $J_E$ . Similarly, if the material expressed an unrecoverable loss, this would be evidence of a isolated dashpot element in the analog and, accordingly, a viscous component,  $J_V(t)$  in the compliance. From equation (6), it can be shown that the Newtonian element is the only one capable of asymptotically approaching a constant strain as time approaches infinity. Finally, the nature of the strain, especially during retardation, should be examined. A constant strain rate (i.e. slope of the strain-time curve) would require a Newtonian dashpot as the only time-dependent element. However, a changing strain rate, or more specifically, a negative derivative of strain with respect to time will indicate that the creep compliance has a linear time-dependent component,  $J_D(t)$ . There are an infinite number of linear analogs that will produce a time dependent delayed compliance; however a Kelvin element (Figure 2) is the simplest possible analog. In this case, a Kelvin element was assumed and complexity (i.e. linear elements) added until the observed behavior was adequately modeled. This approach could lead to a unnecessarily complex analog for  $J_D(t)$ . Thus, the final model must be robust enough to adequately describe the shear response, but simple enough to maintain the value of an analogy to the physics of the linear spring and dashpot elements.

The preceding technique was applied to the rheometry data garnered in the experiments detailed throughout Chapter 4. As can be seen in Figures 10 through 17, all

7 creep curves exhibited these three types of viscoelastic responses. The Burger model presented in Chapter 3 is the simplest linear viscoelastic model that can exhibit all three types of compliances. The next step in the process was to optimize the material parameters in a Burger model to each of the seven creep curves and evaluate the resulting strain predictions. A metric of these 'fits' was calculated and used to justify the general applicability of this four element constitutive equation to the mixed culture microbial biofilm.

#### Optimizing Parameters in the Burger Constitutive Equation

Assuming that the biofilm was sufficiently linear and that the creep curves indicated all three modes of linear viscoelastic compliance, a procedure for numerically optimizing the material coefficients in the Burger model, equations (12, 13), was developed. Typically, a non-linear regression method would be used to fit discrete experimental data to an analytical equation. However, gradient methods would be inappropriate for this situation. A creep test involves the application of a constant stress during the retardation period, followed by unloading during a recovery period. This results in a unit step function in the creep equation (11) and, consequently, a discontinuity in the derivatives of this equation (Mase, 1970). Gradient methods, such as the Steepest Ascent method and those employing Hessians make use of the derivatives of an analytical formula to fit discrete data to it (Chapra and Canale, 1998). The discontinuity in linear viscoelastic creep models prohibits the use of such methods. While there are search methods available that do not employ derivatives including pattern search



techniques such as Powell's method, they are equally ill-suited for discontinuous functions. Therefore, a new algorithm was developed to optimize the material parameters in linear viscoelastic constitutive equations. The development of the following algorithm was applied to the Burger Model but could easily be modified to accommodate any number of linear viscoelastic elements.

The method used to optimize the parameters in the linear viscoelastic model was similar to the well-known bisection or binary selection method (Stark, 1970). This algorithm uses a bracketing approach based on the Intermediate Value Theorem. The Intermediate Value Theorem states that if  $g$  is a continuous function on the closed interval  $[LL, UL]$  and  $K$  is any number between  $g(LL)$  and  $g(UL)$ , then there exists a number  $P$  in the open interval  $(LL, UL)$  for which  $g(P) = K$  (Mounem and Foulis, 1984). The bisection method involves iteratively identifying the location of the 'better' solution and eliminating the other half of the range. Similarly, the following algorithm iteratively identifies the location of the 'better' solution and redefines the range centered on that improved solution. Most importantly, like the bisection method, this technique operates without the need to calculate a derivative of  $g(P)$ .

To illustrate how the Burger equation was fit to observed strain, the approach, idealized for one parameter, is as follows: Given a reasonable estimate of the interval  $[LL, UL]$  in which the optimal value of the material parameter  $P$  exists, the technique calls for calculating the mean absolute weighted residual (WR) at discrete points within this range of  $P$ . The WR was defined as the average absolute difference between the observed strains and strains predicted by the Burger model. The optimal material

parameter is the discrete point within the range of  $P$  that produces the lowest WR for the data set. On the first iteration, the WR at three points is calculated  $\{LL, (LL+UL)/2, UL\}$ . In other words, the residuals at the end points and the mid-point of the range of the material parameter,  $P$ , are evaluated. The next step in the algorithm is dependent upon which point produced the lowest WR. If an end point (i.e.  $LL, UL$ ) produces the smallest error, then the parameter range is re-centered upon that end point and the algorithm repeats itself. In this way, the algorithm 'slides' in one direction or the other if the optimal choice is outside the initial range of  $P$ . This sliding will continue until a mid-point in the interval produces the lowest WR. At this stage, the number of discrete points at which the WR is calculated changes from three to two. These two new points of evaluation are the mid points between  $[LL, (LL+UL)/2]$  and  $[(LL+UL)/2, UL]$ . Evaluation of the WR at these two points prevents the algorithm from stalling or stagnating near the midpoint by further dividing the interval. These two values of  $P$  are 'staggered' between the 3 previous points. As before, the interval is re-centered on the point producing the lowest WR. To refine the estimates at each iteration, the magnitude of the interval is then reduced by a reduction factor,  $RF$  ( $0 < RF \leq 1.0$ ). Then the limits and mid-point of this reduced range are evaluated and the procedure repeats itself. This 'slide and stagger' algorithm repeats iteratively until a convergence criterion or maximum number of iterations is reached.

The actual problem involved optimizing 4 parameters (2 elastic constants and 2 viscous constants) in a linear viscoelastic equation. The following is a formal description of the algorithm used to fit observed shear strain to the Burger equation:

Begin by setting the set of initial lower limits  $\{LL_p^1\} = \{LL_p\}$  and the set of initial upper limits  $\{UL_p^1\} = \{UL_p\}$  for all material parameters  $p = 1 \dots 4$ . Let  $\{P_p^1\}$  be a set of evenly spaced discrete points in  $[LL_p, UL_p]$ . Calculate  $g(P_1^1, P_2^1, P_3^1, P_4^1)$  at  $N$  observations. The measure of this fit is the weighted residual,

$$WR = \frac{\sum_{i=2}^N (R_i * (t_i - t_{i-1}))}{\sum_{i=2}^N (t_i - t_{i-1})}. \quad (43)$$

If  $WR(P_1^1, P_2^1, P_3^1, P_4^1) = 0$  then stop. If not, then the range  $[LL, UL]$  is re-centered around the best fit and the process is repeated. This method is presented in detail in the following algorithm. The algorithm was coded in The MathWorks, Inc, MATLAB, version 6.5, and run on an AMD2200 processor with 512 Megabytes of RAM. The MATLAB code is included in Appendix B.

#### Slide & Stagger Algorithm

To optimize the values of the 4 material parameters in the Burger model, a linear viscoelastic constitutive law given creep test data.

INPUT: a constant applied stress,  $\tau_0$ , and  $N$  observations of strain,  $S_i$ , versus time,  $t_i$  taken during a creep test; a weighted residual tolerance, TOL; a maximum number of iterations,  $L_{MAX}$ ; upper bounds,  $UL_p$ , and lower bounds,  $LL_p$ , on the

initial range of 4 viscoelastic parameters; a number of range divisions,  $m$ ; and a range reduction factor, RF.

OUTPUT: an optimized set of viscoelastic material parameters,  $\{P_1^O, P_2^O, P_3^O, P_4^O\}$  or a message of failure.

*Step 1:* Set  $WR^O = 1.0$

$$K = 1$$

$$P_p^O = 0$$

*Step 2:* For  $i = 1, 2, \dots, N$

$$S_i^O = S_i$$

*Step 3:* While ( $WR^O > \text{TOL}$ ) and ( $K \leq K_{\text{MAX}}$ ) Do *Steps 3 through 8*

*Step 4:* For  $p = 1, 2, \dots, 4$

$$\Delta L_p = (UL_p - LL_p) / (m-1)$$

*Step 5:* Set  $WR=0$

*Step 6:* For  $P_1 = LL_1, LL_1 + \Delta L_1, \dots, UL_1$

For  $P_2 = LL_2, LL_2 + \Delta L_2, \dots, UL_2$

For  $P_3 = LL_3, LL_3 + \Delta L_3, \dots, UL_3$

For  $P_4 = LL_4, LL_4 + \Delta L_4, \dots, UL_4$

For  $i = 1, 2, \dots, N$

$$S_i^P = g(\tau_0, t_i, P_1, P_2, P_3, P_4)$$

For  $i = 1, 2, \dots, N$

$$R_i = |S_i^P - S_i|$$

$$WR = \frac{\sum_{i=2}^N (R_i * (t_i - t_{i-1}))}{\sum_{i=2}^N (t_i - t_{i-1})}$$

If  $WR < WR^O$  Then

For  $p = 1, 2, \dots, 4$

$$P_p^O = P_p$$

For  $i = 1, 2, \dots, N$

$$S_i^O = S_i^P$$

$$WR^O = WR$$

Step 7: For  $p = 1, 2, \dots, 4$

If  $SFlag_p = 1$  then

If  $P_p^O = LL_p$  or  $P_p^O = UL_p$  then

$$LL_p = P_p^O - \frac{1}{2}(UL_p - LL_p)$$

$$UL_p = P_p^O + \frac{1}{2}(UL_p - LL_p)$$

Else

$$LL_p = P_p^O - \frac{1}{2}(UL_p - LL_p)RF$$

$$UL_p = P_p^O + \frac{1}{2}(UL_p - LL_p)RF$$

Step 8:  $K = K + 1$

Step 9: If  $(WR^O < TOL)$  Then

OUTPUT  $(P_1^O, P_2^O, P_3^O, P_4^O, S_i^O)$

Else

OUTPUT ("Method failed after ",  $L_{MAX}$ , "iterations")

STOP.

where the Burger model is defined by the function  $g$ :

$$g(\tau_0, t_i) = \tau_0 \left( \left( \frac{1}{P_1} \right) \left( 1 - e^{-t_i \left( \frac{P_1}{P_2} \right)} \right) + \left( \frac{1}{P_3} \right) + \left( \frac{t_i}{P_4} \right) \right) \quad t_i < t_s \quad (44)$$

$$g(\tau_0, t_i) = \tau_0 \left( \frac{t_s}{P_3} \right) + \frac{1}{P_1} \left( e^{t_s \left( \frac{P_1}{P_2} \right)} - 1 \right) e^{-t_i \left( \frac{1}{P_2} \right)} \quad t_i \geq t_s \quad (45)$$

The advantage of the approach is that this 'search' algorithm does not employ a derivative of (44, 45) in fitting the data. Furthermore, the algorithm will converge to weighted residual minima given a large enough  $L$ . Equally advantageous, is that the optimal solution need not be within the initial range  $[LL, UL]$  unlike the bisection method. The technique will re-center the range on the best solution and apply a reduction factor if and only if a set of parameters within this range reduces the weighted residual. Because the algorithm will move beyond the initial limits it cannot be described as a true 'bracketing' method. Neither is it appropriate to label this as an 'open' method. Implicit in the fact that the range is reduced over each iteration, there is a limit to how far outside the initial range that the algorithm will move. It is perhaps most appropriate to classify this algorithm as a 'quasi-open' method. The disadvantage of this method is that the solution can stagnate on local minima of the weighted residual. This would be evident in both a poor fit of the creep curve and in the fact that the method would fail to converge given a properly chosen value of the tolerance. However, this is easily remedied by

choosing alternate initial values and re-running the algorithm. It is also important to note that the algorithm is computationally intensive. However, that was mitigated by a reasonably fast processor.

### Parameter Optimization Results

While the previous algorithm could have been applied with very naive estimates of the parameter ranges, the initial values of  $\{UL_p\}$  and  $\{LL_p\}$  were based on observations of slopes and y-intercepts of the Figures 10 through 17 as outlined by Findley et al, 1976. Though unnecessary, these initial calculations minimized computer time and helped to validate the accuracy of the numerical method. The optimal Burger model functions, together with the experimental creep data, are shown on common time axes in Figures 23 through 29.

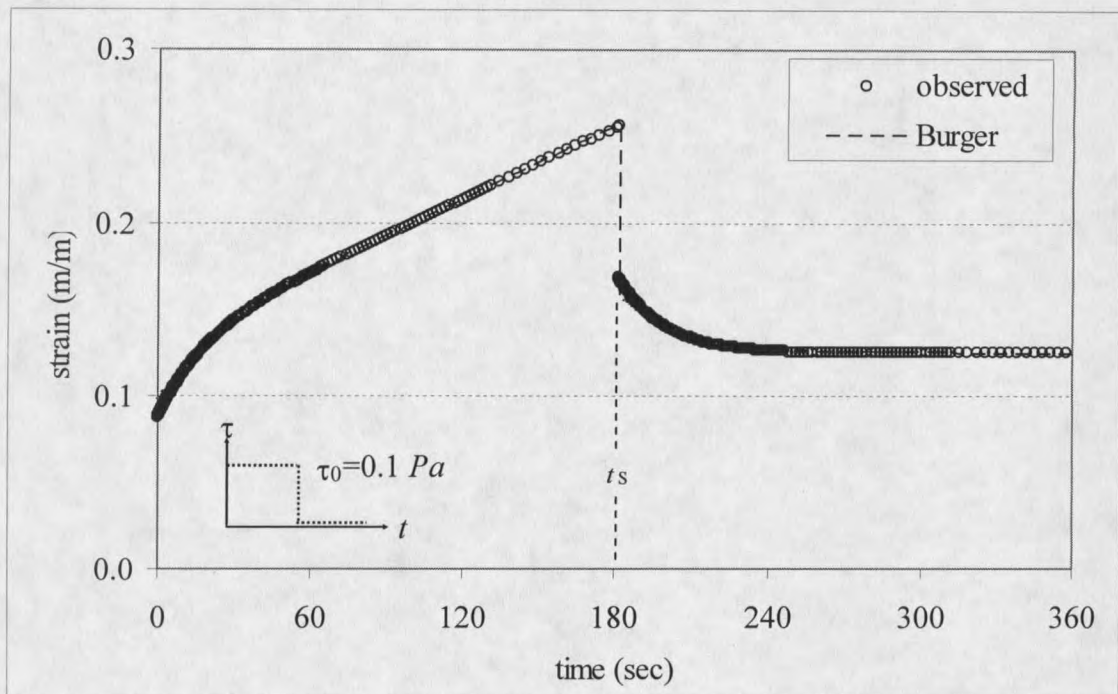


Figure 23. Creep Test Strain and Optimized Burger Model Function for Sample 1.

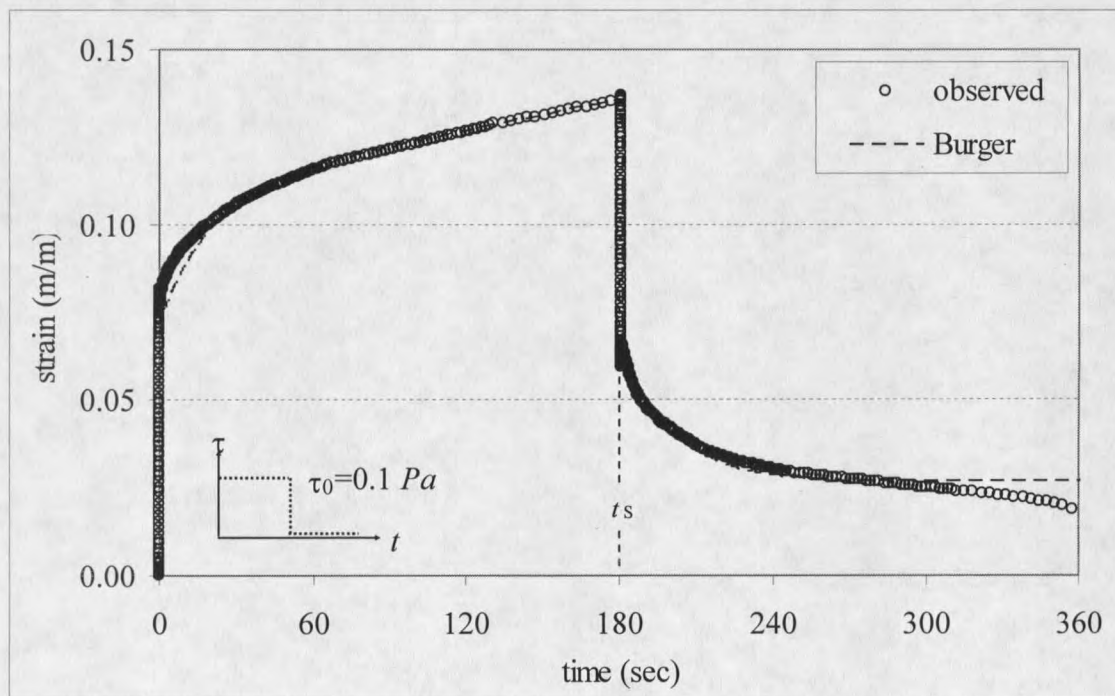


Figure 24. Creep Test Strain and Optimized Burger Model Function for Sample 2.



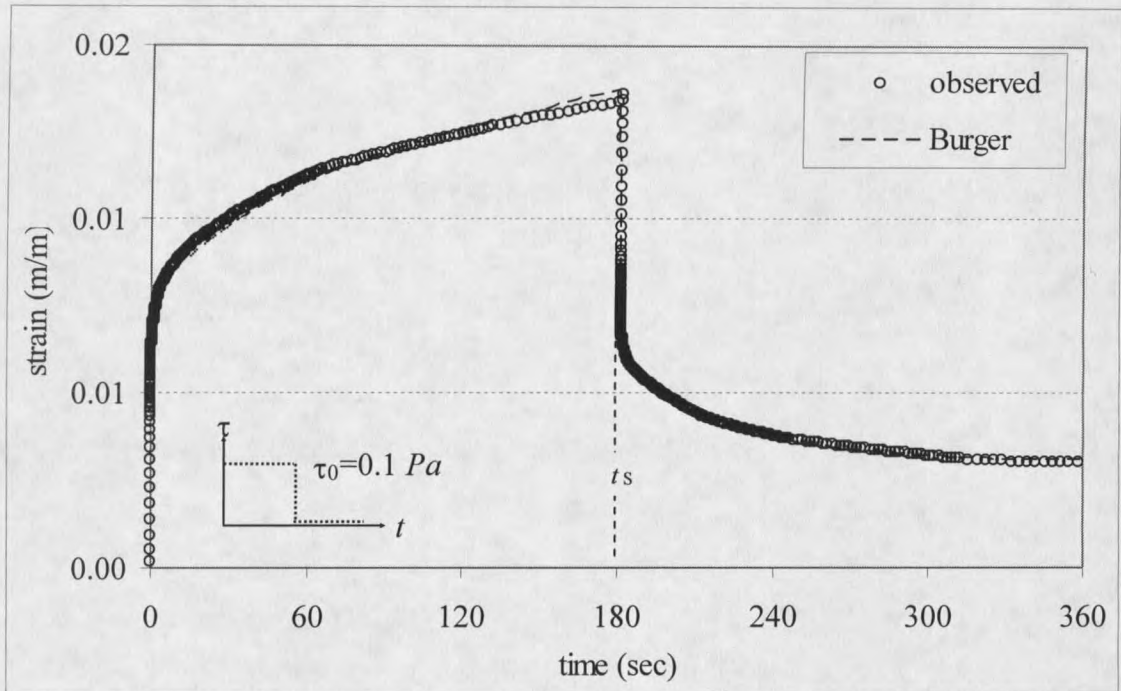


Figure 25. Creep Test Strain and Optimized Burger Model Function for Sample 3a.

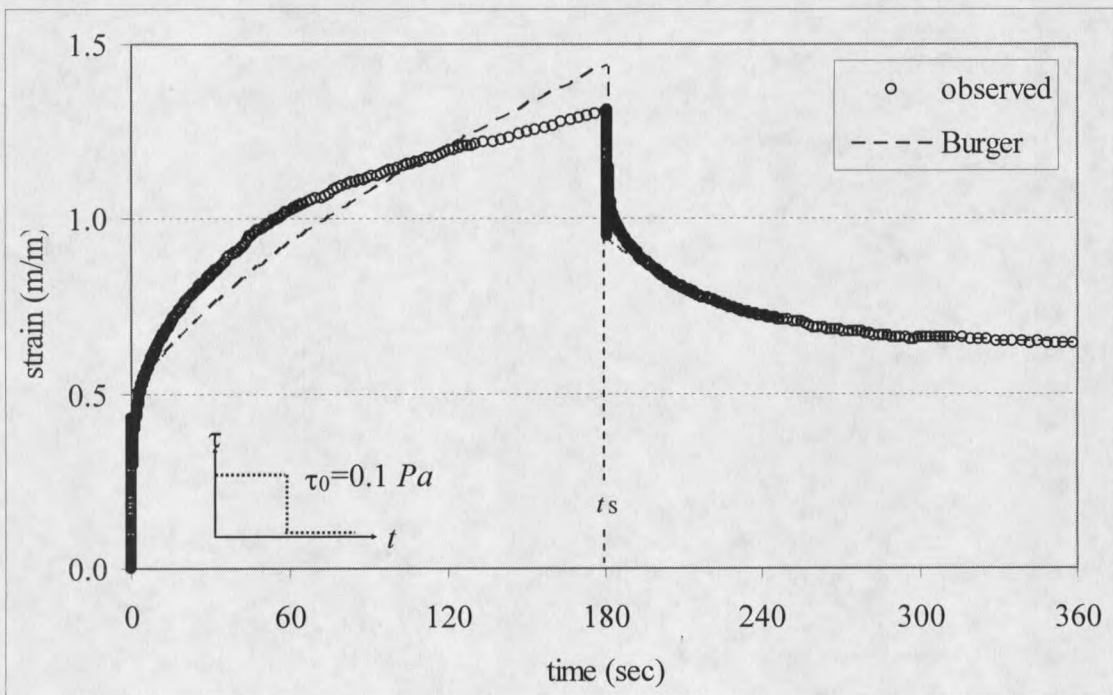


Figure 26. Creep Test Strain and Optimized Burger Model Function for Sample 4.

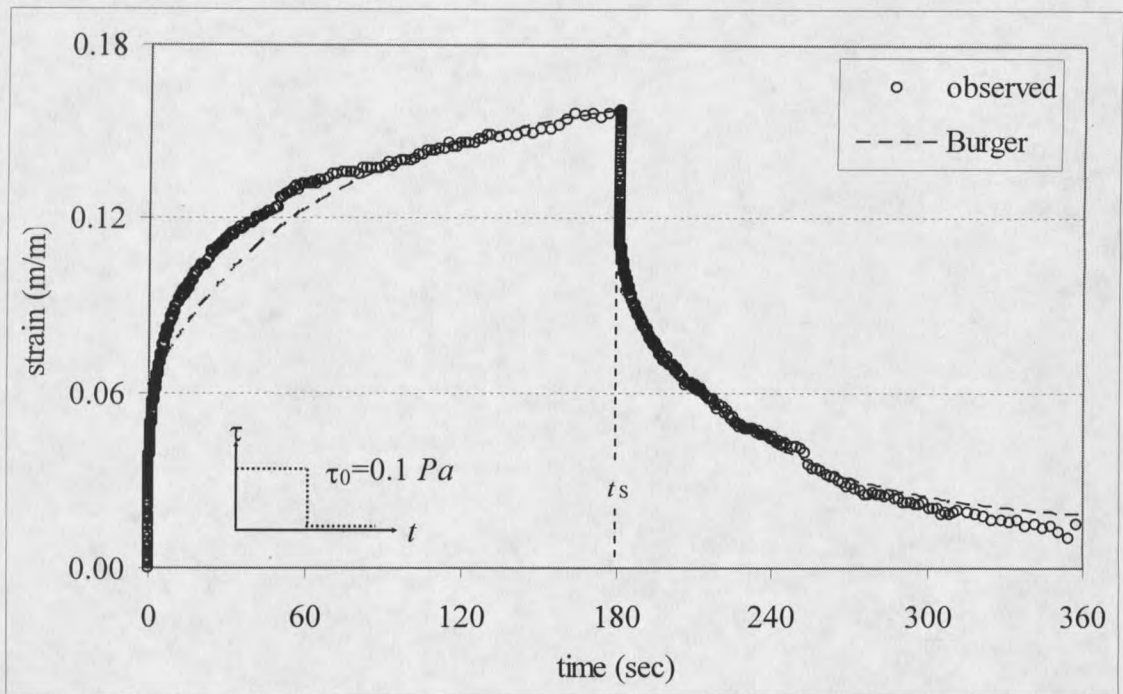


Figure 27. Creep Test Strain and Optimized Burger Model Function for Sample 5.

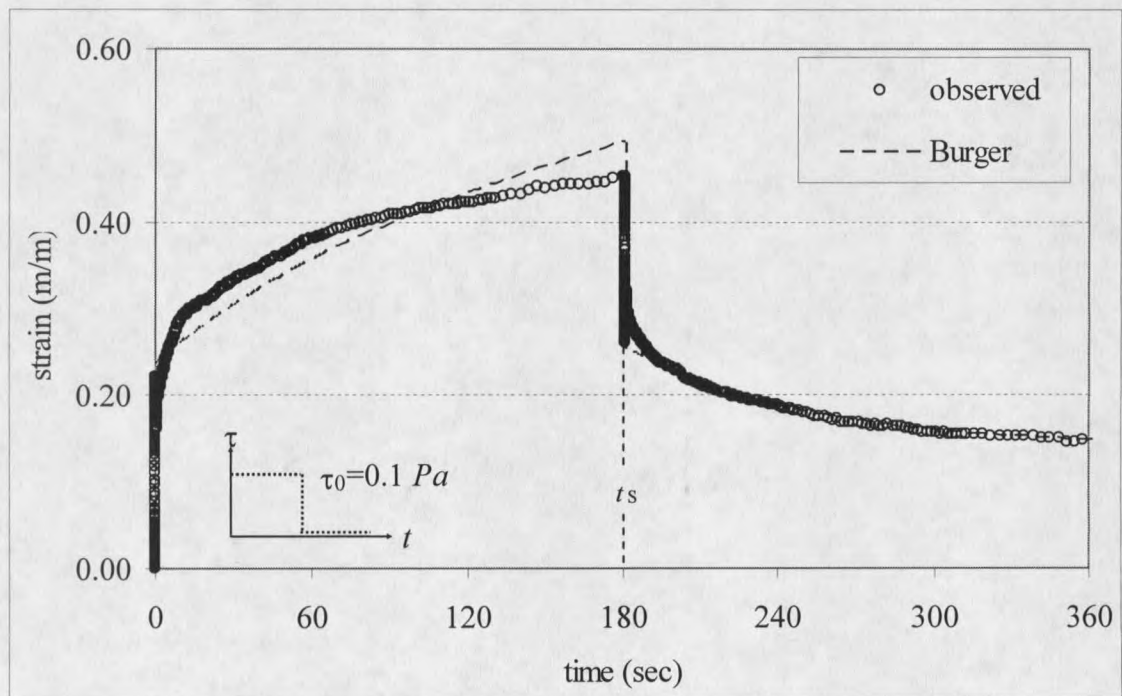


Figure 28. Creep Test Strain and Optimized Burger Model Function for Sample 6.

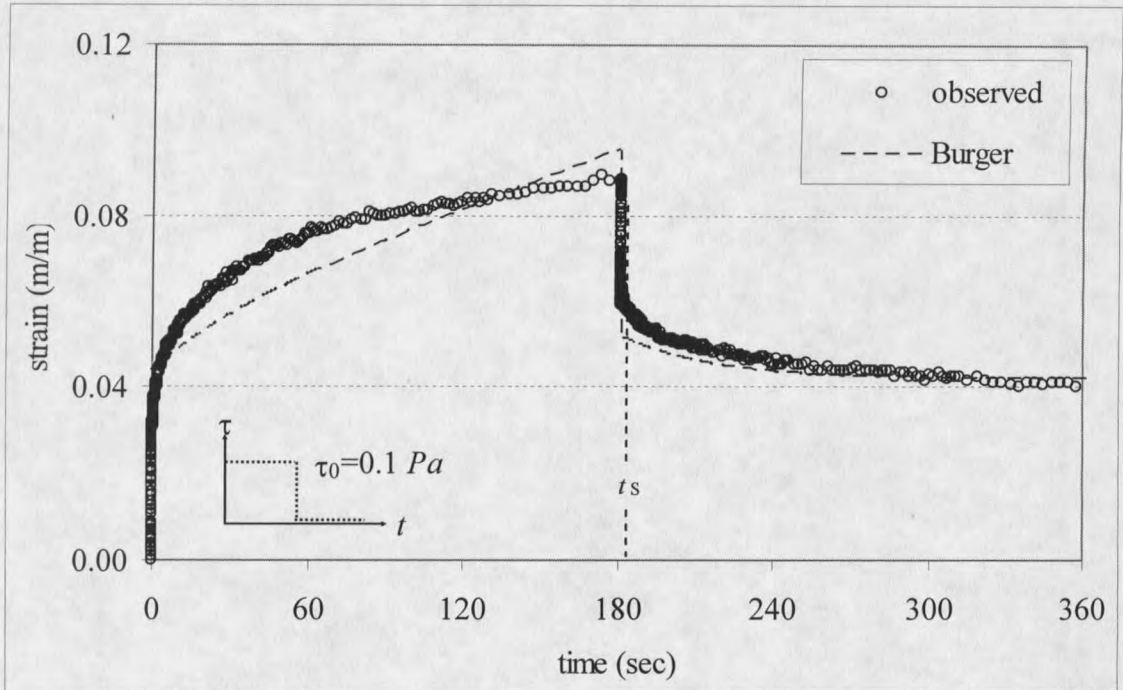


Figure 29. Creep test strain and optimized Burger model function for sample 7.

As discussed in Chapter 4, the strain responses for samples 1, 2, 3a, 4, ..., 7 varied greatly despite identical shear stress loadings ( $\tau_0 = 0.1 \text{ N/m}^2$ ). Not surprisingly, there was considerable variation in the optimized Burger parameters for these same samples. While the weighted residual ( $WR$ ) was used to optimize these curves, it was inappropriate for comparison between samples. An  $R^2$  value would also be a poor measure of the fit because of the unevenly spaced time data. The  $WR$  can be regarded as an absolute error between the Burger approximation and the observed strain that is weighted by the magnitude of the time step. An examination of equation 43 shows that the units on  $WR$  are meters per meter. To compare the quality of the curve fits among samples, the weighted residual was normalized against each samples maximum strain. This non-dimensional weighted residual ( $WR^*$ ), expressed as a percentage of the maximum strain,

varied from 0.28% to 4.36%. The optimized parameters along with the residuals are presented for each sample and the mean,  $\mu$ , in Table 6.

Table 6. Optimized Burger Model Parameters and Weighted Residuals.

|           | $\tau_0$<br>N/m <sup>2</sup> | $G_1$<br>N/m <sup>2</sup> | $\eta_1$<br>kg/m-s | $G_2$<br>N/m <sup>2</sup> | $\eta_2$<br>kg/m-s | $WR$<br>m/m | $WR^*$<br>- |
|-----------|------------------------------|---------------------------|--------------------|---------------------------|--------------------|-------------|-------------|
| Sample 1  | 0.1                          | 2.925                     | 65.88              | 1.026                     | 145.3              | 7.27E-04    | 0.28%       |
| Sample 2  | 0.1                          | 2.647                     | 52.19              | 1.401                     | 680.3              | 1.77E-03    | 1.30%       |
| Sample 3a | 0.1                          | 28.41                     | 1226               | 10.69                     | 5019               | 1.73E-04    | 1.07%       |
| Sample 4  | 0.1                          | 0.3179                    | 16.24              | 0.1994                    | 28.65              | 3.79E-02    | 2.89%       |
| Sample 5  | 0.1                          | 1.205                     | 65.97              | 1.596                     | 1202               | 3.84E-03    | 2.44%       |
| Sample 6  | 0.1                          | 0.7982                    | 54.77              | 0.4224                    | 130.1              | 1.21E-02    | 2.68%       |
| Sample 7  | 0.1                          | 10.00                     | 338.8              | 2.276                     | 432.2              | 3.89E-03    | 4.36%       |
| $\mu$     | 0.1                          | 6.615                     | 259.9              | 2.516                     | 1091               | -           | -           |

Clearly, this viscoelastic model was a reasonable approximation to the strain history in all samples. In fact, the applicability of the Burger model was underscored by the fact that this linear model adequately approximated shear strain despite the wide range of calculated material coefficients.

#### Linear Viscoelastic Constitutive Law

Using the results of the optimization as justification for the validity of the Burger model as a constitutive law for this mixed culture microbial biofilm, the general form of the constitutive equation is now developed. Viscoelastic behavior, linear or non-linear, blends the characteristic responses of both elastic materials and viscous fluids. Thus a constitutive law for a viscoelastic biofilm must have the capacity to dissipate mechanical

energy as a fluid does and store mechanical energy like a purely elastic material. Conceptually these two behaviors are end points on a spectrum of viscoelasticity. Consider first the constitutive equations (Mase and Mase, 1999) for an ideal isotropic elastic material,

$$\sigma_{ij} = 2G\varepsilon_{ij} \quad \forall i \neq j, \quad (46)$$

and

$$\sigma_{ii} = 3K\varepsilon_{ii} \quad \forall i = j, \quad (47)$$

where  $\sigma$  is the stress tensor,  $G$  is the shear modulus,  $K$  is the bulk modulus and  $\varepsilon$  is the strain tensor. The governing relationships for viscous flow (given small displacement gradients) take a similar form (Mase and Mase, 1999),

$$\sigma_{ij} = 2\mu^* \dot{\varepsilon}_{ij} \quad \forall i \neq j, \quad (48)$$

and

$$\sigma_{ii} = 3\kappa^* \dot{\varepsilon}_{ii} \quad \forall i = j, \quad (49)$$

where  $\mu^*$  is a viscosity coefficient and  $\kappa^*$  is the bulk viscosity coefficient. Note that equations (46) and (48) describe the deviatoric response and, similarly, equations (47) and (49) describe the dilatational response. By introducing linear differential operators, equations (46) through (49) can be written concisely as (Findley et al., 1976)

$$\{P\}\sigma_{ij} = \{Q\}\varepsilon_{ij} \quad \forall i, j, \quad (50)$$

where P and Q are linear differential operators with respect to time of the form

$$P = p_0 + p_1 \frac{\partial}{\partial t} + p_2 \frac{\partial^2}{\partial t^2}, \quad Q = q_0 + q_1 \frac{\partial}{\partial t} + q_2 \frac{\partial^2}{\partial t^2}. \quad (51)$$

By comparing equation (51) with equation (17), the differential form of the Burger constitutive law can be described using the following coefficients:

$$p_0 = 1, \quad p_1 = \frac{\eta_1}{G_1} + \frac{\eta_1}{G_2} + \frac{\eta_2}{G_2}, \quad p_2 = \frac{\eta_1 \eta_2}{G_1 G_2}, \quad (52)$$

$$q_0 = 0, \quad q_1 = \eta_1, \quad q_2 = \frac{\eta_1 \eta_2}{G_2} \quad (53)$$

The generalized linear viscoelastic strain function can be obtained by integrating equation (51). If the material is isotropic, the resulting integrand is one of two possible creep functions: one for deviatoric relaxation, another for bulk relaxation. Combined into one equation describing strain as a function of time yields:

$$\varepsilon_{ij}(t) = \delta_{ij} \left[ a_0 \sigma_{kk}(t) + \int_0^t \Psi_1(t-\xi) \frac{\partial \sigma_{kk}(\xi)}{\partial \xi} d\xi \right] + b_0 \sigma_{ij}(t) + \int_0^t \Psi_2(t-\xi) \frac{\partial \sigma_{ij}(\xi)}{\partial \xi} d\xi \quad (54)$$

There is considerable empirical evidence showing that nearly all engineering materials behave elastically in dilatation (Mase and Mase, 1999). Biofilm, as described in Chapter 2, is an aggregation of microbial cells surrounded in a polymeric substance. In an aqueous environment, the interstitial space of any biofilm is largely water (Kornegay and Andrews, 1967, Characklis et al., 1981, Zilver, 1979). Therefore, it is reasonable to assume that biofilm also reacts elastically in dilatation. Clearly, the deviatoric response is viscoelastic. Using the results from the shear creep tests and the supposition that biofilm acts elastically under hydrostatic conditions, equation (54) can be further simplified. Recall that under a constant shear stress

$$\sigma_{12}(t) = \tau_0 H(t), \quad (55)$$

where  $H(t)$  is the heavyside step function. By substituting (55) into (54) the deviatoric strain response is reduced to

$$\varepsilon_{12}(t) = b_0 \tau_0 + \Psi_2(t) \tau_0 \quad (56)$$

Now, equating the Burger creep compliance (12) with (56) and using the boundary condition that the instantaneous strain  $\varepsilon_{12}$  at  $t = 0$  is  $\tau_0/G_1$  then the coefficients  $b_0$  and the second creep function,  $\Psi_2$ , can be determined.

$$b_0 = \frac{1}{G_1}, \quad \Psi_2(t) = \frac{1}{G_2}(1 - e^{-G_2 t/\eta_2}) + \frac{t}{\eta_1} \quad (57)$$

Using a similar argument for the elastic dilatational response of the material, the coefficients  $a_0$  and the first creep function  $\Psi_1$  can be evaluated.

$$a_0 = \frac{1}{R_1} - \frac{1}{G_1}, \quad \Psi_1(t) = -\Psi_2(t) \quad (58)$$

Substituting equations (57) and (58) back into equation (54) yields the general strain equation for a linear viscoelastic Burger model.

$$\begin{aligned} \varepsilon_{ij}(t) = & \delta_{ij} \left[ \left( \frac{1}{R_1} - \frac{1}{G_1} \right) \sigma_{kk}(t) + \int \left[ -\frac{1}{G_2}(1 - e^{-G_2(t-\xi)/\eta_2}) - \frac{(t-\xi)}{\eta_1} \right] \frac{\partial \sigma_{kk}(\xi)}{\partial \xi} d\xi \right] \\ & + \frac{1}{G_1} \sigma_{ij}(t) + \int \left[ \frac{1}{G_2}(1 - e^{-G_2(t-\xi)/\eta_2}) + \frac{(t-\xi)}{\eta_1} \right] \frac{\partial \sigma_{ijk}(\xi)}{\partial \xi} d\xi \end{aligned} \quad (59)$$

This equation now allows for the general description of strain as a function of stress for a mixed culture biofilm that follows the Burger linear viscoelastic response.



Together with the equations (36) and (37), this constitutes a closed set of equations that describe the motion of a linear elastic biofilm. The next step is to incorporate equations (36), (37) and (59) into a numerical model that can be used to evaluate the biofilm response to a variety of flow conditions.

## DEVELOPMENT OF A COMPUTATIONAL MODEL

In the preceding sections, the experimental work outlined in Chapter 4 and curve fitting results presented in Chapter 5 were used to justify the applicability of the Burger linear viscoelastic material model for representing the deformation of mixed culture biofilms over a discrete range of applied loads. In those chapters, the Burger stress-strain relation was presented in general form, a three-dimensional constitutive law empirically validated for stress inputs in the range of  $0.1 \text{ N/m}^2$  to  $0.5 \text{ N/m}^2$ . It has been demonstrated that biofilm strength is dependent upon the conditions under which it was grown. Given this qualification, it is reasonable to assume that the Burger material model is more widely applicable. As such, this stress-strain relation can be used in the development of a computational model of the biofilm response to flow.

### Physical Problem Description

Stoodley et al. (1999b, 2002) developed an experimental technique for testing a biofilms response to turbulent flow conditions. This method involved growing microbial biofilm in flow-through cells under a variety of conditions. Individual biofilm streamers were identified and, using digital time lapse microscopy (DTLM), measurements of elongation versus changes in average flow rate were taken. Qualitatively, these experiments demonstrated, *in vitro*, the viscoelastic nature of biofilm. This work also provides a setting for the development of a computational model. However, the authors' claim that these experiments were "analogous to stress-strain and creep tests on attached

biofilms" is an overstatement. Creep and stress-strain tests are well-defined experimental procedures involving the application of a prescribed or measurable load to a material which responds by exhibiting a measurable deformation. In these studies, biofilm structures certainly deformed in response to fluid-induced surface forces. However, neither the applied loads nor deformations were actually measured. Instead, wall shear was used as an approximation of the true loadings. Also, the angular deformation of a streamer's leading edge was used as an indicator of shear strain. These are oversimplifications. The actual applied fluid loads are complex, three-dimensional, spatially varying forces and not a constant, unidirectional wall shear force. Indeed, by assuming a constant wall shear stress one completely neglects the interplay between biofilm structure and local hydrodynamics. Furthermore, the deformation exhibited by the streamers was equally complex. Thus, the transferability of the calculated elastic moduli and shear coefficients presented in these studies is extremely limited. Despite these criticisms, empirically measuring local velocity vectors and displacement on this scale in two or three dimensions would have been difficult if not impossible. A validated computational simulation of this specific fluid-structure interaction would alleviate the need to measure velocities and deformations locally and act as a tool for further investigation of the biofilm response to turbulent flow.

#### Goal of the Computational Model

It is well known that there are few analytical solutions to turbulent flow problems. This is due primarily to the problem of turbulent closure in the Reynolds averaged

governing equations (Wilcox, 1993). Certainly, the same obstacle prevents one from analytically solving a fluid-structure interface problem as complex as the one outlined in the previous paragraphs. Due to practical limitations, experimentally measuring many of the quantities of interest in the biofilm streamer response are also difficult. Historically, these two approaches, empiricism and theory, constituted the whole discipline of fluid dynamics. However, computational fluid dynamics represents a third approach to solving problems in this field (Anderson, 1995). Properly posed, experimental measurements of mean velocity, problem geometry and a biofilm constitutive model can be combined with a numerical code to more accurately simulate streamer deformation as a function of the bulk fluid velocity. The goal was to develop a computational fluid-structure interface (FSI) model, numerically validate the Burger constitutive law and utilize the model in simulations of the biofilm streamer response to turbulent flow conditions. The ANSYS software was used as a platform for the development of this numerical model.

#### Finite Element Platform

ANSYS version 7.1 is a general-purpose finite element program that uses numerical methods to solve the fundamental equations of motion, equations of state and constitutive laws for different types of problems including structural, thermal, and fluid mechanics. It is a self-contained program in that the pre-processor, processor (i.e. solver), and post-processor are all integrated into a single interface. While the program has been an industry standard for many years, recent advances now allow for coupled field analyses. Coupled structural and fluid problems can be solved iteratively.

Additionally, the ANSYS software allows for a high degree of customization by the user through a scripting language called the ANSYS Parametric Design Language (APDL). The coupled field ability of ANSYS, together with the flexibility gained through APDL, made ANSYS version 7.1 an ideal platform for the development of a numerical biofilm model.

### Solid Mechanics in ANSYS

The partial differential equations governing strain deformation can be developed through the principle of virtual work (ANSYS, Inc., 2003a). The finite element method or FEM employed by the ANSYS program divides the solution domain (i.e. a structural shape, biofilm structure) into several simply shaped regions called elements. Approximate solutions to the partial differential equations are developed for each element and then assembled, all the while ensuring continuity at the inter-element boundaries. These equations, together with boundary conditions (e.g. applied loads, constraints) and a constitutive law, can be used to describe a well-posed solid mechanics problem.

Boundary conditions are problem-specific. In the context of the biofilm streamer model, constraints refer to prescribed displacements, or more appropriately, lack of displacements. For example, the base of the biofilm structure is fixed in the computational model. The other type of boundary condition is an applied load. Clearly, applied loads such as shear and pressure forces involved in this fluid-structure interface (FSI) problem were not known *a priori*. Indeed, this is precisely why the structural field

must be coupled with the fluid field. In that way, the applied loads are derived from the fluid solution.

### Biofilm Constitutive Law in ANSYS

The aforementioned Burger model was used to relate stress and strain in the FEM program. While the Burger model was experimentally validated in Chapter 5, two issues were resolved prior to using this linear viscoelastic description within the ANSYS program. First, the formulation (59) presented in Chapter 5 was converted into a format suitable to ANSYS. Secondly, once incorporated into ANSYS, the material model was numerically validated against an analytical solution.

ANSYS Representation of the Burger Model. In general, non-linear material behavior can be incorporated into ANSYS in several ways. The specific method is dependent upon the element type used which, in turn, is mandated by the type of non-linear behavior one wishes to model. Modeling linear viscoelasticity in a couple-field setting necessitated the use of the PLANE18X family of elements. These are two dimensional structural elements that permit a linear surface force distribution along each face. The PLANE182 element is a 4 node quadrangle. The PLANE183 element is an 8 node quadrangle; there are 4 corner nodes and 4 mid-side nodes. The PLANE182 element was used in validation of the material model. Due to the more realistic behavior one gains from an 8 node element over a 4 node element, the PLANE183 was used in the biofilm streamer and rolling biofilm models.

With the PLANE18X elements, viscoelasticity can be incorporated into an ANSYS model by describing the time-dependent shear relaxation function,  $G(t)$ , in terms of a Prony series of the form (ANSYS, Inc., 2003a)

$$C(t) = C_0 + \sum_{i=1}^k C_i e^{-a_i t}, \quad (60)$$

where  $t$  is the independent variable (e.g. time),  $a_i$  are the  $k$  exponential Prony coefficients and  $C_i$  are the  $k+1$  linear Prony coefficients. This equation is conveniently of the same form as the relaxation function for a generalized Maxwell model. A generalized Maxwell model is a sequence of  $n$  Maxwell units (Figure 2a) arranged in parallel. The relaxation modulus for the generalized Maxwell model is

$$G(t) = G_1 e^{-\frac{t}{\tau_1}} + G_2 e^{-\frac{t}{\tau_2}} + \dots + G_{n-1} e^{-\frac{t}{\tau_{n-1}}} + G_n e^{-\frac{t}{\tau_n}}, \quad (61)$$

where  $G_1$  through  $G_n$  and  $\tau_1$  through  $\tau_n$  are the shear moduli and relaxation times for the  $n$  parallel Maxwell units. Equating (60) with (61) for  $n = k = 2$  results in Prony series representation of two parallel Maxwell units or a 4 element General Maxwell model.

$$G(t) = G_\infty + \sum_{i=1}^n G_i e^{-t/\tau_i} \quad n = 2. \quad (62)$$

In this form,  $G_\infty$  is a representation of the shear modulus at  $t = \infty$ . It has been shown that biofilm acts as a viscoelastic fluid (Stoodley et al, 1999a). As a viscoelastic fluid, by definition biofilm cannot sustain shear stresses at  $t = \infty$ . Therefore,  $G_\infty$  can be neglected. Thus (62) reduces to

$$G(t) = G_1 e^{-t/\tau_1} + G_2 e^{-t/\tau_2}. \quad (63)$$

From the preceding development, it is clear that representing any General Maxwell model in Prony series form is relatively straightforward. This is due to the fact that the linear relaxation modulus is itself the summation of all parallel linear subcomponents (e.g. Maxwell units). Moreover, the shear moduli and relaxation times in each Maxwell unit are easily identifiable as the linear and exponential coefficients, respectively, in the Prony series. This is not the case with a 4 element Burger model. The Burger model is a serial combination of Maxwell and Kelvin elements; its relaxation modulus is not the summation of the subcomponents because they are not arranged in parallel. As outlined in Chapter 3, the relaxation modulus must be determined through a Laplace transform of the creep compliance (see Appendix A). This Laplace transform yields the following form of the Burger relaxation modulus

$$G(t) = \frac{\left( \eta_1 - \frac{\eta_1 \eta_2}{G_2} T_1 \right)}{\sqrt{\left( \frac{\eta_1}{G_1} + \frac{\eta_1}{G_2} + \frac{\eta_2}{G_2} \right)^2 - 4 \frac{\eta_1 \eta_2}{G_1 G_2}}} e^{-T_1 t} + \frac{-(\eta_1 - \frac{\eta_1 \eta_2}{G_2} T_2)}{\sqrt{\left( \frac{\eta_1}{G_1} + \frac{\eta_1}{G_2} + \frac{\eta_2}{G_2} \right)^2 - 4 \frac{\eta_1 \eta_2}{G_1 G_2}}} e^{-T_2 t}, \quad (64)$$



where  $\eta_1$  is the viscosity coefficient in the Maxwell element,  $G_1$  is the (elastic) shear coefficient in the Maxwell element,  $\eta_2$  is the viscosity coefficient in the Kelvin element,  $G_2$  is the shear coefficient in the Kelvin element, and  $T_1$  and  $T_2$  are grouped terms shown in equations (19) and (20). By equating equation (60) with equation (64) the following linear and exponential Prony coefficients can be explicitly represented in terms of Burger model spring and dashpot parameters:

$$C_1 = \frac{\left[ \eta_1 - \frac{\eta_1 \eta_2}{G_2} \left( \frac{\frac{\eta_1}{G_1} + \frac{\eta_1}{G_2} + \frac{\eta_2}{G_2} - \sqrt{\left( \frac{\eta_1}{G_1} + \frac{\eta_1}{G_2} + \frac{\eta_2}{G_2} \right)^2 - 4 \frac{\eta_1 \eta_2}{G_1 G_2}} \right)}{2 \frac{\eta_1 \eta_2}{G_1 G_2}} \right]}{\sqrt{\left( \frac{\eta_1}{G_1} + \frac{\eta_1}{G_2} + \frac{\eta_2}{G_2} \right)^2 - 4 \frac{\eta_1 \eta_2}{G_1 G_2}}}; \quad (65)$$

$$C_2 = \frac{\left[ \eta_1 - \frac{\eta_1 \eta_2}{G_2} \left( \frac{\frac{\eta_1}{G_1} + \frac{\eta_1}{G_2} + \frac{\eta_2}{G_2} + \sqrt{\left( \frac{\eta_1}{G_1} + \frac{\eta_1}{G_2} + \frac{\eta_2}{G_2} \right)^2 - 4 \frac{\eta_1 \eta_2}{G_1 G_2}} \right)}{2 \frac{\eta_1 \eta_2}{G_1 G_2}} \right]}{\sqrt{\left( \frac{\eta_1}{G_1} + \frac{\eta_1}{G_2} + \frac{\eta_2}{G_2} \right)^2 - 4 \frac{\eta_1 \eta_2}{G_1 G_2}}}; \quad (66)$$

$$a_1 = \frac{\frac{\eta_1}{G_1} + \frac{\eta_1}{G_2} + \frac{\eta_2}{G_2} - \sqrt{\left(\frac{\eta_1}{G_1} + \frac{\eta_1}{G_2} + \frac{\eta_2}{G_2}\right)^2 - 4\frac{\eta_1\eta_2}{G_1G_2}}}{2\frac{\eta_1\eta_2}{G_1G_2}}; \quad (67)$$

$$a_2 = \frac{\frac{\eta_1}{G_1} + \frac{\eta_1}{G_2} + \frac{\eta_2}{G_2} + \sqrt{\left(\frac{\eta_1}{G_1} + \frac{\eta_1}{G_2} + \frac{\eta_2}{G_2}\right)^2 - 4\frac{\eta_1\eta_2}{G_1G_2}}}{2\frac{\eta_1\eta_2}{G_1G_2}}; \quad (68)$$

Linear viscoelastic materials can then be modeled in ANSYS by converting Burger parameters into Prony series coefficients through the use of equations (65) through (68).

This was done internally in each ANSYS input file through APDL statements.

#### Numerical validation of the BURGER Model

In the field of computational modeling and simulation the terms validation and verification refer to a determination of the *soundness* of a model or code. Specifically, a verification assessment determines if a code is correctly solving the conceptual model as envisioned by the programmer. For this reason, it is often described as “solving the equations right”. ANSYS 7.1 has been subjected to a thorough verification process and for additional information, the reader is directed to the official ANSYS, Inc. internet site at [www.ansys.com](http://www.ansys.com). Conversely, validation has been described as “solving the right equations”. The American Institute of Aeronautics and Astronautics provides a more

formal definition: "(Validation is) the process of determining the degree to which a model is an accurate representation of the real world from the perspective of the intended uses of the model." (AIAA, 1998) Realistically, models cannot be validated for every possible outcome or solution. The typical approach, however, is to compare model results to a well documented solution or to an analytical solution. If the model matches a reliable solution over a limited range, the model's applicability can be extrapolated beyond that range provided the governing equations are still valid.

Description of the Burger Model Assessment. While it was unnecessary and beyond the scope of this work to verify that the ANSYS code was correctly applying the Prony series representation of a material's relaxation modulus, it was necessary to validate that equations (64) through (68), and the ANSYS interpretation of those equations, correctly represented the relaxation modulus of a Burger material. To accomplish this, a simple finite element model analogous to a relaxation test was developed. This model (Figure 30) consisted of a single PLANE182 element of unit length and width was anchored at the two low-y nodes, 1 and 2. At the start of the test, nodes 3 and 4 were subjected to a fixed 1% strain by instantaneously displacing these nodes a distance of  $\delta = 0.01$ . The duration of the test, 180 seconds, was selected to maintain consistency with the rheometry experiments of Chapter 4. The Burger parameters were then selected to produce a 95% reduction in elemental stresses over the 3 minute effective relaxation time period. The resulting Burger parameters are presented in Table 7. The input file used to generate this ANSYS model is presented in Appendix C.

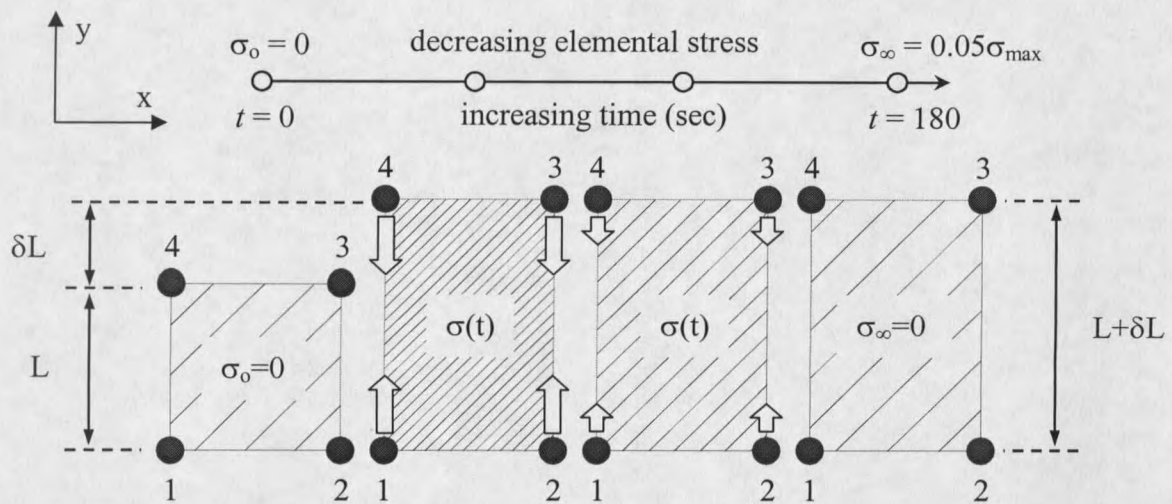


Figure 30. Relaxation Test Simulation. A four node quadrangle was subjected to a constant displacement,  $\delta L$ , by displacing nodes 3 and 4. Moving in time from left to right, the elemental stress state (illustrated by the density of hatching) instantaneously increases at  $t > 0$  then begins to decrease with time.

Table 7. Burger Parameters Used in Validation of ANSYS/Prony Series Conversion

| $E_1$ | $\eta_1$ | $E_2$ | $\eta_2$ | $\nu$ |
|-------|----------|-------|----------|-------|
| 3.54  | 204      | 10.0  | 1500     | 0.33  |

Results of the Burger Model Assessment. Elemental stress outputs,  $\sigma^S(t)$ , from the simulation were written to an ASCII file every 0.05 seconds. The constant strain maintained throughout the simulation,  $\epsilon_0$ , was normalized against these stress values to determine the numerically-calculated time-dependent relaxation modulus,  $G^S(t)$ . Using equation (64) and the Burger parameters of Table 7, the analytically-calculated time-

dependent relaxation modulus,  $G^A(t)$  was also calculated . A comparison of these two moduli is shown in Figure 31.

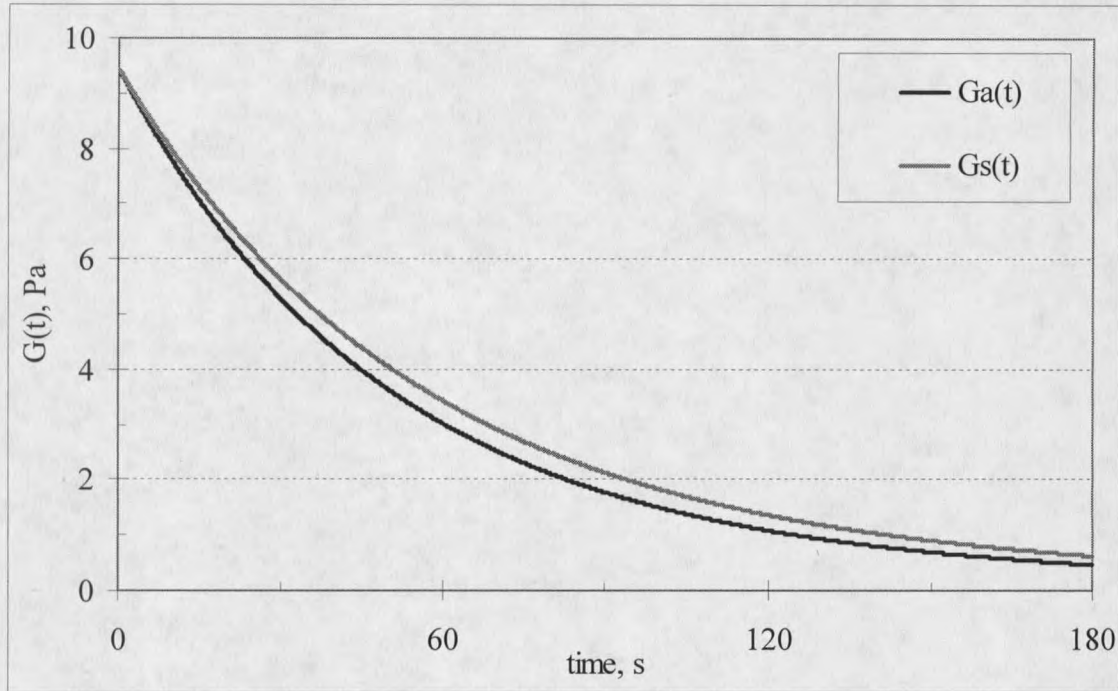


Figure 31. Comparison of the Analytically Calculated Time-dependent Relaxation Modulus,  $G^A(t)$  and the ANSYS/numerically-approximated Time-dependent Relaxation Modulus,  $G^S(t)$ .

Figure 31 indicates close agreement between these two relaxation modulus functions. A more critical evaluation is provided by examining the absolute and cumulative error between the two values. The expressions

$$e(t_i) = |G^S(t_i) - G^A(t_i)|, \quad (69)$$

and

$$E(t_n) = \sum_{i=1}^n e(t_i), \quad (70)$$

represent the absolute and cumulative errors, respectively, where  $i$  is the time index and  $n$  is the number of time divisions. Figure 32 illustrates the changes in these values over time. The tabular results for the entire test, including these error calculations are presented in Appendix D.

At the point of greatest absolute error,  $e(53.95 \text{ sec}) = 0.414$ , the relative error between the true and approximate relaxation moduli was 12 %. While the magnitude of this error was not large enough to significantly affect the results, it was large of enough to warrant a sensitivity analysis. A series of Burger models was run to determine if relative error was dependent on time, burger value parameters, or strain setting. The results indicated this error was relatively insensitive (i.e. no change in order of magnitude) to these variables. This is not surprising given that the relaxation modulus itself is independent of strain as outlined in previous chapters. Further investigation revealed that the error observed in Figure 32 is an expression of the numerical approximation used within the ANSYS (Doyle, 2003). The numerical methods often approximate non-linear solutions through the linearization of the governing equations. While element shape and time step can mitigate this error, the FEM solution is ultimately an approximation. For comparison, a similar simulation was performed using a single Maxwell element. The validated input file for this simulation was provided by technical support at ANSYS, Inc.

(Thompson and Chang, 2003). The results of a Maxwell relaxation simulation produced an error of approximately the same order as that seen in the Burger model. The input file and tabular results from the validation of the Maxwell model, as provided by ANSYS, Inc., are presented in Appendices D and E.

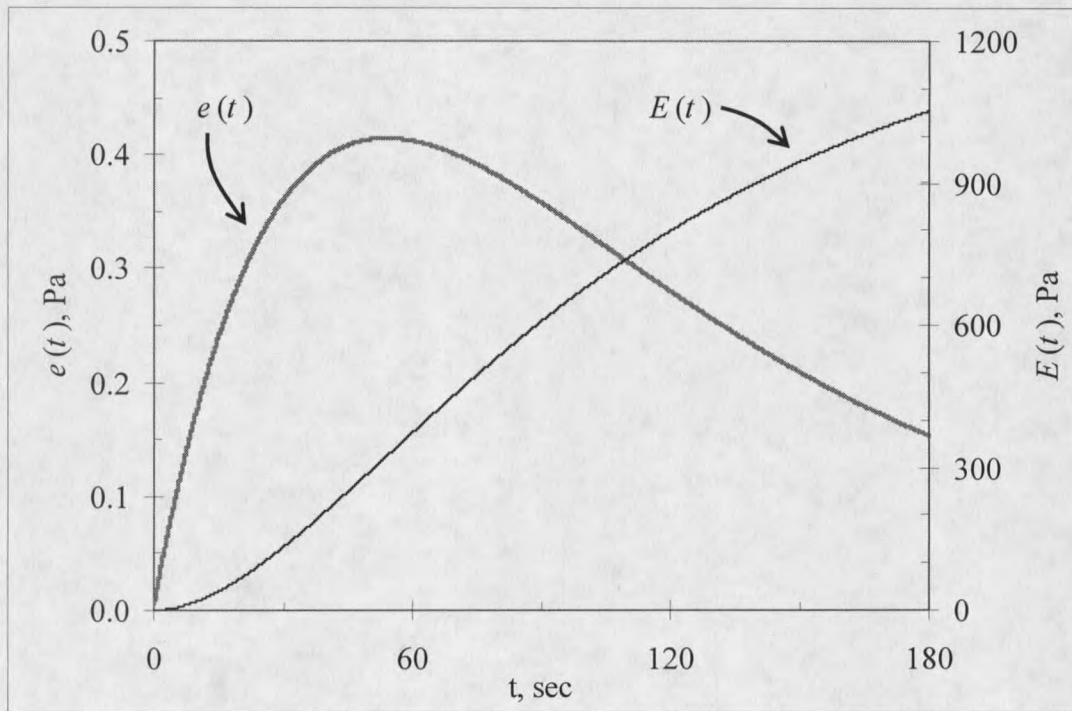


Figure 32. Absolute and Cumulative Errors Between the Analytical Relaxation Modulus,  $G^A(t)$  and the ANSYS Relaxation Modulus,  $G^S(t)$ .

### Computational Fluid Dynamics in FLOTRAN

FLOTRAN is an ANSYS derived computational fluid dynamics (CFD) software package. Although FLOTRAN also employs a finite element method to solve problems, the approach is quite different from that of the ANSYS structural solver. FLOTRAN

partitions a solution domain (i.e. fluid space) into elements. Nodal formulations of the Navier-Stokes equations and, if appropriate, a turbulence model are assembled to complete an FEM solution. A built-in CFD constitutive law describing the behavior of the bulk fluid (i.e. water) is available. A variety of fluid solvers are available, but all involved iteratively solving the linearized, non-linear, governing partial differential equations using a method based on the SIMPLE technique developed by Patankar (1975).

### Fluid Boundary Conditions

As with the solid model, boundary conditions must also be prescribed in the fluid model. Applied loadings, in this context, were average velocity at an inlet and a pressure boundary at the outlet. Constraints included no-slip and zero-displacement conditions along the flow boundaries. The displacement of the interfacial biofilm streamer could not be prescribed. These interfaces were the boundaries that were coupled with the solid solution using an iterative FSI algorithm.

### Fluid-Structure Interaction

A coupled field analysis in ANSYS is carried out by separately creating a two models, each governed by a different field of physics (e.g. fluid mechanics, structural mechanics, heat transfer), and then coupling them across an interface. An interface refers to any similar or dissimilar mesh boundary between the two models. Coupling describes the process by which the solutions to each model are linked. Fluid-structure interactions in ANSYS must be performed using a *weak sequential coupling algorithm*. This



algorithm is described in Figure 33. The algorithm proceeds toward a solution by running through a nested set of loops: the outer or “time loop” and the inner or “stagger loop”. During execution of the stagger loop, the structural FEM analysis is performed, and fluid mesh is updated as a result of the deformed structural mesh. The fluid analysis is then carried out and the resulting shear and pressure loads are transferred across the FSI boundary to the structure. The stagger loop is repeated until changes in the transferred forces falls below specified convergence criteria. The outer loop advances through time toward a converged solution.

#### Test Case: Simple Shear Cube Problem

Once the material model was decided, and a method for incorporating the Burger model into ANSYS was calculated, a test problem was created. One of the motivations for the development of a computational biofilm structure model was to be able to examine the problem in a more complex, multi-dimensional setting than offered by Stoodley, et al. (1999b). Thus, a two-dimensional test problem was created. The geometry of real biofilm structures is highly complex. Often, this complexity can hide results that might be illuminated by a more elementary shape. For this reason, a shear cube problem was used as a test problem. The simple geometry of a 2 dimensional cube, anchored at its base and subjected to fluid forces on the other 3 sides was complex enough to serve as a test case and geometrically simple enough to elucidate the fluid-structure interaction and material behavior. Furthermore, this problem more accurately

approximates the simplification made by Stoodley, et al. (1999a, 1999b) in calculating shear moduli.

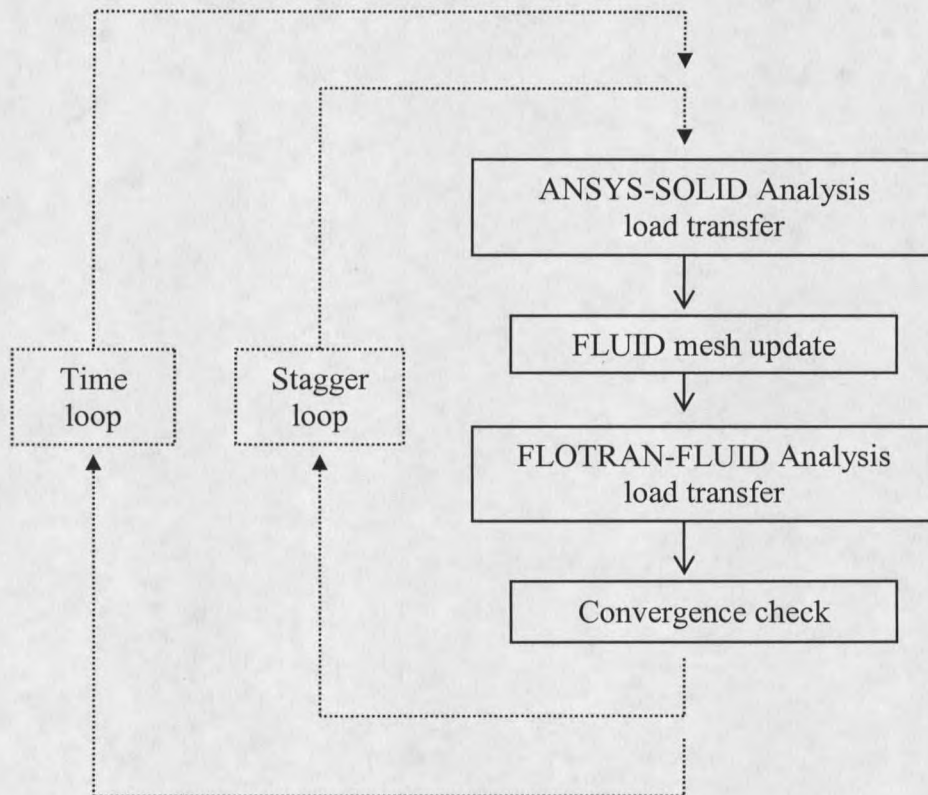


Figure 33. Weakly-Coupled FSI Algorithm. Stagger loop iterates until convergence criteria placed on transferred interfacial forces and displacements are met. Once convergence criteria are achieved, solution advances through time.

### Problem Description

This simulation was intentionally set on the same scale as the problem studied by Stoodley, et al. (1990b, 2002). A 200  $\mu\text{m}$  by 200  $\mu\text{m}$  compliant cube representing an attached biofilm structure was anchored at the low  $y$  side to a channel that was 3 mm in

height (Figure 34). Water ( $\mu = 1.0 \times 10^{-6}$  kg/mm·s,  $\rho = 1.0 \times 10^{-6}$  kg/mm<sup>3</sup>) moved from the left to right at a mean velocity of 1.0 m/s. A Burger material model with the following parameters was used as a constitutive law for the shear cube:  $G_1 = 250$  kg/mm·s<sup>2</sup>,  $\eta_1 = 750$  kg/mm·s,  $G_2 = 500$  kg/mm·s<sup>2</sup>,  $\eta_2 = 5000$  kg/mm·s. Loads were transferred across the dissimilarly-meshed low x, high y, and high x interfaces between the fluid and the cube. The outlet was specified as a pressure boundary. The flow boundaries were prescribed as hydraulically smooth. The duration of the simulation was 60 seconds. The solution was found when convergence criteria fell below a specified value, indicating the simulation had reached a steady-state.

#### Solution and Convergence

The input file (see Appendix E) was created and submitted to the ANSYS processor/solver. The machine used was a Silicon Graphics, Inc. SGI Origin 2100. The Origin 2100 has eight 350 MHz processors and 4 gigabytes of RAM. The time to solution for this simulation was approximately 7.5 hours. The convergence criterion used by the ANSYS FSI algorithm is the Euclidean norm,  $l_2$ , of the following quantities being transferred across the interfaces: force in the x-direction,  $F_x$ ; force in the y-direction,  $F_y$ ; displacement in the x-direction,  $U_x$ ; and displacement in the y-direction,  $U_y$ . The time-marching method of the solution can be regarded as quasi-static. That is, at each time step a 'solution' is reached. A 'converged' solution for each time step is one where the differences between the Euclidean norms of each of these four quantities on either side of the fluid-structure interface approaches zero. The convergence criteria were set at 0.0001 of the square root of the sum of the squares (SRSS) of the applied

forces or displacements. The values of the  $l_2$  norms after the final iteration are presented in Table 8.

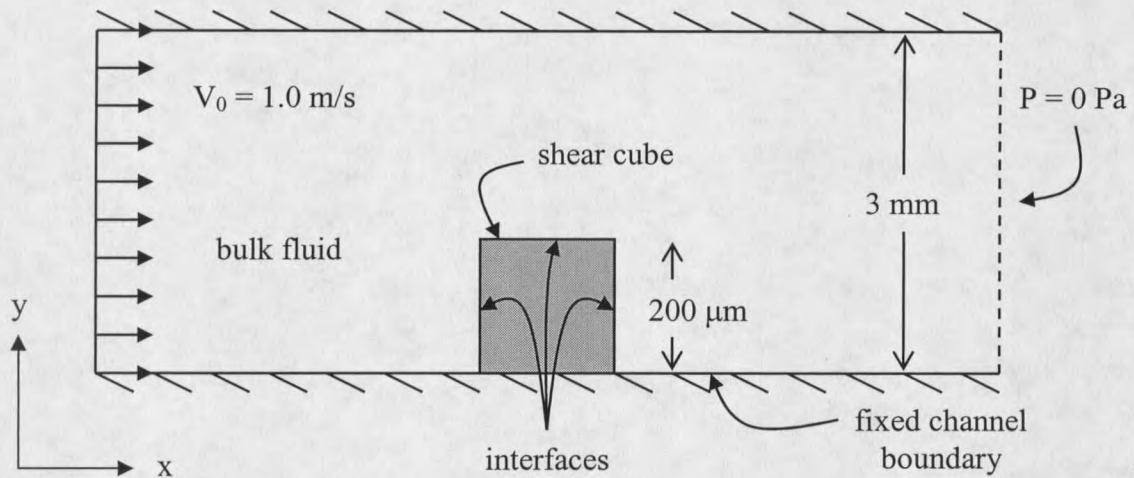


Figure 34. Shear Cube Model. The flowing water impinges on the cube. Pressure and shear forces are transferred across the interfaces which cause the cube to deform.

Table 8. Convergence Criteria After Final FSI Iteration.

| quantity | $F_x$  | $F_y$  | $U_x$  | $U_y$  |
|----------|--------|--------|--------|--------|
| Unit     | mN     | mN     | mm     | mm     |
| $l_2$    | -0.320 | -0.431 | -0.110 | -0.115 |

### Discussion of Results

The coupled solution illustrates the interaction between bulk fluid velocity and the stress state in biofilm structures. Figure 35 shows an ANSYS generated contour plot of the internal von Mises stresses in the structural solution. Figure 36 shows another ANSYS generated contour plot of the fluid velocity in the structural solution. The

maximum tensile stress of  $\sigma_x = 4.64$  mPa,  $\sigma_y = 10.5$  mPa,  $\sigma_{xy} = 3.17$  mPa was observed at the intersection of the low x face and the base of the cube. The maximum compressive stress of  $\sigma_x = -2.99$  mPa,  $\sigma_y = -6.77$  mPa,  $\sigma_{xy} = 1.88$  mPa was observed at the intersection of the high x face and the base of the cube. The implications of these stress extremes, given the homogenous, isotropic material model, are that the leading and trailing edges of the basal area may be most susceptible to flow-induced failure. This may explain empirical observations of detachment and rolling phenomena noted by other investigators (Rupp, et al., 2002).







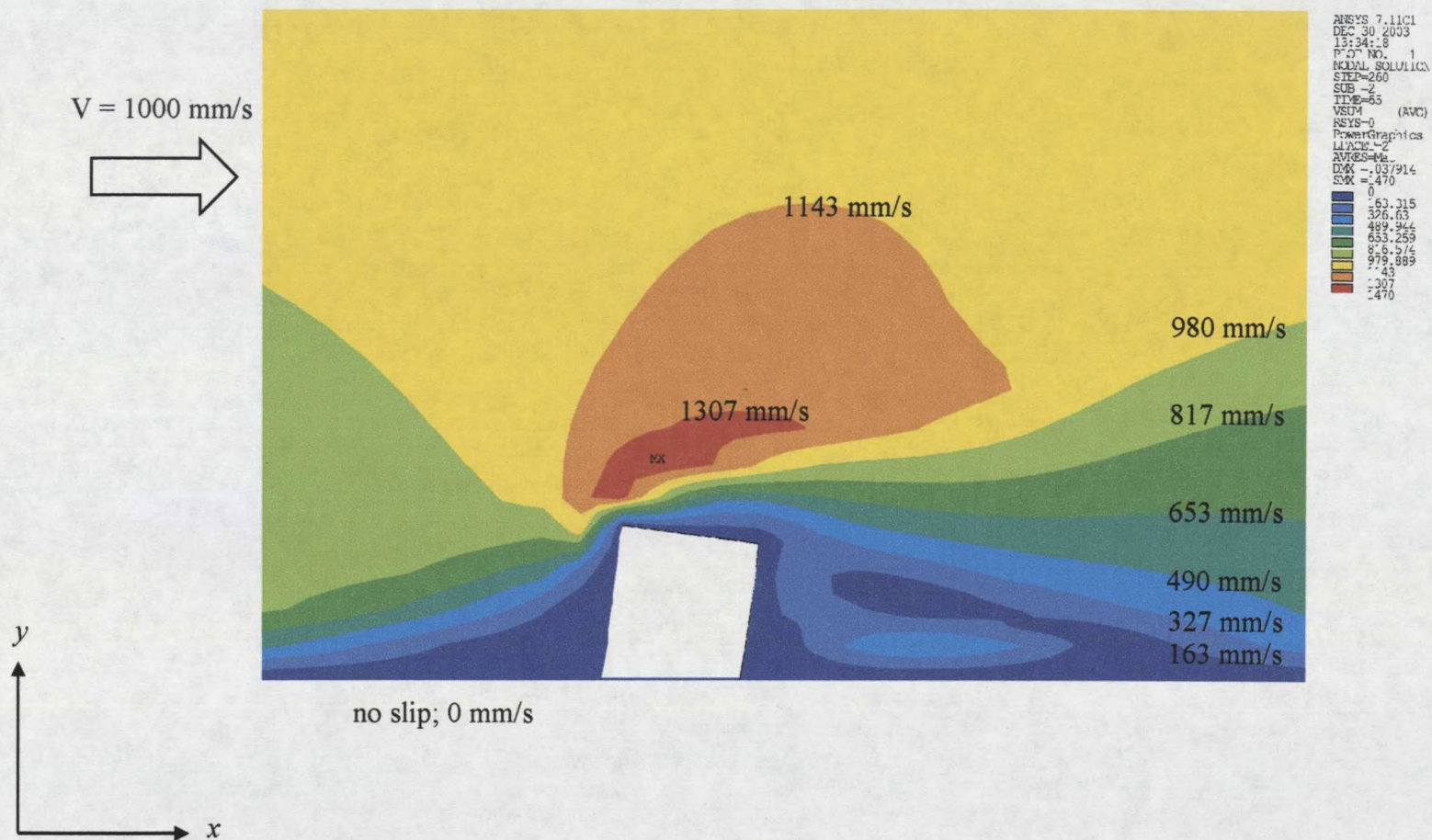


Figure 36. ANSYS-Generated Contour Plot of Nodal Velocity Norms in the Bulk Fluid. The mean channel velocity was 1000 mm/s.

Effective shear angle. The converged solution demonstrated that a mean velocity of 1.0 m/s caused a maximum nodal deformation of 37.0  $\mu\text{m}$  at the intersection of the high y and low x interfaces. While the leading edge of the deformed cube was no longer a straight line, it was close enough to warrant the calculation of an effective shear angle,  $\gamma_{\text{eff}}$ . The angle tended by the original low x face and the line formed by the endpoints of this displaced face was  $\gamma_{\text{eff}} = 7.96$  degrees. Moreover, since this simulation was hydraulically similar to the experimental design of Stoodley, et al. (1999b), an effective shear modulus,  $G_{\text{eff}}$ , was calculated in the same manner.

$$G_{\text{eff}} = \frac{\tau_w}{\gamma_{\text{eff}}}, \quad (71)$$

where  $\tau_w$  is the wall shear stress. This yielded a  $G_{\text{eff}}$  of 36  $\text{N/m}^2$  which was on the same order of magnitude as the value ( $G_{\text{eff}} = 27 \text{ N/m}^2$ ) reported by Stoodley et al. (1999b). These calculations are presented in Appendix F. However, the results of the FEM simulation indicate that the mean force acting on the shear cube was 175  $\text{N/m}^2$ . Since there was little error in the approximation of  $\gamma_{\text{eff}}$ , one can conclude that the discrepancy lies in the use of  $\tau_w$  as an estimate of the forces acting on the biofilm structure. Therefore, if wall shear is an inappropriate estimate for the total force on a biofilm structure, the use of equation (71) is, by extension, inappropriate.

Comments on Material Behavior. As outlined in Chapter 3, the four element Burger model is the simplest of all analogs that can exhibit all three types of linear



viscoelastic strain behavior: an instantaneous elastic response, a time-dependent elastic response and viscous flow. These behaviors were evident in the shear cube simulation.

### Summary

The experimental results of Chapter 4 and the curve fitting of Chapter 5 presented a basis for the development of a computational biofilm model. The ANSYS FEM software was used as a platform for this model. Using a Laplace transform, the Burger formula developed in Chapter 5 was converted into a Prony series for inclusion in the FEM simulation. This numerical representation of the Burger model was then compared to the analytical solution of a relaxation stress response. The results indicated close agreement. Moreover, the difference was an expression of the limitation of the linearized numerical approximation and not a miscalculation in the Prony series conversion. This comparison validated the use of a linear viscoelastic Burger model in ANSYS.

The next step involved the creation of a biofilm structure-fluid interaction test case. The seminal work by Stoodley, et al., (1999b, 2002) provided an ideal scale and setting for the model. The shape of the attached biofilm was simplified to prevent complex geometry from masking otherwise salient behavior. The test simulation converged and demonstrated several facts. The areas of high internal stress within the biofilm structure were located at the leading and trailing edges of the attached basal edge. This observation may provide insight into the nature of detachment and rolling phenomena. The test simulation also highlighted failings in the method proposed by Stoodley, et al., (1999b) for calculating elastic coefficients of biofilm streamers.

With the development of the computational model complete and the insight provided by a simple shear cube evident, the next step was to add complexity to the model. In the next chapter, two applications are explored. First, the model is used to determine the Burger material parameters if the deformation is known *a priori*. Secondly, the numerical simulations are used to investigate the phenomenon of rolling biofilm and to develop a quasi-static rolling equilibrium model.

## APPLICATION OF THE COMPUTATIONAL MODEL

In the preceding chapter, the development of a computational model of the biofilm response to turbulent flow was outlined. This numerical model utilized a linear viscoelastic material description that was experimentally determined. Additionally, the incorporation of this nonlinear stress-strain relation was validated against an analytical solution. This systematic approach to experimentation, model development and validation justified a high degree of confidence in the resulting computer model. The potential uses for this model are numerous. This chapter details a specific application of the computational model; parametric studies were used to examine the rolling biofilm phenomenon.

### Analysis of Rolling Biofilm Phenomena

There is evidence that biofilm structures can move along a channel boundary or across the substratum (Rupp, et al., 2002). This movement is significant because it contradicts assumptions that biofilm movement results from either large-scale detachment or small-scale erosion. This behavior can be described as a rolling translation. In some instances this translation precipitates wholesale detachment. In other instances, the result is not detachment but a downstream migration of the biofilm. The computational model provided an excellent mechanism for investigating rolling biofilm behavior due to fluid forces.

### Problem Description

The biofilm structure was modeled as a half circle in a two dimensional setting. This shape was chosen for two reasons. First, in three dimensions, a hemispherical shape is analogous to an idealized growth pattern for a microbial community in the absence of a strong unidirectional flow field. Thus, this hemispherical shape could be regarded as the starting point of biofilm rolling behavior. Secondly, this elementary shape could illustrate behavior that more complex geometry might mask. For comparison purposes, the scale and setting again mimicked the experimental set-up used by Stoodley et al (1999b). The solution domain was defined by a 40 mm long by 3 mm high two-dimension channel. At  $t = 0$ , the velocity of the bulk fluid was set at 1 m/s. Water, flowing in the positive  $x$  direction, entered through an inlet at the low  $x$  boundary and exited the channel through a pressure outlet at the high  $x$  boundary. The forces generated by the fluid acted to deform the biofilm shape.

Flow Development. The experimental model used glass flow tubes that were 200 mm in length. The simulated channel was 40 mm in length with the biofilm structure placed near the approximate center. Thus, the model allowed for a 20 mm entrance length and a 20 mm exit length. This distance was chosen as a compromise between the computational demands of modeling the entire length and the length necessary to prevent entrance and exit effects. Initial testing indicated that the 20 mm exit length was sufficient to prevent any "back flushing" effects due to the presence of the high  $x$  pressure boundary. However, it was also revealed that the flow field (i.e. velocity

distribution) was not yet fully developed as it approached the biofilm structure. A channel width to length aspect ratio of 30 is a commonly used “rule of thumb” estimate to ensure the development of turbulent flow in smooth conduits. This estimate would have necessitated an entrance length of 90 mm and a total length of 110 mm. Based on the meshing scheme, a model length of 40 mm contained 5426 elements. Modeling 100 mm of the channel would have increased the number of elements nearly threefold. To mitigate the need for an extended entrance length, a non-uniform velocity distribution was applied at the inlet boundary.

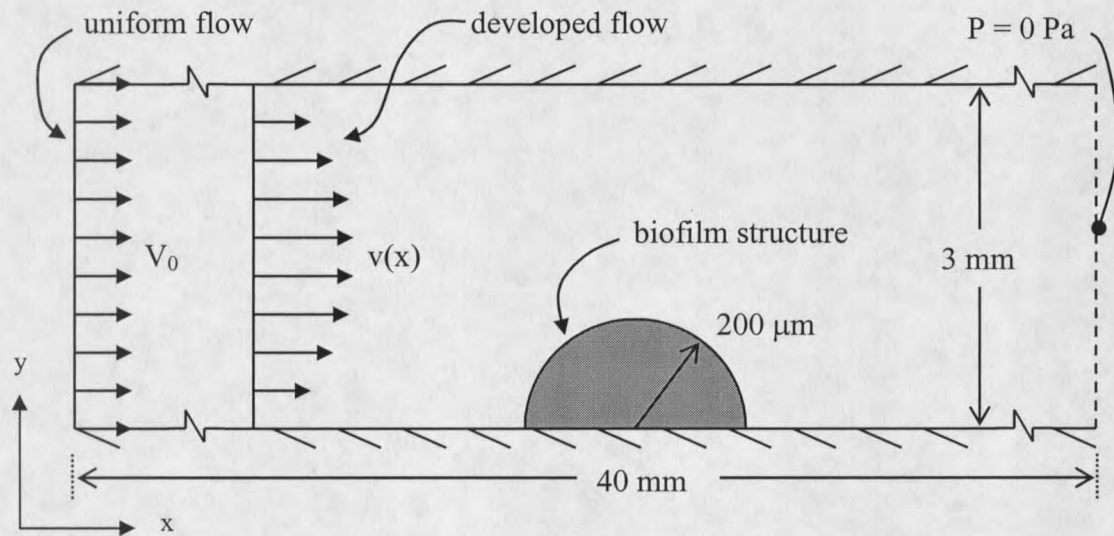


Figure 37. Rolling Biofilm Model. Water moves from left to right impinging on the attached viscoelastic biofilm hemisphere.

The velocity distribution in turbulent flows is not constant but considerably more uniform than the parabolic distribution observed in laminar flows. The velocity

distribution in laminar flow can be determined analytically (Janna, 1993). This is not possible under turbulent conditions. However, a number of empirical formulae exist. For smooth pipes, Giles, et al. (1956) suggests

$$v(y) = V\sqrt{f}\left(2\log\frac{y}{R} + 1.32\right) + V, \quad (72)$$

where  $v(y)$  is the velocity at a distance  $y$  from the center of the conduit,  $R$  is the radius of the conduit,  $V$  is the mean velocity, and  $f$  is the friction factor. This equation (72) requires an estimate of the friction factor. For smooth pipes, the Blasius equation can be used to approximate the friction factor (Debler, 1990),

$$f = \frac{0.316}{RE^{0.25}}. \quad (73)$$

Using custom code written in APDL, these two equations were combined to specify velocities at the nodes on the inlet boundary (see Appendix G).

Centroid of an Arbitrarily Oriented Quadrangle. A means of comparison between simulations resulting in different strain histories (and accordingly different final geometries) was needed. The location of the biofilm structure's centroid was chosen as this metric. The initial centroid location was readily calculated. Unfortunately, the ANSYS program provided no method for determining the deformed shape's centroid as a

function of time. Therefore, an APDL macro was written to perform this calculation at each time step. The details of this calculation follow.

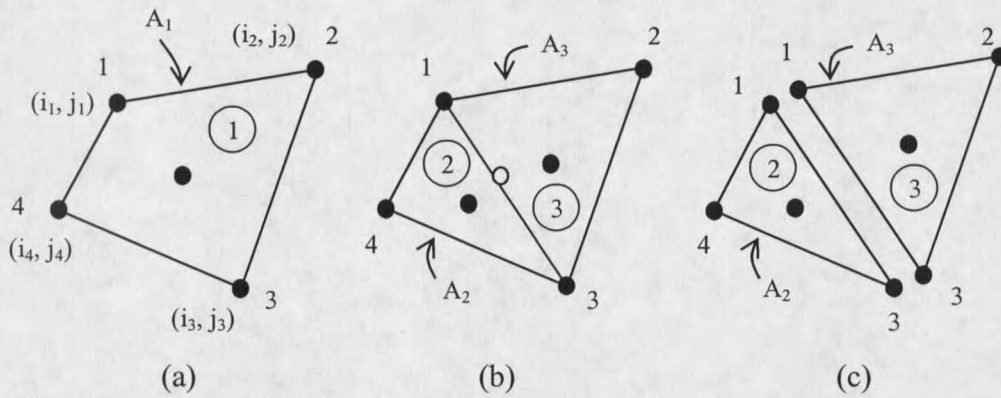


Figure 38. Centroid of an Arbitrarily Oriented Quadrangle. The centroid of a quadrangle element (a) can be determined by dividing it into two triangles (b). A relationship between the centroid of two triangles (c) and the quadrangle exists.

A arbitrarily oriented quadrangle (Fig. 39a) defined by the nodes 1 through 4 located at coordinates  $(i_1, j_1)$  through  $(i_4, j_4)$  can be regarded as two triangular elements (Fig. 39b) defined by nodes 1, 2, 3 and 1, 3, 4. The area of shape 2 is

$$A_2 = \left(\frac{1}{2}\right) \sqrt{(i_3 - i_1)(j_2 - j_1) - (j_3 - j_1)(i_2 - i_1)}, \quad (74)$$

where  $i_1$  through  $i_4$  represent the horizontal coordinates of nodes 1 through 4, and  $j_1$  through  $j_4$  represent the vertical coordinates of nodes 1 through 4. Similarly, the area of shape 3 is

$$A_3 = \left(\frac{1}{2}\right) \sqrt{(i_4 - i_1)(j_3 - j_1) - (j_4 - j_1)(i_3 - i_1)}. \quad (75)$$

The  $i, j$  coordinates of the centroid of shape 2 are

$$C_2 = (C_2^i, C_2^j) = \left( \left(\frac{1}{3}\right)(i_1 + i_2 + i_3), \left(\frac{1}{3}\right)(j_1 + j_2 + j_3) \right), \quad (76)$$

and the  $i, j$  coordinates of shape 3 are

$$C_3 = (C_3^i, C_3^j) = \left( \left(\frac{1}{3}\right)(i_1 + i_3 + i_4), \left(\frac{1}{3}\right)(j_1 + j_3 + j_4) \right). \quad (77)$$

It can be shown then that there exists a relationship between the centroids of these two triangles (Fig. 39c) and the quadrangle (Fig. 39a). The relationship is an area weighted average that takes the form

$$C_1 = (C_1^i, C_1^j) = \left( \frac{A_2}{A_2 + A_3} C_2^i + \frac{A_3}{A_2 + A_3} C_3^i, \frac{A_2}{A_2 + A_3} C_2^j + \frac{A_3}{A_2 + A_3} C_3^j \right). \quad (78)$$

Once the centroids for the individual elements are calculated, the centroid for the composite shape (i.e. deformed biofilm structure) can be determined by applying a



similar form of equation (78). A sample of this APDL code is presented in Appendix G; a sample calculation of a quadrangle centroid is included in Appendix H.

#### Parametric Study of FSI Relationship to Burger Coefficients

A series of simulations was created to investigate the relationship between the material model coefficients and the fluid-structure interaction. Specifically, this study was intended to provide insight into the initial stages of the rolling biofilm phenomenon and how the viscoelastic parameters may relate to this behavior. A set of parameters (set 5) was designated as a baseline response. Then a series of simulations was run by varying one parameter in this baseline set. These permutations of the baseline set were designated as set 1, 2, 3, or 4 if the magnitude of the varied parameter was decreased. Similarly, if the magnitude of the varied parameter increased, the set was designated as 6 through 9. Each series was comprised of 9 parameter sets, one baseline and 8 permutations of a specific parameter. In all, 8 different parameters were studied, each consisting of nine different sets. This necessitated generating 72 different simulations of the biofilm model.

The goal of the parametric study was to examine the effect of material coefficients on biofilm deformation. The numerical strain could not be known *a priori* so an analogy was made to the one dimensional analytical solution to determine an appropriate range of Burger parameters to investigate. The parameter values were based on the theoretical creep response of the one dimensional burger model subjected to a 0.1 Pascal loading over thirty seconds. These values were on the same order of magnitude as

the experimental evidence provided by Stoodley et al. (1999b). The parameter variation within a series was designed to produce a wide range of deformations. In each series, a single parameter (e.g. Kelvin shear modulus) was varied so that the ratio of the final strain to the final strain in the baseline (set 5) was one of the following values: 5.0, 4.0, 3.0, 2.0, 0.8, 0.6, 0.4, and 0.2. A sample input file for series 0, set 5 is presented in Appendix G.

Series 0: Maxwell Elastic Modulus Varied. In Series 0, the value of the Maxwell element elastic modulus,  $G_1$ , was varied. The relaxation time for the Maxwell element,  $\tau_1$ , was held constant. Accordingly, the Maxwell viscous coefficient,  $\eta_1$ , changed in response to  $G_1$ . Table 9 lists the parameter sets investigated in Series 0.

Table 9. Series 0 Parametric Study.

|                          | 01   | 02   | 03   | 04   | 05   | 06    | 07    | 08    | 09     | Unit                 |
|--------------------------|------|------|------|------|------|-------|-------|-------|--------|----------------------|
| $G_1$                    | 1.77 | 2.22 | 3.00 | 4.62 | 10   | 13.04 | 18.74 | 33.31 | 149.26 | kg/mm-s <sup>2</sup> |
| $\eta_1$                 | 26.5 | 33.3 | 45.0 | 69.2 | 150  | 195.6 | 281.2 | 499.6 | 2238.9 | kg/mm-s              |
| $\tau_1$                 | 15   | 15   | 15   | 15   | 15   | 15    | 15    | 15    | 15     | s                    |
| $G_2$                    | 20   | 20   | 20   | 20   | 20   | 20    | 20    | 20    | 20     | kg/mm-s <sup>2</sup> |
| $\eta_2$                 | 100  | 100  | 100  | 100  | 100  | 100   | 100   | 100   | 100    | kg/mm-s              |
| $\tau_2$                 | 5    | 5    | 5    | 5    | 5    | 5     | 5     | 5     | 5      | s                    |
| $\epsilon/\epsilon_{05}$ | 5.00 | 4.00 | 3.00 | 2.00 | 1.00 | 0.80  | 0.60  | 0.40  | 0.20   | mm/mm                |

Seven of the 9 simulations converged successfully. The average time to solution was approximately 4.2 hours for each simulation. Set numbers 01 and 02 represented biofilm structures with the lowest Maxwell elastic modulus. Consequently, the deformations observed in these two simulations were the greatest. Excessive deformations in a FEM model led to convergence failure. Specifically, elements

downstream of the biofilm compressed as the grid was adjusted to account for displacement of the structural elements. The deformations in set 01 and 02 were large enough to compress down stream elements to the point where the element volumes became zero or negative. This resulted in convergence failure. Sets 03 through 09 converged successfully.

Fluid-induced pressure and shear forces acted to deform the biofilm structure. In reaction to these applied loads, internal stresses developed in the biofilm. A sample contour plot of the internal von Mises stresses in the biofilm during the initial state of simulation 05 is presented in Figure 40. The final stress state (i.e  $t = 42$  seconds) is presented in Figure 41.

A maximum stress of 1.19 mN was observed at the upstream base of the biofilm structure. Sample vector plots of the initial and final local fluid velocity fields are presented in Figures 42 and 43, respectively.

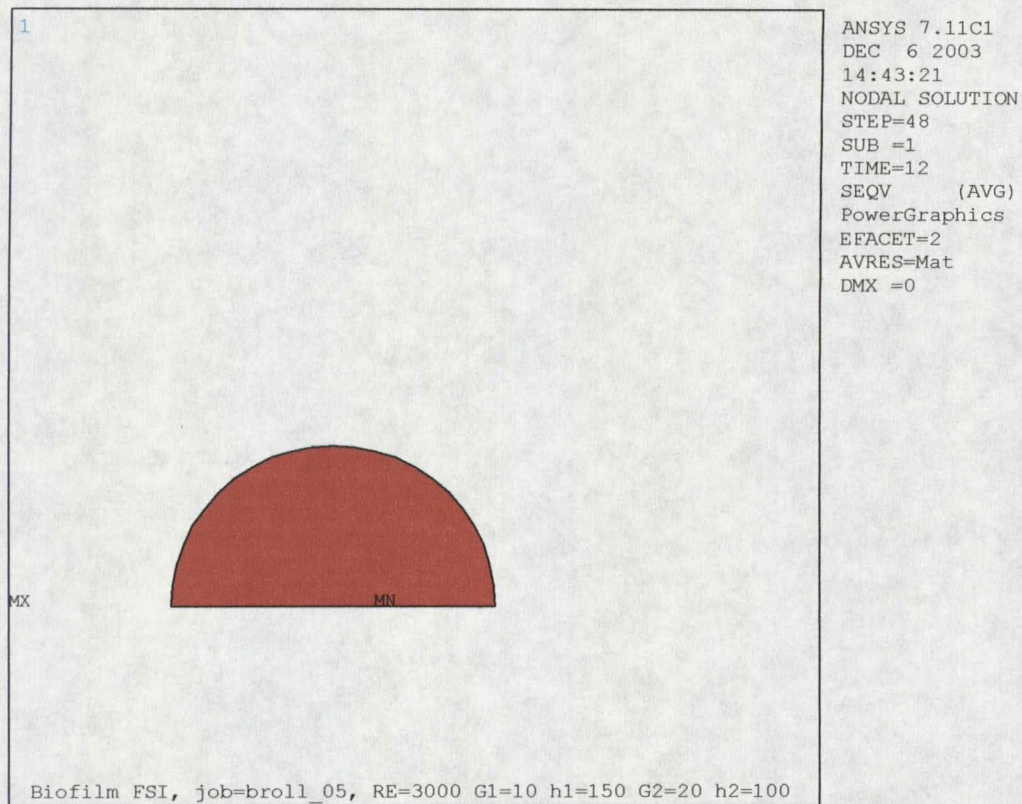


Figure 39. Initial Geometry and Stress State of Simulation 05. At  $t = 12$  seconds, the material is not loaded and unstressed.



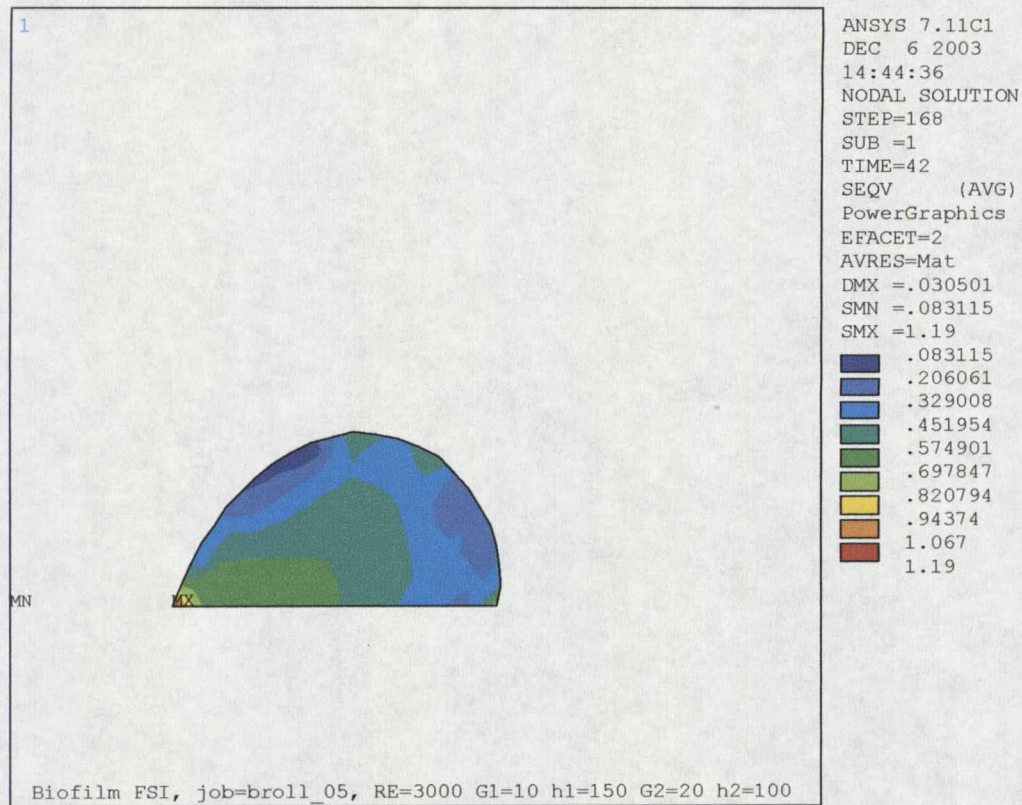


Figure 40. Final Geometry and Stress State of Simulation 05. At  $t = 42$  seconds, the material has been loaded for 30 seconds. Contours represent internal von Mises stress distributions.



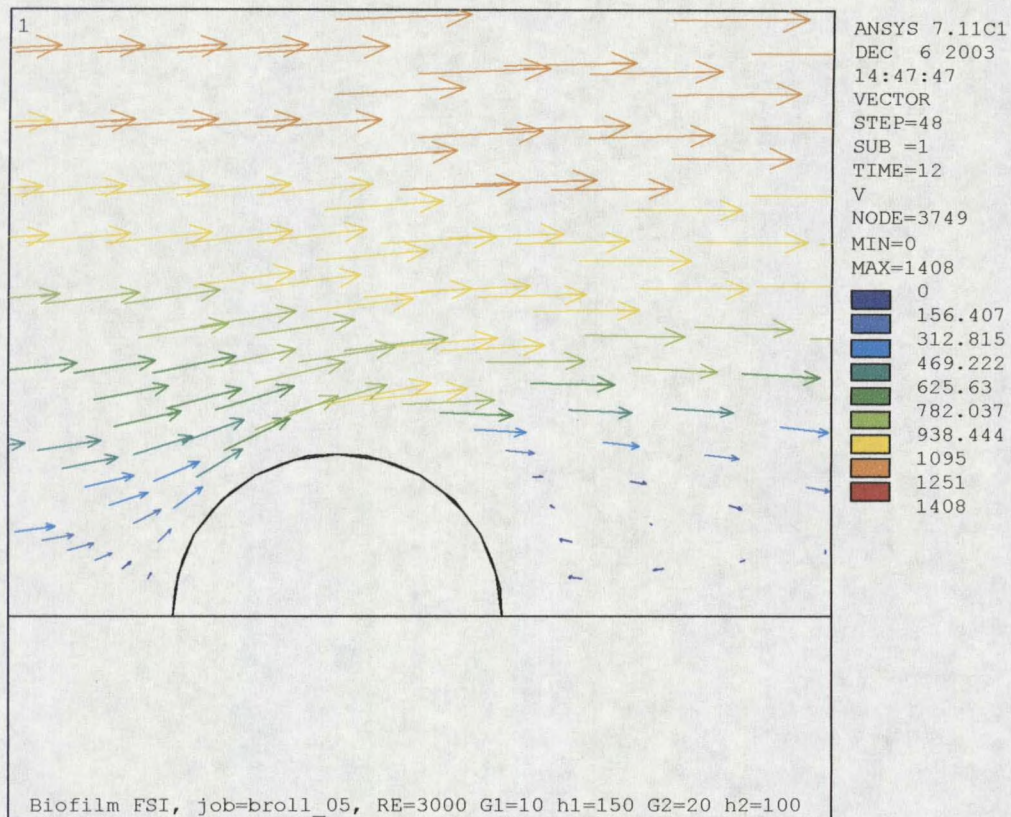


Figure 41. Initial ( $t = 12$  seconds) Velocity Field for Simulation 05. Mean channel velocity was 1000 mm/s.



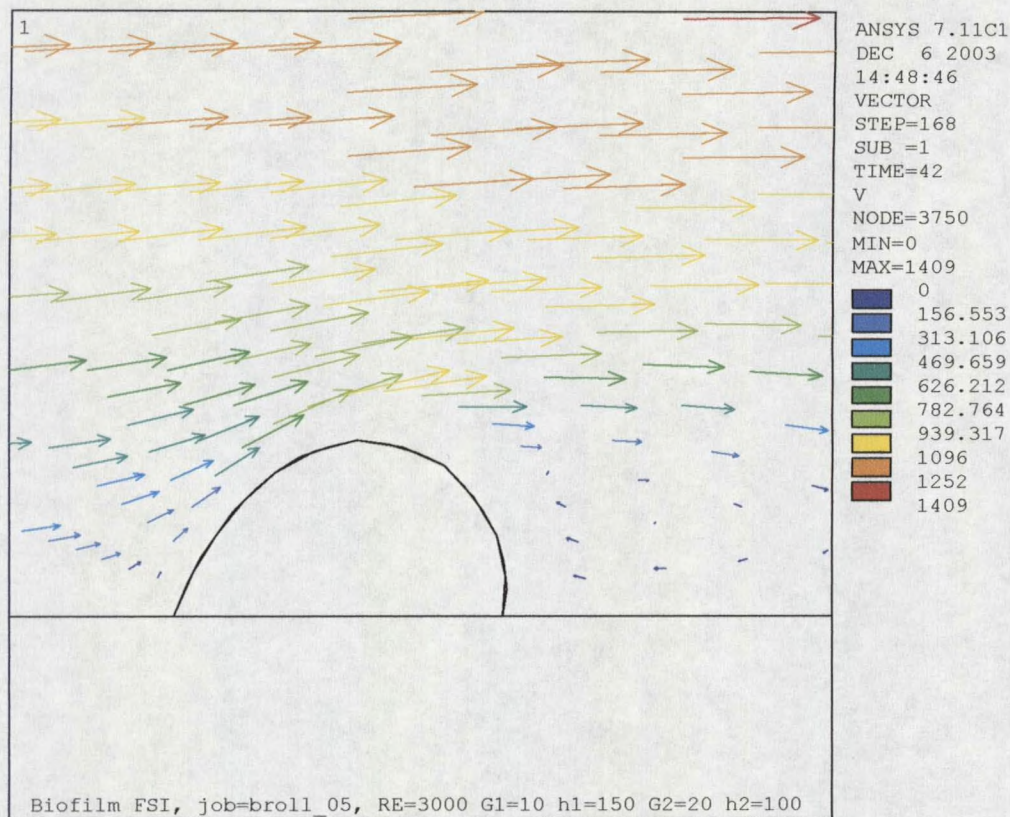


Figure 42. Final ( $t = 42$  seconds) velocity field for simulation 05. Mean channel velocity was 1000 mm/s.

The shape of the viscoelastic biofilm changed with time in response to the pressure and shear forces. In each simulation in Series 0, the centroid of the structure, as determined by equation (78), was used as a measure of this change. Overall the fluid forces deformed the biofilms in the positive  $x$  and  $y$  directions. Thus, the biofilms were lifted and pushed downstream. Figure 44 shows the change in the  $x$  coordinate of the biofilm centroids with time from their initial positions of  $x = 20$  mm. Figure 45 illustrates the change in the  $y$  component of the centroids from their initial position of  $y = 0.0845$  mm.



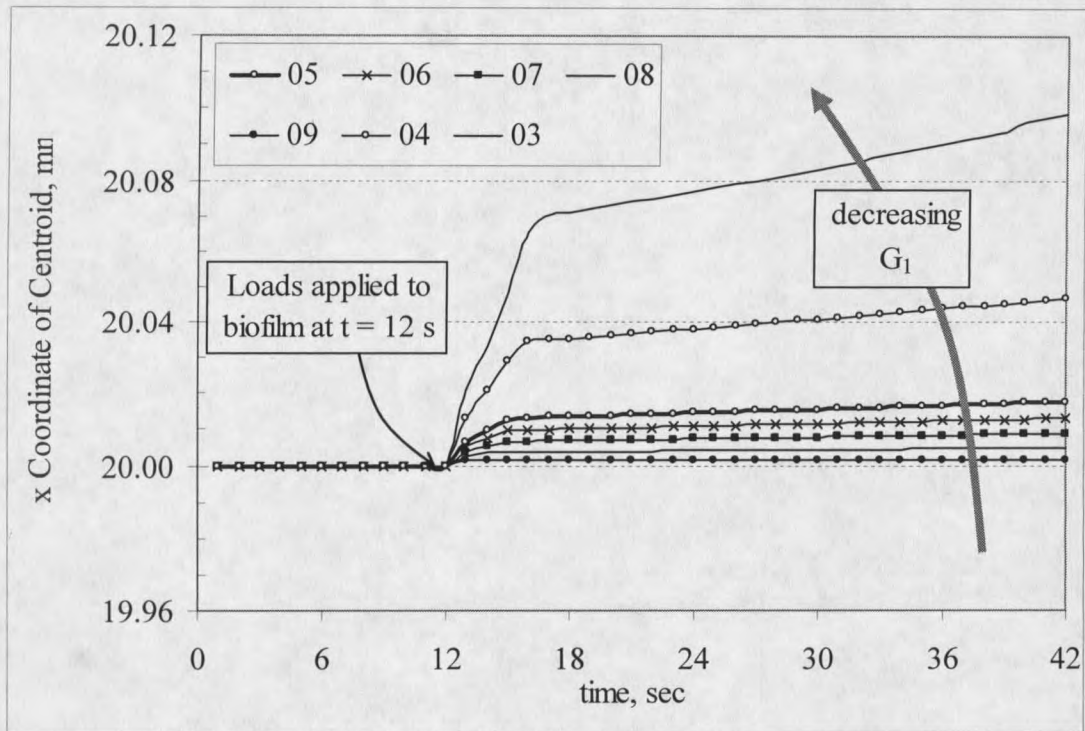


Figure 43. Change in x Component of Centroid with Time, Series 0.

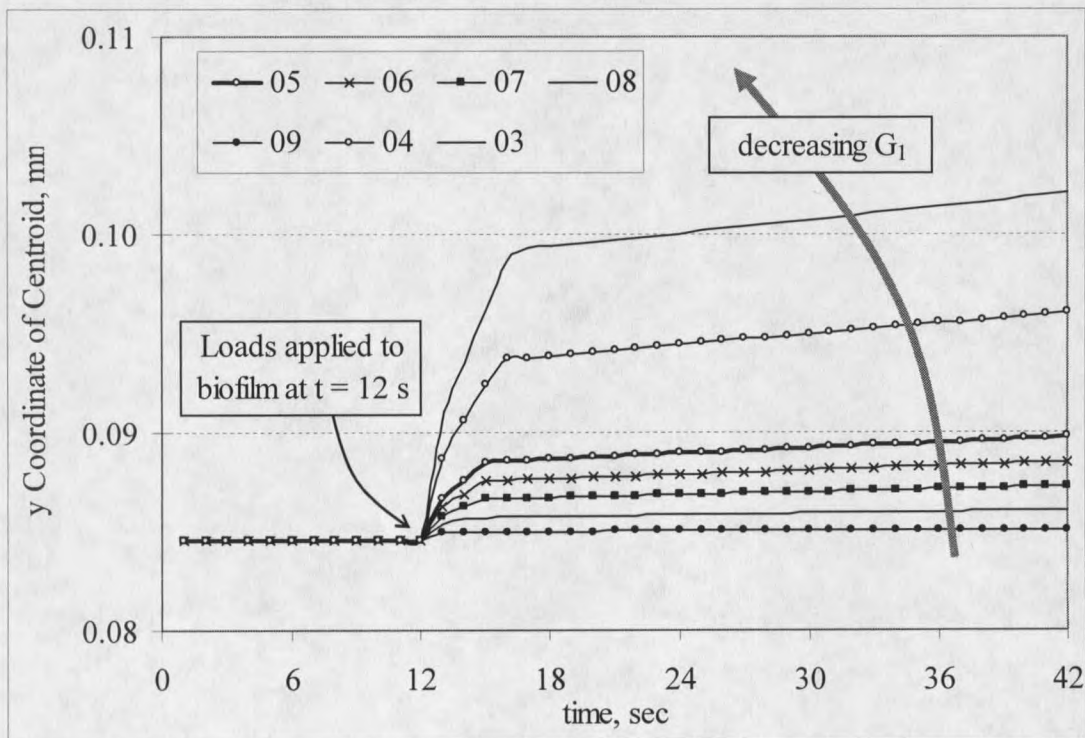


Figure 44. Change in y Component of Centroid with Time, Series 0.



An explanation of the centroid displacement can be found by examining the pressure and shear forces acting on the surface of the structure. Pressure forces act normally to the surface of the biofilm. Shear forces act tangentially. Shear and pressure can be resolved into a single resultant force,  $R$ , with components in both the  $x$  and  $y$  directions. This resultant was calculated for each converged simulation in Series 0. Figure 47 shows the  $x$  component of the resultant force,  $R_x$ , as it changes in time. Figure 48 shows the  $y$  component of the resultant,  $R_y$ , through time. Finally, Figure 49 illustrates the change through time in the magnitude of the resultant force.

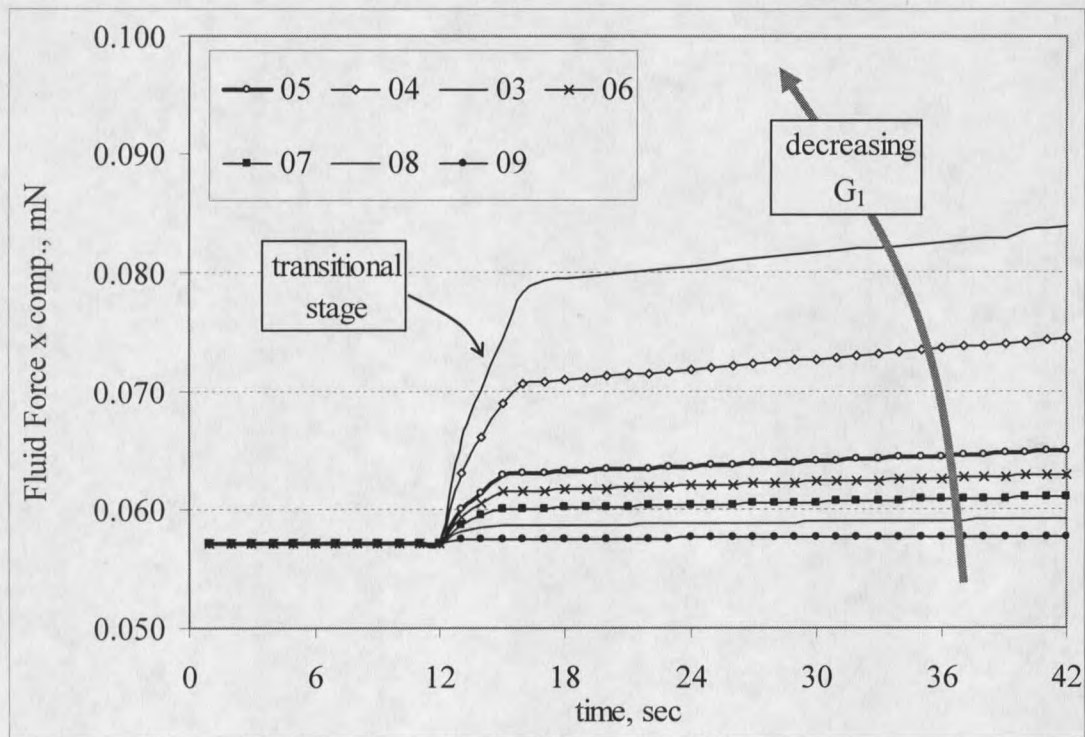


Figure 45. Change in the  $x$  Component of Fluid Forces,  $R_x$ , Through Time, Series 0.

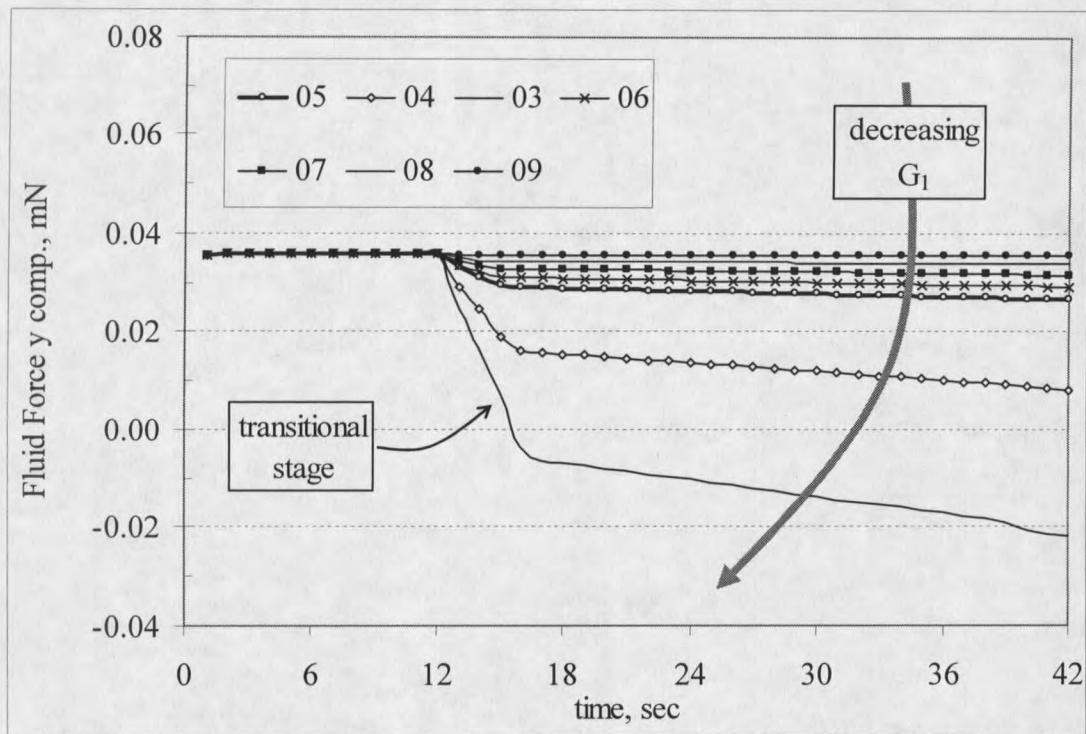


Figure 46. Change in the y Component of Fluid Forces,  $R_y$ , Through Time, Series 0.

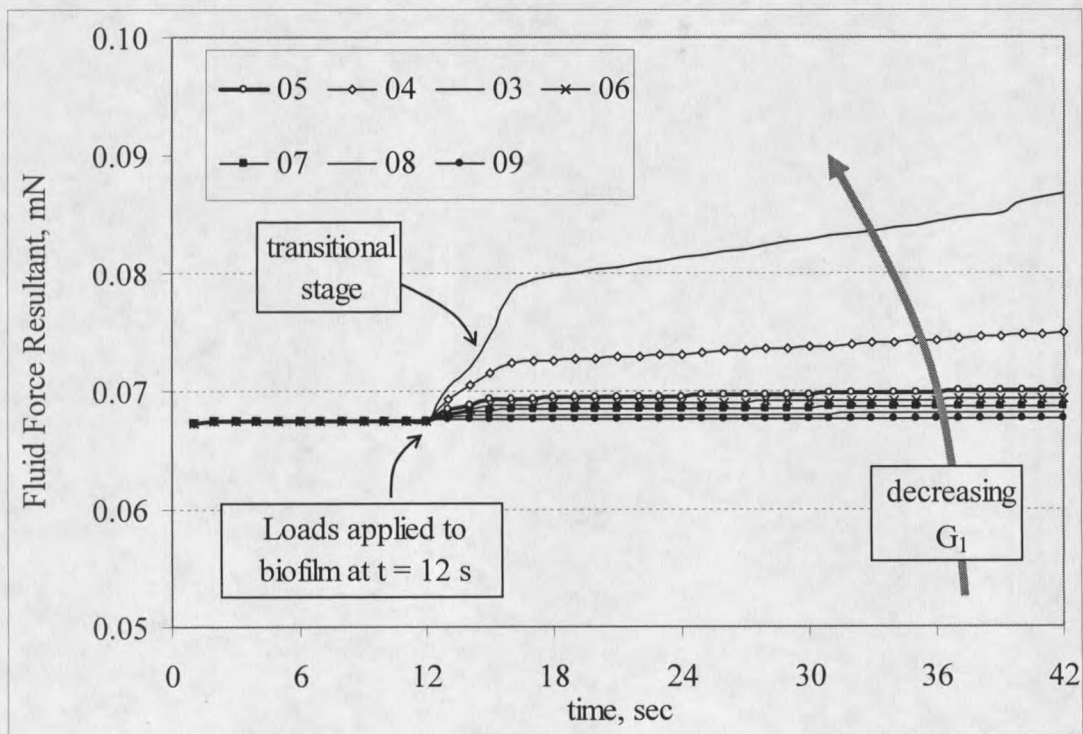


Figure 47. Change in the Resultant of Fluid Forces,  $R$ , Through Time, Series 0.

Not surprisingly, the  $x$  component of the resultant was positive throughout each simulation and became increasingly more positive in time. This positive  $x$  force acted to push the biofilm in the downstream direction. Less surprising, however, was that  $R_x$  increased as the Maxwell element elastic modulus was decreased exposing more surface area to fluid shear interactions. Thus, a softer biofilm was subject to a greater horizontal fluid force. Conversely, the  $y$  component of the resultant fluid forces decreased with time. Furthermore,  $R_y$  also decreased as  $G_1$  decreased. Consequently, the interplay between  $G_1$  and  $R$  was clear. The fluid forces tended to lift the biofilm, and in doing so, exposed the structure to the higher velocities found in the upper parts of the flow field. This effect was magnified by decreasing the Maxwell elastic modulus. The combined effect was that pressure and shear deformed the structure in such a way as to make it more susceptible to those same forces.

Interestingly, a clear change in the rate of increase in fluid forces was evident at approximately 17 seconds into the simulation (5 seconds into the deformation history). This "transitional stage" from  $t = 12$  seconds to  $t = 17$  seconds was characterized by a rapid increase in applied fluid forces. After  $t = 17$  seconds the rate of change in total pressure was much smaller. While the change in  $R$  with respect to time varied with  $G_1$ , they were approximately constant during and after this transition stage within a single simulation. It is likely that this change in the slope of  $R(t)$  was an expression of the relaxation time,  $\tau_2$ , of the Kelvin element. Relaxation times can be regarded as the time required for a viscoelastic material to dissipate internal stresses after being subjected to a constant strain instantaneously applied at  $t = 0$  seconds. As outlined in Chapter 3, the

Burger model possesses two different relaxation times, one for the Maxwell element and another for the Kelvin element. The Kelvin relaxation time for each simulation in Series 0 was 5 seconds. This is approximately equal to the elapsed time between the start of the simulation and the transition in slopes of  $R(t)$ .

Series 1: Kelvin Elastic Modulus Varied. In Series 1, the value of the Kelvin element elastic modulus,  $G_2$ , was varied. The relaxation time for the Kelvin element,  $\tau_2$ , was held constant. Accordingly, the Kelvin viscous coefficient,  $\eta_2$ , changed in response to  $G_2$ . Table 10 lists the parameter sets investigated in Series 1.

Table 10. Series 1 Parametric Study.

|                                | <u>11</u> | <u>12</u> | <u>13</u> | <u>14</u> | <u>15</u> | <u>16</u> | <u>17</u> | <u>18</u> | <u>19</u> | <u>Unit</u>          |
|--------------------------------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|----------------------|
| $G_1$                          | 10        | 10        | 10        | 10        | 10        | 10        | 10        | 10        | 10        | kg/mm-s <sup>2</sup> |
| $\eta_1$                       | 150       | 150       | 150       | 150       | 150       | 150       | 150       | 150       | 150       | kg/mm-s              |
| $\tau_1$                       | 15        | 15        | 15        | 15        | 15        | 15        | 15        | 15        | 15        | s                    |
| $G_2$                          | 0.69      | 0.91      | 1.33      | 2.50      | 20        | 24.25     | 30.80     | 42.20     | 67.1      | kg/mm-s <sup>2</sup> |
| $\eta_2$                       | 3.44      | 4.54      | 6.65      | 12.48     | 100       | 121.27    | 154.02    | 211.02    | 267.32    | kg/mm-s              |
| $\tau_2$                       | 5         | 5         | 5         | 5         | 5         | 5         | 5         | 5         | 5         | s                    |
| $\varepsilon/\varepsilon_{05}$ | 5.00      | 4.00      | 3.00      | 2.00      | 1.00      | 0.975     | 0.95      | 0.925     | 0.90      | mm/mm                |

All nine simulations in this series converged successfully. The average time to solution in Series 1 was 4.1 hours. This time was comparable to Series 0. In general, displacements were not as great in Series 1 as they were in Series 0. Displacements in sets 11 through 19 were never great enough to cause element death (i.e. negative element volumes). This explains why all simulations in the series converged successfully.

As with Series 0, the shape of the viscoelastic biofilm changed with time in response to the pressure and shear forces. Again, the centroid of the structure was used as a measure of this change. The magnitudes of the displacement were not as great in this series as they were in Series 0. A transition of displacement rates was also evident. In Series 1, these transitions to a new displacement rate occurred within a couple of seconds. Furthermore, this change occurred faster as the Kelvin elastic modulus was decreased. However, the ultimate displacement rate was relatively insensitive to  $G_2$ . This behavior agreed with the theoretical response of a Burger model which, at  $t = \infty$ , acts as a fluid governed by the Maxwell dashpot.  $\eta_2$  was held constant in Series 1 thereby explaining why the ultimate strain rates for all 9 simulations were approximately equal. Figures 50 and 51 show the change in the x and y coordinates of the centroids with time.

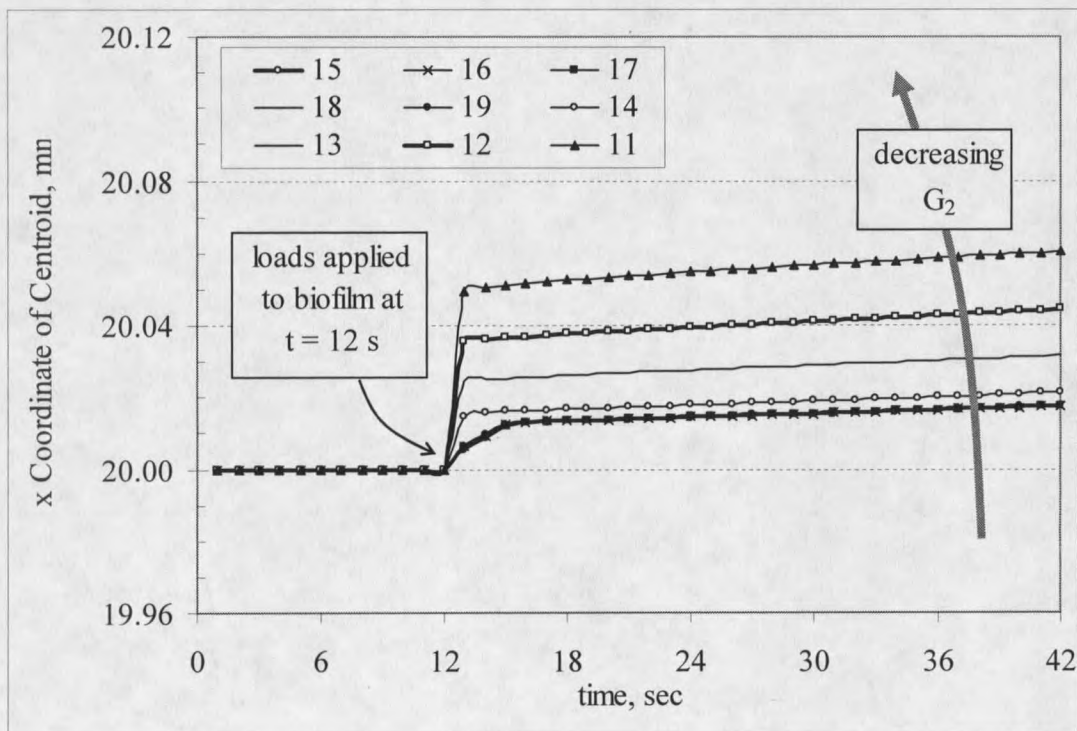


Figure 48. Change in x Component of Centroid with Time, Series 1.



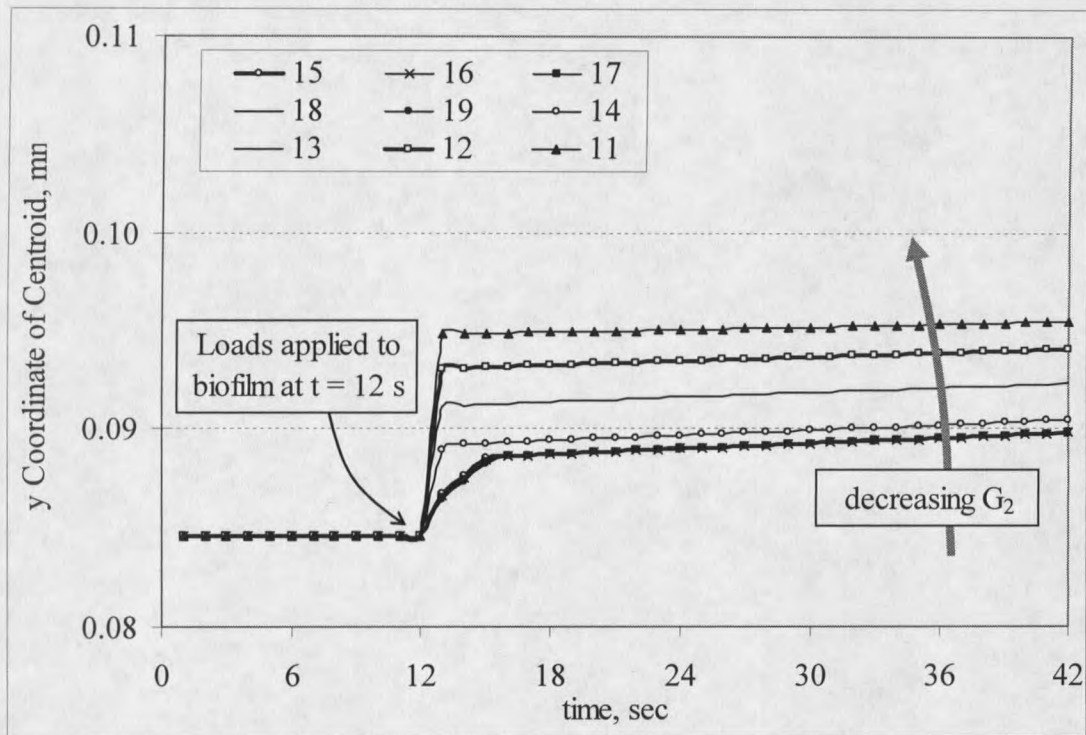


Figure 49. Change in y Component of Centroid with Time, Series 1.

The resultant fluid force and its components varied with time and  $G_2$ . As illustrated in Figures 52 and 53, the variation was not as pronounced as it was in Series 1. In fact, the system approached a near steady state almost immediately after the transition stage. Furthermore, the magnitudes of the resultant were smaller in Series 1 compared with Series 0 (Fig. 54). As explained previously, the transition of displacement rates was likely an expression of the Kelvin relaxation time and a metric of the time required to dissipate internal stresses in the biofilm. However,  $\tau_2$  was no longer distinguishable on the time axis of these plots. It was clear, however, that once the internal stresses in the biofilm were dissipated the resultant forces changed at approximately the same rate in

each simulation. This also agreed with the theoretical response of a Burger material at  $t = \infty$ .

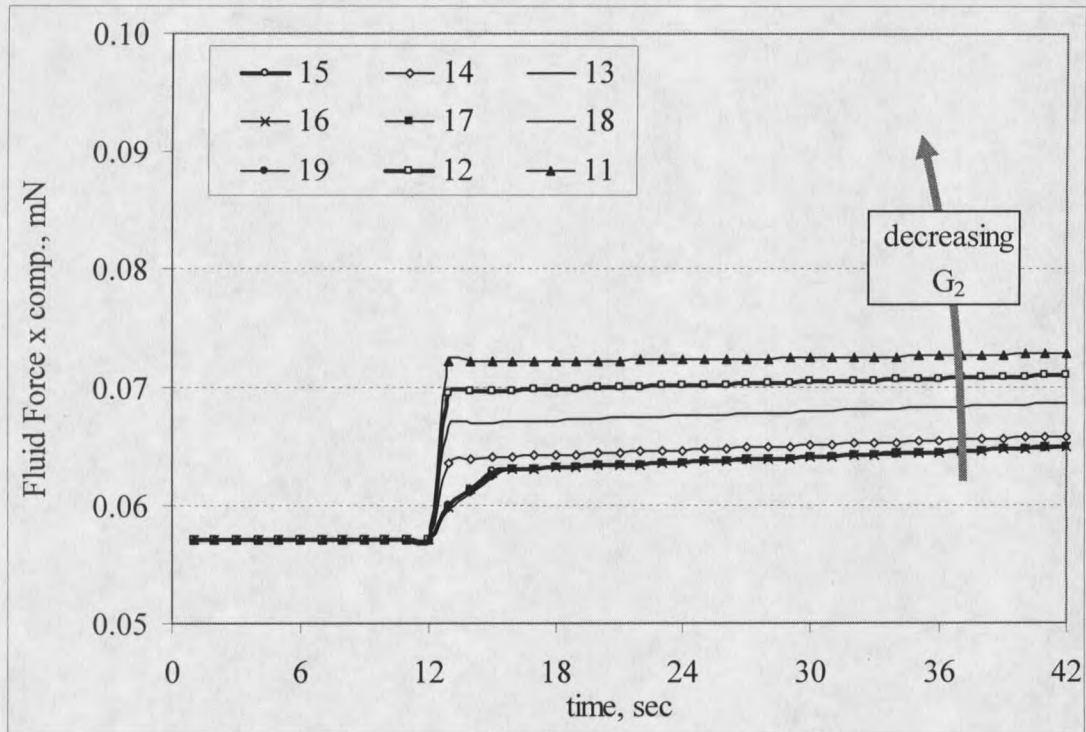


Figure 50. Change in the x Component of Fluid Forces,  $R_x$ , Through Time, Series 1.

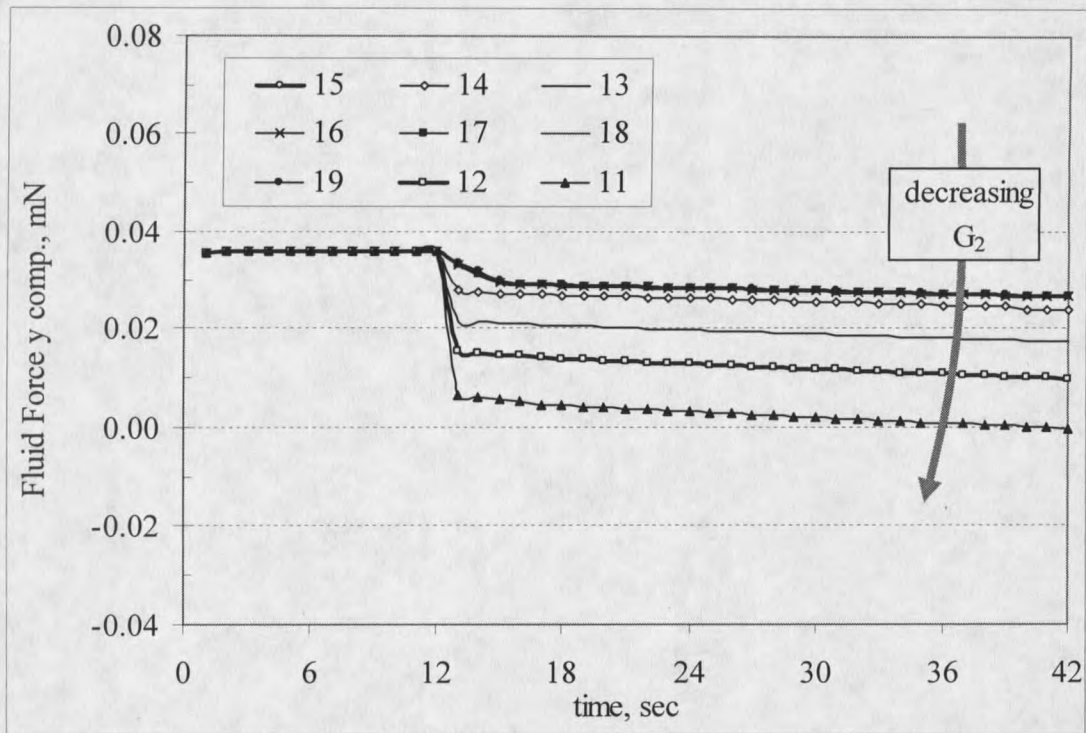


Figure 51. Change in the y Component of Fluid Forces,  $R_y$ , Through Time, Series 1.

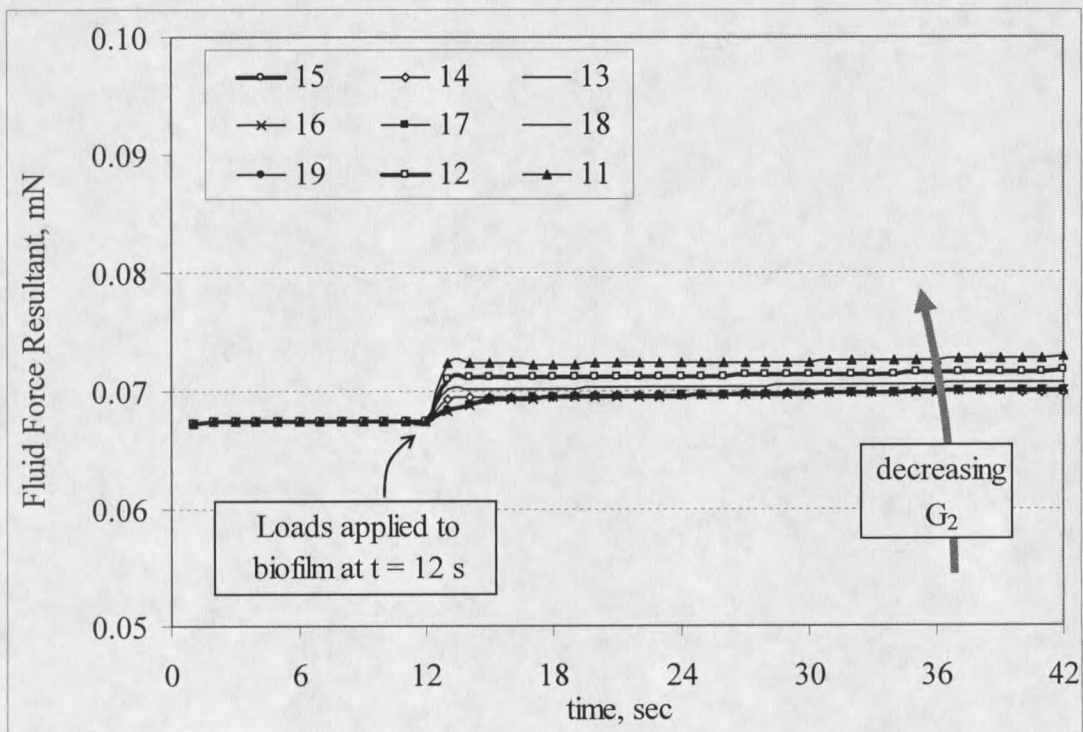


Figure 52. Change in the Resultant of Fluid Forces,  $R$ , Through Time, Series 1.



Series 2: Maxwell Elastic Modulus Varied. In Series 2, the value of the Maxwell element elastic modulus,  $G_1$ , was varied again. In contrast to Series 0, the viscous coefficient,  $\eta_1$ , was held constant. Consequently, the Maxwell relaxation time,  $\tau_1$ , changed in response to  $G_1$ . Table 11 lists the parameter sets investigated in Series 2.

Table 11. Series 2 Parametric Study.

|                                | <u>21</u> | <u>22</u> | <u>23</u> | <u>24</u> | <u>25</u> | <u>26</u> | <u>27</u> | <u>28</u> | <u>29</u> | <u>Unit</u>          |
|--------------------------------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|----------------------|
| $G_1$                          | 0.67      | 0.87      | 1.25      | 2.22      | 10        | 12.12     | 15.38     | 21.04     | 33.31     | kg/mm-s <sup>2</sup> |
| $\eta_1$                       | 150       | 150       | 150       | 150       | 150       | 150       | 150       | 150       | 150       | kg/mm-s              |
| $\tau_1$                       | 224.9     | 172.4     | 119.9     | 67.48     | 15        | 12.38     | 9.75      | 7.13      | 4.50      | s                    |
| $G_2$                          | 20        | 20        | 20        | 20        | 20        | 20        | 20        | 20        | 20        | kg/mm-s <sup>2</sup> |
| $\eta_2$                       | 100       | 100       | 100       | 100       | 100       | 100       | 100       | 100       | 100       | kg/mm-s              |
| $\tau_2$                       | 5         | 5         | 5         | 5         | 5         | 5         | 5         | 5         | 5         | s                    |
| $\varepsilon/\varepsilon_{05}$ | 5.00      | 4.00      | 3.00      | 2.00      | 1.00      | 0.95      | 0.90      | 0.85      | 0.80      | mm/mm                |

Series 2 produced results that were similar to Series 0 and 1. Displacements were not large enough to result in negative element volumes and all 9 simulations converged. The average time to convergence was approximately 4.1 hours. Again, fluid forces tended to both drag and lift the attached biofilm structure.  $G_1$  was varied in both Series 0 and 2. However, by holding the viscous coefficient constant, the overall results of Series 2 more closely resembled those of Series 1, not Series 0. This suggests that the Maxwell viscous coefficient plays a dominant role in the ultimate displacement of the attached biofilm structure. Figure 55 shows the change in the resultant fluid forces for each simulation in Series 2.

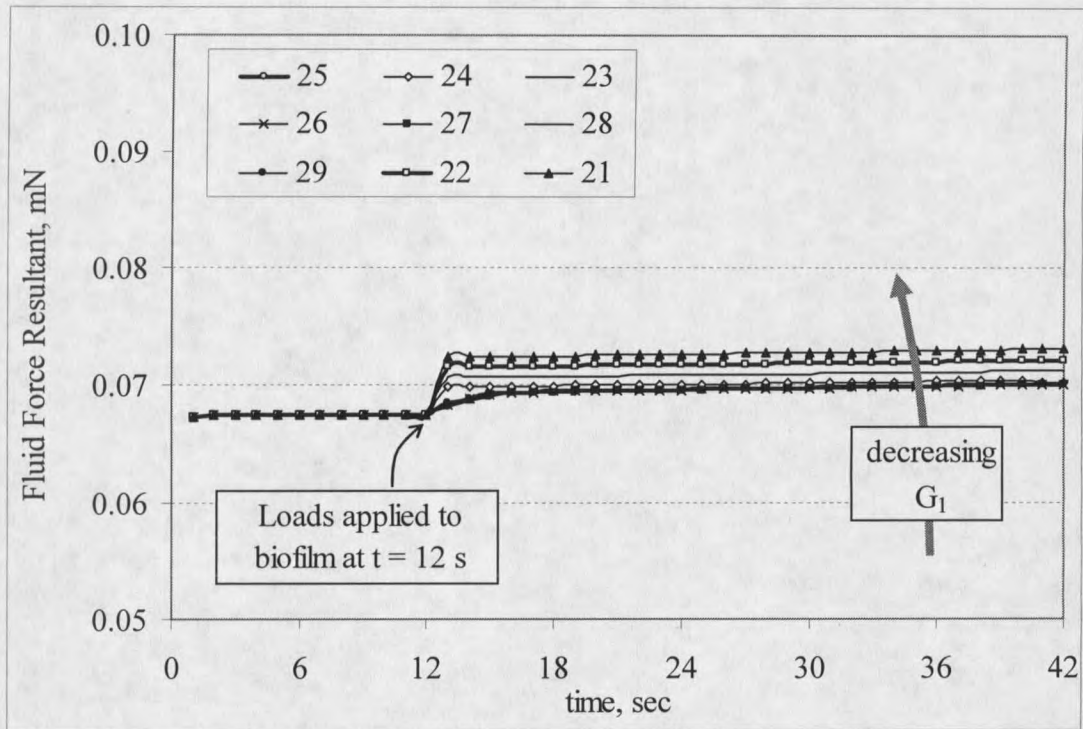


Figure 53. Change in the Resultant of Fluid Forces,  $R$ , Through Time, Series 2.

Series 3: Kelvin Elastic Modulus Varied. The value of the Kelvin element elastic modulus,  $G_2$ , was varied again. In contrast to Series 1, the viscous coefficient,  $\eta_2$ , was held constant. Consequently, the Kelvin relaxation time,  $\tau_2$ , changed in response to  $G_2$ .

Table 12 lists the parameter sets investigated in Series 3

Table 12. Series 3 Parametric Study.

|                                | <u>31</u> | <u>32</u> | <u>33</u> | <u>34</u> | <u>35</u> | <u>36</u> | <u>37</u> | <u>38</u> | <u>39</u> | Unit                 |
|--------------------------------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|----------------------|
| $G_1$                          | 10        | 10        | 10        | 10        | 10        | 10        | 10        | 10        | 10        | kg/mm-s <sup>2</sup> |
| $\eta_1$                       | 150       | 150       | 150       | 150       | 150       | 150       | 150       | 150       | 150       | kg/mm-s              |
| $\tau_1$                       | 15        | 15        | 15        | 15        | 15        | 15        | 15        | 15        | 15        | s                    |
| $G_2$                          | 0.98      | 2.64      | 5.03      | 9.14      | 20        | 24.30     | 30.88     | 42.31     | 67.17     | kg/mm-s <sup>2</sup> |
| $\eta_2$                       | 100       | 100       | 100       | 100       | 100       | 100       | 100       | 100       | 100       | kg/mm-s              |
| $\tau_2$                       | 101.7     | 37.93     | 19.88     | 10.94     | 5         | 4.12      | 3.24      | 2.36      | 1.49      | s                    |
| $\varepsilon/\varepsilon_{05}$ | 1.60      | 1.45      | 1.30      | 1.15      | 1.00      | 0.975     | 0.95      | 0.925     | 0.9       | mm/mm                |

The ratio of baseline strain (Set 35) to the strain produced in Set 31 for a theoretical one dimensional model was approximately 1.60 compared with 5.0 in the other series. Overall, the ultimate strains produced in this series were limited by the choice of baseline parameters in Set 35. Nevertheless, these baseline parameters were used in order to preserve consistency with the other series. Despite the limited displacements produced, the impact of changing  $G_2$  while holding the viscous coefficient constant was clear. As  $G_2$  decreased,  $t_2$  increased and this increase in the Kelvin relaxation time was expressed in the time required for the resultant fluid force (Fig 56) to reach a constant rate of increase.

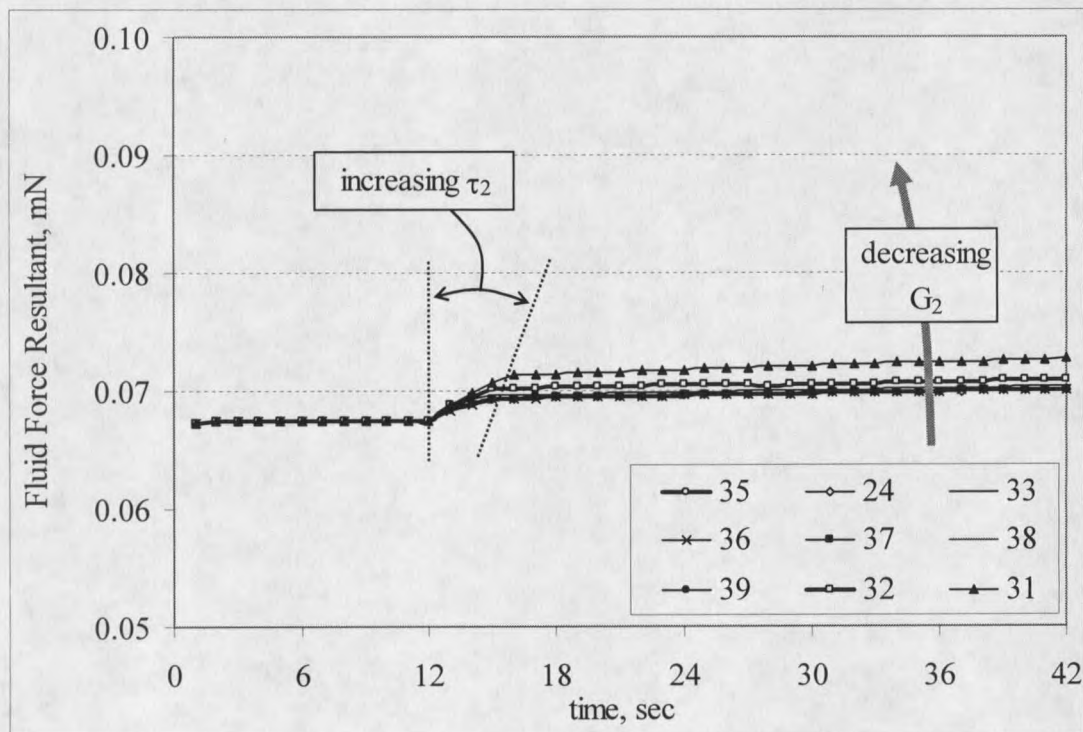


Figure 54. Change in the Resultant of Fluid Forces,  $R$ , Through Time, Series 3.

Series 4 and 5: Mean Channel Velocity Varied. In these two series, the magnitude of the mean stream velocity was altered. In Series 4, simulations 3, 5 and 7 were re-run using a channel velocity of 800 mm/s. In Series 5, simulations 4, 5, and 7 were re-run using a channel velocity of 1200 mm/s. Set 54 was used in lieu of Set 53 which produced displacement great enough to cause convergence failure. Figures 57 and 58 illustrate the effect of decreasing and increasing channel velocity on the resultant applied force on the biofilm.

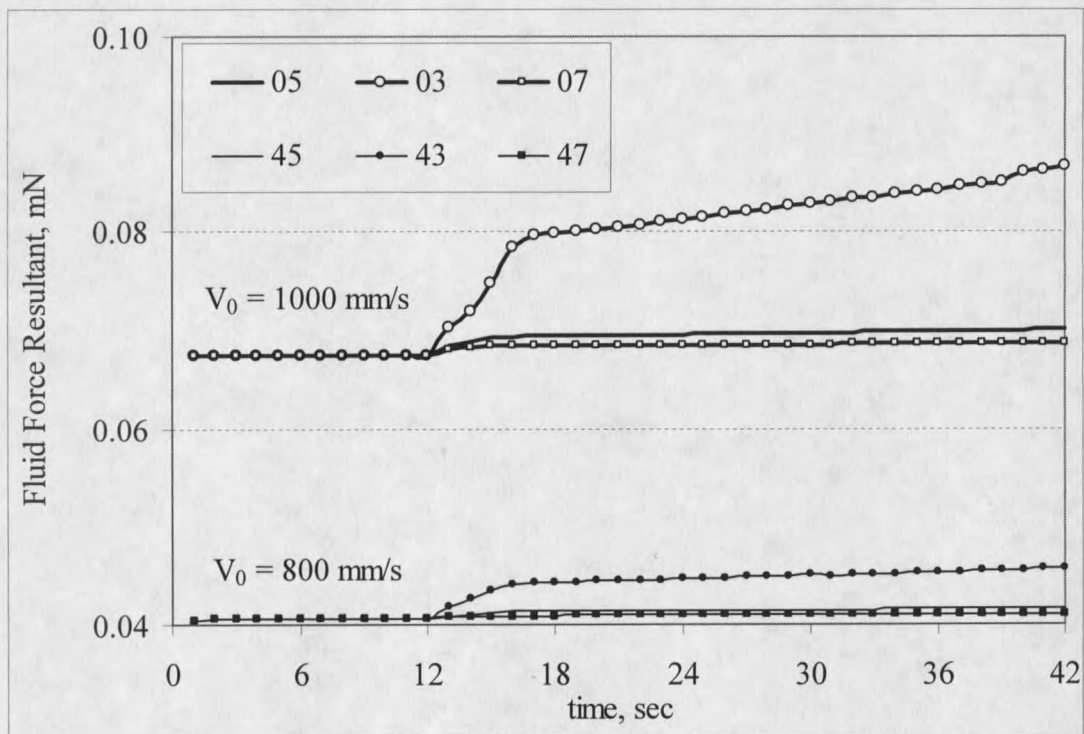


Figure 55. Change  $R$ , Through Time, for 2 Different Channel Velocities, Series 4.



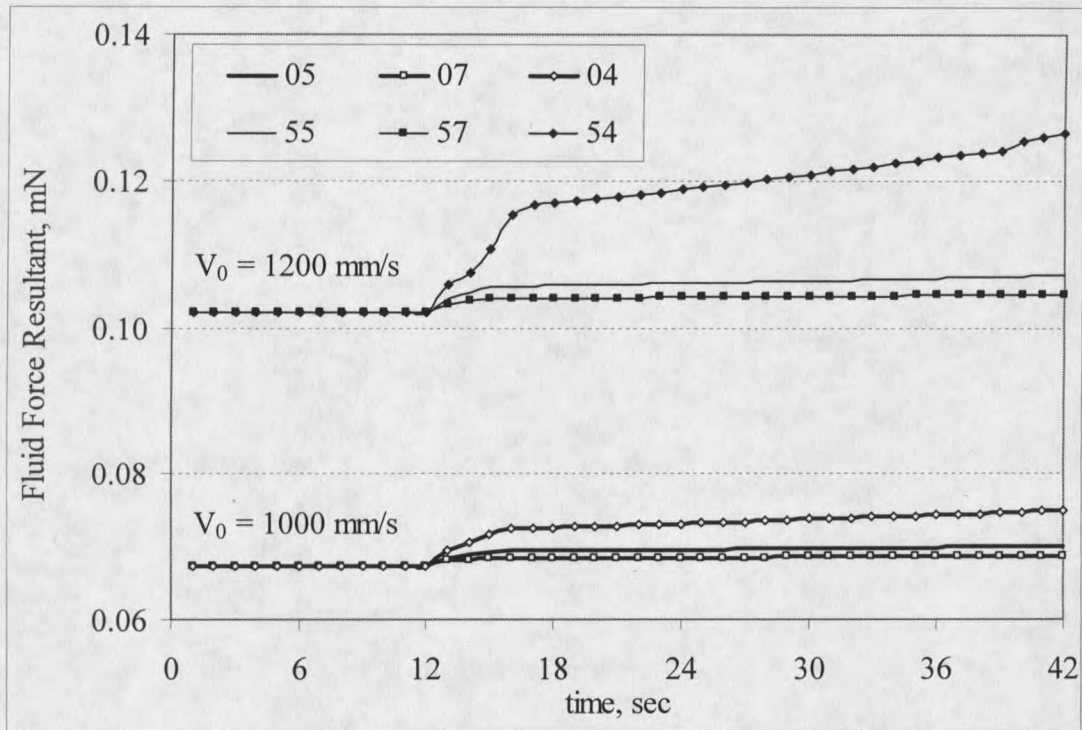


Figure 56. Change  $R$ , Through Time, for 2 Different Channel Velocities, Series 5.

A comparison of the channel velocity to the resultant fluid forces provided insight into biofilm boundary layer penetration. In Series 4, the ratio of channel velocity to the baseline velocity of Series 0 was 0.8. The initial ratio (up to  $t = 12$  seconds) of resultant forces for the two series was 0.601. In all three simulations in Series 4 (43, 45, 47) this force ratio decreased as the biofilm deformed over time. Softer biofilms (lower  $G_1$ ) experienced a greater decrease in the force ratio. Conversely, when comparing the ratios of the resultant in Series 5 to Series 0, it was found that the force ratio increased from 1.514 to 1.686 in the biofilm with the lowest Maxwell elastic modulus. To explain this interplay, it is convenient to resolve the fluid forces into lift forces (y-component forces) and drag forces (x component forces).

Lift causes vertical deformation of the biofilm. The vertical deformation leads to a higher profile in the velocity field. Assuming a uniform velocity distribution, the higher profile would still result in greater momentum transfer to the attached structure. However, the velocity distribution is not uniform in turbulent flow. The distribution is approximately linear within the boundary layer and approximately logarithmic beyond the boundary layer. Quantifying a relationship was not possible because the boundary layer thickness is a function of Reynolds number and resultant forces were a function of the complete strain history. However, in general, the velocity increases as one moves away from the channel boundary. Thus, the lift imposed on the biofilm is doubly important because it tends to create a greater vertical profile for the fluid to impinge upon and the resultant applied forces vary non-linearly because the velocity distribution is non-uniform in turbulent flow.

Series 6 and 7: Biofilm Hemisphere Radius Varied. In these two series, the effective surface area of the attached biofilm was altered by changing the radius of the biofilm hemisphere. In Series 6, the radius was decreased from 0.20 mm to 0.15 mm. In Series 7 the radius was increased from 0.20 mm to 0.25 mm. The nature of the results was similar to those in Series 4 and 5. By increasing the size of the attached structure, the momentum transfer to the biofilm was increased.

Series 8: Pure Elastic Response Examined. Series 8 consisted of a single simulation of Set 85. In this scenario, the biofilm was modeled as a purely elastic

material with a Young's modulus of  $10 \text{ kg/mm-s}^2$ . This value matched the magnitude of the Maxwell elastic modulus in the baseline Set 05.

### Drag and Lift

As described previously, the force a fluid imparts to a boundary can take one of two forms. Pressure forces are those that act normal to the surface and can be further categorized as hydrostatic or stagnation pressures. The total pressure refers to the sum of forces caused by the transfer of momentum normal to a boundary. Shear forces refer to the tractive forces that act tangential to a boundary surface. In a general two dimensional problem, pressure and shear forces will both have x and y components. When considering forces applied by flow around an object, it is often convenient to resolve the forces acting in the downstream direction into a single drag force. Thus, the drag force is comprised of a surface drag due to tractive forces and a form drag due to pressure differentials. Similarly, the fluid forces acting perpendicular to the downstream direction are resolved into a single lift force. In the context of this biofilm model, the fluid forces acting on the biofilm structure in the x direction were resolved into a single drag force. The forces acting in the y direction were resolved into a single lift force.

### Polar Diagrams

In the study of airfoils, drag and lift coefficients are determined by normalizing the drag force by the velocity head (Vennard, 1940). More specifically, the drag coefficient,  $C_D$ , is

$$C_D = \frac{F_D}{\left(\frac{\rho V^2}{2}\right)}, \quad (79)$$

where  $F_D$  is the drag force,  $\rho$  is the density of the fluid, and  $V$  is the mean fluid velocity.

The lift coefficient,  $C_L$ , is calculated in a similar manner and takes the form

$$C_L = \frac{F_L}{\left(\frac{\rho V^2}{2}\right)}, \quad (80)$$

where  $F_L$  is the total lift force acting on the airfoil. By extending this argument to the study of the hydrodynamic impact on attached biofilms, one can use drag and lift coefficients as non-dimensional metrics of the forces on a biofilm structure. Additionally, aeronautical engineers use polar diagrams as a means of relating lift versus the airfoil's angle of attack (Vennard, 1940). A polar diagram is a plot of the drag coefficient versus the lift coefficient. Polar diagrams were created using the results of Series 0 through Series 8 to investigate the relationship between drag and lift in a non-dimensional setting.

Figure 59 is a polar diagram of the drag versus lift coefficients in Series 0. All 7 simulations started with the same initial drag and lift coefficients of 0.114 and 0.072, respectively. In each simulation, the biofilm hemispheres deformed in response to these forces and, consequently, this altered the forces that were applied later in time. As time advanced, the drag forces increased while the lift forces decreased. While the ultimate



value of these coefficients differed among simulations, it is *very* interesting to note that the entire series plotted along the same curve. That is, the paths that each of these simulations took in reaching their final values of lift and drag were nearly identical. They differed only in how far along the path they progressed after 30 seconds. If the ratios of the lift and drag forces are indicators of deformed geometry, this also suggests that the deformation history was identical for each of these simulations.

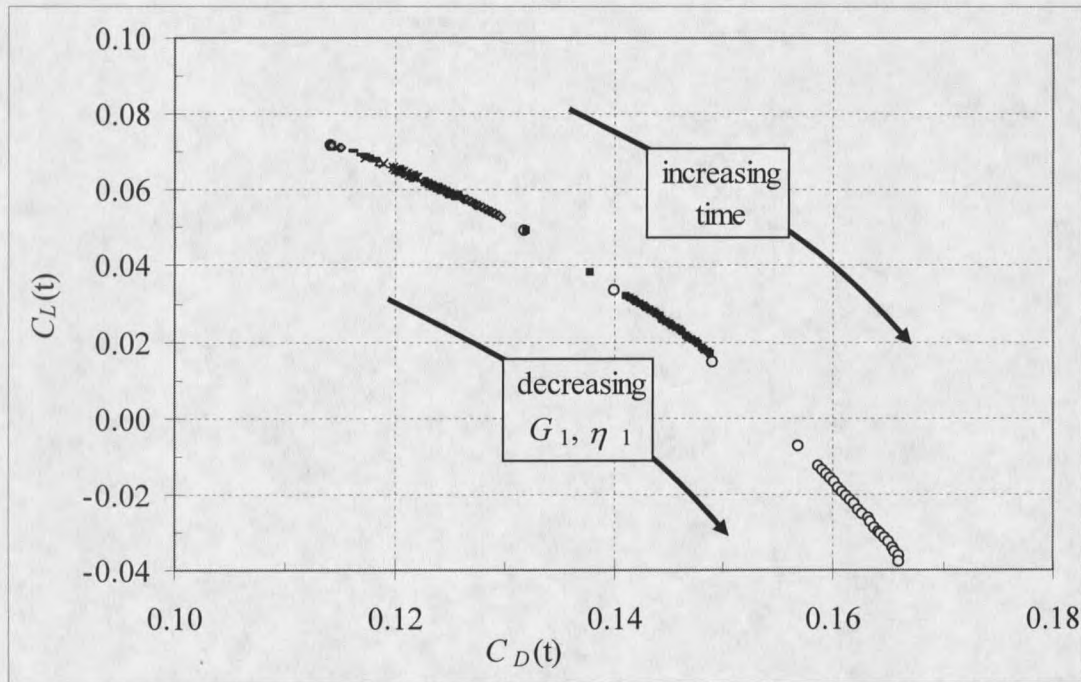


Figure 57. Polar Diagram of Drag ( $C_D$ ) Versus Lift Coefficients ( $C_L$ ), Series 0.

In Series 1, the Kelvin elastic modulus was varied while holding the Kelvin relaxation time constant. The coefficients of drag and lift for all 9 simulations in this series were plotted on a polar diagram. The results indicated a pattern similar to the one seen in Figure 59. This tendency to preserve a common deformation history among a

permutation set was also present in Series 2 and 3. These results are presented in Figures 60, 61 and 62.

These polar diagrams demonstrate that varying a single elastic modulus in the Burger model, while holding either the viscous coefficient or relaxation time constant, will produce a similar deformation history. The effect of varying the modulus is seen in how far into the deformation history (or how far down the path traced out on the polar diagrams) the simulation progress. Although each of these simulations produced a single deformation path and although each path originated at the same point, they did not produce the same path. When displayed on a common plot, a clear bifurcation in these curves was evident and is displayed in Figure 63. There was close agreement between Series 0 and 3 and between Series 1 and 2. The previous examination of Series 0 and 3 demonstrated (Fig. 45 through 49 and 56) that these two series clearly expressed two different modes of behavior or rates of deformation. Furthermore, it was suggested that these different slopes were an expression of viscoelastic relaxation times. It seems likely therefore that the bifurcation of these polar paths in Figure 63 is also due to differences in relaxation times.

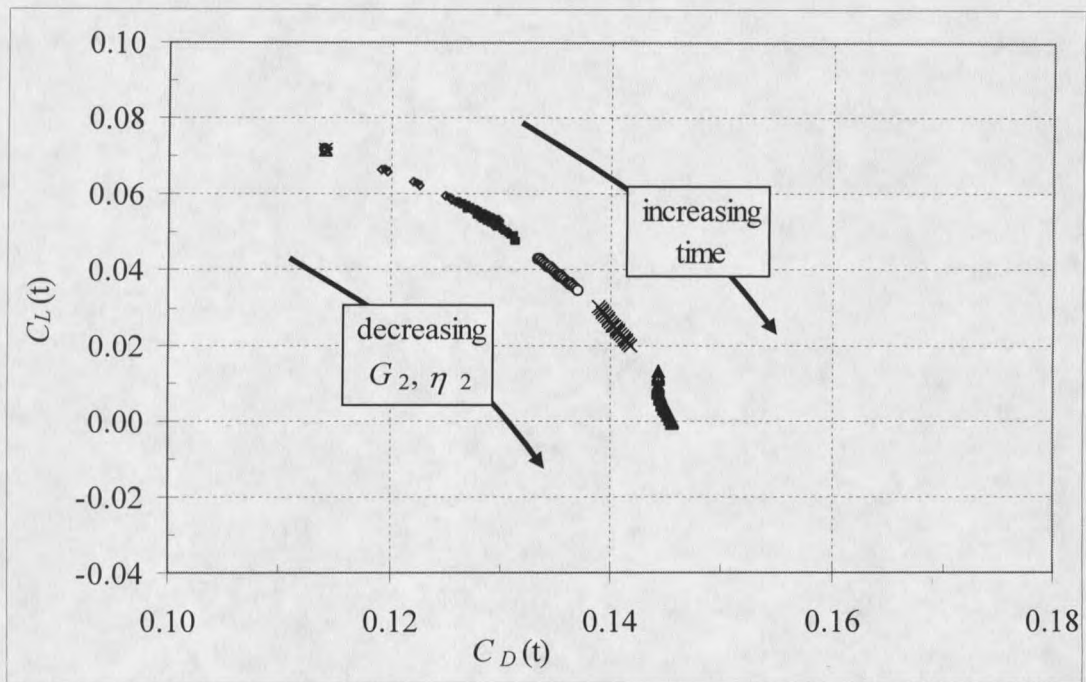


Figure 58. Polar Diagram of Drag ( $C_D$ ) Versus Lift Coefficients ( $C_L$ ), Series 1.

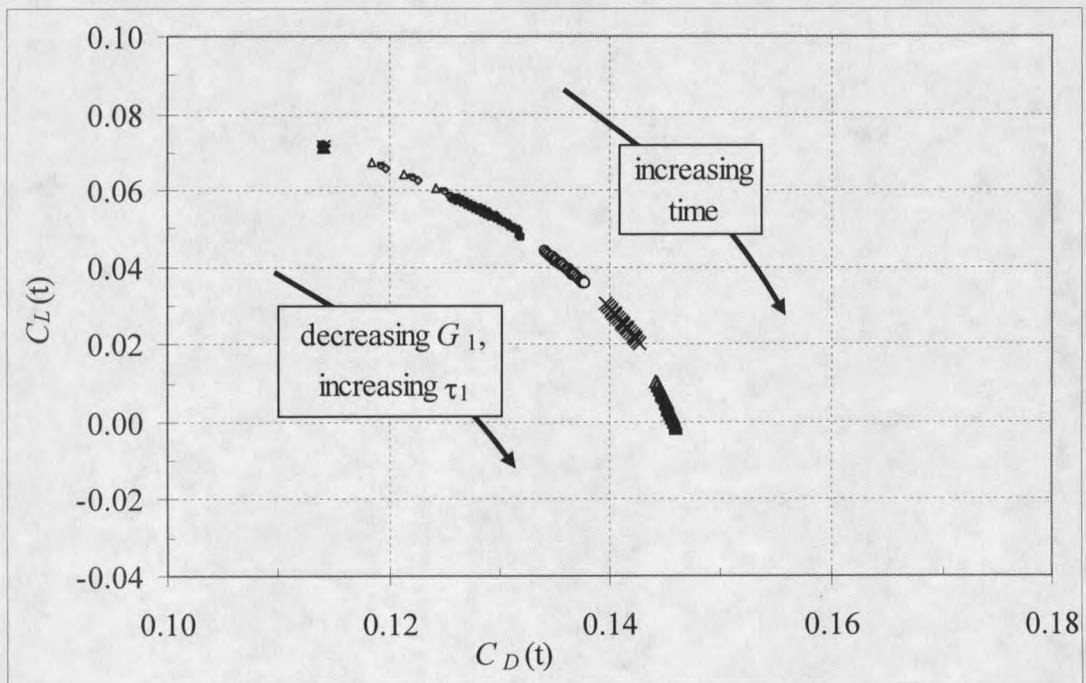


Figure 59. Polar Diagram of Drag ( $C_D$ ) Versus Lift Coefficients ( $C_L$ ), Series 2.

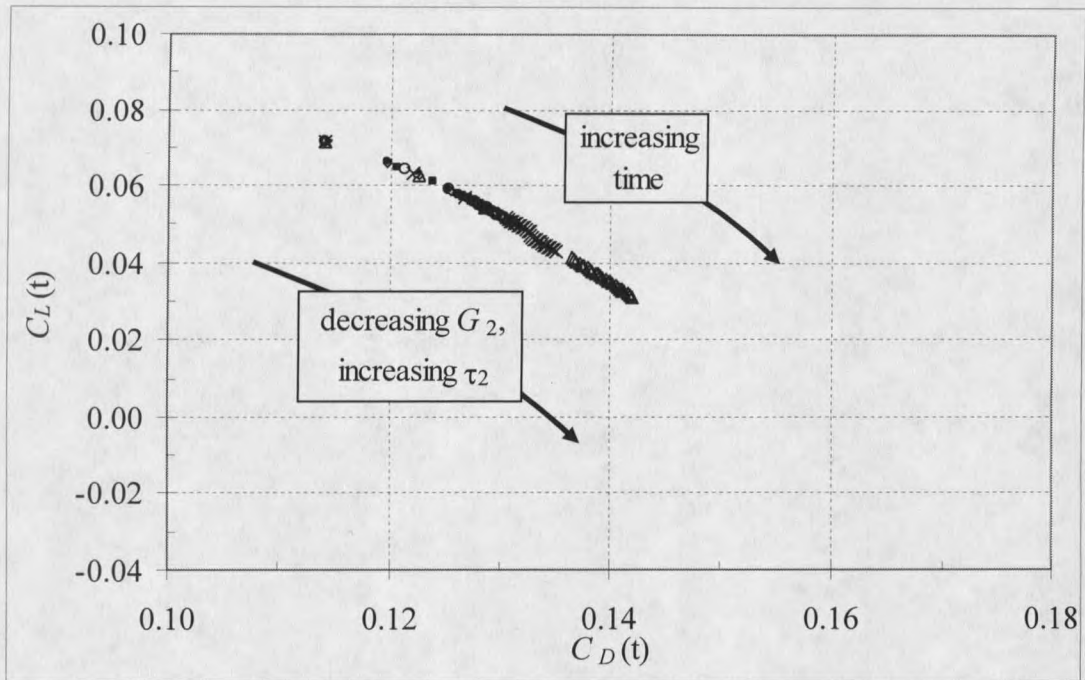


Figure 60. Polar Diagram of Drag ( $C_D$ ) Versus Lift Coefficients ( $C_L$ ), Series 3.

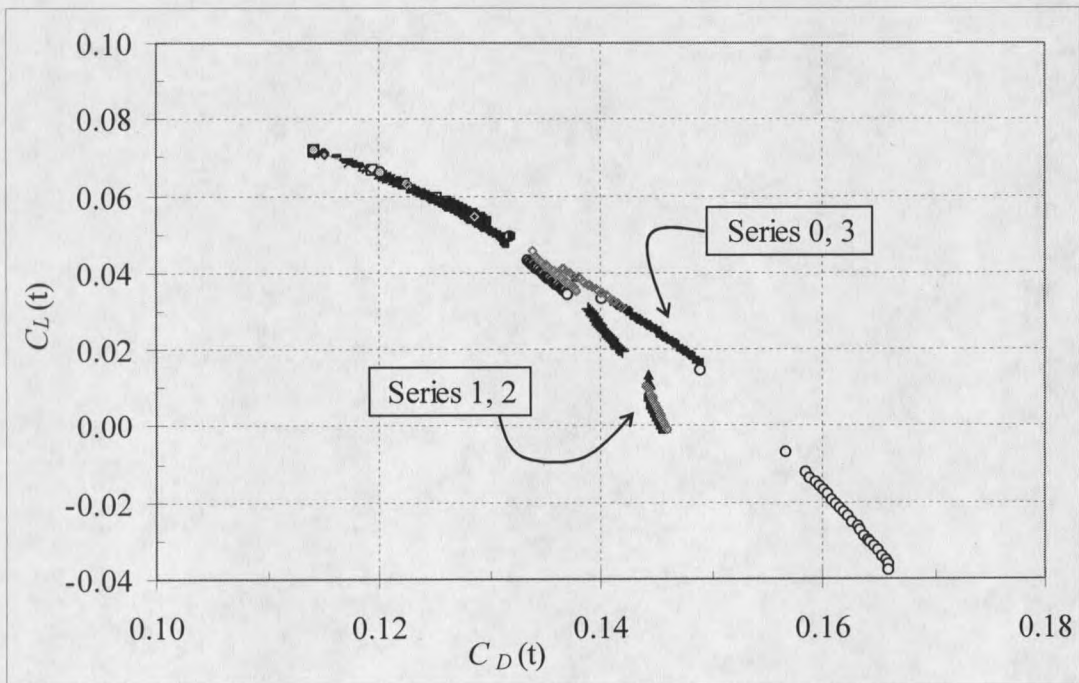


Figure 61. Composite Polar Diagram for Series 0 Through 3.



The effects of changing the mean channel velocity and the radius of the biofilm hemisphere were also examined using polar diagrams. Figure 64 is a polar diagram of the simulations in Series 0, 4 and 5. Figure 65 is a similar plot demonstrating the effect of varying biofilm radius using Series 0, 6 and 7.

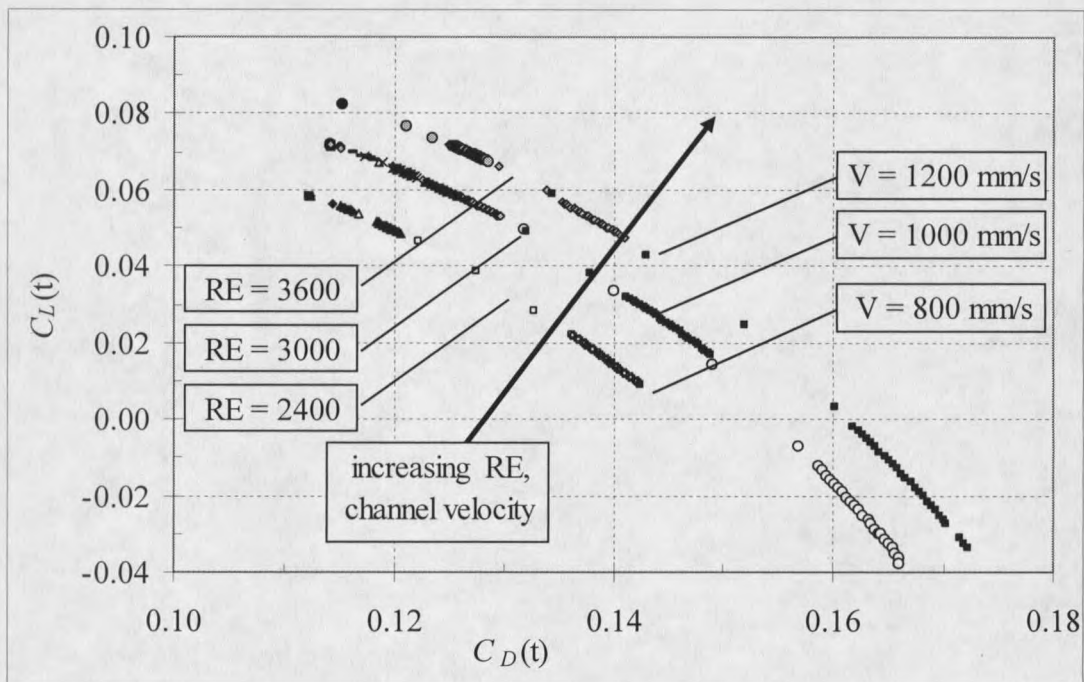


Figure 62. Effect of Channel Velocity,  $RE$ , on  $C_D$  and  $C_L$ . Series 0, 4, 5

As the mean channel velocity was increased from 800 mm/s to 1000 mm/s to 1200 mm/s the forces acting upon the attached biofilm increased. Figure 64 demonstrates a clear shift in the polar diagram as channel velocity is increased. Though more simulations may be required to determine the exact relationship, the shift does not appear to be proportional to the increase in velocity. The offset of the path taken at  $RE = 3600$

from the path taken at  $RE = 3000$  appears to be less than the similar offset from the path taken at  $RE = 2400$ . Nevertheless, the separate paths are similar in character suggesting that a simple relationship may exist between channel velocity (and Reynolds number) and the polar diagram offset.

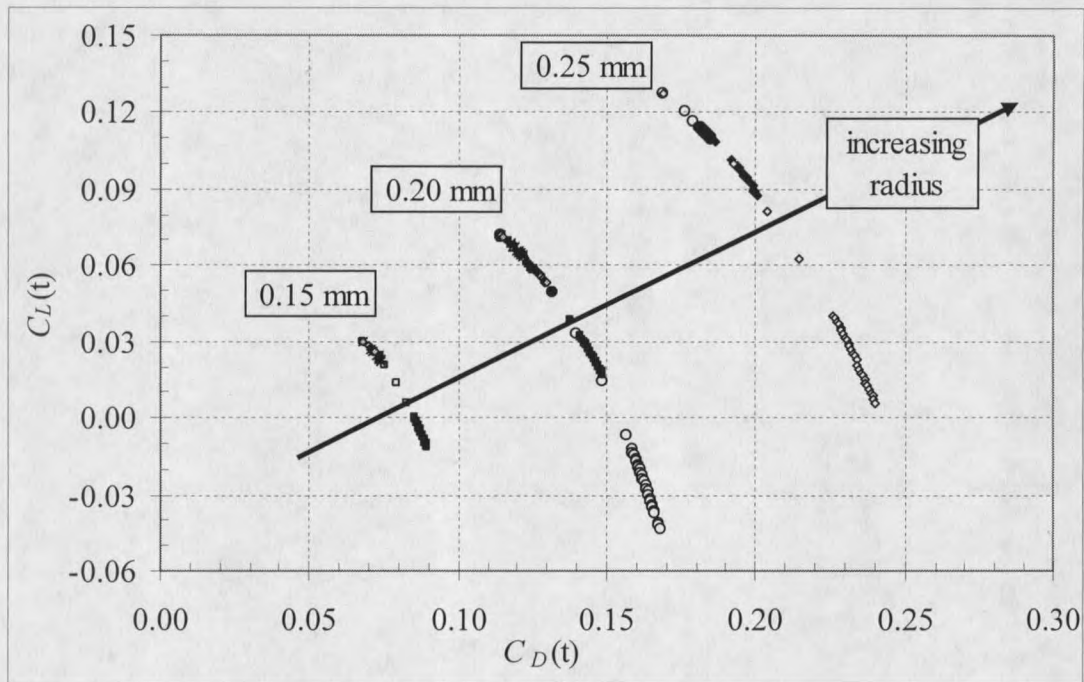


Figure 63. Effect of Biofilm Radius on  $C_D$  and  $C_L$ . Series 0, 6, 7.

The radius of the biofilm structure was varied from 0.15 mm to 0.20 mm to 0.25 mm in Series 0, 6 and 7. All other parameters were the same. A larger radius led to more biofilm surface area and a greater penetration of the fluid boundary layer. The expectation was that the lift and drag forces acting on the larger biofilms would be greater. Figure 65 illustrates the increases biofilm size did indeed increase lift and drag

forces. As with Figure 64, more simulations may be required to determine the actual relationship between radius and applied force. The offset for these paths do not appear proportional to the increase in radius. The offset from the path generated by a biofilm of radius 0.20 mm to the one with a radius of 0.25 mm is slightly greater than the offset to the path for a biofilm of radius 0.15 mm. It is likely that additional simulations may elucidate this relationship, however. Moreover, additional simulations examining the influence of other parameters may be equally insightful.



## SUMMARY

Biofilm is a ubiquitous natural phenomenon that impacts a variety of industrial applications. Research in this field is particularly relevant to environmental and water resources engineering. Despite its pervasiveness, historically, little attention was paid to the mechanical nature of this polymeric substance. More recently, investigators have begun to study the viscoelastic properties of microbial biofilms. However, few studies have properly couched these material properties in the context of a constitutive relation. This dissertation represents an in-depth investigation of the material properties of a mixed-culture microbial biofilm. The project consisted of three distinct stages: experimentation, material analysis, and computational modeling.

In the initial stage, mixed-culture biofilms were grown in a custom made microbial growth reactor. The samples were then subjected to creep tests using a computer-controlled rotating disk rheometer. An analysis of this rheometry data revealed that the samples were viscoelastic in nature. Additionally, these biofilms demonstrated a capacity for unlimited deformation under a finite stress. Accordingly, this allowed for these mixed population biofilms to be further classified as viscoelastic *fluids*. Finally, creep curves in all samples clearly showed the presence of viscous, time-dependent and instantaneous components to the biofilm compliance functions. These results were used in the subsequent phase of the investigation: material analysis.

Material analysis began with a determination of whether or not the biofilms displayed linear viscoelastic behavior. Application of superposition principles were used

to examine creep responses under two different loading conditions ( $\tau_0 = 0.1$  Pa, and  $\tau_0 = 0.5$  Pa). A linear regression analysis of these two different loadings resulted in a coefficient of determination near unity ( $r^2 = 0.8819$ ). This supported the treatment of these samples as linear viscoelastic fluids within the stress range of 0.1 Pa to 0.5 Pa. A specific linear viscoelastic constitutive equation was then determined by fitting experimental results to analytical solutions. The discontinuity in the creep curves prohibited the use of derivative-based regression or optimization routines. Therefore, a unique numerical root-finding algorithm was written, coded in MATLAB and applied to the creep data. The results indicated that the linear viscoelastic Burger material model closely approximated the behavior of these biofilms. This constitutive law was used in the development of a computational model.

In the final stage, the validated material model was incorporated into a computational simulation of biofilm growing on a wetted surface using ANSYS. The scale and boundary conditions used were based on the seminal work by Stoodley, et al. (1999b). The analytical solution to a 1-D relaxation test was used to validate the numerical approximation of the Burger stress-strain relation. The biofilm fluid-structure model was initially developed using a simple shear problem. The model readily demonstrated convergence at these scales. The computational model was then employed in a comprehensive parametric study to investigate biofilm behavior. All four of the independent Burger equation parameters,  $G_1$ ,  $\tau_1$ ,  $G_2$  and  $\tau_2$ , were examined as well as the effects of mean channel flow velocity and biofilm size.

Results clearly demonstrated the effects of increasing and decreasing the material constants. It was shown that weaker or softer biofilms (characterized by lower elastic moduli) were highly susceptible to lift forces and consequently were subject to even greater drag forces found higher in the velocity field. Also, a bimodal deformation rate was observed in Series 0 and 3. It was suggested that these two different rates of deformation were expressions of the two different relaxation times found in the Burger material model. Lift and drag coefficients were calculated by normalizing the x and y components of the fluid forces against the mean velocity head. Polar diagrams were generated by plotting the coefficients of drag versus lift. These diagrams lent insight into different aspects of biofilm behavior, particularly that of rolling biofilm structures. The polar diagrams suggested that in the first few seconds after loading, the deformation paths taken by viscoelastic biofilms are largely insensitive to specific material coefficients. Moreover, the effects of scale could be predicted from these plots. This suggests that given a sufficient number of simulations to more thoroughly define these behaviors, the development of a simple, yet accurate method for predicting the hydrodynamic forces acting on an attached biofilm is possible. In light of the economic and human health impacts of interfacial biofilm growth and detachment, the value of these results is substantial.

## REFERENCES CITED

AIAA. 1998. "Guide for the Verification and Validation of Computational Fluid Dynamics Simulations". *AIAA G-077-1998*, American Institute of Aeronautics and Astronautics, Reston, VA.

Anderson, J.D. 1995. *Computation Fluid Dynamics. The Basics with Applications*. McGraw-Hill, Inc. New York, New York.

ANSYS. 2003. *ANSYS Theory Reference manual*. ANSYS, Inc., Canonsburg, PA

Bakke, R., 1986. Biofilm Detachment. Doctoral Thesis, Montana State University, Bozeman, MT.

Bryers, J., Characklis, W.G. 1981. Early fouling biofilm formation in a turbulent flow system: Overall kinetics. *Wat. Res.* 15:483-491

Callister, W.D. 1985. *Materials Science and Engineering*. John Wiley & Sons, Inc., New York, NY.

Chapra, S.C., Canale, R.P. 1998. *Numerical Methods for Engineering*. 3rd Ed. WCB/McGraw-Hill, Boston, MA.

Characklis, W.G. 1973. Attached microbial growths. *Water Res.* 7: 1113-1127.

Characklis, W.G. 1980. *Biofilm Development and Destruction*, Final Report, EPRI CS-1554, Project RP902-1, Electric Power Research Institute, Palo Alto, CA, 1980.

Characklis, W.G., Nimmons, N.J., Picologlou, B.F. 1981. Influence of fouling biofilms on heat transfer. *J. Heat Transfer Eng.*, 1(3): 23-37

Characklis, W.G. 1981. Fouling Biofilm development: a process analysis. *Biotechnol. Bioeng.* 23: 1923-1960.

Characklis, W.G. 1990. Microbial Fouling. In *Biofilms*, Ed. Characklis, W.G., Marshall, K.C., John Wiley & Sons, Inc., New York, NY

Characklis, W.G., Marshall, K.C. 1990. Biofilms: A Basis for An Interdisciplinary Approach. In *Biofilms*, Ed. Characklis, W.G., Marshall, K.C., John Wiley & Sons, Inc., New York, NY

Christensen, R.M. 1982. *Theory of Viscoelasticity*. 2<sup>nd</sup> Ed. Academic Press, Inc. New York, NY.

Christensen, B.E., Characklis, W.G., 1990. Physical and Chemical Properties of Biofilms. In *Biofilms*, Ed. Characklis, W.G., Marshall, K.C., John Wiley & Sons, Inc., New York, NY

Cooksey, K.E., Wigglesworth-Cooksey, B. 1995. Adhesion of bacteria and diatoms to surfaces in the sea: a review. *Aquatic Microbial Ecology*. 9: 87-96.

Costerton, J.W., Stewart, P.S., Greenberg, E.P. 1999. Bacterial Biofilms: a common cause of persistent infections. *Science*. 284: 1318-1322.

Cunningham, A.C., Bouwer, E.J., Characklis, W.G. 1990. Biofilms in Porous Media. In *Biofilms*, Ed. Characklis, W.G., Marshall, K.C., John Wiley & Sons, Inc., New York, NY

Cunningham, A.C., Warwood B., Sturman P., Horrigan K., James G., Costerton J.W., Hiebert R.. 1997. "Biofilm Process in Porous Media - Practical Applications" In *The Microbiology of the Terrestrial Deep Subsurface*. Ed. Amy P.S., Haldeman D.L., Lewis Publishers, pp. 325-344.

Currie, I.G. 1974. *Fundamental Mechanics of Fluids*. McGraw-Hill, Inc. New York, NY

Debler, W.R. 1990. *Fluid Mechanics Fundamentals*. Prentice Hall. Englewood Cliffs, NJ.

Dockery, J., Klapper, I. 2001. Finger Formation in biofilm layers. *SIAM. J. Appl. Math.* 62: 853-869.

Doyle, J.J. 2003. Phone conversation, 8/12/2003. ANSYS Technical Support, ANSYS, Inc., Canonsburg, PA

Findley, W.N., Lai, J.S., Onaran, K. 1989. *Creep and Relaxation of Nonlinear Viscoelastic Materials*. Dover Publishing, Minneola, NY.

Flemming, H.C. 1987. Microbial Growth on Ion Exchangers. *Wat. Res.* 21(7):745-756

Flemming, H.C., Wingender, J., Mayer, C., Korstgens, V., Borchard, W. 2002. Cohesiveness in biofilm matrix polymers. *SGM symposium* 59

Flugge, W.F. 1975. *Viscoelasticity*, 2<sup>nd</sup> Ed. Springer-Verlag, New York, NY.

Giles, R.V., Evett, J.B., Liu, C. 1956. *Schaum's Outline of Theory and Problems of Fluid Mechanics and Hydraulics*. McGraw-Hill, Inc. New York, NY.

Gittus, J. 1975. *Creep Viscoelasticity and Creep Fracture in Solids*. Applied Science Publishers, Ltd. Essex, England

- Gjaltema, A., Arts, P.A.M., van Loosdrecht, M.C.M., Kuenen, J.G. and Heijnen, J.J. 1994. Heterogeneity of biofilms in rotating annular reactors: occurrence, structure and consequences. *Biotechnol. Bioeng.* 44: 194-204
- Gujer W., Wanner, Wanner, O. 1990. Modeling Mixed Population Biofilms. In *Biofilms*, Ed. Characklis, W.G., Marshall, K.C., John Wiley & Sons, Inc., New York, NY
- Hermanowicz, W.S. 1998. Two-dimensional simulations of biofilm development: effects of external environmental conditions. In *Conference proceedings, international specialty conference. Microbial ecology of biofilms: concepts, tools and applications*. Ed. Rittmann, B.E., Bishop, P., Bouwer, E., Cunningham, A. Lake Bluff, IL.
- Hoehn, R.C., Ray, A.D. 1973. *J. Water Poll. Contr. Fed.* 46: 2302.
- Janna, W.S. 1993. *Introduction to Fluid Mechanics*, 3<sup>rd</sup> Ed. PWS Publishing Co., Boston, MA.
- Klapper, I., Rupp, C.J., Cargo, R., Purevdorj, B., and Stoodley, P. 2002. A viscoelastic fluid description of bacterial biofilm material properties. *Biotechnol. Bioeng.* 80: 289-296.
- Kornegay, B.H., Andrews, J.F. 1967. Characteristics and Kinetics of Mixed Film Biological Reactors, *Final Report*, Grant WP-01181, Federal Water Pollution Control Administration, U.S. G.P.O., Washington, D.C.
- Korstgens V., Flemming H-C., Wingender J., Borchard W., 2001. Influence of calcium ions on the mechanical properties of a model biofilm of mucoid *Pseudomonas aeruginosa*. *Wat. Sci. Technol.* 43: 49-57.
- Lai, W.M., Rubin, D. Krempf, E. 1993. *Introduction to Continuum Mechanics*. Pergamon Press, Oxford, England.
- Lewthwaite, J.C., Molland, A.F., Thomas, K.W. 1985. An Investigation into the Variation of Ship Skin Frictional Resistance with Fouling. *Trans. Royal Inst. Of Naval Arch.*, vol. 127, p. 269-284
- Mase, G.T. and Mase, G.E. 1999. *Continuum Mechanics for Engineers.*, 2<sup>nd</sup> Ed., CRC Press LLC, Boca Raton, FL.
- Morrison, F.A. 2001. *Understanding Rheology*. Oxford University Press. New York, NY.
- Munem, M.A., Foulis, D.J. 1984. *Calculus with Analytic Geometry*. Worth Publishers, Inc., New York, NY.

- Newman, H.N. 1998. Periodontal therapeutics--a viable option? *Int Dent J.* 48(3): 173-9.
- Ohashi, A., Harada, H. 1994. Adhesion strength of biofilm developed in an attached-growth reactor. *Wat. Sci. Tech.* 29:10-11, 281-288
- Phan-Thien, N. 2002. *Understand Viscoelasticity. The Basics of Reology.* Springer-Verlag. Berlin, Germany.
- Picologlou B.F., Zilver N., Characklis W.G. 1980. Biofilm Growth and Hydraulic Performance. *J Hydraul Div Am Soc Civ Eng* 106(HY5): 733-746
- Piciooreanu, C., van Loosdrecht, M.C., Heijnen, J.J. 2001. Two-dimensional model of biofilm detachment caused by internal stress from liquid flow. *Biotechnol. Bioeng.* 72: 205-218.
- Plazek, D.J. 1965. Temperature dependence of the viscoelastic behavior of polystyrene. *J. Phys. Chem.* 69: 3480-3487.
- Pritchard, A.M. 1981. Fouling – science or art? An investigation of fouling and antifouling measures in the British Isles. In *Fouling of Heat Transfer Equipment.* Ed. E.F.C. Somerscales and J.G. Knudsen. Hemisphere, Washington.
- Ridgway, H.F., Justice, C.A., Whittaker, D.G., Argo, D.G. and Olson, B.H. 1984. Biofouling. *J. AWWA.* 76, 94
- Rouse, H. 1946. *Elementary Mechanics of Fluids.* John Wiley & Sons, Inc. New York, NY. Chapman & Hall, Limited. London, U.K.
- Rupp, C., Wilson, S., Stoodley, S. 2002. *Staphylococcus aureus* Biofilm Rolling Along the Lumen of a Glass Tube. <http://www.microbelibrary.org/index.html>. American Society for Microbiology.
- Shackelford, J.F. 1985. *Introduction to Material Science for Engineers,* Macmillan Publishing Company, New York, NY.
- Stark, P.A. 1970. *Introduction to Numerical Methods.* The Macmillan Company, Collier-Macmillan Limited, London, U.K.
- Stoodley, P., Boyle, J.D., Dodds, I., and Lappin-Scott, H.M. 1997. Consensus model of biofilm structure. In *Biofilms: Community Interactions and Control.* Ed. Wimpenny, J.W.T., Handley, P.S., Gilbert, P., Lappin-Scott, H.M. and Jones, M. Cardiff:BioLine



Stoodley, P., Lewandowski, Z., Boyle, J.D., Lappin-Scott, H.M. 1998. Oscillation Characteristics of Biofilm Streamers in Turbulent Flowing Water as Related to Drag and Pressure Drop. *Biotech. Bioeng.* 57: 536-544.

Stoodley, P., Lewandowski, Z., Boyle, J.D., Lappin-Scott, H.M. 1999a. The formation of migratory ripple in a mixed species bacterial biofilm growing in turbulent flow. *Environmental Microbiology.* 1(5): 447-455.

Stoodley, P., Lewandowski, Z., Boyle, J.D., Lappin-Scott, H.M. 1999b. Structural deformation of bacterial biofilms caused by short term fluctuations in flow velocity: an in-situ demonstration of biofilm viscoelasticity. *Biotech. Bioeng.* 65: 83-92.

Stoodley, P., Jacobsen, A., Dunsmore, B.C., Purevdorj, B., Wilson, S., Lappin-Scott, H.M. Costerton, J.W. 2001. The Influence of Fluid Shear and AlCl<sub>3</sub> on the Material Properties of *Pseudomonas aeruginosa* PAO1 and *Desulfovibrio* sp. EX265 Biofilms. *Wat. Sci. Technol.* 43: 113-120.

Stoodley, P., Cargo, R., Rupp C.J., Wilson, S. and Klapper, I. 2002. Biofilm Material Properties as related to shear-induced deformation and detachment phenomena. *J. Ind. Microb. And Biotech.* 29: 361-367.

TA Instruments, Inc. 1999. *Instrument Manual, AR1000 Rheometer.* 1999. TA Instruments, Inc., New Castle, DE.

Thompson, J. and Chang, S. 2003. Phone conversation, 12/19/2002. ANSYS Technical Support, ANSYS, Inc., Canonsburg, PA

Trulear, M.G. 1983. Cellular reproduction and extracellular polymer formation in the development of biofilms. Doctoral Thesis, Montana State University, Bozeman, MT.

Turakhia, M.H. 1986. The influence of calcium on biofilm processes. Doctoral Thesis, Montana State University, Bozeman, MT.

Vennard, J.K. 1940. *Elementary Fluid Mechanics.* John Wiley & Sons, Inc. New York, NY

Walker J.T., Mackerness C.W., Mallon D., Makin T., Williets T., Keevil C.W. 1995. Control of legionella-pneumophila in a hospital water-system by chlorine dioxide. *J. Ind. Microbio.* 15: 384-390.

Wilcox, D.C. 1993. *Turbulence Modeling for CFD.* DCW Industries, LaCanada, CA.

Zelver, N. 1979. Biofilm development and associated energy losses in water conduits. M.S. Thesis, Rice University, Houston, TX.

APPENDICES

APPENDIX A

BURGER COMPLIANCE CONVERSION VIA LAPLACE TRANSFORM

### Symbolic Determination of the Prony series coefficients in terms of Burger parameters

Recall the terms used in the definition of the Relaxation modulus:

$$p_0 := 1 \quad p_1 := \frac{\eta_1}{G_1} + \frac{\eta_1}{G_2} + \frac{\eta_2}{G_2} \quad p_2 := \frac{\eta_1 \cdot \eta_2}{G_1 \cdot G_2}$$

$$q_1 := \eta_1 \quad q_2 := \frac{\eta_1 \cdot \eta_2}{G_2}$$

$$A := \sqrt{p_1^2 - 4 \cdot p_2} \quad r_1 := \frac{p_1 - A}{2 \cdot p_2} \quad r_2 := \frac{p_1 + A}{2 \cdot p_2}$$

A 2-term Prony series takes the form:

$$g(t) := g_0 \cdot \left( \alpha_{\text{inf}} + \alpha_1 \cdot e^{-\frac{t}{\tau_1}} + \alpha_2 \cdot e^{-\frac{t}{\tau_2}} \right)$$

$$\text{where:} \quad g_0 := g_{\text{inf}} + g_1 + g_2 \quad e_0 := e_{\text{inf}} + e_1 + e_2$$

$$\alpha_{\text{inf}} := \frac{g_{\text{inf}}}{g_0} \quad \alpha_1 := \frac{g_1}{g_0} \quad \alpha_2 := \frac{g_2}{g_0}$$

$$e_{\text{inf}} := g_{\text{inf}} \cdot 2 \cdot (1 + \nu) \quad e_0 := g_0 \cdot 2 \cdot (1 + \nu)$$

$$e_1 := g_1 \cdot 2 \cdot (1 + \nu) \quad e_2 := g_2 \cdot 2 \cdot (1 + \nu)$$

Equating terms in the Relaxation modulus with the Prony series

$$g_{\text{inf}} := 0 \quad g_1 := \frac{(q_1 - q_2 \cdot r_1)}{A} \quad g_2 := -\frac{(q_1 - q_2 \cdot r_2)}{A}$$

$$\tau_1 := \frac{1}{r_1} \quad \tau_2 := \frac{1}{r_2}$$

Evaluating the coefficients in the Prony series:

$$g_1 \rightarrow \frac{\eta_1 - \frac{1}{2} \left[ \frac{\eta_1}{G_1} + \frac{\eta_1}{G_2} + \frac{\eta_2}{G_2} - \left[ \left( \frac{\eta_1}{G_1} + \frac{\eta_1}{G_2} + \frac{\eta_2}{G_2} \right)^2 - 4 \cdot \eta_1 \cdot \frac{\eta_2}{G_1 \cdot G_2} \right]^{\frac{1}{2}} \right] \cdot G_1}{\left[ \left( \frac{\eta_1}{G_1} + \frac{\eta_1}{G_2} + \frac{\eta_2}{G_2} \right)^2 - 4 \cdot \eta_1 \cdot \frac{\eta_2}{G_1 \cdot G_2} \right]^{\frac{1}{2}}}$$

$$g_2 \rightarrow \frac{- \left[ \eta_1 - \frac{1}{2} \left[ \frac{\eta_1}{G_1} + \frac{\eta_1}{G_2} + \frac{\eta_2}{G_2} + \left[ \left( \frac{\eta_1}{G_1} + \frac{\eta_1}{G_2} + \frac{\eta_2}{G_2} \right)^2 - 4 \cdot \eta_1 \cdot \frac{\eta_2}{G_1 \cdot G_2} \right]^{\frac{1}{2}} \right] \right] \cdot G_1}{\left[ \left( \frac{\eta_1}{G_1} + \frac{\eta_1}{G_2} + \frac{\eta_2}{G_2} \right)^2 - 4 \cdot \eta_1 \cdot \frac{\eta_2}{G_1 \cdot G_2} \right]^{\frac{1}{2}}}$$

$$\tau_1 \rightarrow \frac{2}{\frac{\eta_1}{G_1} + \frac{\eta_1}{G_2} + \frac{\eta_2}{G_2} - \left[ \left( \frac{\eta_1}{G_1} + \frac{\eta_1}{G_2} + \frac{\eta_2}{G_2} \right)^2 - 4 \cdot \eta_1 \cdot \frac{\eta_2}{G_1 \cdot G_2} \right]^{\frac{1}{2}}} \cdot \eta_1 \cdot \frac{\eta_2}{G_1 \cdot G_2}$$

$$\tau_2 \rightarrow \frac{2}{\frac{\eta_1}{G_1} + \frac{\eta_1}{G_2} + \frac{\eta_2}{G_2} + \left[ \left( \frac{\eta_1}{G_1} + \frac{\eta_1}{G_2} + \frac{\eta_2}{G_2} \right)^2 - 4 \cdot \eta_1 \cdot \frac{\eta_2}{G_1 \cdot G_2} \right]^{\frac{1}{2}}} \cdot \eta_1 \cdot \frac{\eta_2}{G_1 \cdot G_2}$$

### Relaxation Modulus for a 4 element Burger model

Given the material parameters for the linear model:

$$G_1 := 0.5 \cdot \frac{\text{N}}{\text{m}^2} \quad \eta_1 := 100 \cdot \frac{\text{kg}}{\text{m} \cdot \text{sec}} \quad \text{kelvin element (parallel)}$$

$$G_2 := 0.5 \cdot \frac{\text{N}}{\text{m}^2} \quad \eta_2 := 100 \cdot \frac{\text{kg}}{\text{m} \cdot \text{sec}} \quad \text{maxwell element (series)}$$

To simplify the relaxation equation, repeating terms are grouped and defined:

$$p_0 := 1 \quad p_1 := \frac{\eta_1}{G_1} + \frac{\eta_1}{G_2} + \frac{\eta_2}{G_2} \quad p_2 := \frac{\eta_1 \cdot \eta_2}{G_1 \cdot G_2}$$

$$q_1 := \eta_1 \quad q_2 := \frac{\eta_1 \cdot \eta_2}{G_2}$$

$$A := \sqrt{p_1^2 - 4 \cdot p_2} \quad r_1 := \frac{p_1 - A}{2 \cdot p_2} \quad r_2 := \frac{p_1 + A}{2 \cdot p_2}$$

Then the relaxation modulus is defined as (Findley, Lai and Onaran, 1976, pg. 73-75)

$$E(t) := \frac{(q_1 - q_2 \cdot r_1)}{A} \cdot e^{-r_1 \cdot t} - \frac{(q_1 - q_2 \cdot r_2)}{A} \cdot e^{-r_2 \cdot t}$$

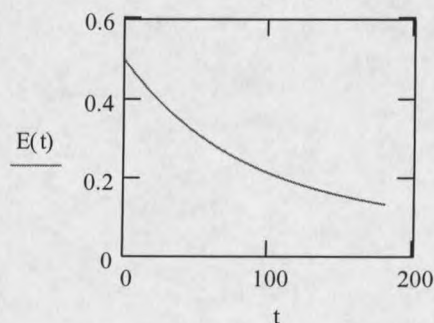
Defining a time period over which to apply the modulus:

$$t := 0.1 \cdot \text{s}, 0.2 \cdot \text{s}.. 180 \cdot \text{sec}$$

The values of  $E(t)$  are calculated and graphed:

$$E(t) =$$

|       |    |
|-------|----|
| 0.5   | Pa |
| 0.499 |    |
| 0.499 |    |
| 0.498 |    |
| 0.498 |    |
| 0.497 |    |
| 0.497 |    |
| 0.496 |    |



To verify these results, the relaxation modulus is calculated from the operational modulus,  $E_s(t)$ :

Then the operational modulus is defined as (Findley, Lai and Onaran, 1976, pg. 73-75)

$$E_s := \frac{q_1 + q_2 \cdot u}{1 + p_1 \cdot u + p_2 \cdot u^2}$$

And the relaxation modulus is the inverse LaPlace transform of the operational modulus:

$$E_2(t) := E_s \text{ invlaplace, } u \rightarrow .50000000000000000000 \cdot \frac{\text{N}}{\text{m}^2} \cdot \exp\left(\frac{-7.50000000000000000000 \cdot 10^{-3}}{\text{kg} \cdot \text{m}} \cdot \text{sec} \cdot \text{N} \cdot t\right)$$

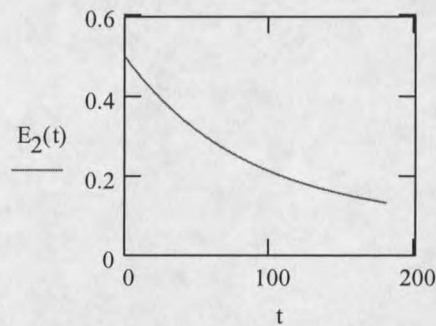
*note: the relaxation modulus is defined as  $E_2(t)$  to differentiate it from  $E(t)$  which was calculated directly above.*

The values of  $E_2(t)$  are calculated and graphed:

$E_2(t) =$

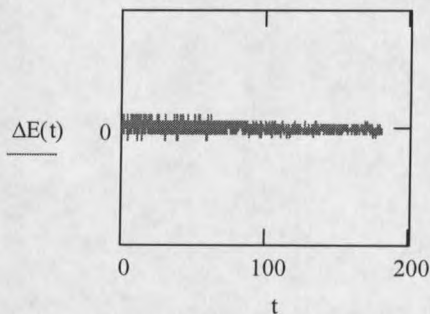
|       |
|-------|
| 0.5   |
| 0.499 |
| 0.499 |
| 0.498 |
| 0.498 |
| 0.497 |
| 0.497 |
| 0.496 |

Pa



For comparison, the residuals are each time,  $t$  are calculated and graphed.

$\Delta E(t) := E(t) - E_2(t)$  residual function is the difference between the two



$$\cdot \cosh\left(\frac{5.5901699437494742410 \cdot 10^{-3}}{\text{kg} \cdot \text{m}} \cdot \text{sec} \cdot \text{N} \cdot \text{t}\right) - .22360679774997896964 \cdot \frac{\text{N}}{\text{m}^2} \cdot \exp\left(\frac{-7.5000000000000000}{\text{kg} \cdot \text{m}}\right)$$



APPENDIX B

MATLAB CODE FOR SLIDE & STAGGER ALGORITHM

Filename: LVE\_i.m

```
% -----
% -----
% VISCOLASTIC MODEL CURVE FITTING 'ITERATIVE' DRIVER
%   Brett Towler
%   Apr 25, 2003
%
%   NOTE: This driver calls the LVEfit.m MATLAB function
%   iteratively through
%   the use of the LVE_s (1st iteration) and LVE_r (2nd +)
%   scripts. Use
%   this driver when the solution is to be iterated upon
%   until the change
%   in the WR is less than the criteria identified in
%   WR_conv.
%
% -----
% Initial Call
% -----
%
% fprintf(1, '\n');
% fprintf(1, '-----\n');
% fprintf('Starting LVE_s, LVE_i iteration = 1\n');
% fprintf(1, '-----\n');
% -----\n');
LVE_s;                                % calls Start-up
routine
%
% -----
% User Input
% -----
%
%                                     % Convergence criterion
WR_Conv=1e-10;                        % minimum number of
min_loops=25;
loops to run
%
% -----
% -----
```

```

% Variable Initializations
% -----
%
loop=1;                                % initialize loop
counter
WR_Stack=zeros(5,1);                  % builds stack for
storing WR_O
WR_Exchange=zeros(5,1);               % builds stack for
exchanging WR_O
WR_Error=1;                          % spike WR_Error for
1st loop
%
% -----
% Restart Iterations
% -----
%
while (WR_Error > WR_Conv) | (loop < min_loops) %
start of loop
    fprintf(1, '\n');
    fprintf(1, '-----\n');
    fprintf('Starting LVE_r, LVE_i iteration = %3.0f,
WR_Error = %g\n', loop+1, WR_Error);
    fprintf(1, '-----\n');
    -----\n');
    LVE_r;                            % restart
LVE
    WR_Exchange=WR_Stack;             % redefine
exchange stack
    for i=2:5
        WR_Stack(i)=WR_Exchange(i-1); % shift
WR_O in stack down
    end
    WR_Stack(1)=WR_O;                % add new
WR_O to stack top
    WR_Error=abs(WR_Stack(1) - (sum(WR_Stack(2:5))/4)); %
redefine WR_Error
    loop=loop+1;                     % advance
counter
end                                  % end of
loop
%

```

```

% -----
% Screen Output
% -----
%
fprintf(1, '\n');
fprintf(1, '*****\n');
fprintf('LVE iteration complete.\n');
fprintf('Total number of LVE_i iterations =
%3.0f\n', loop+1);
fprintf('WR_Error = %g\n', WR_Error);
fprintf(1, '*****\n');
%

```

Filename: LVE\_r.m

```
% -----
% -----
% VISCOLASTIC MODEL CURVE FITTING 'RE-START' DRIVER
%   Brett Towler
%   Apr 25, 2003
%
%   NOTE: This driver calls the LVEfit.m MATLAB function.
Use this driver
%   when parameters are to be taken the previous run. Use
the LVE_s driver
%   when parameters limits are to be entered manually.
%
% -----
% -----
% User Input
% -----
% -----
%
G1_red=2;                % default = 3
n1_red=2;
G2_red=2;
n2_red=2;
%
% -----
% -----
% Variable Initializations
% -----
% -----
%
G1_range=abs(G1_l-G1_u)/G1_red;    % reduces G1 range
n1_range=abs(n1_l-n1_u)/n1_red;    % reduces n1 range
G2_range=abs(G2_l-G2_u)/G2_red;    % reduces G2 range
n2_range=abs(n2_l-n2_u)/n2_red;    % reduces n2 range
%
% -----
% -----
% Restart Input
% -----
% -----
%
G1_l=P_O(1) - G1_range/2;          % units = N/m2
(Pascals)
```

```

G1_u=P_O(1) + G1_range/2;      % units = N/m2
(Pascals)
n1_l=P_O(2) - n1_range/2;      % units = N-s/m2
n1_u=P_O(2) + n1_range/2;      % units = N-s/m2
G2_l=P_O(3) - G2_range/2;      % units = N/m2
(Pascals)
G2_u=P_O(3) + G2_range/2;      % units = N/m2
(Pascals)
n2_l=P_O(4) - n2_range/2;      % units = N-s/m2
n2_u=P_O(4) + n2_range/2;      % units = N-s/m2
%
% builds the limits, L, matrix, 4 x 2, from user input
above
L=[G1_l G1_u;n1_l n1_u;G2_l G2_u;n2_l n2_u];
%
%
% -----
% Screen Output
% -----
%
fprintf(1,'\n');
fprintf('Parameter range sizes: %g, %g, %g,
%g\n',G1_range,n1_range,G2_range,n2_range);
fprintf(1,'-----\n');
%
% -----
% LVEfit call
% -----
%
P_WR=LVEfit(dataset,s0,ts,L);
P_O=P_WR(1:4);
WR_O=P_WR(5);

```

Filename: LVE\_s.m

```
% -----
% -----
% VISCOLASTIC MODEL CURVE FITTING 'START' DRIVER
%   Brett Towler
%   Apr 25, 2003
%
%   NOTE: This driver calls the LVEfit.m MATLAB function.
Use this driver
%   when parameters limits are to be entered manually. Use
the LVE_r
%   driver when parameters are to be taken the previous
run.
%
clear;
clc;
%
% -----
% -----
% User Input
% -----
% -----
%
% The user should change the constant shear stress
according to the test
% parameters used in the creep test 'dataset'.
Additionally, the user
% should select a range of initial estimates for each of the
4 parameters
% in the Burger model. NOTE: The true value of the
parameter need not be
% between these limits. The algorithm will adapt. Note
that a wide range
% in parameter limits will slow convergence and a narrow
range could inhibit
% convergence. Regardless, the values of all material
parameters should be
% positive.
%
dataset='5c1_ob';           % no file extension
needed
s0=0.1;                     % units = N/m2
(Pascals)
ts=181;                     % units = seconds
```

```

G1_l=0.1;                                % units = N/m2
(Pascals)
G1_u=2;                                  % units = N/m2
(Pascals)
n1_l=0.1;                                % units = N-s/m2
n1_u=100;                                % units = N-s/m2
G2_l=0.1;                                % units = N/m2
(Pascals)
G2_u=10;                                  % units = N/m2
(Pascals)
n2_l=0.1;                                % units = N-s/m2
n2_u=100;                                % units = N-s/m2
%
% end user input!!
%
% -----
% Variable Initializations
% -----
%
% builds the limits, L, matrix, 4 x 2, from user input
above
L=[G1_l G1_u;n1_l n1_u;G2_l G2_u;n2_l n2_u];
G1_range=abs(G1_l-G1_u);                  % G1 range
n1_range=abs(n1_l-n1_u);                  % n1 range
G2_range=abs(G2_l-G2_u);                  % G2 range
n2_range=abs(n2_l-n2_u);                  % n2 range
%
%
% -----
% Screen Output
% -----
%
fprintf(1,'-----\n');
fprintf(1,'LVEfit');
fprintf(1,'\n');
fprintf('Parameter range sizes: %g, %g, %g,
%g\n',G1_range,n1_range,G2_range,n2_range);
fprintf(1,'-----\n');
%

```



```
% -----  
-----  
% LVEfit call  
% -----  
-----  
%  
P_WR=LVEfit(dataset,s0,ts,L);  
P_O=P_WR(1:4);  
WR_O=P_WR(5);
```

Filename: LVEfit.m

```
% -----
% -----
% VISCOLASTIC MODEL CURVE FITTING FUNCTION
%   Brett Towler
%   Apr 25, 2003
%
%   NOTE: This function calls the Burger.m MATLAB function
and imports
%   data from the comma-delineated ASCII file designated by
the
%   user in the definition of the variable 'dataset'.
LVEfit is called by
%   either LVE-s.m or LVE-r.m.
%
function LVEreturn=LVEfit(dataset,s0,ts,L)
%
% -----
% -----
% Variable Definitions
% -----
% -----
%
% CreepData      n x 2 matrix containing time, strain
observations
% dataset        text file containing name of creep test
data file
% Del_L          4 x 1 matrix containing node spacing on
parameters
% G1_l           Lower limit on 1st elastic constant in
Burger model
% G1_u           Upper limit on 1st elastic constant in
Burger model
% G2_l           Lower limit on 2nd elastic constant in
Burger model
% G2_u           Upper limit on 2nd elastic constant in
Burger model
% i              dummy variable used in many loops
% k              counter for outer loop iterations
% kmax           Maximum number of outer loop iterations
% L              4 x 2 matrix containing limits on
parameters for current k
% Ltemp          used to temporarily store L(:,1) during
reassignment
```

|            |                                                                     |
|------------|---------------------------------------------------------------------|
| % m        | number of columns in CreepData (typically 2)                        |
| % n        | number of data observations, rows in CreepData                      |
| % n1_l     | Lower limit on 1st newtonian constant in Burger model               |
| % n1_u     | Upper limit on 1st newtonian constant in Burger model               |
| % n2_l     | Lower limit on 2nd newtonian constant in Burger model               |
| % n2_u     | Upper limit on 2nd newtonian constant in Burger model               |
| % P        | 4 x 1 matrix containing current parameter values                    |
| % P_0      | 4 x 1 matrix containing optimal parameter values                    |
| % R        | n x 1 matrix containing residuals in strains                        |
| % RF       | Reduction Factor used in reducing parameter limits                  |
| % Rt_Sum   | numerator in WR calculation                                         |
| % SError   | Allowable relative error in prediction of maximum strain            |
| % SError_P | Current relative error in prediction of maximum strain              |
| % SError_O | Optimal relative error in prediction of maximum strain              |
| % SFlag    | Tracks whether parameter slides(1) or staggers(0)                   |
| % s0       | Constant shear stress used in creep test (units same as G)          |
| % Strain   | n x 1 matrix containing observed strains                            |
| % Strain_O | n x 1 matrix containing optimal strains                             |
| % Strain_P | n x 1 matrix containing predicted (Burger) strains                  |
| % t        | n x 1 matrix containing time only                                   |
| % t_Sum    | denominator in WR calculation                                       |
| % TOL      | Weighted residual convergence criterion                             |
| % Trim     | Controls parameter proximity to 0; prevents div. by 0               |
| % ts       | time of unloading; interface of retardation/recovery                |
| % WR       | current value of (weighted or standard) MAR, mean absolute residual |

```

% WR_O          current value of optimal (weighted or
standard) MAR
% WRFlag        Determines type of residual used for 'best
fit'
%
% -----
% User Input
% -----
%
% The following parameters affect convergence rate and
termination
% criteria.
%
RF=0.95;          % use 0.90 as a
default
kmax=25;          % use 50 as a
default
SError=0.10;     % Allowable rel.
error in max observed strain
WRFlag=1;         % 0 for sum of
residuals, 1 for weighted res.
%
% end user input!!
%
% -----
% Variable Initializations
% -----
%
CreepData=importdata(dataset,',' ); % imports dataset
file into CreepData matrix, n x 2
[n,m]=size(CreepData); % determines size
of the data set, n x m
t=CreepData(:,1); % builds time
matrix from 1st column of CreepData
Strain=CreepData(:,2); % builds strain
matrix from 1st column of CreepData
R=zeros(n,1); % initializes
residual column matrix
Del_L=zeros(4,1); % initializes
change in limits column matrix

```

```

SFlag=ones(4,1);                                % SFlag tracks
parameter slide/stagger
Trim=0.001;                                       % initializes Trim
settings
%
% -----
% -----
% Stagger & Slide Algorithm
% -----
% -----
%
% Step 1: Initialize the Optimized Weighted Residual
variable, WR_0
WR_0=10;                                          % spike WR_0 for
1st loop
SError_0=10.0;                                   % WR_0 selection
criterion initial spike
k=1;
P_0=L(:,1)+(L(:,2)-L(:,1))/2;
%
% Step 2: Initially set Optimized Strain equal to Observed
Strain
Strain_0=Strain;
%
% Step 3: Reduction loop - optimizes parameters iteratively

while (k < kmax+1)                                % start of kth
reduction loop

    % Step 4: Redefine node spacing on parameter loops
    for i=1:4
        Del_L(i)=(L(i,2)-L(i,1))/(SFlag(i)+1);    %
    end
    redefines Delta Limits matrix for current k

    % Step 5: Reset WR to zero
    WR=0;

    % Step 6: Calculate strain, WR_0 at all nodes in
    current limit matrix
    for P1=L(1,1):Del_L(1):L(1,2)                %
    loops thru lower, mid and upper values of P1
        for P2=L(2,1):Del_L(2):L(2,2)            %
    loops thru lower, mid and upper values of P2

```

```

        for P3=L(3,1):Del_L(3):L(3,2) %
loops thru lower, mid and upper values of P3
        for P4=L(4,1):Del_L(4):L(4,2) %
loops thru lower, mid and upper values of P4

        P=[P1;P2;P3;P4];

        for i=1:n
% Calculate n pred. strains for the kth iter.
            Strain_P(i)=Burger(s0,ts,t(i),P);
        end

        for i=1:n
            R(i)=abs(Strain_P(i)-Strain(i));
% residual between kth pred. and obs.
        end

        SError_P=abs(max(Strain)-
max(Strain_P))/max(Strain); % error in current approx.

        Rt_Sum=0;
% reset (residual*time_step) sum to 0
        t_Sum=0;
% reset (time step) sum to 0

        if WRFlag==1
% WRFlag=1-->WR, Else WR is sum of residuals
            for i=2:n
                Rt_Sum=Rt_Sum+R(i)*abs((t(i)-
t(i-1))); % sum of (residual * time step)
                t_Sum=t_Sum+abs((t(i)-t(i-1)));
% sum of (time step)
            end
        else
            for i=2:n
                Rt_Sum=Rt_Sum+R(i);
% sum of residuals
            end
            t_Sum=n;
% number of data points, n
        end

        end

        WR=(Rt_Sum/t_Sum)/max(Strain);
% calculates WR for the kth iteration

```

```

        if (WR < WR_O) & (SError_P < SError)
% checks if current k is the best fit
            P_O=P;
% current is optimal!
            Strain_O=Strain_P;
% current is optimal!
            WR_O=WR;
% current is optimal!
            SError_O=SError_P;
% current is optimal!
        end
    end
end P1 loop
end
end P2 loop
end
end P3 loop
end
end P4 loop

% Output at each reduction loop iteration
fprintf('k = %3.0f, WR_O = %G, SError_O = %4.3f, SFlag=
[ %g %g %g %g ], P_O: [ %4.2f %4.2f %4.2f %4.2f ]\n'...
,k,WR_O,SError_O,SFlag(1),SFlag(2),SFlag(3),SFlag(4),P_O(1)
,P_O(2),P_O(3),P_O(4));

% Step 7: Determine if limits slide or stagger
for i=1:4
% Loop thru all 4 parameters

    if SFlag(i)==1 % 3
points evaluated

        if (P_O(i)==L(i,1) | P_O(i)==L(i,2))
% P_O is at an end point --> slide
            Ltemp=L(i,1);
% temporarily store L(i,1) for L(i,2) calc.
            L(i,1)=P_O(i)-0.5*(L(i,2)-L(i,1));
            if L(i,1)<0
% Prevent lower limit from going negative
                L(i,1)=0+Trim;
            end
            L(i,2)=P_O(i)+0.5*(L(i,2)-Ltemp);
        end
    end
end

```

```

        SFlag(i)=1;
    else
% P_O is NOT at an end point --> stagger
        Ltemp=L(i,1);
% temporarily store L(i,1) for L(i,2) calc.
        L(i,1)=P_O(i)-0.25*(L(i,2)-L(i,1));
        if L(i,1)<0
            L(i,1)=0+Trim;
        end
        L(i,2)=P_O(i)+0.25*(L(i,2)-Ltemp);
        SFlag(i)=0;
    end

    else
% 2 points evaluated
        Ltemp=L(i,1);
        L(i,1)=P_O(i)-(L(i,2)-L(i,1))*RF;
        if L(i,1)<0
% Prevent lower limit from going negative
            L(i,1)=0+Trim;
        end
        L(i,2)=P_O(i)+(L(i,2)-Ltemp)*RF;
        SFlag(i)=1;
    end

end

end

% Step 8: Advance iteration counter
k=k+1;
% advance k, the iteration number

end
% end of kth reduction loop
%
% Step 9: Output
%
% -----
% Plot Generation
% -----
%
out_fig1=strcat(dataset,'_LinearVEfit');    % assign file
name to the figure below

```



```

figure(1)
hold on
plot(t, Strain, 'b o', t, Strain_P, 'k-')
axis([0 max(t)*1.15 0 (max(Strain)*1.15)]);
text(ts*0.4, (max(Strain*0.25)), 'retardation');
text((ts*0.4+ts), (max(Strain*0.25)), 'recovery');
if WRFlag==1
    text((-
ts*0.2+ts), max(Strain*0.1), strcat('WMAR=', num2str(WR_O)));
else
    text((-
ts*0.2+ts), max(Strain*0.1), strcat('MAR=', num2str(WR_O)));
end
text((ts*0.5+ts), (max(Strain*0.85)), strcat('G1=', num2str(P_
O(1))));
text((ts*0.5+ts), (max(Strain*0.775)), strcat('\eta1=', num2st
r(P_O(2))));
text((ts*0.5+ts), (max(Strain*0.70)), strcat('G2=', num2str(P_
O(3))));
text((ts*0.5+ts), (max(Strain*0.625)), strcat('\eta2=', num2st
r(P_O(4))));
title(strcat('Creep Data, LVE Burger Model
:', upper(dataset)));
xlabel('time, t (s)');
ylabel('strain, \gamma');
legend('observed', 'Burger');
hold off
%
saveas(gcf, out_fig1, 'fig'); % save figure
%
% -----
% Screen Output
% -----
%
if WRFlag==1
    WRName='Weighted MAR';
else
    WRName='Standard MAR';
end
%
fprintf(1, '\n');
fprintf(1, '-----\n');

```

```

fprintf('Completed Linear Viscoelastic curve fit');
fprintf('Dataset %s was fit to a Burger model.\n',dataset);
fprintf('Iterations (k) = %g, Rel. Error in max Strain
(SError_O) = %4.4f\n', k-1, SError_O);
fprintf('Optimized Residual (WR_O) = %G, Residual Type =
%s\n',WR_O,WRName);
fprintf('Material parameters (P_O): %4.2f %4.2f %4.2f
%4.2f\n',P_O(1),P_O(2),P_O(3),P_O(4));
fprintf(1,'-----\n');
%
% -----
%
% File Output
% -----
%
close all;
out_run=strcat(dataset,'_LinearVEfit','.fit');
fid2=fopen(out_run,'w');
fprintf(fid2,'-----\n');
fprintf(fid2,'Completed Linear Viscoelastic curve fit');
fprintf(fid2,'Dataset %s was fit to a Burger
model.\n',dataset);
fprintf(fid2,'Iterations (k) = %g, Rel. Error in max Strain
(SError_O) = %4.4f\n', k-1, SError_O);
fprintf(fid2,'Optimized Residual (WR_O) = %G, Residual Type
= %s\n',WR_O,WRName);
fprintf(fid2,'Material parameters (P_O): [ %4.2f %4.2f
%4.2f %4.2f ]\n',P_O(1),P_O(2),P_O(3),P_O(4));
fprintf(fid2,'-----\n');
fclose(fid2);
%
% Thanks for coming to the show, folks! My name is Brett
Towler. I'll be in town all week.
%
LVEreturn=[P_O;WR_O];
%
% -----
%
% BURGER VISCOLASTIC MODEL CONSTITUTIVE EQUATION
% Brett Towler
% Apr 25, 2003

```

```
%
% -----
% -----
```

```
% Usage strain=Burger(s0,ts,t,p)
```

```
function strain=Burger(s0,ts,t,p)
```

```
% Burger model is Kelvin and Maxwell elements in series
```

```
strain_on=s0*((1/p(1))*(1-exp(-
t/(p(2)/p(1))))+(1/p(3))+(t/p(4)));
strain_off=-s0*((1/p(1))*(1-exp(-(t-
ts)/(p(2)/p(1))))+(1/p(3))+((t-ts)/p(4)));
```

```
if t>ts
```

```
    strain=strain_on+strain_off;
```

```
else
```

```
    strain=strain_on;
```

```
end
```

APPENDIX C

ANSYS INPUT FOR BURGER MATERIAL MODEL

FINISH  
/CLEAR

/COM,ANSYS RELEASE 7.1

/COM

\*\*\*\*\*

\*\*\*

/COM \*\*\* Burger Constitutive Model Validation

\*\*\*

/COM

\*\*\*\*\*

\*\*\*

! This model uses a Burger constitutive law with 182 elements to  
validate the  
! Prony series representation of a Burger linear viscoelastic  
constitutive model.

! Brett Towler  
! Civil Engineering Dept., MSU-Bozeman

```
!
!               N4-----N3
!               !         !
!               !         !
!               !         !
!               !         !
!               N1-----N2
!
!
```

/CONFIG,NRES,3650 ! max # of steps allowed is  
3650(1000 default)  
/title, Viscoelastic solution for a single Burger Element

/filename, burger\_182, 1 ! lowercase FILNAME works  
around ansb prompt

```
! Steps in the Structural Model
!   1.0 Model Parameters
!       1.1 User Input
!       1.2 Dependent Parameters
!   2.0 Structural Analysis Set-Up
!       2.1 Structural Nodes
!       2.2 Structural Element Type Defined
!       2.3 Structural Properties Defined
!       2.4 Structural Types, Constants & Properties Enabled
!       2.5 Structural Mesh Created
!       2.6 Structural Boundary Conditions
!   3.0 Solution
!       5.1 Define Monitor File DOFs
!       5.2 Solve model
!   4.0 Database Results Post-Processing
!       6.1 Set Standard View
```

```

!           6.2 Prony Parameter File Output
!           6.3 Deformed Centroid Calculation
!           6.4 Resultant Surface Force Calculation
!           6.5 Reaction Force Calculation
!           6.6 Resultant File Output
!       5.0 Time-History Post-Processing

```

```

!       UNITS:

```

```

!           The base unit for problems of this scale is millimeter
(mm).

```

```

!           L :           1 m           =           1000 mm           height,
length, width
!           L/S :         1 m/s          =           1000 mm/s          velocity
!           M/L^3 :        1 kg/m^3       =           10^-9 kg/mm^3       density
!           M/L-T :        1 kg/m-s       =           10^-3 kg/mm-s       dynamic
viscosity
!           F :           1 kg-m/s^2      =           1000 kg-mm/s^2      force,
load, reaction
!           F/L^2 :        1 kg/m-s^2     =           10^-3 kg/mm-s^2     pressure,
Young's modulus

```

```

/COM *** STEP 1.0 : MODEL PARAMETERS

```

```

*****

```

```

/COM *** STEP 1.1 : USER INPUT

```

```

*****

```

```

! User input of the 4 independant Burger parameters and material
Poisson's ratio

```

```

u_G1 = 3.54592427321466           ! shear modulus for Maxwell
(series) spring
u_h1 = 204.999089375703           ! viscous coeff. for Maxwell
(series) dashpot
u_G2 = 10.011076251886           ! shear modulus for Kelvin
(parallel) spring
u_h2 = 1499.99994345323           ! viscous coeff. for Kelvin
(parallel) dashpot
u_PR = 0.33                       ! Poisson's ratio

```

```

! User input of the constant strain e0

```

```

u_e0 = 0.01

```

```

/COM *** STEP 1.2 : DEPENDENT PARAMETERS

```

```

*****

```

```

! Intermediate parameters used in conversion to Prony series input for
Relaxation modulus

```

```

i_p0 = 1
i_p1 = u_h1/u_G1+u_h1/u_G2+u_h2/u_G2
i_p2 = u_h1*u_h2/(u_G1*u_G2)

```

```

i_q1 = u_h1
i_q2 = u_h1*u_h2/u_G2

i_A = (i_p1**2-4*i_p2)**(1/2)
i_r1 = (i_p1-i_A)/(2*i_p2)
i_r2 = (i_p1+i_A)/(2*i_p2)

i_ginf = 0                                ! Fixed constant for any
Burger model
i_g1 = (i_q1-i_q2*i_r1)/i_A                ! Equivalent 1st Maxwell
coeff.
i_g2 = -(i_q1-i_q2*i_r2)/i_A              ! Equivalent 2nd Maxwell
coeff.
i_g0 = i_ginf+i_g1+i_g2

! Prony series representation of the Burger parameters

s_E0 = i_g0*2*(1+u_PR)
s_PR = u_PR
s_A1 = i_g1/i_g0
s_TAU1 = 1/i_r1
s_A2 = i_g2/i_g0
s_TAU2 = 1/i_r2

! Additional parameters used in analytical solution
s_E1 = i_g1*2*(1+u_PR)
s_E2 = i_g2*2*(1+u_PR)

/COM *** STEP 2.0 : STRUCTURAL ANALYSIS SET-UP
*****

/PREP7

/COM *** STEP 2.1 : STRUCTURAL NODES
*****

N,1,0,0
N,2,1,0
N,3,1,1
N,4,0,1

/COM *** ELEMENT TYPE DEFINED ***
ET,1,182,,,3                                ! PLANE182

/COM *** REAL CONSTANTS TYPE DEFINED ***
R,1,.5,                                     ! Plane thickness (Real
Constants 1)

/COM *** MATERIAL PROPERTIES DEFINED ***
MP,EX,1,s_E0                                ! Youngs Modulus (Material 1)
MP,PRXY,1,s_PR                              ! Poisson ratio, major
(Material 1)
TB,PRONY,1,,2,SHEAR                          ! Creates a table of Prony
coefficients (Material 1)

```

```

TBDATA,1,s_A1,s_TAU1,s_A2,s_TAU2      ! Fills in the values of the
coeff.

/COM *** ELEMENT TYPES, CONSTANTS AND PROPERTIES ENABLED ***
TYPE,1                                ! Enable Element Type 1
MAT,1                                 ! Enable Material 1

/COM *** ELEMENTS CREATED ***
E,1,2,3,4                             ! node numbers listed CCW
order

/COM *** SOLUTION ***
FINISH
/SOLUTION
ANTYPE,STATIC,NEW                     ! static soln w/out restart

/COM *** BOUNDARY CONDITIONS ***
NSEL,S,LOC,Y,0                         ! select nodes at y = 0 (N1 &
N2)
D,ALL,UY                               ! fix UY at these nodes
NSEL,R,LOC,X,0                         ! select subset at x = 0 (N1)
D,ALL,UX                               ! fix UX at this node
NSEL,S,LOC,Y,1                         ! select nodes at y = 1 (N3 &
N4)
D,ALL,UY,u_e0                         ! Apply constant strain
NALL                                   !
TOFF,273                             ! Defines temperature offset
TREF,70                               ! Defines reference
temperature
TUNIF,70                             ! Assigns uniform temperature
AUTOTS,OFF                            ! automatic time-stepping is
turned off
KBC,1                                 ! specifies 'stepped' loading
TIME,180                             ! time for load step
NSUB,3600                             ! number of time substeps in
TIME
OUTRES,ALL,ALL                        ! output solution for ALL
variables at every 20th timestep

ALLSEL,ALL
SAVE

SOLVE                                ! FIRE IN THE HOLE!!!

FINISH

ALLSEL,ALL
SAVE

/POST1

*CFOPEN,prony_param.txt
*VWRITE,s_E0,s_PR

```



```

(2X,'E0:',F16.8,2X,'PR:',F16.8)
*VWRITE,s_A1,s_TAU1
(2X,'A1:',F16.8,2X,'Tau1:',F16.8)
*VWRITE,s_A2,s_TAU2
(2X,'A2:',F16.8,2X,'Tau2:',F16.8)
*VWRITE,u_G1,u_h1,u_G2,u_h2
(2X,'G1:',F16.8,2X,'h1:',F16.8,2X,'G2:',F16.8,2X,'h2:',F16.8)

FINISH

/POST26

ESOL,2,1,3,s,y                                ! store y-comp stress
of element 1, node 3 in variable 2

*CFOPEN,burger_182,txt                          ! open text file called
'burger_182.txt'
*DIM,BURGER,ARRAY,3600,6                      ! dimension 3600 x 6
array called 'BURGER'

VGET,BURGER(1,5),1                             ! Move variable 1
(time) into BURGER(1,5)
VGET,BURGER(1,6),1                             ! Move variable 1
(time) into BURGER(1,6)
*VOPER,BURGER(1,5),BURGER(1,5),MULT,-1/s_TAU1  ! BURGER(1,5) =
BURGER(1,5)*(-1/s_TAU1)
*VOPER,BURGER(1,6),BURGER(1,6),MULT,-1/s_TAU2  ! BURGER(1,6) =
BURGER(1,6)*(-1/s_TAU2)
*VFUN,BURGER(1,5),EXP,BURGER(1,5)              ! BURGER(1,5) =
EXP(BURGER(1,5))
*VFUN,BURGER(1,6),EXP,BURGER(1,6)              ! BURGER(1,6) =
EXP(BURGER(1,6))
*VOPER,BURGER(1,5),BURGER(1,5),MULT,s_E1        ! BURGER(1,5) = s_E1 *
BURGER(1,5)
*VOPER,BURGER(1,6),BURGER(1,6),MULT,s_E2        ! BURGER(1,6) = s_E2 *
BURGER(1,6)
*VOPER,BURGER(1,4),BURGER(1,5),ADD,BURGER(1,6)  ! BURGER(1,4) =
BURGER(1,5) + BURGER(1,6)
*VOPER,BURGER(1,3),BURGER(1,4),MULT,u_e0        ! BURGER(1,3) = u_e0 *
BURGER(1,4)
VGET,BURGER(1,2),2                             ! Move variable 1
(time) into BURGER(1,2)
VGET,BURGER(1,1),1                             ! Move variable 1
(time) into BURGER(1,1)

*VWRITE,BURGER(1,1),BURGER(1,2),BURGER(1,3),BURGER(1,4),BURGER(1,5),BUR
GER(1,6)
(F10.3,' ',3X,F18.10,' ',3X,F18.10,' ',3X,F18.10,' ',3X,F18.10,' ',3X,F
18.10)

FINISH

ALLSEL,ALL
SAVE

```

APPENDIX D

RESULTS OF BURGER MATERIAL MODEL VALIDATION

## Burger Model: ANSYS vs. Analytical Solution

ε0 1.00E-02

| t    | $\sigma(t)$ | G(t)* | $\sigma(t)$ | G(t)     | Prony <sub>1</sub> | Prony <sub>2</sub> | G AbErr  | G RelErr | G CumErr |
|------|-------------|-------|-------------|----------|--------------------|--------------------|----------|----------|----------|
| 0.05 | 0.09428     | 9.4   | 0.094229    | 9.422891 | 0.692774           | 8.730117           | 0.005158 | 0.05%    | 0.01     |
| 0.1  | 0.094198    | 9.4   | 0.094136    | 9.413633 | 0.692581           | 8.721051           | 0.006202 | 0.07%    | 0.01     |
| 0.15 | 0.094116    | 9.4   | 0.094044    | 9.404384 | 0.692389           | 8.711995           | 0.007244 | 0.08%    | 0.02     |
| 0.2  | 0.094034    | 9.4   | 0.093951    | 9.395145 | 0.692197           | 8.702948           | 0.008284 | 0.09%    | 0.03     |
| 0.25 | 0.093952    | 9.4   | 0.093859    | 9.385915 | 0.692004           | 8.693911           | 0.009322 | 0.10%    | 0.04     |
| 0.3  | 0.093871    | 9.4   | 0.093767    | 9.376695 | 0.691812           | 8.684882           | 0.010359 | 0.11%    | 0.05     |
| 0.35 | 0.093789    | 9.4   | 0.093675    | 9.367484 | 0.69162            | 8.675864           | 0.011393 | 0.12%    | 0.06     |
| 0.4  | 0.093707    | 9.4   | 0.093583    | 9.358282 | 0.691428           | 8.666854           | 0.012425 | 0.13%    | 0.07     |
| 0.45 | 0.093625    | 9.4   | 0.093491    | 9.34909  | 0.691236           | 8.657854           | 0.013456 | 0.14%    | 0.08     |
| 0.5  | 0.093544    | 9.4   | 0.093399    | 9.339907 | 0.691044           | 8.648863           | 0.014484 | 0.16%    | 0.10     |
| 0.55 | 0.093462    | 9.3   | 0.093307    | 9.330734 | 0.690852           | 8.639882           | 0.015510 | 0.17%    | 0.11     |
| 0.6  | 0.093381    | 9.3   | 0.093216    | 9.32157  | 0.69066            | 8.63091            | 0.016535 | 0.18%    | 0.13     |
| 0.65 | 0.0933      | 9.3   | 0.093124    | 9.312416 | 0.690468           | 8.621947           | 0.017557 | 0.19%    | 0.15     |
| 0.7  | 0.093218    | 9.3   | 0.093033    | 9.30327  | 0.690277           | 8.612994           | 0.018578 | 0.20%    | 0.17     |
| 0.75 | 0.093137    | 9.3   | 0.092941    | 9.294135 | 0.690085           | 8.60405            | 0.019596 | 0.21%    | 0.19     |
| 0.8  | 0.093056    | 9.3   | 0.09285     | 9.285008 | 0.689893           | 8.595115           | 0.020613 | 0.22%    | 0.21     |
| 0.85 | 0.092975    | 9.3   | 0.092759    | 9.275891 | 0.689702           | 8.586189           | 0.021628 | 0.23%    | 0.23     |
| 0.9  | 0.092894    | 9.3   | 0.092668    | 9.266783 | 0.68951            | 8.577273           | 0.022641 | 0.24%    | 0.25     |
| 0.95 | 0.092813    | 9.3   | 0.092577    | 9.257685 | 0.689319           | 8.568366           | 0.023651 | 0.26%    | 0.27     |
| 1    | 0.092733    | 9.3   | 0.092486    | 9.248595 | 0.689127           | 8.559468           | 0.024660 | 0.27%    | 0.30     |
| 1.05 | 0.092652    | 9.3   | 0.092395    | 9.239515 | 0.688936           | 8.55058            | 0.025667 | 0.28%    | 0.32     |
| 1.1  | 0.092571    | 9.3   | 0.092304    | 9.230445 | 0.688744           | 8.5417             | 0.026672 | 0.29%    | 0.35     |
| 1.15 | 0.092491    | 9.2   | 0.092214    | 9.221383 | 0.688553           | 8.53283            | 0.027675 | 0.30%    | 0.38     |
| 1.2  | 0.09241     | 9.2   | 0.092123    | 9.212331 | 0.688362           | 8.523969           | 0.028677 | 0.31%    | 0.41     |
| 1.25 | 0.09233     | 9.2   | 0.092033    | 9.203288 | 0.688171           | 8.515118           | 0.029676 | 0.32%    | 0.44     |
| 1.3  | 0.092249    | 9.2   | 0.091943    | 9.194255 | 0.687979           | 8.506275           | 0.030673 | 0.33%    | 0.47     |
| 1.35 | 0.092169    | 9.2   | 0.091852    | 9.18523  | 0.687788           | 8.497442           | 0.031669 | 0.34%    | 0.50     |
| 1.4  | 0.092089    | 9.2   | 0.091762    | 9.176215 | 0.687597           | 8.488618           | 0.032662 | 0.36%    | 0.53     |
| 1.45 | 0.092009    | 9.2   | 0.091672    | 9.167209 | 0.687406           | 8.479803           | 0.033654 | 0.37%    | 0.57     |
| 1.5  | 0.091929    | 9.2   | 0.091582    | 9.158212 | 0.687215           | 8.470997           | 0.034644 | 0.38%    | 0.60     |
| 1.55 | 0.091849    | 9.2   | 0.091492    | 9.149225 | 0.687025           | 8.4622             | 0.035631 | 0.39%    | 0.64     |
| 1.6  | 0.091769    | 9.2   | 0.091402    | 9.140247 | 0.686834           | 8.453413           | 0.036617 | 0.40%    | 0.67     |
| 1.65 | 0.091689    | 9.2   | 0.091313    | 9.131277 | 0.686643           | 8.444634           | 0.037601 | 0.41%    | 0.71     |
| 1.7  | 0.091609    | 9.2   | 0.091223    | 9.122317 | 0.686452           | 8.435865           | 0.038583 | 0.42%    | 0.75     |
| 1.75 | 0.091529    | 9.2   | 0.091134    | 9.113367 | 0.686262           | 8.427105           | 0.039564 | 0.43%    | 0.79     |
| 1.8  | 0.09145     | 9.1   | 0.091044    | 9.104425 | 0.686071           | 8.418354           | 0.040542 | 0.45%    | 0.83     |
| 1.85 | 0.09137     | 9.1   | 0.090955    | 9.095492 | 0.685881           | 8.409612           | 0.041518 | 0.46%    | 0.87     |
| 1.9  | 0.091291    | 9.1   | 0.090866    | 9.086569 | 0.68569            | 8.400879           | 0.042493 | 0.47%    | 0.91     |
| 1.95 | 0.091211    | 9.1   | 0.090777    | 9.077655 | 0.6855             | 8.392155           | 0.043466 | 0.48%    | 0.96     |
| 2    | 0.091132    | 9.1   | 0.090687    | 9.068749 | 0.685309           | 8.38344            | 0.044437 | 0.49%    | 1.00     |
| 2.05 | 0.091053    | 9.1   | 0.090599    | 9.059853 | 0.685119           | 8.374734           | 0.045406 | 0.50%    | 1.05     |
| 2.1  | 0.090973    | 9.1   | 0.09051     | 9.050966 | 0.684929           | 8.366038           | 0.046373 | 0.51%    | 1.09     |
| 2.15 | 0.090894    | 9.1   | 0.090421    | 9.042088 | 0.684738           | 8.35735            | 0.047338 | 0.52%    | 1.14     |
| 2.2  | 0.090815    | 9.1   | 0.090332    | 9.03322  | 0.684548           | 8.348671           | 0.048301 | 0.53%    | 1.19     |
| 2.25 | 0.090736    | 9.1   | 0.090244    | 9.02436  | 0.684358           | 8.340002           | 0.049263 | 0.55%    | 1.24     |
| 2.3  | 0.090657    | 9.1   | 0.090155    | 9.015509 | 0.684168           | 8.331341           | 0.050222 | 0.56%    | 1.29     |
| 2.35 | 0.090578    | 9.1   | 0.090067    | 9.006667 | 0.683978           | 8.322689           | 0.051180 | 0.57%    | 1.34     |
| 2.4  | 0.0905      | 9.0   | 0.089978    | 8.997835 | 0.683788           | 8.314047           | 0.052136 | 0.58%    | 1.39     |
| 2.45 | 0.090421    | 9.0   | 0.08989     | 8.989011 | 0.683598           | 8.305413           | 0.053090 | 0.59%    | 1.44     |
| 2.5  | 0.090342    | 9.0   | 0.089802    | 8.980197 | 0.683408           | 8.296788           | 0.054042 | 0.60%    | 1.50     |
| 2.55 | 0.090264    | 9.0   | 0.089714    | 8.971391 | 0.683218           | 8.288173           | 0.054992 | 0.61%    | 1.55     |
| 2.6  | 0.090185    | 9.0   | 0.089626    | 8.962594 | 0.683029           | 8.279566           | 0.055941 | 0.62%    | 1.61     |
| 2.65 | 0.090107    | 9.0   | 0.089538    | 8.953807 | 0.682839           | 8.270968           | 0.056888 | 0.64%    | 1.67     |

|        |          |     |          |          |          |          |          |        |         |
|--------|----------|-----|----------|----------|----------|----------|----------|--------|---------|
| 2.7    | 0.090029 | 9.0 | 0.08945  | 8.945028 | 0.682649 | 8.262379 | 0.057832 | 0.65%  | 1.72    |
| 2.75   | 0.08995  | 9.0 | 0.089363 | 8.936259 | 0.68246  | 8.253799 | 0.058775 | 0.66%  | 1.78    |
| 2.8    | 0.089872 | 9.0 | 0.089275 | 8.927498 | 0.68227  | 8.245228 | 0.059716 | 0.67%  | 1.84    |
| 2.85   | 0.089794 | 9.0 | 0.089187 | 8.918746 | 0.682081 | 8.236666 | 0.060656 | 0.68%  | 1.90    |
| 2.9    | 0.089716 | 9.0 | 0.0891   | 8.910003 | 0.681891 | 8.228112 | 0.061593 | 0.69%  | 1.97    |
| 2.95   | 0.089638 | 9.0 | 0.089013 | 8.90127  | 0.681702 | 8.219568 | 0.062529 | 0.70%  | 2.03    |
| 3      | 0.08956  | 9.0 | 0.088925 | 8.892545 | 0.681512 | 8.211032 | 0.063463 | 0.71%  | 2.09    |
| 3.05   | 0.089482 | 8.9 | 0.088838 | 8.883829 | 0.681323 | 8.202505 | 0.064395 | 0.72%  | 2.16    |
| 3.1    | 0.089404 | 8.9 | 0.088751 | 8.875122 | 0.681134 | 8.193988 | 0.065325 | 0.74%  | 2.22    |
| 3.15   | 0.089327 | 8.9 | 0.088664 | 8.866423 | 0.680945 | 8.185479 | 0.066253 | 0.75%  | 2.29    |
| 3.2    | 0.089249 | 8.9 | 0.088577 | 8.857734 | 0.680756 | 8.176978 | 0.067180 | 0.76%  | 2.35    |
| 3.25   | 0.089172 | 8.9 | 0.088491 | 8.849054 | 0.680567 | 8.168487 | 0.068105 | 0.77%  | 2.42    |
| 3.3    | 0.089094 | 8.9 | 0.088404 | 8.840382 | 0.680378 | 8.160005 | 0.069027 | 0.78%  | 2.49    |
| 3.35   | 0.089017 | 8.9 | 0.088317 | 8.831719 | 0.680189 | 8.151531 | 0.069949 | 0.79%  | 2.56    |
| 3.4    | 0.088939 | 8.9 | 0.088231 | 8.823066 | 0.68     | 8.143066 | 0.070868 | 0.80%  | 2.63    |
| 3.45   | 0.088862 | 8.9 | 0.088144 | 8.814421 | 0.679811 | 8.13461  | 0.071785 | 0.81%  | 2.70    |
| 3.5    | 0.088785 | 8.9 | 0.088058 | 8.805784 | 0.679622 | 8.126162 | 0.072701 | 0.83%  | 2.78    |
| 3.55   | 0.088708 | 8.9 | 0.087972 | 8.797157 | 0.679433 | 8.117724 | 0.073615 | 0.84%  | 2.85    |
| 3.6    | 0.088631 | 8.9 | 0.087885 | 8.788539 | 0.679245 | 8.109294 | 0.074527 | 0.85%  | 2.92    |
| 3.65   | 0.088554 | 8.9 | 0.087799 | 8.779929 | 0.679056 | 8.100873 | 0.075438 | 0.86%  | 3.00    |
| 3.7    | 0.088477 | 8.8 | 0.087713 | 8.771328 | 0.678867 | 8.092461 | 0.076346 | 0.87%  | 3.08    |
| 3.75   | 0.0884   | 8.8 | 0.087627 | 8.762736 | 0.678679 | 8.084057 | 0.077253 | 0.88%  | 3.15    |
| 3.8    | 0.088323 | 8.8 | 0.087542 | 8.754152 | 0.67849  | 8.075662 | 0.078158 | 0.89%  | 3.23    |
| 3.85   | 0.088246 | 8.8 | 0.087456 | 8.745578 | 0.678302 | 8.067276 | 0.079061 | 0.90%  | 3.31    |
| 3.9    | 0.08817  | 8.8 | 0.08737  | 8.737012 | 0.678113 | 8.058899 | 0.079963 | 0.92%  | 3.39    |
| 3.95   | 0.088093 | 8.8 | 0.087285 | 8.728455 | 0.677925 | 8.05053  | 0.080862 | 0.93%  | 3.47    |
| 4      | 0.088017 | 8.8 | 0.087199 | 8.719907 | 0.677737 | 8.04217  | 0.081760 | 0.94%  | 3.55    |
| 4.05   | 0.08794  | 8.8 | 0.087114 | 8.711367 | 0.677549 | 8.033818 | 0.082656 | 0.95%  | 3.64    |
| 4.1    | 0.087864 | 8.8 | 0.087028 | 8.702836 | 0.67736  | 8.025476 | 0.083551 | 0.96%  | 3.72    |
| 4.15   | 0.087788 | 8.8 | 0.086943 | 8.694314 | 0.677172 | 8.017142 | 0.084443 | 0.97%  | 3.80    |
| 4.2    | 0.087711 | 8.8 | 0.086858 | 8.685801 | 0.676984 | 8.008816 | 0.085334 | 0.98%  | 3.89    |
| 4.25   | 0.087635 | 8.8 | 0.086773 | 8.677296 | 0.676796 | 8.0005   | 0.086223 | 0.99%  | 3.98    |
| 4.3    | 0.087559 | 8.8 | 0.086688 | 8.6688   | 0.676608 | 7.992192 | 0.087110 | 1.00%  | 4.06    |
| 4.35   | 0.087483 | 8.7 | 0.086603 | 8.660312 | 0.67642  | 7.983892 | 0.087996 | 1.02%  | 4.15    |
| 4.4    | 0.087407 | 8.7 | 0.086518 | 8.651834 | 0.676232 | 7.975601 | 0.088880 | 1.03%  | 4.24    |
| 4.45   | 0.087331 | 8.7 | 0.086434 | 8.643364 | 0.676045 | 7.967319 | 0.089762 | 1.04%  | 4.33    |
| 4.5    | 0.087255 | 8.7 | 0.086349 | 8.634902 | 0.675857 | 7.959045 | 0.090642 | 1.05%  | 4.42    |
| 4.55   | 0.08718  | 8.7 | 0.086264 | 8.626449 | 0.675669 | 7.95078  | 0.091521 | 1.06%  | 4.51    |
| 4.6    | 0.087104 | 8.7 | 0.08618  | 8.618005 | 0.675481 | 7.942524 | 0.092398 | 1.07%  | 4.60    |
| 4.65   | 0.087028 | 8.7 | 0.086096 | 8.60957  | 0.675294 | 7.934276 | 0.093273 | 1.08%  | 4.70    |
| 4.7    | 0.086953 | 8.7 | 0.086011 | 8.601143 | 0.675106 | 7.926037 | 0.094146 | 1.09%  | 4.79    |
| 4.75   | 0.086877 | 8.7 | 0.085927 | 8.592725 | 0.674919 | 7.917806 | 0.095018 | 1.11%  | 4.89    |
| 4.8    | 0.086802 | 8.7 | 0.085843 | 8.584315 | 0.674731 | 7.909584 | 0.095888 | 1.12%  | 4.98    |
| 4.85   | 0.086727 | 8.7 | 0.085759 | 8.575914 | 0.674544 | 7.90137  | 0.096756 | 1.13%  | 5.08    |
| 4.9    | 0.086651 | 8.7 | 0.085675 | 8.567521 | 0.674357 | 7.893165 | 0.097622 | 1.14%  | 5.18    |
| 4.95   | 0.086576 | 8.7 | 0.085591 | 8.559138 | 0.674169 | 7.884968 | 0.098487 | 1.15%  | 5.28    |
| 5      | 0.086501 | 8.7 | 0.085508 | 8.550762 | 0.673982 | 7.87678  | 0.099350 | 1.16%  | 5.37    |
| 175    | 0.006535 | 0.7 | 0.004923 | 0.492349 | 0.262109 | 0.23024  | 0.161119 | 32.72% | 1036.36 |
| 175.05 | 0.006531 | 0.7 | 0.00492  | 0.492037 | 0.262036 | 0.230001 | 0.161034 | 32.73% | 1036.53 |
| 175.1  | 0.006527 | 0.7 | 0.004917 | 0.491725 | 0.261963 | 0.229762 | 0.160949 | 32.73% | 1036.69 |
| 175.15 | 0.006523 | 0.7 | 0.004914 | 0.491414 | 0.26189  | 0.229524 | 0.160865 | 32.74% | 1036.85 |
| 175.2  | 0.006519 | 0.7 | 0.004911 | 0.491103 | 0.261818 | 0.229285 | 0.160780 | 32.74% | 1037.01 |
| 175.25 | 0.006515 | 0.7 | 0.004908 | 0.490792 | 0.261745 | 0.229047 | 0.160696 | 32.74% | 1037.17 |
| 175.3  | 0.006511 | 0.7 | 0.004905 | 0.490482 | 0.261672 | 0.228809 | 0.160611 | 32.75% | 1037.33 |
| 175.35 | 0.006507 | 0.7 | 0.004902 | 0.490171 | 0.261599 | 0.228572 | 0.160527 | 32.75% | 1037.49 |
| 175.4  | 0.006503 | 0.7 | 0.004899 | 0.489861 | 0.261527 | 0.228334 | 0.160442 | 32.75% | 1037.65 |
| 175.45 | 0.006499 | 0.6 | 0.004896 | 0.489551 | 0.261454 | 0.228097 | 0.160358 | 32.76% | 1037.81 |

|        |          |     |          |          |          |          |          |        |         |
|--------|----------|-----|----------|----------|----------|----------|----------|--------|---------|
| 175.5  | 0.006495 | 0.6 | 0.004892 | 0.489242 | 0.261382 | 0.22786  | 0.160274 | 32.76% | 1037.97 |
| 175.55 | 0.006491 | 0.6 | 0.004889 | 0.488933 | 0.261309 | 0.227624 | 0.160189 | 32.76% | 1038.13 |
| 175.6  | 0.006487 | 0.6 | 0.004886 | 0.488624 | 0.261236 | 0.227387 | 0.160105 | 32.77% | 1038.29 |
| 175.65 | 0.006483 | 0.6 | 0.004883 | 0.488315 | 0.261164 | 0.227151 | 0.160021 | 32.77% | 1038.45 |
| 175.7  | 0.006479 | 0.6 | 0.00488  | 0.488007 | 0.261091 | 0.226915 | 0.159937 | 32.77% | 1038.61 |
| 175.75 | 0.006476 | 0.6 | 0.004877 | 0.487699 | 0.261019 | 0.22668  | 0.159853 | 32.78% | 1038.77 |
| 175.8  | 0.006472 | 0.6 | 0.004874 | 0.487391 | 0.260946 | 0.226444 | 0.159769 | 32.78% | 1038.93 |
| 175.85 | 0.006468 | 0.6 | 0.004871 | 0.487083 | 0.260874 | 0.226209 | 0.159685 | 32.78% | 1039.09 |
| 175.9  | 0.006464 | 0.6 | 0.004868 | 0.486776 | 0.260801 | 0.225974 | 0.159601 | 32.79% | 1039.25 |
| 175.95 | 0.00646  | 0.6 | 0.004865 | 0.486469 | 0.260729 | 0.22574  | 0.159517 | 32.79% | 1039.41 |
| 176    | 0.006456 | 0.6 | 0.004862 | 0.486162 | 0.260657 | 0.225505 | 0.159433 | 32.79% | 1039.57 |
| 176.05 | 0.006452 | 0.6 | 0.004859 | 0.485855 | 0.260584 | 0.225271 | 0.159349 | 32.80% | 1039.73 |
| 176.1  | 0.006448 | 0.6 | 0.004855 | 0.485549 | 0.260512 | 0.225037 | 0.159265 | 32.80% | 1039.89 |
| 176.15 | 0.006444 | 0.6 | 0.004852 | 0.485243 | 0.260439 | 0.224803 | 0.159181 | 32.80% | 1040.05 |
| 176.2  | 0.00644  | 0.6 | 0.004849 | 0.484937 | 0.260367 | 0.22457  | 0.159098 | 32.81% | 1040.21 |
| 176.25 | 0.006436 | 0.6 | 0.004846 | 0.484632 | 0.260295 | 0.224337 | 0.159014 | 32.81% | 1040.36 |
| 176.3  | 0.006433 | 0.6 | 0.004843 | 0.484326 | 0.260222 | 0.224104 | 0.158930 | 32.81% | 1040.52 |
| 176.35 | 0.006429 | 0.6 | 0.00484  | 0.484021 | 0.26015  | 0.223871 | 0.158847 | 32.82% | 1040.68 |
| 176.4  | 0.006425 | 0.6 | 0.004837 | 0.483717 | 0.260078 | 0.223639 | 0.158763 | 32.82% | 1040.84 |
| 176.45 | 0.006421 | 0.6 | 0.004834 | 0.483412 | 0.260006 | 0.223406 | 0.158680 | 32.82% | 1041.00 |
| 176.5  | 0.006417 | 0.6 | 0.004831 | 0.483108 | 0.259933 | 0.223174 | 0.158596 | 32.83% | 1041.16 |
| 176.55 | 0.006413 | 0.6 | 0.004828 | 0.482804 | 0.259861 | 0.222943 | 0.158513 | 32.83% | 1041.32 |
| 176.6  | 0.006409 | 0.6 | 0.004825 | 0.4825   | 0.259789 | 0.222711 | 0.158429 | 32.84% | 1041.48 |
| 176.65 | 0.006405 | 0.6 | 0.004822 | 0.482197 | 0.259717 | 0.22248  | 0.158346 | 32.84% | 1041.63 |
| 176.7  | 0.006402 | 0.6 | 0.004819 | 0.481894 | 0.259645 | 0.222249 | 0.158263 | 32.84% | 1041.79 |
| 176.75 | 0.006398 | 0.6 | 0.004816 | 0.481591 | 0.259573 | 0.222018 | 0.158179 | 32.85% | 1041.95 |
| 176.8  | 0.006394 | 0.6 | 0.004813 | 0.481288 | 0.259501 | 0.221787 | 0.158096 | 32.85% | 1042.11 |
| 176.85 | 0.00639  | 0.6 | 0.00481  | 0.480986 | 0.259429 | 0.221557 | 0.158013 | 32.85% | 1042.27 |
| 176.9  | 0.006386 | 0.6 | 0.004807 | 0.480684 | 0.259356 | 0.221327 | 0.157930 | 32.86% | 1042.42 |
| 176.95 | 0.006382 | 0.6 | 0.004804 | 0.480382 | 0.259284 | 0.221097 | 0.157847 | 32.86% | 1042.58 |
| 177    | 0.006378 | 0.6 | 0.004801 | 0.48008  | 0.259212 | 0.220868 | 0.157764 | 32.86% | 1042.74 |
| 177.05 | 0.006375 | 0.6 | 0.004798 | 0.479779 | 0.25914  | 0.220638 | 0.157681 | 32.87% | 1042.90 |
| 177.1  | 0.006371 | 0.6 | 0.004795 | 0.479478 | 0.259068 | 0.220409 | 0.157598 | 32.87% | 1043.06 |
| 177.15 | 0.006367 | 0.6 | 0.004792 | 0.479177 | 0.258997 | 0.22018  | 0.157515 | 32.87% | 1043.21 |
| 177.2  | 0.006363 | 0.6 | 0.004789 | 0.478876 | 0.258925 | 0.219952 | 0.157432 | 32.88% | 1043.37 |
| 177.25 | 0.006359 | 0.6 | 0.004786 | 0.478576 | 0.258853 | 0.219723 | 0.157349 | 32.88% | 1043.53 |
| 177.3  | 0.006355 | 0.6 | 0.004783 | 0.478276 | 0.258781 | 0.219495 | 0.157266 | 32.88% | 1043.68 |
| 177.35 | 0.006352 | 0.6 | 0.00478  | 0.477976 | 0.258709 | 0.219267 | 0.157183 | 32.89% | 1043.84 |
| 177.4  | 0.006348 | 0.6 | 0.004777 | 0.477676 | 0.258637 | 0.219039 | 0.157101 | 32.89% | 1044.00 |
| 177.45 | 0.006344 | 0.6 | 0.004774 | 0.477377 | 0.258565 | 0.218812 | 0.157018 | 32.89% | 1044.16 |
| 177.5  | 0.00634  | 0.6 | 0.004771 | 0.477078 | 0.258493 | 0.218585 | 0.156935 | 32.90% | 1044.31 |
| 177.55 | 0.006336 | 0.6 | 0.004768 | 0.476779 | 0.258422 | 0.218358 | 0.156853 | 32.90% | 1044.47 |
| 177.6  | 0.006333 | 0.6 | 0.004765 | 0.476481 | 0.25835  | 0.218131 | 0.156770 | 32.90% | 1044.63 |
| 177.65 | 0.006329 | 0.6 | 0.004762 | 0.476183 | 0.258278 | 0.217905 | 0.156688 | 32.90% | 1044.78 |
| 177.7  | 0.006325 | 0.6 | 0.004759 | 0.475885 | 0.258206 | 0.217678 | 0.156605 | 32.91% | 1044.94 |
| 177.75 | 0.006321 | 0.6 | 0.004756 | 0.475587 | 0.258135 | 0.217452 | 0.156523 | 32.91% | 1045.10 |
| 177.8  | 0.006317 | 0.6 | 0.004753 | 0.475289 | 0.258063 | 0.217226 | 0.156440 | 32.91% | 1045.25 |
| 177.85 | 0.006314 | 0.6 | 0.00475  | 0.474992 | 0.257991 | 0.217001 | 0.156358 | 32.92% | 1045.41 |
| 177.9  | 0.00631  | 0.6 | 0.004747 | 0.474695 | 0.25792  | 0.216775 | 0.156276 | 32.92% | 1045.57 |
| 177.95 | 0.006306 | 0.6 | 0.004744 | 0.474398 | 0.257848 | 0.21655  | 0.156193 | 32.92% | 1045.72 |
| 178    | 0.006302 | 0.6 | 0.004741 | 0.474102 | 0.257776 | 0.216325 | 0.156111 | 32.93% | 1045.88 |
| 178.05 | 0.006298 | 0.6 | 0.004738 | 0.473806 | 0.257705 | 0.216101 | 0.156029 | 32.93% | 1046.03 |
| 178.1  | 0.006295 | 0.6 | 0.004735 | 0.47351  | 0.257633 | 0.215876 | 0.155947 | 32.93% | 1046.19 |
| 178.15 | 0.006291 | 0.6 | 0.004732 | 0.473214 | 0.257562 | 0.215652 | 0.155865 | 32.94% | 1046.35 |
| 178.2  | 0.006287 | 0.6 | 0.004729 | 0.472918 | 0.25749  | 0.215428 | 0.155783 | 32.94% | 1046.50 |
| 178.25 | 0.006283 | 0.6 | 0.004726 | 0.472623 | 0.257419 | 0.215205 | 0.155701 | 32.94% | 1046.66 |
| 178.3  | 0.006279 | 0.6 | 0.004723 | 0.472328 | 0.257347 | 0.214981 | 0.155619 | 32.95% | 1046.81 |
| 178.35 | 0.006276 | 0.6 | 0.00472  | 0.472033 | 0.257276 | 0.214758 | 0.155537 | 32.95% | 1046.97 |

|        |          |     |          |          |          |          |          |        |         |
|--------|----------|-----|----------|----------|----------|----------|----------|--------|---------|
| 178.4  | 0.006272 | 0.6 | 0.004717 | 0.471739 | 0.257204 | 0.214535 | 0.155455 | 32.95% | 1047.12 |
| 178.45 | 0.006268 | 0.6 | 0.004714 | 0.471445 | 0.257133 | 0.214312 | 0.155373 | 32.96% | 1047.28 |
| 178.5  | 0.006264 | 0.6 | 0.004712 | 0.471151 | 0.257061 | 0.21409  | 0.155291 | 32.96% | 1047.43 |
| 178.55 | 0.006261 | 0.6 | 0.004709 | 0.470857 | 0.25699  | 0.213867 | 0.155210 | 32.96% | 1047.59 |
| 178.6  | 0.006257 | 0.6 | 0.004706 | 0.470564 | 0.256919 | 0.213645 | 0.155128 | 32.97% | 1047.74 |
| 178.65 | 0.006253 | 0.6 | 0.004703 | 0.47027  | 0.256847 | 0.213423 | 0.155046 | 32.97% | 1047.90 |
| 178.7  | 0.006249 | 0.6 | 0.0047   | 0.469977 | 0.256776 | 0.213202 | 0.154965 | 32.97% | 1048.05 |
| 178.75 | 0.006246 | 0.6 | 0.004697 | 0.469685 | 0.256705 | 0.21298  | 0.154883 | 32.98% | 1048.21 |
| 178.8  | 0.006242 | 0.6 | 0.004694 | 0.469392 | 0.256633 | 0.212759 | 0.154801 | 32.98% | 1048.36 |
| 178.85 | 0.006238 | 0.6 | 0.004691 | 0.4691   | 0.256562 | 0.212538 | 0.154720 | 32.98% | 1048.52 |
| 178.9  | 0.006234 | 0.6 | 0.004688 | 0.468808 | 0.256491 | 0.212317 | 0.154638 | 32.99% | 1048.67 |
| 178.95 | 0.006231 | 0.6 | 0.004685 | 0.468516 | 0.256419 | 0.212097 | 0.154557 | 32.99% | 1048.83 |
| 179    | 0.006227 | 0.6 | 0.004682 | 0.468225 | 0.256348 | 0.211877 | 0.154476 | 32.99% | 1048.98 |
| 179.05 | 0.006223 | 0.6 | 0.004679 | 0.467934 | 0.256277 | 0.211657 | 0.154394 | 32.99% | 1049.14 |
| 179.1  | 0.00622  | 0.6 | 0.004676 | 0.467643 | 0.256206 | 0.211437 | 0.154313 | 33.00% | 1049.29 |
| 179.15 | 0.006216 | 0.6 | 0.004674 | 0.467352 | 0.256135 | 0.211217 | 0.154232 | 33.00% | 1049.45 |
| 179.2  | 0.006212 | 0.6 | 0.004671 | 0.467062 | 0.256064 | 0.210998 | 0.154150 | 33.00% | 1049.60 |
| 179.25 | 0.006208 | 0.6 | 0.004668 | 0.466771 | 0.255992 | 0.210779 | 0.154069 | 33.01% | 1049.75 |
| 179.3  | 0.006205 | 0.6 | 0.004665 | 0.466481 | 0.255921 | 0.21056  | 0.153988 | 33.01% | 1049.91 |
| 179.35 | 0.006201 | 0.6 | 0.004662 | 0.466192 | 0.25585  | 0.210341 | 0.153907 | 33.01% | 1050.06 |
| 179.4  | 0.006197 | 0.6 | 0.004659 | 0.465902 | 0.255779 | 0.210123 | 0.153826 | 33.02% | 1050.22 |
| 179.45 | 0.006194 | 0.6 | 0.004656 | 0.465613 | 0.255708 | 0.209905 | 0.153745 | 33.02% | 1050.37 |
| 179.5  | 0.00619  | 0.6 | 0.004653 | 0.465324 | 0.255637 | 0.209687 | 0.153664 | 33.02% | 1050.52 |
| 179.55 | 0.006186 | 0.6 | 0.00465  | 0.465035 | 0.255566 | 0.209469 | 0.153583 | 33.03% | 1050.68 |
| 179.6  | 0.006182 | 0.6 | 0.004647 | 0.464747 | 0.255495 | 0.209251 | 0.153502 | 33.03% | 1050.83 |
| 179.65 | 0.006179 | 0.6 | 0.004645 | 0.464458 | 0.255424 | 0.209034 | 0.153421 | 33.03% | 1050.98 |
| 179.7  | 0.006175 | 0.6 | 0.004642 | 0.46417  | 0.255353 | 0.208817 | 0.153340 | 33.04% | 1051.14 |
| 179.75 | 0.006171 | 0.6 | 0.004639 | 0.463883 | 0.255282 | 0.2086   | 0.153260 | 33.04% | 1051.29 |
| 179.8  | 0.006168 | 0.6 | 0.004636 | 0.463595 | 0.255211 | 0.208384 | 0.153179 | 33.04% | 1051.44 |
| 179.85 | 0.006164 | 0.6 | 0.004633 | 0.463308 | 0.255141 | 0.208167 | 0.153098 | 33.04% | 1051.60 |
| 179.9  | 0.00616  | 0.6 | 0.00463  | 0.463021 | 0.25507  | 0.207951 | 0.153018 | 33.05% | 1051.75 |
| 179.95 | 0.006157 | 0.6 | 0.004627 | 0.462734 | 0.254999 | 0.207735 | 0.152937 | 33.05% | 1051.90 |
| 180    | 0.006153 | 0.6 | 0.004624 | 0.462447 | 0.254928 | 0.207519 | 0.152856 | 33.05% | 1052.06 |

APPENDIX E

ANSYS INPUT FOR SHEAR CUBE PROBLEM

```

finish
/clear

/COM,ANSYS RELEASE 7.1
/COM *****
/COM ***          FSI Burger Reponse (Sequential Weak Coupling)          ***
/COM *****

!  MODEL NAME
!  -----
!  -----
m_series = 'test_case'
m_number = '01'

!  MODEL NOTES
!  -----
!  The burger response series are used to demonstrate the LVE response of
!  a shear cube in an FSI environment.
!
!  Parent: turb series
!  -----
!
!
!  Brett Towler
!  Civil Engineering Dept., MSU-Bozeman
!
!
!      21-----22-----23-----24-----25-----26
!      !           !           !           !           !           !
!      !           !           !           !           !           !
!      !           !           !           !           !           !
!      con  !      A5      !  A6  !  A7  !  A8  !      A9      !
!      !           !           !           !           !           !
!      INLET--> !           !           !           !           !           !
!      !           !           !           !           !           !
!      !           !           !           !           !           !
!      !           !           !           !           !           !
!      11-----12-----13-----14-----15-----16
!      !           !           !           !           !           !
!      che  !      A1      !  A2  !           !  A3  !      A4      !
!      !           !           !           !           !           !
!      !           !           !           !           !           !
!      1-----2-----3-----4-----5-----6
!
!      dev          for      cle      aft          wit
!
!
!  build the jobname string parameter from the user's definition above
m_job = STRCAT(m_series, '_')
m_job = STRCAT(m_job, m_number)

/filename, %m_job%, 1          ! lowercase FILNAME works around ansb prompt

!  Steps in the SWC FSI model
!      1.0 Model Parameters
!          1.1 User Input
!          1.2 Define Dependent Parameters
!      2.0 Fluid Anaysis Set-Up
!          2.1 Fluid Keypoints & Lines
!          2.2 Fluid Areas
!          2.3 Fluid Nodes
!          2.4 Fluid Element Type Defined
!          2.5 Fluid Types, Constants & Properties Enabled

```



```

!           2.6 Fluid Mesh Created
!           2.7 Fluid Boundary Conditions
!           2.8 Fluid Properties Defined
!           2.9 Fluid Solution Control
!       3.0 Structural Analysis Set-Up
!           3.1 Structural Keypoints & Lines
!           3.2 Structural Areas
!           3.3 Structural Nodes
!           3.4 Structural Element Type Defined
!           3.5 Structural Properties Defined
!           3.6 Structural Types, Constants & Properties Enabled
!           3.7 Structural Mesh Created
!           3.8 Structural Boundary Conditions
!           3.9 Structural Solution Control
!       4.0 Fluid Structure Interaction
!           4.1 Flag Fluid Structural Interfaces
!           4.2 Fluid Structural Interaction Solution Options
!       5.0 Solution
!           5.1 Define Monitor File DOFs
!           5.2 Solve model
!       6.0 Database Results Post-Processing
!           6.1 Set Standard View
!           6.2 Prony Parameter File Output
!       7.0 Time-History Post-Processing

```

```

/COM *** USER-DEFINED PARAMETERS ***

```

```

!       UNITS:
!           The base unit for problems of this scale is millimeter (mm).
!           L :           1 m           =           1000 mm           height,
length, width
!           L/S :           1 m/s           =           1000 mm/s           velocity
!           M/L^3 :           1 kg/m^3           =           10^-9 kg/mm^3 density
!           M/L-T :           1 kg/m-s           =           10^-3 kg/mm-s dynamic viscosity
!           F :           1 kg-m/s^2           =           1000 kg-mm/s^2           force, load,
reaction
!           F/L^2 :           1 kg/m-s^2           =           10^-3 kg/mm-s^2           pressure,
Young's modulus

```

```

/COM *** STEP 1.0 : MODEL PARAMETERS

```

```

*****

```

```

/COM *** STEP 1.1 : USER-DEFINED PARAMETERS

```

```

*****

```

```

!
! -----
! -----
! Start of User-Defined Parameters
! -----
! -----

```

```

! User input: GEOMETRIC PARAMETERS (g)
g_dev = 10           ! flow development (dev) length, mm
g_for = 1           ! fore (for) cube length, mm
g_cle = 0.2         ! cube length (cle), mm
g_aft = 1           ! aft (aft) cube length, mm
g_wit = 4           ! flow withdrawal (wit) length, mm
g_che = 0.2         ! cube height (che), mm
g_con = 3-g_che     ! flow constriction (con) height, mm

```

```

! Note: keep the ratio of g_cle to c_che close to 1.0.

! User input: MESH CONTROL (n)
n_che = 4                ! number of elements along cube height (fluid
side), ND
n_solid = 8              ! number of elements along cube height (solid
side), ND

! User input: FLUID PARAMETERS (f)
f_vx1 = 1000.0           ! step 1 Vx at inlet, mm/s
f_vy1 = 0.0              ! step 1 Vy at inlet, mm/s
f_ro = 0.000001          ! density of water, kg/mm^3
f_nu = 0.00000000000018 ! viscosity of water, N/s/mm^2

! User input: SOLID PARAMETERS (s)
s_G1 = 250               ! shear modulus for Maxwell (series) spring,
N/mm^2
s_h1 = 750               ! viscous coeff. for Maxwell (series) dashpot,
N/s/mm^2
s_G2 = 500               ! shear modulus for Kelvin (parallel) spring,
N/mm^2
s_h2 = 5000              ! viscous coeff. for Kelvin (parallel)
dashpot, N/s/mm^2
s_PR = 0.33              ! Poisson's ratio
s_DE = 0.000001          ! mass density of biofilm

! User input: SOLUTION CONTROL (c)
c_time1 = 65.0           ! time at end of step 1
c_ltime1 = 5.0           ! time to apply load (i.e. velocities at
inlet) in step 1
c_dtFSI = 0.25           ! time step for FSI loop (multiple of c_time)
c_dtS = 0.125            ! time step for SOLID solution loop (multiple of
c_dtFSI)
c_dtF = 0.125            ! time step for FLUID solution loop (multiple of
c_dtFSI)
c_iF = 200               ! global iterations for fluid solution
c_iFSI = 30              ! stagger iterations in FSI loop
c_ALE = 1                ! SWITCH - Mesh-morphing switch. 0 = OFF, 1 =
ON
c_SOLV = 1               ! SWITCH - Solve?. 0 = PREP only, 1 = SOLU
c_TURB = 1               ! SWITCH - Activate turbulence equations?. 0
= laminar only, 1 = turbulent flow

! User input: OPTIMIZATION CONTROL (o)
o_node = 145             ! node number used for target displacement
o_TGT = ((5/4)*g_cle)**2+((1/4)*g_che)**2 ! Target deflection, mm
o_OPT = 0                ! SWITCH - Optimization?. 0 = OFF, 1 = ON

! -----
! End of User-Defined Parameters
! -----

/COM *** STEP 1.2 : DEPENDENT PARAMETERS
*****

! WARNING: Do not change these parameters unless you are SURE you know what
! you are doing!!!

```

```

! Node/element distribution parameters
n_Rdev = -3.0           ! spacing ratio for elements in dev (2-way bias)
n_Rwit = 3.0           ! spacing ratio for elements in wit (1-way bias)
n_Rcon = 3.0           ! spacing ratio for elements in con (1-way bias)
n_ele = g_che/n_che    ! nominal element length based on n_che and g_che
above
n_for = g_for/n_ele     ! number of elements in for region
n_for = NINT(n_for)     ! round n_for to an even integer
n_aft = g_aft/n_ele     ! number of elements in aft region
n_aft = NINT(n_aft)     ! round n_aft to an even integer
n_dev = NINT(3*g_dev/(n_ele*(2+ABS(n_Rdev)))) ! number of elements in dev
region
n_wit = NINT(2*g_wit/(n_ele*(1+n_Rwit)))      ! number of elements in
wit region
n_con = NINT(2*g_con/(n_ele*(1+n_Rcon)))      ! number of elements in
con region

! Intermediate parameters used in conversion to Prony series input for
Relaxation modulus
i_p0 = 1
i_p1 = s_h1/s_G1+s_h1/s_G2+s_h2/s_G2
i_p2 = s_h1*s_h2/(s_G1*s_G2)
i_q1 = s_h1
i_q2 = s_h1*s_h2/s_G2

i_A = (i_p1**2-4*i_p2)**(1/2)
i_r1 = (i_p1-i_A)/(2*i_p2)
i_r2 = (i_p1+i_A)/(2*i_p2)

i_ginf = 0           ! Fixed constant for any Burger
model
i_g1 = (i_q1-i_q2*i_r1)/i_A           ! Equivalent 1st Maxwell shear
coeff.
i_g2 = -(i_q1-i_q2*i_r2)/i_A         ! Equivalent 2nd Maxwell shear
coeff.
i_g0 = i_ginf+i_g1+i_g2

! Prony series representation of the Burger parameters
s_E0 = i_g0*2*(1+s_PR)
s_A1 = i_g1/i_g0
s_TAU1 = 1/i_r1           ! Equivalent 1st Maxwell
relaxation time
s_A2 = i_g2/i_g0
s_TAU2 = 1/i_r2           ! Equivalent 2nd Maxwell
relaxation time

! Additional parameters used in generating the analytical solution
s_E1 = i_g1*2*(1+s_PR)     ! Equivalent 1st Maxwell elastic
coeff.
s_E2 = i_g2*2*(1+s_PR)     ! Equivalent 2nd Maxwell elastic
coeff.

/title, Biofilm FSI, job=%m_job%, RE=%f_RE% G1=%s_G1% h1=%s_h1% G2=%s_G2%
h2=%s_h2%

/PREP7

/COM *** STEP 2.0 : FLUID ANALYSIS SET-UP
*****

```



```
/COM *** STEP 2.1 : FLUID KEYPOINTS & LINES
```

```
*****
```

```
K,1,0,0                      ! level 0 keypoints
K,2,g_dev,0
K,3,g_dev+g_for,0
K,4,g_dev+g_for+g_cle,0
K,5,g_dev+g_for+g_cle+g_aft,0
K,6,g_dev+g_for+g_cle+g_aft+g_wit,0
```

```
K,11,0,g_che                 ! level 1 keypoints
K,12,g_dev,g_che
K,13,g_dev+g_for,g_che
K,14,g_dev+g_for+g_cle,g_che
K,15,g_dev+g_for+g_cle+g_aft,g_che
K,16,g_dev+g_for+g_cle+g_aft+g_wit,g_che
```

```
K,21,0,g_con                 ! level 2 keypoints
K,22,g_dev,g_con
K,23,g_dev+g_for,g_con
K,24,g_dev+g_for+g_cle,g_con
K,25,g_dev+g_for+g_cle+g_aft,g_con
K,26,g_dev+g_for+g_cle+g_aft+g_wit,g_con
```

```
/COM *** STEP 2.2 : FLUID AREAS
```

```
*****
```

```
! areas in flow domain created using keypoints in CCW order
```

```
A,1,2,12,11                  ! Area 1 (A1)
A,2,3,13,12                  ! Area 2 (A2)
A,4,5,15,14                  ! Area 3 (A3)
A,5,6,16,15                  ! Area 4 (A4)
A,11,12,22,21                ! Area 5 (A5)
A,12,13,23,22                ! Area 6 (A6)
A,13,14,24,23                ! Area 7 (A7)
A,14,15,25,24                ! Area 8 (A8)
A,15,16,26,25                ! Area 9 (A9)
```

```
/COM *** STEP 2.3 : FLUID NODES
```

```
*****
```

```
! Parameter definitions of lines created implicitly by "A" command above
```

```
! Mesh controls are applied to component lines
```

```
KSEL,S,KP,,1,2              ! select keypoint 1 to 2
LSLK,S,1                     ! select line that includes all
selected kp's
*GET,line1_2,LINE,0,NUM,MAX   ! name this line "line1_2"
LESIZE,line1_2,,,n_dev,n_Rdev ! generate nodes on this line
```

```
KSEL,S,KP,,11,12            ! select keypoint 11 to 12
LSLK,S,1                     ! select line that includes all
selected kp's
*GET,line11_12,LINE,0,NUM,MAX ! name this line "line11_12"
LESIZE,line11_12,,,n_dev,n_Rdev ! generate nodes on this line
```

```
KSEL,S,KP,,21,22            ! select keypoint 21 to 22
LSLK,S,1                     ! select line that includes all
selected kp's
*GET,line21_22,LINE,0,NUM,MAX ! name this line "line21_22"
```

```

LESIZE,line21_22,,,n_dev,n_Rdev      ! generate nodes on this line

KSEL,S,KP,,2,3                        ! select keypoint 2 to 3
LSLK,S,1                              ! select line that includes all
selected kp's
*GET,line2_3,LINE,0,NUM,MAX           ! name this line "line2_3"
LESIZE,line2_3,,,n_for,1.0           ! generate nodes on this line

KSEL,S,KP,,12,13                      ! select keypoint 12 to 13
LSLK,S,1                              ! select line that includes all
selected kp's
*GET,line12_13,LINE,0,NUM,MAX         ! name this line "line12_13"
LESIZE,line12_13,,,n_for,1.0         ! generate nodes on this line

KSEL,S,KP,,22,23                      ! select keypoint 22 to 23
LSLK,S,1                              ! select line that includes all
selected kp's
*GET,line22_23,LINE,0,NUM,MAX         ! name this line "line22_23"
LESIZE,line22_23,,,n_for,1.0         ! generate nodes on this line

KSEL,S,KP,,13,14                      ! select keypoint 13 to 14
LSLK,S,1                              ! select line that includes all
selected kp's
*GET,line13_14,LINE,0,NUM,MAX         ! name this line "line13_14"
LESIZE,line13_14,,,n_che,1.0         ! generate nodes on this line

KSEL,S,KP,,23,24                      ! select keypoint 23 to 24
LSLK,S,1                              ! select line that includes all
selected kp's
*GET,line23_24,LINE,0,NUM,MAX         ! name this line "line23_24"
LESIZE,line23_24,,,n_che,1.0         ! generate nodes on this line

KSEL,S,KP,,4,5                        ! select keypoint 4 to 5
LSLK,S,1                              ! select line that includes all
selected kp's
*GET,line4_5,LINE,0,NUM,MAX           ! name this line "line4_5"
LESIZE,line4_5,,,n_aft,1.0           ! generate nodes on this line

KSEL,S,KP,,14,15                      ! select keypoint 14 to 15
LSLK,S,1                              ! select line that includes all
selected kp's
*GET,line14_15,LINE,0,NUM,MAX         ! name this line "line14_15"
LESIZE,line14_15,,,n_aft,1.0         ! generate nodes on this line

KSEL,S,KP,,24,25                      ! select keypoint 24 to 25
LSLK,S,1                              ! select line that includes all
selected kp's
*GET,line24_25,LINE,0,NUM,MAX         ! name this line "line24_25"
LESIZE,line24_25,,,n_aft,1.0         ! generate nodes on this line

KSEL,S,KP,,5,6                        ! select keypoint 5 to 6
LSLK,S,1                              ! select line that includes all
selected kp's
*GET,line5_6,LINE,0,NUM,MAX           ! name this line "line5_6"
LESIZE,line5_6,,,n_wit,n_Rwit        ! generate nodes on this line

KSEL,S,KP,,15,16                      ! select keypoint 15 to 16
LSLK,S,1                              ! select line that includes all
selected kp's
*GET,line15_16,LINE,0,NUM,MAX         ! name this line "line15_16"

```

```
LESIZE,line15_26,,,n_wit,1/n_Rwit      ! generate nodes on this line
KSEL,S,KP,,25,26                        ! select keypoint 25 to 26
LSLK,S,1                                ! select line that includes all
selected kp's
*GET,line25_26,LINE,0,NUM,MAX           ! name this line "line25_26"
LESIZE,line25_26,,,n_wit,1/n_Rwit      ! generate nodes on this line
KSEL,S,KP,,1,1                          ! select keypoint 1 (new set)
KSEL,A,KP,,11,11                        ! add keypoint 11 to selection
LSLK,S,1                                ! select line that includes all
selected kp's
*GET,line1_11,LINE,0,NUM,MAX            ! name this line "line1_11"
LESIZE,line1_11,,,n_che,1.0            ! generate nodes on this line
KSEL,S,KP,,2,2                          ! select keypoint 2 (new set)
KSEL,A,KP,,12,12                        ! add keypoint 12 to selection
LSLK,S,1                                ! select line that includes all
selected kp's
*GET,line2_12,LINE,0,NUM,MAX            ! name this line "line2_12"
LESIZE,line2_12,,,n_che,1.0            ! generate nodes on this line
KSEL,S,KP,,3,3                          ! select keypoint 3 (new set)
KSEL,A,KP,,13,13                        ! add keypoint 13 to selection
LSLK,S,1                                ! select line that includes all
selected kp's
*GET,line3_13,LINE,0,NUM,MAX            ! name this line "line3_13"
LESIZE,line3_13,,,n_che,1.0            ! generate nodes on this line
KSEL,S,KP,,4,4                          ! select keypoint 4 (new set)
KSEL,A,KP,,14,14                        ! add keypoint 14 to selection
LSLK,S,1                                ! select line that includes all
selected kp's
*GET,line4_14,LINE,0,NUM,MAX            ! name this line "line4_14"
LESIZE,line4_14,,,n_che,1.0            ! generate nodes on this line
KSEL,S,KP,,5,5                          ! select keypoint 5 (new set)
KSEL,A,KP,,15,15                        ! add keypoint 15 to selection
LSLK,S,1                                ! select line that includes all
selected kp's
*GET,line5_15,LINE,0,NUM,MAX            ! name this line "line5_15"
LESIZE,line5_15,,,n_che,1.0            ! generate nodes on this line
KSEL,S,KP,,6,6                          ! select keypoint 6 (new set)
KSEL,A,KP,,16,16                        ! add keypoint 16 to selection
LSLK,S,1                                ! select line that includes all
selected kp's
*GET,line6_16,LINE,0,NUM,MAX            ! name this line "line6_16"
LESIZE,line6_16,,,n_che,1.0            ! generate nodes on this line
KSEL,S,KP,,11,11                        ! select keypoint 11 (new set)
KSEL,A,KP,,21,21                        ! add keypoint 21 to selection
LSLK,S,1                                ! select line that includes all
selected kp's
*GET,line11_21,LINE,0,NUM,MAX           ! name this line "line11_21"
LESIZE,line11_21,,,n_con,1/n_Rcon      ! generate nodes on this line
KSEL,S,KP,,12,12                        ! select keypoint 12 (new set)
KSEL,A,KP,,22,22                        ! add keypoint 22 to selection
```

```

LSLK,S,1                                ! select line that includes all
selected kp's                            ! name this line "line12_22"
*GET,line12_22,LINE,0,NUM,MAX            ! generate nodes on this line
LESIZE,line12_22,,,n_con,n_Rcon

KSEL,S,KP,,13,13                         ! select keypoint 13 (new set)
KSEL,A,KP,,23,23                         ! add keypoint 23 to selection
LSLK,S,1                                ! select line that includes all
selected kp's                            ! name this line "line13_23"
*GET,line13_23,LINE,0,NUM,MAX            ! generate nodes on this line
LESIZE,line13_23,,,n_con,n_Rcon

KSEL,S,KP,,14,14                         ! select keypoint 14 (new set)
KSEL,A,KP,,24,24                         ! add keypoint 24 to selection
LSLK,S,1                                ! select line that includes all
selected kp's                            ! name this line "line14_24"
*GET,line14_24,LINE,0,NUM,MAX            ! generate nodes on this line
LESIZE,line14_24,,,n_con,n_Rcon

KSEL,S,KP,,15,15                         ! select keypoint 15 (new set)
KSEL,A,KP,,25,25                         ! add keypoint 25 to selection
LSLK,S,1                                ! select line that includes all
selected kp's                            ! name this line "line15_25"
*GET,line15_25,LINE,0,NUM,MAX            ! generate nodes on this line
LESIZE,line15_25,,,n_con,n_Rcon

KSEL,S,KP,,16,16                         ! select keypoint 16 (new set)
KSEL,A,KP,,26,26                         ! add keypoint 26 to selection
LSLK,S,1                                ! select line that includes all
selected kp's                            ! name this line "line16_26"
*GET,line16_26,LINE,0,NUM,MAX            ! generate nodes on this line
LESIZE,line16_26,,,n_con,n_Rcon

/COM *** STEP 2.4 : FLUID ELEMENT TYPE DEFINED
*****

ET,1,FLUID141                            ! Element type 2D fluid #141
*IF,c_ALE,EQ,1,THEN
    KEYOPT,1,4,1                        ! Include if ALE meshing is turned on
w/FLDA,SOLU,ALE,TRUE command
*ENDIF

/COM *** STEP 2.5 : FLUID TYPES, CONSTANTS AND PROPERTIES ENABLED
*****

TYPE,1                                    ! Enable Element Type 1
! No material or real constants are defined for fluid analysis.  Fluid
properties
! are defined with the FLDATA command below.

/COM *** STEP 2.6 : FLUID MESH CREATED
*****

MSHAPE,0,2D                             ! Specifies 2D quad elements
MSHKEY,1                                 ! Specifies mapped meshing
AMESH,1,9,1                             ! Mesh areas 1 thru 9

/COM *** STEP 2.7 : FLUID BOUNDARY CONDITIONS
*****

```

```

! Velocities in X & Y applied to the inlet
DL,line1_11, ,VX,f_vx1,1      ! Apply velocity in X direction
on line1_11
DL,line11_21, ,VX,f_vx1,1    ! Apply velocity in X direction on
line11_21
DL,line1_11, ,VY,f_vy2,1      ! Apply velocity in Y direction
on line1_11
DL,line11_21, ,VY,f_vy2,1    ! Apply velocity in Y direction on
line11_21

*IF,c_ALE,EQ,1,THEN
! Displacement of morphing mesh restricted at inlet
DL,line1_11, ,UX,0,1          ! UX direction on line1_11
DL,line11_21, ,UX,0,1        ! UX direction on
line11_21
DL,line1_11, ,UY,0,1          ! UY direction on line1_11
DL,line11_21, ,UY,0,1        ! UY direction on
line11_21
*ENDIF

! No Slip condition applied to top boundaries
DL,line21_22, ,VX,0,1         ! no slip X
DL,line21_22, ,VY,0,1         ! no slip Y
DL,line22_23, ,VX,0,1         ! no slip X
DL,line22_23, ,VY,0,1         ! no slip Y
DL,line23_24, ,VX,0,1         ! no slip X
DL,line23_24, ,VY,0,1         ! no slip Y
DL,line24_25, ,VX,0,1         ! no slip X
DL,line24_25, ,VY,0,1         ! no slip Y
DL,line25_26, ,VX,0,1         ! no slip X
DL,line25_26, ,VY,0,1         ! no slip Y

! Displacement of morphing mesh restricted at top boundaries
! NOTE: UX is not fixed because this is the direction of flow
*IF,c_ALE,EQ,1,THEN
DL,line21_22, ,UY,0,1
DL,line22_23, ,UY,0,1
DL,line23_24, ,UY,0,1
DL,line24_25, ,UY,0,1
DL,line25_26, ,UY,0,1
*ENDIF

! No Slip condition applied to bottom boundaries
DL,line1_2, ,VX,0,1           ! no slip X
DL,line1_2, ,VY,0,1           ! no slip Y
DL,line2_3, ,VX,0,1           ! no slip X
DL,line2_3, ,VY,0,1           ! no slip Y
DL,line4_5, ,VX,0,1           ! no slip X
DL,line4_5, ,VY,0,1           ! no slip Y
DL,line5_6, ,VX,0,1           ! no slip X
DL,line5_6, ,VY,0,1           ! no slip Y

! Displacement of morphing mesh restricted at bottom boundaries
*IF,c_ALE,EQ,1,THEN
DL,line1_2, ,UX,0,1
DL,line1_2, ,UY,0,1
DL,line2_3, ,UX,0,1
DL,line2_3, ,UY,0,1
DL,line4_5, ,UX,0,1
DL,line4_5, ,UY,0,1

```



```

        DL,line5_6, ,UX,0,1
        DL,line5_6, ,UY,0,1
*ENDIF

! No Slip condition applied to obstacle boundaries
DL,line3_13, ,VX,0,1           ! no slip X
DL,line3_13, ,VY,0,1           ! no slip Y
DL,line13_14, ,VX,0,1          ! no slip X
DL,line13_14, ,VY,0,1          ! no slip Y
DL,line4_14, ,VX,0,1           ! no slip X
DL,line4_14, ,VY,0,1           ! no slip Y

! Pressure boundary at outlet
DL,line6_16, ,PRES,0,1
DL,line16_26, ,PRES,0,1

! Displacement of morphing mesh restricted at outlet boundary
*IF,c_ALE,EQ,1,THEN
        DL,line6_16, ,UX,0,1
        DL,line6_16, ,UY,0,1
        DL,line16_26, ,UX,0,1
        DL,line16_26, ,UY,0,1
*ENDIF

/COM *** STEP 2.8 : FLUID PROPERTIES DEFINED
*****

FLDATA,PROT,DENS,CONSTANT      ! fluid density is constant
FLDATA,PROT,VISC,CONSTANT      ! fluid viscosity is constant
FLDATA,NOMI,DENS,f_ro          ! fluid density specified by
parameter ro
FLDATA,NOMI,VISC,f_nu          ! fluid viscosity specified by
parameter nu

ALLSEL,ALL
SAVE

FINISH

/COM *** STEP 2.9 : FLUID SOLUTION CONTROL
*****

/SOLUTION

! general fluid solution options/controls
FLDATA,ITER,EXEC,c_iF          ! number of fluid sol'n global
iterations
FLDATA,TIME,STEP,c_dtF        ! time step increment for fluid
analysis
FLDATA,SOLU,TRAN,TRUE          ! fluid solution is transient
FLDATA,ALGR,SEGR,SIMPLEN       ! enhanced SIMPLEN solution
algorithm
*IF,c_ALE,EQ,1,THEN
        FLDA,SOLU,ALE,TRUE      ! activates ALE mesh-morphing
*ENDIF

! Pressure and momentum equation settings
FLDATA,SOLU,FLOW,TRUE          ! solves pressure and momentum
eqns.
FLDATA,ADVM,MOME,SUPG          ! Use streamline upward
discretization on momentum eqn.

```

```

*IF, c_ALE, EQ, 1, THEN
    FLDATA, QUAD, MOMD, 2
    ! Accuracy of quadrature increased for
ALE
    FLDATA, QUAD, MOMA, 2
    FLDATA, QUAD, MOMS, 2
    FLDATA, QUAD, PRSD, 2
    FLDATA, QUAD, PRSS, 2
*ENDIF

! Turbulence settings
*IF, c_TURB, EQ, 1, THEN
    FLDATA, SOLU, TURB, TRUE
    ! activates turbulence model
    FLDATA, TURB, WALL, EQLB
    ! specifies 'Wall Equilibrium'
model for boundary layer
    FLDATA, ADVN, TURB, SUPG
    ! Use streamline upward
discretization on turbulence eqn.
    FLDATA, OUTP, TAUW, TRUE
    ! write wall shear to output
    FLDATA, OUTP, YPLU, TRUE
    ! write turb Y+ to output
    FLDATA, STAB, TURB, 0.01
    ! turbulence inertial relaxation
factor
    FLDATA, RELX, VX, 0.35
    ! Momentum equation relaxation factor,
VX
    FLDATA, RELX, VY, 0.35
    ! Momentum equation relaxation factor,
VY
    FLDATA, RELX, PRES, 0.35
    ! Momentum equation relaxation
factor, PRES
    FLDATA, TURB, RATI, 100000
    ! Increases initial effective
viscosity
    FLDATA, MIR, MOME, 0.5
    ! Mod. initial relaxation for Momentum
eqns. (fixs neg. diag.)
    FLDATA, MIR, TURB, 0.5
    ! Mod. initial relaxation for
Turbulence eqns. (fixs neg. diag.)
*ENDIF

```

ALLSEL, ALL  
SAVE

```

/COM *** STEP 3.0 : STRUCTURAL ANALYSIS SET-UP
*****

```

```

!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
```

FSI

113-----114  
!  
!  
!  
!  
  
FSI    !               OBS               !               FSI  
!  
!  
!  
!  
  
102-----103-----104-----105

ANCHORED

```

!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
```

: Note: kp 2,3,4,5,13,14 are in the same physical locations as  
102,103,104,105,113,114

FINISH  
/PREP7

/COM \*\*\* STEP 3.1 : STRUCTURAL KEYPOINTS & LINES  
\*\*\*\*\*

K,103,g\_dev+g\_for,0  
K,104,g\_dev+g\_for+g\_cle,0  
K,113,g\_dev+g\_for,g\_che  
K,114,g\_dev+g\_for+g\_cle,g\_che

/COM \*\*\* STEP 3.2 : STRUCTURAL AREAS  
\*\*\*\*\*

! areas in solid domain created using keypoints in CCW order  
A,103,104,114,113 ! Area 10 (A10)

/COM \*\*\* STEP 3.3 : STRUCTURAL NODES  
\*\*\*\*\*

! Parameter definitions of lines created implicitly by "A" command above  
! Mesh controls are applied to component lines  
KSEL,S,KP,,103,104 ! select keypoint 103 to 104  
KSEL,A,KP,,113,114 ! select keypoint 113 to 114  
LSLK,S,1 ! select line that includes all  
selected kp's  
LESIZE,ALL,,,n\_solid,1.0 ! generate nodes on these lines

/COM \*\*\* STEP 3.4 : STRUCTURAL ELEMENT TYPE DEFINED  
\*\*\*\*\*

ET,2,PLANE183,,,3 ! Element type 2D 8 node solid  
(Element Type 2)  
R,2,.5, ! Plane thickness (Real Constants  
2)

/COM \*\*\* STEP 3.5 : STRUCTURAL PROPERTIES DEFINED  
\*\*\*\*\*

MP,EX,2,s\_E0 ! Youngs Modulus (Material 2)  
MP,PRXY,2,s\_PR ! Poisson ratio, major (Material  
2)  
MP,DENS,2,s\_DE ! Mass density  
TB,PRONY,2,,2,SHEAR ! Creates a table of Prony coefficients  
(Material 2)  
TBDATA,1,s\_A1,s\_TAU1,s\_A2,s\_TAU2 ! Fills in the values of the coeff.

/COM \*\*\* STEP 3.6 : STRUCTURAL TYPES, CONSTANTS AND PROPERTIES ENABLED  
\*\*\*\*\*

TYPE,2 ! Enable Element Type 2  
REAL,2 ! Enable Real Constants 2  
MAT,2 ! Enable Material 2

/COM \*\*\* STEP 3.7 : STRUCTURAL ELEMENTS CREATED  
\*\*\*\*\*

MSHAPE,0,2D ! Specifies 2D quad elements  
MSHKEY,1 ! Specifies mapped meshing  
AMESH,10,10,1 ! Mesh area 10

```
/COM *** STEP 3.8 : STRUCTURAL BOUNDARY CONDITIONS
*****
```

```
! The following D commands fix the base of the cube
```

```
ESEL,S,TYPE,,2           ! select all solid elements
NSLE,S,ALL               ! select all nodes in solid
elements
NSEL,R,LOC,Y,0           ! select subset of nodes at y = 0
mm
D,ALL,UX,0               ! fix x displ.
D,ALL,UY,0               ! fix y displ.
```

```
/COM *** STEP 3.9 : STRUCTURAL SOLUTION CONTROL
*****
```

```
DELTIM,c_dtS,,OFF       ! time step increment for solid
analysis
CNVTOL,F,,,,0.00000001  ! set force converge threshold
value (MENREF)
```

```
ALLSEL,ALL
SAVE
```

```
/COM *** STEP 4.0 : FLUID-STRUCTURAL INTERACTION
*****
```

```
/COM *** STEP 4.1 : FLAG FLUID-STRUCTURAL INTERFACES
*****
```

```
! The following two SF commands flag the low x face of the obstacle
SFL,line3_13,FSIN,1      ! FSIN flag 1 on fluid side line
ESEL,S,TYPE,,2           ! select all solid elements
NSLE,S,ALL               ! select all nodes in solid
elements
NSEL,R,LOC,X,g_dev+g_for ! select subset of nodes at x = (g_dev
+ g_for) mm
SF,ALL,FSIN,1            ! FSIN flag 1 on solid side nodes
```

```
! The following two SF commands flag the high y face of the obstacle
SFL,line13_14,FSIN,2     ! FSIN flag 2 on fluid side line
ESEL,S,TYPE,,2           ! select all solid elements
NSLE,S,ALL               ! select all nodes in solid
elements
NSEL,R,LOC,Y,g_che       ! select subset of nodes at y = g_che
mm
SF,ALL,FSIN,2            ! FSIN flag 2 on solid side nodes
```

```
! The following two SF commands flag the high x face of the obstacle
SFL,line4_14,FSIN,3      ! FSIN flag 3 on fluid side
ESEL,S,TYPE,,2           ! select all solid elements
NSLE,S,ALL               ! select all nodes in solid
elements
NSEL,R,LOC,X,g_dev+g_for+g_cle ! select subset of nodes at x = (g_dev
+ g_for + g_cle) mm
SF,ALL,FSIN,3            ! FSIN flag 3 on solid side nodes
```

```
ALLSEL,ALL
```



SAVE

/COM \*\*\* STEP 4.2 : FLUID-STRUCTURAL INTERACTION SOLUTION OPTIONS  
\*\*\*\*\*

! Basic FSI Analysis Options

|                  |                                   |
|------------------|-----------------------------------|
| FSAN,ON          | ! FSI switched on                 |
| FSOR,FLUID,SOLID | ! FSI solution order              |
| FSTR,FLUID,      | ! Fluid soln is transient         |
| FSTR,SOLID,      | ! Solid soln is transient         |
| FSIN,CONS        | ! Interpolate conservative values |

! End time, Load time and time increments

|                       |                                        |
|-----------------------|----------------------------------------|
| FSIT,c_time1,c_ltime1 | ! End time, load time for FSI          |
| analysis              |                                        |
| FSDT,c_dtFSI          | ! time step increment for FSI analysis |

! Maximum number of stagger iterations

FSIT,c\_iFSI

! FSI Convergence values

|                |                     |
|----------------|---------------------|
| FSCO,FX,0.0001 | ! convergence value |
| FSCO,FY,0.0001 | ! convergence value |
| FSCO,FZ,0.0001 | ! convergence value |
| FSCO,UX,0.0001 | ! convergence value |
| FSCO,UY,0.0001 | ! convergence value |
| FSCO,UZ,0.0001 | ! convergence value |

! Output frequency

|          |                             |
|----------|-----------------------------|
| FSOU,ALL | ! Output at every time step |
|----------|-----------------------------|

! Relaxation values

|              |                             |
|--------------|-----------------------------|
| FSRE,ALL,0.8 | ! All relaxation values set |
|--------------|-----------------------------|

ALLSEL,ALL

SAVE

\*IF,c\_SOLV,EQ,0,THEN

! Exit if SOLU switch is off.

FINISH

/EXIT

/EOF

\*ENDIF

FINISH

/COM \*\*\* STEP 5.0 : SOLUTION

\*\*\*\*\*

/SOLUTION

/COM \*\*\* STEP 5.1 : DEFINE MONITOR FILE DOFS

\*\*\*\*\*

! the following identifies the node number on the upper right of cube.

|                                |                                         |
|--------------------------------|-----------------------------------------|
| ESEL,S,TYPE,,2                 | ! select element type 2 (cube)          |
| NSLE,S,ALL                     | ! select all nodes on these             |
| elements                       |                                         |
| NSEL,R,LOC,Y,g_che             | ! select subset of nodes at top of cube |
| NSEL,R,LOC,X,g_dev+g_for+g_cle | ! select subset of nodes on right side  |
| of cube                        |                                         |

```

*GET,n_mntr1,NODE,0,NUM,MAX          ! assign node number to parameter
"n_mntr1"

! the following identifies the node number on the lower right of cube.
ESEL,S,TYPE,,2                      ! select element type 2 (cube)
NSLE,S,ALL                          ! select all nodes on these
elements
NSEL,R,LOC,Y,g_che/n_solid          ! select subset of nodes at y = g_che /
n_solid
NSEL,R,LOC,X,g_dev+g_for+g_cle      ! select subset of nodes on right side
of cube
*GET,n_mntr2,NODE,0,NUM,MAX          ! assign node number to parameter
"n_mntr2"

! set solution monitoring nodes and DOFs
MONITOR,1,n_mntr1,UX                ! track UX of upper right corner
MONITOR,2,n_mntr1,UY                ! track UY of upper right corner
MONITOR,3,n_mntr2,UY                ! track UY of lower right corner

! Apply displacement constraints for nodes moving below the channel base (y =
0).
! This approach does not work with ALE!!! (you'll get negative volumes in
fluid elements.)

/COM *** STEP 5.2 : SOLVE MODEL
*****

ALLSEL,ALL

SOLVE                                ! RUN STEP 1

ALLSEL,ALL
SAVE

/COM *** SET STATE VARIABLES FOR OPTIMIZATION ***

o_SV = ((UX(o_node)**2-UY(o_node)**2)**(1/2)) ! Straight line deflection of
node o_node

/COM *** DEFINE OBJECTIVE FUNCTION FOR OPTIMIZATION ***

o_OF = ABS(o_SV-o_TGT)               ! Minimize difference between SV
and target deflection

ALLSEL,ALL
SAVE

FINISH

/OPT

/COM *** OPTIMIZATION OF MATERIAL PARAMETERS ***

*IF,o_OPT,EQ,1,THEN

    OPVAR,o_OF,OBJ,,g_cle/5
    OPVAR,o_SV,SV,0,1
    OPVAR,s_G1,DV,s_G1*0.1,s_G1*10,s_G1/100
    OPVAR,s_h1,DV,s_h1*0.1,s_h1*10,s_h1/100
    OPVAR,s_G2,DV,s_G2*0.1,s_G2*10,s_G2/100

```

```

OPVAR,s_h2,DV,s_h2*0.1,s_h2*10,s_h2/100

OPTYPE,FIRST
OPFRST,100
OPEXE                                     ! ready, set, optimize!

OPLIST,ALL,,1                             ! List all optimization
parameters

PARSAV,,all_param,txt                     ! Write parameters to a
file

/AXLAB,Y,U(o_node)

PLVAROPT,o_OF                             ! Plot Obj. Function

FINISH

*ENDIF

ALLSEL,ALL
SAVE

FINISH

/COM *** STEP 6.0 DATABASE RESULTS POST PROCESSING
*****

/POST1

/COM *** STEP 6.1 SET STANDARD VIEW
*****

! load results
RESUME,%m_job%,db
INRES,ALL
FILE,%m_job%,rst
SET,LAST
ALLSEL,ALL

/DSCALE,ALL,1.0                           ! set displacement scaling to 1.0
/VIEW,1,0,0,1.0000                        ! view from the z dir
/FOC,1,g_dev+g_for+(3*g_cle/4),g_che/2,0 ! center on cube
/DIST,1,0.25                               ! zoom scale
PLNSOL,S,EQV,2                             ! plot von Mises stresses

/COM *** STEP 6.2 PRONY PARAMETER FILE OUTPUT
*****

! The following generates a text file for recording Prony parameters:
p_filename = STRCAT(m_job,'_prony')        ! builds filename as string
parameter
*CFOPEN,%p_filename%,txt

*VWRITE,s_E0,s_PR
(2X,'E0:',F16.8,2X,'PR:',F16.8)
*VWRITE,s_A1,s_TAU1
(2X,'A1:',F16.8,2X,'Tau1:',F16.8)
*VWRITE,s_A2,s_TAU2
(2X,'A2:',F16.8,2X,'Tau2:',F16.8)

```

```
*VWRITE,s_G1,s_h1,s_G2,s_h2
(2X,'G1:',F16.8,2X,'h1:',F16.8,2X,'G2:',F16.8,2X,'h2:',F16.8)
```

```
FINISH
```

```
/COM *** STEP 7.0 TIME HISTORY POST PROCESSING
*****
```

```
/POST26
FINISH
```

```
ALLSEL,ALL
SAVE
```

```
/EOF
/COM *** END OF FILE
*****
```



APPENDIX F

EFFECTIVE SHEAR ANGLE & MODULUS CALCULATIONS

Shear Angle Calculations

from Shear Cube deformation in Test Case Problem

1. Initial Height of Cube  mm

2. Deformation of upper left node (626)

node 626

UX 3.08E-02  mmUY 2.05E-02  mm

U 0.037014 mm

% strain 0.185072 mm/mm

3. Node 626 deforms from A to B

Initial: Ax  mmAy  mmFinal: Bx  mmBy  mm

4. Shear Angle Calculations

 $\theta$  1.43 rad $\theta$  82.03907 deg $\gamma$   deg $\gamma$   rad

5. Wall Shear Stress Calculations

V 1000 mm/s

D 3 mm

r 0.000001 kg/mm<sup>3</sup>

m 0.000001 kg/mm-s

RE 3000

RE\* 3600

f 0.01

 $\tau_w$   mN/mm<sup>2</sup> $\tau_w$   N/m<sup>2</sup>

6. Effective Shear Modulus

G  N/m<sup>2</sup>

APPENDIX G

SAMPLE ANSYS INPUT FOR ROLLING BIOFILM PROBLEM

```
finish
/clear
```

```
/COM,ANSYS RELEASE 7.1
/COM
```

```
*****
*****
/COM ***          FSI LVE Roll Constant Times (Sequential Weak Coupling)
***
/COM
*****
*****
```

```
!  MODEL NAME
```

```
!  -----
```

```
m_series = 'broll'
```

```
m_number = '05'
```

```
!  MODEL NOTES
```

```
!  -----
```

```
!  Series is used to develop shape function for hemisphere.
```

```
!
```

```
!  Parent: PARENT
```

```
!
```

```
!  Motivation:
```

```
!
```

```
!  Changes:
```

```
!
```

```
!  Result:
```

```
!
```

```
!  Runtime est.:
```

```
!
```

```
!  -----
```

```
!  Brett Towler
```

```
!  Civil Engineering Dept., MSU-Bozeman
```

```
!
```

```
!          21-----23-----25-----
```

```
27
```

```
!
```

```
!
```

```
!
```

```
!
```

```
!
```

```
!
```

```
!
```

```
!
```

```
!
```

```
!
```

```
!
```

```
!
```

```
!
```

```
!
```

```
!
```

```
!
```

```
!
```

```
17
```

```
!  INLET -->
```

```
!  --> OUTLET
```

```
!          11-----203.          .202-----
```

```

! Steps in the SWC FSI model
!
! 1.0 Model Parameters
!     1.1 User Input
!     1.2 Define Dependent Parameters
!
! 2.0 Fluid Analysis Set-Up
!     2.1 Fluid Keypoints & Lines
!     2.2 Fluid Areas
!     2.3 Fluid Nodes
!     2.4 Fluid Element Type Defined
!     2.5 Fluid Types, Constants & Properties Enabled
!     2.6 Fluid Mesh Created
!     2.7 Fluid Boundary Conditions
!     2.8 Fluid Properties Defined
!     2.9 Fluid Solution Control
!
! 3.0 Structural Analysis Set-Up
!     3.1 Structural Keypoints & Lines
!     3.2 Structural Areas
!     3.3 Structural Nodes
!     3.4 Structural Element Type Defined
!     3.5 Structural Properties Defined
!     3.6 Structural Types, Constants & Properties Enabled
!     3.7 Structural Mesh Created
!     3.8 Structural Boundary Conditions
!
! 4.0 Fluid Structure Interaction
!     4.1 Flag Fluid Structural Interfaces
!     4.2 Fluid Structural Interaction Solution Options
!
! 5.0 Solution
!     5.1 Define Monitor File DOFs
!     5.2 Solve model
!
! 6.0 Database Results Post-Processing
!     6.1 Set Standard View
!     6.2 Prony Parameter File Output
!     6.3 Deformed Centroid Calculation
!     6.4 Resultant Surface Force Calculation
!     6.5 Reaction Force Calculation
!     6.6 Resultant File Output
!
! 7.0 Time-History Post-Processing

```

! UNITS:

```

!           The base unit for problems of this scale is millimeter (mm).
!           L :           1 m           =           1000 mm           height,
length, width
!           L/S :           1 m/s           =           1000 mm/s           velocity
!           M/L^3 :           1 kg/m^3           =           10^-9 kg/mm^3 density
!           M/L-T :           1 kg/m-s           =           10^-3 kg/mm-s dynamic viscosity
!           F :           1 kg-m/s^2           =           1000 kg-mm/s^2           force, load,
reaction
!           F/L^2 :           1 kg/m-s^2           =           10^-3 kg/mm-s^2           pressure,
Young's modulus
!
/COM *** STEP 1.0 : MODEL PARAMETERS
*****

/COM *** STEP 1.1 : USER-DEFINED PARAMETERS
*****

!
! -----
! Start of User-Defined Parameters
! -----

! User input: GEOMETRIC PARAMETERS (g)
g_Hc = 3           ! height of the channel, mm
g_Lc = 40          ! length of the channel, mm
g_Cl = 20          ! center of biofilm hemisphere in x dir, mm
g_Rl = 0.2         ! radius of biofilm hemisphere, mm

! User input: MESH CONTROL (n)

! number of elements used in fluid mesh generation
n_r1 = 6           ! # of elements on L_R21, L_R11
n_r2 = 6           ! # of elements on L_R22, L_R12
n_r3 = 6           ! # of elements on L_R23, L_R13
n_theta = 6        ! # of elements on L_TH1, L_TH2, L_TH3, L_TH4

! bias ratios used in fluid mesh generation
n_R_r1 = 1.0       ! 1 way bias on L_R21, L_R11
n_R_r2 = 1.0       ! 1 way bias on L_R22, L_R12
n_R_r3 = 1.0       ! 1 way bias on L_R23, L_R13
n_R_theta = 1.0    ! 1 way bias on L_TH1, L_TH2, L_TH3, L_TH4
n_R_hy = -3.0      ! on L_HYinlet, L_HYmid1, L_HYmid2, L_HYoutlet (- OK)
n_R_lx = 3.0       ! on L_LXbase, L_LXmid, L_LXtop
n_R_hx = 3.0       ! on L_HXbase, L_HXmid, L_HXtop

! number of elements used in structural mesh generation
n_box_x = 4        ! # of elements on L_S11, L_BOX1, L_BOX3,
L_S13
n_box_y = 4        ! # of elements on L_BOX4, L_BOX2, L_S12
n_box_pt = 4       ! # of elements on L_PT1..L_PT4

! bias ratios used in structural mesh generation
n_R_box_x = 1.0    ! 1 way bias on L_S11, L_BOX1, L_BOX3, L_S13
n_R_box_y = 1.0    ! 1 way bias on L_BOX4, L_BOX2, L_S12
n_R_box_pt = 1.0   ! 1 way bias on L_PT1..L_PT4

! User input: FLUID PARAMETERS (f)
f_vx = 1000.0      ! Vx at inlet, mm/s
f_vy = 0.0         ! Vy at inlet, mm/s

```

```

f_ro = 0.000001          ! density of water, kg/mm^3
f_mu = 0.000001          ! dynamic viscosity of water, kg/mm-s

! User input: SOLID PARAMETERS (s)
s_G1 = 10                 ! shear modulus for Maxwell (series) spring,
kg/mm-s^2
s_h1 = 150                ! viscous coeff. for Maxwell (series) dashpot,
kg/mm-s^2
s_G2 = 20                 ! shear modulus for Kelvin (parallel) spring,
kg/mm-s^2
s_h2 = 100                ! viscous coeff. for Kelvin (parallel)
dashpot, kg/mm-s^2
s_PR = 0.33               ! Poisson's ratio
s_DE = 0.000001          ! mass density of biofilm

! User input: SOLUTION CONTROL (c)
c_time = 42               ! total time of run, sec.
c_loadtime = 12           ! time to apply the FSI SF loads, sec.
c_dtFSI = 0.25            ! time step for FSI loop (multiple of c_time)
c_dtS = 0.25              ! time step for SOLID solution loop (multiple of
c_dtFSI)
c_dtF = 0.25              ! time step for FLUID solution loop (multiple of
c_dtFSI)
c_iF = 200                ! global iterations for fluid solution
c_iFSI = 20               ! stagger iterations in FSI loop

! ON/OFF Switches
c_ALE = 1                 ! SWITCH - Mesh-morphing switch. 0 = OFF, 1 =
ON
c_SOLV = 1                ! SWITCH - Solve?. 0 = PREP only, 1 = SOLU
c_TURB = 1                ! SWITCH - Activate turbulence eqns?. 0 = NO
only, 1 = YES
c_INVEL = 1               ! SWITCH - Inlet velocity distribution. 0 =
uniform, 1 = Vennard eqn.

! -----
! End of User-Defined Parameters
! -----

/COM *** STEP 1.2 : DEPENDENT PARAMETERS
*****

! WARNING: Do not change these parameters unless you are SURE you know what
! you are doing!!!

Pi=2*ASIN(1)              ! defines Pi

! dependent geometric parameters
g_R2 = g_R1*3              ! radius of boundary hemisphere, mm
g_C2 = g_C1+g_R1           ! center of boundary hemisphere in x dir, mm
g_box = g_R1/2             ! length of biofilm hemisphere interior box
edge, mm

! dependent hydrodynamic parameters
f_nu = f_mu/f_ro           ! kinematic viscosity, mm^2/s
f_RE = f_vx*g_Hc/f_nu      ! channel Reynolds number, ND
f_ff = 0.316/(f_RE)**0.25 ! friction factor estimate (smooth walls)

```



```

! Node/element distribution parameters
n_nel = g_R2*SIN(Pi/3)/n_r1      ! nominal element length
*IF,n_R_hy,GE,0,THEN
  n_hy = NINT(2*(g_Hc-g_R2*SIN(Pi/3))/(n_nel*(1+n_R_hy)))
*ELSE
  n_hy = NINT(3*(g_Hc-g_R2*SIN(Pi/3))/(n_nel*(2+ABS(n_R_hy))))
*ENDIF
n_ly = n_r1                      ! same as n_r1, n_r3
n_R_ly = n_R_r1
n_lx = NINT(2*(g_C2-g_R2)/(n_nel*(1+n_R_lx)))
n_mx = n_r2                      ! same as n_r2
n_R_mx = n_R_r2
n_hx = NINT(2*(g_Lc-(g_C2+g_R2))/(n_nel*(1+n_R_hx)))

! Intermediate parameters used in conversion to Prony series input for
Relaxation modulus
i_p0 = 1
i_p1 = s_h1/s_G1+s_h1/s_G2+s_h2/s_G2
i_p2 = s_h1*s_h2/(s_G1*s_G2)
i_q1 = s_h1
i_q2 = s_h1*s_h2/s_G2

i_A = (i_p1**2-4*i_p2)**(1/2)
i_r1 = (i_p1-i_A)/(2*i_p2)
i_r2 = (i_p1+i_A)/(2*i_p2)

i_ginf = 0                      ! Fixed constant for any Burger
model
i_g1 = (i_q1-i_q2*i_r1)/i_A      ! Equivalent 1st Maxwell shear
coeff.
i_g2 = -(i_q1-i_q2*i_r2)/i_A    ! Equivalent 2nd Maxwell shear
coeff.
i_g0 = i_ginf+i_g1+i_g2

! Prony series representation of the Burger parameters
s_E0 = i_g0*2*(1+s_PR)
s_A1 = i_g1/i_g0
s_TAU1 = 1/i_r1                  ! Equivalent 1st Maxwell
relaxation time
s_A2 = i_g2/i_g0
s_TAU2 = 1/i_r2                  ! Equivalent 2nd Maxwell
relaxation time

! Additional parameters used in generating the analytical solution
s_E1 = i_g1*2*(1+s_PR)          ! Equivalent 1st Maxwell elastic
coeff.
s_E2 = i_g2*2*(1+s_PR)          ! Equivalent 2nd Maxwell elastic
coeff.

ALLSEL,ALL
SAVE

/title, Biofilm FSI, job=%m_job%, RE=%f_RE% G1=%s_G1% h1=%s_h1% G2=%s_G2%
h2=%s_h2%

/PREP7

/COM *** STEP 2.0 : FLUID ANALYSIS SET-UP
*****

```



## /COM \*\*\* STEP 2.1 : FLUID KEYPOINTS &amp; LINES

\*\*\*\*\*

! level 0 keypoints

```

K,1,0,0          ! origin
K,2,g_C2-g_R2,0  ! boundary hemisphere 180
K,3,g_C1-g_R1,0  ! biofilm hemisphere 180
K,4,g_C1,0       ! biofilm hemisphere center
K,5,g_C2,0       ! boundary hemisphere center, biofilm zero
K,6,g_C2+g_R2,0  ! boundary hemisphere zero
K,7,g_Lc,0       ! channel end

```

! level 1 keypoints

```

K,11,0,g_R2*SIN(Pi/3)
K,17,g_Lc,g_R2*SIN(Pi/3)

```

! level 2 keypoints

```

K,21,0,g_Hc
K,23,g_C2-g_R2*COS(Pi/3),g_Hc
K,25,g_C2+g_R2*COS(Pi/3),g_Hc
K,27,g_Lc,g_Hc

```

! biofilm hemisphere generated

```

NUMSTR,KP,101          ! KP start # at 101
NUMSTR,LINE,101        ! LINE start # at 101
CIRCLE,4,g_R1,,5,180,3 ! creates kp's 101,102,103,104, lines
101,102,103 (CCW)

```

! boundary hemisphere generated

```

NUMSTR,KP,201          ! KP start # at 201
NUMSTR,LINE,201        ! LINE start # at 201
CIRCLE,5,g_R2,,6,180,3 ! creates kp's 201,202,203,204, lines
201,202,203 (CCW)
L,2,3                  ! upstream boundary flow base, (L204)
L,5,6                  ! downstream boundary flow base, (L205)

```

NUMMRG,KP,,,HIGH

! merges coincident KPs

! channel boundary lines

```

NUMSTR,LINE,1          ! LINE start # at 1
L,1,2                  ! channel bottom upstream (L1)
L,6,7                  ! channel bottom downstream (L2)
L,21,27               ! channel top (L3)
L,1,11                ! channel inlet (L4)
L,7,17                ! channel outlet (L5)
L,11,203              ! level 1 downstream (L6)
L,202,17              ! level 1 upstream (L7)
L,203,23              ! downstream vertical line (L8)
L,202,25              ! upstream vertical line (L9)
L,103,203             ! boundary flow downstream partition
(L10)
L,102,202             ! boundary flow upstream partition
(L11)

```

## /COM \*\*\* STEP 2.2 : FLUID AREAS

\*\*\*\*\*

! areas in flow domain created using keypoints in CCW order

```

A,2,3,103,203          ! boundary flow area A (A1)
A,103,102,202,203      ! boundary flow area B (A2)

```

```

A,5,6,202,102      ! boundary flow area C (A3)
A,1,2,203,11       ! low y, downstream (A4)
A,11,203,23,21     ! high y, downstream (A5)
A,203,202,25,23    ! high y, midstream (A6)
A,6,7,17,202       ! low y, upstream (A7)
A,202,17,27,25     ! high y, upstream (A8)

```

```

/COM *** STEP 2.3 : FLUID NODES ***

```

```

! set line variable name and assign nodes on line L_R21
KSEL,S,KP,,2
KSEL,A,KP,,203
LSLK,S,1
*GET,L_R21,LINE,0,NUM,MAX
LESIZE,L_R21,,,n_r1,n_R_r1

```

```

! set line variable name and assign nodes on line L_R11
KSEL,S,KP,,3
KSEL,A,KP,,103
LSLK,S,1
*GET,L_R11,LINE,0,NUM,MAX
LESIZE,L_R11,,,n_r1,n_R_r1

```

```

! set line variable name and assign nodes on line L_R22
KSEL,S,KP,,203
KSEL,A,KP,,202
LSLK,S,1
*GET,L_R22,LINE,0,NUM,MAX
LESIZE,L_R22,,,n_r2,n_R_r2

```

```

! set line variable name and assign nodes on line L_R12
KSEL,S,KP,,103
KSEL,A,KP,,102
LSLK,S,1
*GET,L_R12,LINE,0,NUM,MAX
LESIZE,L_R12,,,n_r2,n_R_r2

```

```

! set line variable name and assign nodes on line L_R23
KSEL,S,KP,,202
KSEL,A,KP,,6
LSLK,S,1
*GET,L_R23,LINE,0,NUM,MAX
LESIZE,L_R23,,,n_r3,n_R_r3

```

```

! set line variable name and assign nodes on line L_R13
KSEL,S,KP,,102
KSEL,A,KP,,5
LSLK,S,1
*GET,L_R13,LINE,0,NUM,MAX
LESIZE,L_R13,,,n_r3,n_R_r3

```

```

! set line variable name and assign nodes on partition lines, TH1..TH4
KSEL,S,KP,,2,3
LSLK,S,1
*GET,L_TH1,LINE,0,NUM,MAX
LESIZE,L_TH1,,,n_theta,n_R_theta
KSEL,S,KP,,203
KSEL,A,KP,,103
LSLK,S,1
*GET,L_TH2,LINE,0,NUM,MAX

```

```

LESIZE,L_TH2,,,n_theta,n_R_theta
KSEL,S,KP,,202
KSEL,A,KP,,102
LSLK,S,1
*GET,L_TH3,LINE,0,NUM,MAX
LESIZE,L_TH3,,,n_theta,n_R_theta
KSEL,S,KP,,5,6
LSLK,S,1
*GET,L_TH4,LINE,0,NUM,MAX
LESIZE,L_TH4,,,n_theta,n_R_theta

! set line variable name and assign nodes on LXbase line
KSEL,S,KP,,1,2
LSLK,S,1
*GET,L_LXbase,LINE,0,NUM,MAX
LESIZE,L_LXbase,,,n_lx,1/n_R_lx

! set line variable name and assign nodes on LXmid line
KSEL,S,KP,,11
KSEL,A,KP,,203
LSLK,S,1
*GET,L_LXmid,LINE,0,NUM,MAX
LESIZE,L_LXmid,,,n_lx,1/n_R_lx

! set line variable name and assign nodes on LXtop line
KSEL,S,KP,,21
KSEL,A,KP,,23
LSLK,S,1
*GET,L_LXtop,LINE,0,NUM,MAX
LESIZE,L_LXtop,,,n_lx,n_R_lx

! set line variable name and assign nodes on MXtop line
KSEL,S,KP,,23
KSEL,A,KP,,25
LSLK,S,1
*GET,L_MXtop,LINE,0,NUM,MAX
LESIZE,L_MXtop,,,n_mx,n_R_mx

! set line variable name and assign nodes on HXbase line
KSEL,S,KP,,6,7
LSLK,S,1
*GET,L_HXbase,LINE,0,NUM,MAX
LESIZE,L_HXbase,,,n_hx,n_R_hx

! set line variable name and assign nodes on HXmid line
KSEL,S,KP,,202
KSEL,A,KP,,17
LSLK,S,1
*GET,L_HXmid,LINE,0,NUM,MAX
LESIZE,L_HXmid,,,n_hx,n_R_hx

! set line variable name and assign nodes on HXtop line
KSEL,S,KP,,25
KSEL,A,KP,,27
LSLK,S,1
*GET,L_HXtop,LINE,0,NUM,MAX
LESIZE,L_HXtop,,,n_hx,1/n_R_hx

! set line variable name and assign nodes on LYinlet line
KSEL,S,KP,,1

```

```

KSEL,A,KP,,11
LSLK,S,1
*GET,L_LYinlet,LINE,0,NUM,MAX
LESIZE,L_LYinlet,,,n_ly,n_R_ly

! set line variable name and assign nodes on HYinlet line
KSEL,S,KP,,11
KSEL,A,KP,,21
LSLK,S,1
*GET,L_HYinlet,LINE,0,NUM,MAX
*IF,n_R_hy,GE,0,THEN
    LESIZE,L_HYinlet,,,n_hy,1/n_R_hy
*ELSE
    LESIZE,L_HYinlet,,,n_hy,n_R_hy
*ENDIF

! set line variable name and assign nodes on HYmid1 line
KSEL,S,KP,,203
KSEL,A,KP,,23
LSLK,S,1
*GET,L_HYmid1,LINE,0,NUM,MAX
LESIZE,L_HYmid1,,,n_hy,n_R_hy

! set line variable name and assign nodes on HYmid2 line
KSEL,S,KP,,202
KSEL,A,KP,,25
LSLK,S,1
*GET,L_HYmid2,LINE,0,NUM,MAX
LESIZE,L_HYmid2,,,n_hy,n_R_hy

! set line variable name and assign nodes on LYoutlet line
KSEL,S,KP,,7
KSEL,A,KP,,17
LSLK,S,1
*GET,L_LYoutlet,LINE,0,NUM,MAX
LESIZE,L_LYoutlet,,,n_ly,n_R_ly

! set line variable name and assign nodes on HYoutet line
KSEL,S,KP,,17
KSEL,A,KP,,27
LSLK,S,1
*GET,L_HYoutet,LINE,0,NUM,MAX
LESIZE,L_HYoutet,,,n_hy,n_R_hy

/COM *** STEP 2.4 : FLUID ELEMENT TYPE DEFINED
*****

ET,1,FLUID141                                ! Element type 2D fluid #141
*IF,c_ALE,EQ,1,THEN
    KEYOPT,1,4,1                                ! Include if ALE meshing is turned on
w/FLDA,SOLU,ALE,TRUE command
*ENDIF

/COM *** STEP 2.5 : FLUID TYPES, CONSTANTS AND PROPERTIES ENABLED
*****

TYPE,1                                ! Enable Element Type 1
! No material or real constants are defined for fluid analysis. Fluid
properties
! are defined with the FLDATA command below.

```

```
/COM *** STEP 2.6 : FLUID MESH CREATED
```

```
*****
```

```
MSHAPE,0,2D          ! Specifies 2D quad elements
MSHKEY,1              ! Specifies mapped meshing
AMESH,1,8,1          ! Mesh areas 1 thru 8
```

```
ALLSEL,ALL
SAVE
```

```
/COM *** STEP 2.7 : FLUID BOUNDARY CONDITIONS
```

```
*****
```

```
! gravity force simulated thru application of positive y acceleration
!ACEL,,9810           ! g = 9810 mm/s^2 = 9.81 m/s^2
```

```
! Apply Vennard eqn to velocity distribution
```

```
*IF,c_INVEL,EQ,1,THEN
  ! Velocity distribution applied to inlet nodes
  LSEL,S,LINE,,L_LYinlet
  LSEL,A,LINE,,L_HYinlet
  NSLL,S,1
  *GET,n_inletLow,NODE,,NUM,MIN          ! lowest node number
  *GET,n_inletHigh,NODE,,NUM,MAX         ! highest node number
  ! checks Y loc of node and applies velocity distribution
  *DO,n_index,n_inletLow,n_inletHigh,1
    *IF,NSEL(n_index),EQ,1,THEN
      *IF,NY(n_index),EQ,0,THEN
        f_vxNode = 0
      *ELSEIF,NY(n_index),EQ,g_HC
        f_vxNode = 0
      *ELSEIF,NY(n_index),GT,g_HC/2
        f_vxNode = (f_vx*(f_ff)**(1/2))*(2*LOG10((g_HC-
NY(n_index))/(g_HC/2))+1.32)+f_vx
      *ELSEIF,NY(n_index),LE,g_HC/2
        f_vxNode =
(f_vx*(f_ff)**(1/2))*(2*LOG10(NY(n_index)/(g_HC/2))+1.32)+f_vx
      *ENDIF
      f_vy = 0
      D,n_index,VX,f_vxNode
      D,n_index,VY,f_vyNode
    *ENDIF
  *ENDDO
*ENDIF
```

```
! Apply uniform velocity distribution
```

```
*IF,c_INVEL,EQ,0,THEN
  ! Velocities in X & Y applied to the inlet
  DL,L_LYinlet, ,VX,f_vx,1          ! Apply velocity in X direction
on line1_11
  DL,L_HYinlet, ,VX,f_vx,1          ! Apply velocity in X direction
on line11_21
  DL,L_LYinlet, ,VY,f_vy,1          ! Apply velocity in Y direction
on line1_11
  DL,L_HYinlet, ,VY,f_vy,1          ! Apply velocity in Y direction
on line11_21
*ENDIF
```

```
*IF,c_ALE,EQ,1,THEN
```

```

}
      ! Displacement of morphing mesh restricted at inlet
      DL,L_LYinlet, ,UX,0,1      ! UX direction on line1_11
      DL,L_HYinlet, ,UX,0,1      ! UX direction on
line11_21
      DL,L_LYinlet, ,UY,0,1      ! UY direction on line1_11
      DL,L_HYinlet, ,UY,0,1      ! UY direction on
line11_21
*ENDIF

```

```

! No Slip condition applied to top boundaries
DL,L_LXtop, ,VX,0,1      ! no slip X
DL,L_LXtop, ,VY,0,1      ! no slip Y
DL,L_MXtop, ,VX,0,1      ! no slip X
DL,L_MXtop, ,VY,0,1      ! no slip Y
DL,L_HXtop, ,VX,0,1      ! no slip X
DL,L_HXtop, ,VY,0,1      ! no slip Y

```

```

! Displacement of morphing mesh restricted at top boundaries
! NOTE: UX is not fixed because this is the direction of flow
*IF,c_ALE,EQ,1,THEN
      DL,L_LXtop, ,UY,0,1
      DL,L_MXtop, ,UY,0,1
      DL,L_HXtop, ,UY,0,1
*ENDIF

```

```

! No Slip condition applied to bottom boundaries
DL,L_LXbase, ,VX,0,1      ! no slip X
DL,L_LXbase, ,VY,0,1      ! no slip Y
DL,L_TH1, ,VX,0,1      ! no slip X
DL,L_TH1, ,VY,0,1      ! no slip Y
DL,L_TH4, ,VX,0,1      ! no slip X
DL,L_TH4, ,VY,0,1      ! no slip Y
DL,L_HXbase, ,VX,0,1      ! no slip X
DL,L_HXbase, ,VY,0,1      ! no slip Y

```

```

! Displacement of morphing mesh restricted at bottom boundaries
*IF,c_ALE,EQ,1,THEN
      DL,L_LXbase, ,UX,0,1
      DL,L_LXbase, ,UY,0,1
      DL,L_TH1, ,UX,0,1
      DL,L_TH1, ,UY,0,1
      DL,L_TH4, ,UX,0,1
      DL,L_TH4, ,UY,0,1
      DL,L_HXbase, ,UX,0,1
      DL,L_HXbase, ,UY,0,1
*ENDIF

```

```

! No Slip condition applied to obstacle boundaries
DL,L_R11, ,VX,0,1      ! no slip X
DL,L_R11, ,VY,0,1      ! no slip Y
DL,L_R12, ,VX,0,1      ! no slip X
DL,L_R12, ,VY,0,1      ! no slip Y
DL,L_R13, ,VX,0,1      ! no slip X
DL,L_R13, ,VY,0,1      ! no slip Y

```

```

! Pressure boundary at outlet
DL,L_LYoutlet, ,PRES,0,1
DL,L_HYoutlet, ,PRES,0,1

```

```

! Displacement of morphing mesh restricted at outlet boundaries

```



```

*IF,c_ALE,EQ,1,THEN
    DL,L_LYoutlet, ,UX,0,1
    DL,L_LYoutlet, ,UY,0,1
    DL,L_HYoutlet, ,UX,0,1
    DL,L_HYoutlet, ,UY,0,1
*ENDIF

/COM *** STEP 2.8 : FLUID PROPERTIES DEFINED
*****

FLDATA,PROT,DENS,CONSTANT           ! fluid density is constant
FLDATA,PROT,VISC,CONSTANT           ! fluid viscosity is constant
FLDATA,NOMI,DENS,f_ro                ! fluid density specified by
parameter ro
FLDATA,NOMI,VISC,f_mu                ! fluid viscosity specified by
parameter nu

ALLSEL,ALL
SAVE

FINISH

/COM *** STEP 2.9 : FLUID SOLUTION CONTROL
*****
/SOLUTION

! general fluid solution options/controls
FLDATA,SOLU,TRAN,TRUE                ! fluid solution is transient
FLDATA,ALGR,SEGR,SIMPLEN             ! enhanced SIMPLEN solution
algorithm
*IF,c_ALE,EQ,1,THEN
    FLDA,SOLU,ALE,TRUE                ! activates ALE mesh-morphing
*ENDIF

! Pressure and momentum equation settings
FLDATA,SOLU,FLOW,TRUE                ! solves pressure and momentum
eqns.
FLDATA,ADVM,MOME,SUPG                ! Use streamline upward
discretization on momentum eqn.
*IF,c_ALE,EQ,1,THEN
    FLDATA,QUAD,MOMD,2                ! Accuracy of quadrature increased for
ALE
    FLDATA,QUAD,MOMA,2
    FLDATA,QUAD,MOMS,2
    FLDATA,QUAD,PRSD,2
    FLDATA,QUAD,PRSS,2
*ENDIF

! Turbulence settings
*IF,c_TURB,EQ,1,THEN
    FLDATA,SOLU,TURB,TRUE             ! activates turbulence model
    FLDATA,TURB,WALL,EQLB             ! specifies 'Wall Equilibrium'
model for boundary layer
! FLDATA,METH,PRES,1                  ! Set TDM solver to default solver for
PRES equation
    FLDATA,ADVM,TURB,SUPG             ! Use streamline upward
discretization on turbulence eqn.
    FLDATA,OUTP,TAUW,TRUE             ! write wall shear to output
    FLDATA,OUTP,YPLU,TRUE             ! write turb Y+ to output

```

```

!      FLDATA,STAB,TURB,0.001                ! turbulence inertial relaxation
factor
!      FLDATA,RELX,VX,0.5                    ! Momentum equation relaxation factor,
VX (0.5 default)
!      FLDATA,RELX,VY,0.5                    ! Momentum equation relaxation factor,
VY (0.5 default)
!      FLDATA,RELX,PRES,0.5                  ! Momentum equation relaxation
factor, PRES (0.5 default)
!      FLDATA,TURB,RATI,100000              ! Increases initial effective
viscosity
!      FLDATA,MIR,MOME,0.75                  ! MIR for Momentum eqns to fix
neg. diag., (0 default)
!      FLDATA,MIR,TURB,1.0                  ! MIR for Turbulence eqns to fix neg.
diag., (0 default)
*ENDIF

```

```

ALLSEL,ALL
SAVE

```

```

FINISH

```

```

/PREP7

```

```

/COM *** STEP 3.0 : STRUCTURAL ANALYSIS SET-UP
*****

```

```

/COM *** STEP 3.1 : STRUCTURAL KEYPOINTS & LINES
*****

```

```

K,300,g_C1-g_box/2,0                ! ll hemi box
K,301,g_C1-g_box/2,g_box            ! ul hemi box
K,302,g_C1+g_box/2,g_box            ! ur hemi box
K,303,g_C1+g_box/2,0                ! lr hemi box
K,304,g_C1,0                        ! biofilm hemisphere center
K,305,g_C2,0                        ! biofilm hemisphere zero

! biofilm hemisphere generated
NUMSTR,KP,401                        ! KP start # at 401
NUMSTR,LINE,401                      ! LINE start # at 401
CIRCLE,304,g_R1,,305,180,3          ! creates kp's 401,402,403,404, lines
401,402,403

```

```

! define lines creating box on the interior of biofilm hemisphere
L,303,302                            ! line 404, hemi box right side
L,302,301                            ! line 405, hemi box top side
L,301,300                            ! line 406, hemi box left side
L,300,303                            ! line 407, hemi box bottom side

```

```

! define lines creating partitions within hemishpere
L,300,404                            ! line 408, box to hemi segment radial
(CW)
L,301,403                            ! line 409, box to hemi segment radial
(CW)
L,302,402                            ! line 410, box to hemi segment radial
(CW)
L,303,401                            ! line 411, box to hemi segment radial
(CW)

```

```

/COM *** STEP 3.2 : STRUCTURAL AREAS ***

```

```

NUMSTR,AREA,301                      ! AREA start # at 301

```



```

A,303,401,402,302      ! A301, hemi, right
A,301,302,402,403      ! A302, hemi, center
A,404,300,301,403      ! A303, hemi, left
A,300,303,302,301      ! A304, hemi, box

```

```

/COM *** STEP 3.3 : STRUCTURAL NODES ***

```

```

! set line variable name and assign nodes on line L_BOX1
KSEL,S,KP,,300,301
LSLK,S,1
*GET,L_BOX1,LINE,0,NUM,MAX
LESIZE,L_BOX1,,,n_box_x,n_R_box_x

```

```

! set line variable name and assign nodes on line L_BOX2
KSEL,S,KP,,301,302
LSLK,S,1
*GET,L_BOX2,LINE,0,NUM,MAX
LESIZE,L_BOX2,,,n_box_y,n_R_box_y

```

```

! set line variable name and assign nodes on line L_BOX3
KSEL,S,KP,,302,303
LSLK,S,1
*GET,L_BOX3,LINE,0,NUM,MAX
LESIZE,L_BOX3,,,n_box_x,n_R_box_x

```

```

! set line variable name and assign nodes on line L_BOX4
KSEL,S,KP,,300,303,3
LSLK,S,1
*GET,L_BOX4,LINE,0,NUM,MAX
LESIZE,L_BOX4,,,n_box_y,n_R_box_y

```

```

! set line variable name and assign nodes on line L_S11
KSEL,S,KP,,403,404
LSLK,S,1
*GET,L_S11,LINE,0,NUM,MAX
LESIZE,L_S11,,,n_box_x,n_R_box_x

```

```

! set line variable name and assign nodes on line L_S12
KSEL,S,KP,,402,403
LSLK,S,1
*GET,L_S12,LINE,0,NUM,MAX
LESIZE,L_S12,,,n_box_y,n_R_box_y

```

```

! set line variable name and assign nodes on line L_S13
KSEL,S,KP,,401,402
LSLK,S,1
*GET,L_S13,LINE,0,NUM,MAX
LESIZE,L_S13,,,n_box_x,n_R_box_x

```

```

! set line variable name and assign nodes on line L_PT1
KSEL,S,KP,,404
KSEL,A,KP,,300
LSLK,S,1
*GET,L_PT1,LINE,0,NUM,MAX
LESIZE,L_PT1,,,n_box_pt,n_R_box_pt

```

```

! set line variable name and assign nodes on line L_PT2
KSEL,S,KP,,403
KSEL,A,KP,,301
LSLK,S,1

```

```

*GET,L_PT2,LINE,0,NUM,MAX
LESIZE,L_PT2,,,n_box_pt,n_R_box_pt

! set line variable name and assign nodes on line L_PT3
KSEL,S,KP,,,402
KSEL,A,KP,,,302
LSLK,S,1
*GET,L_PT3,LINE,0,NUM,MAX
LESIZE,L_PT3,,,n_box_pt,n_R_box_pt

! set line variable name and assign nodes on line L_PT4
KSEL,S,KP,,,401
KSEL,A,KP,,,303
LSLK,S,1
*GET,L_PT4,LINE,0,NUM,MAX
LESIZE,L_PT4,,,n_box_pt,n_R_box_pt

/COM *** STEP 3.4 : STRUCTURAL ELEMENT TYPE DEFINED
*****

ET,2,PLANE183,,,3                                ! Element type 2D 4 node solid
#42 (Element Type 2)
R,2,.5,                                           ! Plane thickness (Real Constants
2)

/COM *** STEP 3.5 : STRUCTURAL PROPERTIES DEFINED
*****

MP,EX,2,s_E0                                     ! Youngs Modulus (Material 2)
MP,PRXY,2,s_PR                                  ! Poisson ratio, major (Material
2)
MP,DENS,2,s_DE                                   ! Mass density
TB,PRONY,2,,2,SHEAR                             ! Creates a table of Prony coefficients
(Material 2)
TBDATA,1,s_A1,s_TAU1,s_A2,s_TAU2               ! Fills in the values of the coeff.

/COM *** STEP 3.6 : STRUCTURAL TYPES, CONSTANTS AND PROPERTIES ENABLED
*****

TYPE,2                                           ! Enable Element Type 2
REAL,2                                           ! Enable Real Constants 2
MAT,2                                           ! Enable Material 2

/COM *** STEP 3.7 : STRUCTURAL ELEMENTS CREATED
*****

! Mesh generated
MSHAPE,0,2D                                     ! Specifies 2D quad elements
MSHKEY,1                                         ! Specifies mapped meshing
AMESH,301,304,1                                 ! Mesh areas

/COM *** STEP 3.8 : STRUCTURAL BOUNDARY CONDITIONS
*****

! gravity force simulated thru application of positive y acceleration
!ACEL,,9810                                     ! g = 9810 mm/s^2 = 9.81 m/s^2

! The following D commands fix the base of the structural hemisphere
LSEL,S,LINE,,L_PT1

```

```

LSEL,A,LINE,,L_BOX4
LSEL,A,LINE,,L_PT4
NSLL,S,1                      ! select nodes assoc. with selected lines
D,ALL,UX,0                    ! fix x displacement
D,ALL,UY,0                    ! fix y displacement

ALLSEL,ALL
SAVE

/COM *** STEP 4.0 : FLUID-STRUCTURAL INTERACTION
*****

/COM *** STEP 4.1 : FLAG FLUID-STRUCTURAL INTERFACES
*****

! The following SF command flags the fluid side of the boundary of the
hemisphere
LSEL,S,LINE,,L_R11
NSLL,S,1                      ! select nodes on these lines
SF,ALL,FSIN,1                 ! FSIN flag 1 on fluid side nodes

LSEL,S,LINE,,L_R12
NSLL,S,1                      ! select nodes on these lines
SF,ALL,FSIN,2                 ! FSIN flag 2 on fluid side nodes

LSEL,S,LINE,,L_R13
NSLL,S,1                      ! select nodes on these lines
SF,ALL,FSIN,3                 ! FSIN flag 3 on fluid side nodes

ALLSEL,ALL

! The following SF command flags the structural side of the boundary of the
hemisphere
LSEL,S,LINE,,L_S11
NSLL,S,1                      ! select nodes on these lines
SF,ALL,FSIN,1                 ! FSIN flag 1 on structural side nodes

LSEL,S,LINE,,L_S12
NSLL,S,1                      ! select nodes on these lines
SF,ALL,FSIN,2                 ! FSIN flag 2 on structural side nodes

LSEL,S,LINE,,L_S13
NSLL,S,1                      ! select nodes on these lines
SF,ALL,FSIN,3                 ! FSIN flag 3 on structural side nodes

ALLSEL,ALL
SAVE

/COM *** STEP 4.2 : FLUID-STRUCTURAL INTERACTION SOLUTION OPTIONS
*****

! Basic FSI Analysis Options
FSAN,ON                       ! FSI switched on
FSOR,FLUID,SOLID              ! FSI solution order
FSTR,FLUID,                   ! Fluid soln is transient
FSTR,SOLID,                   ! Solid soln is transient
FSIN,CONS                     ! Interpolate conservative values

FLDA,OUTP,TAUW,TRUE
FLDA,OUTP,YPLU,TRUE

```

```

! End time, Load time and time increments
FSTI,c_time,c_loadtime          ! End time, load time for FSI
analysis
FSDT,c_dtFSI                    ! time step increment for FSI analysis
DELTIM,c_dtS,,,OFF              ! time step increment for solid
analysis
FLDATA,TIME,STEP,c_dtF          ! time step increment for fluid
analysis

! Fluid convergence criteria
FLDATA,TERM,VX,0.001            ! conv. on VX term (0.01 default)

! Maximum number of iterations
! FLDATA,ITER,EXEC,c_iF          ! number of fluid sol'n global
iterations (static analysis)
FLDATA,TIME,GLOB,c_iF           ! number of fluid sol'n global
iterations (transient analysis)
FSIT,c_iFSI                     ! FSI stagger iterations

! FSI Convergence values
FSCO,FX,0.00001                 ! convergence value
FSCO,FY,0.00001                 ! convergence value
FSCO,FZ,0.00001                 ! convergence value
FSCO,UX,0.00001                 ! convergence value
FSCO,UY,0.00001                 ! convergence value
FSCO,UZ,0.00001                 ! convergence value

! Output frequency
FSOU,ALL                         ! Output at every time step

! Relaxation values
FSRE,ALL,0.6                    ! All relaxation values set

ALLSEL,ALL
SAVE

*IF,c_SOLV,EQ,0,THEN              ! Exit if SOLU switch is off.
    FINISH
    /EXIT
    /EOF
*ENDIF

FINISH

/COM *** STEP 5.0 : SOLUTION
*****

/SOLUTION

/COM *** STEP 5.1 : DEFINE MONITOR FILE DOFS
*****

! Select the node at the intersection of L_S12, L_S13
ESEL,S,TYPE,,2                  ! select all type 2 elements
NSLE,S                          ! select all these nodes
n_mntr = NODE(g_C1+g_R1*COS(Pi/3),g_R1*SIN(Pi/3),0) ! return node # at x,y,z

! set solution monitoring DOFs
MONITOR,2,n_mntr,UX              ! track max UX

```



```

MONITOR,3,n_mntr,UY                      ! track max UY
! var 1 is unchanged. The default is CPU time.

/COM *** STEP 5.2 : SOLVE MODEL
*****

ALLSEL,ALL
SOLVE                                     ! CARPE MAMILLA !!!

ALLSEL,ALL
SAVE

FINISH

/COM *** STEP 6.0 DATABASE RESULTS POST PROCESSING
*****

/POST1

/COM *** STEP 6.1 SET STANDARD VIEW
*****

ALLSEL,ALL

! load results
RESUME,%m_job%,db
INRES,ALL
FILE,%m_job%,rst
SET,LAST
ALLSEL,ALL
! These commands center the view on the hemisphere and plot the stresses
/DSCALE,ALL,1.0                          ! set displacement scaling to 1.0
INRES,ALL                                ! retrieve ALL structural data
FILE                                      ! structural data file location
SET,LAST                                 ! load the last data set
/VIEW,1,0,0,1.0000                       ! view from the z dir
/FOC,1,g_C1+g_R1/2,g_R1/2,0              ! center on hemi
/DIST,1,0.5                              ! zoom scale
PLNSOL,S,EQV,2                           ! plot von Mises stresses (dummy
plot command)

ALLSEL,ALL
SAVE

/COM *** STEP 6.2 PRONY PARAMETER FILE OUTPUT
*****

! The following generates a text file for recording Prony parameters:
p_filename = STRCAT(m_job,'_prony')      ! builds filename as string
parameter
*CFOPEN,%p_filename%,txt
*VWRITE,s_E0,s_PR
(2X,'E0:',F16.8,2X,'PR:',F16.8)
*VWRITE,s_A1,s_TAU1
(2X,'A1:',F16.8,2X,'Tau1:',F16.8)
*VWRITE,s_A2,s_TAU2
(2X,'A2:',F16.8,2X,'Tau2:',F16.8)
*VWRITE,s_G1,s_h1,s_G2,s_h2
(2X,'G1:',F16.8,2X,'h1:',F16.8,2X,'G2:',F16.8,2X,'h2:',F16.8)
*CFCLOSE

```

```
ALLSEL,ALL
SAVE
```

```
/COM *** STEP 6.3 DEFORMED CENTROID CALCULATION
*****
```

```
! load results
RESUME,%m_job%,db
INRES,ALL
FILE,%m_job%,rst
```

```
p_numSets = NINT(c_Time)           ! number of sets in results file
```

```
*DIM,p_SetCentroid,ARRAY,p_numSets+1,4      ! set#, time, Xcentroid,
Ycentroid
```

```
! determine which elements comprise the biofilm hemisphere
```

```
ALLSEL,ALL
ESEL,S,TYPE,,2                          ! select all structural (type 2)
elements
*GET,p_eMax,ELEM,,NUM,MAX                ! get max element number in selected
set
*GET,p_eMin,ELEM,,NUM,MIN                ! get min element number in selected
set
*GET,p_eCount,ELEM,,COUNT              ! get number of structural elements
```

```
*DO,p_Set,1,p_numSets,1
  SET,NEAR,,,,p_Set                    ! set to be read
```

```
  ! reset variables that change thru each loop
  p_elNum = 0                          ! reset p_elNum
  *SET,p_eCentroid,                    ! delete previous
p_eCentroid array
  *DIM,p_eCentroid,ARRAY,p_eCount+1,6  ! el #, cent1_x, cent1_y,
cent2_x, cent2_y, area
```

```
  ! calculate the centroid of the deformed shape for the current set
  *DO,i,1,p_eCount                      ! loop thru all el. & calc
orig. and deform. centroid
```

```
    ! define current element number
    p_elNum = i + p_eMin - 1
    ! calculate deformed node locations for current element
    p_dx1 = NX(NELEM(p_elNum,1))+UX(NELEM(p_elNum,1))
    p_dy1 = NY(NELEM(p_elNum,1))+UY(NELEM(p_elNum,1))
    p_dx2 = NX(NELEM(p_elNum,2))+UX(NELEM(p_elNum,2))
    p_dy2 = NY(NELEM(p_elNum,2))+UY(NELEM(p_elNum,2))
    p_dx3 = NX(NELEM(p_elNum,3))+UX(NELEM(p_elNum,3))
    p_dy3 = NY(NELEM(p_elNum,3))+UY(NELEM(p_elNum,3))
    p_dx4 = NX(NELEM(p_elNum,4))+UX(NELEM(p_elNum,4))
    p_dy4 = NY(NELEM(p_elNum,4))+UY(NELEM(p_elNum,4))
    ! calculate new centroid based on old centroid and displacements
    (less accurate)
    p_CentX =
CENTRX(p_elNum)+(UX(NELEM(p_elNum,1))+UX(NELEM(p_elNum,2))+UX(NELEM(p_elNum,3))
+UX(NELEM(p_elNum,4)))/4
    p_CentY =
CENTRY(p_elNum)+(UY(NELEM(p_elNum,1))+UY(NELEM(p_elNum,2))+UY(NELEM(p_elNum,3))
+UY(NELEM(p_elNum,4)))/4
    ! calculate new area based on deformed node locations
```

```

      p_Area1 = (1/2)*(((p_dx3-p_dx1)*(p_dy2-p_dy1)-(p_dy3-
p_dy1)*(p_dx2-p_dx1))**2)**(1/2))
      p_Area2 = (1/2)*(((p_dx4-p_dx1)*(p_dy3-p_dy1)-(p_dy4-
p_dy1)*(p_dx3-p_dx1))**2)**(1/2))
      p_AreaT = p_Area1 + p_Area2
      ! write out element data to centroid array
      p_eCentroid(i,1) = p_elNum
      p_eCentroid(i,2) = CENTRX(p_elNum)
      p_eCentroid(i,3) = CENTRY(p_elNum)
      p_eCentroid(i,4) = p_CentX
      p_eCentroid(i,5) = p_CentY
      p_eCentroid(i,6) = p_AreaT
*ENDDO

      ! loop thru all elements to calculate the composite centroid location
      p_AreaSUM = 0                      ! sum of all element areas
      p_xAreaSUM = 0                    ! sum of all element x
      centroid*area products
      p_yAreaSUM = 0                    ! sum of all element y
      centroid*area products
      *DO,i,1,p_eCount                  ! loop thru all el. for
      summations
      ! define current element number
      p_elNum = i + p_eMin - 1
      ! sum area
      p_AreaSUM = p_AreaSUM + p_eCentroid(i,6)
      ! sum x area product
      p_xAreaSUM = p_xAreaSUM+(p_eCentroid(i,4)*p_eCentroid(i,6))
      ! sum y area product
      p_yAreaSUM = p_yAreaSUM+(p_eCentroid(i,5)*p_eCentroid(i,6))
*ENDDO
      p_xCent = p_xAreaSUM / p_AreaSUM      ! x centroid of composite
shape
      p_yCent = p_yAreaSUM / p_AreaSUM      ! y centroid of composite
shape

      ! write set centroid to array
      p_SetCentroid(p_Set,1) = p_Set/c_dtFSI      ! set number
      p_SetCentroid(p_Set,2) = p_Set              ! time
      p_SetCentroid(p_Set,3) = p_xCent            ! x centroid of deformed
shape
      p_SetCentroid(p_Set,4) = p_yCent            ! y centroid of deformed
shape

*ENDDO

! write out array of centroids by data set
p_filename = STRCAT(m_job,'_centroid') ! builds filename as string parameter
*CFOPEN,%p_filename%,txt
*VWRITE,p_SetCentroid(1,1),p_SetCentroid(1,2),p_SetCentroid(1,3),p_SetCentroid(
1,4)
(2X,F6.1,'      ',F8.3,'      ',F16.8,'      ',F16.8)
*CFCLOS

ALLSEL,ALL
SAVE

/COM *** STEP 6.4 RESULTANT SURFACE FORCE CALCULATION
*****

```

```

! load results
RESUME,%m_job%,db
INRES,ALL
FILE,%m_job%,rfl

p_numSets = NINT(c_Time)           ! number of sets in results file

*DIM,p_SetResultant,ARRAY,p_numSets+1,11      ! set#, time, FxP, FyP, FxT, FyT,
FxR, FyR, MzP..

ALLSEL,ALL

*DO,p_Set,1,p_numSets,1
  SET,NEAR,,,,p_Set                ! set to be read

  ! reset variables that change thru each loop
  p_FxR = 0
  p_FyR = 0
  p_MzR = 0
  ! select FSI boundary nodes on fluid side
  LSEL,S,LINE,,L_R11
  LSEL,A,LINE,,L_R12
  LSEL,A,LINE,,L_R13
  NSLL,S,1
  ! integrate pressure to calculate FX,FX,MX,MY
  INTSRF,PRES                      ! integrate nodal results for
pressure
  *GET,p_FxP,INTSRF,,PRES,FX      ! retrieve X component of resultant
pressure force
  *GET,p_FyP,INTSRF,,PRES,FY      ! retrieve Y component of resultant
pressure force
  *GET,p_MzP,INTSRF,,PRES,MZ      ! retrieve Z component of pressure
moment around origin
  ! integrate wall shear to calculate FX,FX,MX,MY
  INTSRF,TAUW                      ! integrate nodal results for
wall shear
  *GET,p_FxT,INTSRF,,TAUW,FX      ! retrieve X component of resultant
wall shear force
  *GET,p_FyT,INTSRF,,TAUW,FY      ! retrieve Y component of resultant
wall shear force
  *GET,p_MzT,INTSRF,,TAUW,MZ      ! retrieve Z component of wall shear
moment around origin
  ! add to calculate resultant FX,FX,MX,MY
  p_FxR = p_FxP + p_FxT          ! resultant of shear and pressure
in x dir
  p_FyR = p_FyP + p_FyT          ! resultant of shear and pressure
in y dir
  p_MzR = p_MzP + p_MzT          ! resultant of moment around z
axis

  ! write set resultant to array
  p_SetResultant(p_Set,1) = p_Set/c_dtFSI      ! set number
  p_SetResultant(p_Set,2) = p_Set              ! time
  p_SetResultant(p_Set,3) = p_FxP              ! Pressure force in x dir
  p_SetResultant(p_Set,4) = p_FyP              ! Pressure force in y dir
  p_SetResultant(p_Set,5) = p_FxT              ! Shear force in x dir
  p_SetResultant(p_Set,6) = p_FyT              ! Shear force in y dir
  p_SetResultant(p_Set,7) = p_FxR              ! Resultant force in x dir
  p_SetResultant(p_Set,8) = p_FyR              ! Resultant force in y dir
  p_SetResultant(p_Set,9) = p_MzP              ! Pressure moment in z dir

```



```

      p_SetResultant(p_Set,10) = p_MzT          ! Shear moment in z dir
      p_SetResultant(p_Set,11) = p_MzR          ! Resultant moment in z
dir
! write out array of resultants by data set
p_filename = STRCAT(m_job,'_resultant')         ! builds filename as
string parameter
*CFOPEN,%p_filename%,txt
*VWRITE,p_SetResultant(1,1),p_SetResultant(1,2),p_SetResultant(1,3),p_SetResultant(1,4),p_SetResultant(1,5),p_SetResultant(1,6),p_SetResultant(1,7),p_SetResultant(1,8),p_SetResultant(1,9),p_SetResultant(1,10),p_SetResultant(1,11)
2X,F6.1,' ',F8.3,' ',E16.10,' ',E16.10,' ',E16.10,' ',E16.10,' ',E16.10,' '
'E16.10,' ',E16.10,' ',E16.10,' ',E16.10,' ',E16.10)
*CFCLOS

ALLSEL,ALL
SAVE

/COM *** STEP 6.5 REACTION FORCE CALCULATION
*****

! load results
RESUME,%m_job%,db
INRES,ALL
FILE,%m_job%,rst

p_numSets = NINT(c_Time)           ! number of sets in results file

*DIM,p_SetReaction,ARRAY,p_numSets+1,5 ! set#, time, FxB, FyB, MzB

ALLSEL,ALL

*DO,p_Set,1,p_numSets,1
    SET,NEAR,,,p_Set               ! set to be read

    ! reset variables that change thru each loop
    *SET,p_nReacs,

    ! select lines on base of hemisphere
    LSEL,S,LINE,,L_PT1
    LSEL,A,LINE,,L_BOX4
    LSEL,A,LINE,,L_PT4
    NSLL,S,1                        ! select nodes on these lines
    *GET,p_nMax,NODE,,NUM,MAX       ! get max node number in selected set
    *GET,p_nMin,NODE,,NUM,MIN       ! get min node number in selected set
    *GET,p_nCount,NODE,,COUNT     ! get count of nodes in selected set
    *DIM,p_nReacs,ARRAY,p_nCount+1,6 ! dim. Reacs array - node
# , X,Y,FX,FY,MZ (reactions)
    ! loop thru all selected nodes
    p_Row = 0                      ! initialize row count
    *DO,i,p_nMin,p_nMax
        *IF,NSEL(i),EQ,1,THEN
            p_Row = p_Row + 1
            p_nReacs(p_Row,1) = i
            p_nReacs(p_Row,2) = NX(i)
            p_nReacs(p_Row,3) = NY(i)
            *GET,p_nReacs(p_Row,4),NODE,i,Rf,FX
            *GET,p_nReacs(p_Row,5),NODE,i,Rf,FY

```

```

*GET,p_nReacs(p_Row,6),NODE,i,RF,MZ
*ENDIF
*ENDDO

! Sum reactions and moments (rxns) around origin
p_FxB = 0 ! initialize resultant reaction
moment at base nodes
p_FyB = 0 ! initialize resultant reaction Fx
at base nodes (trans to origin.)
p_MzB = 0 ! initialize resultant reaction Fy
at base nodes (trans to origin.)
*DO,i,1,p_nCount
p_FxB = p_FxB + p_nReacs(i,4)
p_FyB = p_FyB + p_nReacs(i,5)
p_MzB = p_MzB + p_nReacs(i,6)+p_nReacs(i,5)*p_nReacs(i,2)
*ENDDO

! write set reactions to array
p_SetReaction(p_Set,1) = p_Set/c_dtFSI ! set number
p_SetReaction(p_Set,2) = p_Set ! time
p_SetReaction(p_Set,3) = p_FxB ! reaction force in x dir.
p_SetReaction(p_Set,4) = p_FyB ! reaction force in y dir.
p_SetReaction(p_Set,5) = p_MzB ! reaction moment in z
dir.

*ENDDO

! write out array of reactions by data set
p_filename = STRCAT(m_job,'_reaction') ! builds filename as
string parameter
*CFOPEN,%p_filename%,txt
*VWRITE,p_SetReaction(1,1),p_SetReaction(1,2),p_SetReaction(1,3),p_SetReaction(
1,4),p_SetReaction(1,5)
(2X,F6.1,' ',F8.3,' ',E16.10,' ',E16.10,' ',E16.10)
*CFCLOS

ALLSEL,ALL
SAVE

FINISH

/COM *** STEP 7.0 TIME HISTORY POST PROCESSING
*****

/POST26

FINISH

ALLSEL,ALL
SAVE

/EOF
/COM *** END OF FILE
*****

```

APPENDIX H

SAMPLE CALCULATION OF A QUADRANGLE CENTROID

### Centroid of a Quadrangle

For an arbitrarily oriented quadrangle defined by four CCW ordered points (1, 2, 3 and 4) :

The area,  $A_1$ , of the triangle defined by points 1, 2, 3 is:

$$A_1(i_1, i_2, i_3, j_1, j_2, j_3) := \left(\frac{1}{2}\right) \cdot \sqrt{[(i_3 - i_1) \cdot (j_2 - j_1) - (j_3 - j_1) \cdot (i_2 - i_1)]^2}$$

and the area,  $A_2$ , of the triangle defined by points 1, 3, 4 is:

$$A_2(i_1, i_3, i_4, j_1, j_3, j_4) := \left(\frac{1}{2}\right) \cdot \sqrt{[(i_4 - i_1) \cdot (j_3 - j_1) - (j_4 - j_1) \cdot (i_3 - i_1)]^2}$$

where  $i$  and  $j$  represent the  $x$  and  $y$  coordinates of those points

then the total area of the quadrangle is,  $A_T$  :

$$A_T(i_1, i_2, i_3, i_4, j_1, j_2, j_3, j_4) := A_1(i_1, i_2, i_3, j_1, j_2, j_3) + A_2(i_1, i_3, i_4, j_1, j_3, j_4)$$

$$A_T(i_1, i_2, i_3, i_4, j_1, j_2, j_3, j_4) \rightarrow \frac{1}{2} \cdot \left[ [(i_3 - i_1) \cdot (j_2 - j_1) - (j_3 - j_1) \cdot (i_2 - i_1)]^2 \right]^{\frac{1}{2}} + \frac{1}{2} \cdot \left[ [(i_4 - i_1) \cdot (j_3 - j_1) - (j_4 - j_1) \cdot (i_3 - i_1)]^2 \right]^{\frac{1}{2}}$$

The centroid,  $G$ , of triangle 1 defined by points 1, 2, and 3 is :

$$G_{y1}(j_1, j_2, j_3) := \left(\frac{1}{3}\right) \cdot (j_1 + j_2 + j_3) \quad G_{x1}(i_1, i_2, i_3) := \left(\frac{1}{3}\right) \cdot (i_1 + i_2 + i_3)$$

The centroid,  $G$ , of triangle 2 defined by points 1, 3, and 4 is :

$$G_{x2}(i_1, i_3, i_4) := \left(\frac{1}{3}\right) \cdot (i_1 + i_3 + i_4) \quad G_{y2}(j_1, j_3, j_4) := \left(\frac{1}{3}\right) \cdot (j_1 + j_3 + j_4)$$

Then the centroid of the quadrangle is :

$$G_x(i_1, i_2, i_3, i_4, j_1, j_2, j_3, j_4) := \left( \frac{A_1(i_1, i_2, i_3, j_1, j_2, j_3)}{A_T(i_1, i_2, i_3, i_4, j_1, j_2, j_3, j_4)} \right) \cdot G_{x1}(i_1, i_2, i_3) + \left( \frac{A_2(i_1, i_3, i_4, j_1, j_3, j_4)}{A_T(i_1, i_2, i_3, i_4, j_1, j_2, j_3, j_4)} \right) \cdot G_{x2}(i_1, i_3, i_4)$$

$$Gy(i_1, i_2, i_3, i_4, j_1, j_2, j_3, j_4) := \left( \frac{A_1(i_1, i_2, i_3, j_1, j_2, j_3)}{A_T(i_1, i_2, i_3, i_4, j_1, j_2, j_3, j_4)} \right) \cdot Gy_1(j_1, j_2, j_3) + \left( \frac{A_2(i_1, i_3, i_4, j_1, j_3, j_4)}{A_T(i_1, i_2, i_3, i_4, j_1, j_2, j_3, j_4)} \right) \cdot Gy_2(j_1, j_3, j_4)$$

Example of centroid calculation of quadrangle :

The four points defining the quadrangle are:

$$x_4 := 0 \quad y_4 := 0$$

$$x_1 := 2 \quad y_1 := 0$$

$$x_2 := 3 \quad y_2 := 1$$

$$x_3 := 0 \quad y_3 := 2$$

The area are :

$$A_1(x_1, x_2, x_3, y_1, y_2, y_3) = 2$$

$$A_2(x_1, x_3, x_4, y_1, y_3, y_4) = 2$$

$$A_T(x_1, x_2, x_3, x_4, y_1, y_2, y_3, y_4) = 4$$

The centroids of the two triangles are :

$$Gx_1(x_1, x_2, x_3) = 1.667$$

$$Gy_1(y_1, y_2, y_3) = 1$$

Triangle 1 : pt.

$$Gx_2(x_1, x_3, x_4) = 0.667$$

$$Gy_2(y_1, y_3, y_4) = 0.667$$

Triangle 2 : pt.

Then the centroid of the combined shape or quadrangle is :

$$Gx(x_1, x_2, x_3, x_4, y_1, y_2, y_3, y_4) = 1.167$$

$$Gy(x_1, x_2, x_3, x_4, y_1, y_2, y_3, y_4) = 0.833$$

If the quadrangle comprise a larger composite shape, the centroid of the composite shape is :

$$GX = \frac{\sum Gx^i A_T^i}{\sum A_T^i} \qquad GY = \frac{\sum Gy^i A_T^i}{\sum A_T^i}$$

MONTANA STATE UNIVERSITY - BOZEMAN



3 1762 10465606 9