Investigation of Nano-Scale, Self-Assembled, Polymeric Systems by Atomic Force Microscopy

by

James K. Li

A thesis submitted in conformity with the requirements for the degree of Doctor of Philosophy
Department of Chemistry
University of Toronto

© Copyright by James K. Li 2010
Atomic Force Microscopy of Nano-Scale, Self-Assembled, Polymeric Systems

James K. Li

Doctor of Philosophy

Department of Chemistry
University of Toronto

2010

Abstract

The atomic force microscope (AFM) was used to study a series of self-assembled systems: alkanethiol self-assembled monolayer (SAM), diblock copolymer thin film, solid supported lipid bilayer membrane, and microgel with double interpenetrating polymer network.

In the first system, packing and restructuring of self-assembled monolayers as exhibited by several alkanethiol systems (1-hexanethiol, 1-decanethiol, 11-ferroceny1-1-undecanethiol) is demonstrated using conducting probe AFM (CP-AFM). Pressure is induced by an AFM tip, and simultaneously, electrical behavior is measured via detection of tunneling currents between metallic tip and substrate. The behavior is fit using a mechanical model that attempts to predict the observed junction resistance as a function of applied force with consideration for mechanical restructuring of the monolayer at higher loads.

CP-AFM is also used to study self-assembled thin film of the diblock copolymer polystyrene-block-polyferrocenylsilane (PS-b-PFS) on gold substrate. Simultaneous height and electrical
current imaging verify the phase separation of the two blocks of the polymer and additionally, distinguish each block due to differential conductivity.

The phase separation of multi-component phospholipid bilayers (phosphatidylcholine/ sphingomyelin/ cholesterol) on supporting substrate into liquid-ordered and liquid-disordered phases is demonstrated using both topographical imaging, and the use of force map analysis through tip indentation and rupture measurements. The segregation and differential mechanical stiffness of the phases help to understand the important role of mechanical stability and rigidity membranes. An automated batch analysis process was implemented to facilitate the procedure.

The mechanical properties of microfluidically produced microgels (cross-linked sodium alginate and poly(N-isopropylacrylamide)) are measured using indentation experiments, to evaluate the suitability of these gels as cell-mimics. Nanoscale heterogeneities were avoided by using a tipless cantilever. This body of work shows that the alginate content of these microgels can be varied to tune their mechanical properties and that a platform for mechanical measurement of cell and cell-mimics is possible.
Acknowledgements

Foremost, I would like to thank my supervisor committee for guidance in the course of my graduate studies and doctoral research: my Ph.D. advisor Professor Gilbert Walker, and Professor Al-Amin Dhirani and Professor Zhenghong Lu. Moreover from Gilbert, I have learned many lessons equally applicable to other areas outside of scientific research: how to be a good leader, how to bring out potential in others, how to be a good mentor, how to learn.

My heartfelt gratitude goes to members of the Walker group, both past and present, for their suggestions and advice during group meetings and in discussion, their help in the lab, and for their support and camaraderie. To Dr. Weiqing Shi, Melissa Paulite, Adrienne Tanur, Isaac Li, Shell Ip, Dr. Nikhil Gunari, Christina MacLaughlin, Toan and Thang Nguyen, Claudia Grozea, Dr. Zahra Fakhraai, Ilia Auerbach-Ziogas, Mandy Koroniak, and Dr. Joyce Guest: graduate student life would not be the same without you! I would especially like to Dr. Shan Zou for inspiring greatness, easing the bumpy transition to research life, and assuaging the concerns of the new graduate student. I am also indebted to Ruby Sullan, who was a constant support throughout graduate school and with whom I've tried to crack the riddle of the Ph.D. 'career'. I am not sure if we were successful, because it was over before we got to the answer.

I would like to thank the esteemed scientists I have had the privilege and pleasure to work with in a number of collaborations: Professor Alexei Tivanski, Dr. David Rider, Dr. Jean-Charles Eloi, Dr. Oksana Protsiv, Neta Raz, Professor Ian Manners and Professor Eugenia Kumacheva. I would also like to thank several colleagues from within and outside the Chemistry Department, for their hard work and efforts in informal collaborations and departmental duties, from which I learned something from and was glad that a working relationship and friendship was established: Ethan Tumarkin, Anna Lee, Dr. Mingfeng Wang, Tingbin Lim, Dr. Joyce Dinglasan-Panlilio, Zhibin Wang, and Michelle Xu.

I would like acknowledge my friends for lending me an ear when needed, and offering their opinions and advice during the past five years. Although there are too many names to mention, I would like to especially thank Sara Khor, Dr. Bernard Fleet, Ivan Lin, and Bhavin Bijlani.
Lastly I would like to thank my family: for knowing how important this was to me, and for caring.

James Li

September 13th, 2010
# Table of Contents

Acknowledgements ........................................................................................................................ iv

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

List of Symbols and Abbreviations ............................................................................................ xi

List of Tables ............................................................................................................................... xiii

List of Schematics ......................................................................................................................... xiv

List of Figures .............................................................................................................................. xvi

1 Introduction ...............................................................................................................................1

1.1 Understanding and Creating at the Nanoscale .....................................................................1

1.2 The Self-Assembly Approach ..............................................................................................2

1.3 Surface Probe Microscopy ...................................................................................................3

1.4 Summary of Thesis ..............................................................................................................4

1.4.1 Use of the Atomic Force Microscope ......................................................................4

1.4.2 Scientific Scope .......................................................................................................4

1.4.3 Self-Assembled Systems ..........................................................................................5

1.4.4 Outline of Experimental Sections ............................................................................5

1.5 References ............................................................................................................................6

2 Tools of the Trade: Concepts in Contact Mechanics and the Atomic Force Microscope ....8

2.1 Preliminary Mechanics Concepts .........................................................................................8

2.1.1 Hooke’s Law for 1-D Linearly Elastic Materials ....................................................8

2.1.2 Extension to Higher Dimensions: Shear and Poisson’s Ratio .................................9

2.2 Contact Mechanics .............................................................................................................12

2.2.1 The Hertz Model for Non-Adhesive Contact ........................................................12

2.2.2 Shortcomings of the Hertzian Model ..................................................................16

2.3 The Atomic Force Microscope ...........................................................................................18
2.3.1 Contact Mode Imaging .......................................................................................... 19
2.3.2 Tapping (Intermittent Contact) Mode Imaging ...................................................... 20
2.3.3 Force Curve Collection .......................................................................................... 21
2.3.4 Force Mapping ....................................................................................................... 21
2.3.5 Conducting Probe AFM ......................................................................................... 22
2.4 References................................................................................................................... 24

3 The Self-Assembled Systems.......................................................................................... 29
3.1 Alkanethiol Self-Assembled Monolayers ..................................................................... 29
3.2 Diblock Copolymer Thin Films .................................................................................. 31
  3.2.1 Diblock Copolymers .............................................................................................. 31
  3.2.2 Microphase Separation of Diblock Copolymers in Bulk ....................................... 32
  3.2.3 Phase-Separation and Self-Assembly as Thin Films ............................................. 34
  3.2.4 Control of Polymer Thin Film Morphology .......................................................... 35
  3.2.5 AFM Study of Diblock Copolymer Thin Films ..................................................... 36
3.3 Substrate-Supported Lipid Membranes ..................................................................... 36
3.4 Microfluidically Self-Assembled Polymer Hydrogels as Cell- Mimics ....................... 37
3.5 References................................................................................................................... 38

4 Pressure Induced Restructuring of Monolayer Film Nanojunction Produces Threshold and Power Law Conduction ................................................................. 44
4.1 Permissions ............................................................................................................... 44
4.2 Abstract .................................................................................................................... 44
4.3 Introduction ............................................................................................................... 44
4.4 Experimental Details ................................................................................................ 46
  4.4.1 SAM Preparation .................................................................................................. 46
  4.4.2 Conducting Probe AFM Measurements ................................................................ 47
4.5 Results and Discussion ............................................................................................. 48
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.2</td>
<td>Introduction</td>
<td>82</td>
</tr>
<tr>
<td>6.3</td>
<td>AFM-tip Indentation with Membrane Breakthrough</td>
<td>84</td>
</tr>
<tr>
<td>6.4</td>
<td>Observables and Calculations</td>
<td>85</td>
</tr>
<tr>
<td>6.5</td>
<td>Experimental</td>
<td>87</td>
</tr>
<tr>
<td>6.5.1</td>
<td>Lipid Bilayer Preparation</td>
<td>87</td>
</tr>
<tr>
<td>6.5.2</td>
<td>AFM Imaging and Force Mapping</td>
<td>87</td>
</tr>
<tr>
<td>6.5.3</td>
<td>Batch Analysis of the Force Curves</td>
<td>88</td>
</tr>
<tr>
<td>6.5.4</td>
<td>Data Analysis and Visualization</td>
<td>90</td>
</tr>
<tr>
<td>6.6</td>
<td>Results and Discussion</td>
<td>91</td>
</tr>
<tr>
<td>6.6.1</td>
<td>Self-assembly and Mechanical Properties of Phosphatidylcholine-Sphingomyelin-Cholesterol Supported Lipid Bilayer Systems</td>
<td>92</td>
</tr>
<tr>
<td>6.6.2</td>
<td>Force Maps and Batch Analysis Performance</td>
<td>96</td>
</tr>
<tr>
<td>6.7</td>
<td>Conclusions</td>
<td>98</td>
</tr>
<tr>
<td>6.8</td>
<td>Contribution</td>
<td>99</td>
</tr>
<tr>
<td>6.9</td>
<td>References</td>
<td>99</td>
</tr>
<tr>
<td>7</td>
<td>Microgels with Interpenetrating Polymer Network as Cell-Mimic</td>
<td>102</td>
</tr>
<tr>
<td>7.1</td>
<td>Permissions</td>
<td>102</td>
</tr>
<tr>
<td>7.2</td>
<td>Abstract</td>
<td>102</td>
</tr>
<tr>
<td>7.3</td>
<td>Introduction</td>
<td>102</td>
</tr>
<tr>
<td>7.4</td>
<td>Materials and Methods</td>
<td>103</td>
</tr>
<tr>
<td>7.4.1</td>
<td>Materials and Microgel Synthesis</td>
<td>103</td>
</tr>
<tr>
<td>7.4.2</td>
<td>Mechanical Characterization of Microgels</td>
<td>103</td>
</tr>
<tr>
<td>7.4.3</td>
<td>Batch Analysis</td>
<td>106</td>
</tr>
<tr>
<td>7.4.4</td>
<td>Characterization of Microgels: Other Methods</td>
<td>106</td>
</tr>
<tr>
<td>7.5</td>
<td>Results and Discussion</td>
<td>106</td>
</tr>
<tr>
<td>7.5.1</td>
<td>Variation of Alginate Content</td>
<td>106</td>
</tr>
</tbody>
</table>
7.5.2 Effect on Swelling Ratio ................................................................. 107
7.5.3 Elastic Modulus and Relaxation Measurements .......................... 108
7.5.4 Evaluation of AFM Indentation as Testing Platform ................. 110
7.6 Conclusion ...................................................................................... 111
7.7 Contribution .................................................................................. 111
7.8 References ..................................................................................... 111
8 Outlook ............................................................................................. 113
9 Appendices ........................................................................................ 118
   9.1 Appendix: Programmed Code Used in Thesis Work for Analysis  118
      9.1.1 Main Functions ........................................................................ 118
      9.1.2 Instrument Specific File Handling ........................................... 127
      9.1.3 Fitting Functions ..................................................................... 139
      9.1.4 Manual Intervention Functions .............................................. 142
      9.1.5 Graphing Functions ................................................................. 145
      9.1.6 File I/O Functions .................................................................. 153
      9.1.7 Clean-up Functions ................................................................ 155
      9.1.8 Helper Functions .................................................................... 158
      9.1.9 Macros ..................................................................................... 162
      9.1.10 Code Version History ............................................................ 165
   9.2 Appendix: Rupture of a Lipid Bilayer- Theory and Derivations .... 168
      9.2.1 Rupture Activation Energy Calculation .................................. 168
      9.2.2 Rupture Energy Derivation ...................................................... 169
List of Symbols and Abbreviations

Below is a list of symbols and abbreviations most commonly referred to throughout the thesis. Other usage of these terms will be explicitly defined within the context of the work in which they are mentioned.

**Symbols**

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>τ</td>
<td>relaxation time (Chapter 7)</td>
</tr>
<tr>
<td>τ</td>
<td>shear stress (Chapter 2)</td>
</tr>
<tr>
<td>a</td>
<td>contact area radius</td>
</tr>
<tr>
<td>E</td>
<td>Young's modulus</td>
</tr>
<tr>
<td>F</td>
<td>force</td>
</tr>
<tr>
<td>G</td>
<td>shear modulus</td>
</tr>
<tr>
<td>k</td>
<td>spring constant</td>
</tr>
<tr>
<td>k_B</td>
<td>Boltzmann constant</td>
</tr>
<tr>
<td>N</td>
<td>degree of polymerization</td>
</tr>
<tr>
<td>R</td>
<td>electrical resistance (Chapter 4)</td>
</tr>
<tr>
<td>R, r</td>
<td>radius of curvature</td>
</tr>
<tr>
<td>T</td>
<td>Kelvin temperature</td>
</tr>
<tr>
<td>γ</td>
<td>interfacial tension</td>
</tr>
<tr>
<td>γ</td>
<td>shear strain</td>
</tr>
<tr>
<td>ε</td>
<td>stress</td>
</tr>
<tr>
<td>ν</td>
<td>Poisson's ratio</td>
</tr>
<tr>
<td>σ</td>
<td>strain</td>
</tr>
<tr>
<td>φ</td>
<td>diblock copolymer volume fraction</td>
</tr>
<tr>
<td>χ</td>
<td>Flory-Huggins parameter</td>
</tr>
</tbody>
</table>

**Abbreviations**

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AFM</td>
<td>atomic force microscopy</td>
</tr>
<tr>
<td>BIS</td>
<td>N,N'-methylenebisacrylamide</td>
</tr>
<tr>
<td>C_{10}SH</td>
<td>1-decanethiol</td>
</tr>
<tr>
<td>C_{6}SH</td>
<td>1-hexanethiol</td>
</tr>
<tr>
<td>CP-AFM</td>
<td>conducting probe AFM</td>
</tr>
<tr>
<td>Fc</td>
<td>ferrocene</td>
</tr>
<tr>
<td>FcC_{11}SH</td>
<td>11-ferrocenyl-1-undecanethiol</td>
</tr>
<tr>
<td>IPN</td>
<td>interpenetrating polymer network</td>
</tr>
<tr>
<td>Acronym</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td>PFS</td>
<td>polyferrocenylsilane</td>
</tr>
<tr>
<td>PNIPAm</td>
<td>poly(N-isopropylacrylamide)</td>
</tr>
<tr>
<td>PS-b-PFS</td>
<td>polystyrene-<em>block</em>-polyferrocenylsilane</td>
</tr>
<tr>
<td>SAM</td>
<td>self-assembled monolayer</td>
</tr>
<tr>
<td>SEM</td>
<td>scanning electron microscopy</td>
</tr>
<tr>
<td>SPM</td>
<td>scanning probe microscopy</td>
</tr>
<tr>
<td>STM</td>
<td>scanning tunnelling microscopy</td>
</tr>
<tr>
<td>TEM</td>
<td>transmission electron microscopy</td>
</tr>
</tbody>
</table>
List of Tables

Table 2.1 Systems studied via CP-AFM in literature. ................................................................. 23

Table 2.2 Related conducting / electrical based scanning probe techniques. ......................... 23

Table 3.1 Parameters controlling thin film morphology, important aspects, and methods of study. .......................................................................................................................................................... 35

Table 4.1. The two power-law scaling regimes observed for each SAM molecule. ................. 52

Table 4.2. Fitting parameters used in the multiple conduction pathway model. The molecular length and intermolecular distances are held constant. While the number of hops is not fixed, it is constrained to a lowerbound (0 hops) and an upperbound value (maximum number shown in table) as dictated by physical constraints..................................................................................................................... 55

Table 4.3. Best-fit elastic modulus and initial angle values for each molecule using the multiple conduction pathways model, for the fits in Figure 4.3. ................................................................. 58

Table 4.4. Slopes of line fits to Figure 4.4 for each molecule. Low and high-load regimes are distinguished. Behaviors for each molecule / slope combination is categorized into three groups. The estimated slope for a model close-packed SAM is also provided. ........................................ 61

Table 6.1 Rupture activation energies of the coexisting phases (liquid ordered domains $L_o$, and fluid disordered phase $L_d$, in DOPC/SM (1:1) with 10-40% cholesterol bilayers. (Adapted from 17)................................................................................................................................. 95

Table 6.2. Data analysis run times for 3 force maps with very similar data collection parameters, except for data points per force curve............................................................... 96

Table 7.1 Compositional variation of IPN microgels (Adapted from 11) ................................ 107
List of Schematics

Schematic 2.1 Hooke's law in terms of force and extension, and stress and strain for a 1D linearly elastic material. ........................................................................................................................................................................ 9

Schematic 2.2 Linear superposition of stresses and strain as a scale-up to higher dimensions.... 10

Schematic 2.3 Shear stress due to shear strain.............................................................................. 10

Schematic 2.4 Axial stresses couple to transverse strain through Poisson's ratio. .................... 11

Schematic 2.5 Hertzian contact between two bodies with radius of curvature $R_1$ and $R_2$ respectively. ........................................................................................................................................................................ 12

Schematic 2.6 The plane of contact under Hertzian mechanics has characteristic radius $a$. ....... 13

Schematic 2.7 The supposition of a) non-conforming contact (contact surface is a very small area) over b) conforming contact (bodies can ‘nest’ against each other) ensures the assumption that $a<<R$ holds true........................................................................................................................................................................ 14

Schematic 2.8 Scheme for a paraboloidal object impinging on an elastic half plane.............. 15

Schematic 2.9 Schematic of AFM operation. ............................................................................... 18

Schematic 2.10 Components of a conducting probe AFM setup................................................. 22

Schematic 3.1 Block copolymer categorization using (A-B)$_n$ notation..................................... 32

Schematic 4.1 Scheme of tip-monolayer-substrate showing relevant distances in multiple conduction pathway model: chain-to-chain distance $d_{cc}$, natural alkanethiol length $d_a$, effective thickness $d_t$ as a function of tilt $\theta$. Tunneling through chain (blue) has decay coefficient $\beta_c$ and through space (red) has coefficient $\beta_s$. ........................................................................................................................................................................ 54

Schematic 6.1. Flowchart of force map collection, subsequent analysis and data output. ........ 83
Schematic 6.2. Schematic of AFM tip-supported lipid bilayer interaction- (a) non-contact; (b) tip-bilayer contact; (c) bilayer indentation; (d) bilayer rupture (tip breakthrough); (e) tip-substrate contact.

Schematic 6.3. Program flow depicting (a) data import and intermediate calculations, smoothing, and initial threshold calculation; (b) data parsing with filter, critical point searching, and extraction and calculation of physical parameters; and (c) data output: force maps, the set of force curves, histogram, and scatter plots.

Schematic 7.1 Schematic of tipless cantilever in contact with microgel on Au substrate.

Schematic 8.1 Conduction through molecules under tension.

Schematic 8.2 Charge storage array using diblock copolymer with intermediate tunneling layer.
List of Figures

Figure 3.1 Static water contact angle as a function of mol % of mercaptohexadecanoic acid (MHDA, COOH- terminated) in a mixed monolayer with hexadecanethiol (HDT, CH$_3$-terminated) demonstrating the tunability of hydrophobicity of the surface. Dashed line is a linear trend to guide the eye................................................................. 30

Figure 3.2 AFM tapping mode height image of polystyrene-block-polymethylmethacrylate (PS-b-PMMA) on silicon substrate. Polymer was cast from acetone solution, and solvent annealed in a toluene-rich chamber. Pattern retains fidelity for areas up to (A) 1 µm × 1 µm, and (B) 2.5 µm × 2.5 µm large.................................................................................................................. 34

Figure 4.1. Force (solid line) and absolute current (dashed line) profiles measured simultaneously as a function of vertical piezo displacement for the Pt-coated tip over a FcC$_{11}$SH SAM. Only approach data is shown................................................................. 49

Figure 4.2. Log-log plots of junction resistance versus interaction force for (a) FcC$_{11}$SH, (b) C$_6$SH and (c) C$_{10}$SH SAMs show threshold and two power law-scaling regimes. Symbols are averaged data and solid lines are the corresponding linear fit..................................................... 51

Figure 4.3. Plots of junction resistance versus interaction force for FcC$_{11}$SH (×), C$_6$SH (◊) and C$_{10}$SH (□) SAMs. Solid black and grey lines are fits to Equation 4.8 for 0 and 1 through-space hops, respectively................................................................. 57

Figure 4.4. Plot of ln(RF$^{2/3}$) as a function of F$^{2/3}$ for (a) FcC$_{11}$SH (points), C$_6$SH (triangles) and (b) C$_{10}$SH (crosses) SAMs with the corresponding linear fit (solid lines)................................................. 60

Figure 5.1 Schematic illustration of (a) CP-AFM setup, and (b) cross-section showing expected morphology of polystyrene-b-poly(ferrocenylethylmethysilane) solid-state self-assembly on Au substrate. .............................................................................................. 70

Figure 5.2. AFM images of PS-b-PFEMS thin film, 1 µm × 1 µm. (a) Height image with (b) corresponding current image of sample scanned at –5 V. (c-e) Current images at large negative applied biases (–7 V, –9 V and –8 V respectively) show stronger response of PFEMS region... 72
Figure 5.3. (a) PS-b-PFEMS (○) shows a non-zero (negative) linear correlation of current and height with large negative voltage bias, while a film of PS-b-PMMA (■) has poor correlation. (b) Current-voltage graph of PFEMS (grey) and PS (black) swept from –9 V to +9 V. 

Figure 5.4. (a) Reproducibility of current signal in PS-b-PFEMS. Shown in (i)→(iii) is the same area of film imaged sequentially under biases that turn the current response on-off-on. (b) Current image of PS-b-PFEMS at (i) ambient conditions (20.9% O₂) and (ii) low-oxygen conditions (2-3% O₂). 

Figure 5.5. (a) Log-log plot showing the current averaged over twenty PFEMS domains in each current image. Solid line is a fit to a quadratic equation (See Section 5.3). (b) Observed widening of PFEMS cylinders in the current image (Figure 5.4) with increasing magnitude of applied voltage. Average width was obtained by binarizing each current image using a threshold of 0.5 pA, and calculating an average width for twenty PFEMS cylinders. 

Figure 6.1 Force curves indicating quantities extracted: breakthrough force, indentation region, and adhesion force. Points of interest- A: contact point, B: onset of rupture/ breakthrough, C: onset of substrate, D: maximum adhesion, E: sufficiently far from the surface with negligible adhesion effects. 

Figure 6.2. A set of data output (a) AFM height image (3 μm × 3 μm); corresponding recreated (b) force map of breakthrough forces and of; (c) Young’s modulus values; (d) representative graph depicting force curve (line), contact and breakthrough points (hollow circles) and curve fit to the Sneddon model (thick dashed line); (e) histogram of corresponding breakthrough forces from (b); (e) scatter plot showing correlation of breakthrough and adhesion forces. 

Figure 6.3 AFM images and height profiles of a DEC-Ceramide bilayer on mica in contact mode using a low setpoint of $F \sim 0.5$ nN (A, E) with corresponding lateral deflection image (B), and using a high setpoint of $F \sim 5.0$ nN (C, F), and in the intermittent contact mode (D, G). Regions outlined in green in the height profiles indicate the subdomains. (Adapted from¹⁸) 

Figure 6.4 Over 200 force curves over both $L_o$ (red) and $L_d$ (black) phases which show the breakthrough event, and the distance (height) difference between the two phases.
Figure 6.5 Output of force map analysis of DEC bilayers: A) recreated map of breakthrough force magnitudes, B) corresponding histogram showing bimodal distribution for breakthrough forces in $L_d$ (light shade) and $L_o$ (dark shade) phases, C) recreated elastic modulus map using Hertz contact mechanics and Sneddon model. ................................................................. 94

Figure 7.1 Typical force-time curve acquired for microgel particle showing the approach (blue), dwell (black) and retract (red) portions. ................................................................. 105

Figure 7.2 Effect of alginate concentration on degree of swelling in hydrogels. (from Set B, D, E) (Adapted from 11).................................................................................................................... 108

Figure 7.3 A) Elastic modulus and B) relaxation time as a function of alginate concentration (wt. %). (Adapted from 11).................................................................................................................... 109
1 Introduction

1.1 Understanding and Creating at the Nanoscale

It can be argued that scientists have been studying systems at the nanometer length-scale well before the term nanotechnology was invented and popularized, before Eric Drexler's popular non-fiction book *Engines of Creation* was published in 1986\(^1\), and perhaps prior to Richard Feynman's famous lecture *There's Plenty of Room at the Bottom* was delivered in 1959\(^2\). For example, many chemical reactions and biological events occur at the molecular level which, depending on the system, do reach or go below this length scale; and thus, scientists' success in observing, controlling and quantifying (macroscopic) changes in these systems confirms their mastery of the nanoscale. However, the progress of the nanotechnology field is not only driven by the need to gain an understanding of phenomena occurring at this length regime, but also to be able to create technology at this scale.

The foremost example of the last statement is in electronics technology. The trend towards miniaturization has been occurring since the 1970s, and generally following Moore's Law which dictated the exponential doubling growth in the number of transistors able to fit in a given area\(^3\). For example, the total transistor count in Intel\(^\text{®}\) microchips has grown from containing less than 50 transistors from when Gordon E. Moore made his observation in 1965, to over 500 million transistors by the time of 40\(^{th}\) anniversary of this law\(^4\). The motivation for such miniaturization is increased computing performance in smaller form factors. The direct result of this advancement, is the ubiquity of powerful personal computers that far exceed supercomputing capability from decades past, *smart* devices which trade computing capability for size to operate unobtrusively in common appliances, and mobile technology for computing (laptops), communication (cellular phones) and entertainment (mobile gaming consoles and tablets).

The progress of this miniaturization in the aforementioned industry has been hindered by the limitations of top-down fabrication technologies, namely photolithographic processing because of the inherent wavelength-dependence required in creating the desired features. Briefly, lithography depends on the passing of light through a mask to polymerize a sacrificial photoresist layer on top of a silicon substrate, which ultimately determines the geometry of the final pattern. However, light can diffract at the edges of the mask and lower the resolution\(^5\). Since the actual
extent of the diffraction depends on the light's wavelength, then the smallest feature sizes achievable via photolithographic techniques also succumb to this wavelength dependence. It can be concluded that this mode of fabrication has imposed a length scale limit to the feature size with the rule of thumb being approximately half the vacuum wavelength of the light utilized. The limitation of top-down technologies is not merely confined to computer chip fabrication, but in other areas as well. For example, the synthesis of nanoparticles by the mechanical pulverization of larger chunks also has a performance ceiling. It is clear that in order to be able to further create at the nanoscale, a different paradigm is needed.

An additional concern is that going below ~100 nm length scales introduces effects that are not merely extensions of macroscopic observations. For example, the smaller number of interacting species in a particular system may cause individual behavior to become significant and macroscopic observables may not be sufficient in describing actual phenomena, since, the latter quantities are usually the average behavior of an ensemble of species. In short, statistical mechanical and quantum size effects may dominate (for example, quantum confinement of electrons affects its transport in nanoscale and smaller systems). Also at these length scales, the balance of forces is much different from macroscopic observation: Coulombic forces and Van der Waals forces are large because of the small masses of the species and the short distances between them. Surface area to volume ratio is also much higher, and so, surface interactions are yet another issue. In order to study objects at nanoscale, tools are needed that can probe and distinguish these factors.

Therefore, in nanotechnology, two questions are important: how can we create? and how can we study systems at this length scale?

1.2 The Self-Assembly Approach

Self-assembly techniques are the proposed workaround to the limitations of top-down approaches. They are usually classified as bottom-up because rather than taking away components to achieve nanometer lengths (as in removal of sections photore sist from a thin film, or breaking apart bigger chunks of particles to create nanoparticles), instead, individual molecules are built up by taking advantage of intrinsic intermolecular forces and surface
interactions, until the resulting product is of an appropriate size. The challenge for this self-assembly procedure is in being able to halt this spontaneous assembly at the desired size, and also ensure that the desired form or morphology (ultimately, the functionality) is maintained.

There are numerous examples of self-assembly in literature: self-assembled monolayers (SAMs) on solid substrate, block copolymers morphologies formed in solution and when confined to surfaces, the field of supramolecular chemistry, and the inorganic synthesis of nanorods, nanotubes, and buckyballs. Even in nature self-assembly is observed when phospholipid bilayers self-assemble to encapsulate cells. Use of self-assembly techniques can be used to mimic nature's building blocks and create an investigative platform to study model lipid bilayers and model 'cells' through microfluidically formed microgels.

1.3 Surface Probe Microscopy

One important tool in the investigation of nanoscale systems is scanned probe microscopy, which includes instruments such as the scanning tunneling microscope (STM) and the atomic force microscope (AFM). Underlying this technology is a small probe that is allowed to interact with the surface. The probe may be used to scan laterally (as in the general umbrella term: scanning probe microscopy SPM), or to scan vertically (as in AFM force curve collection), using a feedback that is known to be distance-sensitive and effective in the nanometer range. For STM this feedback is tunneling current, and for AFM it is force. Factors that affect the resolution of such instruments include the sharpness of the probe, the experimental setup and environmental conditions.

These type of probed instruments can be used to study molecules and surfaces that are substrate supported, and therefore are excellent tools to study thin films and monolayers. Substrate supported micelles, vesicles and even individual molecules may also be studied. The AFM and STM provide complementary information, although the AFM is generally more widely used since it can operate under different conditions (in vacuum, in air, in water, and in physiological buffer, for example).

In addition to using the AFM as a force probe, it can be used to apply a force to the system and to observe the response: morphological or structural changes, kinetics, etc. Although a crude
mimic to a stimulus that the system may potentially come in contact with *in-situ*, the probe can be accurately controlled and the force response can be accurately measured.

### 1.4 Summary of Thesis

The aim of this thesis is to investigate self-assembled systems using force-based microscopy. It is divided into several sections, the first few of which provide relevant background to the science, systems and techniques utilized during the course of research. Following these are the main experimental sections, which are logically arranged by self-assembled system and coincides with the author's publications. Therefore, the experimental section is in part comprised of the compilation of manuscripts, with modifications depending on extent of intellectual contribution.

#### 1.4.1 Use of the Atomic Force Microscope

The AFM is the primary instrument used in this body of work. The AFM served three purposes:
1) obtain topographical information using raster scanning; 2) obtain mechanical information using force curve spectroscopy; 3) obtain complementary electrical information using a conducting probe and substrate setup (CP-AFM). With its ability to detail in the nanometer range, and with its primary feedback mechanism (i.e. force) being very distance-sensitive, the AFM is very appropriate in elucidating mechanical stiffness, surface restructuring and other changes due to compression (compressing force stimulus). The atomic force microscope will be briefly reviewed in Chapter 2.3.

#### 1.4.2 Scientific Scope

The majority of the work carried out involved the study of soft (relative to AFM tip or substrate) systems placed under a compressive force stimulus. The response to such an applied load would be indentation, with possible structural changes at a threshold magnitude of force/ pressure. Models of mechanical indentation, primarily the Hertz model will be reviewed in Chapter 2.2.1.
Relevant models for electrical conduction are introduced as needed in the experimental chapters themselves in Chapter 4 and 5.

### 1.4.3 Self-Assembled Systems

Several self-assembled systems are studied: self-assembled monolayers, diblock copolymer thin films, solid supported lipid bilayers membranes, and microfluidically created polymer microgels. Each one will be briefly reviewed in Chapter 3.

### 1.4.4 Outline of Experimental Sections

In Chapter 4, the packing and restructuring of several self-assembled monolayers (SAMs) as exhibited by alkanethiol systems on gold (hexanethiol, decanethiol, and ferrocenyl-undecanethiol) are demonstrated when pressure is induced by an AFM tip. The tip and substrate are both conducting, so tunneling currents are also concurrently measured while using the AFM tip as a force probe and stimulus. The behavior is fit using two mechanical models which try to predict the observed junction resistance as a function of applied force; however only by interpretation of the model that assumes mechanical restructuring can the behavior be satisfactorily explained. A fundamental aspect of nanoscale electronics is the formation of a good electrical junction/contact, and therefore elucidation of conduction and conduction changes upon mechanical contact is vital.

In Chapter 5, the self-assembly of the diblock copolymer polystyrene-\textit{block}-polyferrocenylsilane (PS-\textit{b}-PFS) as a thin film on a conducting, planar substrate is studied using conducting probe AFM. Electrical current imaging is used to verify the phase separation of the two blocks of the polymer and additionally, distinguish each block due to their differing conductivities. The results are corroborated by the simultaneously collected topography.

In Chapter 6, the phase separation of multi-component phospholipid bilayers (phosphatidylcholine/ sphingomyelin/ cholesterol) on supporting substrate is demonstrated using both topographical imaging and the use of force map analysis to calculate mechanical quantities and properties through tip indentation measurements with position identification. The
segregation and differential mechanical stiffness of the phases, even when studied on solid support, help to understand the important role of membranes in cells. Since force maps involve length analysis of a large number of force curves, an automated process was implemented to facilitate the procedure.

In Chapter 7, the mechanical properties of microfluidically produced microgels (double interpenetrating polymer network of cross-linked sodium alginate and poly(N-isopropylacrylamide)) are measured using indentation experiments, to evaluate the suitability of these gels as cell-mimics. Nanoscale heterogeneities were avoided by using a tipless cantilever. This body of work shows that a platform for mechanical measurement of cell and cell-mimics is possible.

1.5 References


(2) The transcript of Richard Feynman's lecture There's Plenty of Room at the Bottom in the website: http://www.zyvex.com/nanotech/feynman.html Last accessed 2010-07-10

(3) Moore, G. E. In Electronics Magazine 1965; Vol. 38.


2 Tools of the Trade: Concepts in Contact Mechanics and the Atomic Force Microscope

2.1 Preliminary Mechanics Concepts

2.1.1 Hooke’s Law for 1-D Linearly Elastic Materials

The understanding of mechanics at the nanoscale must begin with introduction of some preliminary concepts. The most basic concept was introduced in the 17th century by Robert E. Hooke with the Latin phrase, *ut tensio sic vis*, translated as *as the extension, so the force*. In mathematical terms, this is Hooke’s Law for linearly elastic materials:

$$ F = k \cdot \Delta x $$

Equation 2.1

where $F$ is the load or force applied, and $k$ and $\Delta x$ are the spring constant and deformation respectively of a material along its length $x$ (Schematic 2.1). This relationship can be expressed in an alternative manner with the stress-strain equation:

$$ \sigma = E \cdot \varepsilon $$

Equation 2.2a

$$ \varepsilon = \frac{\sigma}{E} $$

Equation 2.2b

where $\sigma$ is elastic stress, $\varepsilon$ is strain, and $E$ is the Young’s modulus, a material dependent property.

Stress is defined as

$$ \sigma = \frac{F}{A} $$

Equation 2.3

which is the force $F$ acting over an area $A$. It has the units of pressure. Strain is defined as the fraction of the deformed length $\Delta x$ to its original length $x$:

$$ \varepsilon = \frac{\Delta x}{x} $$

Equation 2.4

Strain is unitless. The equivalence of Equation 2.1 and Equation 2.2 can be shown by substitution of Equation 2.3 and Equation 2.4 into Equation 2.2a:
\( \sigma = \frac{F}{A} \)
\( \epsilon = \frac{\Delta x}{x} \)
\( \left( \frac{F}{A} \right) = E \cdot \left( \frac{\Delta x}{x} \right) \)
\( F = \left( \frac{EA}{x} \right) \cdot \Delta x \)
\( F = k \cdot \Delta x \)

The material dependence of a property lies in \( E \), and in addition, the stress-strain relationship is also affected by the geometry, which in this case, is encapsulated in \( A \) and \( x \). The role of \( E \) as a stiffness parameter is readily seen in Equation 2.2b, where it serves to reduce the effect of stress on strain.

Schematic 2.1 Hooke’s law in terms of force and extension, and stress and strain for a 1D linearly elastic material.

2.1.2 Extension to Higher Dimensions: Shear and Poisson’s Ratio

The first step to extend the stress-strain relationship to higher dimensions is to begin using vectors to represent stresses and strains in the \( x \)- and \( y \)-directions (for example), and the net stress and strain due to linear superposition (Schematic 2.2). However, one must also consider that the direction stress is acting (for greater than one dimension) is not necessarily in the same direction that the strain is being produced. Physically, this is observed in the phenomenon of shear, and in the Poisson’s ratio.
Shear stress and strain are the rotational analogue to linear stress and strain. When a stress $\tau$ is applied to a 2-dimensional object as in Schematic 2.3, it creates a shear strain $\gamma$, which is the radian angle in the direction of material deformation. This produces the analogue to Equation 2.2:

$$\tau = G \cdot \gamma$$  \hspace{1cm} (Equation 2.5a)

$$\gamma = \frac{\tau}{G}$$ \hspace{1cm} (Equation 2.5b)

with $G$, the shear modulus, replacing $E$. 

Another material dependent parameter, the Poisson’s ratio $\nu$, quantifies the coupling of stresses and strains in the orthogonal direction (shown in Schematic 2.4). For example, the stress in the
axial direction, $\sigma_{\text{axial}}$ produces the corresponding strain $\varepsilon_{\text{axial}}$, but additionally induces strain in the transverse direction, $\varepsilon_{\text{transverse}}$, by the relation:

$$\varepsilon_{\text{transverse}} = -\nu \cdot \varepsilon_{\text{axial}}$$

Equation 2.6a

$$\nu = -\frac{\varepsilon_{\text{transverse}}}{\varepsilon_{\text{axial}}}$$

Equation 2.6b

Therefore, in higher dimensions, the expressions for stress-strain relationship (for 2-dimensions, for an isotropic material) are:

$$\sigma_x = \frac{E}{(1-\nu^2)} \left( \varepsilon_x + \nu \varepsilon_y \right)$$

Equation 2.7

$$\sigma_y = \frac{E}{(1-\nu^2)} \left( \nu \varepsilon_x + \varepsilon_y \right)$$

$$\tau = G \gamma = \frac{E \gamma}{2(1+\nu)}$$

The extension to 3-dimensions becomes more complicated, with $\sigma_x$, $\sigma_y$, and $\sigma_z$ representing the normal stresses in the orthogonal directions, and $\tau_{xy}$, $\tau_{yz}$, and $\tau_{zx}$ representing the shear stresses in the planes identified in the suffix (there are actually 6 shear stresses, but only 3 are necessary due to symmetry considerations). Therefore, the description requires a 6-element tensor to describe the stresses felt by a 3D object.

It is observed that the response (strain) to stresses (force) is determined by geometry and by material mechanical properties, of which $E$, $G$, and $\nu$ have been mentioned. However, the
2.2 Contact Mechanics

2.2.1 The Hertz Model for Non-Adhesive Contact

Heinrich Hertz first introduced the conceptual, mathematical framework of contact mechanics in 1881\(^1\) (English version 1896\(^2\)). He used the theory of elasticity to study two elastic bodies in contact by accounting for the displacement and stresses in the \(x\), \(y\), and \(z\) directions, and furthermore, accounting for what happens both within and outside of the surface of contact (Schematic 2.5).

![Schematic 2.5 Hertzian contact between two bodies with radius of curvature \(R_1\) and \(R_2\) respectively.](image)

If two elastic bodies, analytically described by 3-dimensional quadratic equations are in contact (and having characteristic radii of \(R_1\) and \(R_2\) respectively), the contact surface occurs along a plane normal to the direction of the applied load, and has characteristic radius \(a\), as shown in Schematic 2.6. For example, if the direction of applied load is the \(z\)-axis in rectilinear Cartesian coordinates, then the \(xy\)-plane forms this contact surface. Given two corresponding points \((A_1\)
and $A_2$), one on each body, the distance between them prior to contact can be found by noting the positional displacements (e.g. subtraction of the analytical expressions). However, after making contact and generating the contact surface, the boundary conditions as set by Hertz apply: if the two points are within the contact surface, the distance between them equals zero, otherwise the distance is a positive value. After making contact, elastic deformations of the two contacting bodies contribute to the distance calculation, in addition to positional displacements.

**Schematic 2.6** The plane of contact under Hertzian mechanics has characteristic radius $a$.

Hertz also accounted for forces. Within the contact surface, the total tangential and normal forces were the surface integral over all infinitesimally small elements. Outside the contact surface, all forces are zero.

Several assumptions are made which effectively translate to the assumption that the contact surface is much much smaller than the characteristic dimension of the elastic bodies ($a \ll R$), hold true. These include: 1) strains are small and are within the elastic limit; 2) each body is an elastic half-space; and 3) objects are non-conforming (Schematic 2.7). Furthermore, it was required that surfaces were frictionless, continuous, and in non-adhesive contact.
Sneddon, in 1965 extended this work by solving for the relation between total applied load and penetration depth for a class of problem where an axisymmetric Boussinesq punch of arbitrary profile is loaded on an elastic half space\(^3\). Since it was written in 1965, its practical applications were for stainless steel punches of various shapes exerting vertical load on a planar surface. In particular, he solved the load versus penetration depth relation for a cone, and a paraboloid of revolution exerting a normal force on a surface. The model introduced has since been adapted to look at nanoscale systems, since much of the physics still applies, and in the case of Sneddon's work, conical or paraboloidal punch geometry can sufficiently model the tip interaction (see Section 2.3).

The general picture of a tip interacting with a sample has been simplified to a paraboloidal object (e.g. the tip) of radius \(r\) interacting with an elastic, infinitely thick, infinitely planar film (e.g. the sample) (Schematic 2.8). Since the curvature can be much higher than the sample (especially when it is film-like), then this “sphere”-plane assumption is valid.
When a paraboloidally-shaped tip is pressed into the film with a loading force \( F \), the contact area's radius \( a \), is given by:

\[
a^3 = \frac{F \cdot r}{K}
\]

where \( K \) is an effective elastic modulus.

\[
K = \frac{4}{3} \left( \frac{E}{1 - v^2} \right)
\]

Equation 2.9

composed of \( E \) and \( v \), the Young’s modulus and Poisson’s ratio respectively. If the two materials in contact are dissimilar, as is often the case in realized AFM experiments (see Section 2.3), then \( K \) can be replaced with:

\[
K = \frac{4}{3} \left[ \frac{\left(1-v_1^2\right)}{E_1} + \frac{\left(1-v_2^2\right)}{E_2} \right]^{-1}
\]

Equation 2.10

where the subscripts 1 and 2 refer to the two bodies in contact.

The displacement \( \delta \) is given by
\[
\delta = \frac{a^2}{r} = \frac{F^{2/3}}{K^{2/3}r^{1/3}}
\]

Equation 2.11

The force-displacement relationship can be more directly expressed as:

\[
F = K\sqrt[r]{\delta^{3/2}}
\]

Equation 2.12

2.2.2 Shortcomings of the Hertzian Model

2.2.2.1 Adhesive Contact: JKR and DMT Models

The seminal work by Hertz has since been extended to factor in the effect of adhesion. The work by Johnson, Kendall, and Roberts in 1971 built upon Hertz's work by first demonstrating the actual area of contact can be larger than predicted by Hertzian mechanics in several sphere-elastic half plane experiments using rubber or gelatin materials\(^4\). They also offer up a revised model that reflects the increased area by accounting for surface energy and elastic energy.

\[
a^3 = \frac{r}{K} \left( F + 3\gamma r + \sqrt{6\gamma r F + (3\gamma r)^2} \right)
\]

Equation 2.13

The extra terms contributing to the increased area are shown to depend on \(\gamma\), the interfacial tension between the two objects in contact. At zero loading force, this equation still gives a finite contact area, and this persists even when the forces become negative. This means that adhesion forces are contributing, since extra pull-off forces are necessary to reduce the contact area to zero. The applicability of the equation to physical events is exceeded when the terms under the square root become negative. The pull-off force that is required for this equation to hold true is:

\[
F \geq -\frac{3}{2}\gamma r
\]

Equation 2.14

Therefore this value at equality is the force at which the elastic bodies just separate. It is noteworthy that this equation is independent of the mechanical parameter \(E\).

Another model by Derjaguin et al, (DMT Model) is a more general solution, which also accounts for Van der Waals interaction forces outside of the contact surface\(^5\).
### 2.2.2.2 Effects of Finite Sample Thickness

It should be noted that the Hertz model assumes the contact of elastic half-spaces. In other words, the bodies in contact are assumed to be infinitely thick. However, when studying thin films, this may not necessarily be true. The ramifications and prescribed solution to this problem was worked out by Akhremitchev et al.\(^6\).

If the logarithm of both sides of the force-displacement relation in Equation 2.12 is taken, then the expression becomes:

\[
\log(F) = \log(K\sqrt{r}) + \frac{3}{2}\log(\delta)
\]

which depicts a linear relationship in \(\log(F)\) and \(\log(\delta)\), with the \(y\)-intercept containing the term with mechanical parameters. Therefore, this provides a means to extract mechanical parameters from force-indentation (force curve collection, see Section 2.3.3) experiments. However, they showed that very large indentations into a polymer thin film exhibits a force-displacement profile that does not exactly show a completely linear trend in the log-log scale. And, in fact, the apparent elastic modulus obtained via this method can be different, for example from the same material probed under conditions more amenable to the Hertz theory (thicker film, or lower loads/penetration depths applied). Their prescribed solution to this problem is to determine correction factors that can be used to multiply with the apparent elastic modulus to obtain a more accurate value. This correction factor was obtained from numerical simulations that do factor in finite sample thickness, and also consider the difference mechanical properties of the film, and its supporting substrate. In particular, they utilize the ratio of Lamé coefficients (rigidity moduli) of the film and substrate. This method requires the practitioner to either perform their own numerical simulation, or utilize a set of compiled tabulated multipliers or curves.

Therefore, in deciding on which mechanics model to use, one must carefully determine the significance of adhesion in the experiment, given the experimental system in question, and its environment\(^7\). In addition, corrections to obtained mechanical parameters are necessary when the finite thickness of the film being studied results in interaction with the underlying substrate, invalidating the assumption of two elastic half spaces.
2.3 The Atomic Force Microscope

The atomic force microscope was invented in 1986 just 5 years after the invention of the scanning tunneling microscope. Like the latter instrument, it is a scanned probe technique, used to study surfaces much smaller than the 100μm length scale.

The main aspects of AFM operation (Schematic 2.9) include:

- Sharp probe (the AFM tip) with small surface area at the apex, suspended on a cantilever.
- Piezoelectric actuation to allow for very fine lateral and vertical translation of either the tip holder head or the sample stage.
- Utilization of a distance-sensitive parameter (force) as the main feedback.
- Utilization of laser and photodiode setup to monitor the deflection of the cantilever.

Schematic 2.9 Schematic of AFM operation.
There are numerous examples of AFM applications including the study of surface morphology of all kinds, the measurement of friction, adhesion and other forces, the mapping of temperature, capacitance and surface potential (see for example the conducting probe section Chapter 2.3.5), nanolithography, surface modification and manipulation, and the imaging of biomolecules and polymers.

2.3.1 Contact Mode Imaging

The AFM has two main modes of operation: imaging and force curve collection. In the imaging mode, the tip is raster-scanned across the surface to allow the tip to interact with the sample over a specified area, and produce a topographic image as a result of tip-sample interaction at each position. In contact mode imaging, the cantilever suspending the tip deflects due to attractive or repulsive forces when the tip and sample are in contact. When the tip base is moved towards or away from the sample, this changes the force felt by the tip, and thus also changes the extent of cantilever deflection. Therefore, this interaction force can be used as a feedback parameter, and the tip base moved up or down accordingly as it is raster-scanned across the surface to maintain a constant interaction force. The collection of height adjustments thus generates topography of the area of interest.

The data obtained from contact mode imaging is most intuitive to understand and interpret, as the forces of a bending cantilever can be obtained by taking its deflection distance and multiplying it by its stiffness using Hooke's Law (Section 2.1). Contact mode imaging works extremely well with hard samples rather than soft samples for a number of reasons. Soft samples have a possibility of sustaining damage due to inelastic deformations, when pressed too hard by the AFM tip. Conversely, there is a greater chance for tip contamination as some of the sample may adhere to the tip, thus muddling the interpretation of further data collected. With contact mode imaging, a wider range of cantilever stiffness values can be used. In particular, more flexible cantilevers can be used, which leads to greater sensitivity to tip-sample interactions.

Additionally, the lateral deflections of the cantilever provide complementary information, as the frictional force that arises from sample interaction is material dependent. Therefore, any lateral chemical or compositional differences in the sample being studied, whether or not they differ in
height, can be distinguished if there are differences in friction. This knowledge can be used to extract a friction map or to just qualitatively distinguish phases. One problem with the lateral deflections experienced by the cantilever is that it may convolute vertical deflections recorded especially when torsion is high.

Lastly, contact mode imaging is also preferred in experiments where force spectroscopy will also be performed. Force spectroscopy usually requires the same type of cantilever as contact mode imaging. Therefore, these two modes can be utilized consecutively without the need to switch tips, which would otherwise lead to difficulties in remaining within scanning range of a region of interest.

2.3.2 Tapping (Intermittent Contact) Mode Imaging

In tapping mode imaging\textsuperscript{7,39}, a different mode of operation is utilized to generate the topography. The tip is mounted on a stiffer cantilever, and made to oscillate by piezoelectric actuation. When sufficiently far from any sample, the oscillation of this free tip has a particular amplitude and phase relative to the driving oscillator. As the tip approaches the sample, the oscillation becomes damped, and its oscillation amplitude and phase angle both change. In this case, the amplitude is used as a feedback, and the tip base's distance from the surface is adjusted so that a specified amplitude (actually the ratio relative to the free amplitude) can be maintained. The height adjustments to maintain constant amplitude again generate the topography.

The phase angle changes are also recorded and provide complementary information to the height image. Since the phase angle arises from dampening of the oscillation as the tip intermittently contacts the surface, it is also material dependent, and can be used to distinguish phase the same way as lateral deflections are used. However, the actual value of the phase angle is difficult to ascribe to a particular physical or mechanical parameter, and may be convoluted with other experimental conditions. So, in many cases phase angle is only used qualitatively.

Since the tip is only in intermittent contact with the sample, it is not as damaging, and is therefore more suitable for softer samples. Furthermore, even during the contact, force is only in the vertical direction so lateral deflections are kept to a minimum.
Tapping mode imaging was not used extensively in this thesis, beyond preliminary topography verification, since most AFM work required the subsequent collection of force curves, for which contact-mode tips were more amenable.

2.3.3 Force Curve Collection

While the imaging modes are best used to give topography, it is somewhat restricted in quantification of the forces, since a zero-force reference is only established prior to the scan, and therefore may change or drift during imaging. A more accurate mode of AFM operation is force spectroscopy, or force curve collection \(^{28,40-42}\). It does provide a zero-force reference, which is established by human interpretation for each force curve collected. In a force curve collection, the tip deflection is collected over a period of time, during which the tip base is moved towards and then away from the sample in a predefined way. The evolution of the forces as a function of the base's vertical traversal distance provides one iteration of a force curve. This force curve is only collected on a laterally fixed spot, so it does not contain location/area information by itself. However, it does provide rich mechanical information and fully utilizes deflection sensitivity of the cantilever. The elasticity of single molecules\(^{43,44}\), and nanoindentation experiments\(^{17,37,45-55}\) are just two examples of the many systems studied with force spectroscopic techniques.

The collection and analysis of force curves were used extensively in this research, most prominently in Chapter 4, Chapter 6, and Chapter 7. Chapter 6 contains further information on force curves.

2.3.4 Force Mapping

The workaround to force curves not being able to provide area information is to collect force curves at many spots. Therefore, by systematically dividing a region of interest into a grid, force curves can be collected with \((x,y)\) positional information to generate a force map. This method now provides force curve information, and any information that imaging modes provide can theoretically be reported as well. The tradeoff however, is that since each individual force curve needs the AFM tip base to approach and retract from a spot over a period of time, then force
curve collection for the whole map is very time consuming. This will result in experimental drift (feedback is turned off) and is compounded by the fact that the subsequent analysis of force curves is laborious.

These were the problems faced when performing force map collection for supported lipid membrane systems with our solution to address them described in Chapter 6. Chapter 6 contains further information on force maps.

### 2.3.5 Conducting Probe AFM

In addition to application of force and the collection of height, amplitude, phase angle, other stimulus-response pairs can be used during either force curve collection or imaging modes. One that was used in this thesis was voltage application with current collection—known as conducting probe AFM (CP-AFM). In this mode, a metal-coated tip was used in conjunction with a conducting substrate, and the necessary electronics to apply a voltage bias, and record the current via a transimpedance amplifier (Schematic 2.10).

![Schematic 2.10 Components of a conducting probe AFM setup.](image-url)
A conducting probe-equipped AFM can be used in imaging mode to collect a current image, in force curve collection mode to collect a current versus distance curve while a fixed voltage bias is applied.

Additionally, a voltage waveform can be applied and the current response collected to create current-voltage \((I-V)\) curves of the material's response. This is usually performed with the tip in contact with the sample at a constant deflection of the cantilever (maintained with feedback).

CP-AFM has been used to study a wide variety of systems, as can be seen in Table 2.1.

<table>
<thead>
<tr>
<th>System</th>
<th>References</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alkanethiol monolayers and molecules</td>
<td>50,56-63</td>
</tr>
<tr>
<td>Biomolecules (e.g. carotene)</td>
<td>64-66</td>
</tr>
<tr>
<td>Organic semiconductors</td>
<td>11,67,68</td>
</tr>
<tr>
<td>Conjugated polymers</td>
<td>36,69-71</td>
</tr>
<tr>
<td>Organic Photovoltaics / Solar cells</td>
<td>12,72</td>
</tr>
<tr>
<td>Proteins</td>
<td>73-76</td>
</tr>
<tr>
<td>Inorganic nanostructures</td>
<td>15,77</td>
</tr>
</tbody>
</table>

CP-AFM belongs to a group of specialized AFM techniques developed to measure local electrical characteristics, for which dedicated conducting probes and specialized software and electronics have been created (Table 2.2).

Table 2.1 Systems studied via CP-AFM in literature.

Table 2.2 Related conducting / electrical based scanning probe techniques.
<table>
<thead>
<tr>
<th>Name</th>
<th>Method of Operation</th>
<th>References</th>
</tr>
</thead>
<tbody>
<tr>
<td>Electrostatic Force Microscopy (EFM)</td>
<td>Electrostatic force (from capacitance) causes cantilever deflection</td>
<td>78-81</td>
</tr>
<tr>
<td>Scanning Capacitance/Impedance Microscopy (SCM)</td>
<td>Response to AC voltage bias contains impedance information</td>
<td>55,82-88</td>
</tr>
<tr>
<td>Scanning Electrochemical Microscopy (SECM)</td>
<td>Use of the tip is a scanning nanoelectrode in analogue to an electrochemical setup</td>
<td>14,82,83,89-93</td>
</tr>
<tr>
<td>Kelvin Probe Microscopy (KPM)</td>
<td>Detection of work function difference</td>
<td>34,94-96</td>
</tr>
</tbody>
</table>

In addition to electrical measurement, these type of scanning probe instruments can also be used to induce local changes electrically, e.g. tip-induced oxidation\(^9^7\) and, electrostatic pressure-induced deformation\(^2^9\), which may have possible applications in nanoscale patterning and lithography.

CP-AFM is used in force curve collection mode in Chapter 4, and in imaging mode in Chapter 5.

2.4 References

(1) Hertz, H. *Journal für die reine und angewandte Mathematik* **1881**, *2*, 156.


(10) Xuan, Y.; Peng, J.; Cui, L.; Wang, H.; Li, B.; Han, Y. Macromolecules 2004, 37, 7301.
(38) The best source of AFM technical information come from the instrument manufacturers themselves: Veeco, Asylum Research, JPK, for example.


3 The Self-Assembled Systems

3.1 Alkanethiol Self-Assembled Monolayers

These systems are comprised of an alkane-based chain with a thiol end-group on one end, and either a simple methyl-termination on the other terminal, or another functional group (ω-substituted alkanethiols). Alkanethiols can chemisorb onto metals via formation of a metal-thiol bond, and thus form a layered or film structure on a surface. Gold is typically used as the metal substrate due to its inertness. And, with careful control of the self-assembly procedure and choice of chain length and functional group, a very well-ordered, close-packed single layer (monolayer) can be formed.

Self-assembled alkanethiol systems were first studied in 1989 in a series of seminal papers by the groups of Whitesides and Nuzzo\textsuperscript{1-3} and have since become a mature, established technique for creating model surfaces\textsuperscript{4,5}. Alkanethiol self-assembled monolayers are perfect model surfaces for the following reasons:

a) single monolayer formation means thickness of film will depend on chain-length\textsuperscript{3}, and thus can be controlled simply by choice of alkanethiol.

b) well-established self-assembly/ bottom-up technique makes creation of a modified surface or monolayer coating very easy\textsuperscript{5}.

c) tuning of surface properties is highly achievable, via the use of different functional group\textsuperscript{1}, or by varying the ratio of a mixture of several alkanethiols\textsuperscript{7}. In this way, wettability, for example, can be tuned simply by adjusting the ratio of hydrophilic- and hydrophobic- terminated alkanethiols\textsuperscript{4}. (see for example, Figure 3.1)

d) Device applications achievable via incorporation of appropriate molecules, and/or pairing functionality with metallic substrate. For example electrochemical and electrical devices can utilize the metallic substrate as an electrode. Surface plasmon-based sensors require an interface between materials with a positive dielectric constant and negative dielectric constant (e.g. the metal).
Many studies on gold- alkanethiol systems have been carried out, including the nature of the adsorption kinetics\(^8,9\), the packing and ordering of the monolayers\(^10\), stability, electron transfer processes\(^11-21\), and using a number of techniques including X-ray based techniques\(^22\), Fourier Transform-Infrared (FT-IR) spectroscopy\(^10,23\), contact angle measurements\(^4\), electrochemical measurements\(^24,25\), quartz crystal microbalance (QCM) measurements\(^26\), surface plasmon resonance (SPR) spectroscopy\(^27\), and the probed microscopy techniques: STM\(^28-31\) and AFM\(^7,32-35\). Furthermore, the development of alkanethiol self-assembly was crucial in a number of techniques, including soft lithography\(^36-39\) which itself is an attempt to provide an alternative route to patterning at the sub-micron scale and thereby address future needs and shortcomings of aforementioned top-down techniques such as photolithography.

The AFM can be used to study SAMs in a number of ways. Imaging using AFM probes the tip-end group interaction. For example, mixed alkanethiol SAMs may be distinguished via the interaction with the tip, and may show phase contrast via frictional force\(^34,40,41\) or phase angle

---

Figure 3.1 Static water contact angle as a function of mol % of mercaptohexadecanoic acid (MHDA, COOH-terminated) in a mixed monolayer with hexadecanethiol (HDT, CH\(_3\)-terminated) demonstrating the tunability of hydrophobicity of the surface. Dashed line is a linear trend to guide the eye.
images. As well, since the AFM is very sensitive in the Z-axis, accurate determination of the height of the SAM can be made, if the substrate's relative height can also be simultaneously measured (if substrate is not exposed via defect, a blade can be used to scratch away the SAM to reach it).

Alkanethiol SAMs are studied in Chapter 4 as a platform for studying nano-sized electrical contacts in electronics application. This study highlights the generation of tunneling currents which arise from pressure-induced evolution of mechanical structure in these thin films.

### 3.2 Diblock Copolymer Thin Films

#### 3.2.1 Diblock Copolymers

Polymers are macromolecules generated by the covalent attachment of repeating sub-units or monomers in a controlled manner that can guarantee their final overall size and physical and chemical properties (by monomer selection). They are an important class of materials because of their ability to span various length scales by this controlled polymerization. This fact, in combination with low cost, excellent processability, and amenability to mass production, has guaranteed the ubiquity of polymeric materials in technological and everyday applications. A sub-category of polymers is block copolymers- these are systems having blocks of smaller polymers covalently linked together. Each block is distinct in that it is formed from the repeat of a single monomer.

If we use the notation as put forward by Bates et al.\(^42\), then a block copolymer with blocks \(A\) and \(B\) can be represented as \((A-B)_n\) where \(n\) is the number of covalent bonds formed between the 2 blocks. When \(n=1\), the system is referred to as a diblock copolymer, as there is only one covalent linkage between the 2 blocks \((A-B)_1\). Triblock copolymers are represented by \(n=2\), \((A-B)_2\) and when \(n\geq 4\), the blocks form a star (Schematic 3.1). Within the thesis, discussion will be restricted to diblock copolymer systems.
3.2.2 Microphase Separation of Diblock Copolymers in Bulk

The bulk equilibrium behavior of diblock copolymers is controlled by three primary factors: the choice of monomers, the polymer’s overall size, and the relative size of each block. Respectively, these are quantified by the Flory-Huggins parameter $\chi$, the degree of polymerization $N$, and the relative volume fraction of the two blocks ($\phi_A$ or $\phi_B$).

The choice of monomer affects the self-assembly of block copolymers because it establishes how the blocks interact with each other. Intuitively, when many chains of a particular diblock copolymer interact in the solution- or solid-state, they tend to rearrange themselves in a manner such that $A$ (or $B$) blocks are close to neighboring $A$ (or $B$) blocks, and the two block types are as far apart as possible. The extent of interaction is formally quantified by the Flory-Huggins interaction parameter $\chi$, which is related to the contact energy between the monomer units within the diblock system. If block $A$ is composed of monomers $a$, and block $B$ with monomers, $b$, then the contact energy ($\varepsilon_{ij}$) between monomers are $\varepsilon_{aa}$, $\varepsilon_{ab}$, $\varepsilon_{bb}$ with the subscript referring to the pair of monomers $i$ and $j$ in contact. The Flory-Huggins parameter can be obtained approximately by:
\[
\chi = \frac{1}{k_B T} \left( \epsilon_{ab} - \frac{(\epsilon_{aa} + \epsilon_{bb})}{2} \right)
\]

Equation 3.1

where \(k_B\) is Boltzmann’s constant and \(T\) is Kelvin temperature. Based on the dependence on \(T\), one can conclude that \(\chi\) is largely enthalpic, although it has a more general form:

\[
\chi = \alpha T^{-1} + \beta
\]

Equation 3.2

with \(\alpha\) and \(\beta\) representing enthalpic and entropic contributions respectively. When \(\chi\) is large, then the energy of the system is high, and therefore the system favors a configuration that reduces the \(A-B\) monomer contacts, which is accomplished by losing some configurational or translational entropy via ordering (microphase separation). Additionally, a large degree of polymerization, \(N\), also leads to a more ordered state. Therefore, one often uses the product \(\chi N\) as a determinant of state. For block copolymers, if \(\chi N\) exceeds a value that is approximately 10.5, the system microphase separates and becomes ordered. The phase separation, depending on the three factors mentioned here, can result in regions or patterns that are nanoscale in one or more dimensions.

To induce phase separation, a given diblock system can be taken from a disordered to an ordered state (ODT transition) by varying temperature, which from Equation 3.2 affects \(\chi\). Experimentally, this can be performed by quenching from a thermal annealing step, although in practice this is difficult, as polymer thermal degradation prevents the exploration of some temperature ranges. In other words, inducing an ODT transition by manipulation of temperatures is not very effective. Another approach is to attempt to alter the Flory-Huggins parameter by addition of a neutral solvent, which effectively modifies \(\chi\) through new \(A\)-solvent and \(B\)-solvent interactions.

The volume fraction \(\phi_A\) also affects the phase segregation of the diblock copolymer system. Depending on the relative fractions of the \(A\) and \(B\) block copolymers different morphologies are possible. These include hexagonal(cylindrical), lamellar, spherical or bicontinuous gyroid patterns\(^{43}\). For example, with \(\phi_A = 0.5\), diblock copolymers tend to form lamellae structures with sheets of \(A\) and \(B\) of approximately equal thickness.
3.2.3 Phase-Separation and Self-Assembly as Thin Films

If these diblock copolymers are deposited on a substrate via a solution deposition technique, then they form thin films, with behavior differing from the bulk since the polymer is now also interacting with a surface. As well, there is a shift in the importance of the interfacial area between the $A$ and $B$ blocks as this area becomes more dominant as the film gets thinner. Lastly, the film thickness itself represents a geometric barrier as the block copolymer becomes confined to a surface. The two-dimensional analogues to the above 3D patterns are formed: e.g. cylindrical morphology and lamellar patterns (whether perpendicular or parallel to the surface). The self-assembly of diblock copolymer thin films then, is able to give nanoscale patterns comprised of two phases, each being composed of different polymer blocks and having distinct properties. Figure 3.2 shows an example of diblock pattern on silicon substrate by careful control of factors outlined in Section 3.2.4.

Figure 3.2 AFM tapping mode height image of polystyrene-block-polymethylmethacrylate (PS-b-PMMA) on silicon substrate. Polymer was cast from acetone solution, and solvent annealed in a toluene-rich chamber. Pattern retains fidelity for areas up to (A) $1 \mu m \times 1 \mu m$, and (B) $2.5 \mu m \times 2.5 \mu m$ large.

Control of the self-assembly process of polymer thin films has again historically been motivated by the electronics industry, since nanoscale phase separation is potentially useful for lithography applications. Some pioneering work by Russell et al. at IBM Almaden labs is direct evidence$^{44}$. As long as perfect control of patterns is guaranteed for $\sim 10^2$-$10^4$ square microns, polymer films
can already be used as lithographic masks for electronic components. In addition to patterning, polymer thin films are also potentially useful in a host of other applications, including photovoltaic, biofouling, and nanoscale devices.

### 3.2.4 Control of Polymer Thin Film Morphology

The control of thin film morphology is primarily focused on thickness, chemical composition, phase separation and morphology. There has been much success in the last decade in controlling the formation of nanoscale patterns with experiment protocols and furthering of theory. The table below outlines some of these experimental considerations in designing polymer thin films with desired phase separation, and the tools used to study them.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>How it Can Be Controlled</th>
<th>How it is Studied</th>
</tr>
</thead>
<tbody>
<tr>
<td>Thickness</td>
<td>film casting techniques: e.g. solvent drop casting, spin casting</td>
<td>- Ellipsometry/profilometry&lt;br&gt;- AFM height image with scratch test&lt;br&gt;- TEM with cross-sectional profiling</td>
</tr>
<tr>
<td>Chemical composition</td>
<td>choice of A and B blocks</td>
<td>- Tapping mode AFM (also Section 2.3.2) phase image (qualitative, planar only, nanoscale)&lt;br&gt;- X-ray photoelectron spectroscopy (XPS): poor x-y resolution, averaged over micron scales, depth information available.&lt;br&gt;- Near-field AFM (planar)&lt;br&gt;- Electron microscopes (selective staining may be required)</td>
</tr>
<tr>
<td>Morphology</td>
<td>$\phi$, $\chi$ and $N$, tuning of surface energy, $T$</td>
<td>- Any topographic imaging technique with post-processing (Fourier transforms, etc)&lt;br&gt;- X-Ray or neutron scattering (GISAXS/ GISANS)</td>
</tr>
</tbody>
</table>
3.2.5 AFM Study of Diblock Copolymer Thin Films

Studying of diblock copolymer thin films with AFM has primarily been used as a visual confirmation of nanoscale phase separation by imaging the differences in height or friction, or phase as shown in the corresponding Figure 5.2 for example.

In Chapter 5, we will be looking at thin films of the diblock copolymer system polystyrene-$block$-polyferrocenylsilane (PS-$b$-PFS) which additionally show differential conductivity. Therefore, the phases of this diblock, suitably deposited on a conducting substrate can be distinguished by looking at the electrical current image generated.

3.3 Substrate-Supported Lipid Membranes

A lipid membrane is a double layer of lipid molecules that encapsulates a cell. This barrier or surface is the primary interface with which the cell interacts with its surroundings and therefore serves a number of important functions. Membranes enriched with glycosphingolipids and cholesterol microdomains in particular (referred to as lipid rafts) act as sites for attachment of proteins\(^80\), and have been implicated in protein sorting and cell signaling functions, and even in the assembly and release mechanisms for diseases such as HIV\(^81\)-\(^83\).

A critical aspect of these lipid membranes is their ability to phase segregate into micro- or nano-domains with membranes containing phosphatidylcholine (PC), sphingomylein (SM) and cholesterol, being known to form liquid-ordered, and liquid-disordered phases\(^84\)-\(^88\). It is believed that the lateral heterogeneity of these sub-domains controls membrane interactions, since for example, the receptor distribution on cells are known to be of the same length scale\(^89\)-\(^92\). Therefore, the study of their phase segregation and physicochemical properties helps to advance our understanding of biological interactions at the cellular level.
The study of lateral heterogeneity is most intuitively understood by methods with spatial resolving capacity. However, owing to the challenges of studying lipid membranes in-situ, model membranes are instead constructed and studied while resting on a substrate support. This utilizes vesicle fusion protocols that ensure the deposition of a bilayer on the substrate. It has been shown that these substrate-supported lipids do phase-segregate in a similar manner, and therefore, with selection of biologically relevant molecules, can serve as a platform to study real membrane behavior.

Planar supported lipid bilayers can be studied using scanned-probe microscopy, fluorescence correlation, Raman and other light spectroscopy, electrical probe-based techniques, secondary ion mass spectrometry (SIMS) and many other techniques. AFM has two advantages that make it most amenable as the tool for investigation:

1. imaging mode can distinguish phase segregation via height differences.
2. force-spectroscopic techniques can be used to determine mechanical properties.

In Chapter 6, we will be looking at quantifying the mechanical properties and demonstrating lateral heterogeneity of lipid membranes using force map analysis. The challenges of implementing this analysis will be highlighted, and the nanomechanics results will also be summarized.

### 3.4 Microfluidically Self-Assembled Polymer Hydrogels as Cell-Mimics

A small part of this thesis includes work on polymer microgels. These gels are comprised of two polymers forming an interpenetrating network (IPN), and are formed from a microfluidic synthesis method. They produce hydrogels that are approximately 100 μm in diameter, which is on the order of magnitude of the size of cells.

The inclusion of two polymers allow for the fine-tuning of mechanical properties, so as that they fall within a range typical for cells. The motivation for this work arises from the extreme mechanical deformation required of white blood cells (neutrophils) as they flow through the pulmonary capillary network. Therefore, the suitability of cell mimics, among other factors,
must be able to have similar mechanical response. Given the wide tunability of these microgels, a systematic approach must be utilized to screen for the mechanical properties of these cell-mimics. Moreover, such an approach, in turn can be used to study real cells.

The AFM will be used in Chapter 7 to study these microgels via indentation measurements using a tipless cantilever to measure its mechanical response to stress, expressed in terms of elastic modulus and relaxation time. While not actually a problem in the nanoscale, this study nonetheless highlights the importance of understanding the mechanical behavior of materials, the role of AFM in furthering this understanding, and foremost, the development of a novel approach made possible by self-assembly.

3.5 References


(50) Rider, D. A.; Manners, I. *Thesis 2007*.


(64) Peng, J.; Wei, Y.; Wang, H.; Li, B.; Han, Y. Macromol Rapid Comm 2005, 26, 738.


(107) Chiantia, S.; Ries, J.; Schwille, P. *Biochimica et Biophysica Acta - Biomembranes* 2009, 1788, 225.


4 Pressure Induced Restructuring of Monolayer Film Nanojunction Produces Threshold and Power Law Conduction

4.1 Permissions

Content in this chapter is adapted with permission from Tivanski, A. V.; Li, J. K.; Walker, G. C. *Langmuir* 2008, 24, 2288. Copyright 2008 American Chemical Society.

4.2 Abstract

The electrical conduction of metal-molecule-metal junctions formed between Au-supported self-assembled monolayers of structurally different 1-hexanethiol (C₆SH), 1-decanethiol (C₁₀SH), 11-ferrocenyl-1-undecanethiol (FcC₁₁SH), and a Pt-coated atomic force microscope (AFM) tip have been measured under different compressing forces using conducting probe AFM. The observed junction resistance had two distinct power law scaling changes with compressing force. Different scaling regions were assigned to the change in the contact area, tunneling distance, number of conduction pathways and structure of the film under compression.

4.3 Introduction

Recently, charge transport through metal-molecule-metal (m-M-m) junctions has been studied intensively due to their potential applications to molecule-based electronics. In these junctions, a molecular film is typically sandwiched between two metal or semiconductor electrodes. One of the fundamental goals of all electrical conduction measurements in nanojunctions is to understand how structural and electrical properties of nanocontacts and molecules forming junctions influence charge transport through such junctions. The scanning probe microscope (SPM), with the probe serving as one of the metal electrodes has been commonly used to form and study the electrical and structural properties of nanojunctions directly, with examples including scanning tunneling microscopy (STM)¹-⁶ and conducting probe atomic force microscopy (CP-AFM)²⁷-²⁰. In SPM, the electronic properties of a nanocontact are sensitive to
the effects of deformation caused by the interaction force between the probe and the sample,9-14,16,17 thus it is important to measure force and current simultaneously to understand the mechanical and electrical properties of nanojunctions. This can be achieved directly using CP-AFM rather than STM where the contact force is not precisely known because it is not controlled independently.

Recent studies9,11-14,16,17 employed CP-AFM to demonstrate the influence of the contact force on the electrical conduction through SAMs involving alkanethiols and conjugated molecules. These measurements showed two power law scaling changes in junction resistance with applied force. While one of the power regions was related to the change in the contact area between the probe and compressing sample, another was less understood and was assigned to apparent changes in the mechanical properties of the film. Here, we report detailed measurements and quantitative modeling of electrical conduction using CP-AFM for the nanojunction formed between a Pt-coated AFM tip and an Au-supported self-assembled monolayer under different compressing forces with the objective to correlate electrical conduction through m-M-m junctions with mechanical and structural properties of the film. Specifically, we examine whether the measured dependence of the junction resistance upon compressing force can be adequately described by the power law scaling relation originating from the exponential change in the elastic contact area, tunneling gap between the probe and the sample, and through-space and through-bond hopping of electrons. While similar studies earlier performed on C₈SH - C₁₂SH ordered alkanethiol SAMs7,12,16 showed that the observed scaling changes are nearly independent of the molecular length (apart from obvious current magnitude change), we have chosen different molecular systems that are not expected to form closely packed and ordered films in the absence of applied force.

An important aspect of these measurements is that they were undertaken under an insulating bicyclohexyl solvent, which significantly decreases the capillary forces due to water condensation and, as will be discussed below, prevents a formation of the solvent layers at the AFM probe-sample interface. Earlier CP-AFM electrical conduction measurements12,16 performed in air were unable to examine the low stress (directly proportional to the sum of the adhesion force and the loading force) region in detail because of the ~10-15 nN adhesion force between the AFM probe and the monolayer due to capillary forces. The presence of a water layer at the interface between the probe and the monolayer could also lead to undesired and
complicated conduction-force dependence under applied compressing forces. In this work capillary effects are mitigated since adhesion forces are ~0.5 nN or smaller, therefore allowing one to directly probe the low stress region at nanoscale. We explore a significantly lower force regime than has been previously reported for the conductive measurements in air\textsuperscript{12,16}. These improvements enable quantitative modeling that takes into account mechanical and structural properties of the film to describe the change in the electrical conduction with applied pressure.

The molecular SAMs studied here were 1-hexanethiol (C\textsubscript{6}SH), 1-decanethiol (C\textsubscript{10}SH) and ferroceny1-1-undecanethiol (FcC\textsubscript{11}SH). Molecular systems were chosen for the following reasons: while the charge transport mechanism through these SAMs under relatively small applied biases (within ± 0.25 V) is expected to be nonresonant tunneling\textsuperscript{21}, the structure of C\textsubscript{6}SH assemblies is more disordered than for longer C\textsubscript{10}SH\textsuperscript{22,23}. The ferrocenyl-1-undecanethiol was chosen over similar length dodecanethiol due to the presence of ferrocene-end groups that are expected to decrease the contribution from chain-to-chain coupling to the electron tunneling through the film in the absence of applied force. It is important to mention that the FcC\textsubscript{11}SH molecules can undergo redox transitions under relatively large absolute biases (~±1.7 V) and negative differential resistance in current-voltage curve can be observed\textsuperscript{3,24}. Here significantly smaller applied biases are used and the charge transport mechanism through FcC\textsubscript{11}SH SAM is expected to be nonresonant tunneling. Having different structural properties of the films yet the same charge transport mechanism, allowed us to probe how the film structure influences the measured nanojunction electrical conduction under compression.

### 4.4 Experimental Details

#### 4.4.1 SAM Preparation

Self-assembled monolayers of 1-decanethiol (C\textsubscript{10}SH) were formed by exposing gold substrates (11 mm × 11 mm, 250 nm Au on 2 nm Cr on borosilicate glass, Metalhandel Schröer GmbH, Lienen, Germany) to 1 mM 1-decanethiol (Sigma-Aldrich Corp., St. Louis, MO) prepared in filtered ethanol (PTFE 0.2 μm pore filter, Whatman, NJ) with a soaking time of approximately 24 hours. Prior to soaking in the alkanethiol solution, the gold substrates were cleaned in piranha solution (1:3 of 30% H\textsubscript{2}O\textsubscript{2}/98% H\textsubscript{2}SO\textsubscript{4}) for 5 minutes, rinsed in excess ultrapure water (> 18
MΩ·cm) then excess filtered ethanol, followed by drying under a stream of nitrogen gas. 

**Caution!** Piranha solution is a very strong oxidant and is extremely dangerous to work with; gloves, goggles, and a face shield should be worn. Description of SAM preparation for FeC_{11}SH or C_{6}SH monolayers is available in the published work^{25}. After assembly, each sample was rinsed in excess solvent and dried in a stream of nitrogen gas. All preparations were performed at room temperature, and all samples were used within one day of preparation.

### 4.4.2 Conducting Probe AFM Measurements

The CP-AFM measurements on C_{10}SH SAMs were performed using a commercial AFM (MFP-3D, Asylum Research, Santa Barbara, CA) in contact-mode using a conducting probe module (ORCA, Asylum Research, Santa Barbara, CA)–essentially a modified cantilever holder with a transimpedance amplifier, and related software for operating mode-control, voltage biasing, and electrical current collection (see for Section 2.3.5 for further detail). Measurements performed on C_{6}SH and FeC_{11}SH SAMs utilized an external electronics setup with a picoammeter (Chem-Clamp, Dagan Corp., Minneapolis, MN)^{25}. In order to obtain detailed measurements of the current-contact force relationship, different fixed tip biases were applied and currents through the film were measured as a function of vertical piezo displacement, simultaneously with independent force detection between the tip and the sample. The sample was not scanned in horizontal directions. Measured currents for different contact forces over different surface locations were averaged over the number of repeated measurements performed with different AFM tips to obtain averaged force-dependent current-voltage (I-V) characteristics of the nanojunction. Results obtained with different tips were similar to within experimental error, especially for contact forces smaller than ~15 nN. The experimental error is dominated primarily by the uncertainty in the exact number of molecules forming the junction. Because the tip drifts over the surface, variations in the number of contacting molecules may cause fluctuations in the measured current.

Experiments were performed in insulating bicyclohexyl solvent (99.0%, Fluka, Switzerland) using Pt/Ir-coated rectangular Si cantilevers (CONT-Pt, Nanoworld AG, Neuchâtel, Switzerland) with spring constant nominally between 0.1 to 0.4 N/m and more accurately determined by the thermal noise method^{26}. Description of conducting probe measurements for C_{6}SH and FeC_{11}SH
experiments, as well as a control experiment utilizing n-tetradecane solvent (99.0%, Fluka, Switzerland) are described in the published work\textsuperscript{25}.

Electrical conduction measurements were performed at different fixed applied tip biases with magnitudes smaller than 0.25 V so that the current through the junction could be measured as a function of compressing interaction force. Measured currents for different interaction forces were averaged over several hundred force plots to obtain averaged, force-dependent quasi-static current-voltage ($I$-$V$) plots of the junction.

4.5 Results and Discussion

4.5.1 Force Curves Explained

The interaction of an AFM tip with the sample is captured in a force curve, which shows the loading or interaction force while the probe’s base is moving via piezoelectric actuation. In CP-AFM, a voltage bias is applied between conducting tip and substrate and the current is also collected simultaneously. Typical force and current curves for the FcC\textsubscript{11}SH SAM in bicyclohexyl solvent under a fixed tip bias of –2.8 V are shown in Figure 4.1. When the approaching AFM tip is far away from the surface, it experiences little interaction with the sample, and this is reflected by the relatively constant force values. The y-axis value in this non-contact range is therefore used as the zero force reference, with the convention that positive values are repulsive, and negative values are attractive forces. Once the tip reaches the sample surface- known as the contact point- the piezo displacement (controlling the motion of the AFM cantilever base) is re-zeroed accordingly. In this case, the convention is positive piezo displacements indicate distances away from the surface, and negative ones reflect the continued motion of the cantilever beyond the point of contact. This results in indentation of the relatively soft sample by the stiffer AFM tip. The forces after surface contact in Figure 4.1 are mainly repulsive and physically represent the force with which the cantilever pushes the probe into the surface. The negative forces experienced by the cantilever prior to surface contact are primarily the attractive electrostatic forces due to tip-sample capacitance\textsuperscript{10,13}. 
4.5.2 Adhesion Force Contribution to Interaction Force

Since the results in Figure 4.1 clearly demonstrate an increase in current with applied loading force, it confirms that this type of measurement can indeed be used to measure, with high force sensitivity, load-dependent nanojunction electrical conduction under compression. It should be noted that strictly, the net interaction force between the tip and sample is the vectorial sum of the loading force and any contributing adhesion forces. The adhesion force between the probe and the sample is the maximum attractive force measured when the tip is retracting from the surface. These forces arise in large part from the capillary effects since a water layer can exist on the film surface and also to a lesser extent from bias-dependent adhesion. However, when performing force curve measurements under solvent, adhesion force due to the former becomes negligible. For the electrical conduction measurements performed, bias magnitudes less than 0.25 V were applied and therefore any bias-dependent adhesion force contribution to the interaction force was also not significant. It should be pointed out however that under higher applied biases this factor would need to be considered. Additionally, due to a finite AFM tip-SAM contact resistance, a nonzero loading force was required to make a good electrical contact between the conductive probe and sample and effect observable currents of ~10 pA or larger. In other words, current is only observed beyond the mechanically-defined contact point with the film experiencing some indentation.
4.5.3 Choice of Operating Solvent

CP-AFM experiments were performed under fluid to reduce contamination and capillary forces during measurements. The solvent chosen was bicyclohexyl solvent due to its low volatility and high resistivity. CP-AFM measurements over just the gold surface with no molecular film present under this solvent showed that sufficient electrical contact was achieved for loading forces smaller than 1 nN (current increases steadily until saturation value of 20 nA was reached). However, this was not the case for other solvents. Earlier control experiments in n-tetradecane solvent showed that an additional loading force of at least 15 nN was required to cause observable currents larger than ~10 pA and also depicted a current-force dependence that was significantly more complicated, possibly due to the formation of solvent layers over the film surface. The absence of this highly undesirable conduction-force dependence that convolutes the interpretation of load-dependent nanojunction electrical conduction change under compression justifies the selection of bicyclohexyl solvent for our CP-AFM measurements.

4.5.4 Force Dependence of Junction Resistance

The observed current-voltage characteristics from the quasi-static I-V plots were linear at all interaction forces less than ~30 nN within ±0.25 V bias range. Ohmic dependence observed in the I-V curves are consistent with the Simmons model in the low-bias region for nonresonant tunneling through a m-M-m interface which was expected due to the large molecular gap between highest occupied molecular orbital and lowest unoccupied molecular orbital. Forces higher than 30 nN were not applied due to the noticeably quick decrease in the magnitude of observed currents indicating an apparent damage of the conductive coating on the tip. The Ohmic portion of the I-V curves were fit to straight lines, and the inverse of the determined slopes were used to define junction resistances.
As intuitively expected the junction resistances, decreased when experiencing greater interaction forces. Figure 4.2 shows log-log plots of the junction resistances for (a) FeC_{11}SH, (b) C_{6}SH, and (c) C_{10}SH SAMs show threshold and two power law-scaling regimes. Symbols are averaged data and solid lines are the corresponding linear fit.

Figure 4.2. Log-log plots of junction resistance versus interaction force for (a) FeC_{11}SH, (b) C_{6}SH and (c) C_{10}SH SAMs show threshold and two power law-scaling regimes. Symbols are averaged data and solid lines are the corresponding linear fit.
(c) $C_{10}$SH SAMs *versus* the compressing interaction force. Each data symbol shown in Figure 4.2 represents the obtained mean value of resistance for a series of repeated measurements typically with three different tips over the different surface locations. For these three molecular systems, two distinct power law-scaling regimes were observed, but with a very abrupt transition for FcC$_{11}$SH SAM films. The regions can be fit closely by straight lines in the log-log plot (shown as solid lines in Figure 4.2), with the scaling exponents for each presented in Table 4.1. It is important to point out that the threshold forces at which the resistance dependency transitions from the low to high load regimes were somewhat similar (15 nN for FcC$_{11}$SH and $\sim$13 nN for C$_6$SH and C$_{10}$SH), but the power-law scaling exponents were significantly different for these SAMs. In the next section, we will introduce several models to attempt to explain the load-dependent behavior of conduction, and explore the possible factors that can give rise to the observed junction resistances under an applied force.

<table>
<thead>
<tr>
<th>Molecule</th>
<th>Power Law Scaling Exponents ($x$)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$Resistance=(Force)^x$</td>
</tr>
<tr>
<td></td>
<td>Low Load Regime</td>
</tr>
<tr>
<td>FcC$_{11}$SH</td>
<td>-0.7</td>
</tr>
<tr>
<td>C$_6$SH</td>
<td>-9.8</td>
</tr>
<tr>
<td>C$_{10}$SH</td>
<td>-1.4</td>
</tr>
</tbody>
</table>

4.5.5 Combining the Simmons Model and Hertzian Mechanics

In the indentation of a SAM surface by an AFM tip, the following mechanical changes can occur which affects the electrical behavior of the junction:

(a) changes in the contact area between the AFM tip and the sample,

(b) change in the tunneling distance between two electrodes due to film compression,
(c) intra- and interchain order of the molecular film, and

(d) changes in conduction pathways that arise from (a)-(c).

First, a quantitative model that includes change in the elastic contact area and conduction pathways through ordered and close-packed SAM will be introduced in attempt to explain the different power law scaling regions observed in Figure 4.2.

Change in the junction resistance can be related to change in contact area and film indentation using a Hertzian elastic contact model with inclusion of adhesion forces between the probe and sample 27. The contact area, \( a^2 \), between a spherical tip of radius \( r \) penetrating into a uniform elastic film may be estimated as

\[
a^2 = \left( \frac{F \cdot r}{K} \right)^{\frac{2}{3}} \tag{Equation 4.1}
\]

where \( F \) is the interaction force and \( K \) an effective modulus equal to

\[
K = \frac{4}{3} \left[ \frac{1-v_i^2}{E_i} + \frac{1-v_s^2}{E_s} \right]^{-1} \tag{Equation 4.2}
\]

\( E_s, v_s, E_t, v_t \) are the Young’s modulus and Poisson’s ratio of the sample and Pt-coated AFM tip, respectively. The Poisson ratio for most materials is between 0.25 and 0.5. Therefore, assuming \( v_t \sim v_s \sim 0.33 \), an effective modulus can be approximated as

\[
K \approx \frac{3}{2} \frac{E_t \cdot E_s}{E_i + E_s} \tag{Equation 4.3}
\]

The indentation of the film is given by

\[
\delta = \frac{a^2}{r} = \frac{F^{\frac{2}{3}}}{K^{\frac{2}{3}} r^{\frac{1}{3}}} \tag{Equation 4.4}
\]

Although appropriate measured values for elastic modulus are not available, one can assume \( E_t=170 \text{ GPa}^{28}, E_s=20 \text{ GPa}^{29} \) for the close-packed SAMs and the tip radius of curvature to be 40 nm.
The tunneling current can be related to the contact area by

\[ I \approx a^2 \exp(-\beta d) \]  

Equation 4.5

where \( d \) is the tunneling gap and \( \beta \) is the tunneling decay coefficient which represents both electron tunneling through-bond (\( \beta_{tb} \)) and chain-to-chain (or through-space, \( \beta_{ts} \)) transport pathways. In the case of ordered and close-packed SAMs, most deformation due to film compression increases tilting of molecules with respect to the substrate normal. Therefore, the change in molecular tilt angle \( \theta \) (relative to the substrate normal) with film indentation can be easily expressed as

\[ \theta = \cos^{-1}\left(\frac{d \cos \theta_i - \delta}{d}\right) \]  

Equation 4.6

where \( \theta_i \) is the tilt angle in the absence of applied force and the indentation of the film \( \delta \) as a function of applied force is giving by Equation 4.4. (see Schematic 4.1)

Schematic 4.1 Scheme of tip-monolayer-substrate showing relevant distances in multiple conduction pathway model: chain-to-chain distance \( d_{cc} \), natural alkanethiol length \( d_n \), effective thickness \( d_i \) as a function of tilt \( \theta \). Tunneling through chain (blue) has decay coefficient \( \beta_c \) and through space (red) has coefficient \( \beta_s \).
4.5.6 The Multiple Intermolecular Hopping Model

To describe the tunneling current through such ordered and close-packed SAM in terms of both through-bond and through-space tunneling transport, the multiple intermolecular hopping model for molecular tunneling pathways is employed. According to this model, for the maximum of $N$ possible through-space hops, the chain-to-chain tunneling distance along the molecular chain with the length $d_m$ tilted at an angle $\theta$ with the intermolecular distance of $d_{cc}$ decreases by $N d_{cc} \tan \theta$. The total current can then be described as the sum of two contributions- through-bond and through space tunneling- by the following equation:

$$I \propto a^{\frac{d_m \cos \theta}{N_{tb}}} \left\{ \exp\left(-\beta_{tb} d_m \right) + \sum_{N=1}^{d_{cc}} \frac{n_s!}{(n_s - N)!N!} \exp\left(-\beta_{tb} (d_m - N d_{cc} \tan \theta)\right) \exp\left(-\beta_{ts} N d_{cc}\right) \right\}$$

where $n_s$ equals to the number of available hopping sites and is assumed equal to the number of carbon atoms along the molecular chain. By substituting Equation 4.1 and Equation 4.4 into Equation 4.7, the junction resistance $R$, which is inversely proportional to the tunneling current in the bias range applied here, depends on the interaction force as:

$$R = A F^{-2/3} \left\{ \frac{d_{cc} \cos \theta}{\sum_{N=0}^{N_{tb}}} \frac{n_s!}{(n_s - N)!N!} \exp\left(-\beta_{tb} (d_m - N d_{cc} \tan \theta) - \beta_{ts} N d_{cc}\right) \right\}^{-1}$$

where $A$ is a scaling factor and the dependence of the tilt angle $\theta$ with interaction force is given by Equation 4.4 and Equation 4.6. The molecular lengths for molecules $C_6$SH were estimated using bond lengths and van der Waals radii of each headgroup. The intermolecular distances for $C_6$SH and $C_{10}$SH SAMs were those used by Slowinski et al. and for Fc$C_{11}$SH SAM the approximate diameter of the ferrocene headgroups was used. The maximum number of hops $N$ is restricted by the size of the system, since the total through-space distance is practically limited by the molecular chain length. The molecule-specific parameters used in the model fitting are shown in Table 4.2. The through-space and through-bond tunneling decay coefficients of 1.31 Å$^{-1}$ and 0.91 Å$^{-1}$ are assumed, respectively.

Table 4.2. Fitting parameters used in the multiple conduction pathway model. The molecular length and intermolecular distances are held constant. While the number of hops is not fixed, it is constrained to a lowerbound (0 hops) and an upperbound value (maximum number shown in table) as dictated by physical constraints.
The two regions of the resistance versus interaction force data for C₆SH, C₁₀SH and FcC₁₁SH SAMs were fit separately using Equation 4.8 by adjusting three parameters: the scaling factor $A$, the elastic modulus of the sample $E_s$ and the initial tilt angle $\theta_i$. The model follows the data closely only for the lower force region for FcC₁₁SH SAM and higher force regions for C₆SH and C₁₀SH SAMs. The other regions cannot be fit satisfactorily using this model (these fits are not shown). The successful fits are plotted in Figure 4.3 by solid lines with black lines corresponding to a zero through-space hops and grey lines to a single through-space hop. The low force region for the FcC₁₁SH SAM and high force region for the C₆SH SAM can be closely fit with either 0 or 1 through-space hops, while the model with only 0 hops provides a close fit to the high force region for the C₁₀SH SAM. Within our experimental error we were not able to distinguish between the 0 and 1 hop fits as both provide an accurate fit to the data for both FcC₁₁SH and C₆SH SAMs.
Figure 4.3. Plots of junction resistance versus interaction force for FcC\textsubscript{11}SH (×), C\textsubscript{6}SH (∇) and C\textsubscript{10}SH (□) SAMs. Solid black and grey lines are fits to Equation 4.8 for 0 and 1 through-space hops, respectively.

From the analysis, the parameters of the best-fit curves from Figure 4.3 are presented in Table 4.3. The initial tilt angles determined here agree reasonably well with values reported from literature\textsuperscript{23}. The elastic moduli for FcC\textsubscript{11}SH and C\textsubscript{10}SH SAMs are similar to the elastic modulus for the close-packed SAMs ($E_s \approx 20$ GPa). On the other hand, the elastic modulus for C\textsubscript{6}SH deviates significantly from this model value and is instead similar to modulus for the gold
substrate with $E_{4u} \sim 77$ GPa. It appears that the underlying Au substrate defines the elastic response of the film to compression in the region of larger applied forces.

Table 4.3. Best-fit elastic modulus and initial angle values for each molecule using the multiple conduction pathways model, for the fits in Figure 4.3.

<table>
<thead>
<tr>
<th>Molecule</th>
<th>Best-fit Elastic Modulus ($E_s$)</th>
<th>Best-fit Initial Angle ($\theta_i$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>FcC$_{11}$SH</td>
<td>16 GPa</td>
<td>20°</td>
</tr>
<tr>
<td>C$_6$SH</td>
<td>65 GPa</td>
<td>35°</td>
</tr>
<tr>
<td>C$_{10}$SH</td>
<td>17 GPa</td>
<td>20°</td>
</tr>
</tbody>
</table>

These results demonstrate that a single force region for each monolayer can be described by a model that includes zero hops and these regions can also be described by the multiple intermolecular hopping and increased tilting of molecules due to the monolayer compression for FcC$_{11}$SH and C$_6$SH SAMs. However, since the change in the number of through-space hops cannot adequately describe other force regions, the possibility that the two observed power-law scaling regions in the resistance versus interaction force plots shown in Figure 4.2 originate only from a changing number of conduction pathways is excluded. The inability of this model to accurately describe other force regions indicates that the monolayer film compression does not simply tilt the molecules and that another type of the pressure-induced restructuring must take place. It is therefore necessary to further explore the possibility that force induces structural changes, which alter the film stiffness under compression, and thereby induce load-dependent conduction. In the following section, a quantitative model that includes changes in the elastic contact area and tunneling distance will be introduced together with consideration of the film structure to explain the different power law scaling regions observed in Figure 4.2.

4.5.7 Elastic Contact Model with Mechanical Restructuring

Rather than separate the contributions of through-space and through-bond tunneling pathways to conduction as in Equation 4.7 and Equation 4.8, we use an effective decay coefficient of 12 nm$^{-1}$
to remove the combinatorial/statistical aspect, which would have complicated the curve fitting\textsuperscript{24,31}. Starting from Equation 4.5, the junction resistance $R$, which is inversely proportional to the tunneling current in the bias range applied here, depends on the interaction force as:

$$R \approx F^{-2/3} \exp\left(-S \cdot F^{2/3}\right)$$  \hspace{1cm} \text{Equation 4.9a}

$$RF^{2/3} \approx \exp\left(-S \cdot F^{2/3}\right)$$  \hspace{1cm} \text{Equation 4.9b}

where,

$$S = \frac{\beta}{K^{2/3} \cdot r^{1/3}}$$  \hspace{1cm} \text{Equation 4.10}

with the tunneling distance $d$ expressed in terms of force and $S$ being a collection of constants. Hence, from Equation 4.9b, a plot of $ln(RF^{2/3})$ versus $F^{2/3}$ should be linear with a slope of $-S$.

For a typical tip radius of curvature of 40 nm, the expected slope for the electrical conduction change with compressing force over a closely packed and ordered SAM with the elastic modulus of $E_s = 20 \text{ GPa}$ is $-S = -0.4 \text{ nN}^{2/3}$. It is worthwhile to note that if the tunneling process through such a film is dominated by the through-bond tunneling (for a particular range of forces), the slope in a plot $ln(RF^{2/3})$ versus $F^{2/3}$ becomes zero and, from Equation 4.9a, the resistance is expected to be directly proportional to $F^{-2/3}$. This possibility would indicate that most of the deformation only changes the tilt angle of the molecules forming a closely packed and ordered SAM and does not compress the molecular length. Since the molecular length is equal to the tunneling distance for the through-bond tunneling process, the latter remains unchanged in this case. The tunneling current then is defined by the change in the contact area between the AFM probe and the monolayer, which depends on the interaction force in accordance with Equation 4.1. In terms of the previously described multiple intermolecular hopping model, it corresponds to the 0 through-space hops.
Figure 4.4. Plot of $\ln(R^{2/3})$ as a function of $F^{2/3}$ for (a) FcC$_{11}$SH (points), C$_6$SH (triangles) and (b) C$_{10}$SH (crosses) SAMs with the corresponding linear fit (solid lines).

Figure 4.4a, b shows a plot of $\ln(R^{2/3})$ versus $F^{2/3}$ for FcC$_{11}$SH, C$_6$SH and C$_{10}$SH SAMs where data in both scaling regions closely followed a linear dependence, as predicted by the above model. Data in different regions were fit to straight lines, as shown by the solid lines in Figure 4.4, and whose $-S$ slopes are reported in Table 4.4. Also as shown in the table, the overall dependence for these films can be separated into three common regions: the first region with a slope of -0.01 nN$^{-2/3}$ for FcC$_{11}$SH, a second region with slopes of -0.93, -2.3 and -0.36 nN$^{-2/3}$ for FcC$_{11}$SH, C$_6$SH and C$_{10}$SH, respectively, and a third region with slopes of -0.12 and ~0 nN$^{-2/3}$ for C$_6$SH and C$_{10}$SH, respectively. The three regions are distinguished according to the calculated $-S$ slopes and the apparent structural state of the SAM, which will be discussed below. For reference, the model calculation for a close-packed SAM with slope of -0.40 nN$^{-2/3}$ falls into Region II.
Table 4.4. Slopes of line fits to Figure 4.4 for each molecule. Low and high-load regimes are distinguished. Behaviors for each molecule / slope combination is categorized into three groups. The estimated slope for a model close-packed SAM is also provided.

<table>
<thead>
<tr>
<th>Force Regime</th>
<th>Molecule</th>
<th>Slope [nN^{-2/3}]</th>
<th>Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>Low-load</td>
<td>FcC_{11}SH</td>
<td>-0.01</td>
<td>I</td>
</tr>
<tr>
<td></td>
<td>C_6SH</td>
<td>-2.3</td>
<td>II</td>
</tr>
<tr>
<td></td>
<td>C_{10}SH</td>
<td>-0.36</td>
<td>II</td>
</tr>
<tr>
<td>High-load</td>
<td>FcC_{11}SH</td>
<td>-0.93</td>
<td>II</td>
</tr>
<tr>
<td></td>
<td>C_6SH</td>
<td>-0.12</td>
<td>III</td>
</tr>
<tr>
<td></td>
<td>C_{10}SH</td>
<td>-0</td>
<td>III</td>
</tr>
<tr>
<td></td>
<td>Model SAM</td>
<td>-0.4</td>
<td>II</td>
</tr>
</tbody>
</table>

Region I in Table 4.4 only contains the low-load regime for FcC_{11}SH. It shows a linear decrease with a slope of almost zero, thus the junction resistance is directly proportional to $F^{-2/3}$, which represents the case when the contact area change is the dominant contribution to the junction resistance. The presence of Fc-end groups in the FcC_{11}SH SAM decreases the chain-to-chain coupling to the electron tunneling through the film (the coverage density of the FcC_{11}SH SAM is roughly two times smaller than that for the similar length alkanethiol\textsuperscript{22}), so under relatively small applied forces (less than 15 nN in our case), the through-bond charge transfer is the dominant tunneling mechanism. Again, within the small force regime, the change in the junction resistance appears to be limited by a change in the number of conductive contacts between the AFM tip and the sample, which is proportional to the contact area. The slight deviation from zero slope is likely due to experimental error and possibly a small contribution coming from chain-to-chain tunneling. These results are consistent with the multiple intermolecular hopping model described above, where this region can be fit with both 0 and 1 hops corresponding to the through bond and through space tunneling mechanisms, respectively.

Region II in Figure 4.4 shows a linear decrease with slopes of -0.93, -2.3 and -0.36 nN$^{-2/3}$ for FcC_{11}SH, C_{6}SH and C_{10}SH, respectively. The observed slope for closely packed and ordered
C_{10}SH monolayer is similar to the one predicted from the above model, however, slopes for the C_{6}SH and FcC_{11}SH films are ~5.8 and ~2.4 times larger than expected, respectively. The origin of this deviation can be found in the structural properties of the films. The structure of C_{6}SH assemblies was shown to be significantly more disordered than that of longer chains such as C_{10}SH. Such a structure would result from a combination of conformationally disordered and thermally disordered alkyl chains because of the presence of gauche kinks and weak interchain interactions, respectively. Under compression, the film structure becomes more ordered due to increase in molecular packing, eventually forming closely packed C_{6}SH assemblies at a compression force around 13 nN, similar to the longer chains. Barrena et al.\textsuperscript{32} studied the change of the molecular packing in SAMs of similar molecular length alkanethiols on Au induced by external pressure. They observed, consistent with our work, changes in film thickness accompanied by simultaneous change in friction versus applied load that were attributed to a collective transition to denser molecular configurations. In the case of FcC_{11}SH SAM, its structure is also expected to be more disordered as compared with C_{10}SH due to the presence of Fc-end groups that prevent formation of a closely packed monolayer. In terms of the previously described model, a change in the molecular packing under compression forces can be represented as the tip indents through more compressible film; assuming the elastic moduli of C_{6}SH and FcC_{11}SH sample are $E_s=1.2$ GPa and $E_s=4.9$ GPa, respectively, the observed slope of -2.3 and -0.95 nN^{-2/3} can be obtained. The change in the molecular packing under compression invalidates in this range of applied forces, the assumption that film compression only increases the tilting of molecules, and explains the inability of the multiple intermolecular hopping model to accurately describe these regions.

The third region in Figure 4.4 showed a linear decrease with slopes of -0.12 and ~0 nN^{-2/3} for C_{6}SH and C_{10}SH, respectively. A zero slope for C_{10}SH similar to Region I for FcC_{11}SH represents the case when the contact area change is the dominant contribution to the junction resistance and the charge transport is limited by the through bond tunneling mechanism. This is also consistent with the multiple intermolecular hopping model that fits these two force regions for both molecular films assuming 0 hops. The two force regions for C_{10}SH therefore correspond to the change from combined through-space and through-bond tunneling pathways to a single through-bond pathway. It is important to mention that the determined slope of ~0 for C_{10}SH is not exact due to somewhat higher variations in the measured resistance values in this
region of applied forces as compared with the lower force region. An increase in the number of the gauche defects in the alkyl chains under such large compression forces (15 – 30 nN in this case) is likely the origin of these variations. Such drastic pressure induced structural changes in the alkyl chains decrease chain-to-chain coupling and leads to the predominantly through-bond charge transport mechanism.

The observed slope of $-0.12 \, \text{nN}^{-2/3}$ in Region III for C$_6$SH is significantly smaller than the one predicted by the elastic compression model over a closely packed and compressed monolayer film presented above. The observed deviation in this range of applied forces ($> \sim 13 \, \text{nN}$) likely lies in the change of the mechanical properties of the compressed films. For these films, $\sim 13 \, \text{nN}$ corresponds to a film stress of $\sim 1-1.5 \, \text{GPa}$ (calculated from the force with the contact area estimated using Equation 4.1). Others have observed significant changes in the elastic properties of alkanethiols SAMs at a stress of $\sim 1 \, \text{GPa}$, which supports our hypothesis. It appears that in this force range the elastic modulus of C$_6$SH becomes similar or greater than that for the underlying Au substrate and the substrate dominates the elastic properties of the film. Assuming an elastic modulus of the C$_6$SH-Au nanostructure similar to the Au substrate with $E_s \sim 77 \, \text{GPa}$, the above model predicts a slope of $-0.19 \, \text{nN}^{-2/3}$, reasonably close to $-0.12 \, \text{nN}^{-2/3}$ observed here. The significantly higher elastic modulus obtained here is also in agreement with the multiple intermolecular hopping model, where an elastic modulus of 65 GPa was determined.

We note that unlike the C$_6$SH and C$_{10}$SH SAMs, the FcC$_{11}$SH sample shows a discrete transition between two scaling regions occurring at around 15 nN. We argue that the origin of the discrete change between the first and second regions for the FcC$_{11}$SH SAM is due to significant structural reorganizations of ferrocene-end groups. The absence of such a discrete change for the C$_6$SH SAM is consistent with different structural properties of the films, as discussed previously. Additional experimental studies, beyond the scope of the present work, would be necessary to determine the origin of this discrete change.

To underline the importance of the size of a nanocontact, we note a comparison to work from A. Joyce et al., where a much larger tip radius of curvature ($\sim 300 \, \text{nm}$) was used than here ($\sim 40 \, \text{nm}$). This difference leads to several orders of magnitude larger contact area between the tip and the monolayer and number of molecules forming the junction. The corresponding stresses ranged from 0 to $\sim 100 \, \text{MPa}$ in this earlier work - significantly narrower than the range explored.
here. The fact that no distinct scaling changes were observed there suggests that a significantly larger contact area can hide nanoscopic properties of the film.

### 4.6 Conclusion

In summary, electrical conduction through molecular m-M-m junctions under compression was investigated using CP-AFM. Observed junction resistances showed two distinct power laws scaling changes with compressing interaction force for 1-hexanethiol, 1-decanethiol and ferrocenyl-1-undecanethiol SAMs. These regions were quantitatively described using two analytical models for the change in contact area, tunneling gap between the AFM tip and the sample, number of conduction pathways and different molecular structures of the films. The model based primarily on conduction pathways could not sufficiently explain the changes in conduction behavior under greater compression, and therefore, the model that also factored in structural changes was more appropriate.

### 4.7 Contribution

The self-assembled monolayer preparation, AFM data measurement and subsequent analysis of the decanethiol (C\textsubscript{10}SH) system were performed by the author. Additionally, evaluation of the suitability of the multiple-hopping model by fitting of data for all three systems (FcC\textsubscript{11}SH, C\textsubscript{6}SH, C\textsubscript{10}SH) were the work of the author as well. J.K.L. was a coauthor on a paper published on this\textsuperscript{25}. This chapter is based on that manuscript, with the emphasis on the parts undertaken by J.K.L.

### 4.8 References

(5) Xu, B.; Tao, N. Science 2003, 301, 1221.

5 Differential Conductivity in Self-Assembled Nanodomains of a Diblock Copolymer using Polystyrene-\textit{b}-Polyferrocenylethylmethylsilane

5.1 Permissions

Content in this chapter is adapted with permission from the pre-peer reviewed version of Li, J. K.; Zou, S.; Rider, D. A.; Manners, I.; Walker, G. C. \textit{Adv Mater} \textbf{2008}, 20, 1989. Copyright 2008 Wiley-VCH Verlag GmbH & Co.

5.2 Abstract

Conducting probe atomic force microscopy (CP-AFM) was used to investigate the electronic properties of polystyrene-\textit{block}-polyferrocenylsilane (PS-\textit{b}-PFS) diblock copolymer thin films. In this system, cylindrical domains of polyferrocenylsilane, a weak semiconductor, are surrounded by polystyrene, an insulating material. Mapping the electrical current response of the film to an applied voltage bias showed a correlation with the location of PS and PFS domains as evidenced by concurrent topographical imaging. PFS was additionally observed to exhibit diode-like behavior. Experiments were performed using contact mode AFM operation under ambient conditions and also in low-oxygen environments. This work effectively demonstrates two-terminal, vertically configured nanoscale electronic elements that are self-assembled into electrically isolated regions in a thin-film configuration. Therefore, the electronic properties of PFS, in addition to the locally ordered configuration afforded by the self-assembly process, represents a system that may have possible device applications.

5.3 Introduction

There has been much work in the field of nano-scale electronics based on polymeric materials instead of the traditional semiconductor platform, with success in the fabrication of analogues to silicon-based devices, and promising development of novel configurations\textsuperscript{1-13}. However, the paradigm shift is not so easily achieved; for example, many such devices must be fabricated with
adaptations of top-down technology (e.g., focused-ion-beam, etc.) that may not be well suited to large-scale processing, or alternatively, require operation in highly-specialized conditions (e.g., vacuum, low-temperature, etc). Here, we take advantage of the properties inherent in electroactive polymers along with the diblock copolymer class to demonstrate two-terminal, vertically configured nanoscale electronic elements that are self-assembled into electrically isolated regions in a thin-film configuration.

Diblock copolymers are archetypal systems where self-assembly leads to nanoscale features\textsuperscript{14-16}. Because these materials are polymers, they are highly processible and are able to conform to various surface morphologies. This suggests their application as conductive or resistive elements in polymer electronics. We provide herein the first report of the differential conductivity of the blocks in such a system, specifically, polystyrene-\textit{block}-polyferrocenylethylmethysilane (PS-\textit{b}-PFEMS). We chose PS-\textit{b}-PFEMS because one block (PFEMS) is expected to be electroactive and the other is not\textsuperscript{17,18}.

\section*{5.4 Experimental}

\subsection*{5.4.1 Polymer Synthesis}

Synthesis of PS-\textit{b}-PFEMS are as described elsewhere\textsuperscript{18,19}. Briefly the polymers were were synthesized from styrene and ethylmethysila[1]ferrocenophane via sequential anionic polymerization. The diblock copolymer used in this study was PS-\textit{b}-PFEMS with PFEMS volume fraction of $\phi_{\text{PFEMS}} = 0.36$. It contains a 40.5 kDa PS segment, and a 27.7 kDa PFEMS block, and an overall polydispersity index index (PDI) of 1.04.

\subsection*{5.4.2 Thin Film Preparation}

PS-\textit{b}-PFEMS films were deposited on gold substrates (11 mm $\times$ 11 mm, 250 nm Au on 2 nm Cr on borosilicate glass, Metalhandel Schröer GmbH, Lienen, Germany). Gold substrates were cleaned with piranha solution(1:3 of 30% H$_2$O$_2$/98% H$_2$SO$_4$) for 5 minutes and rinsed in ultrapure water (> 18 M$\Omega$·cm) for 1 minute prior to thin film deposition. Caution! Piranha solution is a very strong oxidant and is extremely dangerous to work with; gloves, goggles, and a face shield should be worn.
PS-b-PFEMS films were prepared by spin-coating 0.25-1.0% w/w polymer solutions in toluene onto substrates. Solutions were filtered with a Whatman 13 mm GD/X nylon syringe filter (0.45 μm pore size) immediately prior to spin-coating. The gold substrates were coated with the polymer solution and immediately accelerated to 600 rpm for 18 s, followed by 60 s at 1000 rpm.

5.4.3 Conducting Probe Atomic Force Microscopy Measurements

Topographic imaging, and force and electrical measurements were performed using a commercial atomic force microscope (MFP-3D, Asylum Research, Santa Barbara, CA) with conducting probe module—essentially a modified cantilever holder with a transimpedance amplifier, and related software for operating mode-control, voltage biasing, and electrical current collection - all commercially available (ORCA kit, Asylum Research, Santa Barbara). The AFM tips used were Pt/Ir-coated rectangular Si cantilevers (CONT-Pt, Nanoworld AG, Neuchâtel, Switzerland), having spring constants in the range of 0.1 to 0.4 N/m.

Current imaging was collected in contact mode, simultaneously with height imaging. Current-voltage curves were collected over both PFS and PS regions of the thin film. Experiments were carried out in air under ambient conditions, as well as under a low O₂ environment (2-3% O₂). For the latter, nitrogen gas was flowed into the instrument chamber, and a gas sensor was used to monitor oxygen levels. N₂ inflow was stopped once a set oxygen level was reached in order to maintain low-noise conditions (N₂ outflow was sufficiently slow). Voltage biases used were within the range of –10 V to +10 V, as these were the limits of the internal voltage source within the MFP controller.

5.4.4 Modeling the Quasi-Static I-V Data

The I-V data was fit to a quadratic equation, with the result:

\[ I(V) = 5.8 \times 10^{-14}(V - 1.1)^2 + 2.1 \times 10^{-13} \]  
Equation 5.1
The $V^2$ dependence of current (current density) indicates a space charge limited current. Therefore, the second-order coefficient was utilized in the Mott-Gurney law to determine an effective hole mobility.

## 5.5 Results and Discussion

PS-$b$-PFEMS films were prepared on Au substrates and placed in a conducting probe AFM (CP-AFM) setup as shown in Figure 5.1a. The film is approximately 30 nm in thickness with cylindrical domains that are approximately the same magnitude in diameter. Based on volume fraction of PFEMS ($\phi_{PFEMS}=0.36$), the expected morphology of the thin-film is PFEMS cylindrical domains in a PS matrix (Figure 5.1b)$^{19,20}$. The diameter and center-to-center distance of the hexagonally packed cylindrical array has been estimated as 30 nm and 43 nm, respectively$^{21}$.

Figure 5.1 Schematic illustration of (a) CP-AFM setup, and (b) cross-section showing expected morphology of polystyrene-$b$-poly(ferrocenylethylmethylsilane) solid-state self-assembly on Au substrate.
In the CP-AFM setup, a tip is rastered across the sample surface using a fixed tip-sample interaction force as a feedback parameter to generate topography (height), and simultaneously, a voltage bias is applied across the tip and sample (relative to the sample) to generate a map of the corresponding electrical current response. A platinum/iridium alloy-coated tip is used in contact mode to obtain the height images, with additional software and hardware to bias the sample at a fixed voltage and collect the current signal. Experiments were performed mainly in ambient air but also in low-oxygen conditions. Each image was taken on a fresh area of film, but the reproducibility of current images was also separately studied by scanning the same area at different voltages. The information made available by electrical current imaging can give insight into the characteristics of the thin film, in particular, the arrangement of conducting and non-conducting regions.
Figure 5.2. AFM images of PS-b-PFEMS thin film, 1 µm × 1 µm. (a) Height image with (b) corresponding current image of sample scanned at −5 V. (c-e) Current images at large negative applied biases (−7 V, −9 V and −8 V respectively) show stronger response of PFEMS region.

Figure 5.2a shows a typical height image under −5 V bias in which we observe the predicted morphology with the cylinders lying horizontally. The corresponding current image Figure 5.2b shows the same morphology as in the height image with a negative correlation in signal (i.e. lower height corresponds to higher current). Under this applied bias, the current signal over the
matrix region remains at the zero level whereas the cylindrical domains exhibit currents that are approximately 1 pA higher. Although the contrast is low, the two domains are distinguishable. Since it is expected that electro-active PFEMS is responsive and insulating PS is non-responsive to current, these images confirm the assignment of polymer blocks to particular domains on the basis of their electrical properties, consistent with the height images in Figure 5.2a. Increasing the magnitude of the negative bias induced greater current response over the PFEMS but not the PS region as shown in Figure 5.2c, d, and e, which were imaged at –7 V, –9 V and –8 V bias, respectively.

Applying positive voltage biases, however, did not produce a response in the PFEMS region of the film. In direct contrast to Figure 5.2e, f shows a current image at +8 V in which PFEMS and PS regions are indistinguishable. It is also worthwhile to note that switching the bias in the middle of a scan showed consistent observations, with non-zero currents only appearing at negative but not positive voltage bias.
Figure 5.3. (a) PS-b-PFEMS (○) shows a non-zero (negative) linear correlation of current and height with large negative voltage bias, while a film of PS-b-PMMA (■) has poor correlation. (b) Current-voltage graph of PFEMS (grey) and PS (black) swept from −9 V to +9 V.

The plot in Figure 5.3a was compiled from a series of image scans performed at different fixed voltage biases and depicts a height to current correlation in PS-b-PFEMS and a reference polystyrene-block-polymethylmethacrylate (PS-b-PMMA) film. With the former, a strong correlation can be observed with large negative biases, and poor correlation with positive voltages. This is not present in the reference sample of PS-b-PMMA. To investigate this phenomenon further, current-voltage (I-V) sweeps were performed by using a ‘pick-a-point’ technique wherein the AFM tip selectively probed a PFEMS or PS region as determined by an
initial image scan step. The voltage was swept from $-9$ to $+9$ V ($\text{scan rate} = 4 \, \text{V} \cdot \text{s}^{-1}$) and currents measured for each representative region (Figure 5.3b). The current response of the PS curve was flat and remained at the zero level over the entire voltage range, whereas PFEMS demonstrated an asymmetric behavior with large current response when using negative voltages. Note also from the current image (Figure 5.2d and b) that as bias magnitude increased, the apparent diameters of the cylinders seemed to increase.

The stability of these images at room temperature and ambient pressure is excellent, owing to the rigid nature of the PS ($T_g = 95 \, ^\circ \text{C}$), which holds the PFEMS with much lower ($T_g = 15 \, ^\circ \text{C}$) in place and preserves the overall morphology. In contrast, preparing an equivalent film of physisorbed PFEMS homopolymer was relatively difficult to achieve as it readily dewets from the substrate.

Signal reproducibility is also well-demonstrated. Figure 5.4a shows a series of three images depicting the same area of the film scanned in succession at three different voltage biases: $-9$ V, $+9$ V, and $-9$ V, respectively. One can observe that the signal response has been effectively turned on, off, and on again. Moreover, the two images that are ‘on’ show nearly identical pattern and equal signal intensity. The experiments were performed repeatedly over a span of around 6 months and images taken have not shown any discernible degradation in either film quality or electrical response.
Figure 5.4. (a) Reproducibility of current signal in PS-b-PFEMS. Shown in (i)→(iii) is the same area of film imaged sequentially under biases that turn the current response on-off-on. (b) Current image of PS-b-PFEMS at (i) ambient conditions (20.9% O₂) and (ii) low-oxygen conditions (2-3% O₂).

We compile a quasi-static I-V curve in the voltage range -4 to -9V by finding the averaging current over 20 PFEMS domains in each current image. We analyze these average currents as a function of voltage and attempt to fit it to a conduction model. (See Figure 5.5a) The quasi-static I-V curve exhibits a power law relationship, with $I \sim V^{2.4}$, which is indicative of a space-charge limited current. As this is very nearly a quadratic relationship, we fit the data accordingly to
yield an equation with a coefficient of $5.8 \times 10^{-14}$ A V$^{-2}$ for the second order term (See Experimental Section above). We analyzed this result using the Mott-Gurney law for current density in a sandwich-type structure, defined as$^{22}$:

$$J = \frac{9}{8} \Theta \mu \varepsilon_r \varepsilon_0 \left( \frac{V^2}{L^3} \right)$$

Equation 5.2

where $\mu$ is the carrier mobility (assumed to be holes), $\varepsilon_r$ and $\varepsilon_0$ are the relative and vacuum permittivities respectively, $\Theta$ is the trapping factor, and $L$ is the separation between the two electrodes. Assuming the three spatial dimensions of the (tip-PFEMS-Au) system are the same order of magnitude, ($\sim L$), which is only a first-order simplifying approximation, we can rewrite the equation in terms of current as: $I = \frac{9}{8L} \Theta \mu \varepsilon_r \varepsilon_0 V^2$. We note that we did not apply large enough voltages to probe the region where current would no longer be space-charge limited.

Estimating $\varepsilon_r$ as the square of the refractive index ($n=1.66$)$^{23}$ and equating this to the fitted coefficient gives a value for the effective hole mobility $\mu_{\text{eff}} = \Theta \mu = 6.3 \times 10^{-7}$ cm$^2$ V$^{-1}$ s$^{-1}$, which is about an order of magnitude smaller than the hole mobility measured at similar field strengths ($\sim10^5$ V cm$^{-1}$) for ferrocene-based polymers such as amorphous poly(ferrocenylmethylphenylsilane) ($\mu = 7.0 \times 10^{-6}$ cm$^2$ V$^{-1}$ s$^{-1}$)$^{18}$. 

Figure 5.5. (a) Log-log plot showing the current averaged over twenty PFEMS domains in each current image. Solid line is a fit to a quadratic equation (See Section 5.4). (b) Observed widening of PFEMS cylinders in the current image (Figure 5.4) with increasing magnitude of applied voltage. Average width was obtained by binarizing each current image using a threshold of 0.5 pA, and calculating an average width for twenty PFEMS cylinders.

The smaller calculated hole mobility suggests the existence of trap states whose nature and distribution were not specifically probed. The existence of trap states is further corroborated by the observed widening of PFEMS cylinders in the current image with increasing magnitude of applied voltage. Figure 5.5b was obtained by binarizing each current image using a threshold of 0.5 pA, and calculating an average width for a number of PFEMS cylinders. It is also of note that
the apparent widths of PFEMS cylinders as observed in the current images are smaller than the expected value of 30 nm. This value is only achieved at the highest voltages.

The increase in diameter is nearly 100% over a span of 5 V, which is much greater than what can be accounted for from electrostatic repulsions between neighboring charged ferrocene groups\textsuperscript{24,25}. Instead, this suggests trap states are spatially distributed in a radial manner from the center of the PFEMS cylinders, with the apparent width increases being a result of the probing of new trap states- only energetically possible at higher bias magnitudes.

Conduction is likely via a hopping transport occurring through Fe centers in the ferrocene unit of the PFEMS, although elucidating the exact mechanism as well as the asymmetric current behavior is still in progress and a series of control experiments performed have begun to rule out several possibilities. Tunneling, for example, is not the mechanism involved, since the exponential distance dependence of this mechanism would inhibit any conduction across the 30 nm film. By comparison, the PS block of this polymer (of similar thickness) and another reference PS-\textit{b}-PMMA sample expected to self-assemble to the same morphology but with no electroactive component, did not show any current response. Considering also the observed swelling phenomenon, we conclude that electrochemical oxidation is occurring in the ferrocene unit, with Fe(II) oxidizing to Fe(III) ions during the application of a voltage bias\textsuperscript{26,27}. Additional CP-AFM measurements carried out under a low-oxygen environment (2-3% O\textsubscript{2}) reveal the same magnitude of electrical signal as those collected under ambient conditions, indicating that ambient O\textsubscript{2} does not participate in this system. (See Figure 5.4b and Experimental Section). It may be that in air for a phase segregated polar- nonpolar diblock film, the outermost molecules even on the predominately polar domains are nonpolar. XPS data (not shown) support this assertion for our sample, but are not conclusive. If the outermost layer is PS, then an asymmetric I-V curve would result, independent of the space-charge model used above.

5.6 Conclusions

In summary, CP-AFM imaging of PS-\textit{b}-PFEMS thin films reveals patterns comprised of semiconducting and non-conducting regions corresponding to each segment of the polymer. The demonstration of differential conductivities in a nanoscale phase segregated diblock copolymer
has never been reported and is in contrast to other works such as nanowire systems grown via templated techniques\textsuperscript{28} or metallic or semiconducting particles which preferentially aggregate to one block of a diblock copolymer\textsuperscript{29}. These self-assembled thin films introduce a viable platform for patterning stable and reproducible elements for polymeric nanoscale electronics. Additionally, the asymmetric I-V behavior exhibited by the PFEMS system shows that it is not merely a passive element. One can envision the PS-\textit{b}-PFEMS system as a patterning layer for nanocapacitors. Finally, the electromechanical behavior of poly(ferrocenylethylmethylsilane) introduces another dimension to the design of electrical components that can induce or respond to mechanical actuation.

5.7 Contribution

All experimental work except diblock copolymer synthesis were carried out by the J.K.L\textsuperscript{30}: thin film preparation, AFM measurements, data analysis.

5.8 References


(14) Alexandridis, P.; Lindman, B. Amphiphilic Block Copolymers; Elsevier: Amsterdam, 2000.
(16) Park, C.; Yoon, J.; Thomas, E. Polymer 2003, 44, 6725.
(21) The bulk center-to-center distance $d_{cc} = 4\pi/(\sqrt{3} \times q^*)$; where $q^*$ is the primary scattering peak as determined by small angle X-ray scattering. The bulk cylindrical diameter $d_{cyl} = 2\sqrt{(d_{cc}^2 \times f_{PFEMS}/\pi)}$; where the volume prism defined by three close packed cylinders contains half a cylinder in its volume.
(22) Lampert, M. Phys Rev 1956, 103, 1648.
6 Force Map Analysis Toolset for Batched, Automated, and High-throughput Processing in the Study of Supported Lipid Bilayers by AFM

6.1 Abstract

Investigating the structural and mechanical properties of lipid bilayer membrane systems is vital in elucidating their biological function. One route to directly correlate the morphology of phase-segregated membranes with their indentation and rupture mechanics is the collection of AFM force maps. These force maps, while containing rich mechanical information, require lengthy processing time due to the large number of force curves needed to attain a high spatial resolution. A force curve analysis toolset was created to perform data extraction, calculation and reporting specifically in studying lipid membrane morphology and mechanical stability. The procedure was automated to allow for high-throughput processing of force maps with greatly reduced processing time. The resulting program was successfully used in systematically analyzing a number of supported lipid membrane systems in the investigation of their structure and nanomechanics.

6.2 Introduction

Since its introduction nearly twenty-five years ago, the AFM has been extensively used in a large number of systems to investigate interactions at sub-micron length scales, and forces in the pico-to nano-Newton regime\(^1\)\(^-\)\(^6\). In AFM imaging, a tip at the end of a cantilever is raster-scanned across an area of interest, and the cantilever’s deflection for example, is used as feedback to adjust the height of the cantilever base relative to the sample surface. In this way, heterogeneities on the sample can be distinguished via relative differences in actual heights of the region’s domains, or, from chemical differences that still effectively cause the cantilever base to adjust its height.

AFM force spectroscopy\(^7\), offers a more rigorous and quantitative approach by inducing a tip-sample interaction event and recording the same channels of data over a period of time. In addition, the induced tip-sample interaction is done without lateral movement of the sample
relative to the cantilever, so the interpretation of the response is simplified. And, the data that is collected over time can be analyzed for kinetics and mechanics for example, and with repetition, statistics.

By extension, AFM force mapping- the collection of an array of force curves with record of each one’s spatial position - would be able to elucidate the spatial dependence of the interaction forces. The collection and analysis of force maps can be logically organized in the flowchart shown in Schematic 6.1. While there have been some mechanical investigations via AFM force maps in literature, it is not as commonplace (especially at high resolutions) because of the inherent difficulties in maintaining stability during the lengthy data collection stage, and in the subsequent analysis of a very large amount of force curves, which are both necessary in obtaining sufficiently spatially dense data to have high resolution.

The issue of data collection is largely resolved due to stable setups and increased functionality in current instrumentation as well as automated force curve-collection over the region of interest. Force maps of greater than 64 × 64 pixels performed over a 3 μm × 3 μm area with spatial resolution of ~50 nm is readily achievable and typically takes half an hour to collect. The challenge is in the extraction or calculation of several parameters during analysis of the thousands of force curves for each force map. As mentioned, while data collection is already
automated by sophisticated software packaged with the instrument, batched data analysis is not provided. One large barrier to automation is the wide range and specificity of end-user needs which makes the creation of a universally applicable program very difficult. The aim of this work, therefore, is to create a tool for the automated, batched analysis of force curves.

The system that we chose to investigate is that of supported lipid bilayers. The study of lipid bilayer membranes is a very active area of research in the biophysical sciences due to the ubiquity and importance of these types of platforms in a host of cellular functions. For example, membranes enriched with cholesterol and sphingolipids have been implicated in protein binding, signal transduction, and in programmed cell death. AFM imaging mode has been used with much success for these systems, due to its ability to visualize and probe at nanometer length scales. However, as studies evolve from model lipid membranes to more biologically-relevant, multi-component mimics, the AFM’s capabilities in distinguishing among heterogeneous micro-domains in these systems may not be sufficient, and hence “imaging” by the generation of force maps provides a more rich and accurate picture. While there have been earlier work on batched data analysis, in this particular investigation, we establish a comprehensive package comprised of an automation process to extract mechanical properties of lipid membranes undergoing AFM–tip indentation with the occurrence of membrane rupture. The universality of the tool is also demonstrated by its adaptation to process data from three different brands of commercial AFMs, as well as the ability to utilize several different mechanical models for the analysis of lipid membranes.

6.3 AFM-tip Indentation with Membrane Breakthrough

The schematic for an AFM tip approaching, indenting, breaking through, and finally withdrawing from a supported lipid bilayer is shown in Schematic 6.2. When the approaching tip is still sufficiently far from the surface (the non-contact region), interaction forces are negligible (a) until the tip contacts the surface (b). As the approach is continued, an increasing, non-zero force is observed, and, since the sample is softer than the tip, the sample is indented (c). When the lipid can no longer sustain these loading forces, the tip breaks through the lipid (d), until it reaches the substrate (e). If tip approach continues beyond this point, the cantilever starts to deflect as the hard substrate cannot be indented.
The aforementioned set of events makes up the approach portion of the force curve. The tip is then withdrawn from the surface, and the remaining event it experiences, is some adhesion to the sample, before the tip pulls away and is again not in contact.

### 6.4 Observables and Calculations

From the approach and retract force curves, several useful parameters can be extracted: breakthrough forces, bilayer thickness (or penetration depth), Young’s modulus values, and adhesion forces. (Figure 6.1)

![Figure 6.1](image)

Figure 6.1 Force curves indicating quantities extracted: breakthrough force, indentation region, and adhesion force. Points of interest: A: contact point, B: onset of rupture/breakthrough, C: onset of substrate, D: maximum adhesion, E: sufficiently far from the surface with negligible adhesion effects.
Breakthrough force is the magnitude of the force at which the lipid membranes finally rupture (relative to the ‘zero’ force felt when very far away from contact). Indentation until rupture reveals the location of the substrate, and therefore allows the membrane thickness to be ascertained- it is the distance from the contact point to the substrate. (Figure 6.1 A-C)

Adhesion force can be calculated by taking the magnitude of the force (relative to non-contact state) from the retract data just prior to pull-off of the AFM tip from the surface. (Figure 6.1 D-E)

The elastic modulus is a mechanical property of the lipid layer that can be obtained by fitting the indentation region of the approach curve (Figure 6.1 A-B) to an appropriate mechanical model for tip-sample interaction. Several models were utilized and their suitability evaluated based on fit errors and comparison of calculated Young’s modulus to literature data. The one most appropriate model to describe the mechanics of this system was that of Sneddon’s, which considers a semi-infinite sample contacted by a paraboloidal-shaped tip:

\[
F = \frac{4E\sqrt{R}}{3(1-\nu^2)}\delta^{3/2}
\]

where \(E\) is Young’s modulus, \(\nu\) is Poisson’s ratio, \(R\) is the tip radius, and \(\delta\) is the indentation (separation distance re-zeroed at the contact point)\(^{24}\). The above equation can also be modified with conical-shaped tip geometry in the equation when sharper AFM tips are used in our experiments:

\[
F = \frac{2E\tan(\alpha)}{\pi(1-\nu^2)}\delta^2
\]

where \(\alpha\) is the tip’s semi-vertical angle.

These parameters arise from, or are affected by, specific and non-specific tip-sample interactions. And, although there is some interdependence (e.g. breakthrough force and modulus values are calculated from the same region of the approach curve), when combined, they provide a more complete physical description of the system.
6.5 Experimental

6.5.1 Lipid Bilayer Preparation

The preparation of lipid membranes supported on mica substrates utilized established vesicle fusion protocols, and is described in greater detail elsewhere\textsuperscript{18}. Briefly, lipid mixtures were obtained by combining ratios of the different lipid components- dioleoylphosphatidylcholine, egg sphingomyelin, cholesterol, and ceramide- using chloroform and methanol as solvents. The resulting solution was then exposed to a gentle stream of N\textsubscript{2} gas and placed under vacuum overnight to further remove the solvents. The lipid film was hydrated to a final lipid concentration of 0.5-1.0 mg/ mL prior to use. Small unilamellar vesicles were obtained by sonicating the lipid solution to clarity (~20-30 min) using a bath sonicator (Cole Parmer, Montreal, QC). Vesicle solutions were deposited on freshly cleaved mica substrates (20-30 μm thick) glued on glass coverslips affixed to an AFM fluid cell. The sample was incubated at 45 °C for about an hour and slowly brought to room temperature prior to AFM measurements.

6.5.2 AFM Imaging and Force Mapping

AFM experiments were performed primarily using the Nanowizard ® II BioAFM (JPK Instruments, Berlin, Germany) mounted on an Olympus 1X81 inverted confocal microscope, operating in contact mode. The Asylum MFP-3D® (Asylum Research, Santa Barbara, CA) and the Picoforce Scanning Probe Microscope (Veeco, CA) were also utilized. Silicon nitride cantilevers (DNP-S, Veeco, CA) were used in contact mode imaging and force mapping measurements unless stated otherwise. The spring constant, which was typically in the range of 0.15-0.28 N/m was determined by the thermal noise method\textsuperscript{25}. In force mapping, arrays of force distance curves were collected on bilayer samples with selected grid sizes (e.g., 64 × 64 pixels), and the 2D visual maps were reconstructed using custom analysis code to be described below. Applied loads within the range of 4-25 nN were typically used.
6.5.3 Batch Analysis of the Force Curves

The sets of force curves, ranging from approximately 1000-8000 curves comprising the force map were batch analyzed using a self-developed algorithm implemented in IGOR Pro 6 (Wavemetrics, Portland, OR). For each curve breakthrough force, Young’s modulus, bilayer thickness, and adhesion force were calculated.

Schematic 6.3. Program flow depicting (a) data import and intermediate calculations, smoothing, and initial threshold calculation; (b) data parsing with filter, critical point searching, and extraction and calculation of physical parameters; and (c) data output: force maps, the set of force curves, histogram, and scatter plots.
The program flow is depicted in Schematic 6.3. In the initial loading step (Schematic 6.3a), the set of force curves comprising the force map are read into the computer sequentially. This portion of the code was adapted to be able to process raw force map data from the three commercial instruments utilized. In addition to deflection force ($F$) versus distance ($z$) data, the spring constant $k$, and $(x,y)$ position of each are recorded. The extension and retraction portion of the curves are separated. From the spring constant and using Hooke’s Law, deflection distance $d$ can be calculated by

$$d = \frac{F}{k}$$

Equation 6.3

and the tip-sample separation $s$ can be calculated by

$$s = z - \frac{d}{c}$$

Equation 6.4

where $c$ is a correction factor that is related to instrument sensitivity, and can be obtained via calibration, and fine-tuned at a later stage to counter the effects of minor drift.

Each force curve is then analyzed, and an internal parameter is calculated so that the subsequent analysis would be dataset-specific. The force curves are box smoothed, using a box size that would be appropriate for the desired distance resolution. The force curve $F(z)$ is then differentiated to obtain $F'(z)$. The differentiated curve will show a sharp peak in the regions when the force-distance curve indicates a breakthrough event (Schematic 6.3a). The standard deviation of $F'(z)$ is calculated for the region where the tip is in the non-contact region. The magnitude of this peak is compared to the standard deviation of the baseline region to calculate a standard-deviation multiple. This value will be used for threshold-detection of breakthrough events by calculating the average for the entire batch of force curves, and adjusting this average to accept more or less curves. This addresses dataset-specific tuning.

In the next step (Schematic 6.3b), this dataset-averaged value is used as a threshold in searching for the $F'(z)$ peak again in each force curve. In this second pass, if the breakthrough peak of the specific force curve does not meet the threshold, it can be dropped from further analysis. Once the peak is found in $F'(z)$, three important reference points in $F(z)$, the contact point and the two points indicating the onset and end of rupture (i.e. the substrate), can be determined. Local
maxima and minima searches in $F(z)$ in this region can be initiated to look for the latter two points.

For the contact point, a search algorithm was implemented to facilitate the batch analysis process. The algorithm assumes that the breakthrough point has already been found, and searches points only prior to it. To speed up the analysis, only a portion of the curve was searched, restricted to a predefined distance from the breakthrough point. Each point in the selected portion is evaluated as a candidate contact point with the following algorithm: the candidate point is the junction of a piece-wise function; any points before the candidate contact point are considered the non-contact region, and are fit to a straight line; points beyond this point up to the breakthrough point belong to the indentation region and are also fit to an appropriate function. Although a 3/2 power fit in this region may have been more accurate, use of a second linear fit is much quicker, and given the evaluation scheme to be described, was not deemed necessary. The average mean square error of the proposed piecewise function and the experimental data is calculated and the candidate point that provided the lowest mean square error becomes the contact point to be used for subsequent calculations. To further prevent false identification, an added criterion to ensure the candidate point is sufficiently close in force value to the non-contact region was implemented.

Data which are physically-improbable (e.g. negative forces and negative separation distances) are filtered out and earmarked for manual inspection.

Data extracted from each force map can be presented in a number of ways as seen in Schematic 6.3c, and explained below.

6.5.4 Data Analysis and Visualization

The collection of data from a force map provides two types of information: spatially correlated values, and statistics. In order to visualize the former, two types of spatial maps were generated—contour plots and image plots—utilizing the $(x,y)$ position recorded for each force curve. Images of breakthrough force, indentation, and bilayer thickness along with adhesion as the $z$-scale were created to ascertain any correlation of clusters of data, with the spatial distribution of the phases of the lipid bilayer.
Secondly, the obtained values can also be analyzed statistically, since they represent many repetitions (albeit spatially-separated) of tip-indentation experiments. Histograms and scatter plots reveal modalities and correlations in the data, respectively. Figure 6.2 shows representative output showing both types, as generated by the analysis procedure: an AFM height image (a) and the corresponding force map of breakthrough forces (b) and Young’s Modulus values (c), an individual force curve (d), a histogram (e), and a scatter plot for a pair of parameters (f).

6.6 Results and Discussion

In the process of building up this force curve analysis tool, we have created a comprehensive package that facilitates AFM investigation of supported lipid membranes. Recently this automated process has been successfully utilized in elucidating the structure-mechanical property correlation of multi-component lipid bilayers\textsuperscript{16-19}. 
6.6.1 Self-assembly and Mechanical Properties of Phosphatidylcholine-
Sphingomyelin-Cholesterol Supported Lipid Bilayer Systems

By examining the intrinsic breakthrough forces in bilayers comprised of
dioleoylphosphatidylcholine, egg sphingomyelin, cholesterol (DEC), insights into the
organization of the different phases, were elucidated\(^\text{18}\). The DEC system phase separates into a
liquid-ordered (\(L_o\)) phase that is sphingomyelin and cholesterol-enriched, and a liquid-disordered
phase (\(L_d\)) that is composed of phosphatidylcholine\(^\text{26-29}\). And, with addition of ceramide (a highly
hydrophobic sphingolipid), a ternary ceramide-enriched domain\(^\text{12,30}\) is formed through the
displacement of cholesterol\(^\text{31-34}\). In the latter case, AFM imaging was able to distinguish the two
phases from height/ topography, with the shorter of the two phases assigned to the \(L_d\) phase, and
the intermediate one assigned to the \(L_o\) phase, and the tallest ones to the ceramide-enriched
domains- consistent with other studes\(^\text{35,36}\). However, the absolute height differences between the
three phases were found to very much depend on the setpoint used in imaging. Figure 6.3 shows
AFM (A- D) images and cross-sectional profiles (E-G) of DEC bilayers with ceramide on mica
using three setpoints: 0.5 nN setpoint (A, E) with lateral deflection image (B), 5.0 nN setpoint
(C,F), and with tapping mode (D, G). The height profiles in (E-G) clearly show three phases yet,
the height differences are markedly different, depending on instrument parameters used.
This result motivates the utilization of force curves and force maps to make conclusions about heights and properties of the phases of the lipid bilayers. By performing a force curve over $L_d$ and $L_o$ phases, a distance reference is established by identification of the contact point, from which indentation begins. Additionally, for this system, indentation is performed until lipid breakthrough, and thus the tip eventually meets with the substrate. Therefore, the substrate can also be used as a reference and bilayer thickness can be quantified from these two reference points as shown in Figure 6.1, A-C region. Figure 6.4 shows the overlay of more than 200 force curves recorded in both phases, which shows two distinct profiles of breakthrough events. It also confirms a height difference of ~0.8 nm between the $L_d$ (shorter) and $L_o$ (taller) phase, with literature data.\textsuperscript{28,35,37}
Force map analysis of a representative DEC bilayers is shown in Figure 6.5. The recreated maps showing breakthrough force (A), and elastic modulus (C) using Hertzian contact and the Sneddon model, differentiate fairly well the two phases in the bilayers. In (B) the corresponding histogram of the breakthrough force values from (A) depict a bimodal distribution with peaks at $F \sim 1.4 \text{ nN}$ corresponding to the fluid-disordered phase, and $F \sim 3.2 \text{ nN}$ corresponding to the fluid-ordered phase. A histogram of the Young’s modulus values maps (not shown) report $E \sim 80 \text{ MPa}$ and $E \sim 140 \text{ MPa}$ for the $L_d$ and $L_o$ phases, respectively. These magnitudes are in agreement with those reported in literature using electrocompression experiments$^{38,39}$. 

Furthermore, comparison of mechanical properties on DEC with addition of ceramide demonstrated the latter’s effect on organization by displacing cholesterol in creating ceramide-enriched domains$^{18}$. However, breakthrough events were largely unobserved over the ceramide region, which suggests a highly organized packing and great mechanical rigidity and so quantifying the mechanical stability of ceramide-enriched lipid bilayers became the objective of a follow-up study, also utilizing the force map batch analysis tool$^{19}$. In this study, several methods were utilized in weakening the ceramide-enriched domains, including methyl β-cyclodextrin and chloroform treatments. Since ceramide is known to expel cholesterol from the sphingolipid/cholesterol domains$^{40}$, methyl β-cyclodextrin was used on the DEC-ceramide bilayers to directly extract cholesterol. Also, chloroform vapor treatment is known to have a strong fluidizing effect on the membranes$^{41}$. The fact that a combination of both treatments were required to sufficiently weaken the ceramide-enriched domain (e.g. for tip-breakthrough to be possible) supports the hypothesis that the generation of ceramide in cells drastically alters the
structural organization of biological membranes, which leads to a greater nanomechanical stability and rigidity for the coexisting domains.

A third study looked at the effects of cholesterol by looking at force maps of a series of lipid membranes with systematic variation of cholesterol content\textsuperscript{17}. The morphology of DEC bilayers was shown to evolve with increasing cholesterol concentration- from small domains at low (5-10 mol\%) concentrations of cholesterol, to large elongated domains (25-35 mol\%), and finally to an inversion of the percolating phase (at \(\sim\)40 mol\%). In the same study, the effect of tip loading rate on rupture kinetics and activation energies was also investigated by varying velocity (loading rate) values, and using a model of bilayers rupture\textsuperscript{42} to determine activation energy (See Appendix Section 9.2 for details and derivation). Table 6.1 shows the calculated rupture activation energy, in both \(L_d\) and \(L_o\) phases, as a consequence of cholesterol content variation, and force curve collection at different velocities.

<table>
<thead>
<tr>
<th>Cholesterol Concentration</th>
<th>(L_d) phase</th>
<th>(L_o) phase</th>
</tr>
</thead>
<tbody>
<tr>
<td>10%</td>
<td>73 ± 50</td>
<td>78 ± 123</td>
</tr>
<tr>
<td>15%</td>
<td>123 ± 12</td>
<td>93 ± 8</td>
</tr>
<tr>
<td>20%</td>
<td>114 ± 23</td>
<td>105 ± 83</td>
</tr>
<tr>
<td>25%</td>
<td>73 ± 36</td>
<td>75 ± 28</td>
</tr>
<tr>
<td>30%</td>
<td>76 ± 18</td>
<td>85 ± 21</td>
</tr>
<tr>
<td>35%</td>
<td>108 ± 15</td>
<td>109 ± 45</td>
</tr>
<tr>
<td>40%</td>
<td>85 ± 25</td>
<td>98 ± 74</td>
</tr>
</tbody>
</table>

Table 6.1 Rupture activation energies of the coexisting phases (liquid ordered domains \(L_o\), and fluid disordered phase \(L_d\), in DOPC/SM (1:1) with 10-40\% cholesterol bilayers. (Adapted from \textsuperscript{17})

The data did not show dependence in the activation energy for the different lipid phases, nor was there an obvious correlation between \(\Delta E_a\) and cholesterol content. This result highlights the complex nature of the rupture process in bilayers of this ternary mixture\textsuperscript{43}. Nevertheless it was in agreement with activation energies of membrane fusion and other cellular processes using different techniques\textsuperscript{44-48}.
The use of force mapping and the batch analysis was also further utilized in phase segregation of the same DEC lipid bilayers deposited on a gold substrate\textsuperscript{16}. The motivation for attempting the self-assembly on gold was in the potential for making this system accessible to surface-plasmon resonance techniques\textsuperscript{49-51} which also opens up the possibility for biosensing applications\textsuperscript{52,53}. Hertzian contact mechanics were utilized in the elastic modulus fits, however Van der Waals forces, surface charge density differences, and finite sample thickness limitations\textsuperscript{54} (see Section 2.2.2.2) led to markedly different results in comparison to bilayer-on-mica systems.

### 6.6.2 Force Maps and Batch Analysis Performance

Batch analysis is extremely useful in studies exploring the effects of multiple variables. For example, the variation of two parameters (cholesterol, and loading rate) in the previously mentioned work\textsuperscript{17} (Section 6.6.1), forms a matrix of experiment conditions, and necessitated the collection and analysis of over 30 force maps. The utilization of an automated analysis procedure facilitated this process primarily by ensuring it was accomplished within a realistic timeframe.

<table>
<thead>
<tr>
<th>FORCE MAP PARAMETERS</th>
<th>4096</th>
<th>4096</th>
<th>4096</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Force Curves</td>
<td>4096</td>
<td>4096</td>
<td>4096</td>
</tr>
<tr>
<td>Collection Frequency</td>
<td>10 kHz</td>
<td>10 kHz</td>
<td>10 kHz</td>
</tr>
<tr>
<td>Force Distance</td>
<td>~100 nm</td>
<td>~100 nm</td>
<td>~100 nm</td>
</tr>
<tr>
<td>Tip Velocity</td>
<td>2000 nm/s</td>
<td>800 nm/s</td>
<td>200 nm/s</td>
</tr>
<tr>
<td>Points per Force Curve</td>
<td>~1000 points</td>
<td>~2500 points</td>
<td>~10000 points</td>
</tr>
<tr>
<td>Total Collection Time</td>
<td>~25 min</td>
<td>~45 min</td>
<td>~60 min</td>
</tr>
</tbody>
</table>

Table 6.2: Data analysis run times for 3 force maps with very similar data collection parameters, except for data points per force curve.
Force maps have an advantage of direct correlation of topography (morphology) to mechanical properties, which is the reason this AFM technique is important to membrane research. However, due to lengthy manual analysis, one often had to compromise by reducing the resolution of the force map. Extracting contact point, breakthrough force, and indentation region, and then subsequently utilizing them in a series of calculations (elastic modulus, breakthrough force, adhesion force, bilayer thickness), would have made the manual route very tedious, and was the original motivation for designing the program. Table 6.2 is a comparison of the analysis times of three force volume maps with nearly identical collection parameters. The primary difference between the three force maps is the tip velocity used, which effectively determined the number of points collected per force curve since collection frequency was fixed and force curve distance was set to ~100 nm. The datasets were analyzed on a mid-end desktop computer (2.8 GHz Intel® Core™ 2 Quad processor computer with 4GB RAM) and as the table shows, took between 1 and 6 hours to finish, depending on the force curve data point density. At the slowest time and with 4096 curves, this translates to approximately 5 seconds per force curve, even in light of the fact that the data was largely serially processed and no parallel processing algorithms were explicitly utilized. The software, IGOR Pro, does have a number of built-in, parallel-optimization routines in its curve-fitting functions for multi-core processors. However, their effect may be minimal as these were only applied during the elastic modulus calculation step, which consistently took less than three minutes for each of the three datasets—a very small fraction of the total processing time. Because the three datasets exhibited the same fitting times, the curve-fitting portion seems to be dependent only on the size of the dataset (4096) and not on data point density.
A second advantage is in the use of an automated contact point finder. This part of the software is very crucial, as many subsequent calculations rely on the correct location of the contact point reference. Manual determination of the contact point is usually performed by visual inspection, and is a source of reading error and inconsistency\textsuperscript{55,56}. In this program, the location of contact point is quantified by evaluating mean square error of candidate points, with allowance for manual correction if needed.

In designing the code, we also understood that not all force curves were relevant, and had to allow for rejection of some data. One feature that allows adjustment of data acceptance is the average threshold multiplier used in finding the breakthrough force peak in the first derivative of the force curve. This value is calculated per force map and attempts to minimize false positive peaks from significant signal noise, and also missed peaks due to gradual breakthroughs (i.e. non-sharp ruptures). The latter cases would have been viewed as regions that are too stiff to be ruptured, which is not necessarily the correct conclusion if the breakthroughs were just missed. The adjustment of the average threshold multiplier and other factors however, is somewhat empirical in nature because some physics is circumvented with mathematics, and vice versa.

In the scope of supported lipid membrane experiments, the code lends itself well to utilization of different physical or mathematical models to explain the mechanics. Therefore, its adaptation to other systems- e.g., other thin films, membranes, surfactants-which require force-curve indentation analysis, is straightforward. Moreover, the use of the peak-finding algorithm is also extensible to pulling experiments such as single-molecule pulling studies\textsuperscript{57-60}, where in the case of AFM experiments, it is the retract portion of the force curve that reveals the physics. There has been some work already in the creation of automated algorithms for the analysis of single-molecule pulling\textsuperscript{61} and receptor-ligand binding\textsuperscript{62,63} experiments.

### 6.7 Conclusions

The creation of an automated procedure in the study of lipid membrane systems was motivated by the rich information available from AFM force curves and force maps, whose extraction required a lengthy and tedious process if performed manually. The automated process was able to reduce the total force map processing time, due to heavy reliance on computing power to
perform unattended sequential analysis. This therefore enabled the execution of force map experiments requiring a large number of permutations of experimental conditions. The resulting program was successfully used to analyze the indentation and subsequent rupture of supported lipid membranes via AFM tip to extract relevant information that was used to demonstrate structure and mechanics correlation in these systems.

6.8 Contribution

The majority of the author’s contribution to this body of work is the creation of the automated batched analysis code for the extraction of mechanical parameters, and spatial correlations that can be ascertained from force volume mapping (majority of work with Sullan et al.\textsuperscript{17-19,23} and small contribution to Ip et al.\textsuperscript{16}). The author was also involved in the data analysis, especially as pertaining to the interpretation and use of mechanical models.

6.9 References

(37) Ira; Johnston, L. J. Biochimica et Biophysica Acta - Biomembranes **2008**, 1778, 185.
(51) Tawa, K.; Morikagi, K. *Biophys J* 2005, 89, 2750.
7 Microgels with Interpenetrating Polymer Network as Cell-Mimic

7.1 Permissions


7.2 Abstract

This work evaluates microfluidically generated interpenetrating polymer network (IPN) microgels as a mimic to neutrophil granulocytes (neutrophils). The microgels were generated from ionically cross-linked alginate and chemically-gelled poly(N-isopropylacrylamide). The mechanical properties of the microgels such as Young’s modulus and characteristic relaxation time were studied under atomic force microscope (AFM) indentation experiments for microgels with varying compositions. The lower limits of the elasticity found in this study is within the same range of the elasticity reported in the literature for neutrophils, and thus make these microgels a promising model system for cell studies.

7.3 Introduction

The movement of white blood cells such as neutrophil granulocytes (neutrophils) through the cardiovascular system is integral to the operation of the immune system. It is known that neutrophils traversing pulmonary capillaries often encounter natural constrictions which require them to undergo substantial deformation to pass through\(^1\),\(^2\). However with plaque buildup, blood vessel pathways constrict further (atherosclerosis). This leads to decreased blood flow and ultimately oxygen deprivation if vessel is an artery, and gives rise to other serious diseases and complications due to blood clots or emboli.

These observations therefore highlight the importance in understanding how cells such as neutrophils flow, and in particular, the mechanical properties that enable them to traverse
through constrictions. Due to the inherent challenges in studying real cells, model systems of artificial cells or cell-mimics are instead utilized\(^3\)-\(^7\). The first challenge therefore, is to be able to make a cell-mimic of the appropriate size, surface, and mechanical properties\(^8\).

Here, microfluidically synthesized microgels with a double interpenetrating polymer network structure (IPN)\(^9\),\(^10\) containing Ca\(^{2+}\) cross-linked alginate and poly(N-isopropylacrylamide) (PNIPAm) will be used as the model cell. The mechanical properties will be tuned by variation of the relative quantities of polymers and cross-linking component comprising the microgel. These microgels will then be characterized by a number of techniques, along with AFM indentation measurements to extract mechanical properties.

### 7.4 Materials and Methods

#### 7.4.1 Materials and Microgel Synthesis

The synthesis of microgels containing two interpenetrating polymer networks: sodium alginate (cross-linked with Ca\(^{2+}\) ions) and PNIPAm) is described in greater detail in the original manuscript\(^11\). Briefly, microfluidic generation of precursor droplets of an aqueous solution containing a mixture of sodium alginate, NIPAm, ammonium persulfate (NIPAm initiator), and N,N’-methylenebisacrylamide (BIS: a NIPAm cross-linker) is followed by a gelation step when Ca\(^{2+}\) ions, and N,N,N’N’-tetramethylethylenediamine (TEMED) are introduced to cross-link the alginate, and polymerize NIPAm, respectively\(^12\)-\(^15\).

#### 7.4.2 Mechanical Characterization of Microgels

The Young’s modulus of the microgels was determined in AFM indentation experiments (Schematic 7.1) under fluid using the MFP-3D (Asylum Research, Santa Barbara, CA) equipped with a 100 μm-wide tipless cantilever (Nano World, U.S.A.). The spring constant, \((k \sim 0.03 \text{ N/m})\), was determined by the thermal noise method\(^16\). To mitigate the problem of microgels slipping laterally while undergoing indentation during the measurement, gold-coated substrates (200 nm thick Au on glass with Cr-adhesion layer, 10 mm × 10 mm) (Ssens B.V., Hengelo, The Netherlands) were utilized and suitably functionalized with a SAM of mercaptoethylamine\(^17\). Due to the presence of deprotonated carboxylic groups in alginate, the microgels carried a
negative charge and were attracted to the positively charged substrate. The substrates were prepared as follows: gold substrates were cleaned with piranha solution (1:3 of 30% H₂O₂/98% H₂SO₄) for 5 minutes, rinsed in excess deionized water (> 18 MΩ·cm) and immersed in a solution of 1.0 mM mercaptoethylamine in ethanol for 2 hours. Afterwards, samples were again washed with deionized water and dried under a stream of N₂ gas. Caution! Piranha solution is a very strong oxidant and is extremely dangerous to work with; gloves, goggles, and a face shield should be worn.

Schematic 7.1 Schematic of tipless cantilever in contact with microgel on Au substrate

All AFM measurements were carried out in phosphate buffer solution (PBS). For each microgel composition (the “sets” as found in Table 7.1), approximately five individual microgel particles were examined and up to 10 consecutive force curves collected for each. To fit the elastic modulus, the indentation region of the approach portion of the force-distance curve was fit to Equation 7.1.18:

\[
\delta = (3\pi)^{\frac{3}{2}} F^\frac{3}{2} \left(\frac{1 - \nu^2}{E}\right)^{\frac{3}{2}} \left(\frac{1}{2R}\right)^{\frac{3}{2}}
\]

Equation 7.1

where \(\delta\) is the indentation distance, \(F\) is the force, \(\nu\) is the Poisson’s ratio (assumed to be 0.5), \(E\) is the Young’s modulus, and \(R\) is the microgel radius.
The relaxation process of the microgels was studied in force-time measurements. Figure 7.1 illustrates a typical force-time curve acquired in such experiments. First, a loading force was applied to compress a microgel particle until a setpoint of 1.5 – 2 nN was reached (the Approach region). This segment of the force curve is used in the indentation analysis mentioned above by instead looking at the distance data channel. Then, the driving piezo was halted for 1-2 s (the Dwell region), and during this time, a decay of the force $F$, occurs, due to the relaxation of the stress on the microgel particle. After, the dwell, the cantilever was retracted from the microgel (the Retraction step).

![Figure 7.1 Typical force-time curve acquired for microgel particle showing the approach (blue), dwell (black) and retract (red) portions.](image)

The dwell portion of the force vs. time curves were fit to the Kohlrausch-Williams-Watts function (stretched exponential function)$^{19}$:

$$F(t) = F_r \exp \left[ -\left( \frac{t}{\tau} \right)^\beta \right] + F_c$$  

Equation 7.2
where $F_r$ is the amplitude of the relaxation force, $F_\infty$ is the force after a length of time much longer than the mean relaxation time $\tau$ and $\beta$ is the stretching exponent which is a measure of distribution of the relaxation times.

### 7.4.3 Batch Analysis

A simplified version of the batch analysis code from Section 6.5.3 and Appendix 9.1 was used in this chapter and also programmed with IGOR Pro (Wavemetrics, Portland OR). Adjustments include the addition of relevant fitting models, removal of force map-specific functions (coordinates, force map recreation), and addition of the time variable and time-dependent model fits. Other functions, including file loading, automatic contact point detection and output, were used as-is.

### 7.4.4 Characterization of Microgels: Other Methods

Other methods for the characterization of microgels were: optical microscopy for size determination, confocal fluorescent microscopy to determine internal structure, and measurement of swelling ratio when dry microgels are immersed in aqueous solution. Swelling ratio ($Q$) follows the simple formula:

$$Q = \frac{m_h}{m_d} \times 100\%$$  

Equation 7.3

where $m_h$ and $m_d$ correspond to the mass of the hydrated and dry microgel, respectively.

### 7.5 Results and Discussion

#### 7.5.1 Variation of Alginate Content

The relative content of polymers can be varied to determine the effect of overall microgel composition on mechanical behavior. While all three (sodium alginate, NIPAm, and cross-linker (BIS)) were varied, only the effects of sodium alginate variation will be discussed here. The compositions studied are outlined in Table 7.1 with set B, D, and E containing the alginate variations.
Table 7.1 Compositional variation of IPN microgels (Adapted from 11)

<table>
<thead>
<tr>
<th>Set</th>
<th>Sodium Alginate (wt. %)</th>
<th>NIPAm (wt. %)</th>
<th>BIS (wt. %)</th>
<th>Diameter (μm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1.0</td>
<td>1.0</td>
<td>10.0</td>
<td>103 ± 3</td>
</tr>
<tr>
<td>B</td>
<td>1.0</td>
<td>4.0</td>
<td>10.0</td>
<td>89 ± 6</td>
</tr>
<tr>
<td>C</td>
<td>1.0</td>
<td>4.0</td>
<td>1.0</td>
<td>99 ± 10</td>
</tr>
<tr>
<td>D</td>
<td>0.5</td>
<td>4.0</td>
<td>10.0</td>
<td>138 ± 8.6</td>
</tr>
<tr>
<td>E</td>
<td>1.5</td>
<td>4.0</td>
<td>10.0</td>
<td>98 ± 4.6</td>
</tr>
<tr>
<td>F</td>
<td>2.0</td>
<td>4.0</td>
<td>10.0</td>
<td>103 ± 5.5</td>
</tr>
</tbody>
</table>

7.5.2 Effect on Swelling Ratio

Since sodium alginate is the component of the microgel that is strongly cross-linked with Ca$^{2+}$, then it is expected that an increase in sodium alginate weight % will tend to restrict the ability of the microgel to swell when immersed in buffer. This trend is readily seen in Figure 7.2 where a swelling ratio of 2700% is observed at 0.5 wt. % sodium alginate, decreasing to ~2400% at 2 wt. %.
7.5.3 Elastic Modulus and Relaxation Measurements

The mechanical properties of the microgels are expressed in terms of their Young’s modulus and characteristic relaxation time respectively. The results of AFM indentation experiments are shown in Figure 7.3.
With increasing alginate concentration, the elastic modulus increased from 1.86 kPa at 0.5 wt. % alginate to 29.0 kPa at 2 wt. %. It is noteworthy that the elastic modulus at 0.5 wt. % is within the range of modulus values for neutrophils, as reported in literature, and the broad range of Young’s modulus values obtained compare well with moduli of living cells (up to 100 kPa). The moduli were found using Equation 7.1, for which was the only free-fit parameter. and were measured by AFM, and Poisson’s ratio was assumed equal to 0.5, which is typical of polymeric materials, and microgel diameter was approximately equal to 100 µm and independently verifiable by optical microscope.

Over the same range of concentrations, average relaxation times were reduced from 305 to 16 milliseconds (Figure 7.3B). The trend persists even considering the large standard deviations in the data. In equation Equation 7.2, , , , and are free-fit parameters. The stretching
exponent $\beta$ is a measure of distribution of the relaxation times, and its best fit values remained in the range of 0.4 to 0.6. This weak power law dependence is in agreement with trends observed for other biological systems, and indicates that these IPN microgels have cell-like relaxation behavior.

Subsequently, the microgels were also flowed through a microfluidic polydimethylsiloxane (PDMS) channel with a constricted segment in a demonstration of an artificial neutrophil traversing a constricted a pulmonary capillary segment$^{11}$.

7.5.4 Evaluation of AFM Indentation as Testing Platform

The tipless cantilever is a source of uncertainty due to the fact that its alignment with respect to the microgel was performed manually with the aid of optical microscope in the AFM instrument. However for each microgel set, the deviation did not exceed 20%, of the average Young’s modulus value, suggesting that the effect of misalignment was insignificant. Microgel contrast in PBS buffer was also an issue during AFM measurements, and was one of the reasons for using a metallic substrate instead of mica. Although, the surface modification of gold using alkanethiol assembly was the more important reason for substrate choice.

Equation 7.1 was used primarily to match the geometry of the cantilever-microgel interaction, which can be sufficiently described as the contact of a planar and spherical body. However, the 2/3 power dependence of distance on force (conversely the 3/2 power dependence of force on distance) indicates Hertzian contact assumptions (see Section 2.2.1). However, the presence of large adhesion (Section 2.2.2.1) forces as seen in Figure 7.1, and the tendency of lateral translation of the microgel during indentation, suggests caution in interpreting the modulus fits from a purely elastic contact model. Lastly, the measurement of shear modulus $G$ is a very important factor missing in the evaluating of microgels as neutrophil-mimics, since neutrophils traversing a capillary must experience shear stress during contact with capillary walls. Perhaps the microfluidic PDMS channels (along with video-assisted flow analysis) can be used to address this mechanical parameter.
7.6 Conclusion

Microfluidically produced microgels were evaluated for their mechanical similarity to cells, in particular to neutrophils, which undergo significant deformations when traversing capillaries. Microgels with varying sodium alginate content were characterized by their Young's modulus and relaxation time. The trend towards greater stiffness and faster relaxation times was seen with increasing alginate content which is attributed to the greater degree of cross-linking. The AFM, with its ability to perform indentation experiments, can be used as a platform for evaluating the mechanical behavior of cell and cell-mimics, bearing in mind that models and techniques presently used may need modification.

In its early development, microfluidics was used in advancing the state-of-the-art in inkjet printing, a wholly technological development of the 20th century. However, in a rather appropriate example of art imitating life, it is now being applied to understanding the nature of blood flow.

7.7 Contribution

Gold substrate preparation, AFM indentation measurements of the microgels, and data analysis including batched extraction of mechanical parameters, were performed by the author.

7.8 References


(20) Concentrations of reagents in the table are given for the droplet phase feed solutions.

8 Outlook

The AFM was instrumental in investigating self-assembled systems, mostly at the nanoscale, using force as a stimulus and as a measured response. The implications of these studies are many:

1. The mechanical restructuring of self-assembled monolayers played an important role in its electrical behavior. Design of nanocontacts for electrical applications therefore must take into consideration pressure-induced effects.
2. The demonstration of differential conductivity in PS-b-PFS thin films presents a configuration of planar, spatially-localized, vertical electrical elements which may have possible device applications.
3. Force map analyses of supported lipid bilayers demonstrate the important role of lateral heterogeneity, and mechanical stability and rigidity to the function of membranes.
4. The evaluation of mechanically tunable alginate- PNIPAm IPN microgels demonstrates their potential use as model cell-mimics due to similarity to cells.

The use of the AFM to measure the molecular packing of SAM monolayers is ideal because of its ability to apply a force stimulus and measure the force response. This was used in conjunction with a conducting probe to utilize conduction behavior as an indicator of mechanical restructuring by monitoring the change in tunneling behavior through straight alkane chains (Chapter 4). This work can be extended by looking at systems that have different conduction mechanisms. For example, recently Salmeron and coworkers have investigated the mechanical and electrical properties of aromatic molecules with a conducting AFM. In particular, they were comparing \(\pi-\pi\) stacking strengths by looking at several model systems in the same manner that different alkane chain lengths were used to test tunneling behavior. It is interesting to note that distinct conductivity changes were also witnessed for some of their films when placed under compressive loads\(^1\).

Since the electrical resistance versus force behavior of initially close-packed films such as a SAM of pure C\(_{10}\)SH, differs significantly from that of one that is initially disordered, such as a SAM of pure C\(_6\)SH, then this characteristic may be used to distinguish phases in mixed monolayer (phase segregated) systems. In studying these systems, force map analysis could be
utilized to collect electrical resistance values at a fixed force setpoint to create a resistance map. This method may also be used with lateral friction measurements.

Quasi-static resistance-force plots (resistance was determined from the force curve at a fixed force setpoint, for many force setpoints), were collected to mitigate the unpredictable current response over the span of a single force curve primarily due to the changing (and unknown) number of contacts with the AFM tip. With careful control of the tip-sample interaction, or possibly going towards single molecule experiments, the dynamic electrical response may be investigated.

Lastly the electrical behavior of molecules may also be studied under tension. Pulling on several strands of molecules may for example reduce the chain-to-chain distance, and for conjugated molecules, increase the $\pi-\pi$ stacking strength. In this case, the relevant conduction model can be combined with the worm-like chain model to look at stretched single or bundles of molecules (Schematic 8.1).

Schematic 8.1 Conduction through molecules under tension.

In Chapter 5, we have shown the use of AFM to distinguish phases in phase-separated diblock copolymer thin films by using differential conductivity of the blocks in support of topography. And, as was shown, the current images produced were highly resolved (more so than
topography), likely due to conduction through tip asperities. This increased lateral resolution, combined with vertical conduction, may be able to provide insights into the interface between the two blocks in cylindrical, or other amenable self-assembled diblock morphologies. In particular, the distribution of the two blocks in that interface— in other words the gradient of the transition— can be investigated in diblock copolymers with appropriately selected blocks. Related work has been performed before by Winnik and coworkers using ruthenium tetroxide staining\(^2\).

Related to this direction, the distribution of incorporated elements (e.g. nanoparticles, quantum dots) in a diblock copolymer thin film\(^3\)\(^-\)\(^5\) can be studied in this manner, as long as each block and the nanoparticles differ in electrical response, and, there is a conduction pathway from the film surface to the substrate. Furthermore, with electrical force microscopy or Kelvin probe AFM, an insulating substrate can be utilized since the method of detection does not rely on conduction from tip to substrate (See Section 2.3.5).

The PFS cylinders in a PS-\(b\)-PFS diblock thin film have the potential to serve as laterally organized, vertical resistive elements in nanoscale electronics. With further engineering of the diblock copolymer and possibly the incorporation of other layers and materials, nanoscale analogues to common electrical components can be demonstrated: diodes, capacitors, electromagnetic metamaterials. For example, to build a device that can store charge in the cylindrical regions, one can insert a thin insulating layer between substrate and polymer film, chosen so that electron or hole tunneling through this layer is sufficiently slow to allow for timely writing/ reading of information from the cylindrical array (Schematic 8.2).

Schematic 8.2 Charge storage array using diblock copolymer with intermediate tunneling layer.
In the study of lipid bilayers (Chapter 6), force map analysis was shown to be a useful technique in extracting phase-dependent mechanical properties as a measure of rigidity. With this platform, the controlled study of lipid membranes can be carried out to determine the influence or roles of different biomolecules (e.g. trans-membrane proteins) to their stability and phase-segregation. Furthermore, with suitable functionalization of the tip (e.g. ligand protein), then re-enactments of biologically-relevant events/interactions can be thoroughly investigated.

One of the most important contributions in this thesis is the development of the force map analysis toolset for the batched, high-through-put processing of AFM force curves, whose utility has been demonstrated in the study of supported bilayer films. This toolset will be valuable to those investigating the adhesive interactions between molecules and surfaces as well as molecular assemblies on surfaces. This toolset enables the acquisition and subsequent statistical analysis of large data sets, which is crucial for thorough interpretation of the physico-chemical phenomena accompanying mechanically-induced contact and rupture.

Regarding aspects of force map analysis, any improvements to contact point or membrane rupture detection are welcome, especially if user-intervention can be minimized. Some possibilities include fuzzy logic\textsuperscript{6,7} and Bayesian analysis\textsuperscript{8} which have been used in force curve analysis in compressive and single molecule spectroscopy (pulling) experiments, respectively.

One improvement in the investigation of microgels with AFM (Chapter 7) would be to control the cantilever's lateral position to perform indentations at a fixed position. The combination of microindentation experiments, force maps, and a geometrical/mechanical model to account for non-axisymmetric deformations for spherical objects may be able to statistically extract the elastic modulus regardless of indentation spot. Alternatively, nanoindentations performed with a tipped cantilever can extract lateral heterogeneities across the microgel surface\textsuperscript{9,10}.

Additionally, for some of these systems, the mechanical deformations may be more suitably described by non-Hertzian mechanics, especially to factor in adhesion effects\textsuperscript{11}. One approach is to utilize the retract portion of the force curve to extract the mechanical properties\textsuperscript{12}, since the pull-off force due to adhesion can be measured and quantified.
The varied collection of systems in this thesis is a direct reflection of the many applications of self-assembly. It is the author’s hope that this body of work has contributed towards understanding and creation at the nanoscale.

References


9 Appendices

9.1 Appendix: Programmed Code Used in Thesis Work for Analysis

9.1.1 Main Functions

The functions below are the primary functions used to analyze each force curve. They include the pre-parsing of force curves to calculate a dataset average, parsing of the force curves again to compute mechanical properties. These functions call on graphing, fitting, and helper functions. (See Section 6.5.3 for details on overall algorithm)

9.1.1.1 IGOR Pro Code

```c
#pragma rtGlobals=1  // Use modern global access method.
#include <KBColorizeTraces>

Function BreakthruPointsBreakdown (namesAndPaths,numsAndPaths)
    WAVE/T namesandpaths
    WAVE numsandPaths
    WAVE constants
    variable index,outer_index=0
    variable dimension_orig,dimension
    variable remainder,quotient
    variable batches=1200
    variable factor=constants[3] // what percent to look into for cut Off
    variable timerRefNum
    variable elapsedTime=0
    WAVE statistics
    statistics[1]=0
    statistics[2]=0
    dimension_orig=dimsizes(namesAndPaths,0)
    Make/T/O/N=(0,15) subNamesAndPaths
    Make/O/N=(0,15) subNumsAndPaths
    for (index=0;index<dimsizes(namesandpaths,0);index+=1)
        namesandpaths[index][13]="orig row index"
        numsandpaths[index][13]=index
    endfor
    duplicate /t/o namesandpaths, subnamesandpaths
    duplicate /o numsandpaths, subnumsandpaths
    do
        print "Iteration ",outer_index+1
        dimension=dimsizes(subnamesAndPaths,0)
        if (100*dimension/dimension_orig>0.5) //if there are more than 5% left to re-process
            remainder=mod(dimension,batches)
            quotient=floor(dimension/batches)
            timerRefNum=elapsedTime+quotient
            elapsedTime=timerRefNum
            fprintf("Print was made on row ",dimension,\n
```
print "Multiplier pre-parsing..."
for (index=0;index<quotient+1;index+=1)
    timerRefNum=startMSTimer
    if (index==quotient)
        if (remainder>0)
            print "Analyzing waves from ",index*batches+1
            print "to ",index*batches+remainder, "of ",dimension
            preParser(subnamesAndPaths,subnumsAndPaths,
                index*batches,index*batches+remainder,batches)
        endif
    else
        print "Analyzing waves from ",index*batches+1
        print " to ",index*batches+batches, "of ",dimension
        PreParser(subnamesAndPaths,subnumsAndPaths,
            index*batches,index*batches+batches,batches)
    endif
    elapsedTime+=stopMSTimer(timerRefNum)/60e6
    print "Parsing, time elapsed so far: ",floor(elapsedTime/60), "hours "
    print mod(round(elapsedTime),60), "minutes."
endfor
imagestats /M=1/G={0,dimension-1,10,10} subnumsAndPaths
variable multiplier=V_avg
if(multiplier<(10/factor))
    print "Multiplier close to 10: ",multiplier
    multiplier*=factor
elseif (multiplier>(50/factor))
    print "Multiplier over 50: ", multiplier
    multiplier=50
else
    print "Multiplier 10<x<50: ", multiplier
    multiplier*=factor
endif
print "Parsing everything else...
for (index=0;index<quotient+1;index+=1)
    timerRefNum=startMSTimer
    if (index==quotient)
        if (remainder>0)
            print "Analyzing waves from ",index*batches+1
            print " to ",index*batches+remainder, "of ",dimension
            Parser(subnamesAndPaths,subnumsAndPaths,
                index*batches,index*batches+remainder,multiplier,batches)
        endif
    else
        print "Analyzing waves from ",index*batches+1
        print " to ",index*batches+batches, "of ",dimension
        Parser(subnamesAndPaths,subnumsAndPaths,
            index*batches,index*batches+batches,multiplier,batches)
    endif
    elapsedTime+=round(stopMSTimer(timerRefNum)/60e6)
    print "Parsing, time elapsed so far: ",floor(elapsedTime/60), "hours "
    print mod(round(elapsedTime),60), "minutes."
endfor
print "Copying stuff over to master lists"
for (index=0;index<dimsize(subnumsandpaths,0);index+=1)
namesandpaths[subnumsandpaths[index][13]]=subnamesandpaths[index][q]
numsandpaths[subnumsandpaths[index][13]]=subnumsandpaths[index][q]
endfor
print "Paring down sub list from ",dimsize(subnumsandpaths,0)
for (index=dimsize(subnumsandpaths,0)-1;index>=0;index-=1)
if (cmpstr(subnamesAndPaths[index][3],"Indent_none")!=0)
    DeletePoints index, 1, subNumsAndPaths
    DeletePoints index, 1, subNamesAndPaths
endif
endfor
print dimsize(subnumsandpaths,0), "left to re-process"
if (outer_index==0)
    duplicate/o subnamesandpaths firstPassNaPs
endif
else
    print "Too few leftovers for re-processing"
    break
endif
outer_index+=1
while (outer_index<constants[5]) //# of iterations to carry out
    timerRefNum=startMSTimer
    rebuildQuantities()
    elapsedTime+=round(stopMSTimer(timerRefNum)/60e6)
    print "Total parsing time: ",floor(elapsedTime/60), "hours ", mod(round(elapsedTime),60)
print "minutes."
End // Function BreakthruPointsBreakdown

Function preParser(namesAndPaths,numsAndPaths,startPos,endPos,batches)
    WAVE/T namesAndPaths //local access to namesAndPaths wave, again
    WAVE numsAndPaths
    variable startPos,endPos,batches
    string localWavePath
    variable index,flag,index2
    variable minPixel
    WAVE constants
    for (index=startPos;index<endPos;index+=1)   //cycle through each name in namesAndPaths list.
        localWavePath=namesAndPaths[index][0];  // string-ize current namesAndPaths entry
        loadwave /o/q/p=FC localWavePath+"extF"
        loadwave /o/q/p=FC localWavePath+"extZ"
        loadwave /o/q/p=FC localWavePath+"extS"
        WAVE localWaveForce=$(localWavePath+"F_smth_box") // make a wave placeholder for box smoothing, and smoothed, differentiated wave.
        Duplicate /O localWaveForce $(localWavePath+"F_smth_boxDIF")
        WAVE ForceBoxSDVRoll=$(localWavePath+"F_smth_sdvRoll")
        variable namesAndPathsDim=DimSize(namesAndPaths,0)
        variable dimension=DimSize(localWaveForce,0)  // dimension of the present wave.
        Duplicate /O localWaveForce $(localWavePath+"F_smth_box")
        Duplicate /O localWaveForce $(localWavePath+"F_smth_boxDIF")
        WAVE ForceBoxSmthDIF=$(localWavePath+"F_smth_boxDIF")
        Duplicate /O localWaveForce $(localWavePath+"F_smth_sdvRoll")
        WAVE ForceBoxSDVRoll=$(localWavePath+"F_smth_sdvRoll")
minPixel=PixelPerDist(localWaveSep,constants[7])
// smooth using a box_width of 0.5% of the number of data points.

// possible hard-coded tweak region
if (minPixel>1)
  Smooth/B minPixel, ForceBoxSmth
endif

wavestats/q/z ForceBoxSmth //calculate wave stats.
variable avg_sdev=V_sdev //get out the average standard deviation.
index2=0
flag=0
Differentiate ForceBoxSmth/X=localWaveZed/D=$(localWavePath+"F_smth_boxDIF")
//differentiate the smoothed Force
// the point of this section is to find roughly the baseline of non contact-ness
do
  wavestats/q/z/R=[0,index2] ForceBoxSmth
  // calculate a rolling stdDev, to calculate percent of rolled sdev to avg_sdev.
  ForceBoxSDVRoll[index2]=(V_sdev)/avg_sdev
  if (ForceBoxSDVRoll[index2]>0.03&&abs((100*index2/DimSize(ForceBoxSDVRoll,0)>5))
    //if the rollingSDV value suddenly spikes to 3%, and a spike has not been previously
    // detected, and the data is at least 5% of x-width
    // possible tweak value
    numsAndPaths[index][7]=index2
    namesAndPaths[index][7]="noncontact index"
    flag=1
  enddo
endwhile (index2<DIMENSION&&!flag)

End // Function preParser(namesAndPaths,numsAndPaths,startPos,endPos,batches)

Function Parser(namesAndPaths,numsAndPaths,startPos,endPos,multiplier,batches)
  WAVE/T namesAndPaths //local access to namesAndPaths wave, again
  WAVE numsAndPaths
variable startPos, endPos, multiplier, batches
WAVE statistics
string localWavePath
WAVE/T subnamesandPaths
WAVE subnumsandPaths
WAVE constants
variable index, box_width, flag, cutoff, dimension, index2, numOfPeaks, index3
variable namesAndPathsDim = DimSize(namesAndPaths, 0)
variable minPixel

for (index=startPos; index<endPos; index+=1)  // cycle through each name in namesAndPaths list.
  flag = 0;
  cutoff = numsAndPaths[index][9] - multiplier * numsAndPaths[index][8]
llocalWavePath = namesAndPaths[index][0];  // string-ize current namesAndPaths entry

  loadwave /o/q/p=FC localWavePath+"extF"
  loadwave /o/q/p=FC localWavePath+"extZ"
  loadwave /o/q/p=FC localWavePath+"extS"
  loadwave /o/q/p=FC localWavePath+"extD"
  loadwave /o/q/p=FC localWavePath+"F_smth_box"
  loadwave /o/q/p=FC localWavePath+"F_smth_boxDIF"

  WAVE localWaveForce = $(localWavePath+"extF");  // local "pointer" to the retract stuff.
  WAVE localWaveZed = $(localWavePath+"extZ");
  WAVE localWaveSep = $(localWavePath+"extS");
  WAVE localWaveDef = $(localWavePath+"extD");
  WAVE localWaveRForce = $(localWavePath+"retF");
  WAVE ForceBoxSmth = $(localWavePath+"F_smth_box")
  WAVE ForceBoxSmthDIF = $(localWavePath+"F_smth_boxDIF")

  DIMENSION = DimSize(localWaveZed, 0)  // dimension of the present wave.
  numOfPeaks = 0
  Make/O/N=(0,3) $(localWavePath+"Peaks")
      // a Peaks wave to store the recorded peaks, for each FC.
  WAVE forcePeaks = $(localWavePath+"Peaks")
  Setscale d 0,0,"N",forcePeaks
  index2 = 0
  wavestats/q/z ForceBoxSmthDIF
  index2 = V_minloc-2*minPixel

  do
      if (ForceBoxSmthDIF[index2] < cutoff)
          // if the smoothed FC exceeds 4 SD's away from average.
          statistics[1] += 1
          flag = 1
          index3 = 3*minPixel  // tweaky : width minimum

      do // keep incrementing index3 to span the peak,
          // until the Force value dips back to pre- spanning level.
          index3 = ceil(0.5*minPixel)
          while (ForceBoxSmthDIF[index3+index2] <= ForceBoxSmthDIF[index2])
              && (index2+index3) < DIMENSION)

          if (index2+index3 == DIMENSION)
// ---- THIS FINDS THE POSITION OF EXIT POINT OF BREAKTHROUGH
Redimension /N=(Dimsize (forcePeaks,0)+1,-1) forcePeaks
forcePeaks[Dimsize (forcePeaks,0)][0]=
localWaveSep[Dimsize(localWaveSep,0)-5]
forcePeaks[Dimsize (forcePeaks,0)][1]=
localWaveForce[Dimsize(localWaveSep,0)-5]
forcePeaks[Dimsize (forcePeaks,0)][2]=Dimsize(localWaveSep,0)-5

// ---- THIS FINDS THE POSITION OF ENTRANCE POINT
// OF BREAKTHROUGH
Redimension /N=(Dimsize (forcePeaks,0)+1,-1) forcePeaks
forcePeaks[Dimsize (forcePeaks,0)][0]=
localWaveSep[Dimsize(localWaveSep,0)-5]
forcePeaks[Dimsize (forcePeaks,0)][1]=
localWaveForce[Dimsize(localWaveSep,0)-5]
forcePeaks[Dimsize (forcePeaks,0)][2]=Dimsize(localWaveSep,0)-5

variable contactPosn=contactPtFinder(ForceBoxSmth,localWaveSep,
round(Dimsize(localWaveSep,0)-5))
Redimension /N=(Dimsize (forcePeaks,0)+1,-1) forcePeaks
forcePeaks[Dimsize (forcePeaks,0)][0]=localWaveSep[contactPosn]
forcePeaks[Dimsize (forcePeaks,0)][1]=localWaveForce[contactPosn]
forcePeaks[Dimsize (forcePeaks,0)][2]=contactPosn
numsAndPaths[index][6]=AdhGetter(localWaveRForce)
breathe
endif
variable UBound=index2+index3+2*minPixel //tweakable
// even after finding the index3, add a bigger window
variable LBound=index2+index3-2*minPixel //tweakable
// (x-dx, x, x+ 1.5dx)
if (LBound<0) //what happens if im already at the left edge.
wavestats/q/z/m=1/R=[0,UBound] localWaveForce
elseif ((UBound)>(DIMENSION-1))
// what happens if I'm at the right edge.
wavestats/q/z/m=1/R=[LBound,DIMENSION-1] localWaveForce
else // all else
wavestats/q/z/m=1/R=[LBound,UBound] localWaveForce
endif
// the wavestats is primarily to find the lowest point in the range,
// I need V_minloc, which is the index of the MIN.
WAVE w_coef
CurveFit/Q/NTHR=0 line localWaveDef[V_minloc,V_minloc+round
(0.4*(DIMENSION-V_minloc)) /X=localWaveZed]

// ---- THIS FINDS THE POSITION OF EXIT POINT OF BREAKTHROUGH
if (abs(enoise(1))>0.75)
print "Index:,namesandpaths[index][0],"suggested correction coef",w_coef[1]
print "from:,V_minloc,"to",V_minloc+round(0.4*(DIMENSION-V_minloc))
endif

Redimension /N=(Dimsize (forcePeaks,0)+1,-1) forcePeaks
forcePeaks[Dimsize (forcePeaks,0)][0]=localWaveSep[V_minloc]
forcePeaks[Dimsize (forcePeaks,0)][1]=localWaveForce[V_minloc]
forcePeaks[Dimsize (forcePeaks,0)][2]=V_minloc

// ---- THIS FINDS THE POSITION OF ENTRANCE POINT OF BREAKTHROUGH
wavestats/q/z/m=1/R=[0,V_{minloc}] localWaveForce//->
Redimension /N=(Dimsize (forcePeaks,0)+1,-1) forcePeaks
forcePeaks[Dimsize (forcePeaks,0)][0]=localWaveSep[V_{maxloc}]
forcePeaks[Dimsize (forcePeaks,0)][1]=localWaveForce[V_{maxloc}]
forcePeaks[Dimsize (forcePeaks,0)][2]=V_{maxloc}

// ---- THIS FINDS THE POSITION OF CONTACT POINT
contactPosn=contactPtFinder(ForceBoxSmth,localWaveSep,round(1.0*(V_{maxloc}-10)))
Redimension /N=(Dimsize (forcePeaks,0)+1,-1) forcePeaks
forcePeaks[Dimsize (forcePeaks,0)][0]=localWaveSep[contactPosn]
forcePeaks[Dimsize (forcePeaks,0)][1]=localWaveForce[contactPosn]
forcePeaks[Dimsize (forcePeaks,0)][2]=contactPosn

if ((forcePeaks[Dimsize (forcePeaks,0)-3][0]-
            forcePeaks[Dimsize (forcePeaks,0)-2][0])>=constants[4])
   //if contact point is > constants[4] away from point B (invalid breakthrough)
   statistics[1]=1
   namesAndPaths[index][3]="ContactPt_bad"
   namesAndPaths[index][4]="Thickness"
   namesAndPaths[index][5]="Breakthrough"
   namesAndPaths[index][6]="Adhesion"
   numsAndPaths[index][3]=forcePeaks[Dimsize (forcePeaks,0)-2][0]-
                              forcePeaks[Dimsize (forcePeaks,0)-1][0]; //Indents
   numsAndPaths[index][4]=forcePeaks[Dimsize (forcePeaks,0)-3][0]-
                              forcePeaks[Dimsize (forcePeaks,0)-1][0]; //Thickness
   numsAndPaths[index][5]=forcePeaks[Dimsize (forcePeaks,0)-2][1]-
                              forcePeaks[Dimsize (forcePeaks,0)-1][1]; //Breakthrough Forces
   numsAndPaths[index][6]=AdhGetter(localWaveRForce) //adhesion
elseif (forcePeaks[Dimsize (forcePeaks,0)-2][0]-
            forcePeaks[Dimsize (forcePeaks,0)-1][0]>=1e-9)
   //if the difference between contact point and point B is >= 1nm.
   statistics[2]=1
   namesAndPaths[index][3]="Indent_OK"
   namesAndPaths[index][4]="Thickness"
   namesAndPaths[index][5]="Breakthrough"
   namesAndPaths[index][6]="Adhesion"
   numsAndPaths[index][3]=forcePeaks[Dimsize (forcePeaks,0)-2][0]-
                              forcePeaks[Dimsize (forcePeaks,0)-1][0]; //Indents
   numsAndPaths[index][4]=forcePeaks[Dimsize (forcePeaks,0)-3][0]-
                              forcePeaks[Dimsize (forcePeaks,0)-1][0]; //Thickness
   numsAndPaths[index][5]=forcePeaks[Dimsize (forcePeaks,0)-2][1]-
                              forcePeaks[Dimsize (forcePeaks,0)-1][1]; //Breakthrough Forces
   numsAndPaths[index][6]=AdhGetter(localWaveRForce)
elseif (forcePeaks[Dimsize (forcePeaks,0)-2][0]-
            forcePeaks[Dimsize (forcePeaks,0)-1][0]>=0)
   //if the difference between contact point and point B is at least positive.
   statistics[2]=1
   namesAndPaths[index][3]="Indent_Intermediate"
   namesAndPaths[index][4]="Thickness"
   namesAndPaths[index][5]="Breakthrough"
   namesAndPaths[index][6]="Adhesion"
   numsAndPaths[index][3]=forcePeaks[Dimsize (forcePeaks,0)-2][0]-
                              forcePeaks[Dimsize (forcePeaks,0)-1][0]; //Indents
   numsAndPaths[index][4]=forcePeaks[Dimsize (forcePeaks,0)-3][0]-
                              forcePeaks[Dimsize (forcePeaks,0)-1][0]; //Thickness
else
   //if the difference between contact point and point B is <= 1nm. No adhesion.
   statistics[2]=1
   namesAndPaths[index][3]="Indent_OK"
   namesAndPaths[index][4]="Thickness"
   namesAndPaths[index][5]="Breakthrough"
   namesAndPaths[index][6]="Adhesion"
   numsAndPaths[index][3]=forcePeaks[Dimsize (forcePeaks,0)-2][0]-
                              forcePeaks[Dimsize (forcePeaks,0)-1][0]; //Indents
   numsAndPaths[index][4]=forcePeaks[Dimsize (forcePeaks,0)-3][0]-
                              forcePeaks[Dimsize (forcePeaks,0)-1][0]; //Thickness

numsAndPaths[index][5]=forcePeaks[Dimsize (forcePeaks,0)-2][1]-forcePeaks[Dimsize (forcePeaks,0)-1][1]; //Brkthrough Forces
numsAndPaths[index][6]=AdhGetter(localWaveRForce) //adhesion
else
    //if the difference between contact point and point B is negative
    // (more of a correction issue)
    namesAndPaths[index][3]="Indent_Negative"
    namesAndPaths[index][4]="Thickness"
    namesAndPaths[index][5]="Breakthrough"
    namesAndPaths[index][6]="Adhesion"
    numsAndPaths[index][3]=forcePeaks[Dimsize (forcePeaks,0)-2][0]-forcePeaks[Dimsize (forcePeaks,0)-1][0]; //Indents
    numsAndPaths[index][4]=forcePeaks[Dimsize (forcePeaks,0)-3][0]-forcePeaks[Dimsize (forcePeaks,0)-1][0]; //Thickness
    numsAndPaths[index][5]=forcePeaks[Dimsize (forcePeaks,0)-2][1]-forcePeaks[Dimsize (forcePeaks,0)-1][1]; //Brkthrough Forces
    numsAndPaths[index][6]=AdhGetter(localWaveRForce) //adhesion
endif
//tweakable .... to Add, if lowest point = at the UBound, or LBOUND, 
// then something is wrong.
break
elseif (index2==DIMENSION-1)
    //if the condition wasnt met, and I've reached the end of this particular force 
    // curve - normally I would assume that there is no breakthrough and just deal
    // with stuff.
    flag=1
    namesAndPaths[index][3]="Indent_none"
    namesAndPaths[index][4]="Thickness_none"
    namesAndPaths[index][5]="Breakthrough_none"
elseif(0) //temporarily ignore the no breakthrough case.
    //if the condition wasnt met, and I've reached the end of this particular force curve – 
    // normally I would assume that there is no breakthrough and just deal with stuff.
    flag=1
    namesAndPaths[index][3]="Indent_none"
    namesAndPaths[index][4]="Thickness_none"
    namesAndPaths[index][5]="Breakthrough_none"
    namesAndPaths[index][6]="Adhesion"

// ---- THIS FINDS THE POSITION OF EXIT POINT OF BREAKTHROUGH
Redimension /N=(Dimsize (forcePeaks,0)+1,-1) forcePeaks
forcePeaks[Dimsize (forcePeaks,0)][0]=localWaveSep[Dimsize(localWaveSep,0)-5]
forcePeaks[Dimsize (forcePeaks,0)][1]=localWaveForce[Dimsize(localWaveSep,0)-5]
forcePeaks[Dimsize (forcePeaks,0)][2]=Dimsize(localWaveSep,0)-5

// ---- THIS FINDS THE POSITION OF ENTRANCE POINT OF BREAKTHROUGH
Redimension /N=(Dimsize (forcePeaks,0)+1,-1) forcePeaks
forcePeaks[Dimsize (forcePeaks,0)][0]=localWaveSep[Dimsize(localWaveSep,0)-5]
forcePeaks[Dimsize (forcePeaks,0)][1]=localWaveForce[Dimsize(localWaveSep,0)-5]
forcePeaks[Dimsize (forcePeaks,0)][2]=Dimsize(localWaveSep,0)-5

contactPosn=contactPtFinder(ForceBoxSmth,localWaveSep, round(Dimsize(localWaveSep,0)-5))
Redimension /N=(Dimsize (forcePeaks,0)+1,-1) forcePeaks
forcePeaks[Dimsize (forcePeaks,0)][0]=localWaveSep[contactPosn]
forcePeaks[Dimsize (forcePeaks,0)][1]=localWaveForce[contactPosn]
forcePeaks[Dimsize (forcePeaks, 0)][2] = contactPosn
numsAndPaths[index][6] = AdhGetter(localWaveRForce)
endif
index2 += 1
while ((index2 < DIMENSION) && (!flag))
  save/c/o/p = FC localWaveSep, forcePeaks
  killwaves /z localWaveForce, localWaveZed, forcepeaks, localWaveRForce, localWaveSep,
  killwaves /z ForceBoxSmth, localWaveDef
  WAVEClear localWaveForce, localWaveZed, ForceBoxSmthDIF, ForceBoxSmth, forcepeaks
  WAVEClear localWaveRForce, localWaveSep, localWaveDef, w_coef
  if (mod(index, round(.25 * batches)) == 0)
    print "Progress: ", round(100 * index/namesAndPathsDim), "%"
  endif
endfor
WAVEClear namesAndPaths
End // Function Parser

Function AdhGetter(localWaveRForce)
  WAVE localWaveRForce
  wavestats /q/z/r = [0, 20] localWaveRForce
  variable baseLine = V_avg
  wavestats /q/z localWaveRForce
  WAVEClear localWaveRForce
  return abs(baseLine - V_min)
End // Function AdhGetter

Function contactPtFinder(localWaveY, localWaveX, brkThruEntrancePt)
  WAVE localWaveY
  WAVE localWaveX
  variable brkThruEntrancePt
  WAVE constants
  Variable HalfD = nnIndex(localWaveX, "<", constants[4], brkThruEntrancePt)
  Duplicate /O/R = [HalfD, brkThruEntrancePt] localWaveY localWaveDef2
  Duplicate /O/R = [HalfD, brkThruEntrancePt] localWaveX localWaveZed2
  variable dimension = DimSize(localWaveDef2, 0); // dim of current Wave
  variable index2
  variable foundIndex = inf;
  variable foundInd = inf;
  Make/O/N = (dimension) "$\text{MSE}\$";
  WAVE localWaveMSE = "$\text{MSE}\$";
  for (index2 = dimension-2; index2 > 1; index2 -= 1) // index must go from 1 to dim-1.
    // use the last 1/2 of the data to minimize run-time;
    localWaveMSE = 0;
    CurveFit/Q/N/NTHR=0 line localWaveDef2[0, index2] /X = localWaveZed2 /R = localWaveMSE;
    CurveFit/Q/N/NTHR=0 line localWaveDef2[index2, dimension-1] /X = localWaveZed2 /R = localWaveMSE;
  localWaveMSE = sqrt((localWaveMSE)^2);
  Wavestats/Q/M = 1 localWaveMSE;
  variable quota = abs((localWaveY[HalfD] - localWaveY[index2 + HalfD]) /}

if(V_avg<foundIndex&&V_avg!=0&&quota<0.02)
    foundIndex=V_avg;
    foundInd=index2;
endif
endfor

killWaves/z localWaveMSE
killwaves/z localWaveDef2, localWaveZed2
WAVEClear localWaveMSE, localWaveDef2, localWaveZed2
WAVEClear localWaveY,localWaveX,constants
return foundInd+HalfD
End //Function contactPtFinder

9.1.2 Instrument Specific File Handing

This section contains the functions that were used to properly process AFM force curves and
AFM force maps generated by the instrument’s built-in software. In this thesis, the force curves
from three companies: JPK, Veeco/ Digital Instruments, and Asylum Research (MFP) were
analyzed. Importing of force curve information required working with binary or ASCII-based
files, the reading of header sections which contained information such as sensitivity values,
spring constants (at time of data measurements), etc., as well as generating universal force curves
that would work with the rest of the code.

9.1.2.1 IGOR Pro Code

Function JPKLoad(pathName,baseName,noPrecheck)
    string pathName   // Name of symbolic path or empty string, "" to get dialog
    string baseName  //baseName you would like to provide.
    variable noPrecheck
    string fileName=""  
    Make/T/O headerReadingWave
    WAVE statistics
    variable springConst
    string zeros  // a part of the final wavename for the force waves we are about to make.
    variable index=0
    variable bogus=0
    variable toDel=0
    variable recordIndex=0
    variable xPos,yPos
    variable offset
    WAVE constants
    variable timerRefNum
    variable elapsedTime=0
    timerRefNum=startMSTimer
print "Initial file loading stage, wait a moment."
Make/T/O/N=(0,15) namesAndPaths
Make/O/N=(0,15) numsAndPaths

statistics[0]=0
do // Loop through each file in folder
    fileName = IndexedFile(SpathName, index+bogus-toDel,"????")
    if (strlen(fileName) == 0) // No more files?
        break // Break out of loop
    endif
    if (!(CmpStr(fileName,"list.dat")&&CmpStr(fileName,".DS_Store")
    &&CmpStr(fileName,"Results.pic")))
        bogus++=1
    else
        statistics[0]+=1
        LoadWave/o/q/G/N=wave/P=$pathName/O/L={0,0,0,0,2} fileName
        // loads the relevant tabular values into waves with wave baseName wave.
        Grep /E="springConstant:" /P=$pathName fileName as headerReadingWave
        sscanf headerReadingWave[0], "# springConstant: %f", springConst;
        // copies the value of spring constant in headers.
        Grep /E="xPosition:" /P=$pathName fileName as headerReadingWave
        sscanf headerReadingWave[0], "# xPosition: %f", xPos;
        // copies the value of spring constant in headers.
        Grep /E="yPosition:" /P=$pathName fileName as headerReadingWave
        sscanf headerReadingWave[0], "# yPosition: %f", yPos;
        // copies the value of spring constant in headers.
        Grep /E="index:" /P=$pathName fileName as headerReadingWave
        sscanf headerReadingWave[0], "# index: %d", recordIndex;
        //copies the value of spring constant in headers.
        if (recordIndex<10)  //string manipulation step, just adds placeholder 0's.
            zeros="000"
        elseif (recordindex<100)
            zeros="00"
        elseif (recordindex<1000)
            zeros="0"
        else
            zeros=""
        endif
        Duplicate /O wave0 $(baseName+zeros+num2str(recordIndex)+"extZ")
        // copies the tabulated data as extend Z
        Duplicate /O wave1 $(baseName+zeros+num2str(recordIndex)+"extF")
        // copies the tabulated data as extend Force
        Duplicate /O wave1 $(baseName+zeros+num2str(recordIndex)+"extD")
        // prepares a retract Deflection (distance) placeholder
        Duplicate /O wave1 $(baseName+zeros+num2str(recordIndex)+"extS")
        // prepares a retract Separation placeholder
        // the following "WAVE xxxxx" statements just makes local "pointers"
        // to the above waves.
        WAVE localWaveEZed=$(baseName+zeros+num2str(recordIndex)+"extZ")
WAVE localWaveEForce=$(baseName+zeros+num2str(recordIndex)+"extF")
WAVE localWaveESep=$(baseName+zeros+num2str(recordIndex)+"extS")
WAVE localWaveEDef=$(baseName+zeros+num2str(recordIndex)+"extD")

Setscale d 0,0,"N",localWaveEForce
Setscale d 0,0,"m",localWaveEZed
Setscale d 0,0,"m",localWaveEDef
Setscale d 0,0,"m",localWaveESep

Duplicate /O wave3 $(baseName+zeros+num2str(recordIndex)+"retF")
// copies the tabulated data as retract Force
WAVE localWaveRForce=$(baseName+zeros+num2str(recordIndex)+"retF")
Setscale d 0,0,"N",localWaveRForce

Duplicate /O wave2 $(baseName+zeros+num2str(recordIndex)+"retZ")
// copies the tabulated data as retract Force
Duplicate /O wave3 $(baseName+zeros+num2str(recordIndex)+"retD")
// copies the tabulated data as retract Force
Duplicate /O wave3 $(baseName+zeros+num2str(recordIndex)+"retS")
// copies the tabulated data as retract Force

WAVE localWaveRZed=$(baseName+zeros+num2str(recordIndex)+"retZ")
WAVE localWaveRDef=$(baseName+zeros+num2str(recordIndex)+"retD")

Setscale d 0,0,"m",localWaveRZed
Setscale d 0,0,"m",localWaveRDef
Setscale d 0,0,"m",localWaveRSep

// calculates a quick offset for Extend's Z and quick offset for Extend's Force
wavestats /q/z/r=[0,50] localWaveEZed
offset=V_avg
localWaveEZed=offset-localWaveEZed
offset=localWaveEForce[0]
localWaveEForce=localWaveEForce-offset
localWaveEDef=localWaveEForce/springConst
// calculates deflection (retraction) from springConst and Force data.
localWaveESep=localWaveEZed - localWaveEDef/constants[6]
// calculates separation as Z - Defl.

wavestats /q/z/r=[0,50] localWaveRZed
offset=V_avg
localWaveRZed=offset-localWaveRZed

offset=localWaveRForce[0]
localWaveRForce=localWaveRForce-offset
localWaveRDef=localWaveRForce/springConst
// calculates deflection (retraction) from springConst and Force data.
localWaveRSep=localWaveRZed - localWaveRDef/constants[6]
// calculates separation as Z - Defl.

Redimension /N=(Dimsize (namesAndPaths,0)+1,-1) namesAndPaths;
// stores info about this particular set into namesAndPaths
Redimension /N=(Dimsize (numsAndPaths,0)+1,-1) numsAndPaths;
// stores info about this particular set into namesAndPaths
namesAndPaths[Dimsize (namesAndPaths,0)][0]=baseName+zeros+num2str(recordIndex)
namesAndPaths[Dimsize (namesAndPaths,0)][1]="xPos"
namesAndPaths[Dimsize (namesAndPaths,0)][2]="yPos"

numsAndPaths[Dimsize (numsAndPaths,0)][0]=recordIndex
numsAndPaths[Dimsize (numsAndPaths,0)][1]=xPos
numsAndPaths[Dimsize (numsAndPaths,0)][2]=yPos
index += 1

save/c/o/p=FC localWaveEDef, localWaveEF0rce, localWaveESep
save/c/o/p=FC localWaveEZed, localWaveRForce
killwaves/Z localWaveEDef, localWaveEF0rce, localWaveESep
killwaves/Z localWaveEZed, localWaveRForce
save/c/o/p=FC localWaveRDef, localWaveRSep, localWaveRZed
killwaves/Z localWaveRDef, localWaveRSep, localWaveRZed
endif // guarantees list.dat is not included
while (noPrecheck||index<constants[8])
KillWaves/Z headerReadingWave, wave0, wave1, wave2, wave3
WAVEClear localWaveEZed, localWaveEF0rce
WAVEClear headerReadingWave, localWaveRDef
WAVEClear localWaveRZed, localWaveRForce
elapsedTime=stopMSTimer(timerRefNum)/60e6
print "Total file loading time: ",floor(elapsedTime/60), "hours ",
print mod(round(elapsedTime),60), "minutes."
return 0 // Signifies success.
End // Function JPKLoad(pathName,baseName,noPrecheck)

Function VeecoLoad(pathName,baseName,noPrecheck)
string pathName   // Name of symbolic path or empty string, "" to get dialog
string baseName  // baseName you would like to provide.
variable noPrecheck
string fileName=""
Make/T/O headerReadingWave
WAVE statistics
wAVE/T textconstants
WAVE constants
variable springConst, dataOffset, zScale, rampSize, deflSens, sampsLineExt
variable sampsLineRet, zSens, FVsampsLine, FVLines, scanSize
string zeros // a part of the final wavename for the force waves we are about to make.
variable index=0
variable bogus=0
variable recordIndex=0
variable toDel=0
variable xPos,yPos
variable offset
variable timerRefNum
variable elapsedTime=0
timerRefNum=startMSTimer
print "Initial file loading stage, wait a moment."
Make/T/O/N=(0,15) namesAndPaths
Make/O/N=(0,15) numsAndPaths
statistics[0]=0
do // Loop through each file in folder
    fileName = IndexedFile($pathName, index+bogus-toDel,"????")
    if (strlen(fileName) == 0) // No more files?
        break // Break out of loop
endif
if(!(CmpStr(fileName,"list.dat")&&&CmpStr(fileName,".DS_Store")
&;&CmpStr(fileName,"Results.pic")))
    bogus+=1
else
    statistics[0]++

    //----GREP REGION
    Grep /E="\\Scan size:" /P=$pathName fileName as headerReadingWave
    if(dimsize(headerReadingWave,0)>0)
        sscanf headerReadingWave[0], "Scan size: %f", scanSize;
        //copies the value of scanSize for the header
        scanSize*=1e-9
        redimension /N=0 headerReadingWave
    endif

    Grep /E="\\Samps/line:" /P=$pathName fileName as headerReadingWave
    if(dimsize(headerReadingWave,0)>0)
        sscanf headerReadingWave[0], "Samps/line: %i", FVsampsLine;
        //copies the value of FV samps per Line
        sscanf headerReadingWave[1], "Samps/line: %i %i",
        sampsLineExt,sampsLineRet;
        // copies the value of pts per Ext and Ret
        redimension /N=0 headerReadingWave
    endif

    Grep /E="\\Lines:" /P=$pathName fileName as headerReadingWave
    if(dimsize(headerReadingWave,0)>0)
        sscanf headerReadingWave[0] , "Lines: %i", FVLines;
        //copies the value lines in FV
        redimension /N=0 headerReadingWave
    endif

    Grep /E="\\Spring constant:" /P=$pathName fileName as headerReadingWave
    if(dimsize(headerReadingWave,0)>0)
        sscanf headerReadingWave[0], "Spring constant: %f", springConst;
        //copies the value of spring constant in headers
        redimension /N=0 headerReadingWave
    endif

    Grep /E="\\@4:Z scale: V [Sens. Deflection]"
    /P=$pathName fileName as headerReadingWave
    if(dimsize(headerReadingWave,0)>0)
        sscanf headerReadingWave[0], "@4:Z scale: V [Sens. Deflection]"
        (%f V/LSB)\", zScale;   //copies the value of ZScale
        redimension /N=0 headerReadingWave
    endif

    Grep /E="\\@4:Ramp size: V [Sens. Zscan]"
    /P=$pathName fileName as headerReadingWave
    if(dimsize(headerReadingWave,0)>0)
        sscanf headerReadingWave[0], "@4:Ramp size: V [Sens. Zscan]"
        (%f V/LSB) %f V", rampSize, rampSize;
        //copies the value of ramp Size (first one is filler)
        redimension /N=0 headerReadingWave
    endif

    Grep /E="\\Data offset:" /P=$pathName fileName as headerReadingWave
if (dimsize(headerReadingWave,0)>0)
    sscanf headerReadingWave[0], "\Data offset: %i", dataOffset;
    //copies the value of data offset in headers
    redimension /N=0 headerReadingWave
endif
Grep /E="\\@Sens. Zscan:" /P=$pathName fileName as headerReadingWave
if (dimsize(headerReadingWave,0)>0)
    sscanf headerReadingWave[0], "(@Sens. Zscan: %f nm/V", zSens;
    //copies the value of ZSens in headers
    redimension /N=0 headerReadingWave
endif
Grep /E="\\@Sens. Deflection:" /P=$pathName fileName as headerReadingWave
if (dimsize(headerReadingWave,0)>0)
    sscanf headerReadingWave[0], "(@Sens. Deflection: %f nm/V", deflSens;
    //copies the value of DeflSens in headers
    redimension /N=0 headerReadingWave
endif
recordindex=index
   //-----end of GREP REGION

GBLoadWave/Q/B/T={16,4}/S=(dataOffset)/W=1 /P=$pathName fileName
WAVE wave0
   // loads the relevant tabular values into waves with wave baseName wave.
   // Note: this code is only assuming <= 9999 waves to process.
   if (recordIndex<10)  //string manipulation step, just adds placeholder 0's.
       zeros="000"
   elseif (recordindex<100)
       zeros="00"
   elseif (recordindex<1000)
       zeros="0"
   else
       zeros="""
   endif

Make /O/N=(sampsLineExt) $(baseName+zeros+num2str(recordIndex)+"extZ")
   // copies the tabulated data as extend Z
Make /O/N=(sampsLineExt) $(baseName+zeros+num2str(recordIndex)+"extF")
   // copies the tabulated data as extend Force
Make /O/N=(sampsLineExt) $(baseName+zeros+num2str(recordIndex)+"extD")
   // prepares a retract Deflection (distance) placeholder
Make /O/N=(sampsLineExt) $(baseName+zeros+num2str(recordIndex)+"extS")
   // prepares a retract Separation placeholder

Make /O/N=(sampsLineRet) $(baseName+zeros+num2str(recordIndex)+"retZ")
   // copies the tabulated data as extend Z
Make /O/N=(sampsLineRet) $(baseName+zeros+num2str(recordIndex)+"retF")
   // copies the tabulated data as extend Force
Make /O/N=(sampsLineRet) $(baseName+zeros+num2str(recordIndex)+"retD")
   // prepares a retract Deflection (distance) placeholder
Make /O/N=(sampsLineRet) $(baseName+zeros+num2str(recordIndex)+"retS")
   // prepares a retract Separation placeholder

   // the following "WAVE xxxxx" statements just makes local "pointers"
   // to the above waves.
WAVE localWaveEZed=$(baseName+zeros+num2str(recordIndex)+"extZ")
WAVE localWaveEForce=$(baseName+zeros+num2str(recordIndex)+"extF")
WAVE localWaveESep=$(baseName+zeros+num2str(recordIndex)+"extS")
WAVE localWaveEDef=$(baseName+zeros+num2str(recordIndex)+"extD")

WAVE localWaveRForce=$(baseName+zeros+num2str(recordIndex)+"retF")
WAVE localWaveRSep=$(baseName+zeros+num2str(recordIndex)+"retS")
WAVE localWaveRZed=$(baseName+zeros+num2str(recordIndex)+"retZ")
WAVE localWaveRDef=$(baseName+zeros+num2str(recordIndex)+"retD")

localWaveEZed=p*rampSize*zSens/(sampsLineExt-1)
localWaveEDef=wave0[p]
localWaveEDef*=deflSens*zScale
localWaveEForce=localWaveEDef*springConst
localWaveESep=localWaveEDef
redimension /N=(sampsLineExt-10) localWaveEZed
redimension /N=(sampsLineExt-10) localWaveEDef
redimension /N=(sampsLineExt-10) localWaveEForce
redimension /N=(sampsLineExt-10) localWaveESep

localWaveRZed=p*rampSize*zSens/(sampsLineRet-1)
localWaveRDef=wave0[p+sampsLineExt]
localWaveRDef*=deflSens*zScale
localWaveRForce=localWaveRDef*springConst
localWaveRSep=localWaveRDef
redimension /N=(sampsLineRet-10) localWaveRZed
redimension /N=(sampsLineRet-10) localWaveRDef
redimension /N=(sampsLineRet-10) localWaveRForce
redimension /N=(sampsLineRet-10) localWaveRSep

reverse/P localWaveEDef, localWaveEForce, localWaveESep, localWaveEZed
reverse/P localWaveRForce,localWaveRDef, localWaveRSep, localWaveRZed

localWaveEZed*=1e-9
localWaveEForce*=1e-9
localWaveESep*=1e-9
localWaveEDef*=1e-9
localWaveRZed*=1e-9
localWaveRForce*=1e-9
localWaveRSep*=1e-9
localWaveRDef*=1e-9

Setscale d 0,0,"N",localWaveEForce
Setscale d 0,0,"m",localWaveEZed
Setscale d 0,0,"m",localWaveEDef
Setscale d 0,0,"m",localWaveESep
Setscale d 0.0,"N",localWaveRForce
Setscale d 0.0,"m",localWaveRZed
Setscale d 0.0,"m",localWaveRDef
Setscale d 0.0,"m",localWaveRSep

// calculates a quick offset for Extend's Z and quick offset for Extend's Force
offset=localWaveEZed[0]
localWaveEZed=offset-localWaveEZed

offset=localWaveEForce[0]
localWaveEForce=localWaveEForce-offset
localWaveEDef=localWaveEForce/springConst
// calculates deflection (retraction) from springConst and Force data.
localWaveESep=localWaveEZed - localWaveEDef
// calculates separation as Z - Defl.

offset=localWaveRZed[0]
localWaveRZed=offset-localWaveRZed

offset=localWaveRForce[0]
localWaveRForce=localWaveRForce-offset
localWaveRDef=localWaveRForce/springConst
// calculates deflection (retraction) from springConst and Force data.
localWaveRSep=localWaveRZed - localWaveRDef/constants[6]
// calculates separation as Z - Defl.

Redimension /N=(Dimsize (namesAndPaths,0)+1,-1) namesAndPaths;
// stores info about this particular set into namesAndPaths
Redimension /N=(Dimsize (numsAndPaths,0)+1,-1) numsAndPaths;
// stores info about this particular set into namesAndPaths
namesAndPaths[Dimsize (namesAndPaths,0)][0]=
baseName+zeros+num2str(recordIndex)
namesAndPaths[Dimsize (namesAndPaths,0)][1]="xPos"
namesAndPaths[Dimsize (namesAndPaths,0)][2]="yPos"
numsAndPaths[Dimsize (numsAndPaths,0)][0]=recordIndex

returnFVPos(recordIndex, Dimsize (numsAndPaths,0),
FVsampsLine,FVLines,scanSize,numsAndPaths,1,2,textConstants[0])

index += 1

save/c/o/p=FC localWaveEDef, localWaveEForce, localWaveESep
save/c/o/p=FC localWaveEZed, localWaveRForce
killwaves/Z localWaveEDef, localWaveEForce, localWaveESep
killwaves/Z localWaveEZed, localWaveRForce
save/c/o/p=FC localWaveRDef, localWaveRSep, localWaveRZed
killwaves/Z localWaveRDef, localWaveRSep, localWaveRZed,wave0
endif // guarantees list.dat is not included
while (noPreCheck||index<constants[8])
KillWaves /Z headerReadingWave, wave0
WAVEClear localWaveEZed, localWaveEForce, localWaveESep, localWaveEZed
WAVEClear headerReadingWave, localWaveEDef, localWaveRDef
WAVEClear localWaveRZed, localWaveRForce, localWaveRSep, localWaveRZed,
elapsedTime=stopMSTimer(timerRefNum)/60e6
print "Total file loading time: ".floor(elapsedTime/60), "hours "
print mod(round(elapsedTime),60), "minutes."
return 0 // Signifies success.
End // Function VeecoLoad(pathName,baseName,noPrecheck)

Function MFPLoad(pathName,baseName,noPrecheck)
    string pathName   // Name of symbolic path or empty string, "" to get dialog
    string baseName  // baseName you would like to provide.
    variable noPrecheck
    string fileName=""
    Make/T/O headerReadingWave
    WAVE statistics
    variable springConst
    string zeros // a part of the final wavename for the force waves we are about to make.
    variable index=0
    variable bogus=0
    variable toDel=0
    string delFilename=""
    variable recordIndex=0
    variable xPos,yPos
    variable offset
    WAVE constants
    variable timerRefNum
    variable elapsedTime=0
    timerRefNum=startMSTimer
    print "Initial file loading stage, wait a moment."
    Make/T/O=N=(0,15) root:namesAndPaths
    Make/O=N=(0,15) root:numsAndPaths
    WAVE/T namesandPaths=root:namesAndPaths
    WAVE numsandpaths=root:numsAndPaths
    statistics[0]=0
    pathName=fileLoad(pathName)

    string stringinQuestion,substring
    variable beginpos,endpos,temp,extendend
    string folderName
    variable bogusfolders,bogusfiles,countfiles,countfolders=0
    do
        folderName=IndexedDir($pathName, (bogusfolders+countfolders),0)
        print "FolderName: ",folderName
        if ((strlen(foldername)>0))
            if ((cmpstr(foldername,"fc")==0 )||(cmpstr(foldername,"results")==0))|
                (cmpstr(foldername,"pics")==0))
                bogusfolders+=1
            else
                countfiles=0
                do // Loop through each file in folder
                    pathinfo $pathname
                    stringinQuestion=$_path+folderName+":"
                    newpath/o/q/z subPath stringInQuestion
                    fileName =
                    IndexedFile(subpath,(bogusfiles+countfiles-toDel),".ibw")
                    delFilename=fileName
                    if (strlen(fileName) == 0) // No more files?
                        break // Break out of loop
                countfiles+=1
                end // Loop through each file in folder
            end
        end
    end // do
    statistics[0]=0
    pathName=fileLoad(pathName)
endif

filename=filename[0,strrchr(filename,".ibw",0)-1]
statistics[0]+=1
LoadWave/o/q/H/N=wave/P=subpath fileName
// loads the relevant tabular values into waves
// with wave baseName wave.
WAVE wave0=$(filename)
if (recordIndex<10)
//string manipulation step, just adds placeholder 0's.
zeros="000"
else if (recordIndex<100)
zeros="00"
else if (recordIndex<1000)
zeros="0"
else
zeros=""
endif

if (DimSize(wave0)==3)
stringInQuestion= note (wave0);
// pulls out the note containing Wave Info
// for said Wave.
beginPos=strrchr
(stringInQuestion, "SpringConstant: ",0);
//searches for the beginning of the SpringConst line
endPos= strstr(stringInQuestion, ",\r",beginPos);
// searches for the RETURN of that Index line.
substring=stringInQuestion[beginPos,endPos];
// pulls out just that line.(economy of strings)
sscanf substring, "SpringConstant: %f", springConst;
//finds the end of extend region.
beginPos=strrchr(stringInQuestion, "Indexes: ",0);
//searches for the beginning of the SpringConst line
endPos= strstr(stringInQuestion, ",\r",beginPos);
// searches for the RETURN of that Index line.
substring=stringInQuestion[beginPos,endPos];
// pulls out just that line.(economy of strings)
sscanf substring, "Indexes: 0,%d", extendEnd;
//finds the end of extend region.
beginPos=strrchr(stringInQuestion, "XLVDT:",0);
//searches for the beginning of the SpringConst line
endPos= strstr(stringInQuestion, ",\r",beginPos);
// searches for the RETURN of that Index line.
substring=stringInQuestion[beginPos,endPos];
// pulls out just that line.(economy of strings)
sscanf substring, "XLVDT:%f", xPos
beginPos=strrchr(stringInQuestion, "YLVDT:",0);
//searches for the beginning of the SpringConst line
endPos= strstr(stringInQuestion, ",\r",beginPos);
// searches for the RETURN of that Index line.
substring=stringInQuestion[beginPos,endPos];
// pulls out just that line.(economy of strings)
sscanf substring, "YLVDT:%f", yPos
beginPos=strrchr(stringInQuestion, "YLVD:",0);
//searches for the beginning of the SpringConst line
endPos= strstr(stringInQuestion, ",\r",beginPos);
// searches for the RETURN of that Index line.
substring=stringInQuestion[beginPos,endPos];
// pulls out just that line.(economy of strings)
sscanf substring, "YLVD:%f", yPos
Duplicate /\r=[0,extendEnd][1] wave0
S(baseName+zeros+num2str(recordIndex)+"extD")
Duplicate /\r=[0,extendEnd][1] wave0
S(baseName+zeros+num2str(recordIndex)+"extF")
Duplicate /\r=[0,extendEnd][1] wave0
S(baseName+zeros+num2str(recordIndex)+"extS")
Duplicate /\r=[0,extendEnd][2] wave0
S(baseName+zeros+num2str(recordIndex)+"extZ")

Duplicate /\r=[extendEnd+1,][1] wave0
S(baseName+zeros+num2str(recordIndex)+"retD")
Duplicate /\r=[extendEnd+1,][1] wave0
S(baseName+zeros+num2str(recordIndex)+"retF")
Duplicate /\r=[0,extendEnd][1] wave0
S(baseName+zeros+num2str(recordIndex)+"retS")
Duplicate /\r=[extendEnd+1,][2] wave0
S(baseName+zeros+num2str(recordIndex)+"retZ")

WAVE localWaveEDef=
S(baseName+zeros+num2str(recordIndex)+"extD")
WAVE localWaveEForce=
S(baseName+zeros+num2str(recordIndex)+"extF")
WAVE localWaveESep=
S(baseName+zeros+num2str(recordIndex)+"extS")
WAVE localWaveEZed=
S(baseName+zeros+num2str(recordIndex)+"extZ")
WAVE localWaveRDef=
S(baseName+zeros+num2str(recordIndex)+"retD")
WAVE localWaveRForce=
S(baseName+zeros+num2str(recordIndex)+"retF")
WAVE localWaveRSep=
S(baseName+zeros+num2str(recordIndex)+"retS")
WAVE localWaveRZed=
S(baseName+zeros+num2str(recordIndex)+"retZ")

wavestats /q/z/r= [0, round(0.10* DimSize(localWaveEZed,0)) ] localWaveEZed
offset=V_avg
localWaveEZed=localWaveEZed-offset

offset=localWaveEForce[0]
localWaveEForce=localWaveEForce-offset

wavestats /q/z/r=[0,round(0.10* DimSize(localWaveEZed,0))] localWaveRZed
offset=V_avg
localWaveRZed=localWaveRZed-offset

offset=localWaveRForce[0]
localWaveRForce=localWaveRForce-offset

// calculates a quick offset for Retract F
//********offset all of ret and ext values first.
localWaveESep=localWaveEZed –
localWaveEDef/constant[6]
// calculates separation as Z - Defl.
localWaveRSep = localWaveRZed – localWaveRDef/constants[6]
// calculates separation as Z - Defl.
localWaveRForce *= springConst
localWaveEForce *= springConst
setdimlabel 1,0,force,localWaveRForce
setdimlabel 1,0,force,localWaveEForce
Setdimlabel 1,0,Sep,localWaveRSep
Setdimlabel 1,0,Sep,localWaveESep

SetScale d 0,0,"N", localWaveRForce
SetScale d 0,0,"N", localWaveEForce
SetScale d 0,0,"m", localWaveRSep
SetScale d 0,0,"m", localWaveESep
SetScale d 0,0,"m", localWaveRDef
SetScale d 0,0,"m", localWaveEDef
SetScale d 0,0,"m", localWaveRZed
SetScale d 0,0,"m", localWaveEZed

redimension /N=(-1,0)
localWaveRForce, localWaveEForce
redimension /N=(-1,0)
localWaveRSep, localWaveESep
redimension /N=(-1,0)
localWaveRDef, localWaveEDef
redimension /N=(-1,0)
localWaveEZed, localWaveRZed

SetScale x 0,0,"", localWaveRForce
SetScale x 0,0,"", localWaveEForce
SetScale x 0,0,"", localWaveRSep
SetScale x 0,0,"", localWaveESep
SetScale x 0,0,"", localWaveRDef
SetScale x 0,0,"", localWaveEDef
SetScale x 0,0,"", localWaveRZed
SetScale x 0,0,"", localWaveEZed

endif

Redimension /N=(Dimsize (namesAndPaths,0)+1,-1)
namesAndPaths;
// stores info about this particular set into namesAndPaths
Redimension /N=(Dimsize (numsAndPaths,0)+1,-1)
numsAndPaths;
// stores info about this particular set into namesAndPaths
namesAndPaths[Dimsize (namesAndPaths,0)][0]=
baseName+zeros+num2str(recordIndex)
namesAndPaths[Dimsize (namesAndPaths,0)][1]="xPos"
namesAndPaths[Dimsize (namesAndPaths,0)][2]="yPos"

numsAndPaths[Dimsize (numsAndPaths,0)][0]=recordIndex
numsAndPaths[Dimsize (numsAndPaths,0)][1]=xPos
numsAndPaths[Dimsize (numsAndPaths,0)][2]=yPos
recordIndex += 1
countfiles += 1
save/c/o/p=FC localWaveEDef, localWaveEForce
save/c/o/p=FC localWaveESep, localWaveEZed
save/c/o/p=FC localWaveRForce
9.1.3 Fitting Functions

These functions contain the code used to perform the curve fitting. These include the Sneddon model for a paraboloidal tip (See Section 2.2.1), as well as several quick exponential fit functions, and other test functions, which may have been used in analysis, but irrelevant to the study (e.g. Feynman-Sanz model).

9.1.3.1 IGOR Pro Code

Function indentFitAll(namesandPaths,numsAndPaths,factor)
  WAVE/T namesandpaths
  WAVE numsandpaths
  variable factor
  variable index
  WAVE constants
  string localWavePath
  variable timerRefNum
  variable elapsedTime=0
  timerRefNum=startMSTimer
  variable minPixel,startA,endB

  for (index=0;index<dimsize(namesandpaths,0);index+=1)
    localWavePath=namesandpaths[index][0]
    loadwave /o/q/p=FC localWavePath+"extF"
    loadwave /o/q/p=FC localWavePath+"extS"
    loadwave /o/q/p=FC localWavePath+"peaks"
    WAVE forcepeaks=$(namesandpaths[index][0]+"peaks")
    WAVE localWaveForce=$(namesandpaths[index][0]+"extF")
WAVE localWaveSep=$(namesandpaths[index][0]+"extS")
minPixel=PixelPerDist(localWaveSep,constants[7])
startA=forcepeaks[2][2]+minPixel
endB=startA+round(.01*factor*(forcepeaks[1][2]-startA))
if ((cmpstr(namesandpaths[index][3],"Indent_none")!=0)||
    (cmpstr(namesandpaths[index][3],"contactpt_bad")!=0))
    if (Dimsize(forcepeaks,0)==3)
        namesandPaths[index][14]="E_Paraboloidal"
        numsandPaths[index][14]=(1/sqrt(constants[0]))*0.75*
        (1-constants[2]^2)*
        indentParaboloidOne(localWaveForce,localWaveSep,startA,endB,"w",index)
    endif
endif
killwaves/Z forcepeaks, localWaveForce, localWaveSep
endif
if ((cmpstr(namesandpaths[index][3],"Indent_none")!=0)||
    (cmpstr(namesandpaths[index][3],"contactpt_bad")!=0))
duplicate/o /R=[0,][14,14] numsandpaths paraboloidalFit
Setscale d 0,0,"Pa",paraboloidalFit
endif
if (waveexists(extractedx1))
kilwaves extractedx1
endif
if (waveexists(extractedy1))
kilwaves extractedy1
endif
elapsedTime=stopMSTimer(timerRefNum)/60e6
print "Modulus fitting time: ",floor(elapsedTime/60), "hours "
print mod(round(elapsedTime),60), "minutes."
End // Function indentFitAll(namesandPaths,numsAndPaths,factor)

Function indentParaboloidOne(localWaveForce,localWaveSep,starting,ending,basename,recordIndex)
    WAVE localWaveForce
    WAVE localWaveSep
    variable starting,ending
    string basename
    variable recordIndex
    variable dimension=ending-starting+1
    variable xPos,yPos
    WAVE/T namesandpaths
    WAVE numsandpaths
    Make/o extractedx1,extractedy1
    duplicate/o /R=[starting,starting+dimension-1] localwavesep extractedx1
    duplicate/o /R=[starting,starting+dimension-1] localwaveforce extractedy1
    extractedy1=extractedy1^2
    wavestats/q/z/m=1 extractedx1
    xPos=(V_min)
    wavestats/q/z/m=1 extractedy1
    yPos=(V_min)
    extractedx1-=xPos
    extractedy1-=yPos
    WAVE w_coef
    K0 = 0;K1 = 0;K2 = 0;
    CurveFit/q/n/H=\"1110\"/NTHR=0 poly 4, extractedy1 /X=extractedx1
Make /O/N=(dimension) $(namesandpaths[recordIndex][0]+"fitF")
Make /O/N=(dimension) $(namesandpaths[recordIndex][0]+"fitS")
WAVE fitF=$(namesandpaths[recordIndex][0]+"fitF")
WAVE fitS=$(namesandpaths[recordIndex][0]+"fitS")
f一时=k3*(extractedx1)^3
fitS=extractedx1
fitF=sqrt(fitF)
fitF+=yPos
fitS+=xPos
numsandpaths[recordIndex][11]=V_chisq
save/c/o/p=FC fitF, fitS
killwaves/z extractedx1, extractedy1, fitF, fitS
WAVEclear extractedx1, extractedy1, localWavesep, localwaveforce, fitF, fitS
return sqrt(K3)
End // Function indentParaboloidOne

Function indentSanzFeynman(localWaveForce, localWaveSep, starting, ending, returnCoeffs)
  WAVE localWaveForce
  WAVE localWaveSep
  variable starting, ending
  WAVE returnCoeffs
  variable dimension=ending-starting+1
  variable xPos, yPos
  Make/o extractedx2, extractedy2
  WAVE extractedx2, extractedy2

duplicate/o /R=[starting, starting+dimension-1] localwavesep extractedx2
duplicate/o /R=[starting, starting+dimension-1] localwaveforce extractedy2
wavestats/q/z/m=1 extractedx2
xPos=(V_min)
wavestats/q/z/m=1 extractedy2
yPos=(V_min)
extractedx2-=xPos
extractedy2-=yPos
Make/D/N=2/O W_coef
  W_coef[0] = {1, 1e-9}
  Make/O/T/N=2 T_Constraints
  T_Constraints[0] ={"K0 > 0","K1 > 0"}
  FuncFit/Q/NTHR=0 Sanz_Feynman W_coef extractedy2 /X=extractedx2 /D /C=T_Constraints
  returnCoeffs[0]=w_coef[0]
  returnCoeffs[1]=w_coef[1]
WAVEclear extractedx2, extractedy2, localWavesep, localwaveforce
End // Function indentSanzFeynman

Function quickHertz(w0, w) : FitFunc
  WAVE w0
  variable w
  //CurveFitDialog/ These comments were created by the Curve Fitting dialog. Altering them will
  //CurveFitDialog/ make the Function less convenient to work with in the Curve Fitting dialog.
  //CurveFitDialog/ Equation:
  //CurveFitDialog/ f(w) = d0+b*(w-w0)^1.5
  //CurveFitDialog/ End of Equation
  //CurveFitDialog/ Independent Variables 1
  //CurveFitDialog/ w
  //CurveFitDialog/ Coefficients 3
  //CurveFitDialog/ wo[0] = b
  //CurveFitDialog/ wo[1] = d0
9.1.4 Manual Intervention Functions

These functions are crucial to the analysis of force curves. They provide a means for users to enter critical information that pertains to the physics and chemical and physical parameters that are obtained from theory, from reference, or from empirical observation during experiments, and which primarily make batched automated analysis effective. Additionally, some functions described herein allow for manually changing values that were automatically calculated, if the user deems it necessary. This would be applicable in two cases: 1) when a value to be subsequently used in further analysis is erroneous and propagates the error; 2) when the final value for reporting is erroneous and needs to be manually adjusted.

9.1.4.1 IGOR Pro Code

Function changeConstants()
    Make/o/N=20 constants
    Make/o/N=10 textConstants
    constants[0]=25 // radius in nm
    constants[1]=35 // semi-vertical angle in degrees
    constants[2]=0.5 // Poisson ratio
    constants[3]=0.5 // factor to accept more curves default:0.8
    constants[4]=30 // how far leftward from point B (breakthrough entrance) do
                     // we search for a contact point (in nm) and for the separation
143

```plaintext
constants[5] = 3  // amount of iterations
constants[6] = 1.15 // universal sensitivity correction (nm Def/ nm ZED)
constants[7] = 0.1 // desired distance resolution in nm
constants[8] = 50 // how many waves to do a random pre-check on.
constants[9] = -400 // lowerbound ret force histogram [pN]
constants[10] = 200 // upperbound ret force histogram [pN]
constants[12] = 400 // index leftcutoff for plateaus
constants[13] = 700 // index rightcutoff for plateaus

constants[0] *= 1e-9
constants[1] *= pi/180
constants[4] *= 1e-9
constants[7] *= 1e-9
constants[9] *= 1e-12
constants[10] *= 1e-12

textConstants[0] = "LRU"
// Choices of Force Map Progression: "LRD" : Left->Right, Down, Left->Right,...
// "LRU" : Left->Right, Up, Left->Right,...
// "RLD" : R->L, Down, R->L,...
// "RLU" : R->L, Down, R->L,...
End // Function changeConstants()

Function correctA(index, whichPoint)
  variable index, whichPoint
  WAVE/T namesandpaths
  WAVE numsandpaths
  correct(2, index, whichPoint, namesandpaths, numsandpaths)
  WAVEclear namesandpaths, numsandpaths
  return 0
End // Function correctA

Function correctB(index, whichPoint)
  variable index, whichPoint
  WAVE/T namesandpaths
  WAVE numsandpaths
  correct(1, index, whichPoint, namesandpaths, numsandpaths)
  WAVEclear namesandpaths, numsandpaths
  return 0
End // Function correctB

Function correctC(index, whichPoint)
  variable index, whichPoint
  WAVE/T namesandpaths
  WAVE numsandpaths
  correct(0, index, whichPoint, namesandpaths, numsandpaths)
  WAVEclear namesandpaths, numsandpaths
  return 0
End // Function correctC

Function correct(whichOne, index, whichPoint, namesandpaths, numsandpaths)
  variable whichOne, index, whichPoint
  WAVE/T namesandpaths
  WAVE numsandpaths
  variable rowIndex
```

string toexecute="findvalue /v=\"+num2str(index)+\" numsandpaths"
exetc toexecute
NVAR v_value
if (v_value===-1|v_value>dimsize(numsandpaths,0))
    print "Index find failed, Graph# ", index," not found"
else
    string localWavePath=namesAndPaths[v_value][0]; // string-ize current namesAndPaths entry
    rowIndex=(v_value)
endif
loadwave /o/q/p=FC localWavePath+\"extF\"
loadwave /o/q/p=FC localWavePath+\"extS\"
loadwave /o/q/p=FC localWavePath+\"retF\"
loadwave /o/q/p=FC localWavePath+\"Peaks\"

WAVE localWaveForce=$(localWavePath+\"extF\"); // local "pointer" to the retract stuff.
WAVE localWaveSep=$(localWavePath+\"extS\");
WAVE localWaveRForce=$(localWavePath+\"retF\");
WAVE forcePeaks=$(localWavePath+\"Peaks\"");

if (dimsize(forcepeaks,0)==3)
switch(whichOne)
case 0:
    print "Replacing ",localWavePath,"'s breakthru exit with point: ",
    print whichPoint,"(Sep,Force)=",(localWaveSep[whichPoint]
    print ", ",localWaveForce[whichPoint],")
    break
case 1:
    print "Replacing ",localWavePath,"'s breakthru entrance with point: ",
    print whichPoint,"(Sep,Force)=",(localWaveSep[whichPoint]
    print ", ",localWaveForce[whichPoint],")
    break
case 2:
    print "Replacing ",localWavePath,"'s contact point with point: ",
    print whichPoint,"(Sep,Force)=",(localWaveSep[whichPoint]
    print ", ",localWaveForce[whichPoint],")
    break
default:
    print "Something wrong, doing nothing."
    break
endswitch
if (whichOne==0||whichOne==1||whichOne==2)
forcepeaks[whichOne][0]=localWaveSep[whichPoint]
forcepeaks[whichOne][1]=localWaveForce[whichPoint]
forcepeaks[whichOne][2]=whichPoint

namesAndPaths[rowIndex][3]=\"Indent_OK\"
namesAndPaths[rowIndex][4]=\"Thickness\"
namesAndPaths[rowIndex][5]=\"Breakthrough\"
namesAndPaths[rowIndex][6]=\"Adhesion\"
numsAndPaths[rowIndex][3]=forcePeaks[Dimsize (forcePeaks,0)-2][0]-
forcePeaks[Dimsize (forcePeaks,0)-1][0]/Indents
numsAndPaths[rowIndex][4]=forcePeaks[Dimsize (forcePeaks,0)-3][0]-
forcePeaks[Dimsize (forcePeaks,0)-1][0]; //Thickness
numsAndPaths[rowIndex][5]=forcePeaks[Dimsize (forcePeaks,0)-2][1]-
forcePeaks[Dimsize (forcePeaks,0)-1][1]; //Brkthrough Forces
numsAndPaths[index][6]=AdhGetter(localWaveRForce)
9.1.5 Graphing Functions

These functions pertain to the display of force curves for intermediate and final reporting purposes. Intermediate display refers to low-quality images of force curve which report elastic modulus, and position of critical points, which may be of use in determining the accuracy of the automated analysis. A quick manual scan of several hundred (randomized selection) low-quality images saves time overall, especially when there is the possibility that an entire force curve batch (>1000 curves) will be analyzed with incorrect input parameters. Other graphs generated include histograms and scatter plots, and re-created force volume maps as a contour plot, or color image.

9.1.5.1 IGOR Pro Code

Function killGraphs(namesAndPaths)
    WAVE namesAndPaths
    variable dimension=DimSize(namesAndPaths,0)
    variable index;
    string name
    for (index=0;index<dimenion;index+=1)
        name="graph"+num2str(index)
        killwindow $(name)
    endfor
    WAVEClear namesAndPaths
End // Function killGraphs(namesAndPaths)

Function contour ()
    WAVE numsAndPaths
    Make /o/n=(DimSize(numsAndPaths,0),3) AdhXYZ
    Make /o/n=(DimSize(numsAndPaths,0),3) IndentXYZ
    Make /o/n=(DimSize(numsAndPaths,0),3) WidthXYZ
    Make /o/n=(DimSize(numsAndPaths,0),3) ForceXYZ
    AdhXYZ[[0]]=numsAndPaths[p][1]
    AdhXYZ[[1]]=numsAndPaths[p][2]
    AdhXYZ[[2]]=numsAndPaths[p][6]
Display;AppendXYZContour AdhXYZ;DelayUpdate
ModifyContour AdhXYZ ctabLines={*,*,YellowHot,0},labels=0
ColorScale/C/N=text1/F=0/A=RC/E contour=AdhXYZ
ColorScale/C/N=text1" Adhesion Forces [N]"
IndentXYZ[[0]]=numsAndPaths[p][1]
IndentXYZ[[1]]=numsAndPaths[p][2]
IndentXYZ[[2]]=numsAndPaths[p][3]

Display;AppendXYZContour IndentXYZ;DelayUpdate
ModifyContour IndentXYZ ctabLines={*,*,YellowHot,0},labels=0
ColorScale/C/N=text1/F=0/A=RC/E contour=IndentXYZ
ColorScale/C=N=text1" Indentation Lengths [m]"
WidthXYZ[[0]]=numsAndPaths[p][1]
WidthXYZ[[1]]=numsAndPaths[p][2]
WidthXYZ[[2]]=numsAndPaths[p][4]

Display;AppendXYZContour WidthXYZ;DelayUpdate
ModifyContour WidthXYZ ctabLines={*,*,YellowHot,0},labels=0
ColorScale/C/N=text1/F=0/A=RC/E contour=WidthXYZ
ColorScale/C=N=text1" Bilayer Thicknesses [m]"
ForceXYZ[[0]]=numsAndPaths[p][1]
ForceXYZ[[1]]=numsAndPaths[p][2]
ForceXYZ[[2]]=numsAndPaths[p][5]
Display;AppendXYZContour ForceXYZ;DelayUpdate
ModifyContour ForceXYZ ctabLines={*,*,YellowHot,0},labels=0
ColorScale/C/N=text1/F=0/A=RC/E contour=ForceXYZ
ColorScale/C=N=text1"Breakthrough Force [N]"
End // Function contour

Function image ()
WAVE numsAndPaths

Display numsAndPaths[][2] vs numsAndPaths[][1]
ModifyGraph zColor(numsAndPaths)= {numsAndPaths[*][6],*,*,Grays,0}
ModifyGraph mode=3,marker=16
ColorScale/C/N=text1/F=0/A=RC/E trace=numsAndPaths
ColorScale/C=N=text1" Adhesion Forces [N]"

Display numsAndPaths[][2] vs numsAndPaths[][1]
ModifyGraph zColor(numsAndPaths)= {numsAndPaths[*][3],*,*,Grays,0}
ModifyGraph mode=3,marker=16
ColorScale/C/N=text1/F=0/A=RC/E trace=numsAndPaths
ColorScale/C=N=text1" Indentation Lengths [m]"

Display numsAndPaths[][2] vs numsAndPaths[][1]
ModifyGraph zColor(numsAndPaths)= {numsAndPaths[*][4],*,*,Grays,0}
ModifyGraph mode=3,marker=16
ColorScale/C/N=text1/F=0/A=RC/E trace=numsAndPaths
ColorScale/C=N=text1"Bilayer Thicknesses [m]"

Display numsAndPaths[][2] vs numsAndPaths[][1]
ModifyGraph zColor(numsAndPaths)= {numsAndPaths[*][5],*,*,Grays,0}
ModifyGraph mode=3,marker=16
ColorScale/C/N=text1/F=0/A=RC/E trace=numsAndPaths
ColorScale/C=N=text1"Breakthrough Force [N]"
Display numsAndPaths[[2]] vs numsAndPaths[[1]]
ModifyGraph zColor(numsAndPaths)={numsAndPaths[[*]][14],*,*,Grays,0}
ModifyGraph mode=3,marker=16
ColorScale/C/N=text1/F=0/A=RC/E trace=numsAndPaths
ColorScale/C/N=text1"Young's Modulus (Paraboloidal [Pa])"
End // Function image

Function realImage (X_dim,Y_dim)
variable X_dim
variable y_dim
WAVE numsAndPaths
variable dimension=dimsize(numsandpaths,0)

imagestats /M=1/G={0,dimension-1,1,1} numsandpaths
variable X_not=V_min
variable X_delta=(V_max-X_not)/(X_dim-1)
imagestats /M=1/G={0,dimension-1,2,2} numsandpaths
variable y_not=V_min
variable y_delta=(V_max-y_not)/(y_dim-1)
Make/o/n=(X_dim,Y_dim) AdhFRealImage
setscale/P x,0,x_delta,"m", AdhFRealImage
setscale/P y,0,y_delta,"m", AdhFRealImage
setscale/P z,0,0,"N", AdhFRealImage

Duplicate /o AdhFRealImage YoungMRealImage
Duplicate /o AdhFRealImage BrkFRealImage
Duplicate /o AdhFRealImage ThickRealImage

setscale/P z,0,0,"m", ThickRealImage
setscale/P z,0,0,"Pa", YoungMRealImage

variable index,x_index,y_index
for (index=0;index<dimension;index+=1)
    x_index=round((numsandpaths[index][1]-X_not)/x_delta)
    y_index=round((numsandpaths[index][2]-y_not)/y_delta)
    AdhFRealImage[x_index][y_index]=numsandpaths[index][6]
    BrkFRealImage[x_index][y_index]=numsandpaths[index][5]
    YoungMRealImage[x_index][y_index]=numsandpaths[index][14]
    ThickRealImage[x_index][y_index]=numsandpaths[index][4]
endfor

newimage/F adhfrealimage
ModifyGraph height={Aspect,1},width=250,margin=36, margin(top)=20, margin(right)=80
ColorScale/C/F=0/N=text0 /A=RC/E=2/X=0/Y=0"Adhesion Force \UN"

newimage/F BrkFrealimage
ModifyGraph height={Aspect,1},width=250,margin=36, margin(top)=20, margin(right)=80
ColorScale/C/F=0/N=text0 /A=RC/E=2/X=0/Y=0"Breakthrough Force \UN"

newimage/F Thickrealimage
ModifyGraph height={Aspect,1},width=250,margin=36, margin(top)=20, margin(right)=80
ColorScale/C/F=0/N=text0 /A=RC/E=2/X=0/Y=0"Bilayer Thickness \Um"

newimage/F YoungMrealimage
ModifyGraph height={Aspect,1},width=250,margin=36, margin(top)=20, margin(right)=80
ColorScale/C/F=0/N=text0 /A=RC/E=2/X=0/Y=0"Young's Modulus \UPa"
End // Function realimage
Function histo()
  if (dimsize(deltaPeakForces,0))
    Make/N=50/O DeltaPeakForces_Hist;DelayUpdate;
    Histogram/B=1 DeltaPeakForces,DeltaPeakForces_Hist;DelayUpdate;
    Setscale y 0.0,"Counts",DeltaPeakForces_Hist
    Display DeltaPeakForces_Hist; ModifyGraph mode=5,bbFill=2;
    TextBox/C/N=text0/F=0/A=MT "Histogram of Breakthrough Forces";
  endif
  if (dimsize(deltaPeakWidths,0))
    Make/N=50/O DeltaPeakWidths_Hist;DelayUpdate
    Histogram/B=1 DeltaPeakWidths,DeltaPeakWidths_Hist;DelayUpdate;
    Setscale y 0.0,"Counts",DeltaPeakWidths_Hist
    Display DeltaPeakWidths_Hist
    ModifyGraph mode=5,bbFill=2
    TextBox/C/N=text0/F=0/A=MT "Histogram of Bilayer Thicknesses"
  endif
  if (dimsize(deltaPeakIndents,0))
    Make/N=50/O DeltaPeakIndents_Hist;DelayUpdate
    Histogram/B=1 DeltaPeakIndents,DeltaPeakIndents_Hist;DelayUpdate;
    Setscale y 0.0,"Counts",DeltaPeakIndents_Hist
    Display DeltaPeakIndents_Hist
    ModifyGraph mode=5,bbFill=2
    TextBox/C/N=text0/F=0/A=MT "Histogram of Indentation Lengths"
  endif
  if (dimsize(deltaPeakAdhs,0))
    Make/N=50/O DeltaPeakAdhs_Hist;DelayUpdate
    Histogram/B=1 DeltaPeakAdhs,DeltaPeakAdhs_Hist;DelayUpdate;
    Setscale y 0.0,"Counts",DeltaPeakAdhs_Hist
    Display DeltaPeakAdhs_Hist
    ModifyGraph mode=5,bbFill=2
    TextBox/C/N=text0/F=0/A=MT "Histogram of Adhesion Forces"
  endif
  if (dimsize(paraboloidalfit,0))
    Make/N=50/O paraboloidalfit_Hist;DelayUpdate
    Histogram/B=1 paraboloidalfit,paraboloidalfit_Hist;DelayUpdate;
    Setscale y 0.0,"Counts",paraboloidalfit_Hist
    Display paraboloidalfit_Hist
    ModifyGraph mode=5,bbFill=2
    TextBox/C/N=text0/F=0/A=MT "Histogram of Young's Moduli via Paraboloidal Fit"
  endif
End //Function histo

Function scatter()
  display deltaPeakForces vs deltaPeakAdhs;
  ModifyGraph mode=2;
  Label left "Breakthrough Force [\U]";
  Label bottom "Adhesion Force [\U]"
  display deltaPeakIndents vs deltaPeakAdhs;
  ModifyGraph mode=2;
  Label left "Indentation [\U]";
  Label bottom "Adhesion Forces [\U]"
  display deltaPeakWidths vs deltaPeakAdhs;
ModifyGraph mode=2;
Label left "Thickness \[\mu\U\]"
Label bottom "Adhesion Force \[\mu\U\]"
display deltaPeakForces vs deltaPeakWidths;
ModifyGraph mode=2;
Label left "Breakthrough Force \[\mu\U\]"
Label bottom "Thickness \[\mu\U\]"
display deltaPeakIndents vs deltaPeakWidths;
ModifyGraph mode=2;
Label left "Indentation \[\mu\U\]"
Label bottom "Thickness \[\mu\U\]"
display deltaPeakIndents vs deltaPeakForces;
ModifyGraph mode=2;
Label left "Indentation \[\mu\U\]"
Label bottom "Breakthrough Forces \[\mu\U\]"
End //Function scatter

Function graphAll(namesAndPaths)
    WAVE/T namesAndPaths //local access to namesAndPaths wave, again
    WAVE numsAndPaths
    string localWavePath
    variable index=0
    string listStr=IndexedFile(fc,-1,".ibw")
    for (index=0;index<DimSize(namesAndPaths,0);index+=1)
        localWavePath=namesAndPaths[index][0]
        graphOne(localWavepath,listStr)
        saveOne(localwavepath)
        removeOne(localwavepath)
    endfor
End // Function graphAll

Function graphIndex(index)
    variable index
define index
    WAVE T namesandpaths
    WAVE numsandpaths
    string toexecute="findvalue /v=\"+num2str(index)+\" numsandpaths"
    execute toexecute
    NVAR v_value
    string listStr=indexedFile(fc,-1,".ibw")
    if (v_value==-1||v_value>dimsize(numsandpaths,0))
        print "Graphing failed, JPKIndex: ", index," not found"
    else
        graphone(namesandpaths[v_value][0],listStr)
    endif
    WAVEClear namesAndPaths,numsandpaths
End // Function graphIndex

Function graphRange(indexstart,indexend)
    variable indexstart,indexend
    variable index
    WAVE numsandpaths
    WAVE T namesandpaths
    string listStr=indexedFile(fc,-1,".ibw")
    for (index=indexstart;index<indexend+1;index+=1)
        string toexecute="findvalue /text=strNumwithZeros(\"+num2str(index)+\"") namesandpaths"
execute to execute
NVAR v_value
if (v_value==-1||v_value>dimsize(numsandpaths,0))
    print "Graphing failed, JPKIndex: ", index, " not found"
else
    graphone(namesandpaths[v_value][0],listStr)
endif
endfor
WAVE Clear namesAndPaths, numsandpaths
End // graphRange(indexstart, indexend)

Function graphOne(name, listStr)
    string name, listStr
    string nameVname
    variable index
    if (cmpstr(listStr,"")==0)
        listStr=IndexedFile(fc,-1,".ibw")
    endif
    variable waveCount=-1
    variable val2, val=0
    if(doesIBWexist("fc",name+"extF",listStr) && doesIBWexist("fc",name+"extS",listStr))
        loadwave /q/o/p=FC name+"extF"
        loadwave /q/o/p=FC name+"extS"
        nameVname="Display "+name +"extF"+" vs "+name +"extS"
        Execute nameVname
        waveCount+=1
        Modifygraph/z mode[waveCount]=0
        Modifygraph/z rgb[waveCount]=(0,0,0)
    endif

    if(doesIBWexist("fc",name+"Peaks",listStr))
        loadwave /q/o/p=FC name+"Peaks"
        nameVname="AppendtoGraph "+name +"Peaks[[1]]"+" vs "+name +"Peaks[[0]]"
        Execute nameVname
        waveCount+=1
        Modifygraph/z mode[waveCount]=3
        Modifygraph/z marker[waveCount]=8
        Modifygraph/z rgb[waveCount]=(65535,0,0)
        WAVE w=$(name +"Peaks")
        val=w[1][2]
        WAVE clear w
    endif

    if(doesIBWexist("fc",name+"fitF",listStr) && doesIBWexist("fc",name+"fitS",listStr))
        loadwave /q/o/p=FC name+"fitF"
        loadwave /q/o/p=FC name+"fitS"
        nameVname="AppendtoGraph "+name +"fitF"+" vs "+name +"fitS"
        Execute nameVname
        waveCount+=1
        Modifygraph/z mode[waveCount]=0
        Modifygraph/z rgb[waveCount]=(0,35535,0)
        Modifygraph/z lsize[waveCount]=2
    endif

    if(doesIBWexist("fc",name+"retF",listStr) && doesIBWexist("fc",name+"retS",listStr))
        loadwave /q/o/p=FC name+"retF"
        loadwave /q/o/p=FC name+"retS"
        nameVname="AppendtoGraph "+name +"retF"+" vs "+name +"retS"
Execute nameVname
waveCount+=1
Modifygraph/z mode[waveCount]=0
Modifygraph/z rgb[waveCount]=(0,0,35535)
endif

if(0&&!doesIBWexist("fc",name+"F_smth_boxDIF",listStr)&&!doesIBWexist("fc",name+"extS",listStr))
    loadwave /q/o/p=FC name+"F_smth_BoxDIF"
    nameVname="AppendtoGraph/R "+name +"F_smth_BoxDIF"+" vs "+name+"extS"
    Execute nameVname
    waveCount+=1
    Modifygraph/z mode[waveCount]=0
    Modifygraph/z rgb[waveCount]=(0,0,35535)
endif

WAVE numsandpaths
Execute "ShowInfo"
Execute "GraphStyle4()"

string toexecute="findvalue /v="+num2str(returnindex(name))+" numsandpaths"
execute toexecute
NVAR v_value
if (v_value==-1||v_value>dimsize(numsandpaths,0))
    print "Graphing failed, JPIndex: ", name," not found"
else
    sprintf nameVname,"Elastic Modulus: %.2W1PPa",numsandPaths[v_value][14]
    TextBox/C/B=1/A=LC/N=text1/F=0 "\Zr100\F'Arial'"+nameVname
endif
TextBox/C/B=1/A=MT/N=text0/F=0 "\Zr120\F'Arial Black'"+name;
End // Function graphOne(name,listStr)

Function saveOne(name)
    string name
    name+=".jpg"
    SavePICT/O/P=pics/E=-6/RES=72 as name;
    DoWindow/K Graph0;
End // Function saveOne

Function removeOne(name)
    string name
    WAVE localWaveForce=$(name+"extF")
    WAVE localWaveSep=$(name+"extS")
    WAVE forcepeaks=$(name+"Peaks")
    WAVE fitF=$(name+"fitF")
    WAVE fitS=$(name+"fitS")
    WAVE retF=$(name+"retF")
    WAVE retS=$(name+"retS")
    killwaves/z localWaveForce, localWaveSep, forcepeaks,fitF,fitS,retF,retS
End // Function saveOne

Function graphtoSingleE (range_min,range_max,total_count)
    variable range_min, range_max,total_count
    graphtoSingle(range_min,range_max,total_count,"ext")
End // Function graphtoSingleE
Function graphtoSingleR (range_min, range_max, total_count)
    variable range_min, range_max, total_count
    graphtoSingle(range_min, range_max, total_count, "ret")
End // Function graphtoSingleR

Function graphtoSingle(range_min, range_max, total_count, suffix)
    variable range_min, range_max, total_count
    string suffix
    range_min*=1e-9
    range_max*=1e-9
    variable index
    WAVE numsandpaths
    WAVE/T namesandpaths
    Make/o/T/N=0 names
    variable flag=0
    variable randNum
    string name, nameVname
    for (index=0; index<dimsize(numsandpaths, 0); index+=1)
        if ((numsandpaths[index][5] >= range_min) && (numsandpaths[index][5] <= range_max) && (numsandpaths[index][3] > 0))
            Redimension /N=(Dimsize(names, 0)+1) names
            names[Dimsize(names, 0)]=namesandpaths[index][0]
        endif
    endfor
    if (Dimsize(names, 0)<=total_count)
        Make/T/o/N=(Dimsize(names, 0)) randomList
        randomList=names
    else
        Make/T/o/N=(total_count) randomList
        for (index=0; index<total_count; index+=1)
            randNum=abs(round(enoise(Dimsize(names, 0))))
        // this is a poor but quick and dirty randomizer
            randomList[index]=names[randNum]
            deletepoints randNum, 1, names
        endfor
    endif

    for (index=0; index<dimsize(randomList, 0); index+=1)
        print index, randomlist[index]
        if (flag==0)
            flag=1
            name=randomList[index]
            loadwave /q/o/p=FC name+suffix+"F"
            loadwave /q/o/p=FC name+suffix+"S"
            nameVname="Display "+name +suffix+"F";" vs "+name+suffix+"S"
            Execute nameVname
        else
            name=randomList[index]
            loadwave /q/o/p=FC name+suffix+"F"
            loadwave /q/o/p=FC name+suffix+"S"
            nameVname="AppendtoGraph "+name +suffix+"F";" vs "+name+suffix+"S"
            Execute nameVname
        endif
    KBColorizeTraces( 0.5,1,0)
    endfor
    killwaves/Z names, randomList
    WAVEClear namesAndPaths, numsandpaths, names, randomList
End // Function graphtoSingle(range_min, range_max, total_count, suffix)

Function histoplateau()
    changeconstants()
    WAVE/T namesAndPaths // local access to namesAndPaths wave, again
    WAVE numsAndPaths
    WAVE statistics
    string localWavePath
    WAVE constants
    variable lowerBoundF=constants[9]
    variable upperBoundF=constants[10]
    variable resF=constants[11]

    variable index_outer, index_inner, dimension, placement
    variable namesAndPathsDim=DimSize(namesAndPaths, 0)
    variable bins=floor((upperBoundF-lowerBoundF)/resF)
    make/o/n=(bins) plateauWave
    WAVE plateauWave
    plateauwave=0
    setscale d,0,0,"counts" plateauWave
    setscale x,lowerboundF,upperBoundF,"N" plateauWave

    for (index_outer=constants[12]; index_outer<constants[13]; index_outer+=1)
        // cycle through each name in namesAndPaths list.
        localWavePath=namesAndPaths[index_outer][0]; // string-ize current namesAndPaths entry
        loadwave /o/q/p=FC localWavePath+"retF"
        WAVE localWaveRForce=$(localWavePath+"retF");

        dimension=dimsize(localWaveRForce, 0)
        wavestats /q/z/r=[0,20] localWaveRForce
        for (index_inner=0; index_inner<dimension; index_inner+=1)
            placement=round((localWaveRForce[index_inner]-lowerBoundF)/resF)
            if (placement>-1)&&(placement<dimension)
                plateauWave[placement]+=1
            endif
        endfor
        killwaves/z localWaveRForce
    endfor
    DeletePoints bins-1, 1, plateauWave
End // Function histoplateau

9.1.6 File I/O Functions

These functions pertain to loading the universal force curves to and from the disk, as they were sequentially analyzed. The serial analysis of force curves required the serial loading and unloading of files from disk to/from memory. This was done so that both high and low-
resolution force maps can be analyzed from one code, without concern for low memory (RAM) errors and crashes.

9.1.6.1 IGOR Pro Code

Function saveToDisk()
    string pathname=fileLoad(""")
    save/c/o/p=results constants
    save/c/o/p=results statistics
    save/c/o/p=results statisticsStr
    save/c/o/p=results namesAndPaths
    save/c/o/p=results numsAndPaths
    save/c/o/p=results subnumsAndPaths
    save/c/o/p=results subnamesAndPaths
    save/c/o/p=results firstPassNaPs
    save/c/o/p=results DeltaPeakForces
    save/c/o/p=results DeltaPeakWidths
    save/c/o/p=results DeltaPeakIndents
    save/c/o/p=results DeltaPeakAdhs
    save/c/o/p=results paraboloidalFit
End // Function saveToDisk()

Function loadFromDisk()
    string pathname=fileLoad(""")
    loadwave/q/o/p=results "constants"
    loadwave/q/o/p=results "statistics"
    loadwave/q/o/p=results "statisticsStr"
    loadwave/q/o/p=results "namesAndPaths"
    loadwave/q/o/p=results "numsAndPaths"
    loadwave/q/o/p=results "subnumsAndPaths"
    loadwave/q/o/p=results "subnamesAndPaths"
    loadwave/q/o/p=results "firstPassNaPs"
    loadwave/q/o/p=results "DeltaPeakForces"
    loadwave/q/o/p=results "DeltaPeakWidths"
    loadwave/q/o/p=results "DeltaPeakIndents"
    loadwave/q/o/p=results "DeltaPeakAdhs"
    loadwave/q/o/p=results "paraboloidalFit"
End // Function loadFromDisk()

Function/S fileLoad(pathName)
    string pathName
    string pathTemp
    if (strlen(pathName)==0) // If no path specified, create one
        NewPath/O temporaryPath // This will put up a dialog
        if (V_flag != 0)
            return "user cancelled" // User cancelled
        endif
        pathName = "temporaryPath"
    endif
    PathInfo temporaryPath
    pathTemp=S_path
    print "Creating fc directory"
NewPath/C/O/Q fc, pathTemp+"fc"
print "Creating pics directory"
NewPath/C/O/Q pics, pathTemp+"pics"
print "Creating results directory"
NewPath/C/O/Q results, pathTemp+"results"
return pathName
End //Function fileLoad(pathName)

9.1.7 Clean-up Functions

These functions were created to delete temporarily created files, or in-program data elements so that file directory structure, and memory usage, respectively is not bloated. This is not a minor issue, as around ten files are created per force curve to be analyzed, and a single directory may hold more than 40,000 files at some point before the automated analysis completes.

9.1.7.1 IGOR Pro Code

Function deletefiles(suffix)
  string suffix
  variable index=0
  string filename
  do // Loop through each file in folder
    fileName = IndexedFile(temporaryPath, index, suffix)
    if (strlen(fileName) == 0) // No more files?
      break // Break out of loop
    else
      print "to delete: ", filename
      endif
    index+=1
  while(1)
return 0
End // Function deletefiles(suffix)

Function rebuildQuantities()
  WAVE/T namesandpaths
  WAVE numsandpaths
  variable index
  Make/O/N=(0) $("DeltaPeakForces")
  Make/O/N=(0) $("DeltaPeakWidths")
  Make/O/N=(0) $("DeltaPeakIndents")
  Make/O/N=(0) $("DeltaPeakAdhs")
  WAVE deltaPeakForces=$("DeltaPeakForces")
  WAVE deltaPeakWidths=$("DeltaPeakWidths")
  WAVE deltaPeakIndents=$("DeltaPeakIndents")
  WAVE deltaPeakAdhs=$("DeltaPeakAdhs")
  Setscale d 0,0,"N",deltaPeakForces
Setscale d 0.0,"N",deltaPeakAdhs
Setscale d 0.0,"m",deltaPeakWidths
Setscale d 0.0,"m",deltaPeakIndents

for (index=0;index<dimsize(namesandPaths,0);index+=1)
    if ((cmpstr (namesandpaths[index][3],"Indent_Ok")==0)||(cmpstr (namesandpaths[index][3],"Indent_Intermediate")==0))
        Redimension /N=(Dimsize (deltaPeakIndents,0)+1) deltaPeakIndents
        Redimension /N=(Dimsize (deltaPeakWidths,0)+1) deltaPeakWidths
        Redimension /N=(Dimsize (deltaPeakForces,0)+1) deltaPeakForces
        Redimension /N=(Dimsize (deltaPeakAdhs,0)+1) deltaPeakAdhs
        deltaPeakIndents[Dimsize (deltaPeakIndents,0)]=numsandpaths[index][3]
        deltaPeakWidths[Dimsize (deltaPeakIndents,0)]=numsandpaths[index][4]
        deltaPeakForces[Dimsize (deltaPeakIndents,0)]=numsandpaths[index][5]
        deltaPeakAdhs[Dimsize (deltaPeakIndents,0)]=numsandpaths[index][6]
    endif
endfor

WAVEClear deltaPeakForces, deltaPeakWidths, deltaPeakIndents, deltaPeakAdhs
WAVEClear namesAndPaths,numsandpaths
End //Function rebuild Quantities ()

Function deleteIndex (index)
    variable index
    WAVE/T namesandpaths
    WAVE numsandpaths
    string toexecute="findvalue /v="+num2str(index)+" numsandpaths"
    execute toexecute
    NVAR v_value
    if (v_value==-1||v_value>dimsize(numsandpaths,0))
        print "Delete failed, JPK #",index,"not found"
    else
        print 'Deleting ... JPK wave# ", numsandpaths[v_value][0]
        deletepoints v_value,1, namesandpaths, numsandpaths
        endif
        rebuildquantities()
        WAVEClear namesAndPaths,numsandpaths
    End //Function deleteIndex (index)

Function killUselessIBW(namesAndPaths)
    WAVE/T namesandpaths
    string localwavepath
    variable index
    string listStr=IndexedFile(fc,-1,".ibw")
    print "Erasing useless ibw files on disk, wait a moment."
    for (index=0;index<dimSize(namesandpaths,0);index+=1)
        localWavePath=namesandpaths[index][0]
        if (doesIBWexist("fc",localwavepath+"F_smth_box",listStr) && \
            doesIBWexist("fc",localwavepath+"F_smth_boxDIF",listStr))
            loadwave /o/q/p=FC localWavePath+"F_smth_box"
            loadwave /o/q/p=FC localWavePath+"F_smth_boxDIF"
            WAVE localWave1=$localWavePath+"F_smth_box"
            WAVE localWave2=$localWavePath+"F_smth_boxDIF"
            killwaves /f/z localWave1
            killwaves /f/z localWave2
        endif
endfor
print "Done"
WAVEclear localWave1
End //killUselessIBW(namesAndPaths)

Function cleanup()
  string toExecute,list
  list=WinList("*",","WIN: 1")
  variable total=itemsinlist (list)
  if (total>0)
    do
      toExecute="killwindow 
      toExecute+=stringfromlist(total-1,list,";")
      execute toExecute
      total-=1
    while (total>0)
  else
    print "No graphs to delete"
  endif
  list=WaveList("w*ext*","","")
  list+=WaveList("w*ret*","","")
  list+=WaveList("w*peaks","","")
  list+=WaveList("w*fit*","","")
  total=itemsinlist (list)
  if (total>0)
    do
      toExecute="killwaves /z 
      toExecute+=stringfromlist(total-1,list,";")
      execute toExecute
      total-=1
    while (total>0)
  else
    print "No data to delete"
  endif

End //Function cleanup

Function pareDown(namesAndPaths,numsAndPaths)
  WAVE/T namesAndPaths
  WAVE numsAndPaths
  Duplicate /T/O namesAndPaths paredNamesAndPaths
  WAVE/T paredNamesAndPaths
  Duplicate /O numsAndPaths paredNumsAndPaths
  variable index=0
  do
    if (!cmpstr(parednamesAndPaths[index][3],"Indent_OK")||
       !cmpstr(parednamesAndPaths[index][3],""))
      DeletePoints  index, 1, paredNamesAndPaths;
      DeletePoints  index, 1, paredNumsAndPaths;
    else
      index+=1
    endif
  while ((index<DimSize(parednamesAndPaths,0)))
End // Function pareDown(namesAndPaths,numsAndPaths)
9.1.8 Helper Functions

This category contains frequently used functions that calculate an intermediate value, or perform an intermediate step, such as a Boolean check prior to executing follow-up actions. These may include: string-to-number conversions, calculation of a pixel density count, total size of data elements, etc. It also includes the code that can be used to analyze a random selection of force curves from the entire data set, so that: 1) a randomized subset can be selected to double check if input parameters are appropriate; 2) a randomized subset can be used to report a statistic without the need to execute the analysis of the entire force curve set.

9.1.8.1 IGOR Pro Code

Function returnFVPos(recordIndex,index,sampsLine,Lines,scanSize, numsAndPaths,xPosCol,yPosCol,FMapProgression)

    variable recordIndex,index,sampsLine,Lines,scanSize
    WAVE numsAndPaths
    variable xPosCol,yPosCol
    string FMapProgression
    variable xOffset=(scanSize/sampsLine)
    variable yOffset=(scanSize/Lines)
    variable dividend=floor(recordIndex/sampsLine)
    variable remainder=mod(recordIndex,sampsLine)
    variable x,y

    if (cmpstr(FMapProgression,"RLD")==0)
        numsAndPaths[index][yPosCol]=scansize-(0.5*yOffset+dividend*yOffSet)
        if (mod(dividend,2)==0) // if even
            numsAndPaths[index][xPosCol]=scansize-(0.5*xOffset+remainder*xOffSet)
        else // if odd
            numsAndPaths[index][xPosCol]=scansize-(0.5*xOffset+(sampsLine-remainder-1)*xOffSet)
    endif
    elseif (cmpstr(FMapProgression,"RLU")==0)
        numsAndPaths[index][yPosCol]=0.5*yOffset+dividend*yOffSet
        if (mod(dividend,2)==0) // if even
            numsAndPaths[index][xPosCol]=scansize-(0.5*xOffset+remainder*xOffSet)
        else // if odd
            numsAndPaths[index][xPosCol]=scansize-(0.5*xOffset+(sampsLine-remainder-1)*xOffSet)
    endif
    elseif (cmpstr(FMapProgression,"LRD")==0)
        numsAndPaths[index][yPosCol]=scansize-(0.5*yOffset+dividend*yOffSet)
        if (mod(dividend,2)==0) // if even
            numsAndPaths[index][xPosCol]=0.5*xOffset+remainder*xOffSet
        else // if odd
            numsAndPaths[index][xPosCol]=0.5*xOffset+remainder*xOffSet
numsAndPaths[index][xPosCol] = 0.5 * xOffset + (sampsLine - remainder - 1) * xOffSet

endif

else // assume FMapProgression=LRU
    numsAndPaths[index][yPosCol] = 0.5 * yOffset + dividend * yOffSet
    if (mod(dividend, 2) == 0) // if even
        numsAndPaths[index][xPosCol] = 0.5 * xOffset + remainder * xOffSet
    else // if odd
        numsAndPaths[index][xPosCol] = 0.5 * xOffset + (sampsLine - remainder - 1) * xOffSet
    endif
endif
End //Function returnFVPos()

Function nnIndex(localWave, greater_lesser, value, givenIndex)
    WAVE localWave
    string greater_lesser
    variable value
    variable givenIndex
    variable index = givenIndex
    variable increment = 1
    variable checkDifference
    if (localWave[0] > localWave[dimsize(localWave, 0)])
        // this is a decreasing Function
        increment = -1
    elseif (localWave[0] < localWave[dimsize(localWave, 0)])
        // this is an increasing Function
        increment = 1
    else
        print "Error in nnIndex Function"
        return index
    endif
    if (cmpstr(greater_lesser, ">") == 0)
        // returns the index(position) that is just greater than givenIndex by value
        checkDifference = localWave[givenIndex] + value
        do
            index += increment
            while((localWave[index] <= checkDifference) && index < dimsize(localWave, 0))
        elseif (cmpstr(greater_lesser, "<") == 0)
            // returns the index(position) that is less than givenIndex by value
            checkDifference = localWave[givenIndex] - value
            do
                index -= increment
                while((localWave[index] >= checkDifference) && index > -1)
        else
            print "ERROR in nnIndex Function"
        endif
    return index
End // Function nnIndex

Function randomizeNamesAndPaths(namesAndPaths, numsAndPaths, percent)
    WAVE/T namesAndPaths
    WAVE numsAndPaths
    variable percent
    Duplicate /T/O namesAndPaths randNamesAndPaths
    Duplicate /O numsAndPaths randNumsAndPaths
variable DIMENSION=DimSize(namesAndPaths,0)
variable newDIMENSION=round(DimSize(namesAndPaths,0)*percent/100)
variable chunkToCut=round(0.01*(100-percent)*DIMENSION/100)
if (chunkToCut<1)
    chunkToCut=1
endif
if (newDIMENSION==0)
    DIMENSION=1
endif
variable toCut=DIMENSION-newDIMENSION
if (toCut<0)
    toCut=0
endif
do
    variable candidate=abs(round(enoise(DIMENSION-1)))
    if (candidate<0)
    elseif (percent==100)
        break
    else
        if ((candidate+chunkToCut-1)>DIMENSION)
            DeletePoints candidate, 1, randNamesAndPaths;
            DeletePoints candidate, 1, randNumsAndPaths;
            toCut-=1
            DIMENSION-=1
        else
            DeletePoints candidate, chunkToCut, randNamesAndPaths;
            DeletePoints candidate, chunkToCut, randNumsAndPaths;
            DIMENSION-=chunkToCut
            toCut-=chunkToCut
        endif
    endif
while (toCut>0&&DIMENSION>0)
    WAVEClear namesAndPaths
End // Function randomizeNamesAndPaths

Function slopechange(index,slope)
    variable index,slope
    WAVE/T namesandpaths
    WaVE numsandpaths
    variable rowIndex
    string toexecute="findvalue /v=\"+num2str(index)+\" numsandpaths"
    execute toexecute
    NVAR v_value
    variable i
    if (v_value==-1||v_value>dimsize(numsandpaths,0))
        print "Index find failed, Graph# ", index," not found"
    else
        string localWavePath=namesAndPaths[v_value][0];
        // string-ize current namesAndPaths entry
        rowIndex=(v_value)
        loadwave /o/q/p=FC localWavePath+\"extD\"
        loadwave /o/q/p=FC localWavePath+\"extF\"
        loadwave /o/q/p=FC localWavePath+\"extS\"
        loadwave /o/q/p=FC localWavePath+\"extZ\"
        loadwave /o/q/p=FC localWavePath+\"retF\"
loadwave /o/q/p=FC localWavePath+"Peaks"
WAVE localWaveDef=$(localWavePath+"extD");  // local "pointer" to the retract stuff.
WAVE localWaveSep=$(localWavePath+"extS");
WAVE localWaveForce=$(localWavePath+"extF");
WAVE localWaveZed=$(localWavePath+"extZ");
WAVE localWaveRForce=$(localWavePath+"retF");
WAVE forcePeaks=$(localWavePath+"Peaks");
graphindex(index)
localWaveSep=localWaveZed-localWaveDef/slope
for (i=0;i<dimsize(forcepeaks,0);i+=1)
    forcepeaks[i][0]=localWaveSep[(forcepeaks[i][2])]
    forcepeaks[i][1]=localWaveForce[(forcepeaks[i][2])]
endfor
if (dimsize(forcepeaks,0)==3)
    namesAndPaths[rowIndex][3]="Indent_OK"
    namesAndPaths[rowIndex][4]="Thickness"
    namesAndPaths[rowIndex][5]="Breakthrough"
    namesAndPaths[rowIndex][6]="Adhesion"
    numsAndPaths[rowIndex][3]=forcePeaks[Dimsize (forcePeaks,0)-2][0]-
                           forcePeaks[Dimsize (forcePeaks,0)-1][0]; //Indents
    numsAndPaths[rowIndex][4]=forcePeaks[Dimsize (forcePeaks,0)-3][0]-
                           forcePeaks[Dimsize (forcePeaks,0)-1][0]; //Thickness
    numsAndPaths[rowIndex][5]=forcePeaks[Dimsize (forcePeaks,0)-2][1]-
                           forcePeaks[Dimsize (forcePeaks,0)-1][1]; //Brkthrough Forces
    numsAndPaths[index][6]=AdhGetter(localWaveRForce)
    rebuildquantities()
else
    print "Number of peaks isn't three, something wrong, doing nothing to the peaks."
endif
save/c/o/p=FC forcePeaks
killwaves/Z localWaveDef,localWaveSep,forcepeaks,localWaveZed,localwaveForce
killwaves/Z localWaveRForce
WAVEclear localwaveDef,localwavesep,namesandpaths,numsandpaths
WAVEclear localwaveForce,localwaveRForce
return 0
End // Function slopechange

Function/T strNumwithZeros(recordIndex)
variable recordindex
string zeros
if (recordIndex<10)  //string manipulation step, just adds placeholder 0's.
    zeros="000"
elseif (recordIndex<100)
    zeros="00"
elseif (recordIndex<1000)
    zeros="0"
else
    zeros=""
endif
zeros+=num2str(recordIndex)
return zeros
End // Function/T strNumwithZeros

Function difZero(difWave)
    WAVE difWave
    variable index
for (index=0;index<dimsize(difwave,0);index+=1)
    if (difWave[index]>0)
        difWave[index]=0
    endif
endfor
End // Function difZero(difWave)

Function pixelPerDist(localWaveSep,value)
    WAVE localWaveSep
    variable value
    variable answer
    wavestats/q/z localWaveSep
    if (V_max-V_min)
        answer=ceil(V_npnts/abs(V_max-V_min)*value)
    else
        answer=1
    endif
    variable dimension=ceil(0.1*dimsize(localWaveSep,0))
    answer=ceil(value*(dimension/abs(localWaveSep[0]-localWaveSep[dimension])))
    if (answer==0)
        answer=1
    endif
    return answer
End // pixelPerDist(localWaveSep,value)

Function returnIndex(name)
    string name
    variable index
    sscanf name, "w%d",index
    return index
End

Function doesIBWExist(pathName,fileName,listStr)
    string pathname,filename,listStr
    fileName+=".ibw"
    if (cmpstr(listStr,"")==0)
        listStr=IndexedFile($pathname,-1,".ibw")
    endif
    if (whichlistitem (filename,listStr)>-1)
        return 1
    else
        return 0
    endif
End //Function doesIBWExist(pathName,fileName)

9.1.9 Macros

This section contains macros that are used to call a succession of functions quickly, with the appropriate initialization of variables. These macros are displayed in the menu toolbar within
IGOR Pro, and are therefore more convenient for mouse-based repeated code execution by the user. The other alternative would be to type in the full commands in the command line prompt.

9.1.9.1 IGOR Pro Code

Macro cleanupMacro()
    cleanup()
End // Macro cleanupMacro()

Macro AFMLoadMacro ()
    changeconstants()
    print "pre-cleanup..."
    cleanup()
    string pathName=fileload(""")
    string baseName="w"
    Make/O/N=(4) statistics
    Make/T/O/N=(4) statisticsStr
    statisticsStr[0]="files_processed"
    statisticsStr[1]="breakthroughs_found"
    statisticsStr[2]="valid_indentations"
    //JPKLoad(pathName,basename,1)
    VeecoLoad(pathName,basename,1)
    //MFPLoad(pathName,baseName,1)

    secondpart()
    graphallMacro()
    savetoDisk()
End // Macro AFMLoadMacro ()

Macro PreCheckAFMLoadMacro ()
    changeconstants()
    print "PRECHECK:pre-cleanup..."
    cleanup()
    // Is a macro, initiates the loading of JPKLoad and BreakThroughPoint Functions
    string pathName=fileload(""")
    string baseName="w"

    Make/O/N=(4) statistics
    Make/T/O/N=(4) statisticsStr
    statisticsStr[0]="files_processed"
    statisticsStr[1]="breakthroughs_found"
    statisticsStr[2]="valid_indentations"

    JPKLoad(pathName,basename,0)
    //MFPLoad(pathName,baseName,0)

    BreakthruPointsBreakdown(namesAndPaths,numsAndPaths)
    print "PRECHECK: Files Processed: ", statistics[0]
    print "Breakthroughs found: ", statistics[1]
    print "Valid indentations: ", statistics[2]
Macro YoungsModFits()
    changeConstants()
    indentFitAll(namesAndPaths, numsAndPaths, 70)
End // Macro YoungsModFits()

Macro graphAllMacro()
    graphAll(namesAndPaths)
End // Macro graphAllMacro()

Macro secondpart()
    BreakthruPointsBreakdown(namesAndPaths, numsAndPaths)
    print "PRECHECK: Files Processed: ", statistics[0]
    print "Breakthroughs found: ", statistics[1]
    print "Valid indentations: ", statistics[2]
    YoungsmodFits()
    killUselessIBW(namesAndPaths)
End // Macro secondpart()

Macro plateauhistoMacro()
    histoplateau()
    display plateauWave
    ModifyGraph mode=5, hbFill=2
End // Macro plateauhistoMacro()

Proc GraphStyle4() : GraphStyle
    PauseUpdate; Silent 1   // modifying window...
    // ModifyGraph/Z mode[1]=3
    // ModifyGraph/Z marker[1]=8
    // ModifyGraph/Z mode[3]=0
    // ModifyGraph/Z Isystyle[3]=0
    // ModifyGraph/Z Isize[3]=2
    // ModifyGraph/Z rgb[0]=(0,0,0), rgb[1]=(65535,16385,16385), rgb[2]=(0,0,65535), rgb[3]=(0,35535,0)
    Label/Z left "Force [\U]"
    Label/Z bottom "Separation [\U]"
EndMacro // Proc GraphStyle4() : GraphStyle
9.1.10 Code Version History

Below is the version history of the code. The code evolved from other pieces that are outside the scope of this thesis, and hence there appears to be a jump in the numbering sequence of the versions. A code version system was not utilized for revision control, as there was only one programmer.

<table>
<thead>
<tr>
<th>Filename Version</th>
<th>Date</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>JPKLoad3a</td>
<td>&lt; 2009-03</td>
<td>• Transferred code with series version 2, from a different purpose (AFM retract curve analysis using Veeco AFM Instruments)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Adapted code to load JPK AFM Instrument files</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Fixed contact point finding to 2 line fits, rather than 3/2 power + line fits for accuracy, and speed.</td>
</tr>
<tr>
<td>JPKLoad3b</td>
<td>&lt; 2009-03</td>
<td>• try to see where I can implement WAVEclear ops.</td>
</tr>
<tr>
<td>JPKLoad3c</td>
<td>&lt; 2009-03</td>
<td>• Also add adhesion finding.</td>
</tr>
<tr>
<td>JPKLoad4a</td>
<td>&lt; 2009-03</td>
<td>• Remove &lt;1nm Indentations data sets, collect adhesion, indent, breakthrough force, breakthrough width and name as a set.</td>
</tr>
<tr>
<td>JPKLoad4b</td>
<td>&lt; 2009-03</td>
<td>• Positions, plus index name is correct</td>
</tr>
<tr>
<td>JPKLoad5a</td>
<td>&lt; 2009-03</td>
<td>• Requires fixing the name index.</td>
</tr>
<tr>
<td>JPKLoad7e</td>
<td>&lt; 2009-03</td>
<td>• Up to indentation</td>
</tr>
<tr>
<td>JPKLoad7g</td>
<td>&lt; 2009-03</td>
<td>• File reading /writing to disk rather than maintain all the files in memory.</td>
</tr>
<tr>
<td>JPKLoad7h</td>
<td>&lt; 2009-03</td>
<td>• Optimized version of 7g</td>
</tr>
<tr>
<td>JPKLoad7i</td>
<td>&lt; 2009-03</td>
<td>• Perform paraboloid/conical fits on curves without breakthrough forces.</td>
</tr>
<tr>
<td>JPKLoad7j</td>
<td>&lt; 2009-03</td>
<td>• Fixed some bugs regarding UBound/ LBound. Also need to fix: what happens when the cutoff condition is met, but there really is no breakthrough.</td>
</tr>
<tr>
<td>JPKLoad8a</td>
<td>2009-03-14</td>
<td>• Final form of JPK Loader</td>
</tr>
<tr>
<td>JPKLoad8b</td>
<td>2009-03-14</td>
<td>• Removed conical fit everywhere</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Took out graphAllIT</td>
</tr>
<tr>
<td>JPKLoad</td>
<td>Date</td>
<td>Notes</td>
</tr>
<tr>
<td>---------</td>
<td>------------</td>
<td>----------------------------------------------------------------------</td>
</tr>
<tr>
<td>8c</td>
<td>2009-03-15</td>
<td>• Put in multiple iterations</td>
</tr>
<tr>
<td>8d</td>
<td>2009-03-17</td>
<td>• Added correction of individual force peaks.</td>
</tr>
<tr>
<td>8e</td>
<td>2009-03-28</td>
<td>• Added &quot;Rebuild quantities&quot; to the correction in 8d.</td>
</tr>
<tr>
<td>8f</td>
<td>2009-04-01</td>
<td>• updated the numsandpaths index as well</td>
</tr>
<tr>
<td>8g</td>
<td>2009-04-02</td>
<td>• many good things</td>
</tr>
<tr>
<td>8h</td>
<td>2009-04-30</td>
<td>• minor improvements</td>
</tr>
<tr>
<td>8i</td>
<td>2009-04-30</td>
<td>• minor improvements</td>
</tr>
<tr>
<td>8gg</td>
<td>2009-05-12</td>
<td>• Double letter suffixes are offshoot of 8g used to address special</td>
</tr>
<tr>
<td></td>
<td></td>
<td>problems quickly for publication purposes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• generate its own pics/fe directories</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• fix to the curve fits</td>
</tr>
<tr>
<td>8gh</td>
<td>2009-05-14</td>
<td>• remove legend</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• add randomSampling graphtoSingle</td>
</tr>
<tr>
<td>8gi</td>
<td>2009-05-15</td>
<td>• many little patches</td>
</tr>
<tr>
<td>8gj</td>
<td>2009-05-18</td>
<td>• manual correction patch</td>
</tr>
<tr>
<td>8gk</td>
<td>2009-06-16</td>
<td>• made changes to use minPerPixel arguments.</td>
</tr>
<tr>
<td>9</td>
<td>2009-07-09</td>
<td>• made a bunch of changes to troubleshoot errors related to sparse</td>
</tr>
<tr>
<td></td>
<td></td>
<td>data.</td>
</tr>
<tr>
<td>9a</td>
<td>2009-07-09</td>
<td>• only performs a fit to factor distance between A and B.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• at the moment, made startA to be really the point found + minPixel,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>just in case.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• implemented Function &quot;nnIndex&quot; to return the index that is value</td>
</tr>
<tr>
<td></td>
<td></td>
<td>less/greater than a givenIndex.</td>
</tr>
<tr>
<td>9b</td>
<td>2009-08-19</td>
<td>• fixed a forever loop in nnIndex</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• made some code to circumvent bad curves, not 100% complete</td>
</tr>
<tr>
<td></td>
<td></td>
<td>though.</td>
</tr>
<tr>
<td>9c</td>
<td>2009-10-04</td>
<td>• added the paraboloid fit stuff on top of the printed curves</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• there might still be a problem with the x, y origin location,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>which right now</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• may be undetected because the position of (x,y) is near (0,0).</td>
</tr>
<tr>
<td>Date</td>
<td>Changes</td>
<td></td>
</tr>
<tr>
<td>------------</td>
<td>--------------------------------------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>MFPLoad9c</td>
<td>2009-11-11</td>
<td></td>
</tr>
<tr>
<td>JPKLoad9c</td>
<td>• converting JPKLoad to MFPLoad to handle Asylum Research AFM files</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Successful conversion to MFP load. Commented out peaks graphing to fix the problem of having no peaks.</td>
<td></td>
</tr>
<tr>
<td>JPKLoad9d</td>
<td>2009-11-23</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• added doesIBWExist, to handle absence of FitF, FitS, and Peaks when graphing ext.</td>
<td></td>
</tr>
<tr>
<td>AFMLoad1a</td>
<td>2009-11-30</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• converting to AFMLoad1a, fusing JPKLoad 9d and MFPLoad9d.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• try to verify that it is seamless.</td>
<td></td>
</tr>
<tr>
<td>AFMLoad1b</td>
<td>2009-12-01</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Make a saveToDisk and loadFromDisk</td>
<td></td>
</tr>
<tr>
<td>AFMLoad1c</td>
<td>2009-12-07</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• AFMLoad 1c: added a Function in JPKLoad to erase the ***.out files after we are done with them</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• procedure is in place in MFP load, but need to discuss with Ruby first if she really wants it.</td>
<td></td>
</tr>
<tr>
<td>AFMLoad1d</td>
<td>2010-01-18</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• added the precheck code, that runs on a specified number of curves first</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• files wont be erased when it does a pre-check, but still will when it doesn’t - for AFMLoad</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• at the moment, MFPLoad won’t erase anything.</td>
<td></td>
</tr>
<tr>
<td>AFMLoad1e</td>
<td>2010-01-18</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• added plateauhistogram</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• put bounds, so that more of the x-axis can be cut off.</td>
<td></td>
</tr>
<tr>
<td>AFMLoad1f</td>
<td>2010-01-25</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• added Real Z-Map Imaging.</td>
<td></td>
</tr>
<tr>
<td>AFMLoad1g</td>
<td>2010-06-28</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• added loading from DI AFM instrument</td>
<td></td>
</tr>
</tbody>
</table>
9.2 Appendix: Rupture of a Lipid Bilayer - Theory and Derivations

9.2.1 Rupture Activation Energy Calculation

We used the theory introduced by Butt et al. to calculate the activation energy of rupture of lipid membranes. This is based on the following works:


Details of the rupture kinetic derivation are presented in Section 9.2.2. In summary, we utilized the general relation between the breakthrough force of film rupture and the activation energy:

\[
\Delta E_a(F_0) = -k_B T \ln \left( \frac{0.693 k}{A} \frac{dv}{dF_0} \right)
\]

where \(\Delta E_a\) is the rupture activation energy as described earlier, \(F_0\) is the mean breakthrough force, \(k_B\) is Boltzmann’s constant, \(T\) is the Kelvin temperature, \(v\) is the loading rate, \(k\) is the cantilever spring constant, and \(A\) is assumed to be the frequency at which the tip attempts to penetrate the film. In the case where the distribution of mean breakthrough forces is relatively narrow (\(\Delta F/F_0 \ll 1\), where \(\Delta F\) is the half width of the yield force distribution), \(F_0\) can be obtained from the histogram of breakthrough forces; the loading rate demonstrates a logarithmic dependence on the breakthrough force, namely,

\[
F_0 = a + b \ln v
\]

where \(a\) and \(b\) are parameters obtained from fits to the experimental data, the dependence of activation energy on loading rate can be explicitly expressed as:

\[
\Delta E_a(F_0) = k_B T \left[ 2.30 \frac{a - F_0}{b} - \ln \left( \frac{1.60 k}{A b} \right) \right]
\]

\(\Delta E_a\) can then be written as a function of force that is proportional to \(k_B T\). When this relation is extrapolated to zero mean yield force (\(F_0 = 0\)), it provides the rupture activation energy of lipid membranes in the absence of an applied force.

In the force curve collection experiments, the mean breakthrough force, \(F_0\) was extracted from histograms compiled for each phase (\(L_d\) and \(L_o\)), at each loading rate (see Chapter 6). In our
calculations, we used a $k$ of ~0.25 N/m, a temperature of 296.2 K, and set $A$ equal to the cantilever’s resonant frequency under water ($\sim 15 \times 10^3$ Hz).

### 9.2.2 Rupture Energy Derivation

In this section, the derivation of the rupture activation energy is derived in greater detail.

Consider an ensemble of $N_0$ simultaneous, identical AFM tip-film indentation experiments occurring at a constant tip loading rate $v$. Let $N$ be the number of films that remain intact after time $t$ has elapsed. Time is referenced to $t=0$, the point at which tip-sample contact begins. The change in number of intact layers $dN$ after time $dt$ can be expressed as:

$$dN = -kN dt$$  \hspace{1cm} \text{Equation 9.4}

where $k$ is a time-dependent constant. Accounting for this, and dividing by $N_0$ to convert to probability $P$.

$$dP = -k(t)P dt$$  \hspace{1cm} \text{Equation 9.5}

Integrating,

$$\ln P(t) = -\int_0^t k(t') dt'$$  \hspace{1cm} \text{Equation 9.6}

Assuming that the rate constant is associated with an activated process following an Arrhenius Law:

$$k(t) = Ae^{\frac{-\Delta U(F')}{k_B T}}$$  \hspace{1cm} \text{Equation 9.7}

where $\Delta U$ is the activation energy necessary for the formation of a hole in the film that is large enough to initiate rupture and let the tip break through, $k_B$ is Boltzmann’s constant $T$ is Kelvin temperature, and $A$ is the frequency at which the tip attempts to penetrate the film.

Time is replaced with force, using its relation with loading rate

$$F = Kvt$$  \hspace{1cm} \text{Equation 9.8}

where $K$ is the tip spring constant. Combining Equation 9.6, Equation 9.7 and Equation 9.8:
\[
\ln P(F) = -\frac{A}{K_v} \int_0^F e^{\frac{\Delta U(F')}{{k_B}T}} dF'
\]

Equation 9.9

The probability distribution that this creates represents the distribution of yield forces. If this distribution is narrow, then at the mean yield force, the probability \( P(F) = 0.5 \).

With \( \ln(0.5) = -0.693 \), Equation 9.6 can be rewritten:

\[
v = \frac{A}{0.693K} \int_0^{F_0} e^{\frac{\Delta U(F)}{{k_B}T}} dF
\]

Equation 9.10

\[
\Delta U(F_0) = -{k_B}T \ln \left( \frac{0.693K}{A} \frac{dv}{dF_0} \right)
\]

Equation 9.11

where \( \Delta U \) is the activation energy, (this is Equation 9.1). If the \( v(F_0) \) relationship is known then its derivative can be calculated to obtain activation energy’s dependence on yield force. This is a useful equation, as \( v \) and \( F_0 \) are experimental observables.

Given that loading rate demonstrates a logarithmic dependence on the mean yield force, namely, (same as Equation 9.2).

\[
F_0 = a + b \ln v
\]

Equation 9.12

\( a \) and \( b \) can derived from geometric parameters and mechanics arguments but more importantly are parameters obtained from fits to experimental data, the dependence of activation energy on loading rate can be explicitly expressed as (Equation 9.3):

\[
\Delta U(F_0) = {k_B}T \left[ 2.30 \frac{a - F_0}{b} - \ln \left( \frac{1.60K}{Ab} v \right) \right]
\]

Equation 9.13

When this relation is extrapolated to zero mean yield force, it provides the intrinsic activation energy of the lipid membranes.

The experimental equivalent of this thought experiment is that the \( N_0 \) simultaneous, identical experiments are actually the set of force curves comprising the map, which although performed sequentially, and presumably are identical (at least true for same-phase force curves and
independent. The probability distribution of forces is the breakthrough force histogram, which shows bimodality corresponding to the two phases ($L_d$ and $L_0$), and at each loading rate.