Oliver Herbert Kress

The Viscoelastic Response of Liquid Crystalline Fibers Formed By Bent-core Molecules

From Microscopic Ordering to Macroscopic Behavior

Georg-August University

Göttingen, 2018

Dissertation

The Viscoelastic Response of Liquid Crystalline Fibers Formed By Bent-core Molecules

From Microscopic Ordering to Macroscopic Behavior

to acquire the degree of Doctor of Philosophy in physics

at the

Georg-August-University

and the

Max Planck Institute of Dynamics and Self-organisation

in

Göttingen, Germany

in the doctoral degree program

Physics of Biological and Complex Systems (PBCS)

of the

Göttingen Graduate School of Neurosciences, Biophysics, and Molecular Biosciences (GGNB)

at the

Georg-August University School of Sciences (GAUSS)

submitted by

Oliver Herbert Kress

from Sigmaringen, Germany

Göttingen, 2018

Thesis Committee

Dr. Tanya Ostapenko

Dynamics of Complex Fluids – Max Planck Institute for Dynamics and Self-Organization (MPIDS), Göttingen

Prof. Dr. Jörg Enderlein

Biophysics – Third Institute of Physics, Georg-August University, Göttingen

Prof. Dr. Simone Techert

Structural Dynamics of (Bio)Chemical Systems – Institute for X-Ray Physics, Georg-August University, Göttingen

Members of the Examination Board

Referee: *Dr. Tanya Ostapenko*

Dynamics of Complex Fluids – Max Planck Institute for Dynamics and Self-Organization (MPIDS), Göttingen

Co-referee: *Prof. Dr. Simone Techert*

Structural Dynamics of (Bio)Chemical Systems – Institute for X-Ray Physics, Georg-August University, Göttingen

Further Members of the Examination Board

Prof. Dr. Jörg Enderlein

Biophysics – Third Institute of Physics, Georg-August University, Göttingen

Prof. Dr. Eberhard Bodenschatz

Laboratory for Fluid Physics, Pattern Formation and Biocomplexity – Max Planck Institute for Dynamics and Self-Organization (MPIDS), Göttingen

Prof. Dr. Helmut Grubmüller

Theoretical and Computational Biophysics – Max Planck Institute for Biophysical Chemistry (MPIBPC), Göttingen

Prof. Dr. Stefan Klumpp

Institute for Nonlinear Dynamics – Georg-August University, Göttingen

Date of Defense

November 23, 2018

ABSTRACT

A free-standing fluid fiber must suppress the radial fluctuations described by the Rayleigh-Plateau (R-P) instability. The stability of such structures are typically restricted to polymers or naturally-occurring spider silk. The first exception was published in 1987 when Van Winkle *et al.* reported that thin films formed by discotic mesogens tended to collapse into thin, stable strands [1]. Nearly two decades passed before the field of liquid crystals reported its second example of free-standing fibers. Thanks to the work of Jákli *et al.* in 2003, it is now known that the ferroelectric SmCsPf (B7) phase formed by certain mesogens known as bent-core liquid crystals (BLCs) is capable of spontaneously forming stable microfibers when subjected to uniaxial extension [2]. These fibers stabilize against R-P fluctuations by means of a layered (smectic) structure which provides the necessary radially-stabilizing compression modulus [3]. While previous studies focused primarily on measuring the static tension of both single fibers and fiber bundles to better understand their structure and stability, their dynamic response was not investigated in convincing detail until a 2016 publication by Salili *et al.* reported on their rupture and recoil behavior [4, 5].

In the present work, specially designed glass capillaries, piezo positioning, a custom heat stage and specialized imaging and control software were combined to develop a compact, nano-extensional rheometer currently undergoing final European patent approval. Experiments to investigate the strain- and strain-rate dependent stress response and the induced structural (phase) changes of these fibers were conducted alongside theoretical work to gradually formulate a generalized viscoelastic model. An improved understanding of the interplay between the viscous and elastic responses of these fibers – both in the bulk and at the surface – will help to uncover ways in which their complex internal geometry can be manipulated by external stimuli such as electric fields or temperature fluctuations in order to achieve actuation in the form of controlled length contraction and dilation. In addition, polymer stabilization and the use of BLCs containing an azo group to enhance the stability and actuation behavior under UV irradiation, respectively, are under continuing investigation. It is expected that such soft, self-healing fiber actuators could provide a key advancement to the fields of robotics, biomedical and cybernetic engineering.

ACKNOWLEDGEMENTS

When thinking back on the many people in my life who directly, or even indirectly, contributed to this achievement I am humbled to know that I could not have done it alone. I would like to thank first and foremost my parents for relentlessly supporting me in the pursuit of my dreams. During high school, I found myself bored by almost every subject besides mathematics and physics. It was by the good graces of Dr. Jon Secaur, my high school physics teacher and a professor of physics at Kent State University who would later become my undergraduate advisor, that I found myself guided towards the study of physics. Once my journey began, I was blessed by kind and stimulating teachers throughout my undergraduate years.

After the completion of my bachelor's degree, I deviated from further studies for over a year during which I felt the need to pursue another calling. I applied to the FBI and began training myself physically instead. Although ferociously focusing on body rather than mind was uniquely rewarding, I realized that I needed to return to a intellectually challenging environment. It was at this point that I began my graduate studies at the Liquid Crystal Institute at Kent State University. I became fascinated by the advanced study of soft matter physics and the seemingly endless complexities of the field of liquid crystals. Having the opportunity to pursue a long-standing interest of developing artificial muscles, I began working with Prof. Antal Jákli for the better half of my two-year enrollment. This period was perhaps the best experience I have had in an academic setting. The LCI felt like a family and Prof. Jákli's kindness and generosity reminded me of Dr. Secaur's many years before. I had the privilege of spending the summer of 2014 in Germany during an internship studying bent-core fibers and was allowed to give a talk at the Ferroelectric Liquid Crystal conference in Prague in 2015.

Nearing the end of my master's studies, I found myself faced with another important decision; to continue in academia or again deviate in search of another path. The choice remained nebulous until suddenly, through the collaboration of Prof. Jákli and his former doctoral advisee Dr. Ostapenko who had recently begun a post-doc in Göttingen, Germany, I received an offer to move to Germany and start doctoral studies at the Max Planck Institute. Immediately recognizing a once-in-a-lifetime opportunity, I quickly prepared for my new life. Having recently been overcome with a fear of flying, I packed my things and drove to New York where I embarked on an seven-day trans-Atlantic voyage aboard the Queen Mary II. After arriving at the port of Southampton, England, catching a ferry to Normandy, France, picking up a rental car and driving 1000 km to Germany, I finally arrived in Göttingen. My cat, however, had arrived 7 days prior and was being kept safe at the home of my new advisor, Dr. Ostapenko, to whom I am forever indebted.

Now, three years later, I must think back on all those who made my time here in Germany one of the happiest periods of my life. I deeply thank the many lifelong friends I have made here (in no particular order): Christian Kreis, Marcin Makowski, Julien Petit, Laura Turco, Hakam Agha, Marco Rivetti, Babak Vajdi, Kyle Baldwin, Kris Hantke, Marco Mazza, and countless others including my neighbors Matthias and Lea Pabst whose kindness and hospitality I will never forget. Throughout my PhD work, I continued to have contact with Prof. Jákli whose insight and kindness never faltered. I would also like to thank Dr. Stefan Karpitschka, a recent addition to our institute as a group leader, who has done nothing less than impress me with his far-reaching understanding and willingness to give input and help even on topics outside of his area of expertise.

Perhaps the most unexpected, and yet the most life-changing experience that starting a new life here in Germany has brought me was meeting an amazing woman named Fátima during a visit to see my father in Switzerland. I cannot thank my fiancé enough for the love and support which helped me through the final stages of my PhD.

Contents

List of Figures

List of Tables

CHAPTER I

INTRODUCTION

When most people think of fibers, they think of the fibers that make up their clothing or bed sheets: nylon, polyester, spandex, cotton and wool, to name a few. While the first synthetic fiber was introduced only at the turn of the 20th century by Sir Joseph Swan in 1889 and followed by its modern predecessors nylon, polyester, and spandex in 1931, 1953 and 1959, respectively, fibers made from natural materials date back to at least 3000 BCE [6–8]. Exceptional cases like parachutes where lightweight fibers with phenomenal strength are required, or body armor where fiber toughness is a must, the use of silk and aramid fibers, respectively, is common practice [9].

All of these fibers share a single hallmark: they are all solids. Whether they are synthetic, extruded polymers or natural fibers spun from plant matter, their stability relies on a solid phase: their mechanical properties are isolated to that of bulk effects. Attempting to form a stable fiber from a simple liquid is not possible due to a surface-driven instability known as the Rayleigh-Plateau instability (RPI) which limits the length of a stable fluid cylinder to π times its diameter [10]. The mechanical behavior of solids only depends very weakly on the effects of their interface with the surrounding medium. For instance, the surface free energy of a crystal is defined as the change in the Helmholtz free energy (F) as the crystal surface area changes while the surface tension is the force acting tangential to the surface about the surface layer: the force trying to pull together a surface towards a given line drawn across that surface [11]. While the surface free energy and the surface tension are typically equal for simple liquids, they differ for crystals [11]. Moreover, the interpretation of these two quantities in liquid crystals – the state of matter that shares properties both otherwise unique to the solid or the liquid state – becomes even more complex.

The first report of long, stable fibers formed in a liquid crystalline phase was published in 1982 by Noel Clark *et al.* showing that fibers could not only form – but remain indefinitely stable – in a unique liquid crystal phase known as a *columnar* phase formed by *discotic* molecules [1]. Such a structure can be thought of as dinner plates or soup bowls stacked to form a column; their shape allows them to stack efficiently. Then, nearly two decades later, a 2003 publication by Jákli *et al.* showed that columnar phases formed by discotic liquid crystals were not the only contender for forming stable, fluid-like fibers. The spotlight was suddenly stolen by two unique phases formed by *bent-core* molecules[1] – the B2 and B7 phases – which demonstrated the ability to spontaneously form stable, free-standing fibers in air by the simple uniaxial extension of the sample between two solid interfaces. This thesis aims to understand the nature of the surface and bulk energy

[1] Also referred to as *bent-core*, *banana- shaped* or simply *banana* liquid crystals

contributions which are responsible for the viscoelastic behavior of liquid crystalline fibers formed by such bent-core liquid crystals.

1.1 Motivation

The age of modern robotics can be considered to have dawned during western industrial revolution which reached its peak at around 1840. Considered by many to be the first example of a modern robot, a trumpet-playing humanoid soldier was revealed in 1810 in Germany and was swiftly presented at various world fairs [12]. Steam and coal powered machines used basic mechanics and thermodynamics to gain the mechanical advantage needed to put machines to work. This transformation accelerated mankind into the current industrialized and globalized setting we find ourselves in today, in many ways improving our lives and bringing into reach technologies never before dreamt of. The field of robotics has moved steadily from being an industrial tool to improving the lives of the disabled to cleaning our carpets and driving our cars.

A key role in the modern advancement in which we find ourselves today can be attributed to decades of materials research. Carbon fiber, metal alloys, plastic and hybrid materials have made robots lighter advances in battery, actuator and sensor technology have given them a way to interact more seamlessly with the physical world many wish to one day share with them on a regular basis. Recent advancements in processors, broadband communications and artificial intelligence algorithms have provided them with the computational power to perform and improve upon complex tasks with little to no human intervention allowing them to optimize their own functionality beyond our own imagination. Even recent developments in the field known as soft robotics have shown how soft membranes formed into specific shapes can be folded, ballooned, or even made to swim or crawl under very specific circumstances [13–15].

However, among all of these advancements, there remains a shortcoming: heavy, bulky, energy consuming electric or magnetic actuators for moving the mechanical components of robots – especially those of the humanoid or biomimetic class – have hindered the realization of agile and natural movements. For many researchers in the field, the effort to give robots a more human touch has become their top priority. In an attempt to allow precise, smooth, biomimetic articulation of small joints such as fingers, even today's most advanced robots and cybernetic enhancements such as limb prosthetics rely on rudimentary methods of actuation using cables, servos, and other outdated technologies. A solution, in theory, exists to free machines from the shackles of mechanically rigid and limited actuation strategies: *soft actuators*.

The term soft actuators has come to encompass a growing class of materials. Those of interest, those which motivated the following work, would be able to provide precise, uniaxial contraction under the influence of an easily controllable stimuli. The actuator should provide a contractile force over a large strain to make it practical and, optimally, be able to provide actuation at or around room-temperature. Should the material become damaged, it would be advantageous for it to be able to self-heal. Moreover, the actuator should be able to lift a relatively high load and respond quickly to the actuation stimulus. Such materials – either theoretical or realized – are appropriately referred to as *artificial muscles* and a few such examples do indeed exist.

One of the first successful classes of materials to have been rightfully classified as artificial muscles are known as liquid crystal elastomers, or LCEs[2]. This fascinating blend of polymer and liquid crystal physics demonstrated the feasibility of synthesizing soft actuators for the application as artificial muscles in the field of robotics [17, 18]. Another approach involved what are called *electro-active polymers* and was shown to have promise in the field of soft actuators in as early as 2004 [19].

Meanwhile, in a lab at Kent State University's Liquid Crystal Institute in Kent, Ohio, a new and different type of material was being studied which showed promise. Stable, free-standing fibers which spontaneously formed during the uniaxial extension of certain liquid crystalline phases of bent-core molecules seemed to hold strong analogues to muscle fibers. Jákli *et al.* demonstrated that two of the eight known phases formed by bent-core molecules tended to form either stable single fibers or bundles of fibers [20]. Subsequent studies over the following decade showed that these fibers were surprisingly strong, flexible, could be pulled to any length, could self-heal, demonstrated strong temperature dependence and responded promisingly to external electric and magnetic fields. This discover – now nearly two decades ago – has since inspired many to understand the nature of these unique liquid crystalline filaments [3–5, 21–29].

1.2 A Brief Introduction to the Liquid Crystal State

Since the dawn of the liquid crystal era just over 100 years ago, to speak of liquid crystalline as a single phase is severely over-generalized. Since the first *nematogen* was discovered and synthesized by German chemist Otto Lehman in 1905 , it has been shown that there exists an immensely broad spectrum of phases which fit the general definition of *liquid crystalline*: local and/or long-range ordering with a degree of freedom allowing for flow. Most commonly, *mesogens* – molecules which are capable of forming a liquid crystal phase – tend to be *calamitic*: they are rod-like molecules. While calamitic mesogens form a wide variety of phases, they can generally be categorized as *nematic, smectic* or *cholesteric*.

The nematic phase is the simplest; it exhibits only short-range ordering, i.e. each molecule is aware only of its nearest neighbor and the director field – the mean alignment of the molecules – is distorted in a way that minimizes the distortion energy of the total field. The field is distorted by boundary conditions where the molecules at some boundary or interface retain a fixed orientation

[2] For more on LCEs, the reader is encouraged to read the review book Liquid Crystal Elastomers: Materials and Applications edited by de Jeu and published in 2010 [16]

due to either steric effects such as surface texture or by the effects of interfacial energies. *Figure 1.1 (a)* shows an undistorted nematic field composed of calamitic molecules where the boundary conditions are *homeotropic* alignment (the long molecular axis oriented normal to the surface) at the top and bottom and free boundaries at the left and right sides. The molecules are ordered in a global direction – the *director* – typically denoted $\hat{n}$. This is the only order parameter of the nematic phase; the molecules are free to change – or trade – their positions anywhere in the volume as long as their orientation minimizes the free energy of the system [30]. While this is a structured (complex) fluid, it is still limited in its ability to form threads longer than dictated by the RPI of $L > \pi D$ due to its ability to flow freely in all thee dimensions, i.e. it is a three-dimensional fluid.

The three major types of distortions associated with phases stemming from calamitic molecules are *splay, twist* and *bend*. Each of these three modes of distortion are formulated as independent curvatures to the total directory field while unique elastic constants weigh each of the distortion curvatures differently. The approach by Oseen and followed more precisely by F. C. Frank in 1958 allows the distortion free energy to be minimized and the equilibrium director field to be calculated [30,31]. *Figure 1.1 (b)* shows the effect of changing the upper boundary condition from homeotropic to planar alignment for a nematic phase. The field then becomes distorted and the orientation of the molecules must reach a globally minimized distortion energy. The dominant distortion seen in *Figure 1.1 (b)* is *bend*. The contribution of the distorted state to the free energy of the undistorted state – the Frank distortion energy – is given by

$$\mathscr{F}_d = \frac{1}{2}K_1(\nabla\cdot\hat{n})^2 + \frac{1}{2}K_2(\hat{n}\cdot\nabla\times\hat{n})^2 + \frac{1}{2}K_3((\hat{n}\cdot\nabla)\hat{n})^2 \tag{1.1}$$

where the elastic constants K_1, K_2 and K_3 independently weigh the contributions of the splay, twist and bend distortions to the overall distortion energy. Minimizing this free energy results in a solution in which each curvature term is solved and the total equilibrium director configuration can be reached [31].

Other phases – namely the smectic and cholesteric phases – increase this complexity by introducing long-rang order in the case of smectics or *chirality* in the case of cholesterics [30]. Generally, in a smectic phase, the molecules not only have orientational order such as in the *nematic* phase but also positional order where the molecules are confined to live within uniform layers. Since bending of the layers requires the molecules to undergo splay distortion, smectic layers typically remain rigid. In the simplest case, these long-range interactions result in what is known as the *Smectic A* phase; a phase similar to the undistorted nematic phase but with a layered structure. *Figure 1.1 (c)* shows the *SmA* phase in which the molecules are confined to layers and are fixed parallel to the layer normal. *Figure 1.1 (d)* shows the SmC phase in which the tilt angle – the angle between the long-axis of the molecule and the layer normal – is non-zero, distinguishing it from the SmA phase. This additional tilt angle provides an extra degree of freedom allowing the layers to more easily undergo bending than in the SmA phase.

Cholesteric phases – also known as chiral nematic phases[3] typically follow from chirality at the molecular level. Molecules that are chiral are built with a rotational asymmetry which allows them to additionally minimize the distortion free energy by rotating about a helical axis. This, in turn, propagates through the liquid crystal sample until the phase itself is said to be chiral. If the mesogen in question is itself *achiral* – exhibiting no chiral structure at the molecular level – then chiral dopants may be added which – at some critical concentration – invoke a chiral phase. The second term in the Frank distortion energy equation – *Equation 1.1* – associated with twisting the director field is then modified to include the additional curvature term q_0 such that

$$\frac{1}{2}K_2(\hat{n}\cdot\nabla\times\hat{n}+q_0)^2 \tag{1.2}$$

where q_0 is defined as $2\pi/P_0$ and P_0 is the pitch of the cholesteric helix [30]. Chirality can also be observed in smectic phases in which the *Sm̲ectic C̲** phase (SmC*) is defined by the continuous rotation of the molecules about a cone of a fixed angle to the layer normal from one smectic layer to the next.

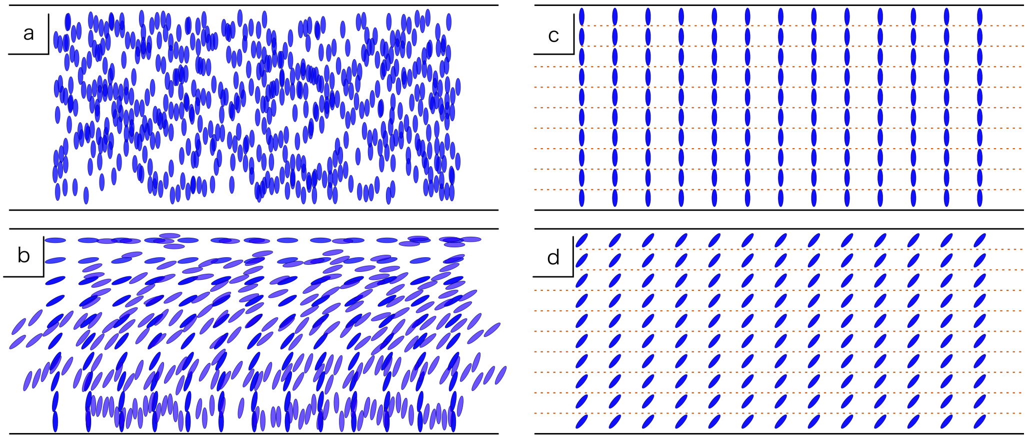


Figure 1.1: Four different key mesophases formed by calamitic molecules. (**a**) The undistorted nematic phase; (**b**) a distorted nematic phase due to specific anchoring conditions at the boundaries; (**c**) The non-tilted smectic A phase; (**d**) the tilted smectic C phase.

[3] The term *cholesteric* in the context of liquid crystal phases originates from the work of the Austrian botanist and chemist Friedrich Reinitzer who discovered the phase while examining the properties of various cholesterol derivatives. The work was presented in Vienna on May 3, 1888 at a meeting of the Vienna Chemical Society and was titled *Beiträge zur Kenntniss des Cholesterins* [32].

Bent-core Liquid Crystals

Apart from the common calamitic mesogens, there exist more complex molecules capable of forming exotic phases beyond the well-known nematics and smectics. In search of new and unique mesophases, German chemist and liquid crystal synthesis pioneer Daniel Vorländer synthesized the first known bent-core mesogens (*Figure 1.2*) in 1929 during his time as professor at the University of Halle [33]. His research focused not only on *bent-core liquid crystals* (BLCs), but on the synthesis and phase behavior of the vast majority of *nematogens* at that time including the discovery of the cholesteric phase observed in the slimy mixture of water and soap commonly found at the bottom of every soap dish [34, 35].

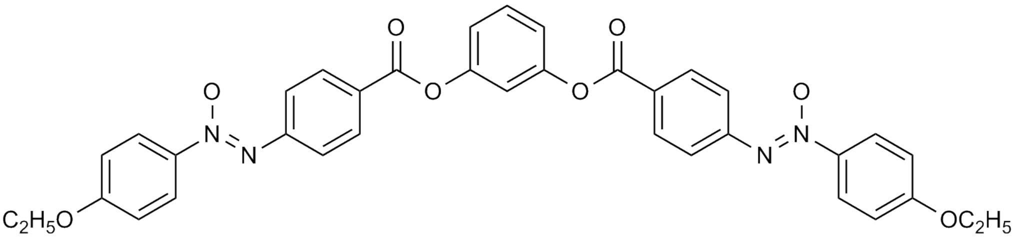


Figure 1.2: Vorländer's first bent-core molecule

While his research directly led to the understanding and synthesis of modern nematogens such as 5CB – responsible for the digital display revolution and present in nearly every flat panel display (LCD) over the past four decades – nearly seventy years would pass before a renewed interest in BLCs would open an entirely new field of study revealing phases, structures and dynamic phenomena beyond anything Vorländer had ever observed. It was the work done by Matsunaga *et al.* at Hokkaido University in 1993 – in which they resurrected the modern synthesis of bent-core mesogens to find that BLCs could exhibit smectic phases – that led to the repeated synthesis of the same compound by Watanaba *et al.* at Tokyo Tech in 1996 [36, 37]. Watanabe's intuition that a bent-core structure in a smectic phase might result in the spontaneous polarization necessary for a ferroelectric mesophase to form paid off when he shared his compound with Hideo Takezoe – also a professor at Tokyo Tech – and observed distinct polar switching [37].

Before the discovery by Takezoe *et al.*, ferroelectric switching behavior had only been observed in tilted smectic (SmC) phases of chiral calamitic and discotic molecules in which polar order is achieved by the simultaneous presence of director tilt and molecular chirality which can be stated as *tilt + chirality = polarity* [38, 39]. Takezoe's discovery that achiral bent-core molecules could also form ferroelectric phases demonstrated that the key requirement for ferroelectric behavior was polar symmetry and that chirality was not needed in every case [37]. Since then, over 500 papers have been published on the topic and eight distinct bent-core phases – denoted B1 through B8 in chronological order of discovery – have been observed and studied in great depth [40].

1.3 Bent-core Fibers: A Recent Development

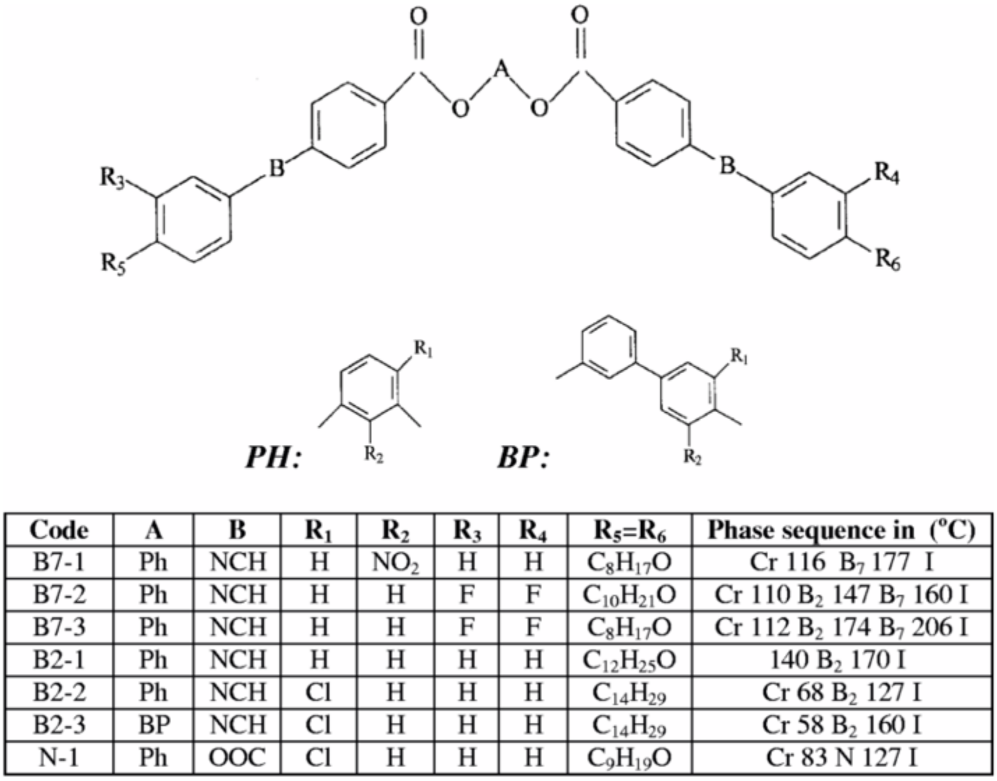
A simple and astonishing statement can be made about the formation of liquid crystal phases by bent-core molecules: *chiral* phases spontaneously grow from *achiral* molecules. This behavior is unique to bent-core mesogens and has fueled countless investigations into the behavior of bent-core phases, both theoretically and experimentally, required many to re-write the book on liquid crystal theories. The bent-core geometry reduces the symmetry otherwise present in calamitic (rod-like) molecules and induces a net polarization perpendicular to the molecular long axis, i.e. in the direction of the bend. This molecular polarity, paired with the bent-geometry, unfolds the range of bent-core phases; each with unique electrical, optical, and mechanical properties.

Of the eight unique phases, just two are known to possess just the right combination of layer-structure, polarity and phase chirality (a consequence of the polarization and bend) to allow stable fibers to form when the material is placed between two surfaces and strained along a single axis: the B2 and the B7 phases [3–5, 21–29]. Both the B2 and the B7 phases share a notable property: they both form a tilted smectic phase which seems to be vital to forming stable, free-standing filaments. Bent-core phases which are not tilted, i.e. where the molecular long axis is parallel to the smectic layer normal, do not form stable fibers [3, 41]. The bent-core geometry, polar ordering, and tilt angle present in the B2 and B7 phases result in layer chirality and has been defined as

$$\frac{\hat{z} \times \hat{n}}{|sin\theta|} = \pm\hat{b} \tag{1.3}$$

where $\hat{n}$ is the long axis of the molecule, $\hat{z}$ the layer normal, θ is the angle between $\hat{n}$ and $\hat{z}$, and $\hat{b}$ is the bend direction which is parallel to the dipole moment for symmetric molecules [40].

It was the keen observation by Jákli *et al.* in 2000 that certain BLC mesogens (*Figure 1.3*) spontaneously formed helical structures within the bulk when confined between two glass slides, cooled slowly from the isotropic phase into the B2 or the B7 phase, and observed under cross polarizers [42] that led to the 2003 discovery that the same material – when heated to the same phases between two glass interfaces – would spontaneously form stable filaments when the two interfaces were moved apart to stretch the material [20]. Once it was established that these liquid crystalline fibers could be consistently pulled and studied, years of research quickly followed to understand everything from their layer structure and optical properties to their elastic and mechanical stability and potential activation response [3–5, 21–29].



Code	A	B	R_1	R_2	R_3	R_4	R_5=R_6	Phase sequence in (°C)
B7-1	Ph	NCH	H	NO_2	H	H	$C_8H_{17}O$	Cr 116 B_7 177 I
B7-2	Ph	NCH	H	H	F	F	$C_{10}H_{21}O$	Cr 110 B_2 147 B_7 160 I
B7-3	Ph	NCH	H	H	F	F	$C_8H_{17}O$	Cr 112 B_2 174 B_7 206 I
B2-1	Ph	NCH	H	H	H	H	$C_{12}H_{25}O$	140 B_2 170 I
B2-2	Ph	NCH	Cl	H	H	H	$C_{14}H_{29}$	Cr 68 B_2 127 I
B2-3	BP	NCH	Cl	H	H	H	$C_{14}H_{29}$	Cr 58 B_2 160 I
N-1	Ph	OOC	Cl	H	H	H	$C_9H_{19}O$	Cr 83 N 127 I

Figure 1.3: The first bent-core compounds shown to form stable, free-standing fibers are shown along with their phase transitions. Only the B2 and B7 phases showed to form fibers. The nematic phase, N-1, did not.

Static Tension Investigation

In 2012, Morys *et al.* introduced a technique for measuring the static tension of a liquid crystal fiber formed by BLCs [4]. In their paper, they used a mesogen (*Figure 1.4*) which tended to form very thick fiber bundles – typically between 40 and 100 μm in diameter compared to the 1 to 6 μm thick single fibers studied before – composed of tightly packed single fibers. These fiber bundles were prepared in the same manner as other BLC fibers; material was placed between two glass rods inside a heating chamber with observation windows, the chamber was heated to a temperature exceeding the isotropic transition of the material and cooled into the smectic phase before the two glass rods were pulled apart to form a long, stable fiber. At this point, a small glass bead was lifted into the heating stage through a small hole in its bottom face. The glass bead was carefully brought into contact with the midpoint of the fiber where it adhered strongly due to capillary forces. The bead was then lowered until it detached from its delivery mechanism and hung freely from the fiber as schematically shown in *Figure 1.5*. The accompanying force balance was written as

$$\Sigma = \frac{F}{sin(\alpha_1) + sin(\alpha_2)} \tag{1.4}$$

or, since $\alpha_1 = \alpha_2 = \alpha$,

$$\Sigma = \frac{F}{2sin(\alpha)} \tag{1.5}$$

where Σ is the fiber's tension – or restoring force – along its axis, α is the deflection angle of the fiber from the horizontal, and $F = mg$ is the weight of the glass bead where m is its mass[4] and g is the acceleration due to earth's gravity.

Plotting this steady restoring force versus fiber perimeter revealed a quadratic increase as the fibers became thicker indicating a quadratic contribution to the restoring force instead of a linear one; this indicated that the restoring force was growing as the cross-sectional area and not as the circumference. Due to this result, it was understood that the bulk of the fibers – not only the surface tension – played a key role in their elastic response.

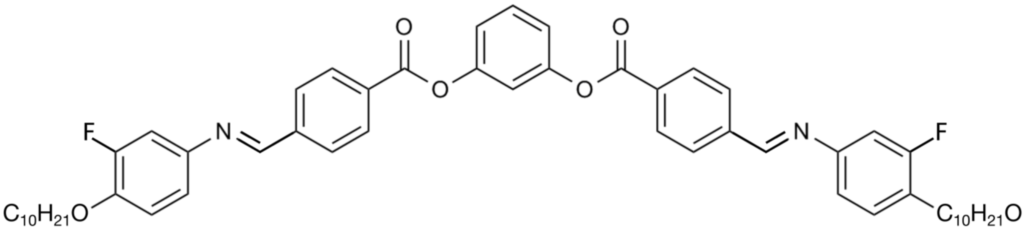


Figure 1.4: The mesogen used by Morys *et al.* in 2012 to measure the tension of bundles of BLC fibers

They noted that while pulling the initial fiber, its diameter was observed to remain constant indicating that material had to be flowing from the excess material at the boundaries to continuously form more fiber. This conclusion made sense for millimeter-long fibers being pulled from an initially fluid bridge with a length on the order of a few microns. The same assumption was then applied to the behavior of the fiber during bead hanging since no observable change in the fiber radius was observed during its deflection. The total tension, in terms of the surface and bulk contributions, was presented as

$$\Sigma = \sigma P(a,b) + \eta \pi ab \tag{1.6}$$

where σ is the surface tension[5], $P(a,b)$ is the perimeter of the fiber bundle corrected for corrugation due to packing, a,b are the major and minor radii to account for fiber profiles deviating from the perfectly circular, and η represents the bulk contribution to the total fiber tension as a volume

[4] The mass of the glass bead was calculated using the known value of the density of glass, assuming a spherical shape and optically measuring its diameter.

[5] In this experiment, and many that followed, the temperature independent surface tension value of 0.02 N/m was used. This value seems to have come from a 2010 paper by Bailey *et al.* [23]. Arguments against using this value are presented in the *Analysis* chapter

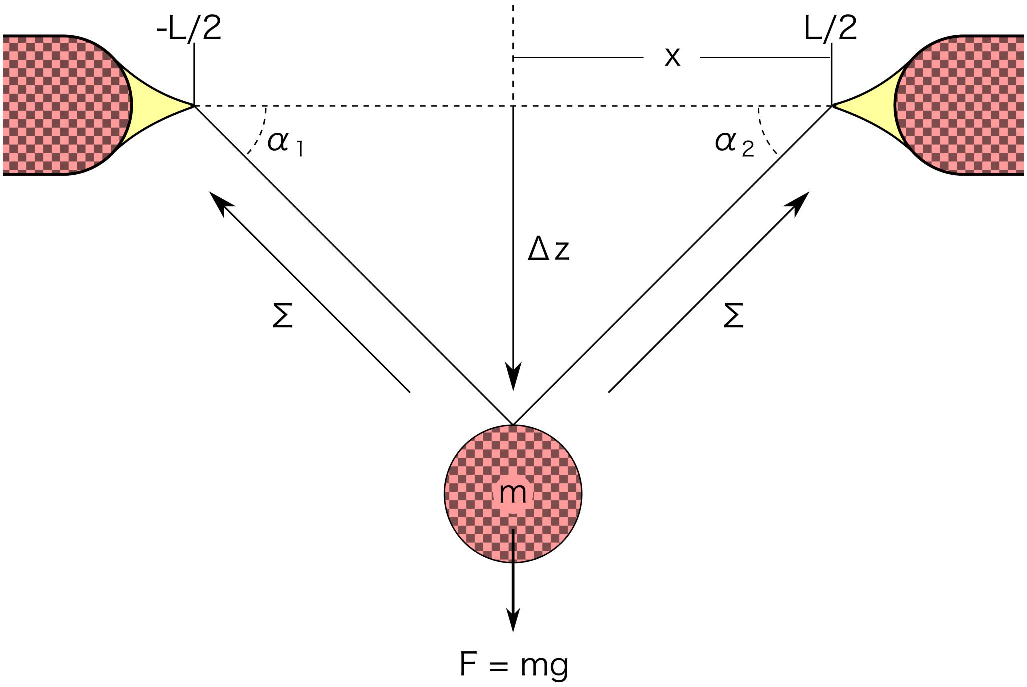


Figure 1.5: The force balance used by Morys *et al.* in their attempt to properly gain insight into the elastic nature of BLC fiber bundles as described in *Equation 1.5*. The red (checkerboard) sections indicate glass, while the yellow (plain) sections indicate the excess material left over on each glass rod from which each fiber was pulled.

energy density. They concluded that the bulk contribution to the fiber tension was coming from the energy needed to add new material to the bulk and that the thicker the fiber, the more total energy would be required to add material in proportion to its cross-sectional area. This, however, does not have to be true.

During bead hanging, the deflection angle α typically did not exceed 30° [6]. Using a constant-volume approach with a Poisson ratio of 0.5, the radial strain (ε_r) of the fiber is related to its extensional strain (ε_{ext}) by

$$\varepsilon_{ext} = \frac{\Delta L}{L_0} = -2\frac{\Delta R}{R_0} = -2\varepsilon_r \tag{1.7}$$

Furthermore, after some algebraic and trigonometric manipulation, the deflection angle α can be related to ε_{ext} using the expression

$$\varepsilon_{ext} = sec(\alpha) - 1. \tag{1.8}$$

For $\alpha = 30°$, *Equation 1.8* yields $\varepsilon_{ext} \approx 0.15$ and $\varepsilon_r \approx -0.075$. Therefore, in the most extreme case, a fiber bundle with an initial diameter of 40 μm would be compressed down to a diameter of 37 μm upon hanging a glass bead. Since both interfaces holding the fiber were visible and fiber lengths were typically on the order of a millimeter, it can be assumed that the low magnification would not have revealed such radial compression (beyond reasonable error) giving the experimenter the

[6] Deduced from the images presented in the publication

illusion that the fiber diameter had remained constant.

This 2012 study by Morys *et al.* inspired the current work to focus on measuring the elastic properties of single fibers instead of fiber bundles to try to observe the limit where the surface effects and bulk energy effects converge. Similar bead hanging experiments were done by Kress in 2015 which showed that the assumed value of the surface tension given by Bailey *et al.* in 2010 which was used in Morys' study was too large by about a factor of 2 [23, 43].

Dynamic Studies

Earlier, in 2005, the dynamic response of fiber bundles formed by the same mesogen later used by Morys *et al.* shown in *Figure 1.4* was examined by means of acoustic driving by Stannarius *et al.* at the O.v.G University in Magdeburg, Germany. [27]. Their study showed that these fibers responded similarly to rigid beams when driven by acoustic compression. Later, Petzold *et al.* demonstrated another interested feature of bent-core fibers: they responded strongly to external electric fields.

Most recently, in 2016, the dynamic behavior of single bent-core fibers was probed by analyzing the kinematics of recoil after sudden rupture. Using the same mesogen as Kress in 2015 – a mesogen that consistently formed single B7 filaments – Salili *et al.* showed that the fibers tended to recoil in a helical fashion when ruptured either thermally at one or or mechanical via impulse.

Albeit the many efforts to understand the material response of these complex fibers, the prior work was forced to implement indirect methods of studying their dynamic behavior. A more straight-forward experimental approach was necessary.

1.4 Extensional Rheology

The field of extensional rheology first gained traction at the beginning of the twentieth century when an Irish physicist by the name of Frederick Thomas Trouton published his 1906 work titled ”On the Coefficient of Viscous Traction and its Relation to that of Viscosity.” After contemplating the torsional methods used up until then which employed rods or cylinders rotating in pitch or tar and the observation that the rate of turning of these rods was ”.... not strictly proportional to the driving couple,” he proposed that ”.... the rate of flow of the material under shearing stress cannot be in simple proportion to stress.” This inspired him to investigate the extensional flow of viscous materials by measuring what he called the ”viscous traction” which would later be to the material parameter now known as the extensional viscosity.

In his experiments, he prepared rods of the material in question with each end of the rod shaped into the form of a cube, much larger in diameter than that of the cylindrical rod itself. The shape was reminiscent of a dumbbell and each ”head” pressed into a metallic box open to one side with a horizontal slot on top so the rod portion stuck vertically from each box. One of the metal boxes was held fixed while the other hung freely and had a hook so a weight could be added. Once released, the lower box with the added weight would stretch the cylindrical rod segment oft he material. The rod’s rate of elongation was then measured as the rate of the lower metal box. In measurements of low-density fluids, the lower metallic box was submerged in a density-matched (to the material in question) bath of salt water and the rod pulled upwards by a weight attached to a pulley in order to eliminate material stresses due to the material’s own weight.

His measurements showed that the rate of extension of the rod under the constant force of gravity was approximately proportional to the cross-sectional area of the rod which he wrote as

$$\frac{dv}{dx} = \lambda \frac{F}{A} \tag{1.9}$$

where F is the applied force, A the cross-sectional area of the rod, v is the velocity (rate of elongation) of the rod and λ a proportionality constant which he called the coefficient of viscous traction.

He then showed that an increase of the applied force, or weight added to the rod, caused the linearity of the traction coefficient λ to degrade such that an increased applied forces resulted in a decreased λ. In addition, the initial rate of elongation was faster than the transient rate, while removal of the weight, or force acting on the rod was followed by a short recovery period. In this experiment, he showed the effects of strain hardening. He found that past a certain threshold applied force that

$$T - T_0 = \lambda \frac{dv}{dx} \tag{1.10}$$

where the rate of elongation was taken past the initial non-linear behavior where the behavior was strongly linear. He showed that, by using different colored materials, that the flow field was such that the midpoint of the rod remained in a plane, i.e. that the particles flowed away from a stagnation point at the point between each end box.

He also described the shape of a stream of material under its own weight. Knowing that the force of traction is everywhere the same, he could state the the product vA was constant everywhere. He then started from

$$\frac{dF}{dx} = g\rho A + A\rho \left(\frac{dv}{dt} + v\frac{dv}{dx} \right) = 0 \tag{1.11}$$

and followed by assuming small rates of elongation in order to neglect the acceleration term, resulting in

$$\lambda \frac{d}{dx} \left(A\frac{dv}{dx} \right) = -g\rho A. \tag{1.12}$$

Following substitutions, differentiation and re-arranging of terms, he reached the expression

$$\frac{1}{y^3}\frac{d^2y}{dx^2} - \frac{1}{y^4}\left(\frac{dy}{dx}\right)^2 - \frac{1}{K^2} = 0 \tag{1.13}$$

where x is the distance from the top of the rod and y the radius at any point x. The profile, for short, elongated rods where the profile near the end of the rod is more apparent is

$$y = \frac{b}{sinh\left(\frac{bx}{K}\right)} \tag{1.14}$$

which reduces to

$$xy = K \tag{1.15}$$

for the limiting case for long filaments where $b \to 0$. This shows the profile of the surface of a long filament, where $K^2 = 2\lambda M/g\rho^2\pi$. This profile was experimentally verified when he measured the rate of elongation and the radius of the rod at different points x from the top.

The contribution of his work came at the end of his paper when he remarked on the connection between his coefficient of viscous traction λ and the coefficient of viscosity which had been used to define resistance of flow due to shear stresses. Decomposing the shear into two equal components and adding dilational stress where all three components have to be equal in an isotropic medium showed that the tractive (elongational) stress is three times that of the shear stress. For the case of shear Trouton defined viscosity as

$$S = \mu\dot{\phi} \tag{1.16}$$

where S is the shear stress, μ is the viscosity and $\dot{\phi}$ is the shear rate. The tractive stress was therefore

$$\sigma = \lambda\dot{\varepsilon} \tag{1.17}$$

resulting in what is now known as Trouton's ratio,

$$\frac{\sigma}{S} = \frac{\lambda}{\mu} = 3. \tag{1.18}$$

Following Trouton's work, the field evolved steadily throughout the 20th century.

Modern Extensional Rheology

The field of modern extensional rheology began with a report of a simple method of confining fluid to uniaxial extension and measuring the coupling between its resistance to flow and its deformation by what is known as the ”falling plate method” in 1990 by Matta *et al.* [44]. The method was simple: A fluid sample was placed between the flat faces of two co-linear cylinders with their axes oriented vertically. The upper cylinder remained fixed while the bottom cylinder was able to be released into a free-fall. The free-fall time and velocity profile of the free cylinder was compared to that of a cylinder being slowed down by the fluid as it was forced to stretch. A camera as used to measure the evolution of the midpoint diameter of the fluid sample during the experiment. The fluid's extensional viscosity could then be calculated from the diameter of the fluid and the acceleration of the falling cylinder versus time. Using lower plates of varying masses resulted in different strain rates which allowed a relatively accurate measurement of the strain rate dependence of the extensional viscosity.

Subsequent iterations of this method were developed throughout the 1990s. In 1991, Sridhar *et al.* replaced the simple falling plate with a motorized plate which could be controlled in order to achieve a ”constant stretch rate” elongation [45]. *Figure 1.6* shows a diagram of a fluid filament being drawn between two flat surfaces and the displacement and strain curves versus time associated with a ”constant stretch rate” – today referred to as a constant *strain rate* $\dot{\varepsilon}$ – mode of elongational deformation.

A few years later, Sridhar *et al.* published another paper on a modified version of their rheometer in which both plates moved apart from each other. Updates included improved force transducers and filament diameter measurement paired with the ability to stretch liquid samples horizontally instead of just vertically [46]. A year later, in 1994, Berg *et al.* reported a further modification to this design which could reduce the end-plate diameters during extension forcing the stretching fluid bridge to retain a cylindrical profile from end-to-end. The experiments by Berg *et al.* were carried out in microgravity via a drop tower [47].

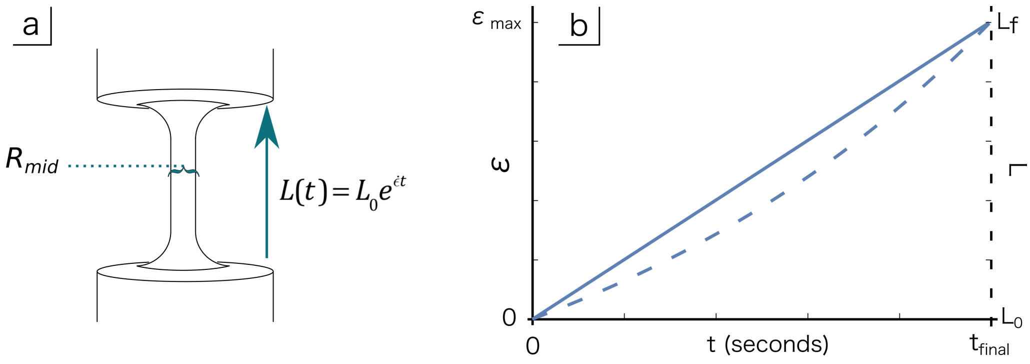


Figure 1.6: **(a)** A fluid sample stretched between two surfaces at a constant strain rate $\dot{\varepsilon}$ while the thinning of the midpoint diameter is being measured. **(b)** Plot showing the relation between a constant strain rate (solid line) and an exponential displacement (dotted line) versus time.

As described by Sridhar *et al.* in 1991, a fluid filament may undergo purely extensional flow if the strain rate – $\dot{\varepsilon}$ – is held constant so that the velocity gradient is zero throughout the fiber. The strain which follows this behavior is commonly known as *true strain*, *Hencky strain*, or *logarithmic* *strain* which is derived by stating that the incrementally small strain

$$\varepsilon = \frac{\delta L_0}{L_0} \tag{1.19}$$

is the incremental change in the fiber's initial length, L_0 and is related to the strain rate, $\dot{\varepsilon}$ by

$$\dot{\varepsilon} = \frac{\delta \varepsilon}{\delta t} = \frac{\delta L_0}{L_0 \delta t} \tag{1.20}$$

Rearranging and integrating over the length of the fiber and over all times,

$$\int \dot{\varepsilon} dt = \int_{L_0}^{L(t)} \frac{dL_0}{L_0} \tag{1.21}$$

yields

$$\dot{\varepsilon} t = ln \frac{L(t)}{L_0} \tag{1.22}$$

finally resulting in the exponential displacement profile

$$L(t) = L_0 e^{\dot{\varepsilon} t} \tag{1.23}$$

and likewise the velocity profile

$$V(t) = L_0 \dot{\varepsilon} e^{\dot{\varepsilon} t} = \dot{\varepsilon} L(t). \tag{1.24}$$

Improvements to the force balance theory were made by Szabo in 1997 which became the standard for analyzing the experimental construction of an extensional rheometer [48]. The effects of surface tension, gravity, inertia, initial aspect ratio and shear at the boundaries (lubrication versus non-slip boundary conditions) were included. Understanding the limits of a fluid's initial state is important and certain parameters, such as the initial aspect ratio needed to avoid initial domination by shear effects, become vital in designing a viable experimental method.

While purely extensional flow is described by a zero-shear state, most experiments involving polymer melts and solutions exhibit boundary effects that are hard to ignore since the aspect ratio of the fiber – its length to diameter ratio – is typically small. Moreover, in extreme cases such as the the boundary behavior of certain semi-dilute polyisobutylene polymer solutions under uniaxial extension, filaments were even shown to break up into many smaller filaments when the initial aspect ratio was close to unity. Such an initial condition creates extremely strong shearing behavior as the filament radius quickly shrinks before the material at the boundaries can flow across the surface towards the fiber's axis [49].

However, for extremely slim fibers with very high aspect ratios, the effects of shear flow at the boundaries can be considered completely negligible. If these fibers are also very thin (low mass) and are stable under static conditions, other terms from the expanded force balance by Szabo – such as inertial terms or sagging due to gravity – can be neglected as well making the analysis of the behavior of liquid crystal fibers formed by bent-core molecules much more straight forward.

The standard equation for measuring the extensional viscosity while neglecting boundary effects is

$$\bar{\eta}^{+} = \frac{\sigma}{\dot{\varepsilon}_0} - \frac{\gamma}{2\pi R_{mid}\dot{\varepsilon}_0} \tag{1.25}$$

where σ is the stress, or the measured forced divided by the cross-sectional area of the midpoint of the fluid column, $\dot{\varepsilon}_0$ is the strain rate, and γ is the liquid/air surface tension. The surface tension term offsets any restoring forces that are measured due to surface effects in order to isolate the bulk behavior.

The next evolution of the extensional rheometer appeared when McKinley *et al.* from MIT developed what they called the *FiSER*, or the Filament Stretching Extensional Rheometer. Based strongly on the updated design of Matta's device by Sridhar [50], the improved version includes a high-speed camera which travels on a rail parallel to the moving plate that pulls the fluid sample. The plate which stretches the sample is moved by a motor at an exponentially increasing velocity to impose a constant strain rate while the surface on which the fluid is placed contains a force transducer. The camera moves along with the fluid sample to record the evolution of the midpoint radius by moving at half the speed of the sample plate. While the falling plate method imposed a constant force on the fluid while measuring changes in the strain rate and midpoint diameter, the FiSER controls the strain rate and measures changes in force and midpoint radius similar to the device presented by Sridhar.

An alternative method of measuring the properties of fluids under uniaxial, shear-free extension exists in the form of a device called *CaBER*, or the Capillary Breakup Extensional Rheometer, currently sold by Thermo Fisher Scientific and also developed by McKinley *et al.* This method uses a laser micrometer to measure the midpoint diameter evolution of a suddenly extended fluid cylinder. This device does not measure the force acting on the fluid sample nor does it control the strain rate. Instead, the fluid sample is quickly stretched to a pre-defined length and allowed to relax until capillary break-up occurs while the midpoint diameter is measured versus time. This is a much less quantitative method of categorizing a fluid and results in plots of relaxation time and pinch-off time characterize fluids based on a qualitative behavioral analysis.

The CaBER and FiSER are in continued development by a spin-off from McKinley's lab called Cambridge Polymer Group, Inc. While the CaBER is sold by Thermo Fisher, the FiSER is sold exclusively by CPG. Both of these devices have limited resolution of the midpoint radius due to using laser micrometers, work with large sample sizes, are expensive, large, and are not user serviceable. The FiSER is a large device which uses are large sample volume, has a limited operating domain and is expensive.

Micro-Extensional Rheometry

In 2017, Paul *et al.* reported the first results of a new filament stretching rheometer called the Microextensional Rheometer (MER) which was patented by Rao *et al.* in 2013 [51,52]. As opposed to the previous extensional rheometers which only imposed constant strain rate extension on the material sample, their device was used to measure the response of a fluid bridge formed between to vertical, flexible fiber optic glass fibers to uniaxial oscillatory driving. One of the two glass fibers was calibrated to obtain a known spring constant, while the other was inserted into a metal tube to effectively make it rigid. Then, the ”rigid” fiber was driven by a piezo transducer at a constant frequency and amplitude while the opposing glass fiber, connected only through the suspended fluid sample deflected in response to the fluid's response. The motion of both glass fibers was tracked by coupling a laser through each one tracking the point signals using a CMOS chip. While their design did address the benefits of introducing oscillatory driving to extensional rheometry, the lack of temperature control, large amplitudes and the need for a multi-camera system and timing adjustments to needed to simultaneously image the motion of the fiber tips and the deformation of the fluid sample were setbacks that needed to be addressed if response of exotic materials such as bent-core fibers were to be studied under similar excitation modes.

1.5 Liquid Crystals Under Extension

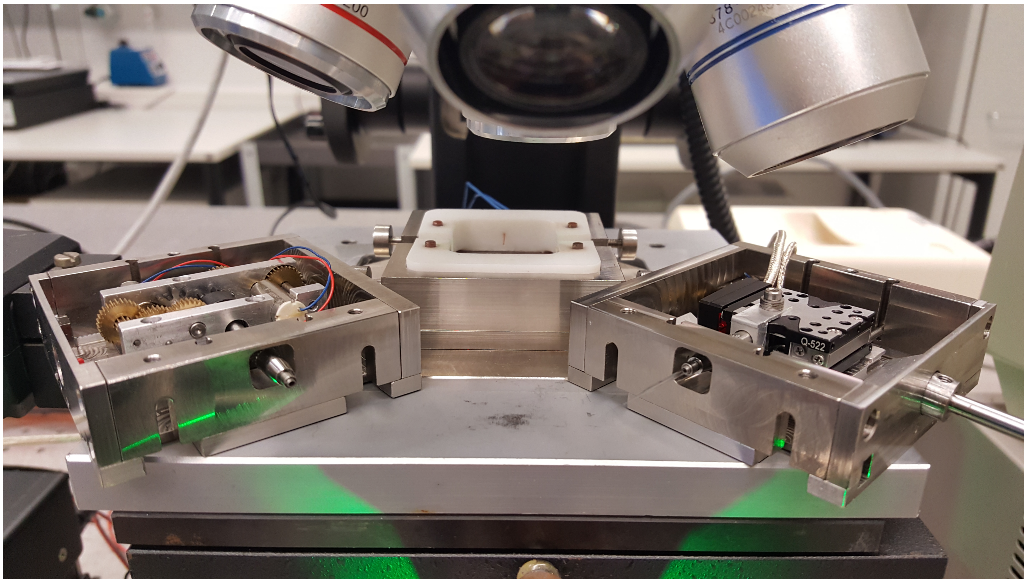
It seemed natural that the field of extensional rheology – almost exclusively developed for the study of the extensional flow behavior of polymeric systems – would need to be merged with the field of liquid crystals if insight were to be gained into the dynamic, rheological response of free-standing fluid-like fibers formed by bent-core liquid crystals. The dynamic studies of the past were difficult to interpret and comparison with other less complex systems – such as polymers most commonly studied using rheological techniques – was not straight forward.

To obtain a more complete understanding of the molecular nature of the viscoelastic properties of such fluid filaments, a new device had to be developed tailored specifically to examine the properties of these bent-core mesogens. The rheometers which had already been developed in the past were not well-suited for this study. They typically required sample volumes far too large for the current work; the bent-core mesogens in question exist in quantities of just a few micrograms and volumes of only a few nano-liters are required to form and study stable fibers. The spatial and temporal resolution of the prior art was also unsuitable for the current study, as was the lack of temperature control and dynamic range of strain, stress and total sample viewing area.

Upon realizing the limitations of the current technologies and the undeniable need for such new measurements to be acquired, the design and construction of a new type of extensional rheometer under the working title of ”Nano-Extensional Rheometer” (”NExt Rheo”) was initiated. The following chapter, titled *Device Development*, tells the story behind the conception of the original design followed by a detailed guide through the rheometer's construction and the control software used. This new, compact, high-resolution tool becomes capable of real-time mechanical analysis of large and small amplitude oscillatory, step-strain, and extensional flow rheology using nano-liter sample volumes with nano-newton force resolution, nanometer spatial resolution and time resolution limited only by the type of camera used.

CHAPTER II

DEVELOPMENT OF A NANO-SCALE EXTENSIONAL RHEOMETER



It was the work of two MPI colleagues and close friends Dr. Christian Kreis and Dr. Marcin Makowski which helped lead to the design and implementation of a force-sensing technique they were using to measuring the adhesive nature of a microalgae called Chlamydomonas [53] and the viscoelastic nature of spherical colonies of Volvox, respectively, which was first introduced to them by the work Dalnoki-Veress et al. in 2009 [54]. Dr. Kreis, a PhD student at the time had been successful in using small glass cantilevers carefully made from hollow glass micropipettes, or capillary glass, to hold on to micro algae while they attempted to adhere, and pull themselves to, certain interfaces under certain lighting conditions [53].

Having noticed the impressive resolution which resulted not only from quality construction of the micropipettes but also from clever approaches and implementations of image analysis – combined with an achievable force sensing range spanning that of the known stress magnitude of the fibers (*Figure 2.1*) – the idea to adapt the technique to measure the internal elastic and viscous forces of very thin fluid-like filaments quickly materialized. Dr. Makowski, a post-doc at the time who had been responsible for most of the lab's setup and pipette forging techniques gained interest in my need to design a compact, high-resolution extensional rheometer. Together, the design of a new micropipette perfectly suited for the chamber geometry of the heat stage, the required range of motion and force sensitivity needed to study the material, and simple implementation and exchangeability in the rheometer was developed. Although the designs for the heat stage portion of the device and the positional control components were already underway, alterations were quickly made to accommodate the new force-sensing pipettes.

After quickly browsing the literature on extensional rheometry and the current state of the art, it became immediately clear that this new device had the potential to not only examine the nature of liquid crystal fibers under uniaxial extension, but of any extremely small sample of a simple or complex fluid. The work by Dr. Gareth H. McKinley at MIT strongly motivated the design and construction of this new extensional rheometer which could surpass nearly all limitations present in the current state-of-the-art. It became obvious that the new device would become a fully-functional extensional rheometer.

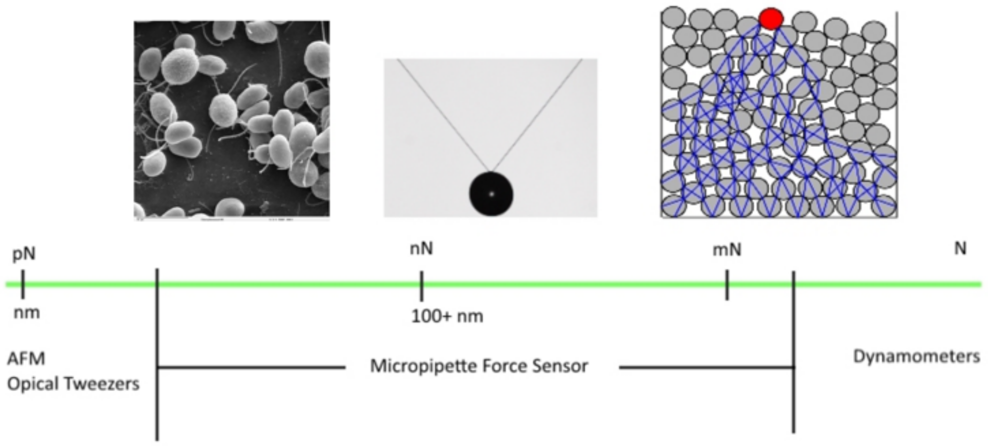


Figure 2.1: The range of forces measurable using the micropipette technique. The technique allows rigid and soft materials to be probed on size and force scales that fill the gap between AFM or optical tweezer methods and dynamometers

The new device was given the working title Nano-Extensional Rheometer (NExt Rheo) and its design was guided by the general requirements (listed on page 22) gathered from past experience with fiber-pulling devices and the current state-of-the-art of extensional rheometers. *Figure 2.2* shows a sketch of the rheometer's final design while *Figure 2.3* shows perspective sketches of the rheometer with and without its lids installed. The following sections describe the design process and assembly of each major component, calibration techniques, control software and rheological programs and algorithms required for the final realization of this new extensional rheometer and are presented in the order *Hardware, Force-sensing, Software, Pipette Calibration*. The last section, titled *Finished Product*, summarizes the completed rheometer and makes comparisons against the current competing technologies and discusses the advantages and disadvantages of the new design.

- The device should be self-contained in that its force detecting mechanism, positioning mechanism(s) and temperature control mechanism may be fitted into single, closed unit.
- The device should be modular in that the positioning and temperature control mechanisms be removable from one another for repair, cleaning, maintenance and modification.
- The heating mechanism of the device should stable, gradient-free temperature zone larger than the sample size or the range of motion of the sample holder or force detection mechanism.
- The force sensing mechanism should not interact either chemically or physically in an unexpected way with the sample to be measured.
- Infinitesimal strains should be achievable in order to isolate linear elastic and viscoelastic behavior of the sample under uniaxial extension. Likewise, larger strains should be achievable in order to examine nonlinear flow behavior due to large-amplitude uniaxial oscillations or constant strain rate stretching.
- Strain rate profiles should be programmable in order to achieve any desired uniaxial motion required for all types of rheological analysis.
- Stresses of the smallest material order, or more precisely stress responses due to the smallest obtainable infinitesimal strains should be measurable with a more-than-adequate signal to noise ratio.
- The contribution of the deflection of the force sensing mechanism should be negligible compared to the imposed strain of the material while measuring the resulting stresses in order to simplify the analysis.
- The device should include both programmable control of the interface and sample positions as well as direct user control via physical input such as that of a joystick.
- The device should allow observation of the sample itself while simultaneously measuring strain and stress responses and temperature.
- The device should have interchangeable viewing windows for use in a variety of experimental environments including fluorescent imaging, second harmonic generation (SHG) and X-ray setups.
- The device should be capable of operating under any orientation, including while fully inverted, to accommodate different experimental environments

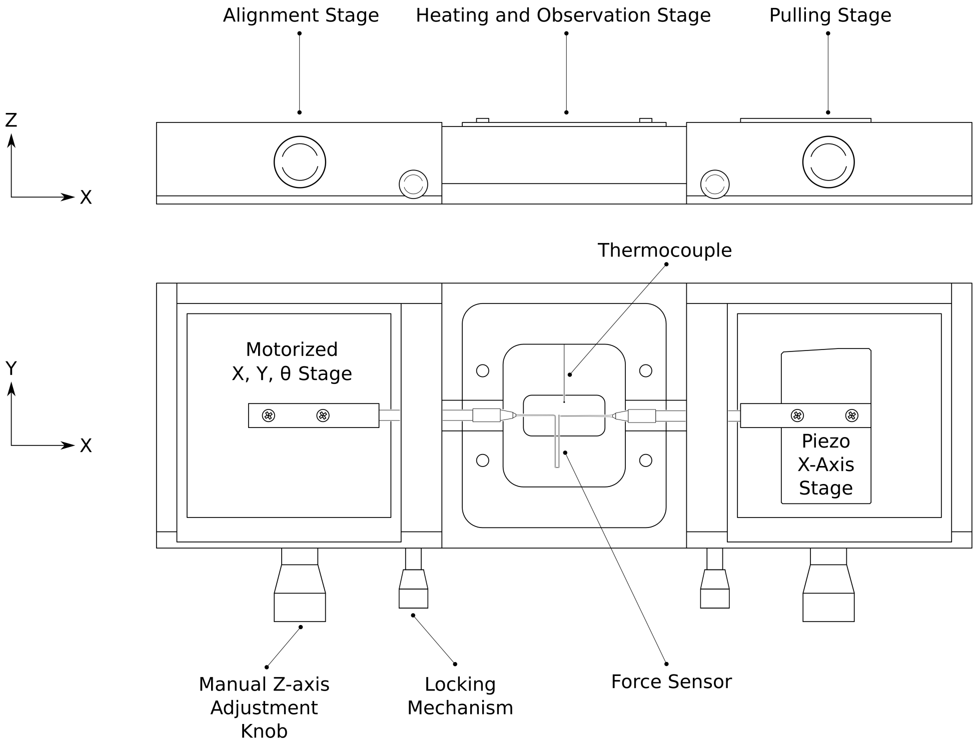


Figure 2.2: Detailed sketch of the completed rheometer

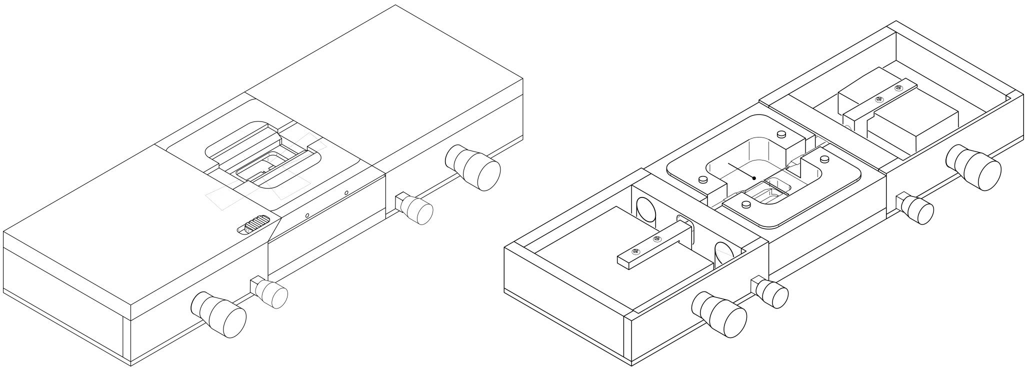


Figure 2.3: Perspective views of the opened and closed rheometer

2.1 Hardware

The following section outlines the design and manufacturing process of the main hardware components. The first subsection discusses the design of the heat stage is is broken down into four parts, each discussing the one if its four major developmental steps; its main body (2.1.1), its lid (2.1.1), the numerical study which drove those designs (2.1.1)and finally the temperature controller. The second sub-section describes the design of the positional control mechanisms and is broken down into two parts; one explaining the design and functionality of the mechanical stages themselves and the second demonstrating how the stages were controlled and the details of the joystick controller. The final subsection briefly describes any additional hardware that needed to be designed such as additional optical components for the microscope.

2.1.1 Heat Stage

Any proper study of liquid crystalline properties, whether they be of the mechanical, optical, electronic, etc., requires complete control over the sample temperature. The study of bent-core fibers is no different, and so a new heat stage compatible with the requirements and goals of the extensional rheometer had to be developed and was the first major design piece of this device. After considering the spatial limitations of the device and the need for a modular and robust design, the agreed-upon specifications were:

- Locking, sliding lid to easily add new sample material to testing interfaces
- Lid with a large[1] viewing window to observe large displacements with interchangeable glass for later experimental compatibility
- Well insulated to protect user and surrounding equipment from high ambient and contact temperatures
- Even (gradient-free) temperature distribution throughout the sample region to avoid temperature inconsistencies in the sample materials
- Data logging and temperature control via a graphical user interface (GUI)
- Stable temperatures to within +/- 0.2 °C from room temperature up to 200 °C
- Modular design allowing quick detachment from the rest of the device for cleaning, maintenance and late upgrades and modifications

[1] Large enough to observe the material's deformation and the full movement of both pipettes during the experiment

The design of the heat stage was done using Solid Edge 7.0 CAD software by Siemens using an institute license. The software was chosen for compatibility reasons in order to allow the designs to be forwarded to the MPI-DS machine shop's designer, Wolf Keiderling, for further input and advice on any changes required before final submission to the machine shop for machining. In order to optimize and numerically verify the functionality of the heat stage before final production, Matlab code was written to simulate and further optimize the functionality of the heat stage by solving the time dependent, conductive heat equation in three dimensions by the finite-difference method. Using the known values for the thermal conductivity of stainless steel, copper, glass, air and Teflon (PTFE), the thermal environment inside the heating chamber could be optimized by adjusting the material geometries and size and power of the heating elements. The method evolved until the simulations showed that the volume of air in which the sample material would be studied had a near-zero temperature gradient in all directions and that ramp rates, insulation and stability were optimized.

Two 80W AC cartridge heaters were embedded in a copper base uniquely designed to provide heat to the chamber while adding a small region of air between it and the PTFE (Teflon)/steel/glass base. They were chosen due to their compact size, high output, and ceramic insulation protecting the wires from thermal degradation. The inner horizontal walls of the main steel chamber were lined with a thick Teflon wall nearly 1 cm thick. Starting from the base, four copper rods rose up through the Teflon walls to passively conduct heat from the lower heating plate to the sliding lid. When the lid is slid into the closed and locked position (locked by a thumb lock and spring-loaded detent), these conductive rods lock into two small plates in the lid. These plates provide the heating needed from above the sample volume to heat the chamber evenly, regardless of ramp rate or absolute temperature. Special attention was given to producing a gradient-free temperature zone due to the strong temperature dependence of the mechanical nature of liquid crystals and the need to often work within a few degrees Celsius of phase transitions and mesophases. The final design is shown in *Figure 2.4*. The tapered machining of the lid in the vicinity of the viewing window serves the purpose of allowing microscope objectives with short working distances to image the sample while still providing adequate thermal protection.

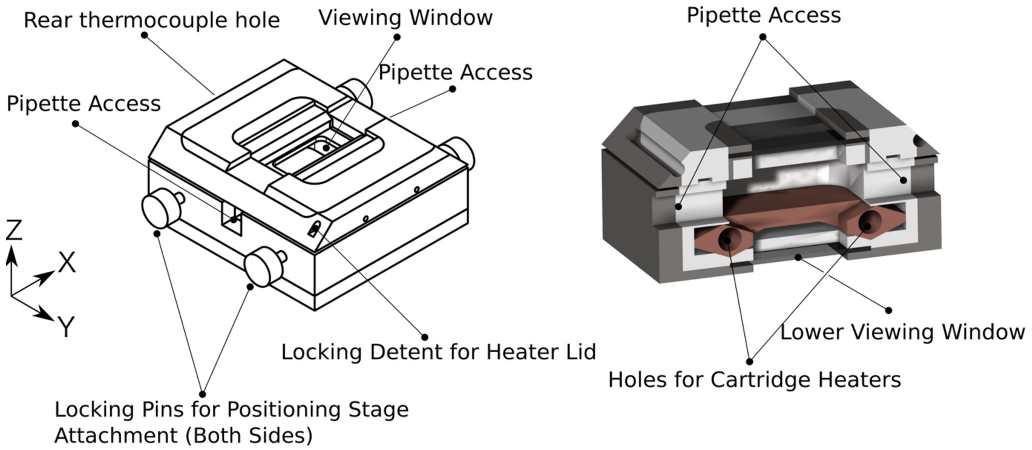

Rear thermocouple hole
Viewing Window
Pipette Access
Pipette Access
Z
X
Y
Locking Detent for Heater Lid
Locking Pins for Positioning Stage Attachment (Both Sides)
Pipette Access
Lower Viewing Window
Holes for Cartridge Heaters

Figure 2.4: Draft (left) and cross-section of a rendered version (right) of the final design of the heat stage

Heat Stage Lid

The lid of the heat stage was designed to be quickly and safely removable from the stage's main body, be well-insulated, resistant to thermal deformations, be easily disassembled, have an easily exchangeable window and include heating elements which – when the lid was locked into place – passively heated the sample volume from above by conducting heat from the lower heating element. The lid itself is composed of a combination of Teflon and stainless steel pieces that optimize rigidity, minimize distortion due to high temperatures and optimize thermal insulation. The components are vertically inserted into each other and then fixed in place by two threaded retaining pins. The viewing window is clamped between the inner Teflon and inner steel frame for stability and can be changed out by the quickly by disassembling the lid. The final design is shown in *Figure 2.5*.

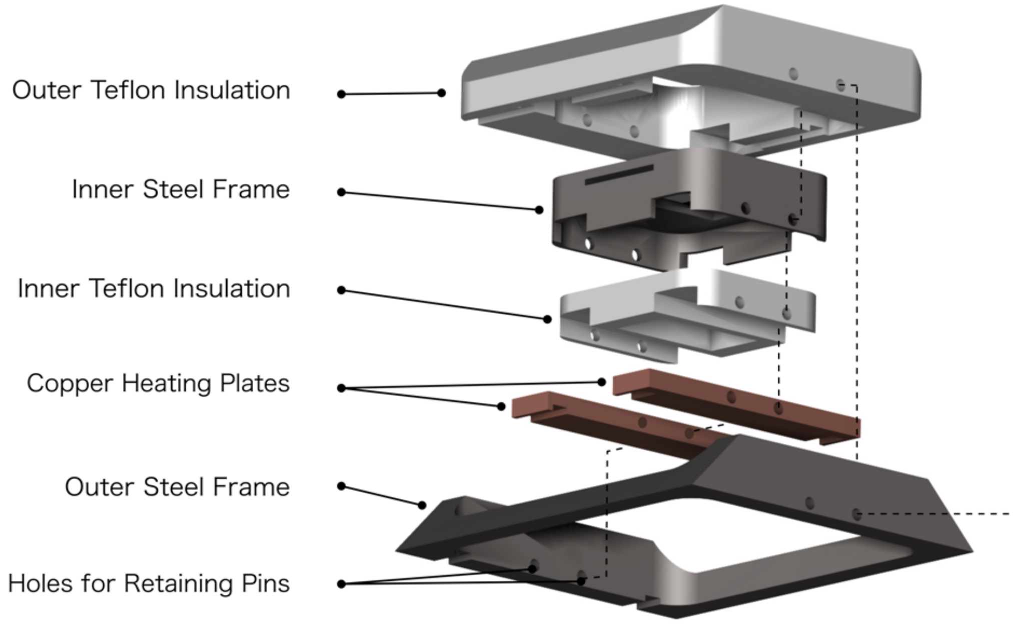


Figure 2.5: Exploded view of the lid of the heat stage. The dotted line shows the path of one of the two threaded pins which hold the lid together once it is assembled. This allows for quick disassembly and reassembly when changing out the viewing window

Heat Stage Body

The main body of the heat stage was designed to provide ample thermal protection from the laterally connected positional stages (*Figure 2.3* and *Figure 2.7*. In addition, the positional stages attach to the sides of the heat stage by being placed vertically downward onto dowels (*Figure 2.4* which lock into place by spring-activated locking mechanisms built into the positional stages (*Figure 2.2*. The lids of the positional stages then act as guides for the lid of the heat stage which slides into place from the front and locks into place with a spring-loaded detent button located on the lid of the left positional stage. Copper rods inserted through the Teflon wall of the main body conduct heat from the lower heating plate (heated by the two cartridge heaters) to the lid's two copper heating plates (*Figure 2.6*). The upper heating places have guides cut into them so that – once the lid is slid into place – they engage positively with the vertical copper rods which protrude from the top of the main body by 1mm. The lower steel plate of the main body can be unscrewed from below to gain access to the lower glass viewing window and the heating elements. The final design of the main body of the heat stage is shown in *Figure 2.6* and *Figure 2.7*.

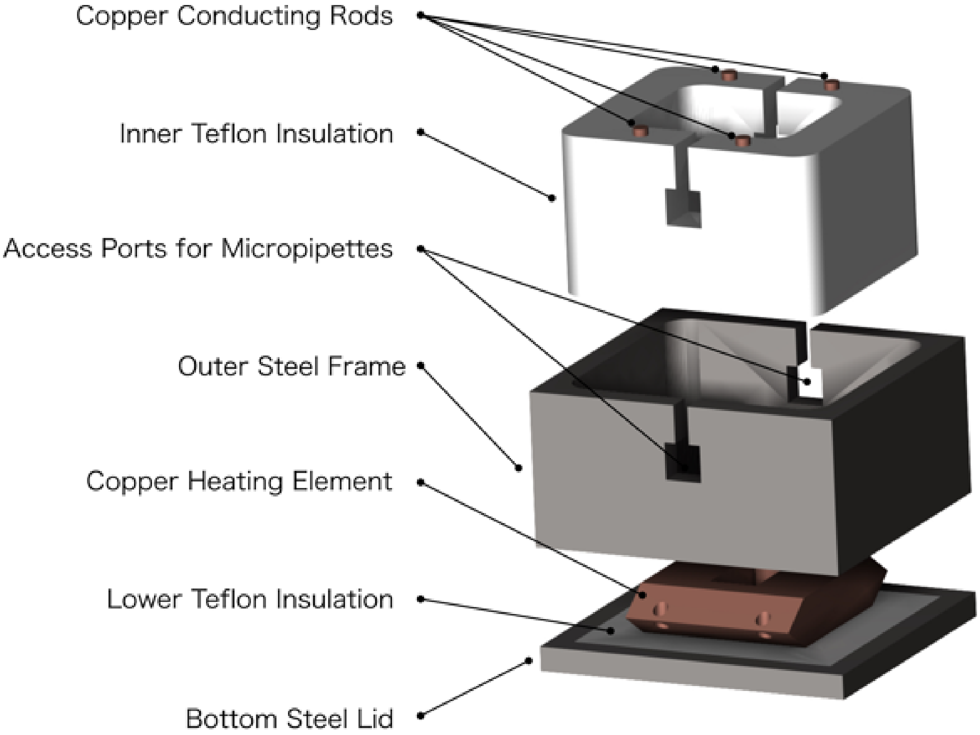


Figure 2.6: Exploded view of the main body of the heat stage

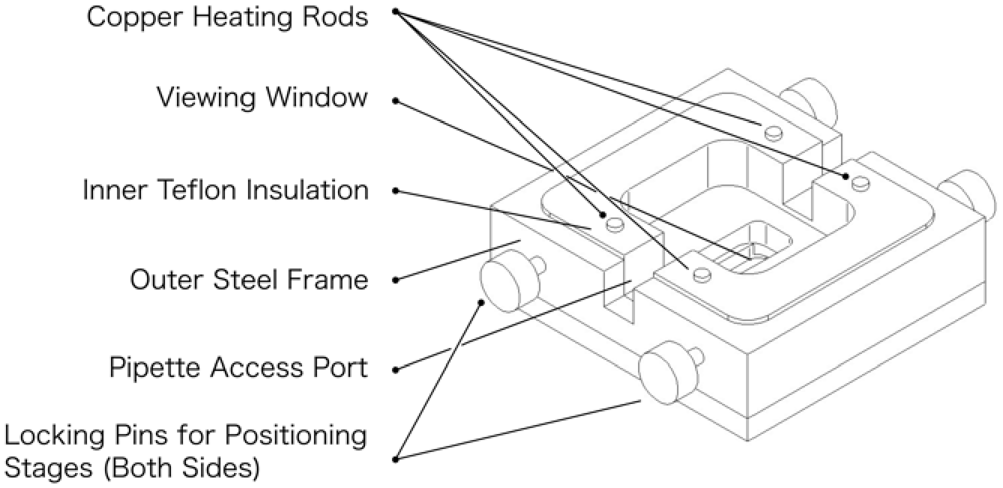


Figure 2.7: Detailed sketch of the heater main body

Thermal Simulation

Using Matlab and applying the finite difference method, the heat equation

$$\frac{\partial u}{\partial t} - \alpha \nabla^2 u = \frac{1}{c_p \rho} \tag{2.1}$$

was solved in three dimensions in rectangular coordinates over a computational space of real units 600 x 600 x 600 mm with boundary conditions fixed at room temperature (20 °C). The heat stage itself was placed at the center of this space with dimensions 65 x 65 x 25 mm. Grid spacing was chosen to be 1 mm in all directions. The geometry of the heat stage and the material properties of its individual components were included in the numerical calculation on *Lines 13 - 121* in *Section A.1* of the *Appendix*. The material properties are present in *Equation 2.1* as a quantity known as the thermal diffusivity

$$\alpha = \frac{\kappa}{c_p \rho} \tag{2.2}$$

where κ is the thermal conductivity, ρ the mass density and c_p the specific heat capacity of each component. The term on the right side of *Equation 2.1* is equal to the internal heat generation (q) and is only present in the regions of the heat stage where the two cartridge heaters are inserted into the lower copper heating plate (*Lines 125 - 138* in *Section A.1* of the *Appendix*). This heat generation is in units of energy per unit volume per unit time, or Watts. Each cartridge heater was rated at 80W, meaning that the total volumetric energy generated by each heater (given the volume of each cartridge) was 30 MW/m^3. This internal generation defines the boundary condition at the surface surrounding the cartridge heaters. Everywhere else, the reduced heat equation

$$\frac{\partial u}{\partial t} - \alpha \nabla^2 u = 0 \tag{2.3}$$

is applied since power is not generated anywhere else in the heat stage. These equations, written in finite difference notation as taken from the Matlab code, take the form

$$\begin{aligned}
H(i,j,k) = {} & (delX * delX/(delX/(2*lambda(i,j,k)) & (2.4)\\
& + delX/(2*lambda(i-1,j,k)))) * (T(i-1,j,k) - T(i,j,k)) & (2.5)\\
& + \text{the 5 remaining lattice terms} & (2.6)\\
& + (I(i,j,k) * (delX^3)) & (2.7)
\end{aligned}$$

where $H(i,j,k)$ is the change in heat at each lattice point (i,j,k) due to the heat of each of its six neighboring lattice points. *delX* is the grid spacing (1 mm) in all directions and *lambda* is the linear conduction coefficient of the material at point (i,j,k).The temperature at each point in space after a small time step *delt* was then calculated by

$$T(i,j,k) = T(i,j,k) + H(i,j,k) * delt/(C(i,j,k) * delX^3) \tag{2.8}$$

where $T(i,j,k)$ on the left is the updated temperature at point (i,j,k) after time *delt* and $C(i,j,k)$ is the specific heat of the material at point (i,j,k). $T(i,j,k)$ was initially set everywhere at 20 °C and was updated for all points except for those at the outer boundary of the computational space. The outer boundary was then held fixed at room temperature 20 °C by re-initializing those points after every iteration. The system was allowed to relax until the temperature change after the time interval *delt* (30 seconds) fell below a reasonable threshold indicating the temperature field had reached equilibrium. It was shown that equilibrium could typically be reached within about five minutes when ramping from room temperature at full power (160W combined) to a maximum of just over 300 °C. The final result of the simulation (*Figure 2.8*) shows that during the ramping process and after reaching equilibrium, the temperature gradient in the region of interest (sample region) remained at virtually zero. The strongest regions of temperature leakage showed to be the thin glass windows which agreed with physical intuition. The full Matlab code can be found in the *Appendix* in *Section A.1*.

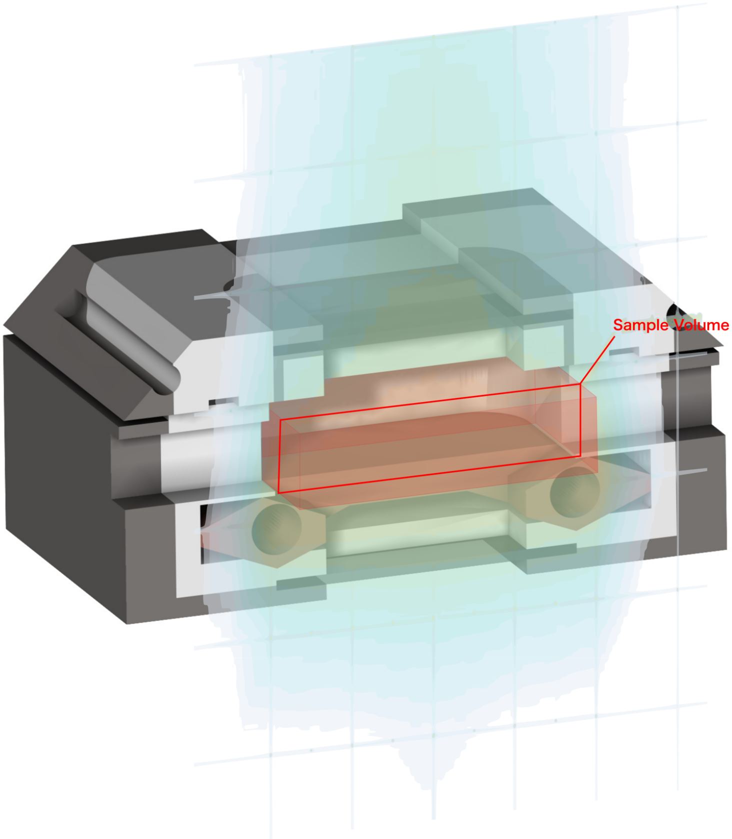


Figure 2.8: 2D slice of the 3D heat conductivity simulation results overlaid atop a cross-sectional view of the heating chamber. Grid lines are spaced by 5 cm. The red section represents the sample volume.

2.1.2 Temperature Controller

A suitable temperature controller was required which would allow simple GUI control, high frequency PID control, data logging and ramp/soak rate programmability. The ”Platinum Series” PID controllers from Omega Engineering provided a USB interface, pulsed 5V output for control of an external relay and free software and a GUI interface to easily configure, program and manage the controller. Auto-tune PID initialization and other features made the controller very user-friendly. In order to read the temperature of the sample region inside the heat stage, a T-type thermocouple from Omega was inserted through a 1mm hole in the rear of the heat stage's main body and its tip positioned near the sample volume without interfering with the sample or the pipettes. The two 80W cartridge heaters used to heat the stage were wired in parallel and powered directly by a 220 VAC source which was routed through a solid state relay. The temperature was controlled by the PID algorithm of the controller used a 5 VDC pulse to a the relay to switch the 220 VAC source at the correct duty cycle required to ramp to, and maintain, the preferred temperature based on the process value (PB), i.e. the reading from the thermocouple. PID settings are typically stable within a $10°C$ window about the set point (SP). Parts specifications can be found in *Table B.2* in the *Appendix*. The heat stage was fully built and calibrated before the positional stages were completed allowing time for its performance to be optimized. A picture of the assembled temperature controller and heat stage are shown in *Figure 2.9*.

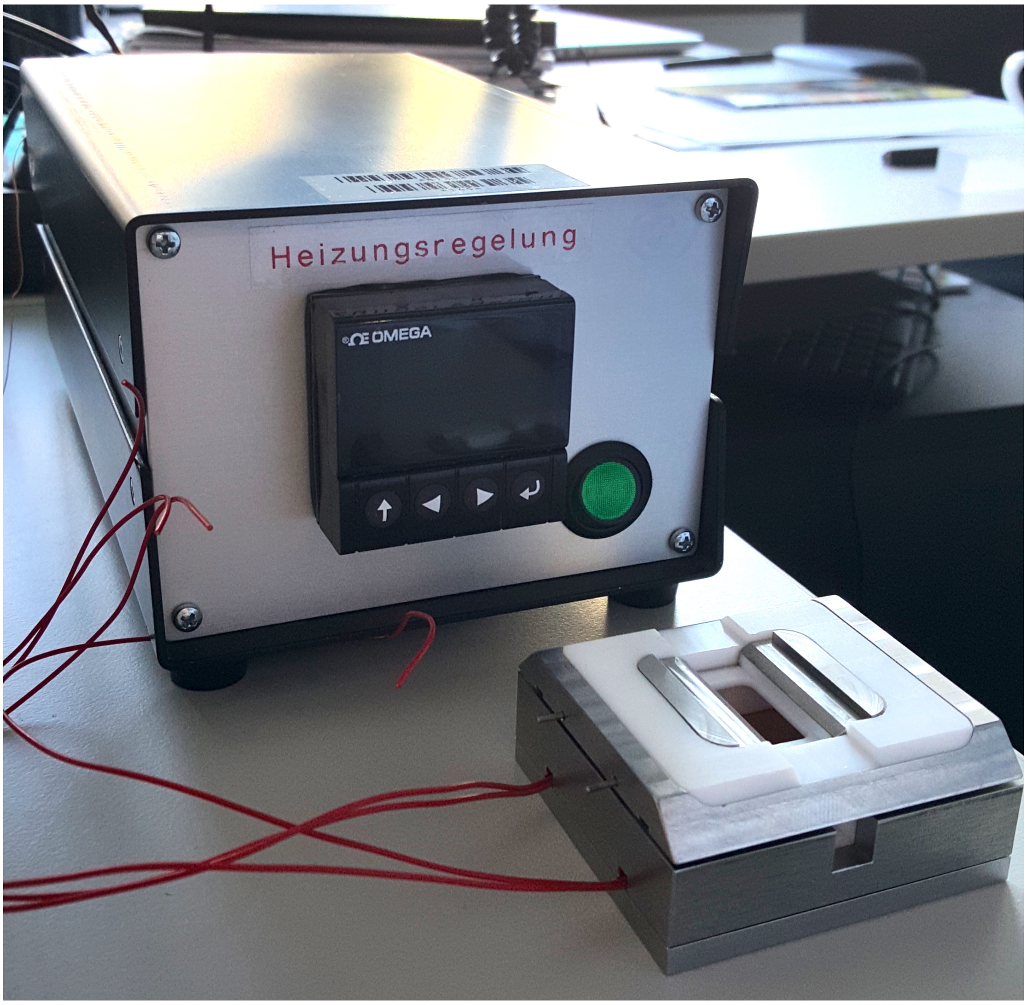


Figure 2.9: The Omega PID and sold-state relay for temperature control built into a housing sitting along side the completed heating stage. The top of the heat stage is machined in a way to allow microscope objectives to achieve a close working distance.

2.1.3 Positional Control

All components of both positional stages – with the exception of the piezo stage purchased from Physikalische Instrumente (PI) – were designed using Solid Edge 7.0 CAD software and were machined in-house with the guidance of Wolf Keiderling. They were designed to be quickly detachable from the heat stage using spring-loaded locking mechanisms to positively engage steel dowels built into the heat stage (*Figures 2.2* and *2.7*) and include modified Narishige pipette holders for quick replacement of the glass micropipettes. Some major initial design requirements included:

- Motorized axes in the horizontal plane, motorized rotation of the glass rod and manual z-positioning
- Invertible for use in upright and inverted microscopes
- Compact enough to fit on most microscope stages and to be used across different labs.
- Modular design for tool-less removal from the heat stage for quick replacement of glass pipettes and easy modification for future upgrades
- Sealed to keep dirt out
- Joystick control
- Stage motion driven by small DC motors integrated gear reduction for smoother motion, slower movement speeds and increased torque.
- Precise control and programmability of sample deformation for rheological studies

The first goal for designing a positional controller was to allow a range of motorized motion in all three orthogonal axes as well as in rotation. These axes are labeled as X (the axis parallel to the axis of material extension), Y (the axis perpendicular to X and the optical axis of the microscope), Z (the axis parallel to the optical axis of the microscope) and ϕ (the rotational degree of movement about the X-axis as shown in *Figure 2.2*.

The second goal was for the hardware (translational components and motors) to be sealed in an enclosed housing (translational stage) which would be compact enough to be fitted directly and tightly to the heat stage (sample testing region) using a tool-less locking interface. This would ensure that the entire rheometer could be assembled as a single, compact unit and placed on the microscope stage as a whole to ensure positional and experimental consistency while keeping dust out of the fine gears of the positioning stage and contributing to the thermal insulation of the rheometer as a whole.

The size restriction of the translational stages imposed many design restrictions on the internal hardware itself. First, no known off-the-shelf motorized multi-axis stage was compact enough or provided the ranges of motion necessary to provide large material strains. Moreover, traditional stepper motors were too large. The solution was to use small DC motors and then custom-design the necessary gearing needed for increased torque and accurate control. Although this method meant ruling out closed-loop control, the movements would be controlled directly by the user via a multi-axis joystick and used only for experimental initialization and preparation of the sample before precise, closed-loop movement was provided by the piezo stage located in the opposing side of the heat stage. In order to fit the components of the positional stage into the steel housing, the Y- and Z-axis were restricted to 3.5 mm of movement in each direction from center (zero), or 7mm in total range. The X-axis was restricted to 5 mm in each direction from center (zero) or 1 cm in total range. Rotation in ϕ was unrestricted and was allowed continuous and unlimited rotation through all angles.

Figure 2.10 shows a top-view of the initial design of the rheometer showing the details of the positional stages on either side of the heat stage. After the CAD designs were completed, nearly 100 unique pieces – from everything to the gears, races, housings, axles, and locking mechanisms were machined by the in-house machinists at the MPIDS .The initial design used one of these DC-motorized stages on either side of the heat stage for manipulation of both the force-sensing pipette and the rigid puller pipette. The current revision only uses the DC-motorized positional stage to control the position of the force-sensing pipette (*Section 2.2*) and to allow precise, joystick-controllable manipulation of the relative position between the two sample surfaces in order to easily form the material's initial geometry; in this case, a liquid crystal fiber as it did did provide the closed-loop control, programmability and repeatability required for the precise strain and strain-rate profiles needed for testing the material's elastic and viscous response. Instead, the second translational stage (the one which controlled the rigid puller pipette) had to be altered by removing the internal components and replacing them with an appropriate linear stage which fulfilled the following requirements:

- Long working range of 6 mm total travel distance, or 3.5 mm in each direction from center
- High translational speed of up to 10 mm/s
- Optical closed-loop positioning feedback with 2 nm resolution
- Minimum step-size of 4 nm
- Full *SDK*[2] package to allow for customizable programming and interfacing in C++
- Compact size to fit inside the stage housing
- Robust against increased ambient temperatures and mechanical wear
- Minimal transferred vibration to the sample during translation

The chosen linear stage was a piezo stage by PI and was installed in the rheometer as shown in *Figure 2.2* where it holds a straight, rigid glass pipette of similar tip diameter to the flexible force-sensing pipette. Both interfaces (the force-sensing pipette held by the multi-axis joystick-controlled stage and the straight, rigid pipette held by the piezo stage) are fastened into place to their respective stages by modified Narishige quick-connect pipette holders (*Figure 2.15* in *Section 2.2*). This allows easy exchange of the pipettes from both stages by simply rotating the tip of the Narishige pipette holder and removing/replacing the pipette while the pipette holders themselves remain fixed (*Figure 2.2*).

[2] Software Developer's Kit

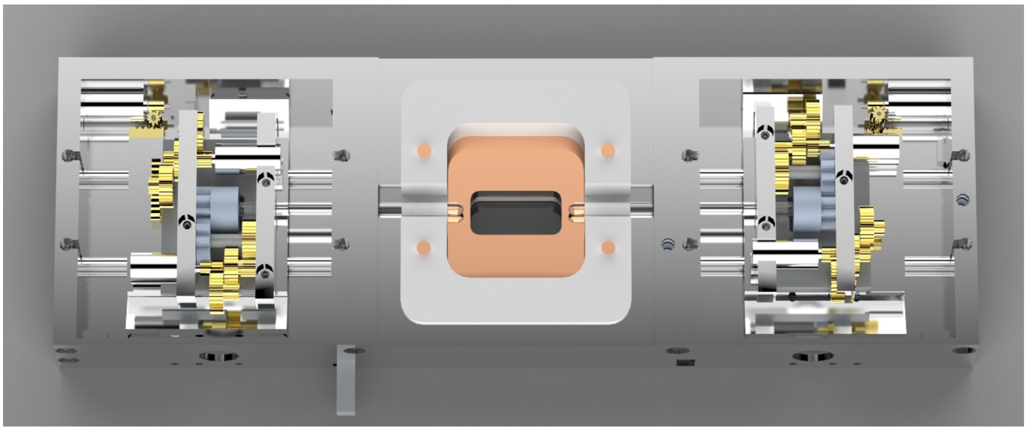


Figure 2.10: Final draft of the CAD drawing of the complete device (top view) with stage covers and heat stage lid removed.

The joystick controller for manipulating the DC-motorized stage was built using a potentiometer-type joystick to control Arduino Uno boards with Motor Shields. Arduino (Objective-C) code was written to control the DC motors using pulse width modulation (PWM) which provided the motors with a reduced mean voltage by changing the duty cycle (on-time versus off-time) of a constant 9V DC relative to the position of the joystick from its zero-position. The joystick at zero-position corresponds to a duty cycle of 0% while a joystick at the end of its range corresponds to a duty cycle of 100%. The range in between is typically linear. The Arduino boards, wiring and joysticks were installed into a hard plastic housing. The final controller can be seen in *Figure 2.11*. In the rear of the housing are 3 USB ports (one for each board), a power connector (for the motors) a self-made 15-pin D-SUB connector to send and receive signals from the motors too all 3 Arduino boards via a self-made cable. Parts specifications can be found in Table B.3 in the Appendix.

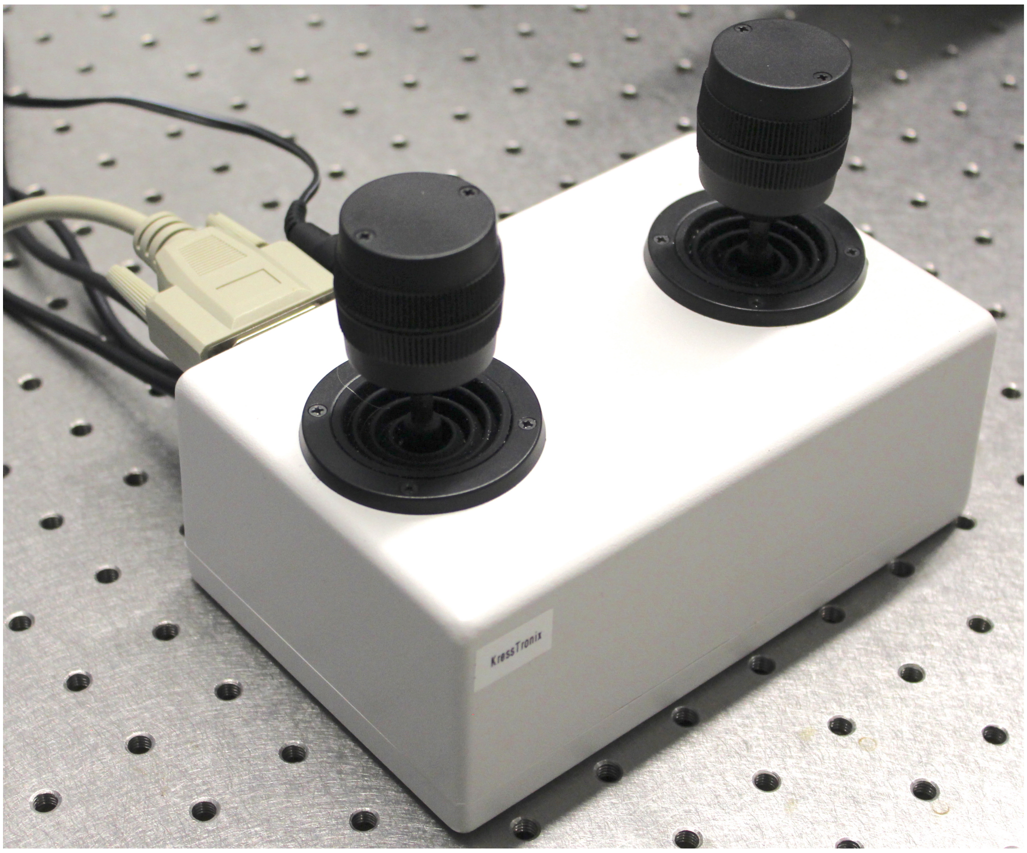


Figure 2.11: Joystick controller built with 3-axis Joysticks and Arduino boards. Housing includes output to motors, power, and USB for Arduino programming

2.1.4 Additional Laboratory Hardware

The microscope which was used was an upright Mitutoyo optical microscope fitted with long working distance objectives of 2X, 5X, 20X and 50X magnifications. This microscope is was the most basic manifestation of a microscope, in that it contained only the most essential optical components. No condenser was included, so one had to be designed and built. Though rudimentary, the microscope's objectives meant they could remain at a safe distance from the heat stage to avoid heat damage or distortion of the optical components (temperatures up to 200°C) and image while its simple construction made it easy to modify.

The sample was illuminated by a 150W halogen source coupled to a fiber optic cable. The original light source included a single-capacitor AC/DC filter which resulted in extremely noticeable intensity fluctuations at frame rates exceeding 40 FPS which made it unusable for even the most modest high-speed imaging. Therefore, the light source was modified and wired directly to an external laboratory DC power supply while retaining current and temperature overload protections. The light was sent through a green light filter to provide monochromatic light to the sample and help to increase contrast and reduce chromatic aberrations and focal depth.

Illumination

In order to properly image the same, open-source ray-tracing software (*Figure 2.12*) was used to design a condenser which mounted directly to Mitutoyo microscope using a standard 1-inch optical cage format. The green light source from the end of the fiber optic cable was fed first into colimating lens which was NA matched to the output of the fiber optic cable to optimize light collection. This parallel beam was then passed through an inverted aspherical lens to converge the beam onto a small aperture. The beam was then sent through a series of biconvex, biconcave and doublet lenses in order to reduce and re-colimate the beam before it was reflect by a mirror placed directly below the sample plane of the microscope. The beam was then reflected upwards towards the sample and send through an achromatic doublet which converged the lens at a focal plane which coincided with the sample plane. Horizontal movement of the components between the initial colimator and the mirror as well as control of the aperature made it possible to adjust the focal plane and NA to match the 5X, 20X and 50X objectives in order to increase transmission into the objectives while the narrow focal depth provided strong contrast. The cage design remained modular so that the inclusion of additional optical components, such as polarizers or liquid crystal retarders, could easily be installed.

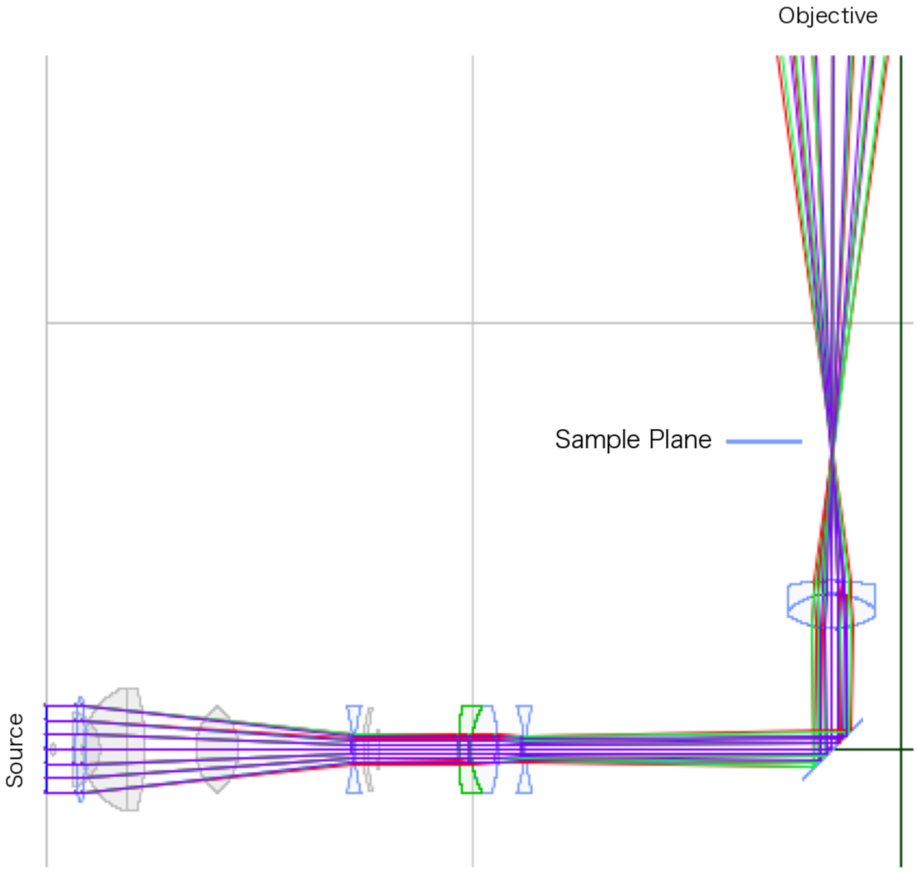


Figure 2.12: Ray trace of the final colimator design. The horizontal components could be slid along the optical cage to change focal plane and the effective NA at the sample plane

Image Capturing

In order to record the motion of the micropipettes and to observe the material's response, a suitable digital imaging device would need to be installed on the microscope. A Point Grey Research Grasshopper digital camera was chosen since it and its FlyCapture software package, met the following requirements:

- Easily mountable to the vertical viewing port of the microscope
- USB3 interface for simplified hardware control via an SDK package and fast data transfer rates
- At lest 4 MP resolution and relatively small pixels
- Absence of an internal fan for quiet operation
- GUI capable of simple control and viewing

2.2 Force Sensing

Once the heating stage was shown to provide the necessary temperature stability required to study the highly-sensitive behavior of liquid crystal fibers and the positional stages shown to operate with the precision and repeatability needed for the acquisition of high-resolution measurements, the design and implementation of an appropriate force sensor followed. Inspired by the implementation of a technique called Micropipette Force Spectroscopy (MFS) by Kreis and Makowski in the study of the adhesion and viscoelastic response of microorganisms, respectively, it was discovered that a simple modification of the pipette geometry was enough to integrate them into the rheometer's pre-existing design [53].

The idea was simple: to place a small sample of the liquid crystal mesogen between two glass interfaces, one of which was capable of deflecting in response to the movement of the other. One interface would be flexible with a known spring constant and provide an optically observable deflection of the glass/sample interface while remaining fixed at its base. From this deflection, the force acting on the pipette could be measured. The other interface would be a straight, rigid glass rod with the same tip diameter as the tip of the flexible pipette. This interface would be used to strain the material inducing a force[3] response. Some requirements for the design of the micropipettes were that

[3] Or stress response if the diameter of the fluid sample could be resolved

- Micropipettes should be easily exchangeable for cleaning, changing to a different spring constant, or replacement should one break

- Pipette geometry should be such that the sample material/pipette interface remains coaxial with the pipette holder and the opposing rigid interface used for providing material strain

- The tips of both interfaces should be fused closed to prohibit fluid from retracting into the interior by capillary forces.

- Precise control of the micropipette diameter and cantilever arms during the forging process result in a wide range of achievable spring constants.

- Relatively large linear regime of the calibrated spring constant.

- Compatible with a simple yet robust calibration procedure which can be implemented from within the rheometer

The manufacturing of a micropipette cantilever began by pulling a 1mm OD borosilicate glass capillary rod using a Narishige pipette puller until the glass rod reaches a diameter of 20 to 80 μm, depending on the final spring constant desired[4]. Once a suitable pipette had been pulled, it was then transferred to a Narishige Micro Forge when it was bent – by hand – into the correct geometry using a heating wire to locally soften the glass and a rigid wire hook – or forming wire – to gently apply pressure and deform the pipette as desired. A brief outline of this process is shown in *Figure 2.13*. The process for forming the rigid pulling pipette was much simply, since only the first step – the pulling of a straight pipette – was required. After both pipettes had been formed, their tips were sealed shut using the heating wire of the micro forge in order to keep the liquid crystal material from being pulled in during experiments.

Different shapes of the tips of each pipette were produced in order to examine the possible effects of geometry on the boundary conditions of the fluid sample (*Figure 2.15*). The final and most vital step – calibration of the force-sensing pipette's spring constant – was done using the methods outlined later in this chapter in *Section 2.4* titled *Calibration Technique*. *Figure 2.15* shows a sketch of the pipettes mounted in their respective pipette holders – each mounted to their respective positional stage[5]. While no quantitative comparison was made to assess the benefits of different tip geometries on the experimental procedure or its outcomes, no clear differences in behavior were observed. In the case of the liquid crystal used here, the boundary conditions played a minute role as most of the behavior was confined to the fiber itself. For experiments where boundary behavior becomes more relevant such as in polymer melts, these different geometries may play a role in how sheer velocities near the interface affect the rheological behavior.

[4] It was typically desirable for the thinned segment of the pipette to have a as-near-as-makes-no-difference constant diameter throughout its length in order to make bending the pipette more controllable

[5] The rigid puller pipette was mounted to the PI piezo stage while the force-sensing pipette to the multi-axis joy-stick controlled stage so that upon straining the fiber using the piezo stage, the force-sensing pipette remained fixed and its deflection could be measured.

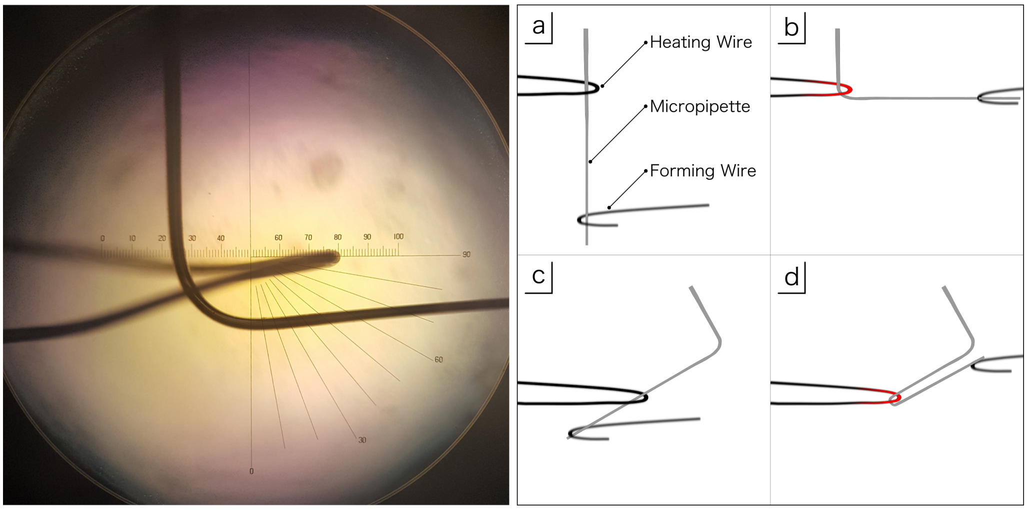


Figure 2.13: A micrograph of a pipette being bent into shape using the Narishige Micro Forge. The minor tick marks are spaced by 10 μm. To the right, the sequence of steps used to form the micropipette. **(a)** Placement of the glass pipette between the heating wire and forming wire, **(b)** heating of the glass and bending via the forming wire, **(c)** repositioning fo the second bend and **(d)** heating and bending into the final U-shaped force-sensing pipette

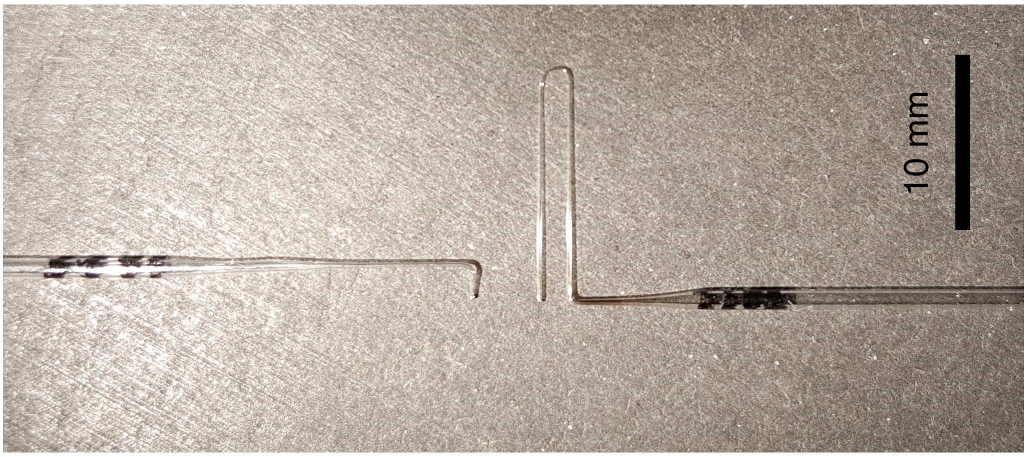


Figure 2.14: Example of a rigid puller pipette (left) and a force-sensing pipette (right)

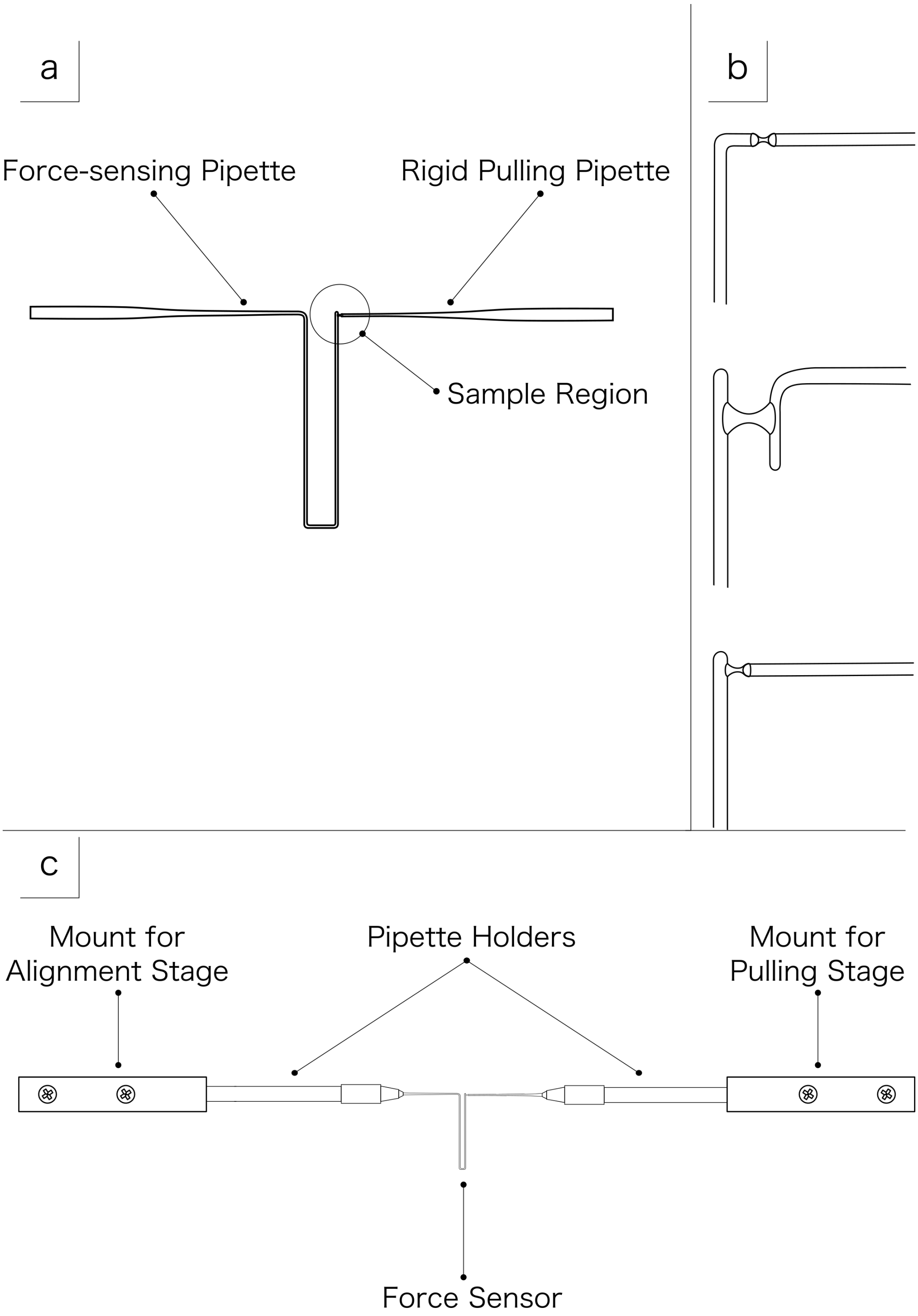


Figure 2.15: **(a)** A sketch of the geometric arrangement of the sensing pipette, the puller pipette and the sample region, **(b)** sketches of the micropipettes and the different tip geometries that were made to address different boundary behavior of the fluid sample between them and **(c)** the of the micropipettes in the pipette holders which were fixed to their respective positional stages.

2.3 Software

Throughout the development of the rheometer, new software had to be written for every aspect of its functionality. This software includes

- Control of the custom-built multi-axis positional stage via multi-axis joysticks
- Control of the piezo linear stage (PI) for rheologically relevant movement profiles
- Image analysis code for detecting pipette deflection for measuring the force response and for measuring fiber diameter (later in real-time)
- Data analysis code for filtering, extracting, analyzing and compiling the rheologically relevant experimental and material parameters
- Code for combining the operation of the positional stage, heating stage and camera into a single, C++ rheometer program for real-time rheological analysis

2.3.1 Joystick Control

The multi-axis stage did not required closed-loop control or high-resolution positioning. Rather, it was important that the user could easily move the force-sensing interface in any direction, with minimal backlash, in a predictable, tactile, immediate manner while observing the movement through the microscope's eyepiece or on a computer screen via digital imaging. The obvious solution was to implement 3-axis joysticks. These joysticks allow lateral movement of the pipette in the focal plane by lateral movement of the joystick and rotation of the pipette by twisting of the joystick. The motors used to move the components of the stage itself were very small DC motors with low torque and rather high RPM. By building a custom stage with custom gear-reduction of 30:1 suited specifically to these motors, the movement control could be refined to provide smooth, continuous motion with sub-micron displacements easily achievable by joystick. The motors and joysticks were interfaced using a set of Arduino UNO boards coupled with Arduino Motor Shields. The Arduino Uno boards provided the programmability and interfacing required to fine-tune the control of the motors, while the Motor Shields allowed external power to be routed to the motors via typical H-bridge transistor arrays for bipolar (multidirectional) DC motor control. The control method used was a pulse-width-modification (PWM) technique coupled with I2C protocol. The joysticks are of the potentiometer type, so their movement results in a voltage change which is translated by the Arduino UNO into a PWM duty cycle which in turn results in modulation of the motor rotational speed. Some requirements of the joystick control were

- Fine motion of the positional stages
- Signal dead-zone around the joystick zero-position to filter out noise or voltage offset
- Removal of signal cross-talk between stage axis
- Non-linear speed-control for slow and precise movement during small joystick displacements and large movements and speeds for large joystick displacements

2.3.2 Piezo Stage Control and Rheological Algorithms

This sub-section will be broken down into further sub-sections, each describing the algorithms used to control the PI piezo stage and for which rheological experiment the algorithm could be used for.

Constant Strain Rate

The NExt rheometer was first and foremost designed to attempt to improve upon – and directly compare the results to – measurements of extensional viscosity such as those those done by McKinley *et al.* at MIT [55–60], Sridhar *et al.* at Monash University in Australia [45,61] and Szabo at Technical University of Denmark [56,62]. A defining challenge shared with those who designed extensional rheometers in the past was to impose a large range of constant constant strain rates over large lengths. In order for the length of the fiber to change at a constant strain rate, the displacement profile needs to evolve in time as

$$L(t) = L_0 e^{\dot{\varepsilon} t}$$

or the velocity as

$$V(t) = L_0 \dot{\varepsilon} e^{\dot{\varepsilon} t} = \dot{\varepsilon} L(t)$$

where $L(t)$ is the length of the fiber at time t, L_0 its initial length and $\dot{\varepsilon}$ the strain rate as was derived in *Section 1.4* titled *Modern Extensional Rheology*, *Equations 1.23* and *1.24*, respectively . Because the PI piezo stage and controller were not ”capable of such movements, since the controller is not compatible with movement profiles beyond simple linear displacement,” [6] C++ code had to be written (*Section* A.4 in the *Appendix*) in order to move the stage according to *Equation* 1.23. Once the initial length (L_0), the desired maximum strain (ε, where entering ”0” means pulling until the stage's limit) and strain rate (*sRate*) are entered by the user (lines 7 - 17) the program starts a

[6] Paraphrased email response from the technical representatives at PI when asked if the controller could move the stage in any way besides from point A to point B at a constant velocity

while loop under the condition that the maximum velocity and position do not exceed the stage's limits. Because the code was not uploaded directly to the controller, but rather had to run on the computer and send the new position to the controller at just the right time in order to produce the desired position and velocity profile, the closed-loop capacity using the optical encoder built into the stage could not be used; the time it took to query the stage's position was greater than one second, making a closed loop approach impossible. Therefore, timing became very important in writing a reliable open-loop program. The position of the stage was controlled by calculating the new length

$$Lnew = L0 * exp(sRate * (tStep * (dT))) \tag{2.9}$$

followed by the new velocity

$$Vel = (Lnew - Lold)/dT \tag{2.10}$$

and converting the new length into the new position

$$Pnew = Pinit - (Lnew - L0) \tag{2.11}$$

before updating the value of the new length for the next iteration by stating that

$$Lold = Lnew \tag{2.12}$$

where the initial parameters are

$$Lold = L0Pinit = InitialPosition \tag{2.13}$$

and *InitialPosition* is the initial position of the controller. *tStep* is incremented by 1 at the end of each loop, and *dT* is set to 0.05 seconds at the start. Assuming the stage has infinite acceleration between steps, the time it takes the stage to get from to each new position can be considered constant for each step and be equal then to *dT*. Because the stage cannot accept new position requests from the controller until the stage has stopped moving and then when ready receives only the most recently calculated position and not the next-in-line, a *Sleep* function had to be added at the end of the loop to pause the code until the new position was reached and only then calculate the new position and send it to the eagerly-awaiting controller. The parameter for the sleep function was then *dT*, or the time it should take the stage to reach its new position. Because of delay time to communicate with the controller and to calculate the new position, a delay time of around 8ms was subtracted.

The final movement of the stage showed to fit exponential displacement perfectly, with only a difference between the user's desired strain rate at the resulting strain rate, which could later be adjusted and calibrated by changing the delay time parameter. When the velocity reached the stage's maximum (10 mm/s), the stage stopped moving. When the stage neared the end of its travel before reaching its maximum velocity, it would decelerate in order to prevent end-of-travel damage. A slight difference in performance resulted in different initial lengths. The performance of the stage and the program `exp_vel_pull.exe` (*Section* A.4 in the *Appendix*) is summarized in *Figure 2.16*.

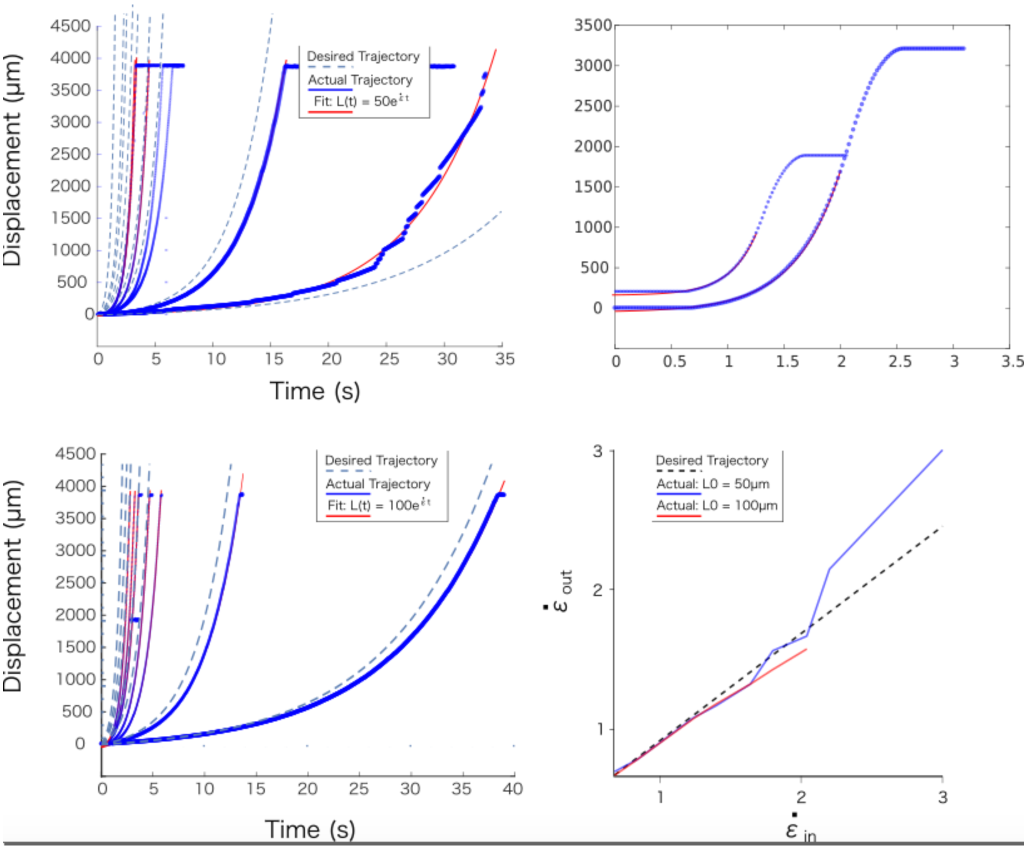


Figure 2.16: Top left: Dotted lines indicate desired trajectories with strain rates from 0.1 to 3 and an initial fiber length of 50μm while the blue points indicate actual positions and the red lines the fits to the data. **Top right**: Examination of the two fastest strain rates: the stage approaches its end-of-travel with a velocity above a safe threshold and decelerates, limiting the range where the exponential pulling takes place. **Bottom left**: Similar to the plot above it, but for $L_0 = 100\mu m$. **Bottom right**: Plot showing how well the input strain rates compare to the output strain rates

Step-strain

The piezo stage from PI interfaced to the PC using an intermediate controller. This controller was very limited in that the user interface provided allowed no programming and only allowed single steps. This would technically have been enough for step-strain experiments, but since extensional viscosity and oscillatory experiments were also planned, it made sense to write a program for the step-strain experiment as well. This C++ program (`PI_step_strain.exe`) was simply written to produce a single step of a length calculated from the user's input of the fiber's initial length (L_0) and desired strain magnitude (ε_0). The result of the piezo stage's attempt to do this, superimposed on the desired result, is shown in *Figure* 2.17. The velocity of the stage was set to its maximum of 10 mm/s, although this rate was not reached due to the nanometer scale of the step and the limit of the stage's maximum acceleration. The code can be found in the *Code Snippets* section *Section A.4* in the *Appendix*.

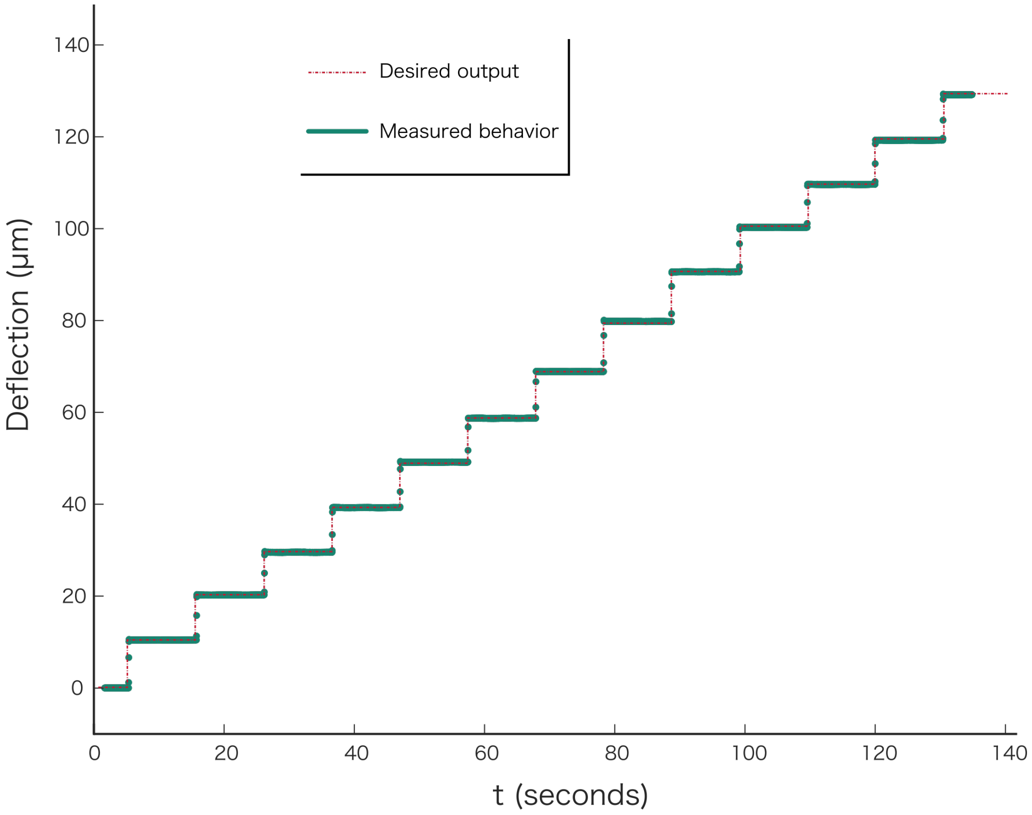


Figure 2.17: A series of single steps done by the PI piezo stage according to the C++ program `PI_step_strain.exe` to verify stability and time accuracy between steps

Step-stress

Applying a constant step stress to a fiber under uniaxial extension while allowing the strain to ”creep” is a challenge which requires a precise closed loop control mechanism which adjusts the length (strain) of the fiber while it is creeping in such a way that the stress remains constant [59,62]. In order to implement this in the new rheometer, the SDKs of the Pointgrey camera and the PI stage were merged using C++ and an open source image processing library called OpenCV (Open Computer Vision). The code allowed the closed loop control of stress and strain by calculating the instantaneous stress by measuring the deflection of the force-sensing pipette and the fiber diameter. The puller pipette's position is then adjusted to keep the stress constant during creep – the relaxation of the strain.

Triangular Excitation

Triangular excitation was done as either a single triangular wave or series of waves. Since triangular excitation can be considered as a constant strain rate extension up until some infinitesimal strain limit, followed immediately be a compression back to the initial fiber length also following a constant strain rate profile, the same displacement and velocity profiles were used as discussed in the *Section 2.3.2* and expressed by *Equations 1.23* and *1.24*. The code implements the algorithms for positive strain using the constant rate code shown in *Section A.4* followed by iterations backwards in time for the compression cycle as shown in *Section A.4* following the algorithms

$$Lnew = (slope/sRate) * log(((tStep) * dT * slope + b)/L0) + b \tag{2.14}$$

and

$$Vel = (Lold - Lnew)/dT \tag{2.15}$$

where *sRate* is the user-defined strain rate, *tStep* is iterated backwards from the final time integer reached during the extensional cycle of the triangular wave. dT is typically set at $5ms$ and $b = L0$ where $L0$ is the user-defined initial length of the fiber. $Lold$ is the length of the fiber during the previous iteration and $slope = (Lnew - L0)/((tStep + 2) * dT)$ is the instantaneously slope of the exponential decay profile.

Sinusoidal Excitation

In order for small and large amplitude extensional oscillation experiments to be performed, a suitable algorithms – or set of algorithms – had to be implemented across different amplitude and frequency ranges to optimize the functionality of the PI stage. A balance had to be found between algorithmic complexity, signal delay time, amplitude and frequency. For larger amplitudes where deviations from the sinusoidal profile would introduce higher frequencies in the excitation spectrum, the sinusoidal profile was calculated using the native C++ math function and iterated forwards in time as shown in *Section A.4* of the *Appendix* chapter titled *Code Snippets*. For higher driving frequencies where the time to calculate the full sine function was too great as compared to the driving period, shortcuts were taken to decrease computation time. These included a truncated

Taylor expansion which was calculated for 8 separate segments of the sine wave if the amplitude was large and accuracy was still a priority (*Section A.4*), or by approximating the sine wave as a triangular wave if the amplitude was small taking advantage of the intrinsic acceleration and deceleration of the stage to ”round out” the tips of the triangle (*Sec, A.4*).

2.3.3 Data Acquisition and Image Analysis

The original Matlab code written by to track the deflection of the micropipettes used to measure the adhesive behavior of *Chlamydomonas* utilized a common autocorrelation-based method complimented by spline fitting the correlation function of an intensity profile drawn across the deflection path of the pipette for sub-pixel displacement measurements.

A similar method was previously described by Kim *et al.* in 2011 [63]. This code was re-written in order to optimize it for use with the rheometer while retaining the fundamental pipette-tracking algorithm. The code was later written in C++ in order to optimize performance and merge the SDK of the camera, the heat stage and the piezo stage to build a comprehensive rheological software package which could provide real-time sample viewing, rheological analysis and plotting. Individual programs were written for each type of rheological test (constant strain-rate, sinusoidal excitation, step strain) in order to analyze the results. The closed-loop feedback from real-time image tracking and motor control even allowed for step stress – or creep tests – where the stress of the fiber could be held constant while the strain was allowed to ”creep,” or relax. The following outlines the vital components of the Matlab code used for prototyping and the development of a real-time, C++/OpenCV based cross-platform software package which offers many advantages over the Matlab scripts.

The Matlab script used the Matlab function `xcorr.m` to autocorrelate the intensity profile across the glass micropipette. As the pipette moved, its unique intensity profile moved with it. The movement was tracked by following the spacial shift of the autocorrelation peak against that of the first frame of a video sequence. The maximum value of this peak was still limited to pixel resolution so a 5th degree polynomial was fitted to the peak and the new peak height was taken as the peak of this fit. Therefore, the peak shift could be defined by sub-pixel displacements. Then, the displacement was multiplied by the spring constant – denoted k_{pip} of the micropipette to extract the force applied to it. More on the tracking procedure of the pipettes is discussed in the *Methods* chapter in *Section 4.1.2*.

After the initial introduction to the aforementioned Matlab script, the code was rewritten: first, in Matlab and then in C++ using some OpenCV libraries. The changes to the Matlab code included some small improvements to speed and the command line interface to make the program more specific to the new device’s needs as well as to suite it to the experimental file structure. Some intermediate versions included custom edge detection algorithms, autocorrelation functions and initial attempts at real-time processing and analysis. These proved, however, not to provide the performance required for real-time processing. The software was then written in C++ utilizing OpenCV libraries. This allowed the C++ SDK from the camera most commonly used during these experiments (Point Gray Grasshopper) to be merged with C++ image analysis code to autocorrelate and plot the stress/strain curves in real time. Together with the SDK of the piezo stage, a complete program was written to include control of the camera, the stage and the autocorrelation process.

Since the calculation of the extensional viscosity required measurement of the evolution of the midpoint radius of the fluid sample during strain, the code also tracked the midpoint diameter of the filament by calculating the FWHM of the intensity profile. The midpoint was tracked by initially selecting its location before the fluid is strained (during the initial stable fluid bridge) and then shifting the scanning region at a rate half that of the velocity of the pulling pipette. Typically, the diameter was sampled over 5 adjacent intensity profiles for constant rate experiments or over 100 profiles for step strain experiments and averaged to reduce the effects of falsely calculated changes in diameter due to pixel shift and image irregularities. More on the measurement of the fiber diameter is discussed in the *Methods* chapter in *Section 4.1.3* while the code can be found in *Section A.3* of the *Appendix*

Another addition which allows an increased amount of automation and user-friendliness is the use of pattern recognition algorithms, also based on image autocorrelation, to detect which micro-pipette is currently installed. Images of all micropipettes are stored in a directory, taken at all magnifications as well as are text files linking these images to their respective spring constants and magnification. When the experiment is started, the pipette is automatically moved into the range of interest of the camera and pattern recognition begins searching for the best match from the directory. Then, once a match is found, the appropriate spring constant and magnification are are stored as variables for the calculation of the correct pixel to micron ratio and spring constants to provide correctly calculated and scaled stress versus strain plots.

2.4 Calibration Technique

Proper, reliable and repeatable calibration of any measurement device meant to probe a material's response and provide trustworthy results worthy of holding up to the vigorous standard on which the field of experimental physics was built is – and must always be – the first step before collecting new data. In this spirit, the following section will outline – in detail – the calibration procedure of the force-sensing glass micropipettes used to examine the stress and strain relation of BLC fibers. The calibration method was a blend of two techniques: the first, a prior art used by Kreis *et al.* during their investigation of the adhesive behavior of algae under different lighting conditions [53] and a second, specific to the extensional rheometry setup used in the experiments described here forth.

2.4.1 Prior Art

Kreis *et al.* wrote Matlab code for calibrating the spring constant of the micropipettes using an *added mass method*. By fixing the micropipette above the objective of an inverted microscope and forming an increasingly large water droplet at the opening of its tip by pressing water through the length of the pipette via a syringe, the mass of the droplet versus the deflection of the pipette could be used to calculate the spring constant. Figure 2.18 graphically outlines the procedure which was carried out in the following order:

1. Place a micropipette in a holder so that it is positioned over the objective of an inverted microscope such that the flexible cantilever end of the pipette can deflect vertically, in the direction of gravity.

2. Attach one end of a hose to the 1mm open end of the pipette while attaching the other end of the hose to a syringe filled with water.

3. Using a mirror positioned at 45 degrees to the optical axis placed over the objective, obtain an image of the pipette from its side so that – via a digital camera attached to the imaging port of the microscope – the pipette's deflection could be observed by image recording software.

4. Ensuring that the tip of the force-sensing segment (cantilever) of the pipette is clearly imaged by the camera, depress the plunger of the syringe until a small water droplet forms at the pipette's opening.

5. Begin recording video.

6. Slowly press the plunger of the syringe in order to form an increasingly larger droplet.

7. Stop recording video.

8. Remove the droplet from the pipette by either retracting the plunger of the syringe or by carefully using an absorbent laboratory cleaning wipe.

9. Repeat the procedure at least five times in order to obtain reliable statistics.

10. Use the custom Matlab software to calculate the spring constant of the pipette.

This procedure proved reliable and robust. Corrections were made for non-spherical droplet geometry. One setback was that the pipettes was tremendously fragile and the procedure included many smalls steps – such as mounting and dismounting the pipette from the holder and the syringe hose – which often lead to breakage. Moreover, the extensional rheology experiments required that tip of the pipette be melted closed after calibration to keep the liquid crystal material from flowing back into the pipette. Although a pipette calibrated using the droplet method can have its tip melted together post-calibration, this extra step was often risky and lead to pipette damage or breakage.

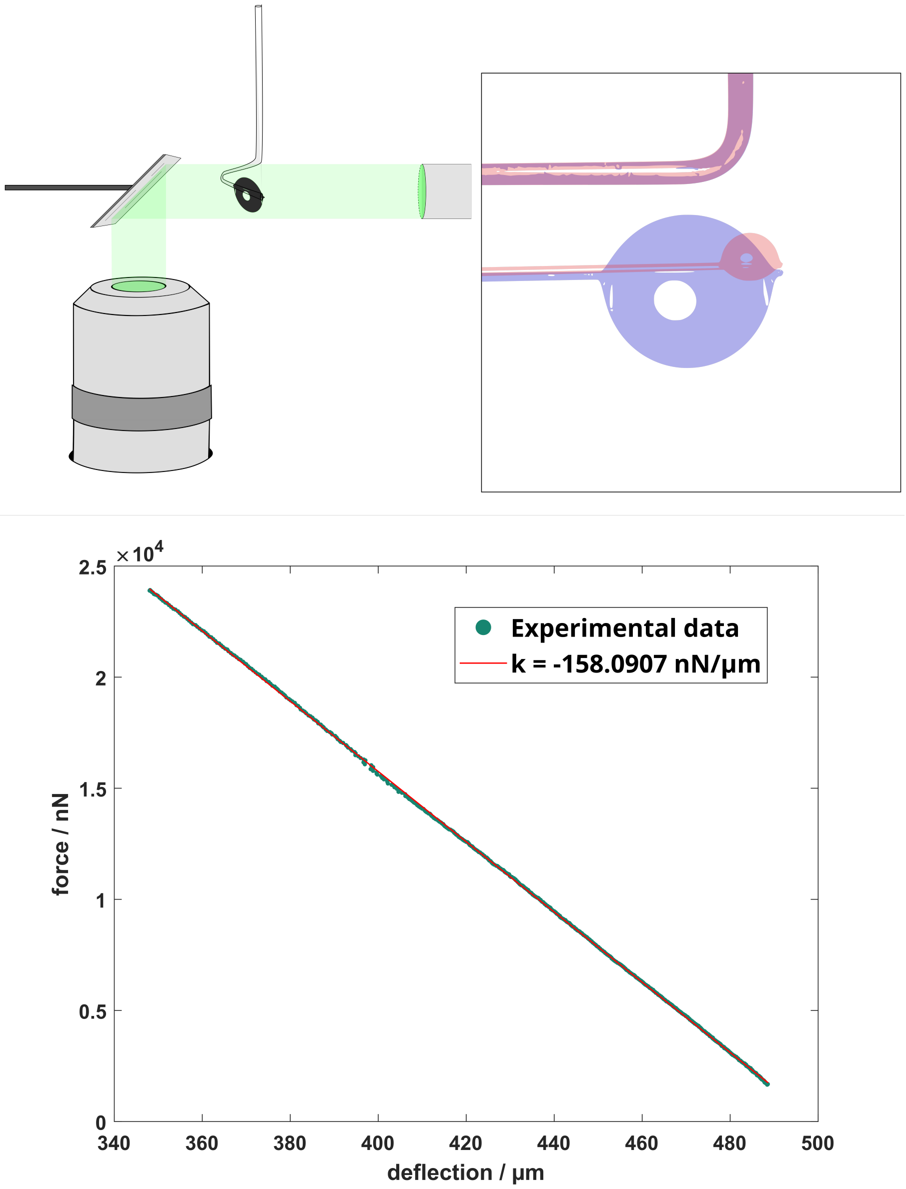


Figure 2.18: A 45° tilted mirror between an inverted objective and horizontal light source allows the vertical displacement of the pipette due to the added mass of a water droplet to be measured. The known change in mass and pipette deflection are then plotted and a linear fit is applied. The slope of this fit is the sprint constant. Red indicates the initial position while blue indicates the final position. The graph is shown as Matlab formatted output.

The water droplet method was also slow to perform and doing multiple trials to get a mean spring constant proved tedious. Due to these shortcomings, a new calibration technique was developed and is discussed in the following section.

2.4.2 Optimized Technique

This new technique allowed freshly-made pipettes to be calibrated against an already reliably-calibrated reference pipette. This reference pipette – denoted pip_{ref} – with a spring constant k_{ref} was initially calibrated by the water droplet method – outlined in *Section 2.4.1* above – with great care to ensure the consistency of all pipettes calibrated against it in the future. The referencing method discussed in this section took advantage of the extensional rheometer itself and its easily-controllable positional stages. The reference pipette was inserted into the pipette holder attached to the multi-axis, joy-stick controlled side of the rheometer while a new and uncalibrated pipette (pip_{new}) was placed in the opposing holder fixed to the linear piezo stage. The idea was to press the tips of the pipettes together and calculate the spring constant of the new pipette (k_{new}) by observing their relative deflections. The process could be automated and repeated many times to collect strong statistics in a matter of seconds.

Because k_{ref} was known, it sufficed to measure the initial velocity of pip_{new} before contact with pip_{ref} (V_{init}) and the velocity of pip_{new} after the point of contact with pip_{ref} (V_{final}). By tracking the displacement of just pip_{new} over a range covering pre-and post contact with pip_{ref}, code could be written to detect the pre-contact slope (V_{init}) and post-contact slope (V_{final}). Since the deflection of pip_{ref} did not need to be measured, the rotational axis of the stage was used to rotate pip_{ref} 45 degrees (out of the focal plane) so that pip_{new} (both pipettes having a diameter of typically 20 μm) could more positively and repeatedly engage pip_{ref}. Calculation of k_{new} was done following that

From a force balance perspective,

$$k_{new}\frac{\Delta x_{new}}{\Delta t} = k_{ref}\frac{\Delta x_{ref}}{\Delta t} \tag{2.16}$$

Since the speed of the pipettes after impact can be considered as their speed relative to the speed before impact, *Equation 2.16* can be written as

$$k_{new}\left(V_{i_{new}} - V_{f_{new}}\right) = k_{ref}\left(V_{i_{ref}} - V_{f_{ref}}\right) \tag{2.17}$$

and since the reference pipette is stationary while the new pipette approaches it at a constant velocity, $V_{i_{ref}} = 0$ and $V_{i_{new}} = V_i$. During contact, both pipettes move at the same speed meaning that

$$V_{f_{new}} = V_{f_{ref}} = V_f \tag{2.18}$$

so that *Equation 2.17* becomes

$$k_{new}(V_i - V_f) = k_{ref}V_f \tag{2.19}$$

or, since only the spring constant of the new pipette – k_{new} – is desired,

$$k_{new} = k_{ref}\frac{V_f}{V_i - V_f} \tag{2.20}$$

Since only the relative change of the velocity of pip_{new} is needed before and after impact with pip_{ref}, the magnification of the objective and pixel-to-micron ratio are not needed, removing another possible source of user error. Imaging of the deflection of the reference pipette is also not needed which means only a thin slice of the viewing area – just enough record the deflection of pip_{new} – is required. This greatly reduces data accumulation and processing time. *Figure 2.19* graphically outlines the following procedure.

1. Align both pipettes in the same focal plane and their contact points nearly in contact with one another.

2. Rotate pip_{ref} 45 degrees about its axis using the rotational axis of the joystick.

3. Using the command line program, give a speed and displacement and number of cycles und run.

4. Run Matlab (or C++) code to repeatedly calculate the spring constant of k_{new} and extract a mean value.

The same reference pipette was used to calibrate all other pipettes. The reference pipette itself was calibrated using the water droplet method using a mean spring constant from 10 trials. Calibration using the reference method was done on each new pipette 10 times and a mean spring constant calculated. The reference method could be repeated as many times as is desired in quick succession. The fitting of the slopes V_i and V_f were done with Matlab code which detected each slope and applied a linear fit. The entire process was automated and took less than one minute to finish. Once the calibration was finished, only the reference pipette needed to be removed from the rheometer; the newly calibrated pipette could stay inside and new rheological measurements could immediately follow. This helped ensure that the new pipette remained safe due to minimal handling.

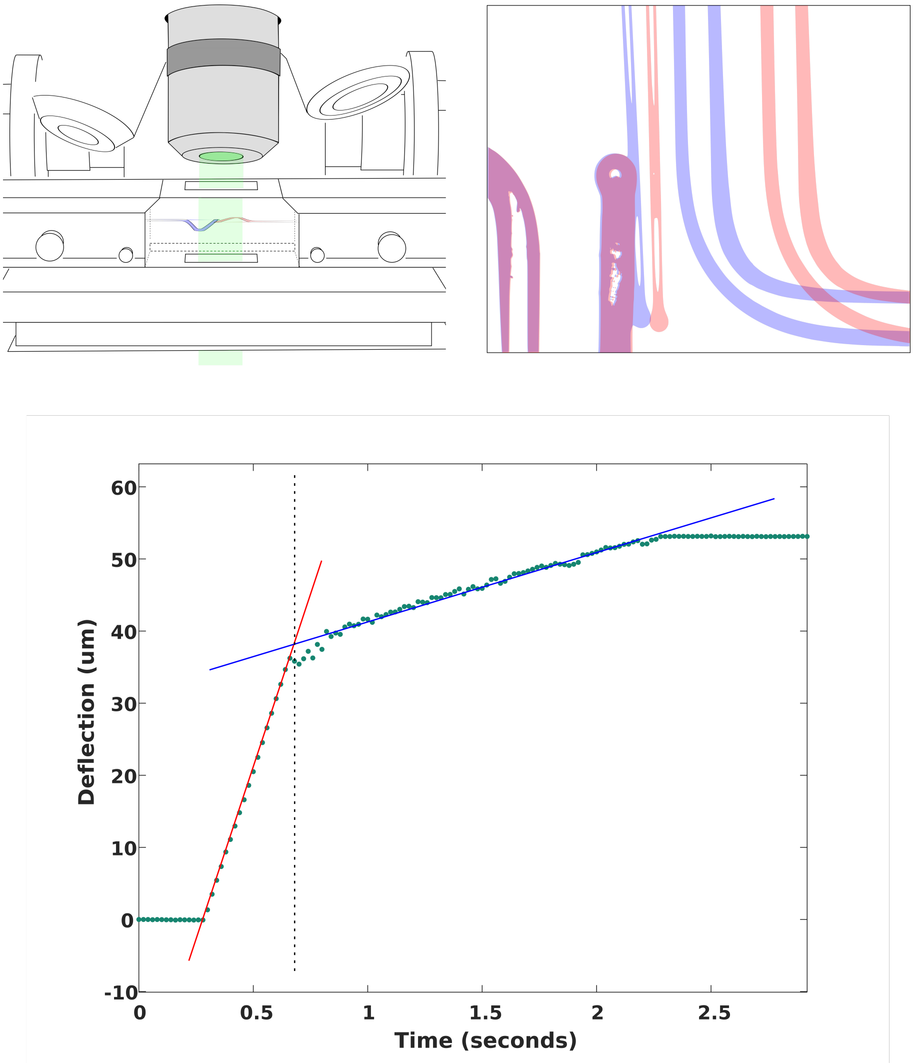


Figure 2.19: The referencing method for pipette calibration takes place inside the rheometer. In the image of the pipettes, red indicates their initial position (before contact) and blue their final position (after contact). In the deflection versus time plot, the red slope indicates the velocity before impact (V_i) and the blue slope the velocity after impact (V_f).

2.5 Finished Product

The result of this invention is a compact extensional rheometer with nano-Newton force resolution, nanometer spatial resolution, time resolution limited only the user's camera system of choice which delivers real-time analysis using the smallest possible sample sizes over a wide range of finely controlled temperatures which is more affordable than the extensional rheometers currently on the market. This opens the doors for users to obtain a low-cost, mobile alternative without sacrificing measurement quality.

Comparative Specifications		
Specifications	NExt Rheo	FiSER
Dimensions (cm)	20 x 6.5 x 2.5	200 x 40[1] x 60[1]
Min. Sample Vol.	< 1 nL	20ml[2]
Min. $\dot{\varepsilon}$ (1/s)	0.001	0.5
Max. $\dot{\varepsilon}$ (1/s)	100	10
Force Resolution	<nN	mN[3]
Diameter Resolution	$< 1\ /mum$	80 μm[4]
Temperature Range	RT - 200°C	0 - 80°C
Portable	Yes	No
User Servicable	Yes	No
[1] Estimates from images where 2m height is known [2] Estimate from videos of FiSER experiments [3] Estimates from scatter of force vs strain data		

In *Figure 2.23*, the red lines which indicate the improved operational space my invention are very conservative. This range is limited by the sampling rate of the camera, maximum acceleration and velocity of the end plate (glass rod in my case) and the minimum initial length of the fluid filament which is limited by the minimum critical aspect ratio which is limited by the minimum sample size. In practice, with a high-speed camera recording at 8000 FPS and a sampling density of 100 data points over a period of 1 second (before the stage reaches its maximum velocity of 10mm/s) and an initial fluid length of 10 microns means an upward bound on the strain rate of over 100 s^{-1} and a lower bound nearly 2 orders of magnitude smaller than what FiSER can offer. This means that the rheological properties of even smaller samples of newly developed materials can be tested over a wide operating domain while exposed to a large range of temperatures on an automated, real-time, compact and inexpensive platform.

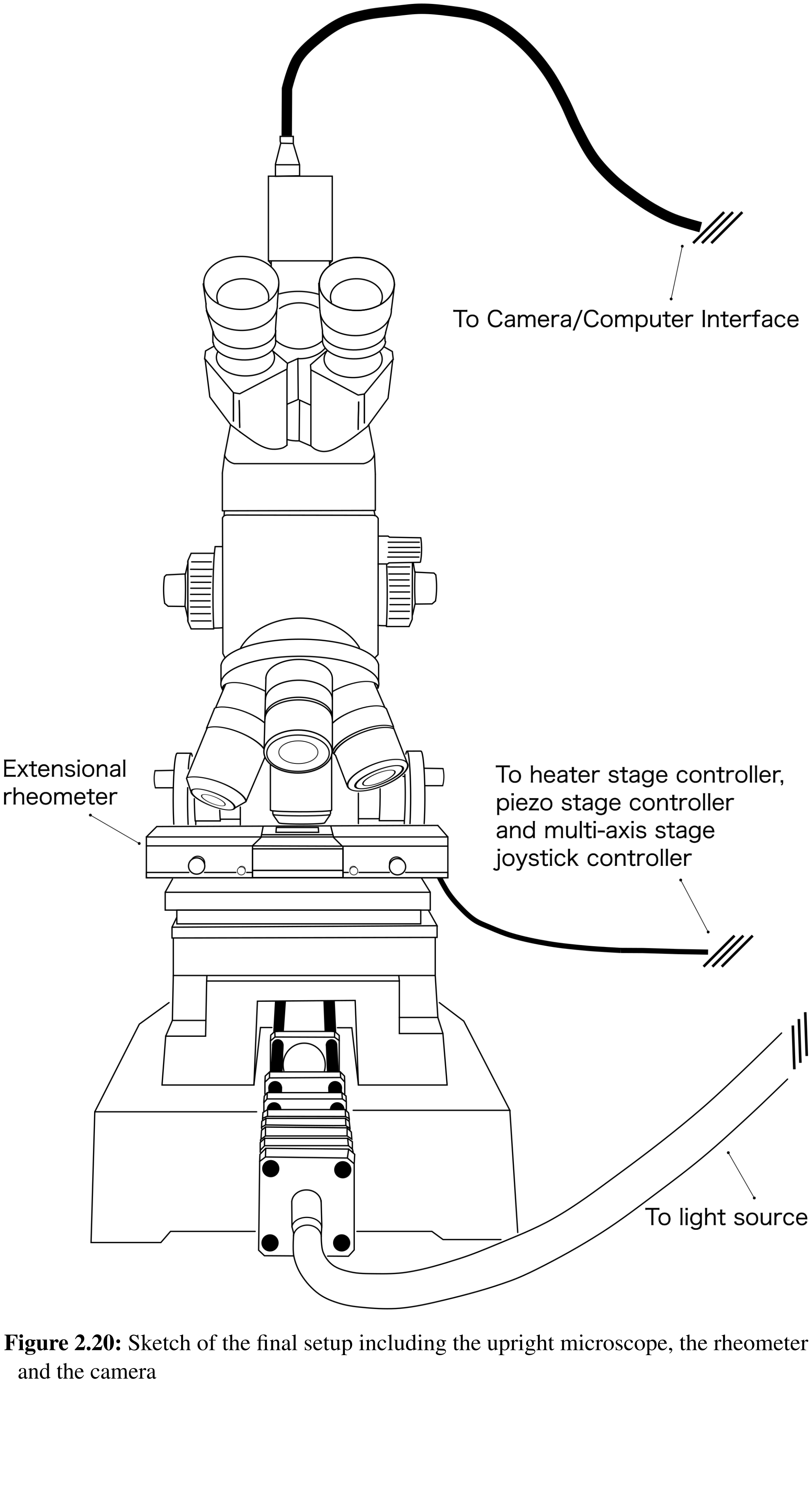

To Camera/Computer Interface
Extensional rheometer
To heater stage controller, piezo stage controller and multi-axis stage joystick controller
To light source

Figure 2.20: Sketch of the final setup including the upright microscope, the rheometer and the camera

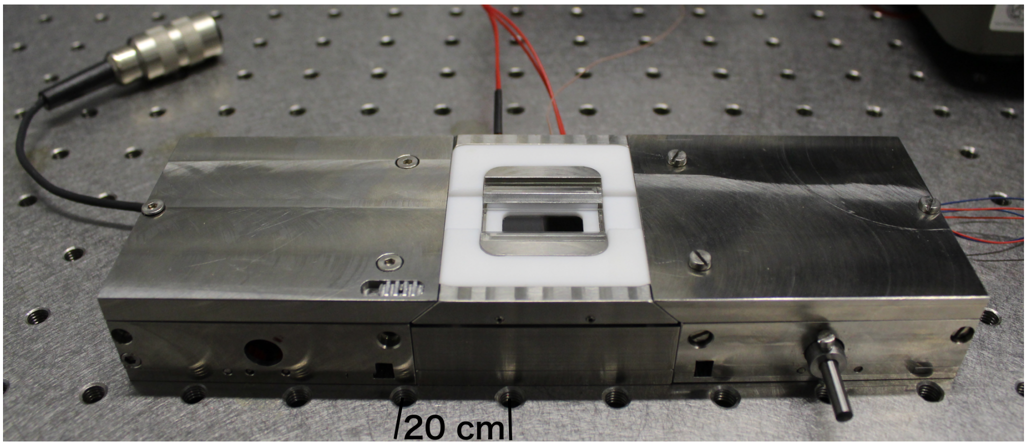


Figure 2.21: Final NExt Rheo device

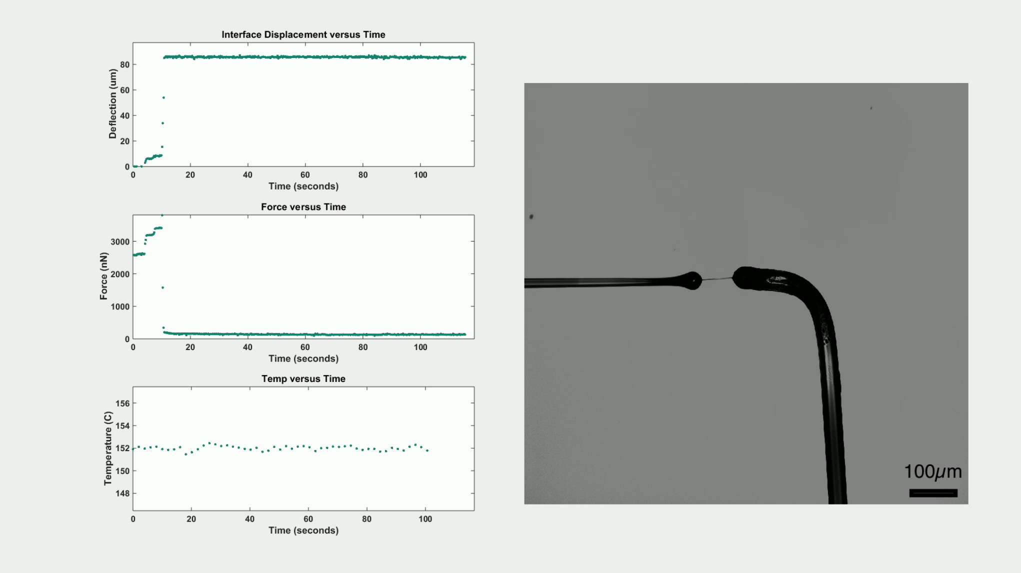


Figure 2.22: An early example of the force, displacement and temperature readings observed during testing of a liquid crystal sample.

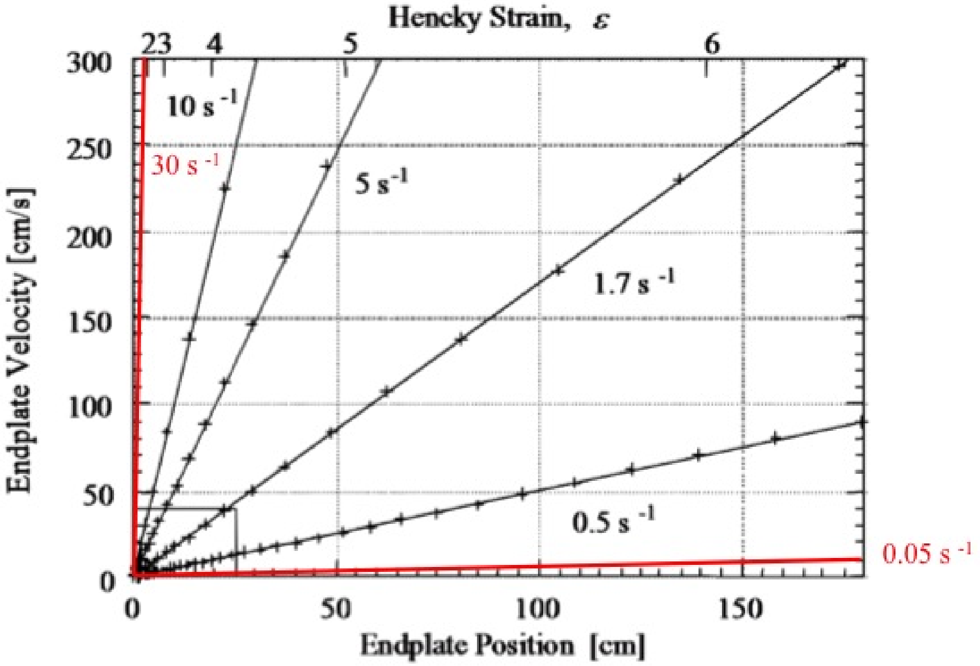


Figure 2.23: The red lines show that this new device exceeds the operational space of the FiSER `http://www.campoly.com/files/2113/5216/6044/006.pdf`

CHAPTER III

MATERIALS

Bent-core, or banana molecules, add a step of complexity on top of the simple and common calimitic geometry. While calimitic, or rod-like molecules such as the widely-known 5CB made popular by its widespread use in the era of liquid crystal displays, simplify optical, mechanical, elastic and viscous theory and analysis due to their uniaxial structure, bent-core molecules are biaxial, resulting in broken rotational symmetry which affects steric interactions as well as bringing a net dipole moment with some finite angle away from the molecular long axis into the picture.

Because certain phases formed by some bent-core mesogens (B2 and B7) spontaneously form highly stable cylindrical fibers, they are a perfect model system for studying the extensional rheology of fluid-like fibers. Most other attempts to study extensional dynamics, especially by uniaxial oscillatory oscillation rely on poly melts which can typically only be studied in the regime only just surpassing a filament length longer than the Rayleigh-Plateau limited fluid bridge allows, or πD, where D is the minimum bridge diameter [51]. While bulk samples typically undergo studies of shear rheology, such as the shear response investigated by Bailey *et al.* in 2009, isolating fluid-like samples to a stable cylindrical geometry is uncommon [64]. The mechanical properties and responses of stable fibers formed by bent-core mesogens have been studied in the static case of bead-hanging to measure filament tension, as well as dynamically by acoustic driving, electric plucking, and filament rupture [4, 5, 26, 27]. These studies sought to derive viscous coefficients, elastic contributions from bulk structure and surface contributions.

3.1 Chemical Structure

The mesogen used in this study was a single homologue of *2-nitro-1,3-phenylene bis [4-(4-n alkyloxy phenylimino methyl)- benzoates]*, and is shown in *Figure 3.1*. The end groups, denoted as *R*, are the methyl chain $C_nH_{2n+1}O$ where $n = 7$ for the homologue of interest. This material, not only in its $n = 7$ homologue but its $n = 16$ homologue as well, is known to form a B7 phase with chiral, helical superstructures [65]. The $n = 7$ homologue has a crystalline to B7 (smectic) phase transition at roughly 100 °C and a B7 to isotropic (liquid) transition at roughly 170 °C.

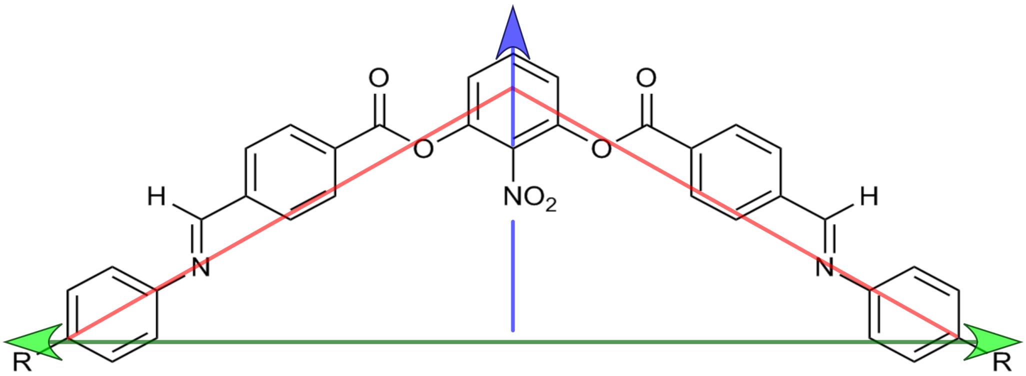


Figure 3.1: The structure of the *2-nitro-1,3-phenylene bis [4-(4-n alkyloxy phenylimino methyl)- benzoates]* bent-core mesogen used in this study. Single fibers are formed in the B7 phase. The vertical blue arrow indicates the direction of the dipole moment while the horizontal green arrow represents the molecular long axis.

3.2 Phase Structure

This phase is considered to be a SmCsPf phase. Sm stands for smectic, in that it exhibits a layered structure. Cs stands for its *clinicity*, or layer-to-layer symmetry. A smectic phase which is synclinic means the molecular tilt angle is in the same direction from one layer to the next whereas *a*ntisynclinic means the tilt angle switches from layer to layer. Similarly, Pf stands for the type of ferroelectric structure. a subscript of *f* stands for ferroelectric meaning the dipole moment face in the same direction from one layer to the next while a subscript of *a* stands for antiferroelectric, alternating polarity form layer to layer. This phase is also said to be a polarization modulated, undulated layer structure, meaning that the smectic layers have axial undulations due to density fluctuations bought on by polarization splay defects [40, 66].

The layered structure of this phase has clearly defined lattice parameters a, b and ϕ where a is the smectic layer spacing (thickness of each layer), b is the undulation periodicity (spacing between the helical defect line), ϕ is the tilt angle of the long molecular axis from the smectic layer normal and

$$a = Lcos(\phi) \tag{3.1}$$

where L is the length of the molecule along its long axis as shown in *Figure 3.2*. The values of the lattice parameters are provided in *Table 3.1*. The proposed layer structure, once in the cylindrical form of a stable fiber, is shown in *Figure 3.3*. The increased distortion energy of the bent layers near the center axis of the fiber require the phase to become unstable and melt into an unstructured liquid phase and a different partially structured phase in order to minimize the core energy. This core region is shown in *Figure 3.3* and *Figure 3.4*.

Phase Specifications					
n	Phase sequence (°C)	L ($\AA$)	a ($\AA$)	b ($\AA$)	ϕ (°)
7	**I** 170 **B$_7$** 100 **Cr**	44.9	34.3	90.0	85.9

Table 3.1: The phase sequence and lattice parameters of the n=7 homologue

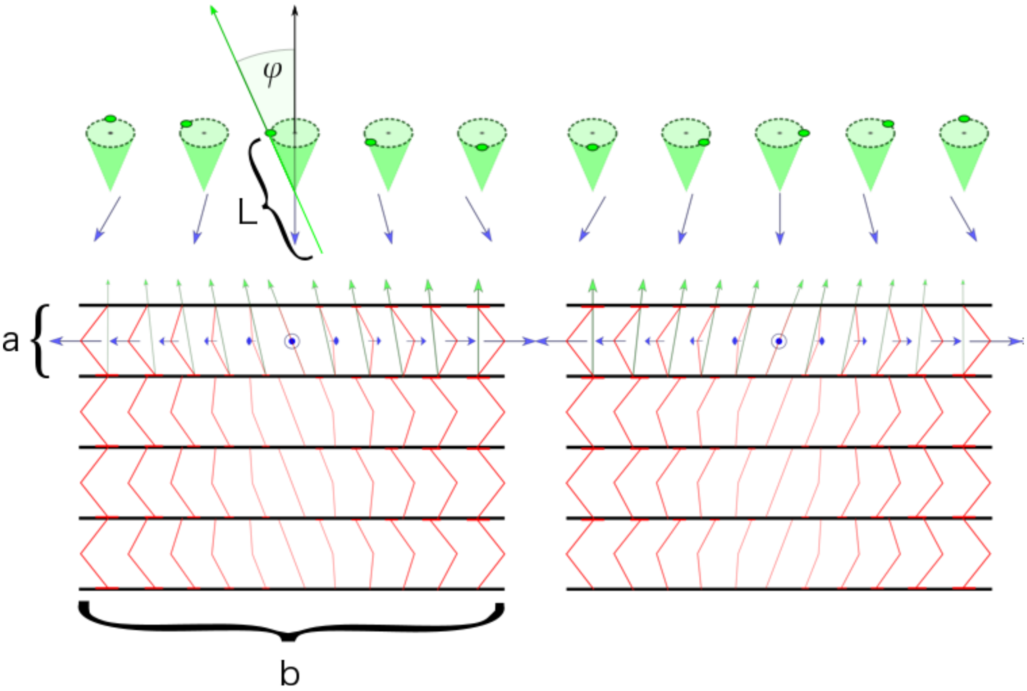


Figure 3.2: Smectic layers formed by the SmCsPf phase of a bent-core molecule. The long axis with length L is at a fixed angle ϕ to the layer normal and is free to rotate around the cone angle. With the polarization vector – in blue – locked in the layer plane, a spontaneous polarization splay induces period defect lines spaced a distance b apart. The resulting layer thickness is a.

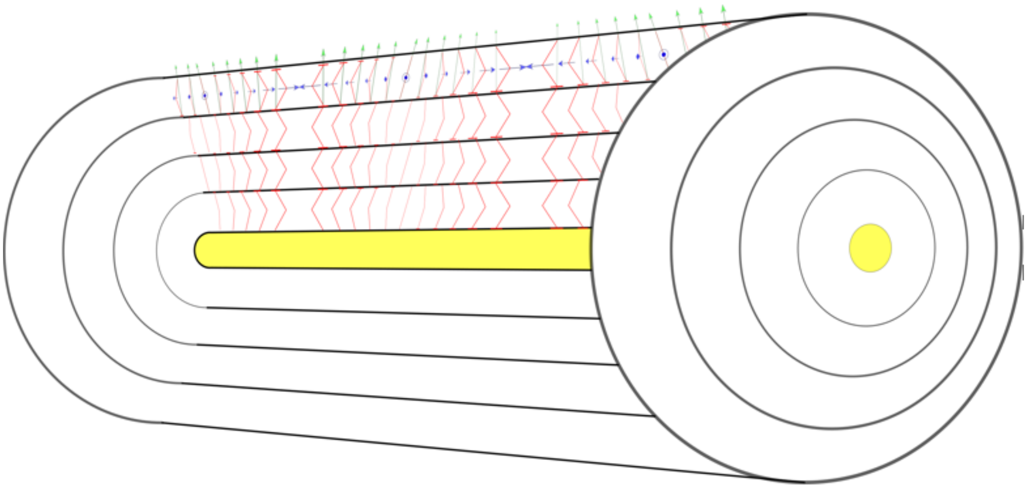


Figure 3.3: A sketch of the proposed smectic structure of the fibers studied in this thesis. Concentric layers around a defect core stabilize against radial fluctuations while spontaneous polarization splay induced axial defect lines.

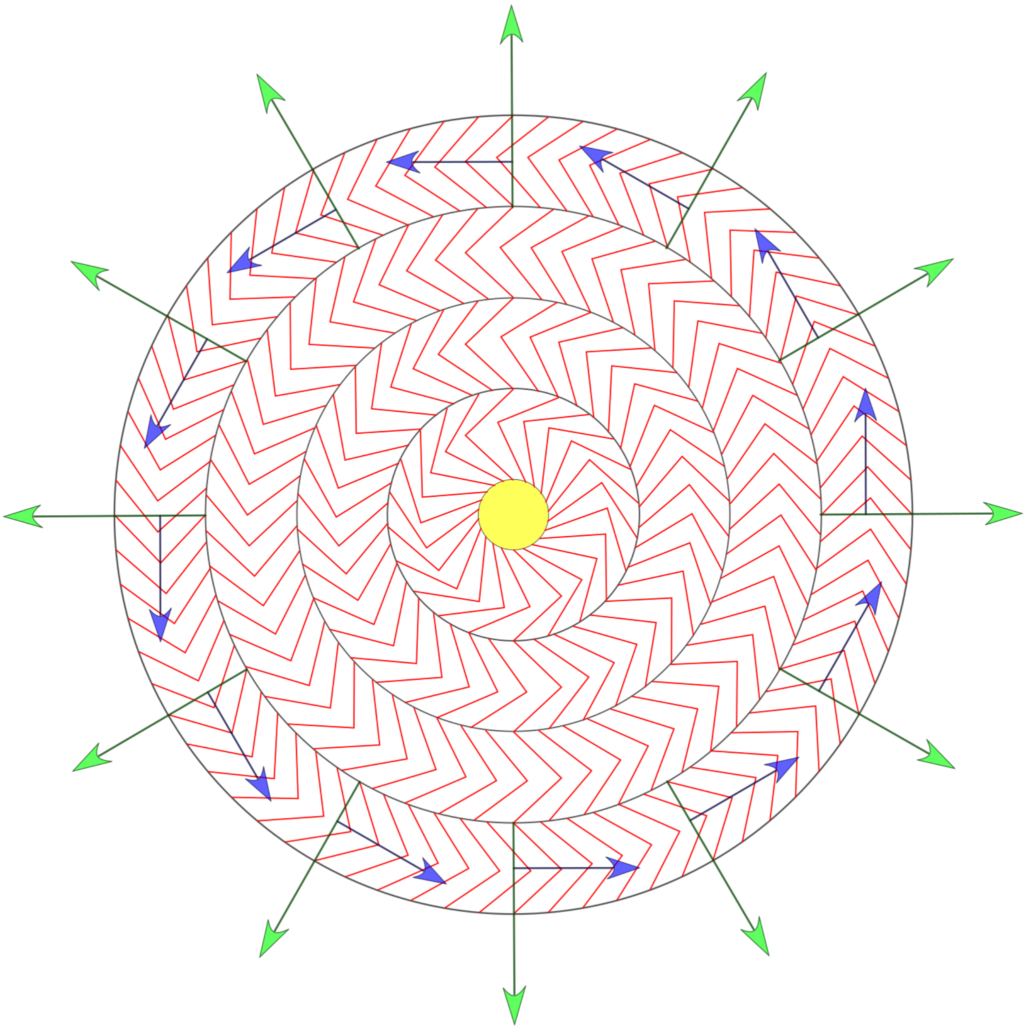


Figure 3.4: In this sketch, the radially-outward arrows indicate that the long axis of the molecules points radially away from the core. The dipole moment wraps around within the layer plane for each layer. At the center, the defect core due to critical curvature is shown in yellow.

CHAPTER IV

METHODS

The purpose of the *Methods* section is to guide the reader through the details of the experimental protocol. From initial setup, to data acquisition (DAQ), to data processing and finally to data structuring with emphasis on contributions to noise, error and the know-how required to repeat these experiments for new materials using a similar or identical experimental setup. The protocol is divided into two parts. *Section 4.1* discusses the general protocol which applies to all experiments such as initialization of the setup, initial fiber formation, data recording and force measurement. The second part is presented as separate sections detailing the protocol for each type of rheological experiment: *Constant-rate Extension, Step Strain and Stress, and Oscillatory Excitation* (*4.2,4.3, 4.4*, respectively). These sections are further broken down into two subsections each: *Experimental Procedure* and *Data Processing and Error Analysis*. While the setup proved extremely useful and reliable, the reader is encouraged to look for any means of improvement – of which there are surely many – for future implementation of these methods.

4.1 General Protocol

Although each type of measurement needed a protocol uniquely tailored to certain spacial, temporal and stability requirements to properly resolve, acquire, and analyze the resulting experimental data, there was a common method in the preparation, initialization and force response analysis for each experimental approach. These procedures are discussed in detail in this section.

4.1.1 Initialization

The initial pulling of bent-core fibers was done following similar approaches used in the past studies of fibers formed by similar mesogens [4, 5, 20–25, 27, 67]. The *mesogenic* powder was carefully placed at the tip of one of the two glass interfaces. Because the powder is composed of small crystalline flakes, a small needle could be used to transfer a single flake and transfer it to the sample region inside of the rheometer. Upon optical inspection and confirmation that the material had adhered itself near the tip of the glass micropipette, the lid of the heat stage was slid into the closed position and the stage heated to just above the mesogen's isotropic transition (*melting point*) of roughly 170 $^{\circ}C$ [28]. Once the mesogen had fully melted, the second glass interface was be moved into place and brought into contact with the sample. A fluid bridge then formed and spanned the distance between the two interfaces as shown in *Figure 4.1 (a).* The bridge could then be pulled to its isotropic limit – dictated by the Rayleigh Plateau instability – of a length less than its minimal circumference, $2\pi D$ [68].

At this point, the temperature was reduced to 5 $^{\circ}C$ below the melting point. For this material, it was known that the first phase below the isotropic phase is the B7 phase; the first known and most stable bent-core fiber phase to date [5,20,21]. The transition was verified by either observing a clear-to-cloudy transition under unpolarized lighting conditions or by observing a dark-to-transmissive transition under crossed polarizers. Another method used to observe the phase transition was to observe an increase in internal stresses due to a sudden increase in density upon entering the B7 phase by measuring the deflecting of the force pipette upon cooling as shown in *Figure 4.1 (b).*

The sample material was then slowly pulled apart by using the 3-axis joystick controller (*Figure 2.11*) to move the force-sensing pipette away from the pulling pipette until the fluid bridge reached a threshold stress followed by the spontaneous formation of a stable, free-standing fiber (*Figure 4.2*). This threshold is likely a measure of the energy barrier that is needed to overcome in order to orient the somewhat disordered B7 bulk made up of mono-domains into the geometry required for a stable fiber – a lower energy state. Once a stable fiber was pulled, its length could be increased or decreased until the desired length was achieved. It is important to note that the rate at which a fiber was pulled after its initial formation – in order to bring it to a new initial length – had to be slow enough to prevent rupture and that 30 seconds to 1 minute of relaxation were necessary after pulling to allow the fiber's stresses to relax.

In order to initialize the rheometer, all hardware systems were first turned on. These components were, in no particular order,

- Camera - Point Grey Grasshopper 5 MP USB3
- Piezo controller - PI Q-series 522.030
- Heat stage - Omega controller via USB
- Light source - 150W halogen with custom power supply for stability and green-light filter for a 530 nm monochromatic source
- Joystick Controller

The experimental initialization was carried out in the following sequence:

1. Initialize the PI stage by opening the appropriate windows executable file in order to run the referencing program.
2. Using the 3-axis joystick controller, move the tip of the force-sensing pipette to the center of

the viewing area (ROI) and bring the microscope into focus.

3. Using the x-axis manual micrometer attached to the piezo stage, move the tip of the puller pipette until it is nearly in contact with the sensing pipette.

4. Using the z-axis manual micrometer attached to the piezo stage, move the tip of the puller pipette until it lies in the same focal plane as the sensing pipette

5. Open the lid of the heat stage by sliding the locking mechanism to the unlocked position and then sliding the lid forward to expose the two glass pipettes.

6. Using a small metal wire or needle, gently place a small flake of the BLC mesogen onto either of the two glass tips.

7. Close the lid and ensure the locking mechanism retracts into the locked position.

8. Set the temperature of the heat stage to just above the isotropic transition temperature of 170 °C and wait for the temperature to stabilize.

9. Pull a new fiber using the methods outlined at the beginning of *General Protocol*, *Section 4.1*.

10. Run the desired experimental program and record new data

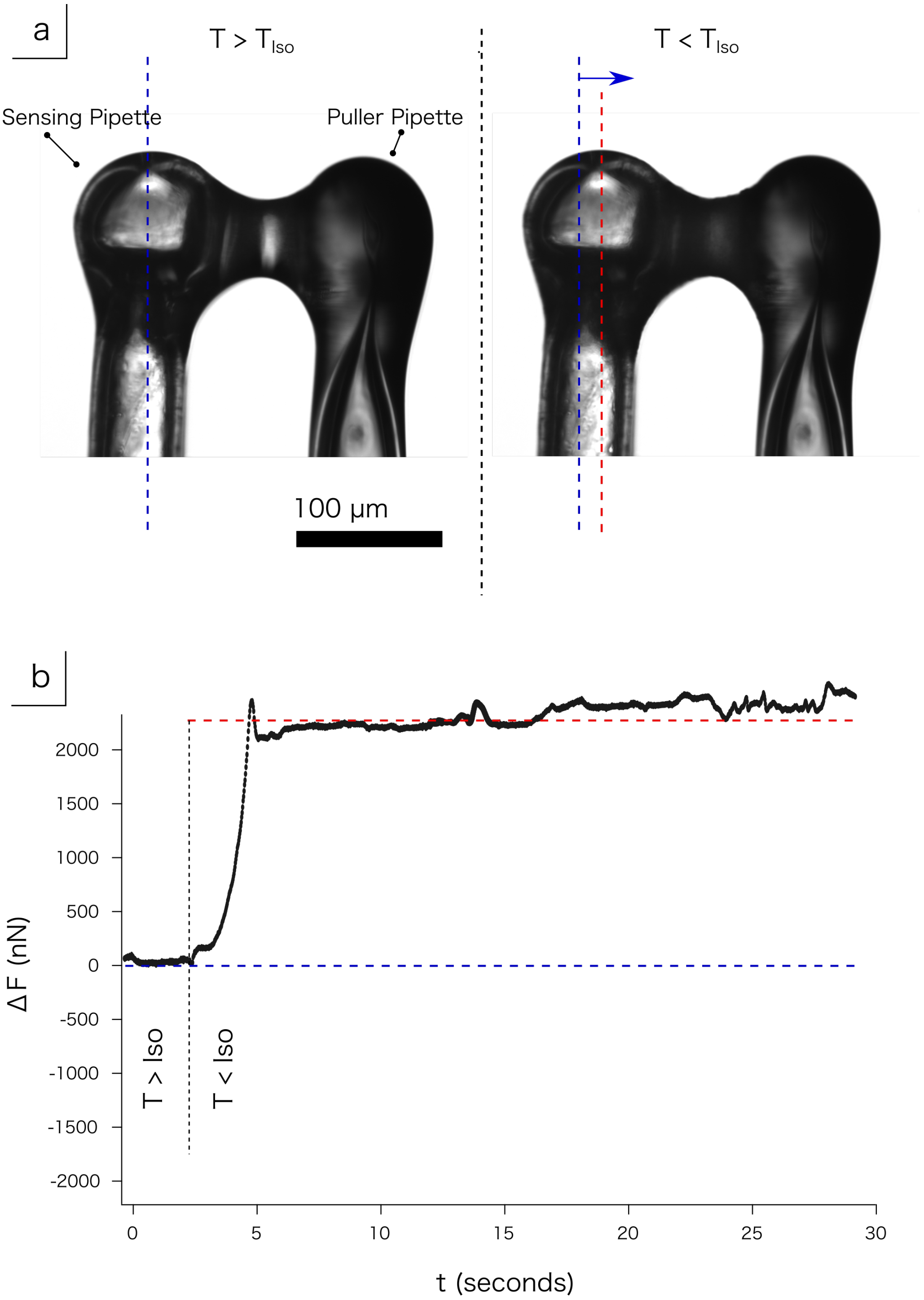


Figure 4.1: **(a)** Upon cooling, the bulk material leaves the isotropic – optically clear – state and enters the B7 – optically opaque – phase accompanied by a sudden density increase seen by an increase in stress and bulk contraction seen in **(b)**, a plot showing the increased force exhibited on the glass pipette due to the temperature-induced phase transition.

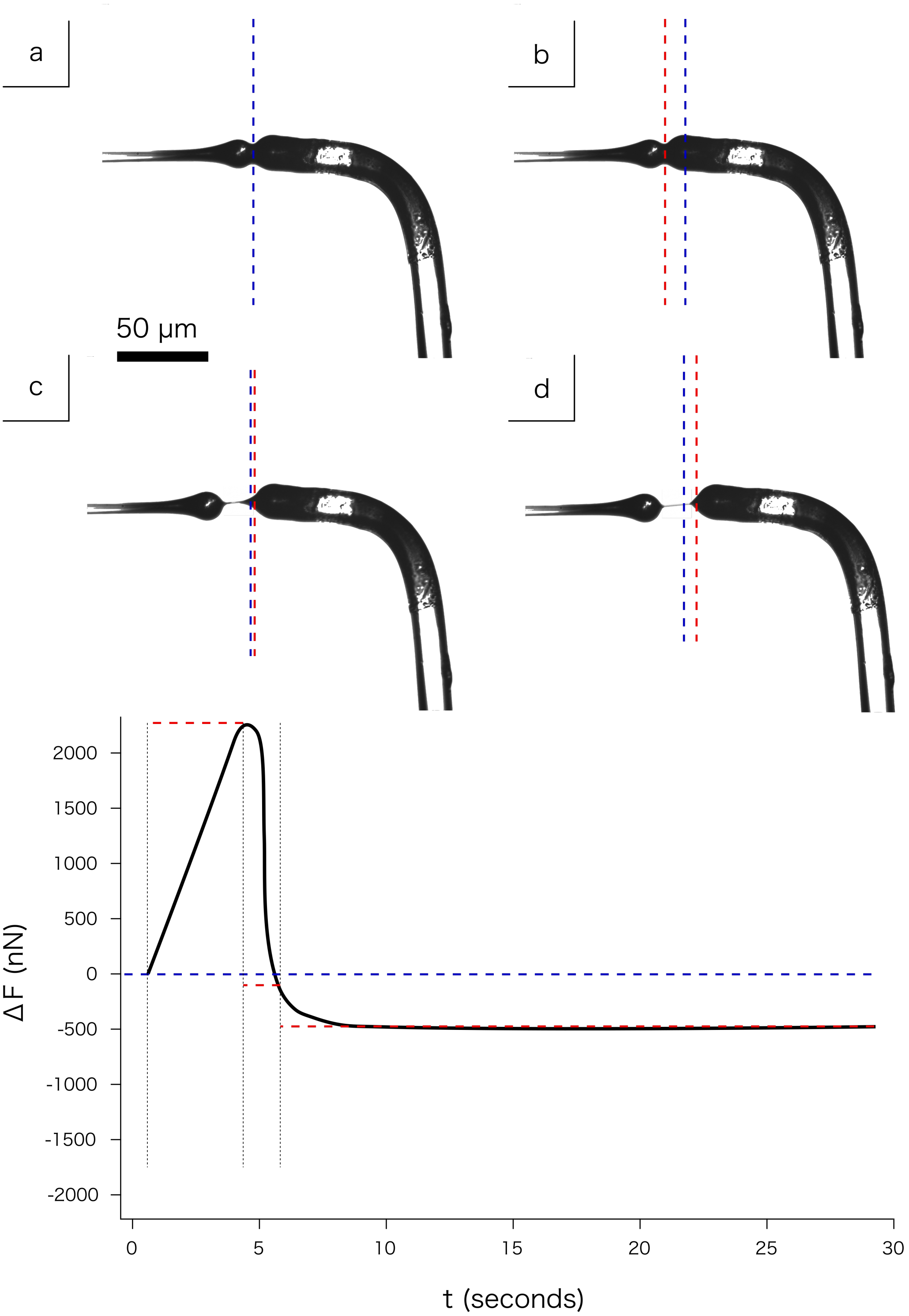


Figure 4.2: The material is strained from its initial state (**a**) until the force-sensing pipette is adequately deflected (**b**). The pulling pipette is held constant until the fiber is spontaneously formed (**c**) and slowly relaxes to its equilibrium state (**d**).

4.1.2 Pipette Tracking and Data Acquisition

All experiments required the tracking of the force-sensing pipette, the rigid puller pipette and the diameter of the fiber in order to measure the force and change in length needed for calculating parameters such as the stress (σ) and strain (ε). Snippets of the Matlab code and its outputs used for pipette tracking, `Deflection_v3_3.m`, as well as the user dialogue are shown in the *Code Snippets* section (*Section A.2*) of the *Appendix*. The basic procedure was as follows.

1. Choose experiment date
2. Choose experiment number
3. Choose which micropipette was used (Many options, each with different spring constants)
4. Choose first and last image in the sequence
5. Using a GUI, choose the endpoints of a line which span the deflection of the force-sensing pipette.
6. Begin image analysis

The image analysis was done by cross-correlating the intensity profile of a line drawn across the micropipette in the direction parallel to its deflection. All frames were compared to the initial image. The Matlab function `xcorr.m` was responsible for this calculation. The peak of the correlation function shifted horizontally according to how much each subsequent image needed to be shifted against the initial image to maximize the correlation of the intensity signals. Each correlation function, one for each frame, was fitted to a spline function. The peak of the correlation function was then taken from the spline fit instead of the peak pixel location so that the distance between peaks is calculated at sub-pixel resolution (*Section A.2*) [53]. Therefore, this technique allows – at 20X magnification (5.88 $pixels/\mu m$) – for pipette movements of around 10 nm or better to be measured depending on noise and vibrations.

The greatest source of noise was thermal fluctuations which affected the BLC sample and thus the force response (deflection) of the pipette. After the deflection of the force-sensing pipette was measured, the program was run a second time to track the movement of the rigid puller pipette so that the change in fiber length over time could be measured. In addition, an algorithm scanned the entire length of the fiber to measure its radial profile at the beginning of each experiment and – in the case of constant rate extension – during the duration of the experiment as well. This measurement technique is described in detail below in *Section 4.1.3*.

4.1.3 Fiber Diameter Measurement

During small amplitude deformations such as those experienced during SAOS or step-strain experiments, only the initial diameter was measured since the fiber typically compressed and expanded much less than was resolvable as discussed in *Section 1.3*. In the case of small strains where only the initial diameter was desired, the fiber was measured at 100 points along its length. For large amplitude deformations where thinning could occur such as those experienced during constant rate extension or LAOS experiments, the diameter was measured in each frame at five points near its midpoint. In the case of small strains where only the initial diameter was desired, the fiber was measured at 100 points along its length.

The program – `Fiber_Geometry_Measurement.m` (*Appendix A.3*) – asks the user to mark the beginning and end of the fiber in the first frame. Then, 100 lines were drawn perpendicular to the fiber to collect intensity profiles. The code then fitted each intensity profile to a Gaussian profile (corrections are made for double peaks) and the fiber diameter calculated as the full-width at half-maximum (FWHM) of the fit.

Finally, a histogram of all collected diameters was plotted and another Gaussian fitted to the distribution. The midpoint of the Gaussian fit determined the best guess of the fiber diameter – as shown in *Figure 4.3* – and was then defined as the R_0 to later calculate the stress, σ, as

$$\sigma = \frac{F}{\pi R_0^2} \tag{4.1}$$

where F is the force measured from the deflection of the force pipette.

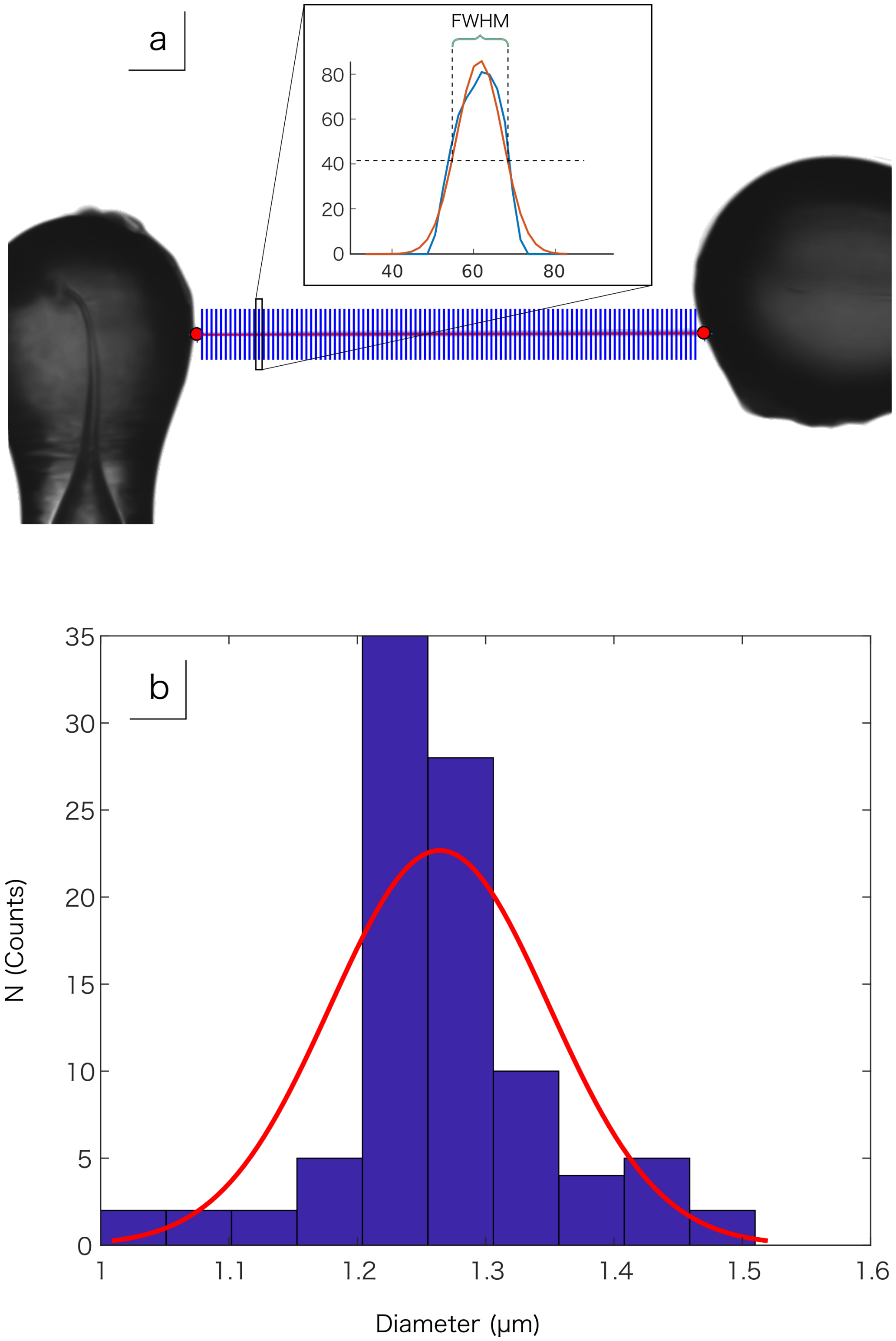


Figure 4.3: Measurement of the fiber diameter. **(a)** Samples (intensity profiles) were taken along the fiber and diameters extracted from the FWHM of each Gaussian fit. The original signal (blue) and the Gaussian fit (red) are shown in the inset for one of one hundred profiles. **(b)** The measured diameters were collected to plot a histogram from which the center of the distribution – once fitted to a Gaussian – was used to calculate the fiber radius, R_0 or $r(t)$.

4.2 Constant Rate Extension

Initially inspired by the work of Garreth McKinley at MIT and others in the field of extensional rheometry, an attempt was made to compare the behavior of bent-core fibers to that of polymer melts and solutions when pulled at a constant strain rate over large strains. The typical material parameter extracted from such constant rate extension is known as the *extensional viscosity* and is discussed in more detail in *Section 1.4* of the *Introduction* [44–46, 48, 55–58, 60–62, 69–72]. The expression for the extensional viscosity is given by *Equation 1.25* and is shown again here for convenience.

$$\bar{\eta}^{+} = \frac{\sigma}{\dot{\varepsilon}_0} - \frac{\gamma}{2\pi R_{mid}\dot{\varepsilon}_0} \tag{4.2}$$

4.2.1 Experimental Procedure

The procedure began by first pulling an initial fiber according to the aforementioned technique. Once a stable fiber had formed, the program `exp_vel_pull.exe` was opened and the parameters L_0, ε (entering ”0” means pulling until the stage’s limit) and $\dot{\varepsilon}$ were entered. The stage then began to strain the fiber according to the exponential displacement curve outlined in *Section 1.4* titled *Modern Extensional Rheology* in the *Introduction* by *Equations 1.19* through *1.23*. The final equation is shown again here for convenience,

$$L(t) = L_0 e^{\dot{\varepsilon}t}. \tag{4.3}$$

Images were recorded at a frame rate adequate for resolving the movement beyond each individual step of the motor, or $FPS > 2 * (1/dT)$ where $dT = 0.05$ seconds is the time interval between steps provided by the piezo stage during pulling and the factor 2 from the minimum sampling requirement introduced by Nyquist [73]. Therefore, the frame rate was typically set between 40 and 50 FPS. The length of travel of the stage was enough to rupture the fiber so that the force-sensing pipette always returned to its initial position, allowing the initial position of the pipette, i.e. the initial stress of the fiber, to be known. From these measurements, stress versus strain curves for different temperatures and strain rates could be plotted.

4.2.2 Data Processing and Error Analysis

The position of the pulling pipette and the force-sensing pipette, as well as the diameter of the fiber, were measured in post-processing. The pipette deflections were measured using the standard auto-correlation method and the diameter of the fiber by the mean FWHM technique for all frames. Because these experiments typically lasted for less than 30 seconds, the temperature fluctuations – with periods of about 30 seconds did not affect the experiment.

Because it was known from the calibration of the exponential displacement profiles (*Figure 2.16*) that although the desired strain rate could deviate from the output, the profile itself always strongly matched an exponential curve with only a slight difference in the strain rate. During these experiments, the puller pipette always exited the viewing area of the microscope and camera meaning its full displacement could not be measured. It was enough to track its movement until it left the frame and then fit its profile to an exponential with the known initial length, L_0, as the amplitude. The fitted curve was then used to extrapolate the full strain of the fiber over the full time of the experiment.

The scanning region of the fiber diameter was shifted in each image so that the region itself moved in the direction of the puller pipette at half the instantaneous pulling velocity, ensuring that the scanning region always scanned the same portion of the fiber until the region left the viewing frame and the fiber diameter could not longer be measured as shown in *Figure 4.4*.

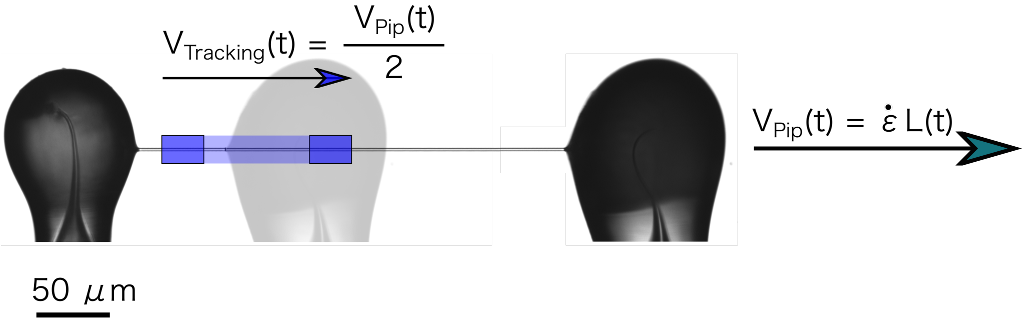

$V_{\text{Tracking}}(t) = \frac{V_{\text{Pip}}(t)}{2}$
$V_{\text{Pip}}(t) = \dot{\varepsilon}\, L(t)$
50 μm

Figure 4.4: A fiber being pulled at an exponentially increasing rate such that the strain rate, $\dot{\varepsilon}$, remains constant. The blue box indicates the region in which the diameter of the fiber is being measured. The box is moved in the direction of elongation at half the velocity of the puller pipette to ensure the same region is scanned for each frame.

4.3 Step Strain and Stress

The relaxation behavior of a material in response to a small, instantaneous and steady jump in strain or stress can provide great insight into the molecular timescales responsible for its viscoelastic nature. Basic polymers respond to a step strain by relaxing their stresses according to a superposition of an infinite set of frequencies given by a power law behavior. In contrast, more complex systems composed of discrete and constant length scales tend to relax according to a finite set of timescales – each one indicative of a certain mode of relaxation present at the molecular level – typically represented by a series of exponential decay functions. Imposing an infinitesimal and instantaneous strain can tell us a lot about how a material behaves over time [74–76]. Further details of step strain theory and the different viscoelastic models and relaxation functions are left to the *Analysis* section and can be further reviewed in books such as ”Viscoelastic Theory: Second Edition” by R.M. Christensen [75].

A material's stress response (*Equation 4.5*) to a step strain (*Equation 4.4*) is presented as a *relaxation function* similar to *Equation 4.6*. Likewise, imposing a step-stress – a maintained stress during which the material experiences a strain response – gives what are called *creep functions*. Creep can be thought of as the response of a rope when put under extensional load by hanging a weight from it (a constant stress) and watching the length of the rope increase in length at a decaying rate [75].

$$\varepsilon_{ij}(t) = \varepsilon_0 h(t) \tag{4.4}$$

$$\sigma_{ij}(t) = G(t)\varepsilon_0 \tag{4.5}$$

$$G(t) = G_0 e^{-t/t1} h(t) \tag{4.6}$$

where $h(t)$ is the unit step function (zero at $t < 0$ and 1 at $t > 0$), ε_0 the total magnitude of the step in strain, $\varepsilon_{ij}(t)$ the strain tensor in time, $\sigma_{ij}(t)$ the stress tensor in time, $G(t)$ the stress relaxation function, G_0 the amplitude of the relaxation function, t the time after the step and t_1 the characteristic relaxation time of the material [77]. The way in which a material relaxes after such a step strain is also used to understand under which classification it falls: elastic (Hookean), viscoelastic solid, viscoelastic liquid, or purely viscous (Newtonian) as shown in *Figure 4.5*.

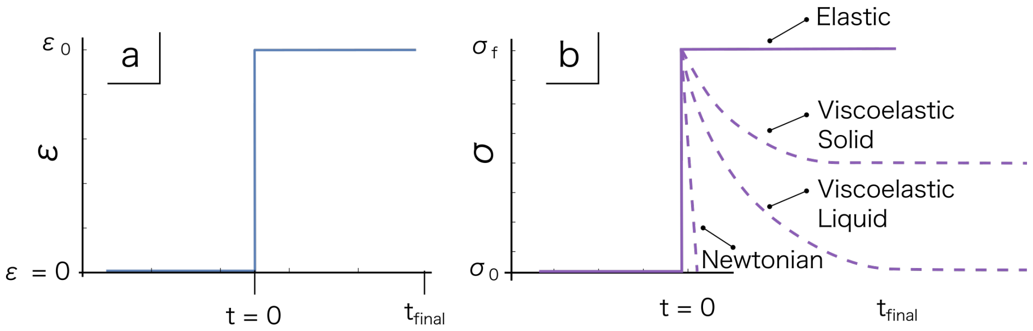


Figure 4.5: A step in the strain **(a)** results in a stress response **(b)** that is either elastic (Hookean), viscoelastic solid, viscoelastic liquid, or purely viscous (Newtonian)

4.3.1 Experimental Procedure

The experimental procedure was straight forward. After the protocol outlined in the aforementioned section *General Protocol* was followed and a stable fiber was formed, the necessary parameters were entered at the command line after opening `PI_step_strain.exe`. For step strain (relaxation tests), these values were the initial length of the fiber (L_0) and the desired magnitude of the strain (ε_0). For step stress (creep tests), the parameters are instead the initial fiber diameter (D_0), the desired magnitude of the stress (σ_0) and the spring constant of the force-sensing pipette (k_{pip}).

In the case of step strain, the pipette is chosen to be rather stiff[1]. In contrast, step stress experiments benefitted from a softer[2] pipette. Spring constants for soft pipettes fell in the range of 5 to 30 $nN/\mu m$ while pipettes with spring constants between 100 and 300 $nN/\mu m$ were considered stiff

4.3.2 Data Processing and Error Analysis

For step strain, the frame rate of the camera was typically around 30 to 40 FPS since the relaxation processes were slow and took on the order of minutes. Temperature stability was crucial since the deflection of a stiffer pipette (above 100 nN/μm) was typically on the order of a micron – already nearing the limit of a suitable signal-to-noise ration before thermal fluctuations. The positions of both the force-sensing pipette and the puller pipette – as well as the fiber diameter – were recorded for each frame. An example of the piezo stage's ability to produce and maintain a sufficiently infinitesimal step is shown in *Figure* 2.17 in the *Device Development* section 2.3.2. The strongest sources of error were movements of the pipettes due to thermal oscillations which either affected the steel structure holding the pipettes or the stresses in the material. Error in the piezo stage's stability due to continuous corrections at the 4nm scale to maintain its position were also accounted for.

4.4 Oscillatory Excitation

The dynamic response of bent-core fibers was a key interest during the research period leading to the study of the their frequency-dependent mechanical response under uniaxial oscillatory excitation. During small amplitude oscillation, or SAOS experiments the rheometer's 0.2 °C temperature stability allowed the temperature dependence of the real and imaginary elastic moduli, viscosities, and the loss coefficient $tan(\delta)$ to be examined. Amplitudes were restricted to very small strains – between 0.6% and 0.9% – to ensure a linear material response.

Non-linear behavior such as buckling and flow from the boundaries was expected not to exist and the fibers were considered volume-conserving during these experiments. The typical frequency range was two decades – between 0.08 and 8 Hz. Initial fiber lengths were typically around 200 μm in order to optimize the operating range of the piezo stage which could produce step sizes no smaller than 4 nm. Therefore, for 0.1% strain, a driving amplitude of 200 μm was required.

[1] *Stiff* here suggests that the pipette does not deflect enough to contribute to the change of length of the fiber during straining

[2] *Soft* here suggests that the pipette deflects strongly so that the stress can be more easily held constant using closed loop control while the strain creeps

4.4.1 Experimental Procedure

The protocol to run a SAOS experiment (or a L(arge)AOS if non-linear effects are pursued) is outlined below. A command-line program written in C++ (*SAOS.exe*) was used to control the piezo stage. The algorithms used to provide the correct positional profiles needed for a clean sinusoidal extensional excitation are discussed in the software section (*Section 2.3.2*) of the *Device Development* chapter. The code is shown in the *Code Snippets* section (*Section A.4*) of the *Appendix*. The command-line asks the user for the following parameters before beginning oscillatory driving.

1. Enter initial Fiber Length in microns, L_0

2. Enter desired strain magnitude, ε_0[3]

3. Enter the desired frequency in Hz, f

4. Enter the desired number of cycles, N

5. Press Enter to run the experiment

Upon running the experiment, the camera began recording video and saving frames at a frame rate reasonably exceeding the sampling rate required given by the excitation frequency, f as first demonstrated by Nyquist [73,78]. For frequencies ranging from 0.08 to 8 Hz, the frame rates ranged accordingly from 1 to 60 FPS in an attempt to exceed the minimal sampling rate by at least a factor of 3 for better signal reconstruction while minimizing unnecessary data accumulation. After the experiment was finished, the camera stopped recording and the images were saved to the hard drive. In order to correct for any temperature fluctuations exceeding reasonably predictable fluctuations, the temperature profile was recorded in parallel during the experiment using the Omega Platinum GUI and was saved to a .CSV file named `temperature.csv`.

4.4.2 Data Processing and Error Analysis

In *Figure 4.6 (a)*, the steps taken to smooth and extract a clear signal from the original sinusoidal movement of the force-sensing pipette as a function of time are shown. The original signal – shown by the blue curve with a circle marking each frame – spans the entire time domain. At around 32 seconds, the orange curve is overlaid and represents some minimal smoothing applied to the original data to account for zero-shift due to thermal fluctuations.Then, at around 44 seconds, the thick line represents a sine wave fitted to the smoothed, orange curve[4]. This sine-fitted curve is

[3] This ε_0 is different from that actual strain magnitude experience by the material, since the force-sensing pipette will need to deflect in order to measure the force. This deflection means the true, material strain will be less

[4] *Figure 4.6* is only intended to show the stages of data smoothing and fitting. The actual portion of the signal used during the analysis was selected by hand upon inspection in order to ensue the cleanest portion of the signal was used.

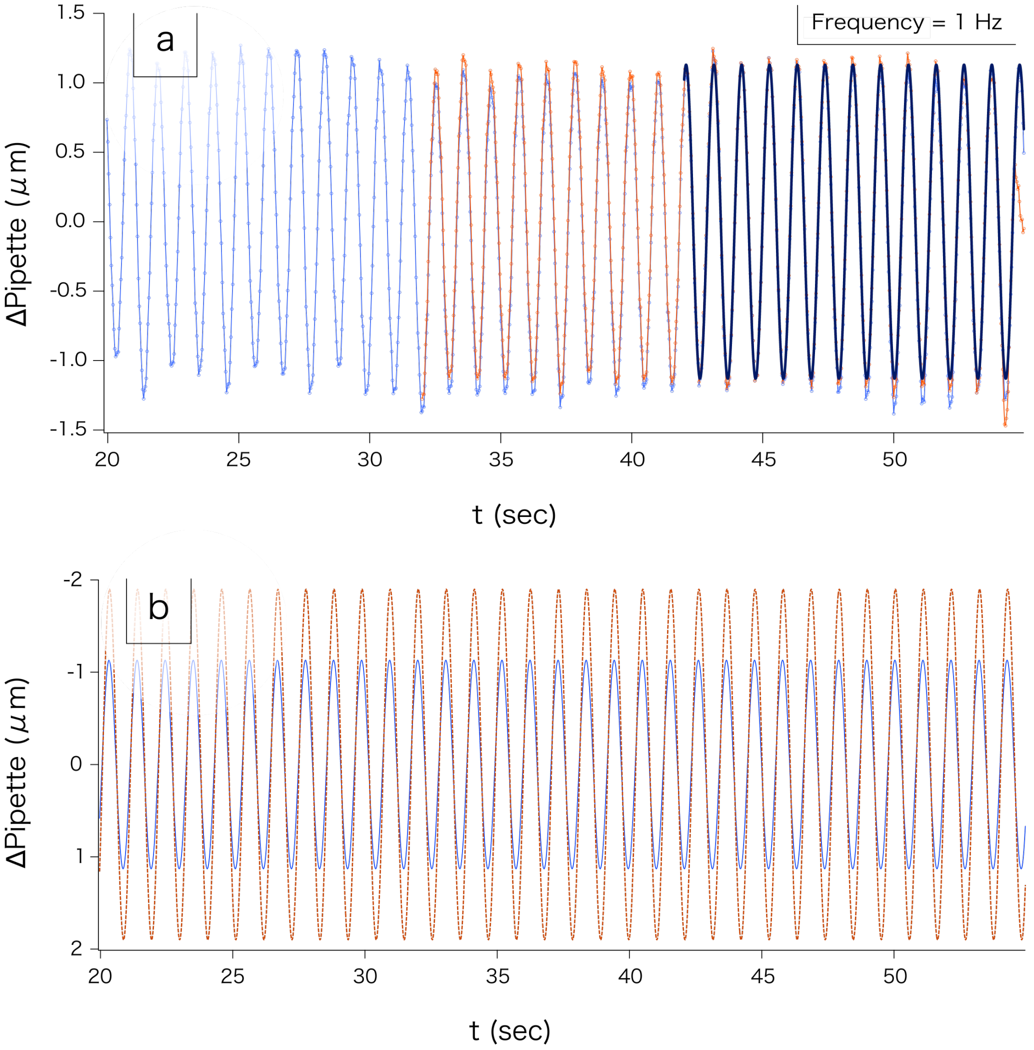


Figure 4.6: (**a**) The original signal of the pipette deflection versus time overlaid with smoothing (middle) and final sine wave fit(right). (**b**) The fitted sinusoidal signal of the puller pipette (larger amplitude) and the force-sensing pipette(smaller amplitude). The actual material strain amplitude, ε_0, is defined as the difference in the amplitudes of these two waves.

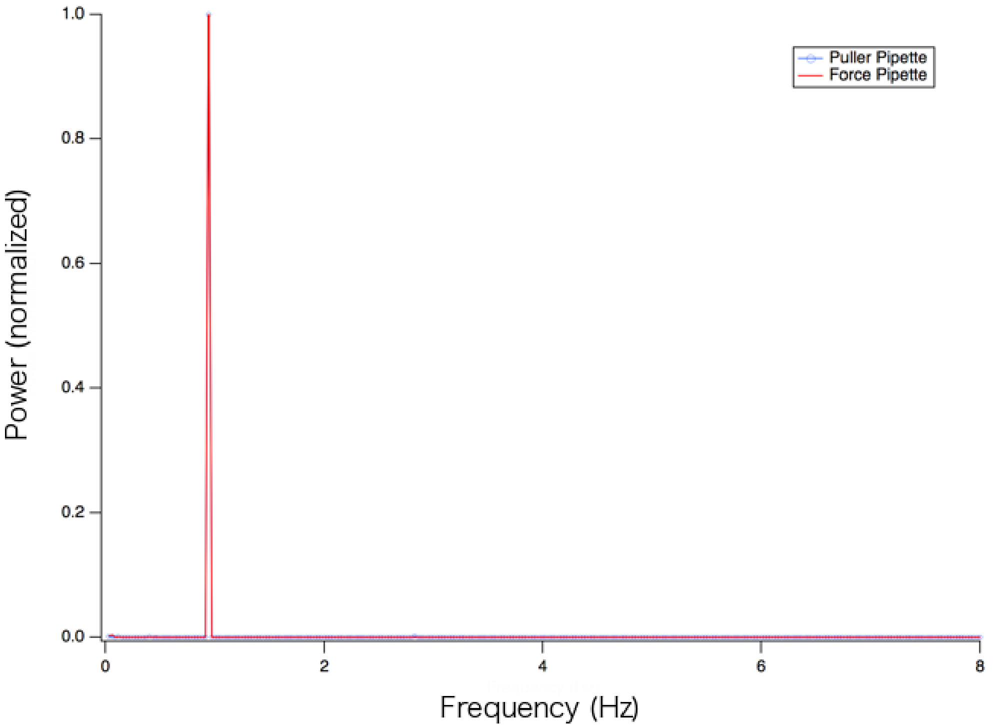


Figure 4.7: The results of a normalized power spectrum resulting from performing a DFT on the position versus time data of the puller pipette and the force-sensing pipette. The spectrums are seen to overlap perfectly with a clear, single spike at the driving frequency.

used to extract the amplitude of the wave. For extracting the phase angle δ between the stress and strain signals, a method using the discrete Fourier transform (DFT) was used instead. *Figure 4.6 (b)* shows the relative displacements of the force-sensing and rigid puller pipettes, with the larger-of-the-two-amplitudes signal belonging to the rigid puller. The difference between these two waves is $\Delta L(t)$, the change in length of the fiber over time. An accurate guess of the strain amplitude for later fitting of a sine wave to the strain wave could be calculated as the amplitude of $\Delta L(t)$ divided by the initial fiber length L_0.

The DFT was performed on both the raw deflections of the force pipette and the puller pipette to extract the peak frequencies. This peak frequency was always the same for both pipettes – as it should have been – since the material should only lag in phase and not frequency. The DFT also showed if any other frequencies, such as those due to thermal oscillations, contributed to the power spectrum in a way that might require further filtering or, in the worst case, dismissal of the experimental data altogether. An example of the normalized power spectrum versus frequency – a typical output of a DFT – is shown in *Figure 4.7*.

The actual stress and strain waves were then calculated from the smoothed, zero-corrected puller and force pipette waves. The strain wave was calculated as

$$\varepsilon(t) = \frac{\Delta puller(t) - \Delta pip(t)}{L_0} \tag{4.7}$$

and the stress as

$$\sigma(t) = \frac{F}{A} = \frac{\Delta pip(t) * k_{pip}}{\pi R_0^2} \tag{4.8}$$

where $\Delta puller(t)$ and $\Delta pip(t)$ are the deflection of the pulling pipette and the force-sensing pipette at time t, respectively. k_{pip} is the spring constant of the force-sensing micropipette and R_0 is the initial fiber radius. Typically, since the deflection curves of the pipettes themselves had already been zero-corrected, the resulting stress and strain curves did not need any further adjustment. Light smoothing and zero-correction were nonetheless applied before sine waves were fitted. The fitting parameters used were typically restricted to within the range of known values with a slight window for error. The equations fitted to stress and strain were

$$\sigma(t) = \sigma_0 \sin(\omega t + \delta_\sigma) \tag{4.9}$$

and likewise

$$\varepsilon(t) = \varepsilon_0 \sin(\omega t + \delta_\varepsilon) \tag{4.10}$$

Parameter	Initial Guess	Lower Bound	Upper Bound
σ_0	$Pip_0 * k_{pip}/A$	$\sigma_i - 5\%$	$\sigma_i + 5\%$
δ_σ	$\delta_{\sigma-DFT}$	$\delta_{\sigma-DFT} - 1\%$	$\delta_{\sigma-DFT} + 1\%$
ε_0	$(Puller_0 - Pip_0)/L_0$	$\varepsilon_i - 5\%$	$\varepsilon_i + 5\%$
δ_ε	$\delta_{\varepsilon-DFT}$	$\delta_{\varepsilon-DFT} - 1\%$	$\delta_{\varepsilon-DFT} + 1\%$
ω	ω_{DFT}	$\omega_{DFT} - 1\%$	$\omega_{DFT} + 1\%$

Table 4.1: Fit parameters used for fitting sinusoidal signals

The initial values and upper and lower bounds are typically set according to *Table 4.1*. Pip_0 is the amplitude of the force-sensing pipette deflection, $Puller_0$ is the amplitude of the rigid puller pipette deflection, A is the cross-sectional area of the fiber. δ_σ and δ_ε are extracted from the imaginary components of the DFT and used to calculate the relative phase angle δ as well as set the initial guess and fitting bounds for the stress and strain sine wave fits. It should be noted that because only the relative phase angle $\delta = \delta_\sigma - \delta_\varepsilon$ between the two signals is needed, there is no need to shift any of the signals to a zero-phase position. An example of the resulting sine fits to the stress and strain signals and the observable phase lag of the stress can be seen in *Figure 4.8*.

Lissajous plots are parametric plots of two sine functions. Although the general form allows for multiple lobes equal to the integer ratio of the two frequencies of the sine functions, Lissajous plots used in oscillatory rheometry are simplified and have a single lobe due to the stress and strain signals (the two sine functions involved in the parametric plot) having equal frequencies such that the ratio of the frequencies is unity resulting in a single lobe. The difference between the amplitudes of the stress and strain signals results in the major axis of this elliptical lobe being steeper for

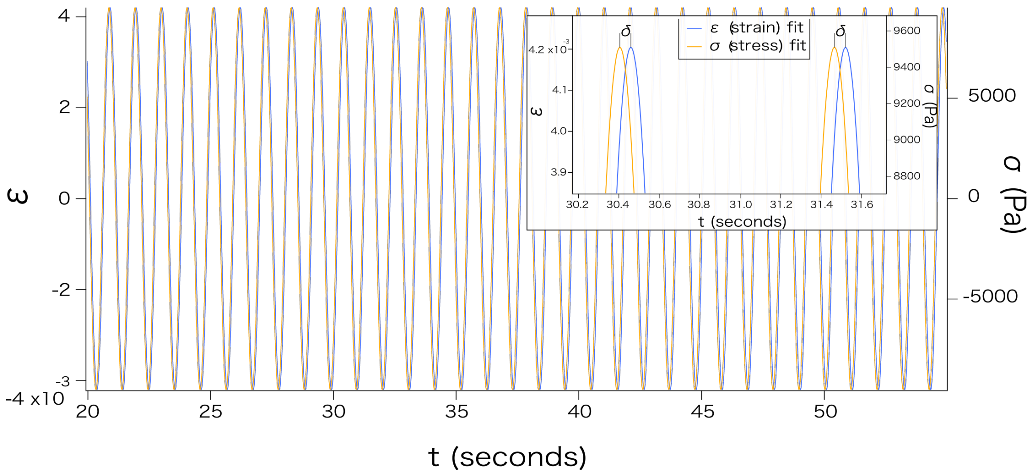


Figure 4.8: Sine fits to the stress (right axis) and stress (left axis) versus time. The inset shows that the stress response lags behind the imposed strain by a phase angle δ.

greater stress/strain ratios or more shallow for lesser ratios. The phase angle (δ) between the stress and strain signals manifests itself in the eccentricity of the ellipse such that a phase angle of zero collapses the ellipse into a straight line co-linear with the major axis while a phase angle of $\pi/2$ results in an eccentricity of zero, or a circle. Such behaviors describe either a fully Newtonian liquid or a fully elastic solid, respectively. An example Lissajous plot from the data above is presented in *Figure 4.9* which includes the original stress and strain signals in blue with the sine fit superimposed in orange.

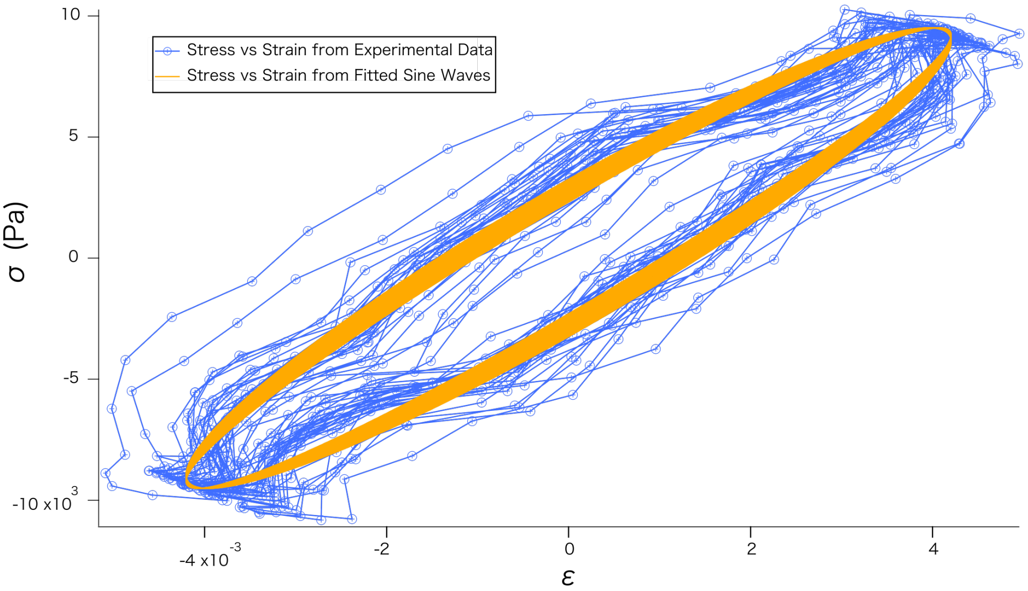


Figure 4.9: Lissajous plot. Blue points are the raw signal and the orange curve the sine-fitted data with initial frequency and phase lag values from the DFT. The thickness of these compound curves represents the fit error.

Error

Full, non-linear error analysis was performed using non-linear propagation of uncertainty. on the stress and strain signals was applied to all resulting parameters. The standard deviation of each variable was calculated using the variance formula,

$$s_f = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 s_x^2 + \left(\frac{\partial f}{\partial y}\right)^2 s_y^2 + \left(\frac{\partial f}{\partial z}\right)^2 s_z^2 + \cdots} \tag{4.11}$$

where the derivatives of the variable f are taken with respect to parameters x, y, and z and s_i are the standard deviations of x, y, and z. *Figure 4.10* shows a visual representation of the signal error. The experimental ellipse (center) is cradled between two other ellipses; one with a steeper slope and a greater phase angle and one with a lesser slope and a lesser phase angle. These two ellipses represent the original (central) ellipse with error either added to, or subtracted from, the signal, respectively. The more tightly these two ellipses cradle the measured signal, the lesser the error. For error ellipses which span a greater deviation in slope of the major axis and broader range of eccentricity, the experimental error is greater. This plot allows a quick and straightforward assessment of the experimental quality.

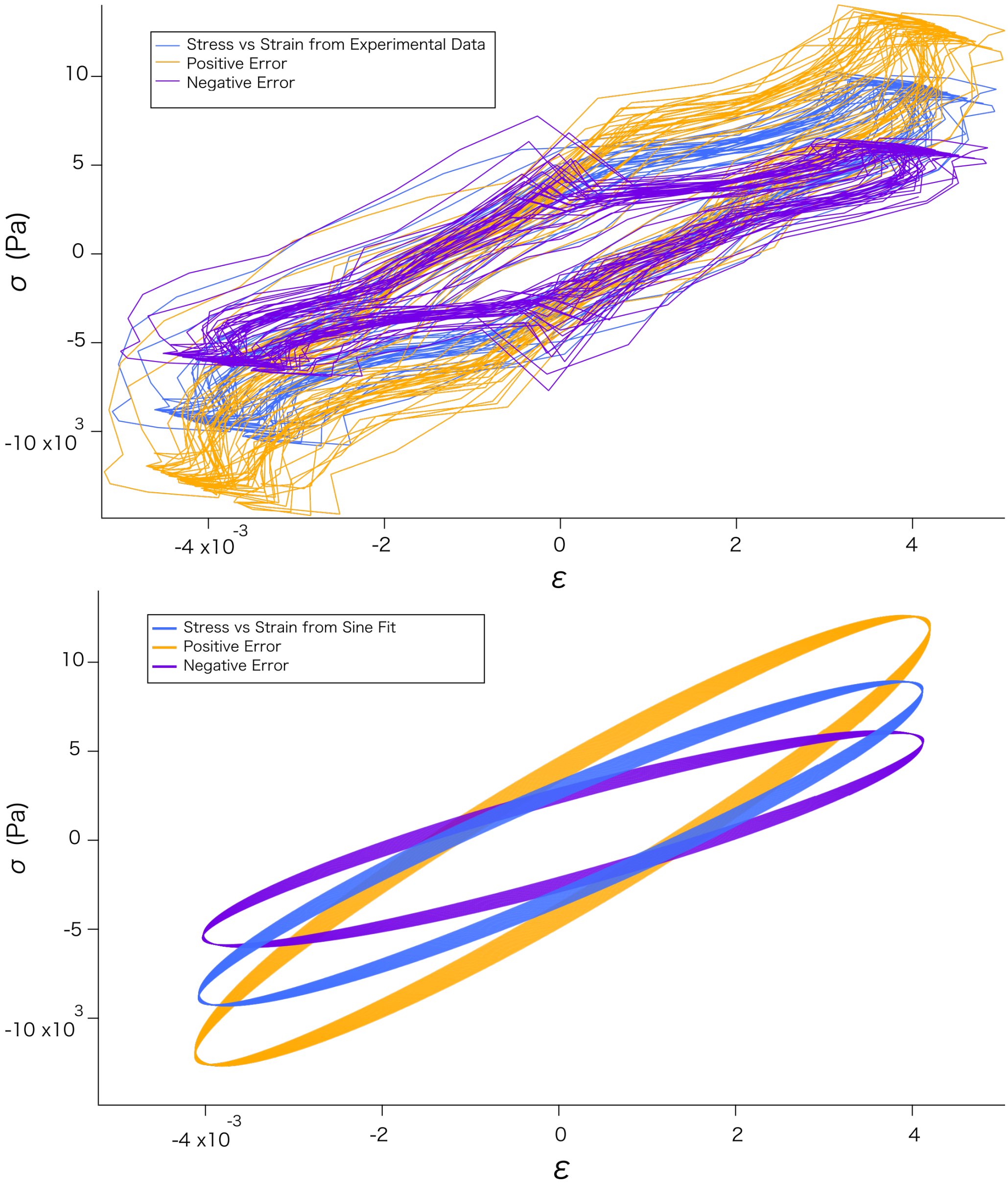


Figure 4.10: Graphical error analysis using positive and negative adjustments to the original stress/strain ellipse.

CHAPTER V

EXPERIMENTAL RESULTS

This chapter will strictly focus on the experimental results. Analysis will be covered in the following *Analysis* chapter. The results will be presented in the order as they are discussed in the *Methods* chapter: *Constant Rate Extension, Step Strain, Small Amplitude Oscillations, Triangular Excitation*. The *Analysis* chapter will then divide the discussion into the different behaviors observed during experiment while analyzing trends and similarities between experimental procedures.

5.1 Constant-Rate Extension

This section will attempt to clearly explain aspects of the recorded data and how its visualization is useful to the understanding of non-linear behavior of bent-core fibers such as necking, flow and rupture. Although the plots obtained from the raw data recorded during a constant strain rate experiment may appear lackluster at first, properly presenting the data under different transformations or scaling reveals unique behavior across discrete regimes of material response. Although the full analysis of these results is reserved for the following *Analysis* chapter, this section will discuss some experimental conclusions.

In order to visualize the response of a single fiber under constant rate extension, we begin with an example of a BLC fiber with an initial diameter $D_0 = 4.2\mu m$ and an initial length $L_0 = 64\mu m$. During this trial, the temperature was held constant at 152 °C and the fiber strained at a constant strain rate of $\dot{\varepsilon} = 5$ while being recorded at 250 FPS at 20X magnification. Presented in *Figure 5.1* are the deflections of both the puller pipette and the force-sensing pipette (left axis) and the radius of the fiber (right axis) versus time, followed by the stress and the strain plotted together versus time to emphasize the stress response of the fiber.The minimum diameter of the fiber was measured during the experiment to calculate the minimum cross-sectional area of the fiber from which the final stress was calculated from the force-sensing pipette's deflection. The strain was calculated as the difference between the deflection of the two pipettes divided by L_0.

The fiber was observed – not just in this trial but in most others as well – to rupture *after* the straining had ceased. The initial regime shows a more-or-less linear behavior up until a peak in the stress signal (yield point) followed by a steady stress dissipation. As seen in *Figure 5.1*, the initial elastic-like behavior was accompanied by a steady reduction in the fiber diameter until, at the yield, the diameter suddenly returned to its initial value before once again thinning. This seems to indicate a *constant volume* elastic response during initial straining until a critical point was reached and flow was initiated at the boundary between the fiber and the excess material at each end allowing the fiber to restore itself. This strain-dependent flow response, however, was occurring while the fiber was still being pulled. Continued straining after the fiber had transitioned into this ”flow” phase shows a stress response more indicative of viscous behavior at the boundaries than flow behavior within the fiber itself. After the pulling stopped, the fiber either remained stable at some new, thinner diameter (while relaxing to the initial stress value) or spontaneously ruptured since rate hardening effects disappeared and the flow continued until rupture.

Having seen the details of the kinematics of a single experiment, the following figures are presented in an attempt to uncover a trend. Each figure covers a different regime of strain rate: *Figure 5.2* covers the lower regime of strain rates from 1.7 to 5.2 while *Figure 5.3* covers the upper regime of strain rates from 5.3 to 9.9. The temperature was held constant at 152 °C for all experiments. In *Figures 5.2 (a)* and *5.3 (a)*, the data is plotted against a linear axis in both the stress σ and the strain ε. The separation of an initial regime, followed by a critical peak, followed by a gradual stress relaxation or possible flow regime are apparent. *Figures 5.2 (b)* and *5.3 (b)* are presented with a logarithmic stress axis and a linear strain axis. The strain axis is cropped to $\varepsilon = 0.5$ to emphasize the initial elastic regime. It is clear that the elastic modulus in this initial regime – the ratio of the stress to the strain – increases monotonically with strain rate indicating rate hardening behavior.

While this behavior is apparent in both the low strain rate regime shown in *Figure 5.2* and in the high rate regime shown in *Figure 5.3*, it is clear that there is a transition in the behavior since the magnitude of the stress for strain rates roughly above 5 are considerably lower than expected. This may be attributed to an immediate hardening at such high rates leading to an equally immediate yield and flow initiation such that the transient flow regime dominates. More on the possible interpretations of such a transition will be discussed in the *Analysis* chapter. The elastic modulus of uniaxial extension under constant rate, E_{ext} is plotted versus the strain rate for both regimes in *Figure 5.4*.

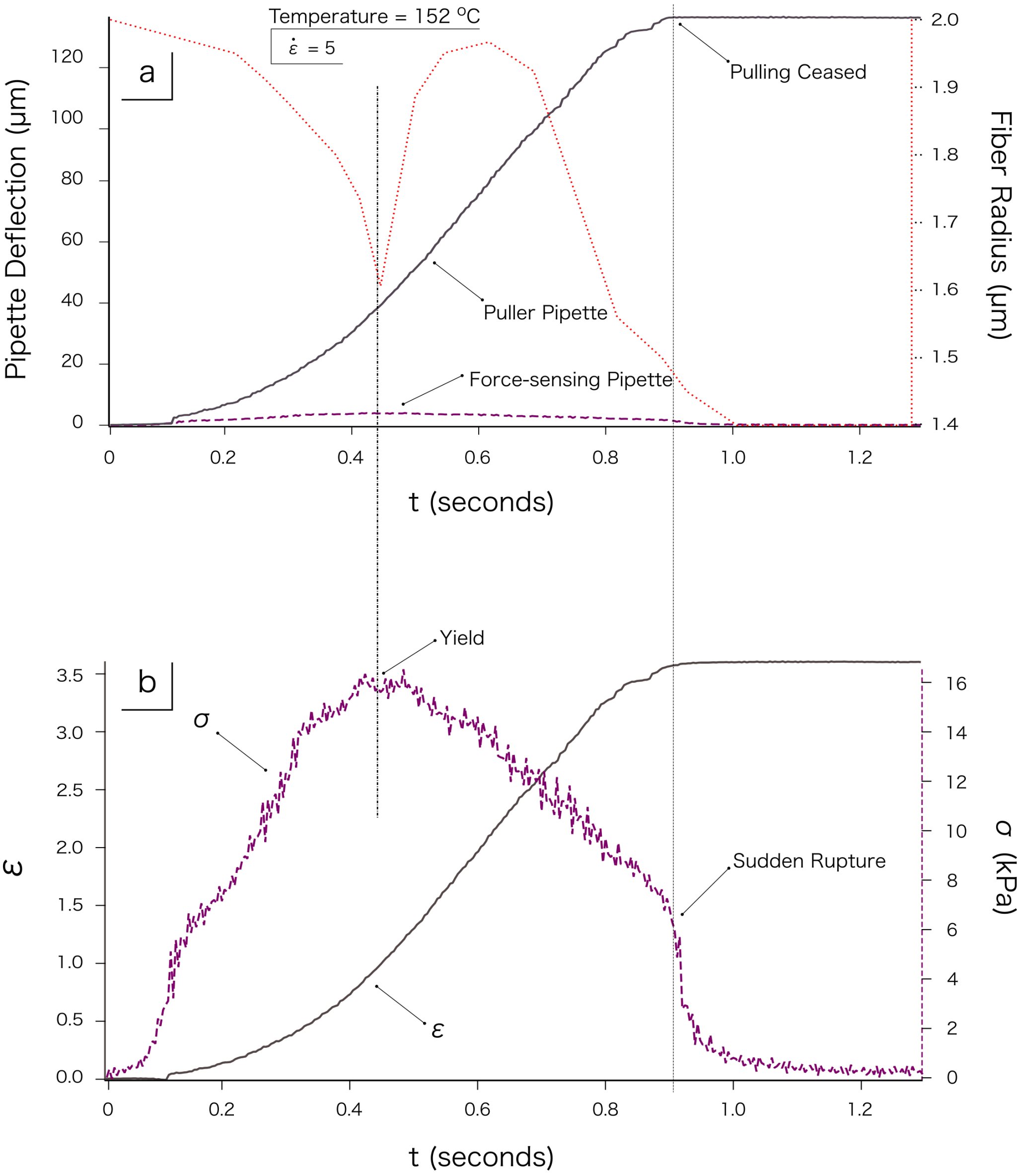


Figure 5.1: (a) The deflection of both pipettes versus time on the left axis and the evolution of the fiber radius versus time on the right axis (red dotted line). **(b)** strain versus time on the left axis and stress versus time on the right axis (dotted line). The fiber was pulled at 152 °C at a strain rate of $\dot{\varepsilon} = 5$. The initial radius ($R_0 = 2.1\ \mu m$ is seen to thin during initial pulling before reached a critical minimum the first vertical dotted line (left) shows this to coincide with the yield in the stress curve below. The second vertical line (right) shows the sudden relaxation after cessation of pulling.

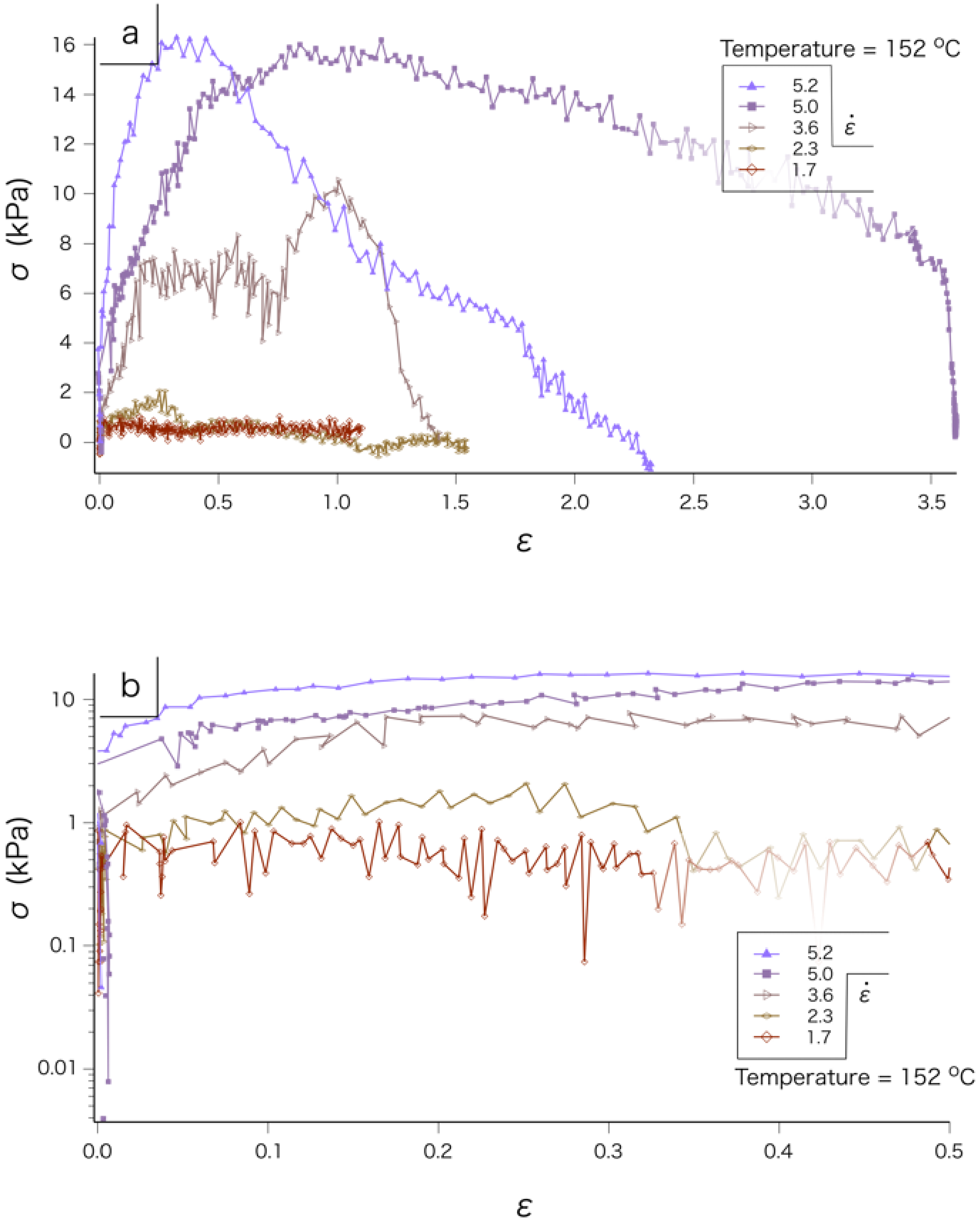


Figure 5.2: **(a)** Stress versus strain curves for the lower range of strain rates. The curve for the fiber pulled at a strain rate of $\dot{\varepsilon} = 5$ (solid squares) represents the data shown in *Figure 5.1*. **(b)** The same data presented on a log-linear plot with the strain axis cropped to $\varepsilon = 0.5$ to emphasize the initial elastic-like regime. Rate dependence of the elastic modulus on extension is clearly seen.

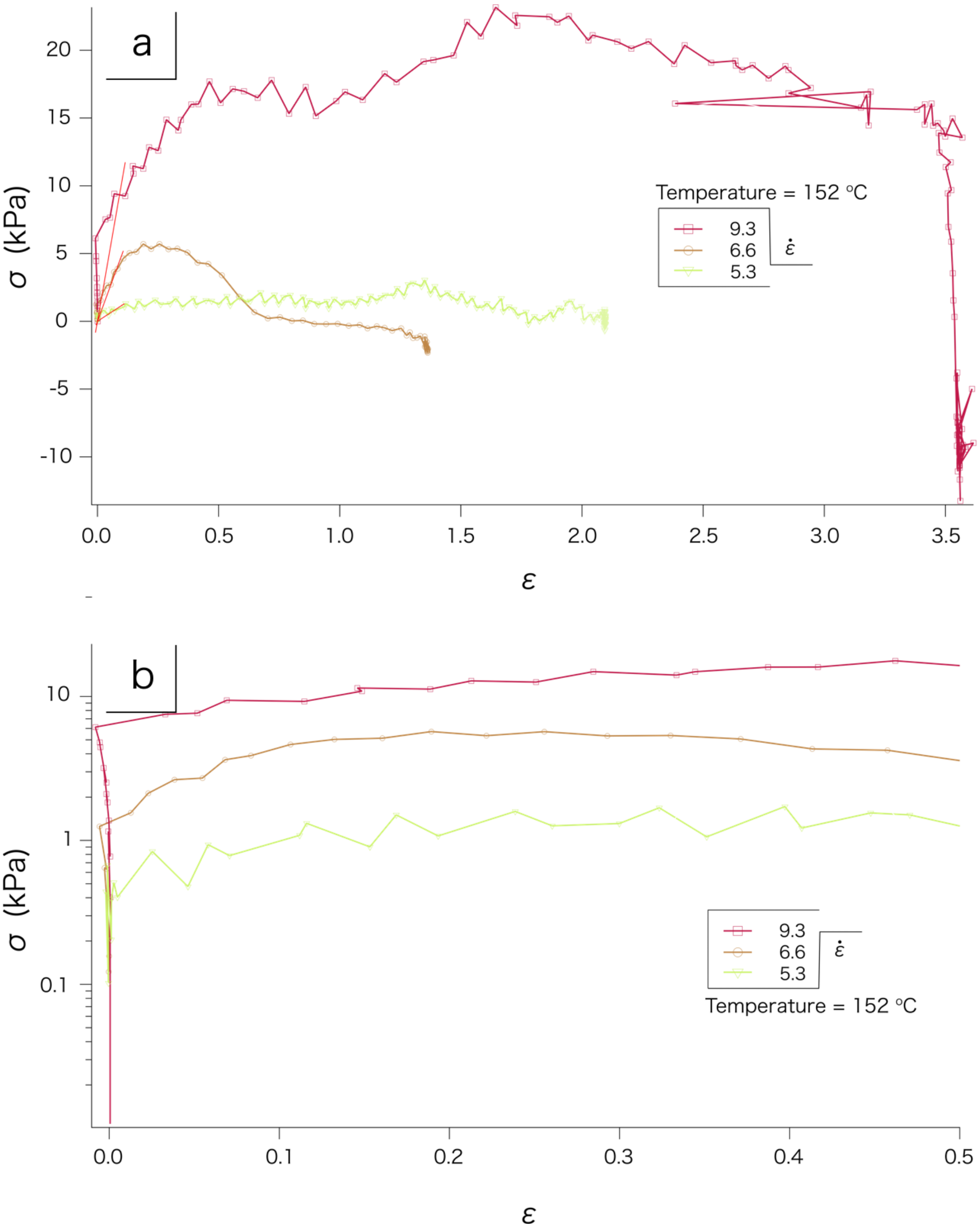


Figure 5.3: (a) Stress versus strain curve for the higher range of strain rates. **(b)** The same data presented on a log-linear plot with the strain axis cropped to $\varepsilon = 0.5$ to emphasize the initial elastic-like regime. Rate dependence of the elastic modulus on extension is clearly seen.

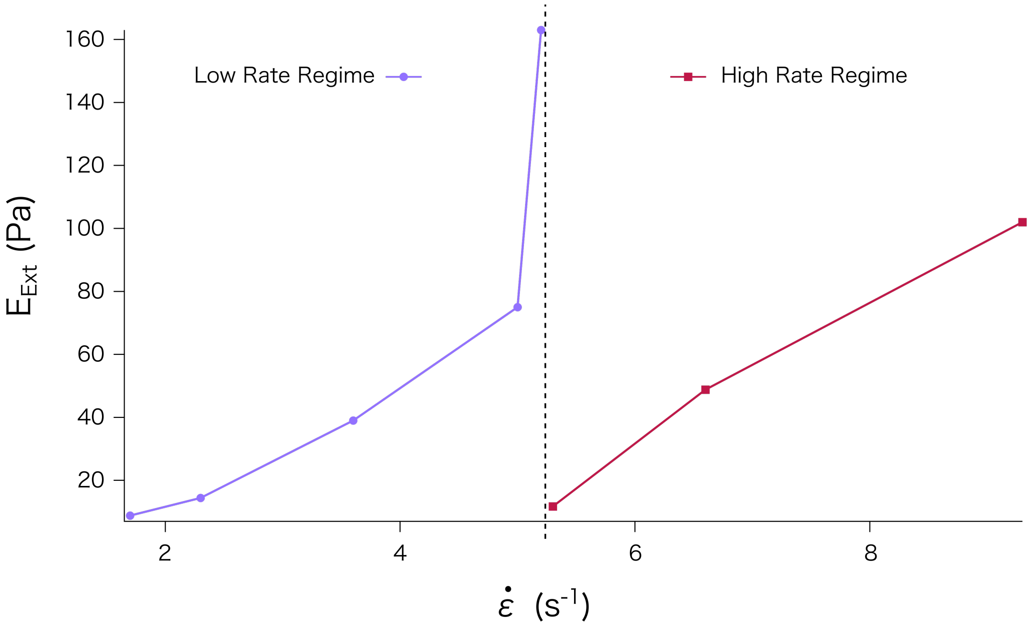


Figure 5.4: Elastic modulus of extension (E_{Ext}) versus strain rate. Both the low rate regime and high rate regime show clear rate hardening trends. The dotted vertical line separates what seems like two different material regimes – one for low rates and another for high rates of constant extension.

5.2 Small Amplitude Oscillations

Data from SAOS experiments are presented in this section. Lissajous curves are shown to visually examine the quality of the experiment and to verify the frequency dependency of Young's modulus (slope of the major-axis of the ellipse) and phase angle (eccentricity). An eccentricity of zero (a circle) indicates pure Newtonian (viscous) behavior where the stress lags the strain by a phase angle of $\delta = \pi/2$. When the eccentricity reaches unity, the stress responds without any lag to the strain ($\delta = 0$) indicating no dissipative or otherwise time-dependent processes and the material response is thus called purely elastic or Hookean. Phase angles between these limits indicate a response which is both viscous and elastic in nature, i.e. viscoelastic. Lissajous plots carry a vertical axis representing the change in stress, $\Delta\sigma$, and a horizontal axis representing the strain, ε.

Stress as a function of time is denoted $\sigma(t)$ while the stress amplitude is denoted σ_0. Likewise, the strain as a function of time is denoted $\varepsilon(t)$ while the strain amplitude is denoted ε_0. It should be noted that $\varepsilon(t)$ and ε_0 refer to the actual *material strain* which is different from the strain that was input into the rheometer program. This input strain – the strain initially desired by the program's user – refers to the displacement amplitude of the puller pipette divided by the initial length of the fiber and is denoted here as ε_{inp}. The actual material strain is then slightly less, depending on the stiffness of the force-sensing micropipette. An in-depth description of how the experiments were performed and the data compiled are presented in the *Methods* chapter in *Section 4.4*.

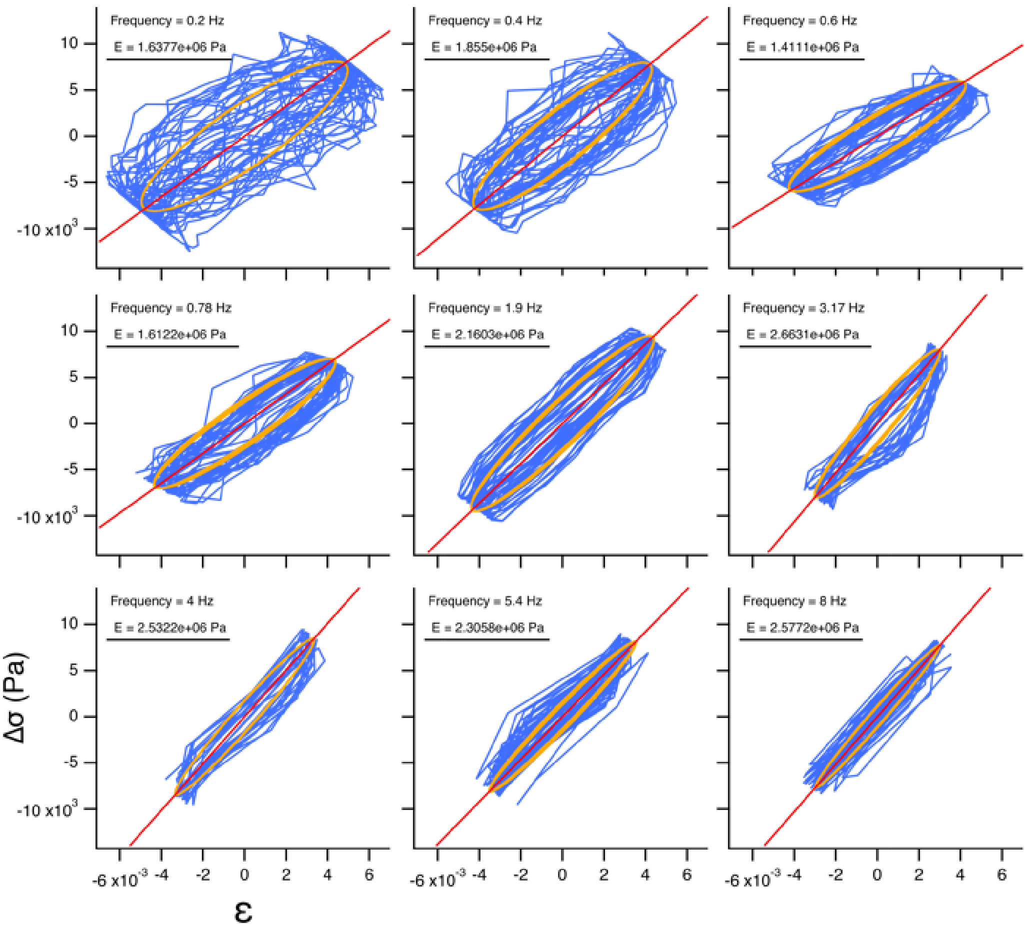


Figure 5.5: Lissajous plots with increasing driving frequency f from top left to bottom right for a BLC fiber held at a constant temperature of 150 °C. The blue curves show the actual data extracted from the motion of the pipettes, the yellow ellipses show the sine fit to the data, and the straight red lines show the storage modulus, E'.

In *Figure 5.5*, nine Lissajous plots are presented in order of increasing driving frequency at a constant temperature of 150 °C. The input strain was kept constant at $\varepsilon_{inp} = 0.01$. 50 driving cycles were completed for each frequency. In this example, the same fiber was used for each experiment and its initial length L_0 remained at 200 μm throughout the entire session. The mean fiber diameter (initial diameter averaged across all experiments) was 1.37 μm and varied by less than $0.12\mu m$. Drift of the raw data – seen in the blue curves for low frequencies – in the direction parallel to the minor axis of the ellipse can be attributed to low-frequency oscillations in the temperature. For low driving frequencies, the experiments lasted longer than the 30-second oscillation period of the temperature. For higher frequencies, this temperature oscillation did not reveal itself in the experimental data. The red lines show the linear fit to the raw data which represents the storage modulus E'.

The six plots shown together in *Figure 5.6* show the material values extracted from *Figure 5.5* in which the temperature was held constant at 150 °C. An increase in the storage modulus and seemingly steady loss modulus with respect to frequency indicate rate hardening behavior.

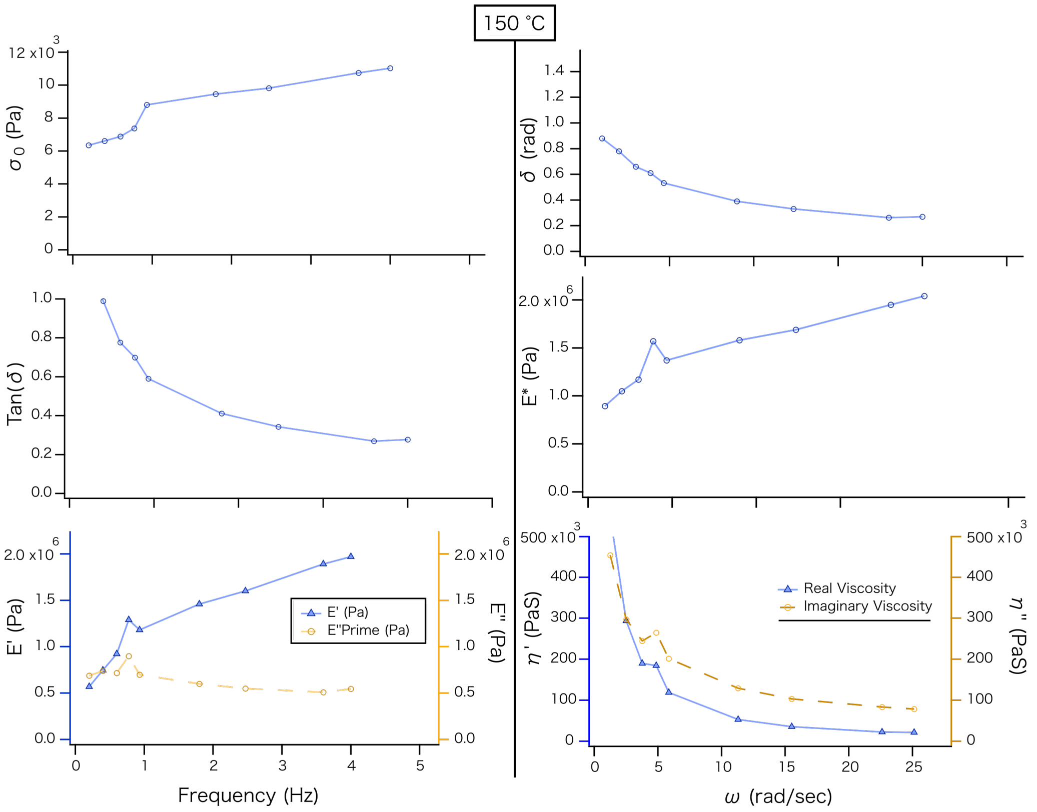


Figure 5.6: Measured and calculated parameters versus frequency. In order from left to right, top to bottom: Stress magnitude (σ_0); phase angle (δ); loss coefficient ($\tan\delta$); magnitude of the complex modulus ($|E|$), storage and loss moduli (E’ and E”); real and imaginary components of the complex viscosity ($\eta' and \eta''$).

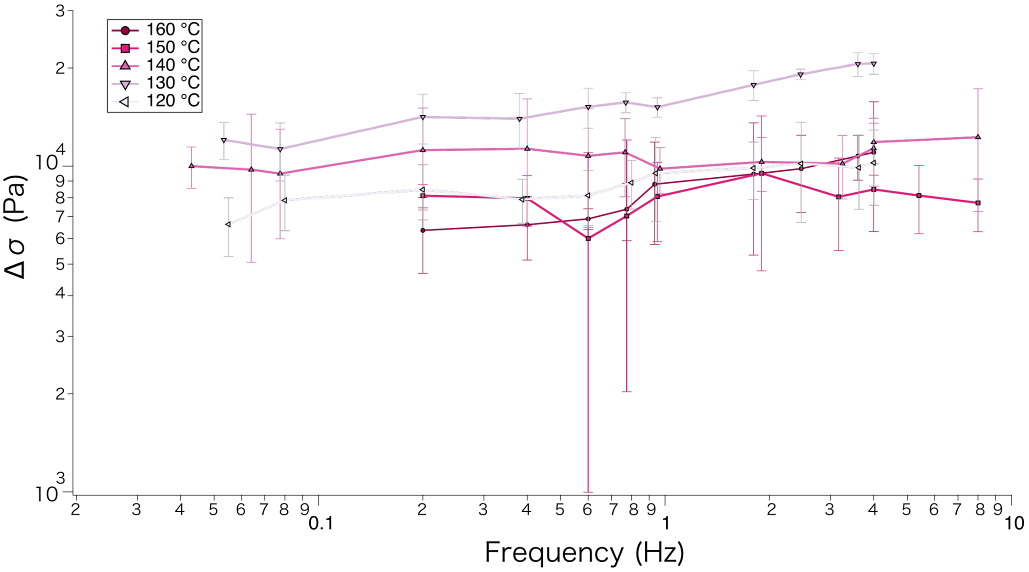


Figure 5.7: Relative stress amplitude $\Delta\sigma_0$ versus frequency for temperatures ranging between 120 and 160 °C

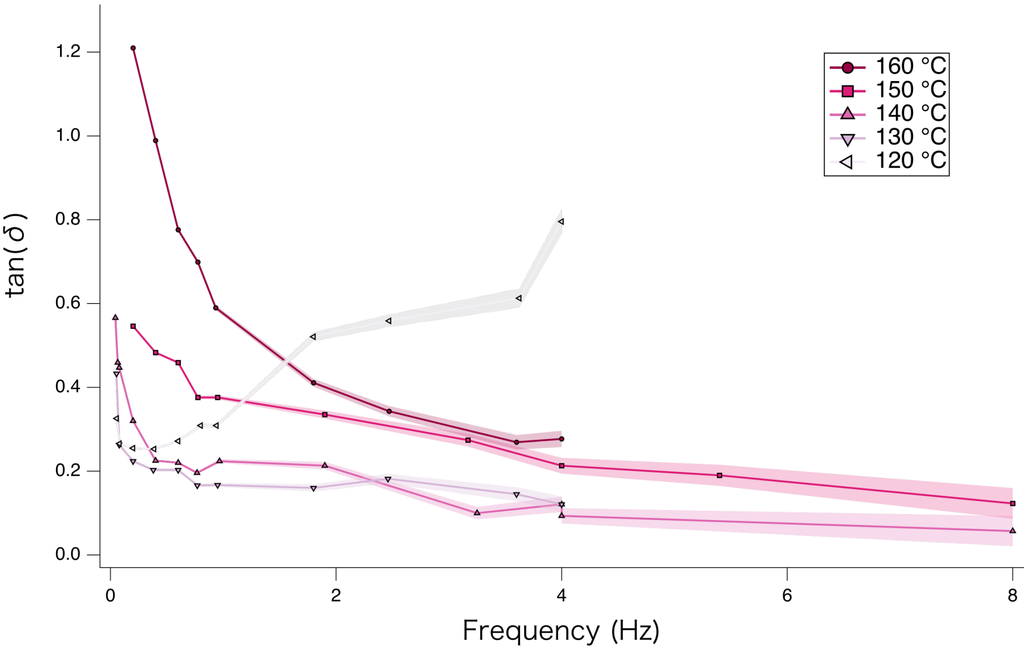


Figure 5.8: Loss coefficient, tan(δ) versus frequency for temperatures ranging between 120 and 160 °C

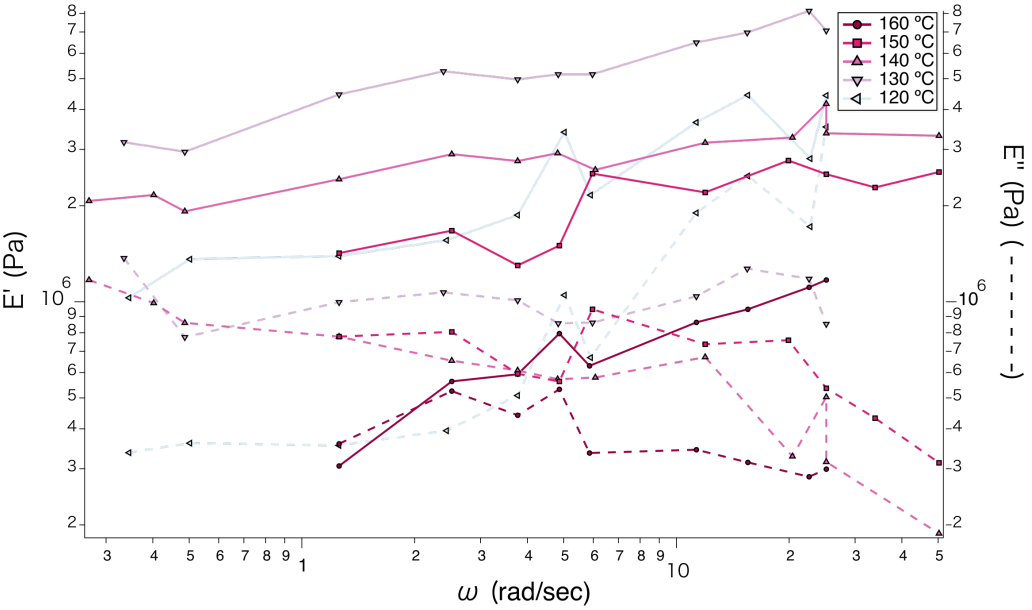


Figure 5.9: Storage modulus (E') and loss modulus (E'') versus frequency for temperatures ranging between 120 and 160 °C

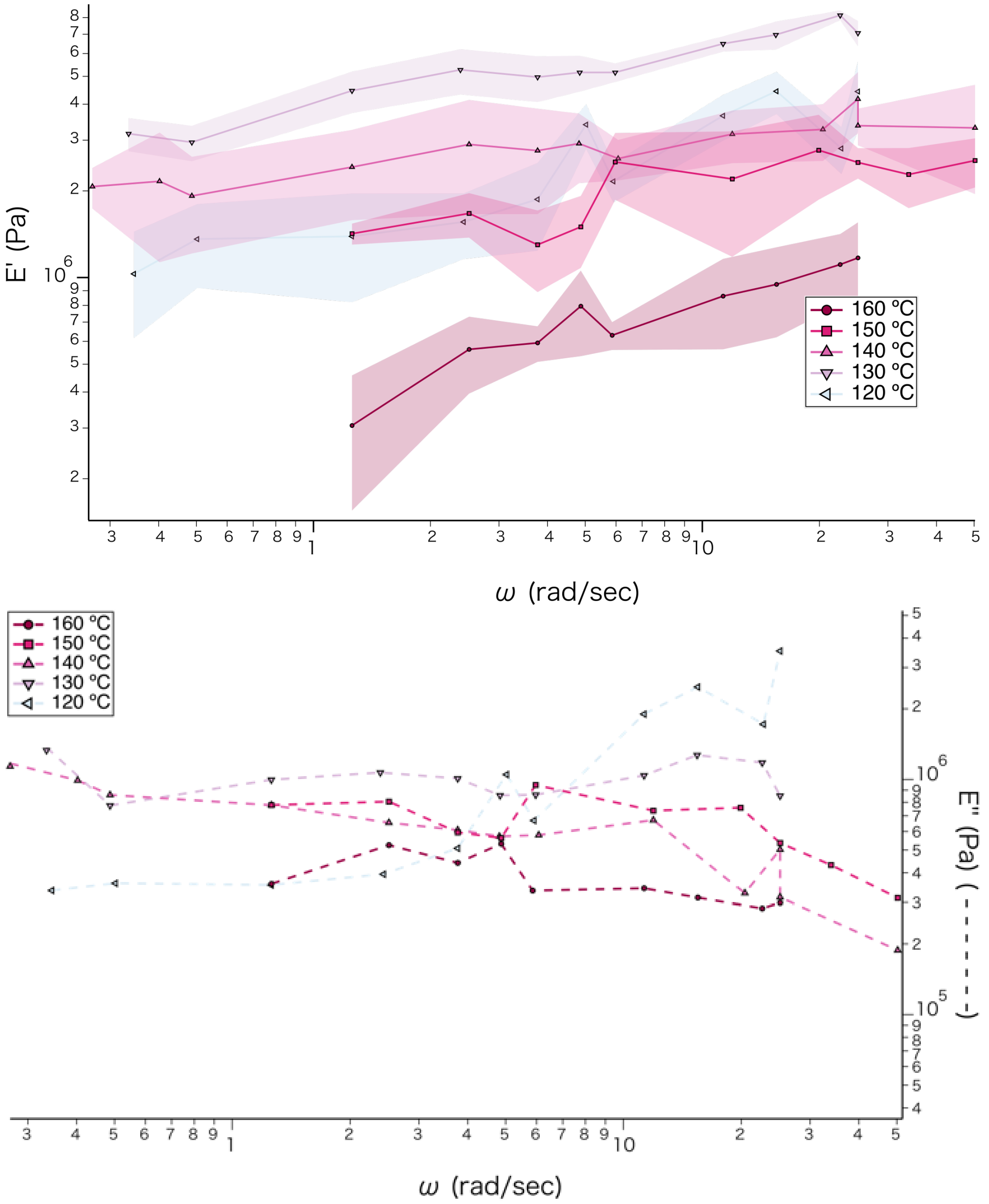


Figure 5.10: Storage Modulus (E') only (upper) and Loss Modulus (E'') only (lower) versus frequency for temperatures ranging between 120 and 160 °C shown with errors

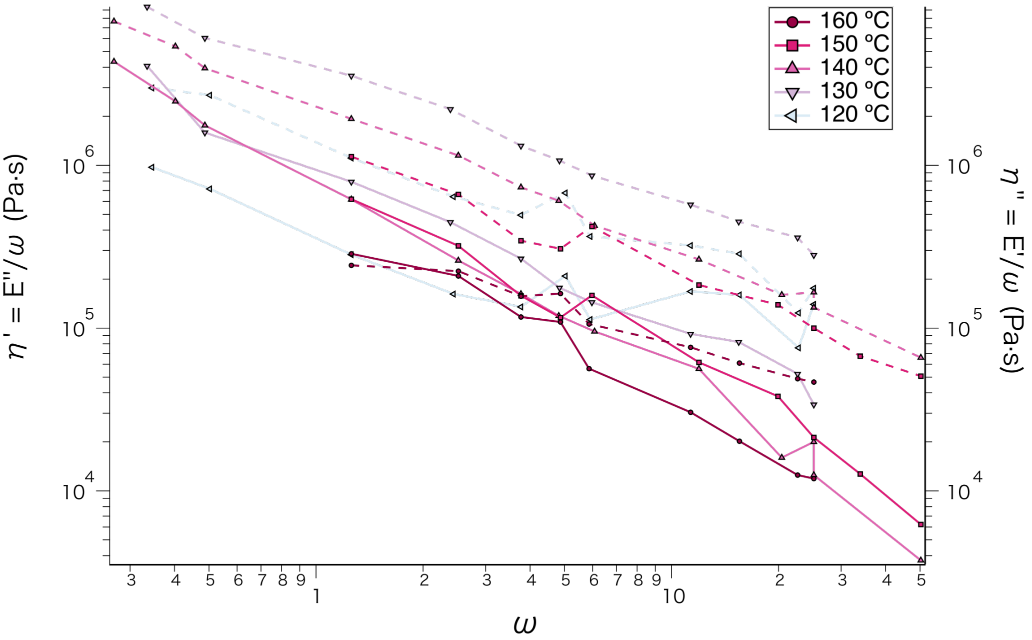


Figure 5.11: Real (solid lines) and imaginary (dotted lines) components of the complex viscosity (η' and η'') versus frequency for temperatures ranging between 120 and 160 °C

Figure 5.15 shows the the smooth surface made by the loss tangent, $tan(\delta)$, when plotted against frequency and time. It shows that the low temperature, high frequency regime tends towards a zero loss angle while the low frequency, high temperature regime grows towards a finite loss tangent.

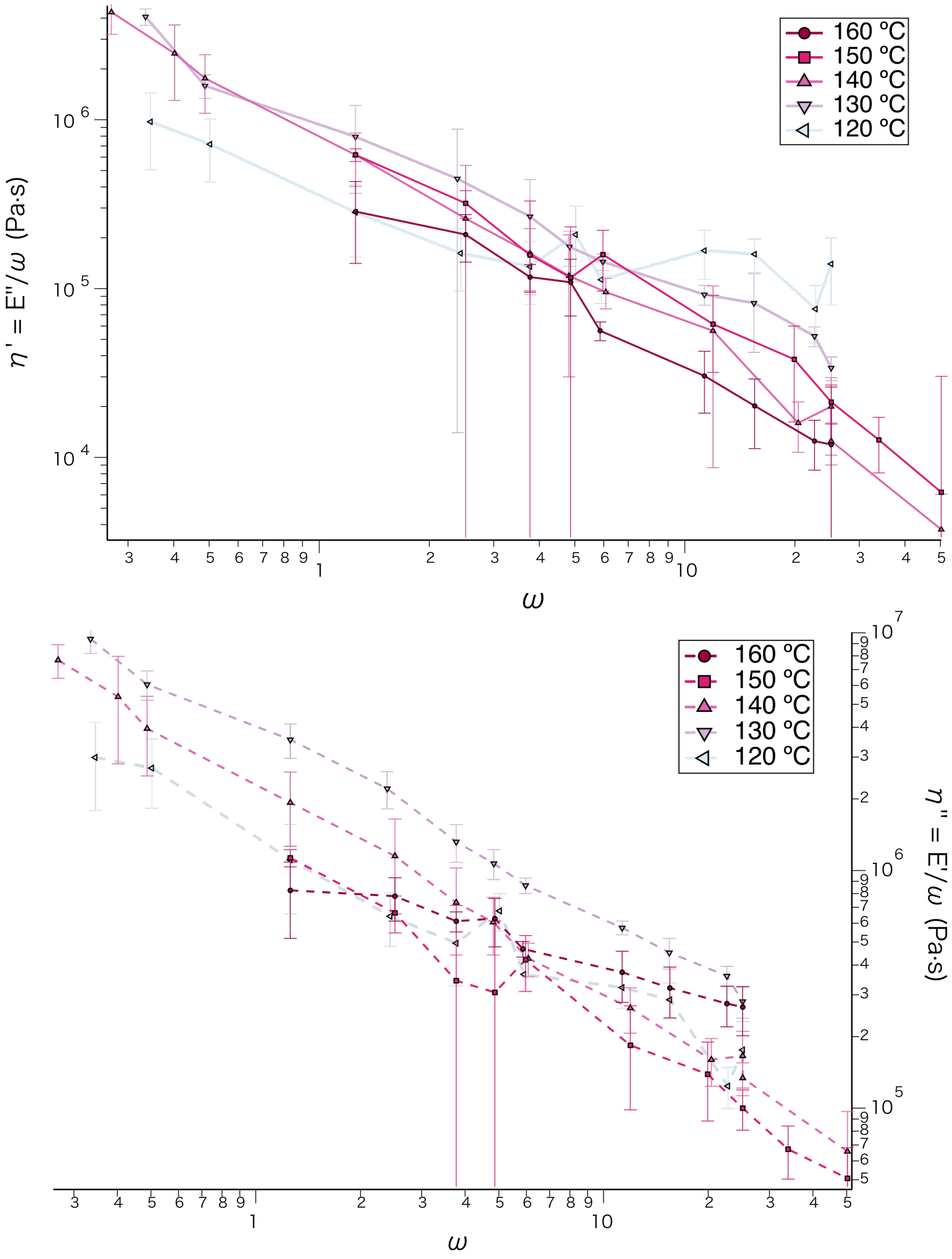


Figure 5.12: Top: Real only and **Bottom:** imaginary only components of the complex viscosity (η' and η'') versus frequency for temperatures ranging between 120 and 160 °C shown with errors

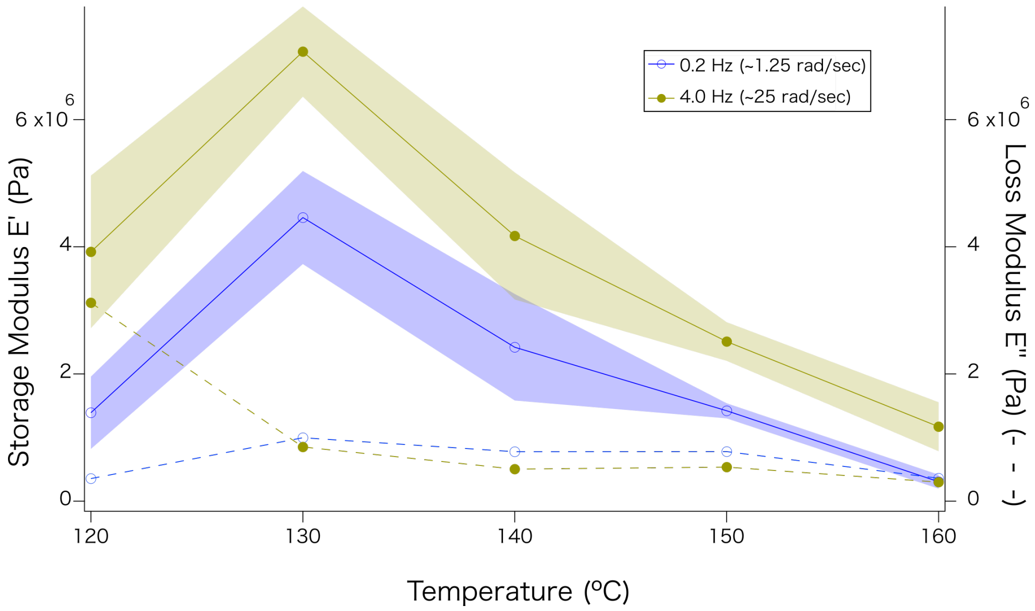


Figure 5.13: E' (solid) and E'' (dashed) versus temperature for the most common lower bound of the driving frequency (0.2 Hz) to the most common upper bound at 4 Hz.

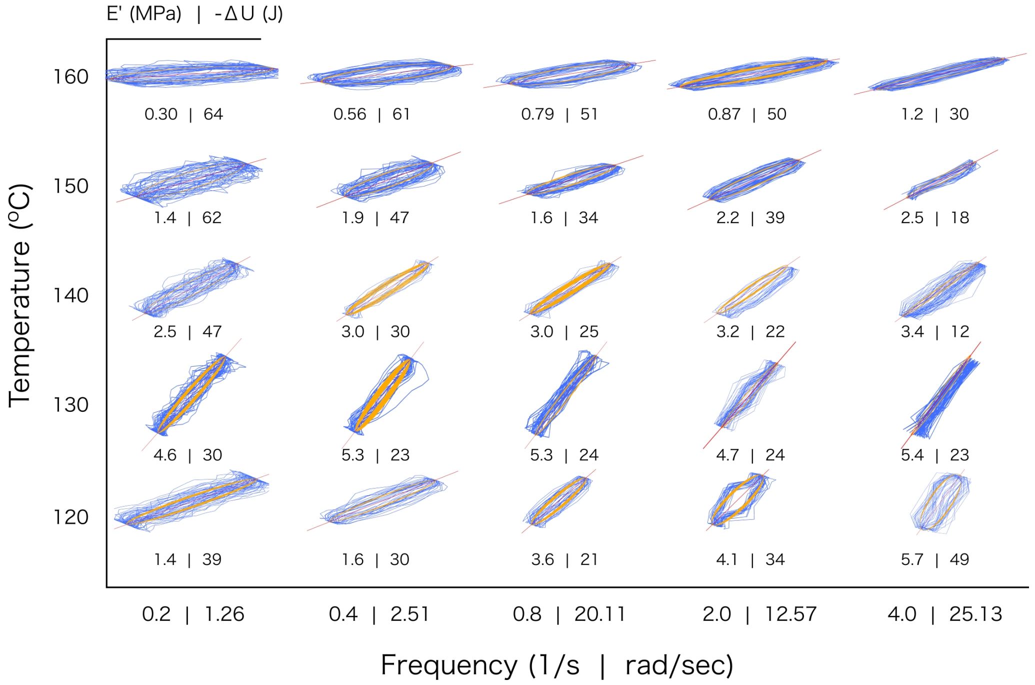
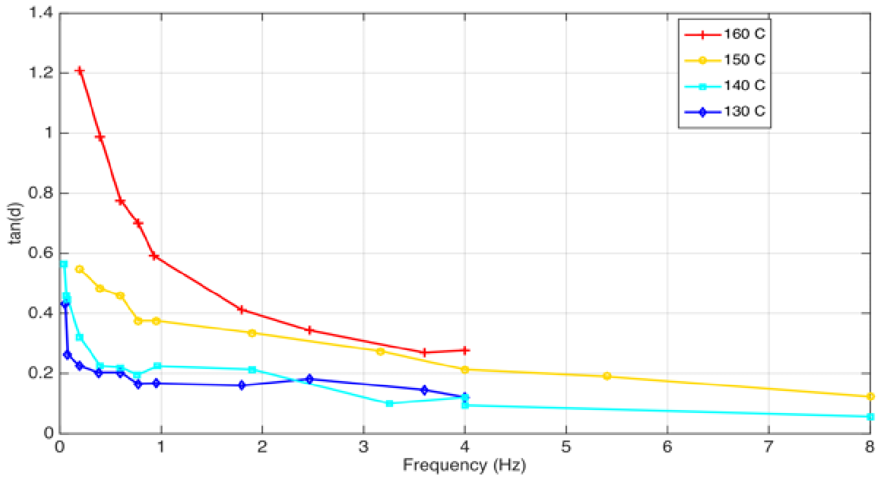


Figure 5.14: A Pipkin diagram summarizing the frequency- and temperature-dependent behavior of BLC fibers. The left axis increases vertically with temperature, while the bottom axis increases to the right with frequency. Each Lissajous plot is labelled with its storage modulus E' and dissipated energy per cycle, ΔU.



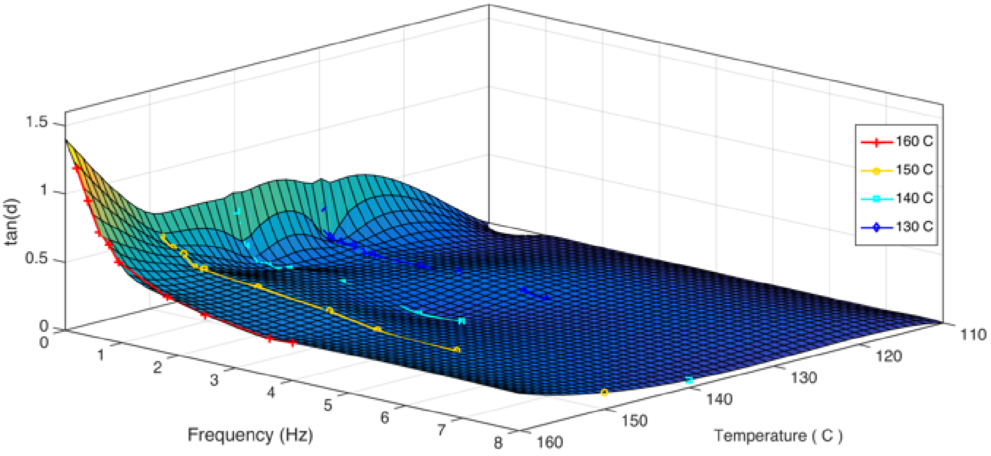


Figure 5.15: Top: The loss tangent $tan(\delta)$ is plotted versus frequency across the temperature range. **Bottom:** The same curves are plotted as a surface plot versus temperature and frequency to show the trend of the loss tangent across the smooth manifold.

5.3 Triangular Excitation

Preliminary data is provided to show the effects of bent-core fibers under triangular excitation. A very brief analysis will be discussed here since triangular excitation analysis did not play a major role in the viscoelastic analysis carried out by the aforementioned methods and experiments. It was intended to show the feasibility of of such a mode of excitation should the user of the rheometer desire such an analysis. *Figure 5.16* shows the stress versus strain, stress versus time, and strain versus time profiles of a BLC fiber at a constant driving frequency at temperatures varying between 125 and 135 °C.

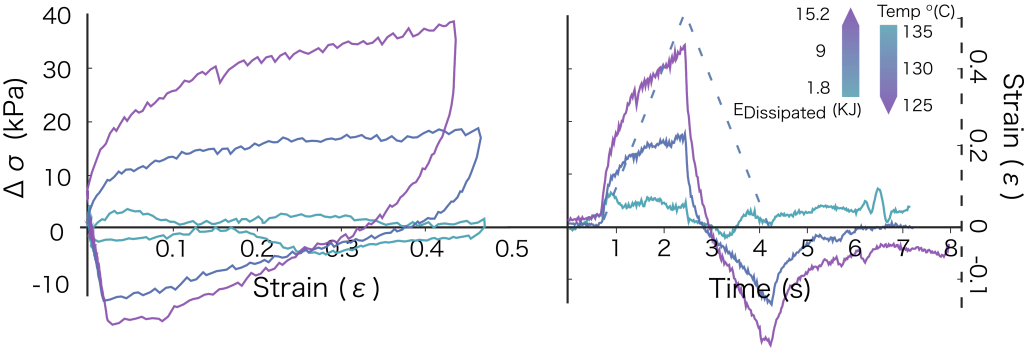


Figure 5.16: Preliminary data is shown for a BLC fiber under triangular, uniaxial extension. The left plot shows stress versus strain while the right plot shows stress versus time. The dotted line in the right plot corresponds to the right-most axis and indicates the strain of the fiber versus time.

CHAPTER VI

ANALYSIS

The linear and nonlinear elastic and viscoelastic behavior of complex fluids, both in their generality and in the context of the analysis of the behavior of the BLC fibers relevant to the current study, will be discussed in this section. Analogues to the behavior of viscoelastic solids – which store energy as time goes to infinity past the moment of excitation – and viscoelastic fluids – which dissipate their stored energy over time such as dilute polymer solutions – and their similarities to BLC fibers will be addressed. It will become apparent that BLC fibers possess characteristics of pre-strained, elastic fluids such as a Boger liquid, show a plastic yield like in solids, store elastic energy over infinite time such as viscoelastic solids and can self-repair through yield-induced flow at the boundaries and smectic layer-restructuring. It is also apparent that a transition between elastic and flow occurs in a way much different than in polymers: a yield strain or stress can induce a phase transition which allows the boundaries of the fiber to flow until straining has stopped, upon which the transition reverses and the fiber re-stabilizes. The possibility of a frequency-temperature-driven phase transition is briefly discussed.

6.1 Elastic Response

The elastic behavior – is the time independent linear relation between stress and strain – is a direct consequence of the nature of solids. While the deformations of fluids are defined by the velocities and velocity gradients of their molecules, solids are defined by the relative change in position of neighboring molecules: the overall molecular arrangement does not change, just their relative distances from each other. In a liquid crystal, the boundary between solid and liquid is blurred; sometimes the material's behavior sits more closely to the boundary of a solid, while for others the behavior more closely emulates that of a fluid [30, 76, 79]. In the case of nematics, the behavior is as close to fluid-like as possible for a liquid crystal. For smectics, the behavior tends to be solid-like along at least one spacial dimension [79, 80]. In the case of smectic BLC Fibers which are confined to a highly-stabilized cylindrical geometry with a phase discontinuity at the boundaries of its termini, the behavior becomes more complex. To understand the most simple aspect of their behavior, the elastic regime is considered. In this case, only the elastic contributions – those responsible for a linear, Hookean response under infinitesimal strain – are considered on the basis of the proposed molecular structure presented in the *Materials* chapter.

Three primary contributions to the elasticity are considered: the surface tension, the bulk (compression) modulus and the polarization splay. The energy of extension is therefore the work needed to overcome surface tension, layer compression, and polarization splay, such that the total energy required to strain the fiber is given by

$$\Delta W(\varepsilon) = \Delta W_T + \Delta W_B + \Delta W_p \tag{6.1}$$

where T, B, and p subscripts refer to surface tension, compression, and polarization splay, respectively and the strain, ε, is

$$\varepsilon = \frac{\Delta L}{L_0}. \tag{6.2}$$

A key to this approach is to assume that for small strains there is no flow at the boundaries and that the fiber acts elastically and thus acts like an elastic cylinder of constant volume as opposed to the analysis by Morys *et al.* in 2012 [4]. There may be flow in the form of mass transport since, during strain, there must be be a displacement field centered around the midpoint of the fiber known as the *stagnation point*. The constant volume is expressed as

$$V = \pi R^2 L \tag{6.3}$$

The change in volume is then

$$dV = \frac{\partial V}{\partial L} dL + \frac{\partial V}{\partial R} dR = \pi R^2 dL + 2\pi R L dR = 0 \tag{6.4}$$

which yields the relationship between extensional ε_{ext} and radial ε_r strain

$$\varepsilon_{ext} = \frac{\Delta L}{L_0} = -2\frac{\Delta R}{R_0} = -2\varepsilon_r. \tag{6.5}$$

The factor of 2 in *Equation 6.5* is the inverse of the *Poisson ratio* – the ratio of a materials response to compression or extension in the directional orthogonal to the applied force. In the simplest case, the Poisson ratio of 0.5 [74, 75] corresponds to volume conservation due to incompressibility. Therefore – in the case of a fluid-like column under uniaxial extension such as is the case for BLC fibers – conservation of volume mandates a radial *compressive* strain (indicated by the minus sign) equal to one half of the extensional strain.

Energy Cost from Surface Tension

The work done by the surface in the form of surface tension as a function of strain, ε, is given by

$$\Delta W_T = T_{surf} \Delta L = T_{surf} L_0 \varepsilon = \gamma 2\pi R L_0 \varepsilon \tag{6.6}$$

where γ is the surface tension (assumed to be $20mN/m$ in our case) and R is the radius of the fiber.

It should be noted that $R = R(\varepsilon)$ since we assume a constant volume during small strain extension as we present here in the current theory, meaning the energy cost of extension due to straining the surface should decrease slightly with strain[1]. More will be discussed at the end of this section.

Energy Cost from Compression Modulus

In a simple fluid, or even a polymer, radial compression to the extension under constant volume conditions does not contribute to the work needed to strain the fluid column: the material can flow in radially to compensate for the extension. Smectic BLC fibers, on the other hand, possess a concentric layered structure which has an intrinsic compression modulus resisting radial fluctuations. The resistance to radial fluctuations provided by the smectic layers is likewise responsible for the fiber's stabile nature in the first place. Without it, the Rayleigh-Plateau stability would dominate and fiber stability would cease [3,41]. Since we are expressing our extension and compression in terms of strain, the percent change in layer thickness is the same as the percent change in total radius. Therefore, we can couple the energy cost of compressing the smectic layers that make up the fiber to the extensional strain.

First, a value for the compression modulus is derived based simply on a Laplace pressure balance. The fiber is considered an infinitely long fluid cylinder with two principle radii of curvature, R_1 and R_2, as shown in *Figure 6.1*.

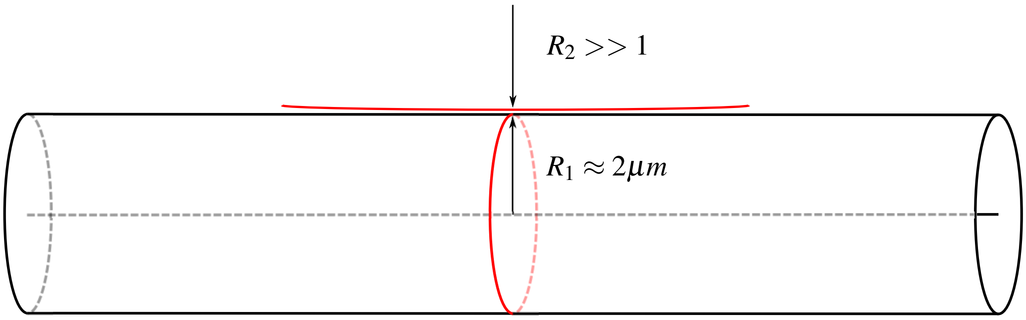


Figure 6.1: A fluid cylinder with the two orthogonal principle radii R_1 and R_2. The first radii, R_1, is that of the fiber itself while the second,R_2 approaches infinity.

The Laplace pressure is given by

$$\Delta P = \gamma \left(\frac{1}{R_1} + \frac{1}{R_2} \right) \tag{6.7}$$

where γ is the surface tension of the interface. Since, in our case, R_2 is taken to be infinitely large,

[1] This can, however, typically be neglected for infinitesimal strains

the curvature ($1/R$) is zero so that

$$\Delta P = \gamma \left(\frac{1}{R_1} \right) \tag{6.8}$$

ΔP is the pressure balance across the interface, i.e. the pressure balance between the ambient air pressure and the internal pressure of the fiber acting normal (radially outward) to the surface. This internal pressure is then equal to the compression modulus, B such that

$$B = \gamma \left(\frac{1}{R_1} \right) + P_{air} \tag{6.9}$$

where $P_{air} = 10^5 Pa$, $\gamma = 0.02 mN/m$ and $R_1 = 2\mu m$ yields a compression modulus of

$$B = 1.1 \times 10^5 Pa. \tag{6.10}$$

Next, the volumetric elastic potential energy density resulting from radial strain is derived. The energy stored in a linear elastic material can be expressed by the integral of Hooke's law over the potential range of strain, which is to say

$$\mu(\varepsilon_r) = \int B\varepsilon_r d\varepsilon_r = \frac{1}{2} B\varepsilon_r^2 \tag{6.11}$$

where ε_r is the radial strain and B is a pressure. The expression for the total work required to strain the fiber due only to the coupling of extension and radial compression, as a function of extensional strain, ε_{ext}, is then

$$\Delta W_B = \int_0^L \int_{R_c}^R (\mu(\varepsilon_r) 2\pi r dr) dl. \tag{6.12}$$

After integrating, we then express the work done due to the compression modulus as

$$\Delta W_B = (R^2 - R_c^2)\pi L \mu(\varepsilon_r) \tag{6.13}$$

where R_c is the radius of the fluid core. We know this radius – typically on the order of 100 nm – is negligible when considering the entire volume of the fiber so that $R_c^2 = 0$ and

$$\Delta W_B = V\mu(\varepsilon_r) \tag{6.14}$$

where V is the constant volume of the fiber. Then, replacing ε_r with ε_{ext} in $\mu(\varepsilon_r)$ (*Equation 6.11*) via the relation in *Equation 6.5*, we arrive at

$$\Delta W_B = \frac{V}{8} B\varepsilon_{ext}^2. \tag{6.15}$$

Energy Cost from Polarization Splay

The third and final contribution to the energy cost of extension comes from increasing the splay of the polarization during strain. This ferroelectric phase prefers the dipole vector of each neighboring molecule in a specific smectic layer to be in the plane of the layer and in the same direction. The dipole vector, or dipole density is then forced to exhibit splay leading to a periodic density discontinuity which manifest as evenly spaced, periodic defect lines. These defect lines can, theoretically, either be entirely separated in the form of rings with the fiber's long axis through their center and normal to their plane, or as a continuously winding helical defect line with a helical axis parallel and collinear with the fiber axis as imagined in *Figure 6.2*. The distance between defects, and also the length scale of the polarization splay is defined as b while the helical angle is α and the periodic length, or pitch of the helix is given by l. As the fiber is initially formed, the angle α is spontaneously formed and remains constant during fiber formation. The strain necessary to make the fiber in the first place is also responsible for the polarization splay, since this is a process of symmetry breaking. *Figure 6.3* illustrates the splay component.

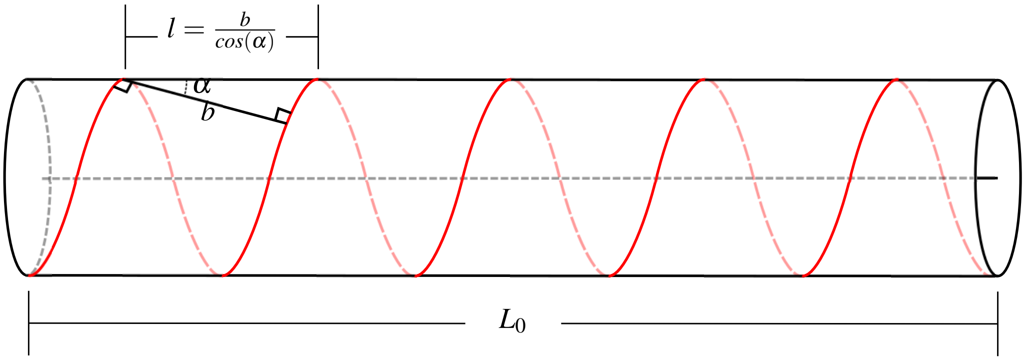


Figure 6.2: Schematic of the geometry of the helical defect line winding along a fiber.

The strain of b is then directly coupled to the increase in splay of the dipoles. Again, although there may be an equilibrium degree of splay associated with the initial symmetry breaking and energy minimization for the phase to reach the stable fiber state, and change in the length b will result in a net energy cost. The strain of b is then the same as the strain in length (ε_{ext}) since the percent change in the total length is the same as the percent change of each segment such that

$$\varepsilon_{ext} = \frac{\Delta L}{L_0} = \frac{\Delta l}{l_0} = \frac{(l_0 + \delta l) - l_0}{l_0} = \frac{\left(\frac{b_0+\delta b}{cos(\alpha)}\right) - \left(\frac{b_0}{cos(\alpha)}\right)}{\frac{b_0}{cos(\alpha)}} = \frac{\Delta b}{b_0} \tag{6.16}$$

The distortion energy associated with splay is given by the Frank distortion energy term

$$F_{splay} = \frac{1}{2} K_s (\nabla \cdot \hat{n})^2 \tag{6.17}$$

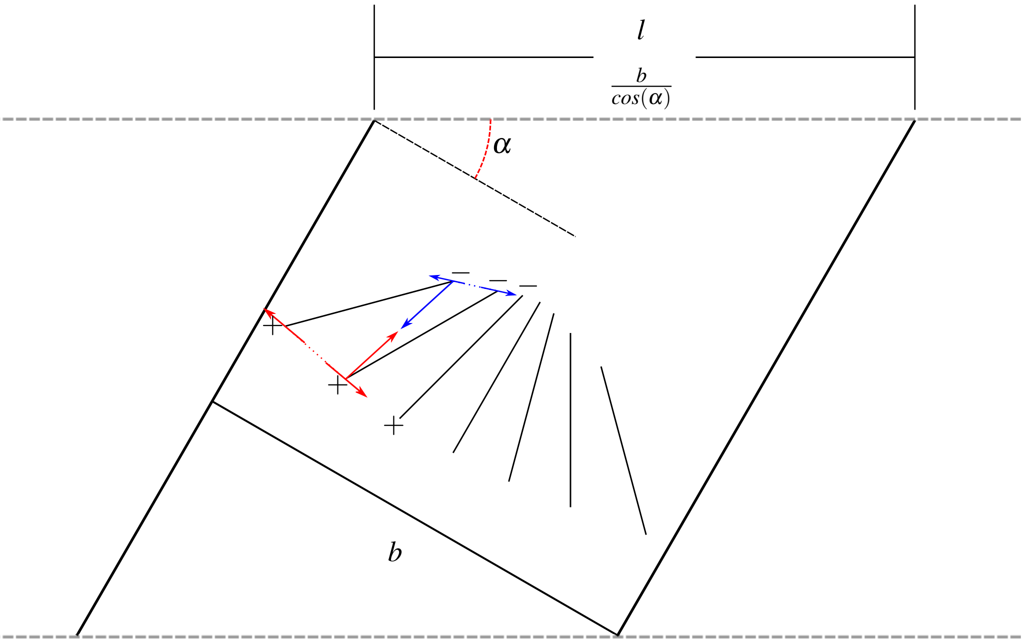


Figure 6.3: The geometry of the polarization splay between defect lines

where the divergence term, in terms of a small change in in b is then the infinitesimal change in splay curvature

$$\left(\frac{\pi}{b+\delta b}-\frac{\pi}{b}\right)^2. \tag{6.18}$$

For the work required to change the polarization splay for a single smectic stripe[2], the distortion free energy is integrated over the length of such a stripe such that

$$\Delta W_{p1}=\frac{K_s}{2}\int_{-l/2}^{l/2}\left(\frac{\pi}{b+\delta b}-\frac{\pi}{b}\right)^2 dx \approx \frac{\pi^2 K_s}{2}\left(\frac{\delta b}{b}\cdot\frac{1}{b}\right)^2\cdot\frac{b}{cos(\alpha)} \tag{6.19}$$

where the approximation comes from

$$\left(\frac{\pi}{b+\delta b}-\frac{\pi}{b}\right)^2=\frac{\pi^2(\delta b)^2}{b^2(b+\delta b)^2}\approx\frac{\pi^2(\delta b)^2}{b^4} \tag{6.20}$$

Then, the curvature term π/b can be rewritten as s_0 and the expression for the work to strain a smectic stripe due to polarization splay expressed as

$$\Delta W_{p1}=\frac{K_s}{2}\left(\frac{\delta b}{b}\right)^2 s_0^2\cdot\frac{b}{cos(\alpha)}. \tag{6.21}$$

where ΔW_{p1} is the energy needed to increase the segment length $\left(\frac{b}{cos(\alpha)}\right)$ due to the change in the segment separation length (b) which is coupled to the elastic deformation energy cost of polarization splay. Then, integrating across all layers through the fiber radius, we arrive at the energy to deform

[2] *Stripe* here refers to a section of the fiber between defect lines

one single defect stripe with a diameter equal to that of a fiber, ΔW_{stripe},

$$\Delta W_{stripe} = \Delta W_{p1} \int_{R_c}^{R} 2\pi r dr = \frac{K_s}{2} \left(\frac{\delta b}{b}\right)^2 s_0^2 \cdot \frac{b}{cos(\alpha)} (R^2 - R_c^2) 2\pi \approx K_s \left(\frac{\delta b}{b}\right)^2 s_0^2 \cdot \frac{b}{cos(\alpha)} R^2 \pi \tag{6.22}$$

Then considering that each stripe has axial length b, the number of stripes n is then L/b so that for the entire fiber the expression becomes

$$\Delta W_p = \Delta W_{stripe} n = \Delta W_{stripe} \frac{L}{b} cos(\alpha) = K_s s_o^2 V \varepsilon^2 \tag{6.23}$$

which shows that there is no α dependence. In other words, the helical geometry and its variation from fiber to fiber doesn't affect the linear elastic behavior. It may, however, result in changes in initial static stress and tension.

It should be noted that there should be a critical amount of strain associated with reaching a maximum, critical energy stored in the splay such that, considering the conventions in *Figure 6.3*, when the $(-)$ ends of the dipoles reach some critical proximity, (if we assume the dipole is rotating around its center) since the $(-)$ end is moving through an arc. If the point of minimum approach is exceeded, then the $(-)$ ends of the dipoles will be repelled from each other while the opposite charged ends will be brought together in a way that will increase the splay infinitely, to the point where all the elastic energy stored in the splay will be dissipated and the length b won't be bound anymore, which could be a cause of the strain-dependent and strain-rate dependent critical failure of the fiber's stability if the fiber reaches its yield strain/stress too quickly, strain hardens and fails.

Final Elastic Expression

Each of the three energy contributions can be summed into an expression for the total deformation energy as initially suggested in *Equation 6.1* and shown below for convenience and then written in terms of the energy contributions in *Equation 6.25*. Since both a ε and L appear in all three energy terms, dividing them out of the entire expression transforms it into an expression for force which is analogous to a tensile force, or tension as shown in *Equation 7.1*.

$$\Delta W(\varepsilon) = \Delta W_T + \Delta W_B + \Delta W_p \tag{6.24}$$

$$\Delta W_{tot} = \gamma 2\pi R L_0 \varepsilon + \frac{V}{8} B \varepsilon_{ext}^2 + K_s s_o^2 V \varepsilon^2 \tag{6.25}$$

$$T_{eff} = \gamma 2\pi R + R^2 \pi \varepsilon \left(\frac{B}{8} + K_s s_0^2\right) \tag{6.26}$$

The first term is the linear contribution to the measured force or stress of the fiber while the second term is quadratic in R and corresponds bulk elastic contribution. This expression may be slightly expanded to include the effects of the reduced radius during extension.

Since the force acting along the fiber due to surface tension depends on the radius, and the radius is compressed with strain, one needs to expect that the surface contribution dies slowly. This is negligible for small strains but would show up in larger ones. The bulk term would also drop off slowly with strain once a critical balance between the quadratic growth is reached. Again, from $\varepsilon_{ext} = -2\varepsilon_r$,

$$R(\varepsilon) = R_0\left(1 - \frac{\varepsilon_{ext}}{2}\right). \tag{6.27}$$

For the bead-hanging experiments that I did, the diameter of the fiber was measured after the bead was hung. Sticking with the conservation of volume assumption, we can then back-calculate what R_0 was before the deformation using $Equation$ 6.27 so that

$$R_0 = \frac{2R_f}{2 - \varepsilon_{final}} \tag{6.28}$$

where R_f is the final (measured) radius at the final (equilibrium) strain. Then, the tension terms can be expressed as

$$T_{eff} = \gamma 2\pi R_0\left(1 - \frac{\varepsilon_{ext}}{2}\right) + \pi\left(R_0\left(1 - \frac{\varepsilon_{ext}}{2}\right)\right)^2 \varepsilon_{ext}(BulkTerm). \tag{6.29}$$

6.2 Linear Viscoelastic Behavior

While viscoelastic behavior can be defined as a material's response to applied stresses resulting from both storage and dissipation of energy, linear viscoelastic behavior narrows this down to behavior that is the linear combination of the effects of linear storage and linear dissipation. Linear storage refers to linear elastic behavior also known as Hookean behavior. Linear dissipation behavior refers to linear viscous behavior also known as Newtonian behavior. A linear viscoelastic response combines Hookean and Newtonian behavior. When these effects are combined in series, the material is said to behave according to the *Maxwell model*[3]. When these effects are combined in parallel, the material is said to behave according to the *Kelvin-Voigt* model[4].

In order to measure this behavior, the material needed to be deformed in a way that was expected to behave linearly in both elastic and viscous responses. This was achieved by ensuring that the magnitude of the strain remained as small as possible. Two types of applied deformation we chosen: step strain and oscillatory. Step strain aimed to impart a sudden and constant strain on the fiber while observing relaxation while small amplitude oscillation (SAOS) aimed to extract the out of phase behavior of the material due to its dissipative nature.

[3] A Maxwell material is a viscoelastic material having the properties both of elasticity and viscosity. It is named for James Clerk Maxwell who proposed the model in 1867. It is also known as a Maxwell fluid.

[4] A Kelvin-Voigt material, also called a Voigt material, is a viscoelastic material having the properties both of elasticity and viscosity. It is named after the British physicist and engineer Lord Kelvin and after German physicist Woldemar Voigt.

First, the SAOS results presented in *Section 5.2* are discussed. From the Lissajous plot example *Figure 5.5*, two trends are clear: the phase angle decreases with increased frequency while the storage modulus increases. For a Newtonian fluid, the phase angle $\delta = 90\circ$. This is due to the stress response of the material being 90 degrees out-of-phase with the excitation as no energy is stored, only dissipated. In a Lissajous plot, this effect is easily spotted when the stress versus strain ellipse approaches a circle. For purely elastic materials, the phase angle, or response lag, is zero: $\delta = 0$. This is seen on Lissajous plots as an ellipse which a minor axis approaching zero: a straight line. A straight line on a stress versus strain plot is consequently interpreted as a linear elastic response. Since viscoelastic materials lie between purely viscous and purely elastic, the phase angle δ shows behavior spanning the intermediate regime as a function of driving frequency. The slope of the long axis of the ellipse is defined as the storage modulus E'. When discussion elastic material, this is equivalent to the Young's modulus, G, which is the slope of the linear stress versus strain relation. When discussing viscoelastic materials, Young's modulus is left in favor of storage and loss moduli. These moduli are defined as

$$E' = \frac{\sigma_0}{\varepsilon_0} cos(\delta) \tag{6.30}$$

$$E'' = \frac{\sigma_0}{\varepsilon_0} sin(\delta) \tag{6.31}$$

and

$$E^* = E' + iE'' \tag{6.32}$$

where σ_0 is the stress amplitude, ε_0 is the strain amplitude, E' is the storage modulus, E'' is the loss modulus and E^* is the complex modulus. The total elastic modulus is often defined as the magnitude of E^*. The phase angle δ is generally used to calculate the value known as the loss tangent, which is written as

$$tan(\delta) = \frac{E''}{E'} = \frac{sin(\delta)}{cos(\delta)} \tag{6.33}$$

and is used as a measure of how liquid-like or solid-like a material behaviors under oscillatory driving. For a solid, the loss tangent approaches 0 and the storage modulus and complex modulus both approach the Young's modulus. When a material makes the transition frombehaving like a fluid at low driving frequencies to behaving like a solid at higher frequencies, it is considered to be rate hardening. It is clear from *Figure 5.5* that the elastic modulus increases significantly and monotonically with frequency, accompanied by a loss tangent approaching zero such that for high frequencies, the material behaves effectively as a linear elastic material. This behavior is shown in *Figure 6.4* for a BLC fiber held at 150 °C. An exponential decay of the loss tangent is observed, while the storage modulus is seen to increase, with frequency. The loss modulus tends to remain constant. This can be interpreted as a dissipative components which is not frequency dependent. Furthermore, the viscous terms are defined as

$$\eta' = \frac{E''}{\omega} \tag{6.34}$$

$$\eta'' = \frac{E'}{\omega} \tag{6.35}$$

and

$$\eta^* = \eta' + i\eta'' \tag{6.36}$$

where η^* is the *complex viscosity*, η' the *dynamic viscosity* or the real part and η'' is the *out-of-phase viscosity* or the imaginary part. These are related to the standard *coefficient of viscosity* η in that a material exhibiting purely Newtonian behavior approaches a storage modulus $E'' = 0$. *Figure 6.4* shows that the real component of the complex viscosity – related to the fluid-like behavior – dies off more quickly than the imaginary part of the complex viscosity – related to the solid-like behavior.

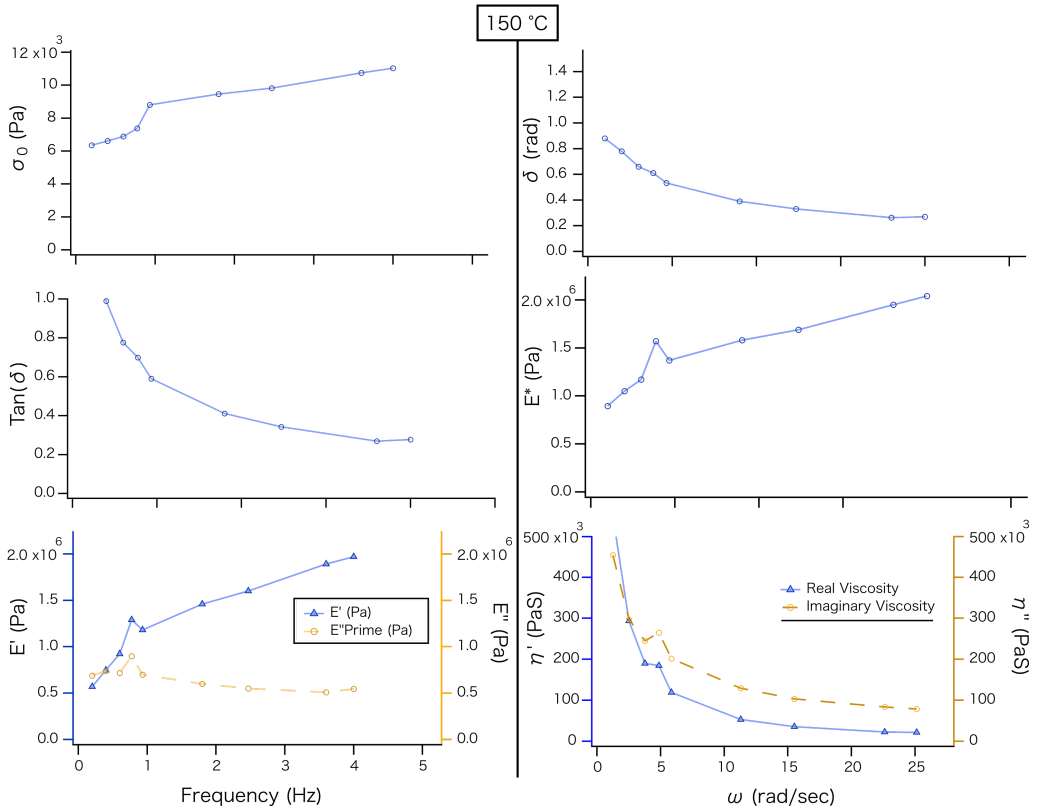


Figure 6.4: Measured and calculated parameters versus frequency. In order from left to right, top to bottom: Stress magnitude (σ_0); phase angle (δ); loss coefficient ($\tan\delta$); magnitude of the complex modulus ($|E^*|$), storage and loss moduli (E' and E"); real and imaginary components of the complex viscosity ($\eta' and \eta''$).

Since the temperature dependent behavior is also of interest, the SAOS experiments were conducted at temperatures spanning the liquid-crystalline phase of the mesogen used. The material typically entered a crystalline transition at just under 120 °C and melted into the isotropic phase at 170 °C. In order to visualize the temperature dependence and frequency dependence most clearly, a Pipkin diagram is shown in *Figure 6.5*. In this figure, a clear time-temperature relation can be observed: The material behaves similarly at low temperatures and low driving frequencies as it does for high temperatures and high driving frequencies. The material , across the frequency range, behaves more like an elastic solid as the temperature decreases. Likewise, the material behaves more like a solid across the temperature range for increasing frequency. In addition, the dissipated energy for a single extension-compression cycle is defined by the area of the strain-strain ellipse which is expressed as

$$-\Delta U = \sigma_0 \varepsilon_0 sin(\delta) \tag{6.37}$$

where $-\Delta U$ is the dissipated energy per cycle in units of *Joules*. *Figure 6.5* clearly shows that the BLC fibers are able to store more kinetic energy (lose less) per cycle when either the temperature is decreased or the frequency increased. It was also observed that the behavior suddenly change for temperatures nearing the crystalline phase. This has not yet been properly analyzed, but it can be seen that near this transition the behavior reverses: the fiber behaves more like a fluid as the frequency increases. Its viscous stresses increase with frequency just as before, as sen by the increase in elastic modulus, but the loss tangent and energy dissipation increase as well. meaning it becomes more fluid-like. Further experiments and analysis into this behavior and its implications of a possible mesophase between B7 crystalline should be followed up.

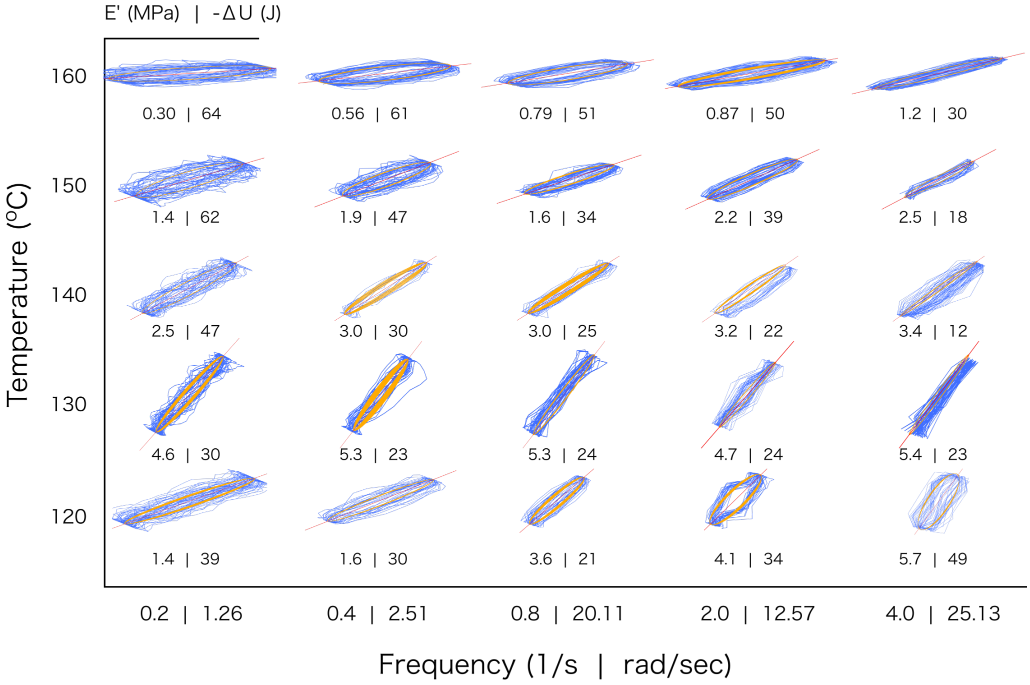


Figure 6.5: A Pipkin diagram summarizing the frequency- and temperature-dependent behavior of BLC fibers. The left axis increases vertically with temperature, while the bottom axis increases to the right with frequency. Each Lissajous plot is labelled with its storage modulus E' and dissipated energy per cycle, ΔU.

6.3 Nonlinear Viscoelastic Behavior

While the step strain and SAOS experiments focused on isolating the linear viscoelastic response of BLC fibers, applying a constant strain rate extension over extremely large strains shows the transition between initial linear behavior and transient flow behavior separated by a critical point. Considering the experimental conclusions presented in the *Results* chapter in *Section 5.1*, three major regimes are discussed. These regimes are both similar and in contrast with those found while straining polymer solutions under the same conditions. In polymers, the initial elastic regime is due to the tangled chains being stretched until they are fully extended. After this point, the extended chains are forced to slide past each other as the fluid column is axially strains and radially compressed resulting in strain hardening. For many polymer systems, the rate hardening under extension is in contrast to rate, or shear thinning observed in shear rheometry experiments.

As shown in *Figure 5.1* in which a BLC fiber held at 150 °C and strained at a constant rate of $\dot{\varepsilon} = 5$, the radius decreases exponentially, indicating the possibility that the fiber is, in fact, conserving volume and compression in accordance with a Laplace ratio of 0.5 in accordance with elastic behavior. To analyze this further, the expression for the fiber radius as a function of time $R(t)$ can be given as a function of extensional strain. *Equation 1.23* is given again here for convenience:

$$L(t) = L_0 e^{\dot{\varepsilon} t} \tag{6.38}$$

and can be re-written in terms of $\varepsilon(t)$ such that

$$L(t) = e^{\dot{\varepsilon} t} - 1 \tag{6.39}$$

From the relation between extensional and radial strains,

$$\varepsilon_r = -\frac{e^{\dot{\varepsilon} t} - 1}{2} \tag{6.40}$$

from which the expression

$$R(t) = \frac{R_0(3 - e^{\dot{\varepsilon} t})}{2} \tag{6.41}$$

can be simply derived. Using the values for $\dot{\varepsilon}$ and R_0 from the experiment, a curve is plotted and shown against the data in *Figure 5.1*.

In *Figure 6.6*, three clearly defined regimes are considered: elastic, yield and flow. Before the elastic regime, the behavior can be said to be *strongly elastic* as the theoretical and experimental curves of the evolution of the radius in time match very closely. Then, the fiber enters the large regime labeled *Elastic Regime* where the expected and radius begin to deviate slightly while the stress versus strain behavior still show a linear relation This regime can also be considered to be *weakly elastic*.

At this point, the radius starts to shrink much more rapidly than expected for a simple elastic response: the fiber has reached its elastic limit and thinning has begun. It is important to consider than during extension, the smectic layers are compressed against the fluid defect core at the center of the fiber. The energy required to compress the inner-most layers is accompanied by the critical curvature. At some critical radial compression, the inner most layer must then *collapse* into the core via a smectic to isotropic – or perhaps other more-highly-disorder bent-core phase – initializing the rapid radial thinning accompanied by flow away from the fiber core and into the bulk material at the menisci. Then, since this transition has been initiated, the fiber is able to attempt to restore itself. The remaining smectic layers at the outside of the fiber are highly curved and would like to reduce this curvature creating a negative pressure inside the fiber. The same threshold that allowed material to enter the core and escape into the menisci now allows the process to run in reverse: material is pulled from the menisci and layers are rebuilt from the inner core outward resulting in a sudden increase in fiber thickness at the yield peak. The width of the yield regime can then be interpreted as the rate at which this transition occurs.

Once the yield regime has passed, post-yield behavior (outright flow) takes over. Here, the fiber cannot keep up its attempt to continuously fill the fiber with more material to preserve its lowest-energy equilibrium diameter. The fiber length is rapidly increasing and the cross-section of the defect core at each termini is not changing. the volume of the core is shrinking more rapidly than the material can be forced to flow inwards. This is, in effect, the same behavior seen in the first half of the yield regime. This interpretation is supported by the observation that the slope of the radial compression in time during the first half of the yield regime is nearly identical to the slope of radial compression in time during the majority of the flow regime. This time, however, there is no restoring response as is seen in the second half of the yield regime: the fiber is forced to continue to compress past the radial minimum seen during yield. At this point, the fiber has passed the critical point in which it can heal itself and, even after pulling has stopped and any stabilizing effects of strain or rate hardening have ceased, the fiber spontaneously ruptures.

The effect of the strain rate itself can be seen clearly in *Figures 5.2* and *5.3*. The elastic modulus calculated from the slope of the elastic regime for different strain rates follows a clear trend of rate hardening and is shown in *Figure 5.4*, presented again here for convenience as *Figure 6.7*. During the shear rheological study by Bailey *et al.*, they found that rate thinning was the dominant response to increased shear rates. Here, however, rate thickening is apparent. This discrepancy is also apparent in polymer systems as reported by McKinley *et al.* where polymers under shear tend to thin while those under extension tend to thicken.

The flow behavior observed in the present experiments is, however, less straightforward to interpret. The rate dependence is not clear: the expression for calculating the extensional viscosity – the stress magnitude during the stable, transient regime after the yield point – differs wildly for different strain rate. This may be due to the many different initial fiber lengths encountered during the constant strain rate experiments discussed here. While the initial length was held constant during

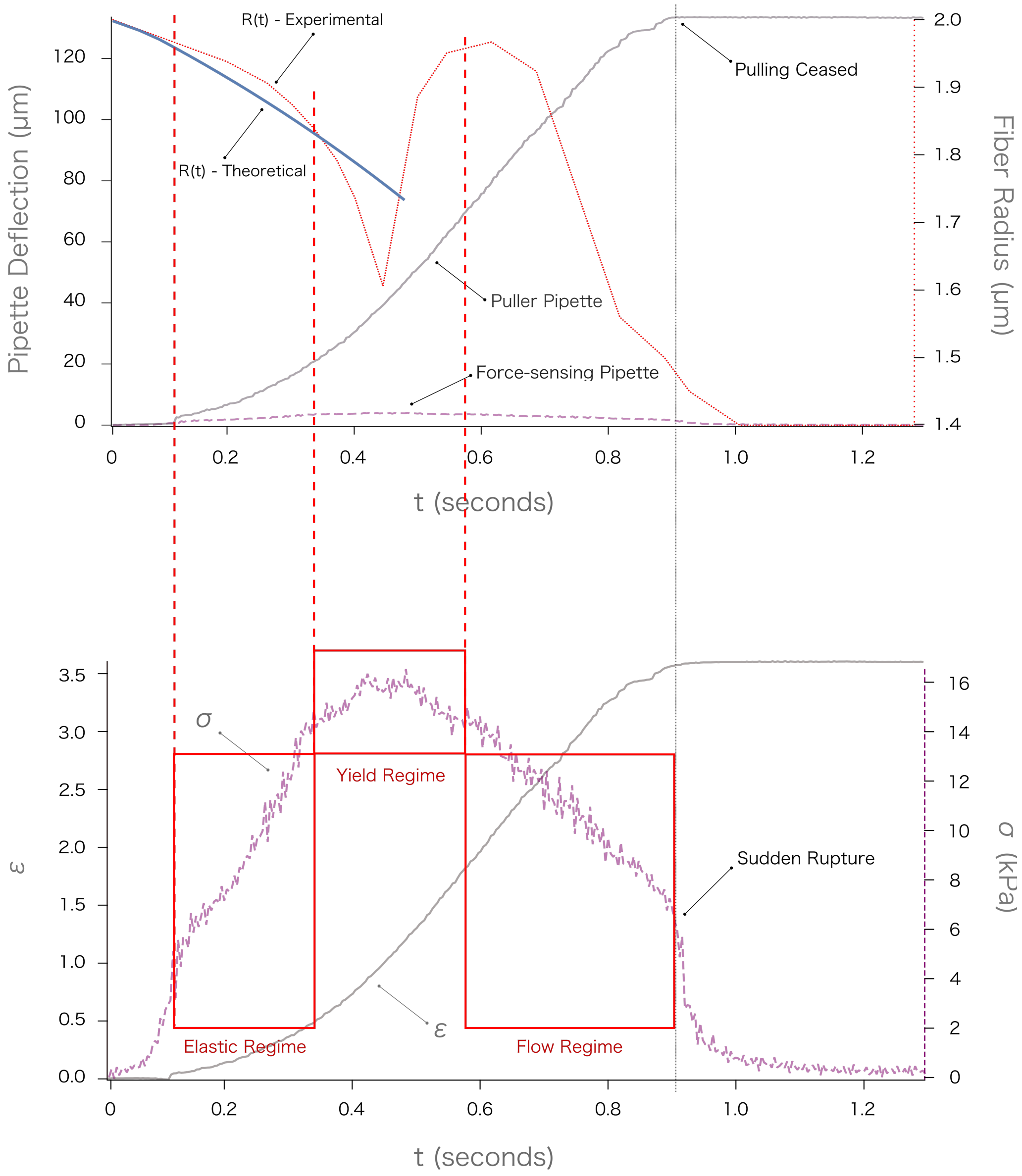


Figure 6.6: Top: The theoretical curve for the decreasing radius as a function of time, $R(t)$, is superimposed against the experimental measurements. The different regimes are then connected to the stress versus strain plot (**bottom**) in which three clearly defined regimes are emphasized.

most SAOS experiments, it tended to vary from 30 to 100 μm during constant rate stretching[5]. However, this does shed some light on the possible dependence of the initial fiber length.

Since it is proposed that flow regime is initiated due to a limited cross-sectional area for new material to flow into the fiber compared to the total core volume, the initial volume of the core may play a role. This would mean that for longer fibers, the slope of the transient flow regime should be steeper while for initially shorter fibers, it should be more shallow. This hypothesis is supported by the behavior observed in *Figure 6.8*. The two curves of interest are those with strain rates of 5.0

[5] These experiments were done early during the research period when the methodology had not yet been consistent

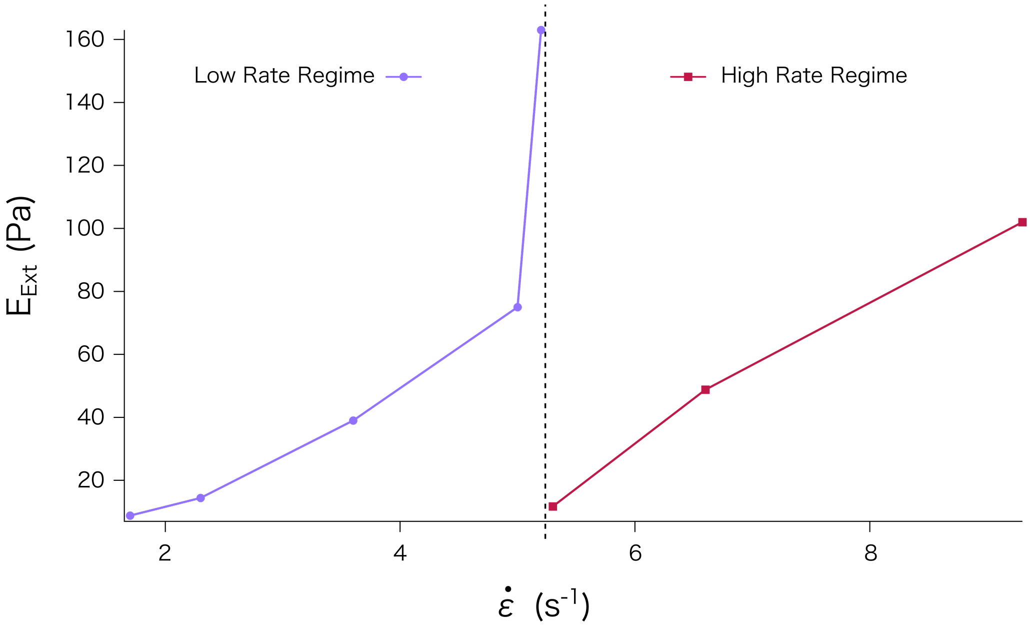


Figure 6.7: Elastic modulus of extension (E_{Ext}) versus strain rate. Both the low rate regime and high rate regime show clear rate hardening trends. The dotted vertical line separates what seems like two different material regimes – one for low rates and another for high rates of constant extension.

and 5.2. Both fibers were pulled at similar rates and both fibers had an initial fiber radius of about 2.1 μm. The only difference was their initial lengths: The fiber pulled at $\dot{\varepsilon} = 5.0$ had an initial length of $L_0 = 35\mu m$ while the fiber pulled at $\dot{\varepsilon} = 5.2$ had an initial length of $L_0 = 95\mu m$. Both fibers reached a critical stress of 16 kPa before yielding. Post-yield, however, the initially longer fiber fails much more quickly than the initially shorter fiber.

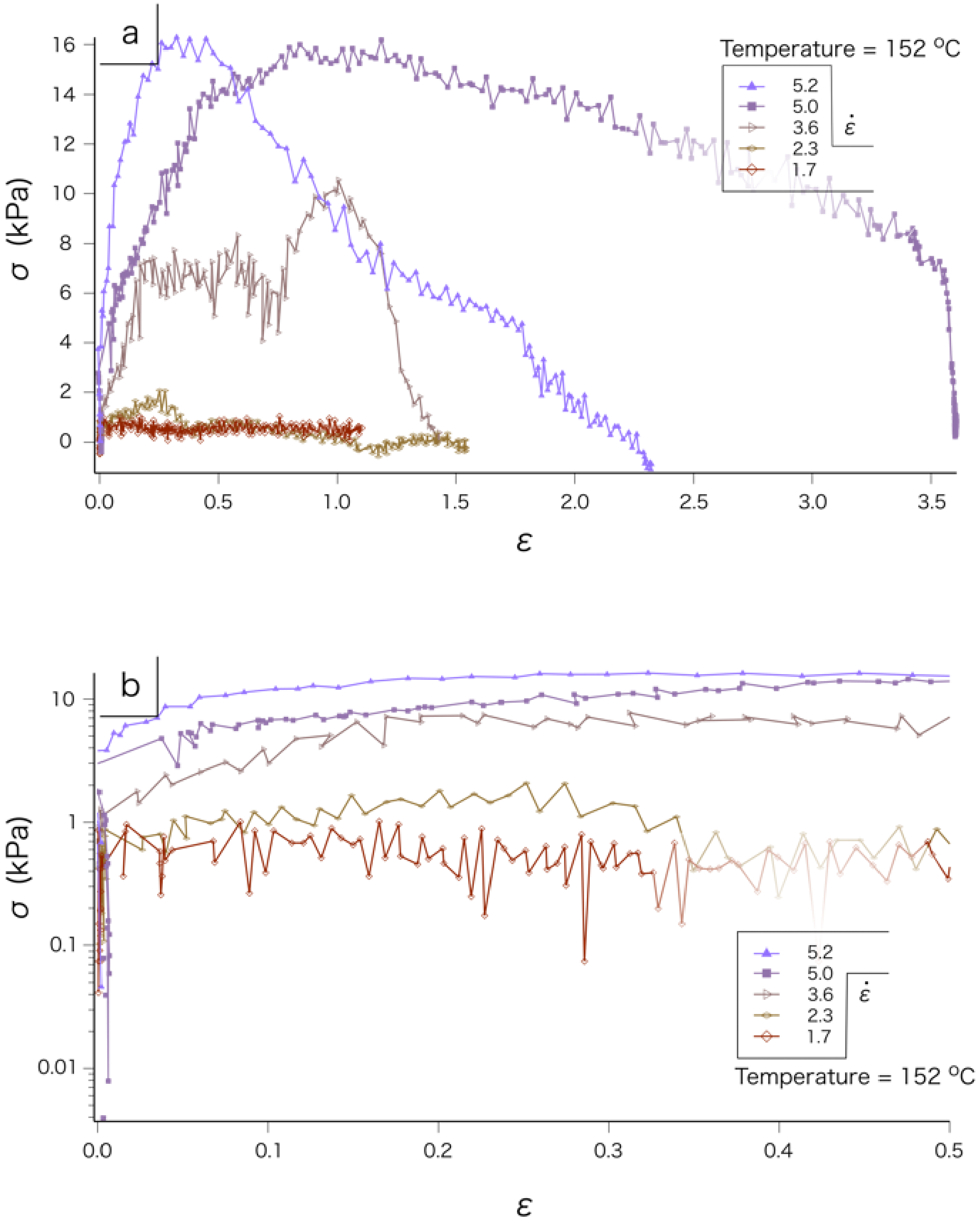


Figure 6.8: (a) Stress versus strain curves for BLC fibers under constant rate stretching. While the two for the two highest rates reach a yield point at the same critical stress, the initially longer fiber (pulled at a rate of 5.2) fails more quickly than the initially shorter fiber (pulled at a rate of 5.0). **(b)** The same data presented on a log-linear plot with the strain axis cropped to $\varepsilon = 0.5$ to emphasize the initial elastic-like regime. Rate dependence on the elastic modulus of extension is clearly seen.

CHAPTER VII

CONCLUSIONS

The continued study of BLCs shows that there is much to be learned and great insight to be found. An improved understanding of such a complex material and phase will yield a fundamental understanding of spontaneous symmetry breaking in geometrically confined systems bound by surface effects and the internal structure of layered systems and reduced-dimension fluids. Further understanding of the mechanics of polar, structured fluids could reveal how rotational and dipole moments play a role in the viscoelastic effects of such systems, while the potential to manipulate internal stresses to control macroscopic geometries may lead to the practical use of BLCs or other similar materials as artificial muscles. In short, developing an understanding of how the molecular structure of a material directly affects its large scale macroscopic behavior is a complex problem and its solution offers benefits.

7.1 Linear Response

From the elastic expression derived in *Section 6.1* and shown here for convenience,

$$T_{eff} = \gamma 2\pi R + R^2 \pi \varepsilon \left(\frac{B}{8} + K_s s_0^2 \right) \tag{7.1}$$

an effective time-independent stress-strain relation can be extracted. Dividing the force (T_{eff}) by the cross-sectional area πR^2 results in the expression

$$\sigma(\varepsilon) = \frac{2\gamma}{R_0} + \varepsilon \left(\frac{B}{8} + K_s s_0^2 \right) \tag{7.2}$$

or more compactly

$$\sigma(\varepsilon) = \sigma_\gamma + \varepsilon E \tag{7.3}$$

where σ_γ is the strain-independent surface stress and E is the effective elastic modulus. *Equation 7.3* suggests that a bent-core liquid crystal fiber, at equilibrium under zero strain conditions, contains only stress due to the surface. Then, upon straining the fiber, the bulk contributions begin to grow linearly with the strain at a rate dictated by E. Comparing this with the bead-hanging experiments to measure the static equilibrium tension such as those presented by Morys *et al.*, the expression relating the deflection angle α to the strain ε is once again

$$\varepsilon = sec(\alpha) - 1 \tag{7.4}$$

which, combined with *Equation 7.3*, gives

$$\sigma(\alpha) = \sigma_\gamma + (sec(\alpha) - 1)E \tag{7.5}$$

The elastic modulus E can then be estimated by using a reasonable value for the splay elastic constant K_s = 1 pN. The compression modulus B was derived earlier to be roughly 110 kPa. s_0 is calculated as π divided by the length of a defect stripe along the fiber axis, b, or $s_0 = 3.5x10^8 m$ which represents the radius of curvature of the splay. From this, we arrive at a value of roughly $E = 140$ kPa showing that while most of the elastic modulus comes from layer compression, there is a non-negligible component due to polarization splay. Without strain data from the bead-hanging experiments, it is hard to conclude that there is a strain-dependent behavior. Therefore, the experiments in this thesis are required. Furthermore, the analysis of the fiber behavior upon hanging a bead is complex: Initially, the fiber experiences an enormous stress as the bead is applied which likely induces a transition into the yield regime immediately before settling to its maximum deflection angle as dictated by surface tension. Then, the fiber's internal structure restores as the angle decreases until equilibrium is reached. It is doubtful that the final strains measured by this approach can be considered comparable to a linear theory since the true initial length before finite strain is achieved, L_0 is unknown.

Considering the value of E estimated above and comparing to the data in *Analysis* for constant rate extension, the value is within reason even for lower rates where it would be expected. When comparing to the results from the SAOS experiments, it can be seen that the lower bound of the storage modulus approaches the value of E estimated here for high temperature and low frequency. The next step would then be to further understand the time dependent nature of the viscoelastic behavior by rewriting *Equation 7.3* as

$$\sigma(\dot{\varepsilon}) = \sigma_\gamma + \varepsilon E(\dot{\varepsilon}) \tag{7.6}$$

where the rate dependence comes from the time dependent nature of the compression modulus and the polarization splay deformation. If these two time scales are known, as well as the temperature dependence of those time scales, then a fully time-dependent viscoelastic constitutive equation can be derived. Extracting these timescales is part of ongoing analysis and will require the further results from step-strain experiments.

7.2 Rheological Conclusions

The results from constant strain rate extension, step strain and oscillatory driving show that fibers formed by bent-core mesogens in the smectic B7 state behave viscoelastically and exhibit a strong linear elastic regime for strain below 5%. Rate hardening is apparent in both constant rate extension and SAOS experiments while temperature dependence is strongly observable. Relaxations times can be attributed to the rotational moment, the dipole moment and the layer compression modulus of the fibers and further extraction of these time scales from the further analysis and comparison of step strain and oscillatory measurements will define a viscoelastic constitutive equation. An expression of this nature will be a powerful tool to further develop the practical use of such fibers; the molecular characteristics that are known to affect compression modulus and polarization splay such as the magnitude of the molecules dipole moment and length of its end chains can be used to fine-tune the material's macroscopic behavior. Furthermore, electric field coupling to manipulate the polarization geometry may be used as an actuation stimulus.

The temperature and rate dependent results show that the storage modulus can be tuned and that, if a fiber is desired as an artificial muscle and its length can be contract by an external stimulus, then adjustment of the temperature and a low-amplitude driving frequency of the optimal rate can result in artificial muscles with a wide range of solid-like or liquid-like behaviors. While solid like behavior is clearly desired for practical use corresponding to colder temperatures and higher driving frequencies, should a fiber be ready to rupture or undergo yield failure, changes in temperature and frequency can be made to induce self-healing. Furthermore, if a fixed length is desired in a solid-like phase and it should suddenly benefit for the actuator to have an increased length, a liquid-like phase can be entered, the fiber pulled, and the solid-like phase re-entered.

This is the first study that attempts to uncover the time dependent nature of liquid crystal fibers formed by bent-core mesogens using a direct method and known rheological techniques. Performing these experiments on more compounds, over wider temperature and frequency rates with a fully capable constitutive equation will not only help progress this material towards its potential as a practical soft actuator but will also help to uncover material properties which may one day lead to a generalized molecular theory of viscoelasticity.

Bibliography

[1] D. H. Van Winkle and N. A. Clark, “Freely suspended strands of tilted columnar liquid-crystal phases: One-dimensional nematics with orientational jumps,” *Physical Review Letters*, vol. 48, no. 20, pp. 1407–1410, 1982.

[2] A. Jákli, D. Krüerke, and G. G. Nair, “Liquid crystal fibers of bent-core molecules.,” *Physical review. E, Statistical, nonlinear, and soft matter physics*, vol. 67, no. 5 Pt 1, p. 051702, 2003.

[3] C. Bailey, E. C. Gartland, and A. Jákli, “Structure and stability of bent core liquid crystal fibers,” *Physical Review E*, vol. 75, p. 031701, mar 2007.

[4] M. Morys, T. Trittel, a. Eremin, P. Murphy, and R. Stannarius, “Tension of freely suspended fluid filaments,” *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics*, vol. 86, pp. 1–5, 2012.

[5] S. M. Salili, T. Ostapenko, O. Kress, C. Bailey, W. Weissflog, K. Harth, A. Eremin, R. Stannarius, and A. Jákli, “Rupture and recoil of bent-core liquid crystal filaments,” *Soft Matter*, vol. 12, no. 21, pp. 4725–4730, 2016.

[6] S. J. Vainker, *Chinese silk : a cultural history*. London : British Museum Press ; New Brunswick, N.J. : in association with Rutgers University Press, 1 ed., 2004.

[7] A. M. A. M. Collier, *A handbook of textiles*. Oxford ; New York : Pergamon Press, [1st ed.] ed., 1970.

[8] M. C. Corporation and M. C. C. Staff, *How It Works: Science and Technology*. How We Lived Series, Marshall Cavendish Corporation, 2003.

[9] J. A. Reglero Ruiz, M. Trigo-López, F. C. García, and J. M. García, “Functional aromatic polyamides,” *Polymers*, vol. 9, no. 9, 2017.

[10] L. Rayleigh, “On the instability of jets,” *Proc London Math Soc.*, vol. 10, pp. 4–13, 1879.

[11] R. Shuttleworth, “The surface tension of solids,” *Proceedings of the Physical Society. Section A*, vol. 63, no. 5, p. 444, 1950.

[12] K. Angelova, “In pictures: How robots have evolved over the last 200 years,” Jul 2011.

[13] A. H. Gelebart, D. Jan Mulder, M. Varga, A. Konya, G. Vantomme, E. W. Meijer, R. L. Selinger, and D. J. Broer, "Making waves in a photoactive polymer film," *Nature*, vol. 546, no. 7660, pp. 632–636, 2017.

[14] R. F. Shepherd, F. Ilievski, W. Choi, S. A. Morin, A. A. Stokes, A. D. Mazzeo, X. Chen, M. Wang, and G. M. Whitesides, "Multigait soft robot," *Proceedings of the National Academy of Sciences*, vol. 108, no. 51, pp. 20400–20403, 2011.

[15] N. W. Bartlett, M. T. Tolley, J. T. B. Overvelde, J. C. Weaver, B. Mosadegh, K. Bertoldi, G. M. Whitesides, and R. J. Wood, "A 3d-printed, functionally graded soft robot powered by combustion," *Science*, vol. 349, no. 6244, pp. 161–165, 2015.

[16] A. Abe, *Liquid Crystal Elastomers: Materials and Applications*, vol. 250 of *Advances in Polymer Science*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012.

[17] P. M. Hogan, A. R. Tajbakhsh, and E. M. Terentjev, "uv manipulation of order and macroscopic shape in nematic elastomers," *Phys. Rev. E*, vol. 65, p. 041720, Apr 2002.

[18] M. Camacho-Lopez, H. Finkelmann, P. Palffy-Muhoray, and M. Shelley, "Fast liquid-crystal elastomer swims into the dark," *Nature Materials*, vol. 3, no. 5, pp. 307–310, 2004.

[19] Y. Cohen, *Electroactive polymer (EAP) actuators as artificial muscles : reality, potential, and challenges*. Bellingham, Wash: SPIE Press, 2004.

[20] A. Jákli, D. Krüerke, and G. Nair, "Liquid crystal fibers of bent-core molecules," *Physical Review E*, vol. 67, p. 051702, may 2003.

[21] A. Nemeş, A. Eremin, and R. Stannarius, "Mechanical Properties of Freely Suspended LC Filaments," *Molecular Crystals and Liquid Crystals*, vol. 449, pp. 179–189, jul 2006.

[22] A. Eremin, A. Nemes, R. Stannarius, M. Schulz, H. Nádasi, and W. Weissflog, "Structure and mechanical properties of liquid crystalline filaments.," *Phys Rev E Stat Nonlin Soft Matter Phys*, vol. 71, no. 3 Pt 1, p. 31705, 2005.

[23] C. A. Bailey, M. Murphy, A. Eremin, W. Weissflog, and A. Jákli, "Bundles of fluid fibers formed by bent-core molecules," *Physical Review E*, vol. 81, pp. 031708–6, mar 2010.

[24] A. Eremin, U. Kornek, S. Stern, R. Stannarius, F. Araoka, H. Takezoe, H. Nádasi, W. Weissflog, and A. Jákli, "Pattern-stabilized decorated polar liquid-crystal fibers," *Physical Review Letters*, vol. 109, no. 1, p. 017801, 2012.

[25] T. Ostapenko, M. Weyland, A. Eremin, M. Lehmann, and R. Stannarius, "Filaments formed in the hexagonal columnar liquid crystal phase of star-shaped oligobenzoates," *Liquid Crystals*, vol. 40, no. April 2013, pp. 345–353, 2013.

[26] J. Petzold, a. Nemeş, a. Eremin, C. Bailey, N. Diorio, a. Jákli, and R. Stannarius, “Acoustically driven oscillations of freely suspended liquid crystal filaments,” *Soft Matter*, vol. 5, p. 3120, 2009.

[27] R. Stannarius, A. Nemeş, and A. Eremin, “Plucking a liquid chord: Mechanical response of a liquid crystal filament,” *Physical Review E*, vol. 72, p. 020702, aug 2005.

[28] G. Pelzl, “Preliminary communication Helical superstructures in a novel smectic mesophase formed by achiral banana-shaped molecules,” *Liquid Crystals*, vol. 26, no. 1, pp. 135–139, 1999.

[29] A. Nemeş, A. Eremin, R. Stannarius, M. Schulz, H. Nádasi, and W. Weissflog, “Structure characterization of free-standing filaments drawn in the liquid crystal state.,” *Physical chemistry chemical physics : PCCP*, vol. 8, pp. 469–476, 2006.

[30] P. M. Chaikin and T. C. Lubensky, *Principles of Condensed Matter Physics*. Cambridge University Press, 1995.

[31] F. C. Frank, “Liquid Crystals–On the theory of liquid crystals,” *Trans Faraday Soc II*, vol. 25, no. I, pp. 19–28, 1958.

[32] F. Reinitzer, “Beiträge zur kenntniss des cholesterins,” *Monatshefte für Chemie und verwandte Teile anderer Wissenschaften*, vol. 9, pp. 421–441, Dec 1888.

[33] D. Vorländer, “Die Richtung der Kohlenstoff Valenzen in Benzol Abkömmlingen,” . . . *der deutschen chemischen Gesellschaft (A and B . . .*, vol. 62, p. 2831, 1929.

[34] D. Dunmur, *Soap, science, and flat-screen TVs : a history of liquid crystals*. Oxford: Oxford University Press, 2014.

[35] C. Weygand, “Daniel Vorländer. 11. 6. 1867-8. 6. 1941,” *Berichte der deutschen chemischen Gesellschaft (A and B Series)*, vol. 76, pp. A41–A58, jun 1943.

[36] Y. Matsunaga and S. Miyamoto, “Mesomorphic behavior of 2,4-bis-(4-alkoxybenzylidene)cyclopentanones and related compounds,” *Molecular Crystals and Liquid Crystals Science and Technology. Section A. Molecular Crystals and Liquid Crystals*, vol. 237, no. 1, pp. 311–317, 1993.

[37] T. Niori, T. Sekine, J. Watanabe, T. Furukawa, and H. Takezoe, “Distinct ferroelectric smectic liquid crystals consisting of banana shaped achiral molecules,” *Journal of Materials Chemistry*, vol. 6, no. 7, pp. 1231–1233, 1996.

[38] A. Eremin and A. Jákli, “Polar bent-shape liquid crystals – from molecular bend to layer splay and chirality,” dec 2013.

[39] R. B. Meyer, “Ferroelectric Liquid Crystals; A Review,” *Molecular Crystals and Liquid Crystals*, vol. 40, no. February 2015, pp. 33–48, 1977.

[40] H. Takezoe and Y. Takanishi, “Bent-core liquid crystals: Their mysterious and attractive world,” *Japanese Journal of Applied Physics, Part 1: Regular Papers and Short Notes and Review Papers*, vol. 45, no. 2, pp. 597–625, 2006.

[41] D. Phillips and P. Bauman, “Analysis and stability of bent-core liquid crystal fibers,” *Discrete and Continuous Dynamical Systems - Series B*, vol. 17, no. 6, pp. 1707–1728, 2012.

[42] A. Jákli, C. Lischka, W. Weissflog, G. Pelzl, and A. Saupe, “Helical filamentary growth in liquid crystals consisting of banana-shaped molecules,” 2000.

[43] O. H. Kress, *Mechanical Tension and Electrical Conductivity of Liquid Crystal Filaments*. PhD thesis, Kent State University Liquid Crystal Institute, 2015.

[44] J. Matta and R. Tytus, “Liquid stretching using a falling cylinder,” *Journal of Non-Newtonian Fluid Mechanics*, vol. 35, no. 2-3, pp. 215–229, 1990.

[45] T. Sridhar, V. Tirtaatmadja, D. A. Nguyen, and R. K. Gupta, “Measurement of extensional viscosity of polymer solutions,” *Journal of Non-Newtonian Fluid Mechanics*, vol. 40, no. 3, pp. 271–280, 1991.

[46] V. Tirtaatmadja, “A filament stretching device for measurement of extensional viscosity,” *Journal of Rheology*, vol. 37, no. 6, p. 1081, 1993.

[47] S. Berg, R. Kröger, and H. Rath, “Measurement of extensional viscosity by stretching large liquid bridges in microgravity,” *Journal of Non-Newtonian Fluid Mechanics*, vol. 55, pp. 307–319, dec 1994.

[48] P. Szabo, “Transient Filament Stretching Rheometer I: Force Balance Analysis,” *Rheologica Acta*, vol. 36, pp. 277–284, 1997.

[49] S. H. Spiegelberg, D. C. Ables, and G. H. McKinley, “The role of end-effects on measurements of extensional viscosity in filament stretching rheometers,” *Journal of Non-Newtonian Fluid Mechanics*, vol. 64, no. 2-3, pp. 229–267, 1996.

[50] V. Tirtaatmadja and T. Sridhar, “A filament stretching device for measurement of extensional viscosity,” *Journal of Rheology*, vol. 37, no. 6, pp. 1081–1102, 1993.

[51] S. Paul, C. Kalelkar, and P. A. Pullarkat, “Oscillatory extensional rheology of microscale fluid filaments,” *Rheologica Acta*, vol. 56, no. 2, pp. 113–122, 2017.

[52] S. R. R. V., C. Kalelkar, and P. A. Pullarkat, “Optical fiber-based force transducer for microscale samples,” *Review of Scientific Instruments*, vol. 84, p. 105107, oct 2013.

[53] C. T. Kreis, M. Le Blay, C. Linne, M. M. Makowski, and O. Bäumchen, "Adhesion of Chlamydomonas microalgae to surfaces is switchable by light," *Nature Physics*, vol. 14, p. 45, sep 2017.

[54] M. J. Colbert, A. N. Raegen, C. Fradin, and K. Dalnoki-Veress, "Adhesion and membrane tension of single vesicles and living cells using a micropipette-based technique," *The European Physical Journal E*, vol. 30, no. 2, p. 117, 2009.

[55] G. H. Mckinley, S. L. Anna, a. Tripathi, M. Yao, M. Engineering, and C. Ma, "Extensional rheometry of polymeric fluids and the uniaxial elongation of viscoelastic filaments," *International Polymer Processing Society*, pp. 1–14, 1999.

[56] P. K. Bhattacharjee, J. P. Oberhauser, G. H. McKinley, L. G. Leal, and T. Sridhar, "Extensional rheometry of entangled solutions," *Macromolecules*, vol. 35, no. 27, pp. 10131–10148, 2002.

[57] G. H. McKinley, "Steady and transient motion of spherical particles in viscoelastic liquids," *Transport Processes in Bubbles, Drops & Particles*, pp. 338–375, 2001.

[58] G. H. Mckinley and T. Sridhar, "Filament-Stretching Rheometry of Complex Fluids," *Annu. Rev. Fluid Mech.*, vol. 34, pp. 375–415, 2002.

[59] S. L. Anna, C. Rogers, and G. H. McKinley, "On controlling the kinematics of a filament stretching rheometer using a real-time active control mechanism," *Journal of Non-Newtonian Fluid Mechanics*, vol. 87, no. 2-3, pp. 307–335, 1999.

[60] G. H. Mckinley, O. Brauner, and M. Yao, "Filament Stretching Rheometry and the Extensional Viscosity of Dilute and Concentrated Polymer Solutions," *Symposium A Quarterly Journal In Modern Foreign Literatures*, pp. 65–75, 2001.

[61] N. V. Orr and T. Sridhar, "Stress relaxation in uniaxial extension," *Journal of Non-Newtonian Fluid Mechanics*, vol. 67, no. 1-3, pp. 77–103, 1996.

[62] P. Szabo, G. H. McKinley, and C. Clasen, "Constant force extensional rheometry of polymer solutions," *Journal of Non-Newtonian Fluid Mechanics*, vol. 169-170, pp. 26–41, feb 2012.

[63] S. Kim, S. R. Aglyamov, S. Park, M. O'Donnell, and S. Y. Emelianov, "An autocorrelation-based method for improvement of sub-pixel displacement estimation in ultrasound strain imaging," *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 58, no. 4, pp. 838–843, 2011.

[64] C. Bailey, K. Fodor-Csorba, J. T. Gleeson, S. N. Sprunt, and A. Jákli, "Rheological properties of bent-core liquid crystals," *Soft Matter*, vol. 5, no. 19, p. 3618, 2009.

[65] G. Pelzl, "Structural and electro-optical investigations of the smectic phase of chlorine-

substituted banana-shaped compounds," *Liquid Crystals*, vol. 26, no. 3, pp. 401–413, 1999.

[66] D. a. Coleman, J. Fernsler, N. Chattham, M. Nakata, Y. Takanishi, E. Körblova, D. R. Link, R.-F. Shao, W. G. Jang, J. E. Maclennan, O. Mondainn-Monval, C. Boyer, W. Weissflog, G. Pelzl, L.-C. Chien, J. Zasadzinski, J. Watanabe, D. M. Walba, H. Takezoe, and N. a. Clark, "Polarization-modulated smectic liquid crystal phases.," *Science (New York, N.Y.)*, vol. 301, no. 5637, pp. 1204–1211, 2003.

[67] J. Petzold, a. Nemeş, a. Eremin, C. Bailey, N. Diorio, a. Jákli, and R. Stannarius, "Acoustically driven oscillations of freely suspended liquid crystal filaments," *Soft Matter*, vol. 5, p. 3120, 2009.

[68] L. Rayleigh, "On The Instability Of Jets," *Proceedings of the London Mathematical Society*, vol. s1-10, pp. 4–13, nov 1878.

[69] P. C. Sousa, E. J. Vega, R. G. Sousa, J. M. Montanero, and M. A. Alves, "Measurement of relaxation times in extensional flow of weakly viscoelastic polymer solutions," *Rheologica Acta*, vol. 56, pp. 11–20, Jan 2017.

[70] S. L. Anna, G. H. McKinley, D. A. Nguyen, T. Sridhar, S. J. Muller, J. Huang, and D. F. James, "An interlaboratory comparison of measurements from filament-stretching rheometers using common test fluids," *Journal of Rheology*, vol. 45, no. 1, pp. 83–114, 2001.

[71] A. Roy, L. Mahadevan, and J.-L. Thiffeault, "Fall and rise of a viscoelastic filament," *Journal of Fluid Mechanics*, vol. 563, p. 283, 2006.

[72] C. J. S. Petrie, "Extensional viscosity: A critical discussion," *Journal of Non-Newtonian Fluid Mechanics*, vol. 137, no. 1-3, pp. 15–23, 2006.

[73] C. Solomon and T. Breckon, *Fundamentals of Digital Image Processing: A Practical Approach with Examples in Matlab*, vol. 14. Wiley-Blackwell, 2011.

[74] L. D. Landau, E. M. Lifshitz, J. B. Sykes, W. H. Reid, and E. H. Dill, "Theory of Elasticity: Vol. 7 of Course of Theoretical Physics," 1960.

[75] R. M. Christensen, *Theory of viscoelasticity*. Mineola, N.Y: Dover Publications, 2003.

[76] N. W. Tschoehl, *The Phenomenological Theory of Linear Viscoelastic Behavior*. Springer-Verlag, 1989.

[77] R. Christensen, *Theory of Viscoelasticity: Second Edition*. Dover Civil and Mechanical Engineering, Dover Publications, 2013.

[78] R. Szeliski, *Computer Vision*, vol. 42. Springer, 2011.

[79] A. Jákli, *One- and two-dimensional fluids : properties of smectic, lamellar and columnar liquid crystals.* New York: Taylor & Francis, 2006.

[80] I. Dahl, "The elasticity theory of the smectic c phase: The case of compressible smectic layers," *Ferroelectrics*, vol. 113, no. 1, pp. 121–141, 1991.

Appendices

APPENDIX A

CODE SNIPPETS

A.1 Heat Stage Thermal Simulation

```
%Heat conduction in 3D for heat stage/fiber puller%
%Finite difference transient solutions%
%Thermal conductivities as a function of temperature%
%Define a temperature gradient outside of the device for the air in the
%near region, but steeply dropping off to room temp



 clc
 clear


%Total Computational Space%

Lx = 0.3;                %meters%
Ly = 0.3;
Lz = 0.3;

%Number of nodes   adjusted for square mesh in x,y,z space%
%xnodes defines mesh coursness%

 xnodes = 60;
 ynodes = Ly/Lx*xnodes;
 znodes = Lz/Lx*xnodes;

%Spacial Step Sizes for all directions : delX = delx = dely = delz%

 delX = Lx/(xnodes);     %meters%
```

```
delY = Ly/(ynodes);
delZ = Lz/(znodes);

%Time%

delt = .05;                %seconds%
tinitial = 0;
tfinal = 600;
tsteps = tfinal/delt;

%Properties of air at room temp   constant for now (later wrt temp)%

kair = 0.0257;              %conduction coefficient at room temp% %W/m K
rhoair = 1.205;             %density of air %kg/m^3%
Cair = 1000;                %specific heat of air  %J/kg K%
alphaair = kair/(rhoair*Cair);   %Diffusivity of air%
% hair =                   %Convection coefficient%


%Poperties of Teflon%

kteflon = 0.25;
rhoteflon = 2200;
Cteflon = 970;

%Properties of Glass%

kglass = 1;
rhoglass = 2500;
Cglass = 750;


%Properties of Copper%
kcu = 387;
rhocu = 8940;
Ccu = 380;
alphacu = kcu/(rhocu*Ccu);
%hcu
```

```

%Properties of Steel ( stainless)%

 ksteel = 16;
 rhosteel = 7850;
 Csteel = 460;
 alphasteel = ksteel/(rhosteel*Csteel);
%hsteel

%Initializing the 3D temperature matrix in i,j,k space%

 Tempinitial = 20;  %20C   room temp%

T(1:xnodes, 1:ynodes, 1:znodes) = Tempinitial; %Initial temp for first time step


%Initial non temp dependent conduction coefficients for all regions of all materials%

lambda(1:xnodes, 1:ynodes, 1:znodes) = kair;  %first make the entire comp space Kair, then replace non air with the materials%
lambda(24:36,24:36,28:31) = ksteel;  %outtersteelframe 1.5 cm high%
lambda(29:31,29:31,28) = kglass; %bottomglass%
lambda(25:35,25:35,29:31) = kteflon; %main teflon liner%
lambda(27:33,26:34,30) = kcu; %copper heater%
    lambda(28,27,30:32) = kcu; % the 4 copper rods
    lambda(28,33,30:32) = kcu;
    lambda(32,27,30:32) = kcu;
    lambda(32,33,30:32) = kcu;
lambda(27:33,27:33,31) = kair; %inner air chamber

%lid%

lambda(24:36,24:36,32) = ksteel;
lambda(25:35,25:25,32) = kteflon;
lambda(29:31,29:31,32) = kglass;

```

```


%volumetric heat capacity for all regions of all materials%

C(1:xnodes, 1:ynodes, 1:znodes) = Cair*rhoair;  %first make the entire comp space Kair, then replace non air with the following%
C(24:36,24:36,28:31)  = Csteel*rhosteel;  %outtersteelframe 1.5 cm high%
C(29:31,29:31,28) = Cglass*rhoglass; %bottomglass%
C(25:35,25:35,29:31) = Cteflon*rhoteflon;  %main teflon liner%
C(27:33,26:34,30) = Ccu*rhocu; %copper heater%
     C(28,27,30:32) = Ccu*rhocu; % the 4 copper rods
     C(28,33,30:32) = Ccu*rhocu;
     C(32,27,30:32) = Ccu*rhocu;
     C(32,33,30:32) = Ccu*rhocu;
C(27:33,27:33,31) = Cair*rhoair; %inner air chamber

%lid%

C(24:36,24:36,32) = Csteel*rhosteel;
C(25:35,25:25,32) = Cteflon*rhoteflon;
C(29:31,29:31,32) = Cglass*rhoglass;



%Heat source values (volumetric watt density) for nodes generating energy
I(1:xnodes, 1:ynodes, 1:znodes) = 0;
I(27:33,26,30) = 30*10^6; %W/m^3%
I(27:33,34,30) = 30*10^6; %W/m^3%


%First scan across 2D x,y plane and then move up through z space %

t = tinitial;
H=zeros(xnodes,ynodes,znodes);
p=0;
plotinterval = 1:50:tsteps; %how frequently to plot   plot speed
```

```

[X,Y,Z] = meshgrid(1:xnodes, 1:ynodes, 1:znodes);

%Z = zeros(xnodes,ynodes); %makes 3d surface plot 2D projection instead

% for h = 2:znodes
% Z(:,:,h) = Z(:,:,h 1)+1;          %z stack matrix
% end

% alf = zeros(znodes,1);                          %Opacity vector
% alfmax = 0.9;
% dalf = 1.8/znodes;
% alf(znodes/2) = alfmax;
%
%
% for a = 1:znodes/2
%
% alf((znodes/2)+a) = alfmax   a*dalf;
%
% if a < znodes/2
% alf((znodes/2) a) = alfmax   a*dalf;
% end
%
%
% end



frame = 0;




while t < tfinal

    averageTbefore = sum(sum(sum(T(:,:,:))))/xnodes^3;

    for k = 2:znodes 1;  %boundaries remain at room temp from Tinitial (BCs)
```

```


        for j = 2:ynodes 1; %boundaries remain at room temp from Tinitial (BCs)


               for i = 2:xnodes 1; %boundaries remain at room temp from Tinitial (BCs)



                   H(i,j,k) =  (delX*delX/(delX/(2*lambda(i,j,k))+delX/(2*lambda(i 1,j,k))))*(T(i 1,j,k) T(i,j,k)) ...
                             + (delX*delX/(delX/(2*lambda(i,j,k))+delX/(2*lambda(i+1,j,k))))*(T(i+1,j,k) T(i,j,k)) ...
                             + (delX*delX/(delX/(2*lambda(i,j,k))+delX/(2*lambda(i,j 1,k))))*(T(i,j 1,k) T(i,j,k)) ...
                             + (delX*delX/(delX/(2*lambda(i,j,k))+delX/(2*lambda(i,j+1,k))))*(T(i,j+1,k) T(i,j,k)) ...
                             + (delX*delX/(delX/(2*lambda(i,j,k))+delX/(2*lambda(i,j,k 1))))*(T(i,j,k 1) T(i,j,k)) ...
                             + (delX*delX/(delX/(2*lambda(i,j,k))+delX/(2*lambda(i,j,k+1))))*(T(i,j,k+1) T(i,j,k)) ...
                             + (I(i,j,k)*(delX^3));

               end

        end

    end

    for k = 2:znodes 1;  %boundaries remain at room temp from Tinitial (BCs)

```

```

        for j = 2:ynodes 1; %boundaries remain at room temp from Tinitial (BCs)


            for i = 2:xnodes 1; %boundaries remain at room temp from Tinitial (BCs)

                T(i,j,k) = T(i,j,k)+ H(i,j,k)*delt/(C(i,j,k)*delX^3);

            end

        end

    end

    t;

%
% figure (1)
%  scatter(t,averageTafter)
%  drawnow
%  hold on

%figure (2)

p = p + 1;

% if any(p==plotinterval) == 1
%
%     frame = frame + 1;
%
% for k = 1:znodes
%     for j = 1:ynodes
%         for i = 1:xnodes
%
%
%                 scatter3(X(i,j),Y(i,j),k,17,T(i,j,k),'filled')
%                 title([num2str(t) ' seconds'])
%                 caxis([0, 200])
%                 axis([1 xnodes 1 ynodes 1 znodes])
%                 set(gcf,'Visible','Off')
%                 hold on
%
%         end
%     end
% end
%
% print(gcf,' dbmp',['scatterheat' num2str(frame) '.bmp']);
%
% end


% if any(p==plotinterval) == 1
%
% frame = frame + 1;
%
%     %no need to loop over layers with slice function
%
% %   for layer = 1:10:znodes
% %
% %
% %     Ttop = T(:,:,layer); %Select which 2D plane to represent. stack z layers
% %     surf(X,Y,Z(:,:,layer+1),Ttop)
% %     title([num2str(t) ' seconds'])
% %     Tmax = 100;
% %     axis([1 50 1 50 0 50])
% %     colorbar
% %     caxis([20 Tmax])                % set color scale
% %     caxis manual                    % hold color scale across loops
% %     set(gcf,'Visible','On')        %Keep from drawing after each loop
% %     shading INTERP
% %     hold on
% %     alphamap('rampdown')
% %     alphamap('increase',.1)                  %opacity
% %
```

```
%%    end
%
%
%
%       h = slice(X,Y,Z,T,[30],[30],[30]);
%       title([num2str(t) ' seconds']);
%       Tmax = 220;
%       axis([1 60 1 60 0 60]);
%       caxis([20 Tmax]);                     % set color scale
%       caxis manual                        % hold color scale across loops
%       colorbar
%       set(h,'Visible','On','EdgeColor','none','FaceColor','interp','FaceAlpha','interp');
%       shading INTERP
%       alpha('color');
%       alphamap('rampup');
%       alphamap('decrease',0);
%       alim([20 Tmax]);
%       alim('manual')
%       set(gcf, 'Visible','On')
%
%
%%print(gcf,'dbmp',['3ddeviceslice' num2str(frame) '.bmp']);
%
%
%
% end
%

    t = t + delt


end


%
% for i = 1:frame
% filename = sprintf('3ddeviceslice%i.bmp', i);
% [bmp, map] = imread(filename, 'bmp');
% M(i) = im2frame(bmp, map);
% end
%
%
% movie2avi(M,'3ddeviceslice','compression','none');
%
%

%
% for k = 2:znodes 1
%    for j = 2:ynodes 1
%       for i = 2:xnodes 1
%
%          stabarray(i 1,j 1,k 1) = C(i,j,k)*(delX*delY*delZ)/((delX*delX/(delX/(2*lambda(i,j,k))+delX/(2*lambda(i 1,j,k)))) ...
%                      + (delX*delX/(delX/(2*lambda(i,j,k))+delX/(2*lambda(i+1,j,k)))) ...
%                      + (delX*delX/(delX/(2*lambda(i,j,k))+delX/(2*lambda(i,j 1,k)))) ...
%                      + (delX*delX/(delX/(2*lambda(i,j,k))+delX/(2*lambda(i,j+1,k)))) ...
%                      + (delX*delX/(delX/(2*lambda(i,j,k))+delX/(2*lambda(i,j,k 1)))) ...
%                      + (delX*delX/(delX/(2*lambda(i,j,k))+delX/(2*lambda(i,j,k+1)))));
%       end
%    end
% end
%
% minstabarray = (min(min(min(stabarray(2:xnodes 2,2:ynodes 2,2:znodes 2)))));
% if delt <= minstabarray
% stability = 'success'
% else
%    stability = 'fail'
% end



% for k = 1:znodes
%     for j = 1:ynodes
%         for i = 1:xnodes
%
%                 scatter3(X(i,j),Y(i,j),k,20,T(i,j,k),'filled')
%                 caxis([0, 200])
%                 axis([1 xnodes 1 ynodes 1 znodes])
%                 hold on
%
%         end
%     end
% end





% for k = 1:5:100
%     figure
% imagesc(squeeze(T(:,:,k)))
% hold on
% end
```

A.2 Pipette Tracking

User Dialogue

```
     Date                           Index



 'August 30 2018'                      1
 'File Tree'                           2
 'July 02 2018'                        3
 'May 16 2018'                         4
 'May 17 2018'                         5
 'May 23 2018'                         6
 'May 30 2018'                         7

Please choose the date of the experiment by typing the
```

```
corresponding value of the index in t he table above , followed
by t he Return key 1

     Exp                            Index


 ' Exp 01'                             1
 ' Exp 02'                             2
 ' Exp 03'                             3
 ' Exp 04'                             4
 ' Exp 05'                             5
 ' Exp 06'                             6
 ' Exp 07'                             7
 ' Exp 08'                             8
 ' Exp 09'                             9
 ' Exp 10'                            10

Please choose the correct experiment by typing the corresponding
value of the index in the table above, followed by the
Return key     7



     Pip ID                         Index



 ' Archive'                            1
 ' Ref Pip 1'                          2
 ' Ref Pip 2'                          3
 ' Ref Pip 3'                          4
 ' Ref Pip 4'                          5
 ' Ref Pip 8'                          6


Which pipette are you using? 4


This is to verify that this experiment was recorded using a
point grey camera using an objective with 20X magnification
```

```
and 1X of intrinsic magnification resulting in a pixel to
micron ratio of 5.88 px /um, while recording at 10 FPS, using
a pipette with a spring constant of 107.3 nN /um.
The first image in the directory has the number 0 .
The last image in the directory has the number 1652.
At which picture do you want to start the analysis? (default
value i s 0) The analsys is will start with the first picture in
the folder
At which picture do you want to end the analysis? (default value
is 1652)
The analysis will end a t image number 1652
The analysis will use 1653 images .


Pick the line you want to use for this set of pictures (top to
bottom or left to right)
Does the position of line please you? yes is 1, no is 0, default
value is 1 .
We can start the calculations!
progress 20
progress 40
progress 60
...
...
...
```

Tracking Code

```
% *********** Correlation and fit correlation *********** %

% Parameters
Max_Find_Step = 0.001 ; % pixels
Deflections = zeros(ImagesNumber, 2) ;
% column 1 are the raw value, column 2 the fit values of the delfections

% Correlation calculation on the first image
PCD_Ref = double(FirstImage(P_1(2) : P_2(2), P_1(1) : P_2(1))) ; % changes the type of numbers so the correlation function works
if vertical == 0
```

```matlab
    PCD_Ref = PCD_Ref' ;
end
Correlation_Values = xcov(PCD_Ref, 'coeff') ;
[~, Deflections(1, 1)] = max(Correlation_Values) ;

% Correlation fit one the first image (auto correlation)
Correlation_Fit = fit((Deflections(1, 1)   Fit_Width : Deflections(1, 1) + Fit_Width)',...
    Correlation_Values(Deflections(1, 1)   Fit_Width : Deflections(1, 1) + Fit_Width), Fit_Method, 'SmoothingParam', Fit_para);

[~, Max_Fit_Position] = max(Correlation_Fit(Deflections(1, 1)   Fit_Width : Max_Find_Step : Deflections(1, 1) + Fit_Width)) ;
Deflections(1, 2) = Deflections(1, 1)   Fit_Width + Max_Find_Step * Max_Fit_Position ;
%% Correlation on all the images
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

for j = 2 : ImagesNumber
    %disp(j) ;
    Image = imread(ImagesList{j}) ;     %reads image

    PCD = double(Image(P_1(2) : P_2(2), P_1(1) : P_2(1))) ;  % changes the type of numbers so the correlation function works
    if vertical == 0
        PCD = PCD' ;
    end

    Correlation_Values = xcov(PCD_Ref, PCD, 'coeff');   % This seems to do the actual correlation calculation
    [~, Deflections(j, 1)] = max(Correlation_Values) ;

    Correlation_Fit = fit(((Deflections(j, 1)   Fit_Width) : (Deflections(j, 1) + Fit_Width))',...
        Correlation_Values(Deflections(j, 1)   Fit_Width : Deflections(j, 1) + Fit_Width), Fit_Method, 'SmoothingParam', Fit_para);

    [~, Max_Fit_Position] = max(Correlation_Fit(Deflections(j, 1)  Fit_Width : Max_Find_Step : Deflections(j, 1) + Fit_Width)) ;
    Deflections(j, 2) = Deflections(j, 1)  Fit_Width + Max_Find_Step * Max_Fit_Position ;

    if mod(j,20)==0
        fprintf('progress %d\n', j);
    end
end
```

A.3 Fiber Diameter Measurement

Measurement Code

```
%% Scan vertically across the length and measure the diameter

%%% Pick some number of lines, N_Lines, to get statistics. Evenly
%%% distributed along the length.
N_Lines_0 = 100;
L0_Pixels = P_Right  P_Left;
Scan_line_length = 20;%microns
Scan_line_pixels = round(Scan_line_length*pixelpermicron); % pixels
Y_0 = (EP_1(2)+ EP_2(2))/2; %mean Y position of the fiber in pixels

%minimize modulo to evenly space integer number of scan lines starting with
%initial number of lines N_Lines_0

j_steps = floor(mod(L0_Pixels,N_Lines_0)/floor(L0_Pixels/N_Lines_0)); %this operation tell you by how much to change N_Lines, in integer steps, to minimize modulo
N_Lines = N_Lines_0 + j_steps

Line_Sep = floor(L0_Pixels/(N_Lines)); %distance in microns between scan lines
```

```
Scan_Line_X_Array(1:N_Lines 1) = P_Left + (1:N_Lines 1)*Line_Sep; % exclude scanning near the interfaces
Scan_Line_Y_Start_Array(1:N_Lines 1) = round(Y_0 Scan_line_pixels/2);
Scan_Line_Y_End_Array(1:N_Lines 1) = round(Y_0+Scan_line_pixels/2);

%Draw the vertical scan lines

for i = 1:N_Lines 1
plot([Scan_Line_X_Array(i) Scan_Line_X_Array(i)],[Scan_Line_Y_Start_Array(i) Scan_Line_Y_End_Array(i)], ' ', 'LineWidth', 2, 'Color', 'b')
hold on
end
hold off

%%
for i = 1:5%N_Lines 1

Fiber_Profile = double(FirstImage(Scan_Line_Y_Start_Array(i) : Scan_Line_Y_End_Array(i), Scan_Line_X_Array(i) : Scan_Line_X_Array(i)));
Fiber_Profile = 255 Fiber_Profile; %Invert for gaussian fitting

Scan_midpoint = Scan_line_pixels/2;

%need x column vector for fitting
x = 1:length(Fiber_Profile);
x=x';

%plot(x,Fiber_Profile)

%concatenate both vectors to account for skipping data between peaks
int_thresh = 10;
thresh_left = find(Fiber_Profile>int_thresh,1); %first meaningful increase in intensity

%find first peak  increase index while derivative is positive
```

```
j=1;
while Fiber_Profile(thresh_left+j)  Fiber_Profile(thresh_left + (j 1)) > 0
    j=j+1;
end
peak_left_X = thresh_left + (j 1); %correct position of first peak (index)

%keep going to find following minimum .step forward while derivative is
%negative
k=1;
while Fiber_Profile(peak_left_X+k)  Fiber_Profile(peak_left_X + (k 1)) < 0
    k=k+1;
end
min_X = peak_left_X + (k 1);

%Are there any other intensity spikes above the threshold to the right of
%the min?
if max(Fiber_Profile(min_X):Fiber_Profile(length(Fiber_Profile))) > int_thresh
    [Max_X2,peak_right_X] = max(Fiber_Profile(min_X):Fiber_Profile(length(Fiber_Profile)));
else
    peak_right_X = peak_left_X;
end

%if peaks are the same, do nothing. else get rid of the stuff inbetween
if peak_right_X == peak_left_X
else
Fiber_Profile(peak_left_X:peak_right_X)= [];
x(peak_left_X:peak_right_X) = [];
end


%hold on
%plot(x,Fiber_Profile)
```

```


gaus_fit_options = fitoptions('gauss1');
if peak_right_X == peak_left_X
    gauss_center_0 = peak_left_X;
else
    gauss_center_0 = (peak_right_X  peak_left_X)/2;
end

center_upper = gauss_center_0 + 2;
center_lower = gauss_center_0  2;

gaus_fit_options.StartPoint = [100 gauss_center_0 3] ;
gaus_fit_options.Lower = [0 center_lower 0];
gaus_fit_options.Upper = [255 center_upper 10];

%Apply fit
Fiber_Diam_Fit = fit(x,Fiber_Profile,'gauss1',gaus_fit_options);

Fiber_Amp = Fiber_Diam_Fit.a1;
Fiber_Center = Fiber_Diam_Fit.b1;
Fiber_Gauss_C = Fiber_Diam_Fit.c1;

gauss_plot = Fiber_Amp.*exp( ((x Fiber_Center)/Fiber_Gauss_C)
    .^2);



Fiber_Sigma = sqrt(Fiber_Gauss_C^2/2);

Fiber_FWHM = 2*sqrt(2*log(2))*Fiber_Sigma  ;%in pixels
Fiber_FWHM_microns(i) = Fiber_FWHM/pixelpermicron; %in microns

end

%% convert to vector in case it's rows
size_fiber_FWHM = size(Fiber_FWHM_microns);
if size_fiber_FWHM(2) > 1
    Fiber_FWHM_microns = Fiber_FWHM_microns';
end
```

```


Low_diam_Val_ind = find(Fiber_FWHM_microns<1); %get indices for all fiber diameters below 1 microns since these are aphysical and therefore bad fits and not needed
High_diam_Val_ind = find(Fiber_FWHM_microns>2.5); %same for upper limit on diameter. typically 2.5 is a good place to start. inspect binning per trial.
Scrap_diams = [High_diam_Val_ind; Low_diam_Val_ind]; %collect all indices corresponding to aphysical diameter values
Fiber_FWHM_microns(Scrap_diams)=[];% erase these values from the array



figure
fiber_hist_fit = histfit(Fiber_FWHM_microns);

%do hist fit and get results
pd = fitdist(Fiber_FWHM_microns,’Normal’);
D_0 = pd.mu
fiber_diam_sigma = pd.sigma
```

A.4 Rheological Strain Profile Algorithms

Initialization

```
// includes
#include ”stdafx.h”
#include <windows.h>
#include <conio.h>
#include <time.h>
#include ”PI_GCS2_DLL.h”
#include <iostream>
#include <fstream>
#include <cmath>
#include <time.h>


// Helper Functions
```

```
bool ReferenceIfNeeded(int piDeviceId, char* axis);
void ReportError(int piDeviceId);
void CloseConnectionWithComment(int piDeviceId, const char* comment);
bool WaitForMovementStop(int piDeviceId, char* axis1);
void ShowAvailablePiControllers ();

// Additional Sample Functions
bool ReadRecordedData(int iD, char * axis, double Position, double MaxPosValue);
int ChangeConnectedStage(int piDeviceId, char * axis);
//bool MoveTwoAxes ( int piDeviceId );

//Movement Functions
int TaylorSin(int piDeviceId, const char* axis1, double Pinit, int NCycles, double omega, double pi, double a1, double a2, double a3, double a4, double a5, double AmpL, double PMax, double SysDelay);
int TriSin(int piDeviceId, const char* axis1, double Pinit, double AmpL, double AmpLo2, int NCycles, double V1, double V2, double SysDelay, double SleepyTime1, double SleepyTime2, double SleepyTime3, double SleepyTime1t2);
int SinOsc(int piDeviceId, const char* axis1, double Pinit, int NCycles, double omega, double pi, double a1, double a2, double a3, double a4, double a5, double AmpL, double PMax, double SysDelay);



int main(int argc, char* argv[])
{
    ////////////////////////////////////////////////////////////
    // Start Connection
    // You MUST EDIT AND ACTIVATE the connection parameters.
    ////////////////////////////////////////////////////////////



    //  // USB
      char controllerSerialNumber[100] = ”416004218”; //”Enter
```

```
         valid serial number here, e.g. 123456789”   // Use ”
         EnumerateUSB(””)” to get all PI controller connected to
          you PC. Or look at the label of the case of your
         controller
      int piDeviceId = PI_ConnectUSB ( controllerSerialNumber );


    if (piDeviceId < 0)
    {
        std::cout<<”Could not connect to controller.\n”;
        _getch();
        return false;
    }

    // Query controller identification
    char deviceIdentificationString[200];
    if (PI_qIDN(piDeviceId, deviceIdentificationString, 199) ==
       false)
    {
        CloseConnectionWithComment(piDeviceId, ”qIDN failed.
           Exiting.\n”);
        return false;
    }
    std::cout << ”\nqIDN returned: ” <<
       deviceIdentificationString << ”\n”;


     ////////////////////////////////////////////////////////
     // Startup stage
     ////////////////////////////////////////////////////////

     // Query controller axes
     char availableAxes[200];
     if (!PI_qSAI_ALL(piDeviceId, availableAxes, 199))
     {
         CloseConnectionWithComment(piDeviceId, ”Could not send
            SAI?\n”);
         return false;
     }
     std::cout << ”Available axes:\n” << availableAxes << ”\n\n”
```

```
        ;

     // Use axis 1
     char axis1 [2] = "1";

    // Query which stage type the controller is configured for

    char currentStageType [256];
    if( !PI_qCST (piDeviceId, axis1, currentStageType, 255) )
    {
        CloseConnectionWithComment(piDeviceId,"Failed to query current stage type.\n");
        return false;
    }
    std::cout << "Axis " << axis1 << " is configured to control " << currentStageType << std::endl;


     // Switch on servo
     const BOOL servoOn = true;
     if (!PI_SVO(piDeviceId, axis1, &servoOn) )
     {
         CloseConnectionWithComment(piDeviceId, "SVO failed. Exiting.\n");
         return false;
     }

         // set stage velocity to max initially
         double Vel0 = 10;
         if (!PI_VEL(piDeviceId, axis1, &Vel0))
         {
                 CloseConnectionWithComment(piDeviceId, "Unable to set stage velocity\n");
                 return false;
         }

     //// Reference stage
     //std::cout << "Referencing axis " << axis1 << "\n";
     //if (!PI_FRF(piDeviceId, axis1))
     //{
```

```
    //    return false;
    //}

    // Wait until the reference move is done.
    BOOL referencingFinished;
    referencingFinished = false;
    while ( TRUE != referencingFinished )
    {
        if ( !PI_IsControllerReady ( piDeviceId, &
           referencingFinished ) )
        {
            return false;
        }
    }

    std::cout << ”Stage is referenced.\n”;
    Sleep ( 500 );


    // ///////////////////////////////////////////
    // Move stage
    // //////////////////////////////////////////


    // Determine boundaries for stage movement
    double travelRangeMinimum;
    double travelRangeMaximum;


    if (!PI_qTMN(piDeviceId, axis1, &travelRangeMinimum))
    {
        CloseConnectionWithComment(piDeviceId, ”qTMN unable to
           query min. position of axis.\n”);
        return false;
    }

    if (!PI_qTMX(piDeviceId, axis1, &travelRangeMaximum))
    {
        CloseConnectionWithComment(piDeviceId, ”qTMX, Unable to
            query max. position of axis.\n”);
            return false;
        }


        std::cout << "Allowed range of movement: min: " << travelRangeMinimum << "\t max: " << travelRangeMaximum << "\n";
```

Constant Strain Rate

```
// Run loop for exponential velocity profile by first asking for L0 and the strain rate (sRate) from the user.

double L0; // In microns
double sRate;
double epsilon;

std::cout << "Please enter the initial length of the fiber in microns" << "\n";

std::cin >> L0;

std::cout << "Please enter the desired strain. Zero (0) means unlimited" << "\n";

std::cin >> epsilon;

std::cout << "Please enter the desired strain rate" << "\n";

std::cin >> sRate;



L0 = L0/1000; //convert from microns to millimeters to match units for velocity.
double SysDelay = 0; //sRate; //ms
SysDelay = SysDelay / 1000; //ms converted into seconds
double sRateMax = 20;
double deltaLMin = 0.000100; // mm
double dT = 0.05; // (1 / sRate) / 100;//.001; //(1/sRateMax)∗(1 (L0/(L0+deltaLMin))); // in seconds
```

```
int tStep = 1;
double Lnew = 0;                //mm
double Vel = 0;                 //mm/s
//double Lold = L0;
double deltaL;                  // mm
double Pnew = RealPosition; // mm
double Pinit = RealPosition; // mm
double PMax = travelRangeMinimum;                   // mm
// double Pnow = Position;
double vMax = 10; // mm/s
//double Pold;
double SleepyTime = (dT+SysDelay)*1000; //in ms for Sleep function
double TravelLimitBuffer = 0.050; //millimeters
double Lold = L0;

// std::cout << ”delay time in milliseconds is ” << SleepyTime << ”\n”;

//while Pnew (target) is still in range... position and velocity loops will kick interrupt with error if out of bounds
// moving from initial position to position = travelrangeminimum.

//double tStepMax = 1000;

if (epsilon > 0)
 PMax = Pinit   (epsilon*L0);


// clock_t t;
// t = clock();

while (Vel < vMax && Pnew > PMax) //(Pnew > (travelRangeMinimum+TravelLimitBuffer))
{


 // Pold = Pnew;
 Lnew = L0*exp(sRate*(tStep*(dT))); //need to add known loop/com
```

```
    delay into exponent to make up for lag
Vel = (Lnew   Lold) / dT;//(Lold*sRate   L0*sRate + Lnew*sRate
     L0*sRate)/2; // mm/s
deltaL = Lnew   L0;
Pnew = Pinit   deltaL; //target position
Lold = Lnew;

if (Vel>vMax)
{
        std::cout << ”Warning: Velocity limit reached.
           Terminating pull sequence . . . \n”;
        break;

}

if (Pnew < PMax)
{
        std::cout << ”Warning: Positional range limit reached.
           Terminating pull sequence . . . \n”;
        break;

}

//std::cout << ”Lnew is now ” << Lnew << ”\n”;
//std::cout << ”velocity is now ” << Vel << ”\n”;
//std::cout << ”delta L is now ” << deltaL << ”\n”;
//std::cout << ”Pnew is now ” << Pnew << ”\n”;

//set new velocity

        if (!PI_VEL(piDeviceId, axis1, &Vel))
        {
                CloseConnectionWithComment(piDeviceId, ”Unable
                   to set stage velocity\n”);
                return false;
        }


        //go to new position

        if (!PI_MOV(piDeviceId, axis1, &Pnew))
        {
                CloseConnectionWithComment(piDeviceId, "MOV, unable to move to target position.\n");
                return false;
        }


/* As soon as encoder reads a value <= Pnew... (moving from positive to negative)

while (Pnow > Pnew)  //keep program stuck in this loop until stage passes target position
{
        if (!PI_qPOS(piDeviceId, axis1, &Pnow)) //re defines Pnow continuously. while loop stops once Pnow passes target mark
        {
                CloseConnectionWithComment(piDeviceId, "Unable to query stage position\n");
                return false;
        }

}
*/

////Query stage position
//double RealPosition;
//if (!PI_qPOS(piDeviceId, axis1, &RealPosition))
//{
       //  CloseConnectionWithComment(piDeviceId, "Unable to query stage position\n");
       //  return false;
//}
//  std::cout << "position is " << RealPosition << "\n";
//

////  Query stage velocity
//double RealVelocity;
//if (!PI_qVEL(piDeviceId, axis1, &RealVelocity))
```

```
    //{
            // CloseConnectionWithComment(piDeviceId, "Unable to query stage velocity\n");
            // return false;
    //}

    //std::cout << "velocity is " << RealVelocity << "\n";


    ////Pnew = Pnew; //now the current position is the destination.

    ////double SleepyTime = ((Pold   Pnew)/RealVelocity)*1000; // To convert to ms for Sleep function.


    Sleep(SleepyTime); //try to time match coms delay with curve resolution. maybe sleep time should increase as tStep*dT??

    //std::cout << "delay time of" << dT << "\n";



    tStep++;

    //std::cout << "tStep is now at" << tStep << "\n";



}
```

Step-Strain Excitation

```
double L0;
double Pnew;
double Initial_Position;
double stress;
double max_strain;
double k_pip;
double f_diam;
double delta_L;
int delta_t;
int k; //position step integer
double disp_total;
double area;
double delta_L_max;
int time_array[10] = {50, 40, 30, 20, 10, 5, 4, 3, 2, 1 }; // seconds
//int max_time = time_array[0] * 1000; //milliseconds
int relax_time;


while (1)

{


std::cout << "Press any key to continue \n";
std::cin.get();


// Prompt User for initial setup parameters
Initial_Position = Position;

std::cout << "Enter initial fiber diameter in microns \n";

std::cin >> f_diam;

std::cout << "Enter initial fiber length in microns \n";

std::cin >> L0;

std::cout << "Enter desired value of maximum strain \n";

std::cin >> max_strain;

std::cout << "Enter desired value of step stress in Pa \n";

std::cin >> stress;

std::cout << "Enter pipette spring constant in nN/um \n";

```

```
std::cin >> k_pip;

area = PI*pow((f_diam / 2), 2);
delta_L = (area*stress / k_pip) / 1000000; // converted from nanometers to millimeters
delta_L_max = L0*max_strain / 1000; //converted from microns to millimeters

std::cout << "Press any key to begin experiment \n";
std::cin.get();


for (int n = 0; n < 10; n++) //run through time_array values
{

        k = 1;
        delta_t = time_array[n] * 1000; //convert to milliseconds

        std::cout << "delta L " << delta_L*1000 << " microns \n";
        std::cout << "delta L max is " << delta_L_max*1000 << " microns \n";
        std::cout << "relaxation time is " << time_array[n] << " seconds \n";

        while (k*delta_L < delta_L_max)
        {

                Pnew = Initial_Position   k*delta_L;

                // Move pipette for step strain

                //go to new position

                if (!PI_MOV(piDeviceId, axis1, &Pnew))
                {
                        CloseConnectionWithComment(piDeviceId, "MOV, unable to move to target position.\n");
```

```
            return false;
        }

        // wait for motion to stop
        BOOL isMoving = true;
        while (isMoving)
        {
            if (!PI_IsMoving(piDeviceId, axis1, &isMoving))
            {
                CloseConnectionWithComment(piDeviceId, "Unable to query movement status.\n");
                return false;
            }
            Sleep(100);
        }

        // Query stage position
        double Position;
        if (!PI_qPOS(piDeviceId, axis1, &Position))
        {
            CloseConnectionWithComment(piDeviceId, "Unable to query stage position\n");
            return false;
        }

        disp_total = abs(Position   Initial_Position) * 1000; //convert from millimeters to microns

        std::cout << "Total distance travelled is " << disp_total << " microns \n";

        Sleep(delta_t); //in ms

        k++;

    }


    relax_time = k*delta_t;

    //go back to initial position

    if (!PI_MOV(piDeviceId, axis1, &Initial_Position))
    {
        CloseConnectionWithComment(piDeviceId, ”MOV,
            unable to move to target position.\n”);
        return false;
    }

    // wait for motion to stop
    BOOL isMoving = true;
    while (isMoving)
    {
        if (!PI_IsMoving(piDeviceId, axis1, &isMoving))
        {
            CloseConnectionWithComment(piDeviceId, ”
                Unable to query movement status.\n”);
            return false;
        }
        Sleep(100);
    }

    // Query stage position
    double Position;
    if (!PI_qPOS(piDeviceId, axis1, &Position))
    {
        CloseConnectionWithComment(piDeviceId, ”Unable
            to query stage position\n”);
        return false;
    }

    std::cout << ”Stage returned to initial position of ” <<
        Initial_Position << ” \n”;

    Sleep(relax_time); //in ms



```

```
}


std::cout << "Experiment concluded   \n";

}
```

Triangular Excitation

```
for (int n=0; n < NCycles; n++)

        {

                int tStep = 0;
                Vel = 0;

                while (Vel < vMax && Pnew > PMax) //(Pnew > (
                    travelRangeMinimum+TravelLimitBuffer))
                {


                        // Pold = Pnew;
                        Lnew = L0*exp(sRate*(tStep*(dT))); //
                            need to add known loop/com delay
                            into exponent to make up for lag
                        Vel = (Lnew   Lold) / dT; // mm/s
                        deltaL = Lnew   L0;
                        Pnew = Pinit   deltaL; //target
                            position


                        tStep++;

                        if (Pnew < PMax)
                        {
                                tStep  ;
                                break;
                        }
                        if (Vel > vMax)
                        {
                                tStep  ;
                    break;
                }


                Lold = Lnew; //allows Lold to be the
                    OLD Lold if the break is excecuted

                //set new velocity

                if (!PI_VEL(piDeviceId, axis1, &Vel))
                {
                    CloseConnectionWithComment(
                        piDeviceId, ”Unable to set
                        stage velocity\n”);
                    return false;
                }


                //go to new position

                if (!PI_MOV(piDeviceId, axis1, &Pnew))
                {
                    CloseConnectionWithComment(
                        piDeviceId, ”MOV, unable to
                        move to target position.\n”)
                        ;
                    return false;
                }


                Sleep(SleepyTime); //try to time match
                    coms delay with curve resolution.
                    maybe sleep time should increase as
                    tStep∗dT??


            }


            tStep  ;
```

```
        double b = L0;
        double slope = (Lnew  L0)/((tStep+2)*dT);

        Vel = 0;
        //std::cout << Lold << ”\n”;
        while (Vel < vMax && Pnew < Pinit)
        {

            // Pold = Pnew;
            Lnew = (slope/sRate)*log(((tStep)*dT*slope+b)/L0)+b; //since srate is now negative, becomes decaying exponential.
            Vel = (Lold  Lnew) / dT; // mm/s need minus sign now that direction is reversed to keep velocity positive
            deltaL = Lnew  L0; //distance from L0 and decreases backwards in time
            Pnew = Pinit  deltaL; //Now, Pnew is approaching Pinit
            Lold = Lnew;
            //std::cout << Lnew << ”\n”;

            if (Pnew > Pinit)
            {
                tStep++;
                break;
            }
            if (Vel > vMax)
            {
                tStep++;
                break;
            }



            //set new velocity
            //std::cout << Vel << ”\n”;
            if (!PI_VEL(piDeviceId, axis1, &Vel))
            {
                CloseConnectionWithComment(piDeviceId, "Unable to set stage velocity\n");

                return false;
            }


            //go to new position

            if (!PI_MOV(piDeviceId, axis1, &Pnew))
            {
                CloseConnectionWithComment(piDeviceId, "MOV, unable to move to target position.\n");
                return false;
            }


            Sleep(SleepyTime); //try to time match coms delay with curve resolution. maybe sleep time should increase as tStep*dT??


            tStep  ;


        }



    }
```

Small Amplitude Oscillation (SAOS)

Multiple algorithms were used for sinusoidal behavior of the PI stage depending on the frequency and amplitude range desired. All three algorithms are presented below.

Taylor Expansion

```
int TaylorSin(int piDeviceId, const char* axis1,double Pinit, int NCycles, double omega, double pi, double a1, double a2, double a3, double a4, double a5, double AmpL, double PMax, double SysDelay) {

std::cout << "running taylor expanded sine approximation \n";

int tStep ;
double Vel = 0;
double Pnew = 0;

double deltaL;
double Pold;
//double PtsPerCycle = 20;
double dT = 0.250; // in seconds
double omega_dt;
double SleepyTime = (dT   SysDelay) * 1000; //seconds to ms for sleep function

if (SleepyTime < 0)
        SleepyTime = 0;


for (int n = 0; n < NCycles; n++) //Run loop for the pre defined number of cycles

{


        tStep = 1;
        omega_dt = omega*tStep*dT;
        Pold = Pinit;



        ///////////////////////////////////////////////
        //Now run the code for the sinusoidal movement. Taylor expansions for each quadrant...
        ///////////////////////////////////////////////
```

```

while (omega_dt < a2) //while in the first quadrant..
{

Pnew = Pinit + AmpL*((omega_dt - a1) - pow(omega_dt - a1, 3) / 6 + pow(omega_dt - a1, 5) / 120); //need to add known loop/com delay into exponent to make up for lag

deltaL = Pnew - Pold; //delta L per step

Vel = abs(deltaL / dT); // mm/s

tStep++;

omega_dt = omega*tStep*dT;

Pold = Pnew;

//set new velocity

if (!PI_VEL(piDeviceId, axis1, &Vel))
{
        CloseConnectionWithComment(piDeviceId, ”Unable to set stage velocity\n”);
        return false;
}

//go to new position

if (!PI_MOV(piDeviceId, axis1, &Pnew))
{
        CloseConnectionWithComment(piDeviceId, ”MOV, unable to move to target position.\n”);
        return false;
}


        Sleep(SleepyTime); //try to time match coms delay with curve resolution. maybe sleep time should increase as tStep*dT??
```

```

}

        while (omega_dt < a3)
{

Pnew = Pinit + AmpL*(1    pow(omega_dt    a2, 2) / 2 + pow(omega_dt    a2, 4) / 24); //need to add known loop/com delay into exponent to make up for lag



deltaL = Pnew    Pold;
Vel = abs(deltaL / dT); // mm/s

                                                            //std::cout << ” Vel =” << Vel << ”\n”;
```

```


Pold = Pnew; //allows Lold to be the OLD Lold if the break is excecuted

                                //set new velocity

if (!PI_VEL(piDeviceId, axis1, &Vel))
{
        CloseConnectionWithComment(piDeviceId, ”Unable to set stage velocity\n”);
        return false;
}


//go to new position

if (!PI_MOV(piDeviceId, axis1, &Pnew))
{
        CloseConnectionWithComment(piDeviceId, ”MOV, unable to move to target position.\n”);
        return false;
}


        Sleep(SleepyTime); //try to time match coms delay with curve resolution. maybe sleep time should increase as tStep*dT??

}

        while (omega_dt < a4)
{

Pnew = Pinit + AmpL*( 1 * (omega_dt   a3) + pow(omega_dt   a3, 3) / 6   pow(omega_dt   a3, 5) / 120); //need to add known loop/com delay into exponent to make up for lag


```

```

deltaL = Pnew    Pold;
Vel = abs(deltaL / dT); // mm/s

                                                            // std::cout << "Vel =" << Vel << "\n";

Pold = Pnew; //allows Lold to be the OLD Lold if the break is excecuted

                          //set new velocity

if (!PI_VEL(piDeviceId, axis1, &Vel))
{
        CloseConnectionWithComment(piDeviceId, "Unable to set stage velocity\n");
        return false;
}
```

```


//go to new position

if (!PI_MOV(piDeviceId, axis1, &Pnew))
{
        CloseConnectionWithComment(piDeviceId, ”MOV, unable to move to target position.\n”);
        return false;
}


        Sleep(SleepyTime); //try to time match coms delay with curve resolution. maybe sleep time should increase as tStep*dT??

}

        while (omega_dt < a5)
{

Pnew = AmpL*( 1 + pow(omega_dt   a4, 2) / 2   pow(omega_dt   a4, 4) / 24); //need to add known loop/com delay into exponent to make up for lag


deltaL = Pnew   Pold;
Vel = abs(deltaL / dT); // mm/s


Pold = Pnew;

//set new velocity

if (!PI_VEL(piDeviceId, axis1, &Vel))
{
        CloseConnectionWithComment(piDeviceId, ”Unable to set stage velocity\n”);
        return false;
}
```

```


//go to new position

if (!PI_MOV(piDeviceId, axis1, &Pnew))
{
        CloseConnectionWithComment(piDeviceId, "MOV, unable to move to target position.\n");
        return false;
}


Sleep(SleepyTime); //try to time match coms delay with curve resolution. maybe sleep time should increase as tStep*dT??

}


}


}
```

C++ Sin Function

```
int SinOsc(int piDeviceId, const char* axis1, double Pinit, int NCycles, double omega, double pi, double a1, double a2, double a3, double a4, double a5, double AmpL, double PMax, double SysDelay) {

std::cout << "running oscillations base on normal sine function output \n";

int tStep;
double Vel = 0;
double Pnew = 0;

double deltaL;
double Pold;
double dT_Err = 0.05; //seconds
double min_step_length = 4 * pow(10, 9); //minimum step size in
```

```
meters
double dT = min_step_length*2*pi/(AmpL*pow(10, 3)*omega) + dT_Err; // in seconds
double omega_dt;
double SleepyTime = (dT   SysDelay) * 1000; //seconds to ms for sleep function

if (SleepyTime < 0)
SleepyTime = 0;


for (int n = 0; n < NCycles; n++) //Run loop for the pre defined number of cycles

{


tStep = 1;
omega_dt = omega*tStep*dT;
Pold = Pinit;


// ///////////////////////////////////////////////////////////////
// Now run the code for the sinusoidal movement. just sine function from math library
// ///////////////////////////////////////////////////////////////

while (omega_dt < a5)
{


Pnew = Pinit + AmpL*sin(omega_dt);
deltaL = Pnew   Pold; //delta L of each step
Vel = abs(deltaL / dT);

Pold = Pnew;

tStep++;
omega_dt = omega*tStep*dT;


//set new velocity

if (!PI_VEL(piDeviceId, axis1, &Vel))
{
CloseConnectionWithComment(piDeviceId, ”Unable to set stage velocity\n”);
return false;
}


//go to new position

if (!PI_MOV(piDeviceId, axis1, &Pnew))
{
CloseConnectionWithComment(piDeviceId, ”MOV, unable to move to target position.\n”);
return false;
}


Sleep(SleepyTime); //try to time match coms delay with curve resolution. maybe sleep time should increase as tStep*dT??


}


}

}
```

Triangular Approximation

```
int TriSin(int piDeviceId, const char* axis1, double Pinit, double AmpL, double AmpLo2, int NCycles, double V1, double V2, double SysDelay, double SleepyTime1, double SleepyTime2, double SleepyTime3, double SleepyTime1t2) {


std::cout << ”running modified triangle/sine approximation \n”;
```

```

double Pnew;
double Vel;


for (int n = 0; n < NCycles; n++) //Run loop for the pre defined number of cycles

{

for (int j = 0; j < 2; j++)
{
//1 of 4
Pnew = Pinit + pow( 1 , j)*AmpLo2; //need to add known loop/com delay into exponent to make up for lag
Vel = V1;

//set velocity
if (!PI_VEL(piDeviceId , axis1 , &Vel))
{
        CloseConnectionWithComment(piDeviceId , ”Unable to set stage velocity\n”);
        return false;
}

//go to new position
if (!PI_MOV(piDeviceId , axis1 , &Pnew))
{
        CloseConnectionWithComment(piDeviceId , ”MOV, unable to move to target position.\n”);
        return false;
}

if (n == 0 && j == 0)
        Sleep(SleepyTime1);
else
        Sleep(SleepyTime1t2); //sleep the loop for the time it should take for the stage to reach the new position to avoid forward calculating. add system delay if needed
```

```

                                                                //2 of 4
Pnew = Pinit + pow( 1, j)*AmpL; //need to add known loop/com delay into exponent to make up for lag
Vel = V2;

//set velocity
if (!PI_VEL(piDeviceId, axis1, &Vel))
{
        CloseConnectionWithComment(piDeviceId, ”Unable to set stage velocity\n”);
        return false;
}

//go to new position
if (!PI_MOV(piDeviceId, axis1, &Pnew))
{
        CloseConnectionWithComment(piDeviceId, ”MOV, unable to move to target position.\n”);
        return false;
}

Sleep(SleepyTime2); //sleep the loop for the time it should take for the stage to reach the new position to avoid forward calculating. add system delay if needed

                                                //3 of 4

Sleep(SleepyTime3); //flat part of tri wave

                                                //4 of 4
Pnew = Pinit + pow( 1, j)*AmpLo2; //need to add known loop/com delay into exponent to make up for lag
Vel = V2;

//set velocity
if (!PI_VEL(piDeviceId, axis1, &Vel))
{
        CloseConnectionWithComment(piDeviceId, ”Unable to set stage velocity\n”);
```

```
    return false;
}

//go to new position
if (!PI_MOV(piDeviceId, axis1, &Pnew))
{
    CloseConnectionWithComment(piDeviceId, ”MOV, unable to move to target position.\n”);
    return false;
}

Sleep(SleepyTime2); //sleep the loop for the time it should take for the stage to reach the new position to avoid forward calculating. add system delay if needed

}

}

}
```

APPENDIX B

HARDWARE AND PARTS SPECIFICATIONS

B.1 Optics Components

Optical Microscope Condenser			
Description	Manufacturer	Part Number	Purpose
Aspherical Condenser Lens	Thorlabs	ACL25416U	Light collimation
Mounted Achromatic Doublet	Thorlabs	AC254-050-A-ML	Beam reduction
Mounted Biconcave Lens	Thorlabs	LD1464-ML	Beam expansion

Table B.1: Optics Components

B.2 Heater Components

Heat Stage			
Description	Manufacturer	Part Number	Purpose
PID Controller	Omega Engineering	CN16Pt	Temperature control and data acquisition
Solid State Relay	Omega Engineering	SSRL240DC100	Current control for cartridge heaters
T-type Thermocouple	Omega Engineering	CO1-T	Temperature sensing

Table B.2: Heater Components

B.3 Positional Stage Components

Positional Stages			
Description	Manufacturer	Part Number	Purpose
Pipette Holder	Narishige International Ltd.	UPN-C	Fixing the micropipettes to the positional stages
Arduino Uno R3	Arduino		Hardware interface for joystick control
Arduino Motor-Shield Rev3	Arduino		Motor control interface or the Hardware controller
DC Motor	Conrad Electronics		Motorization of translational stages
Piezo Linear Stage	PI		To provide precise sample deformation for rheological testing
3-Axis Joystick	Conrad Electronics		Control of the 3-axis translational stage

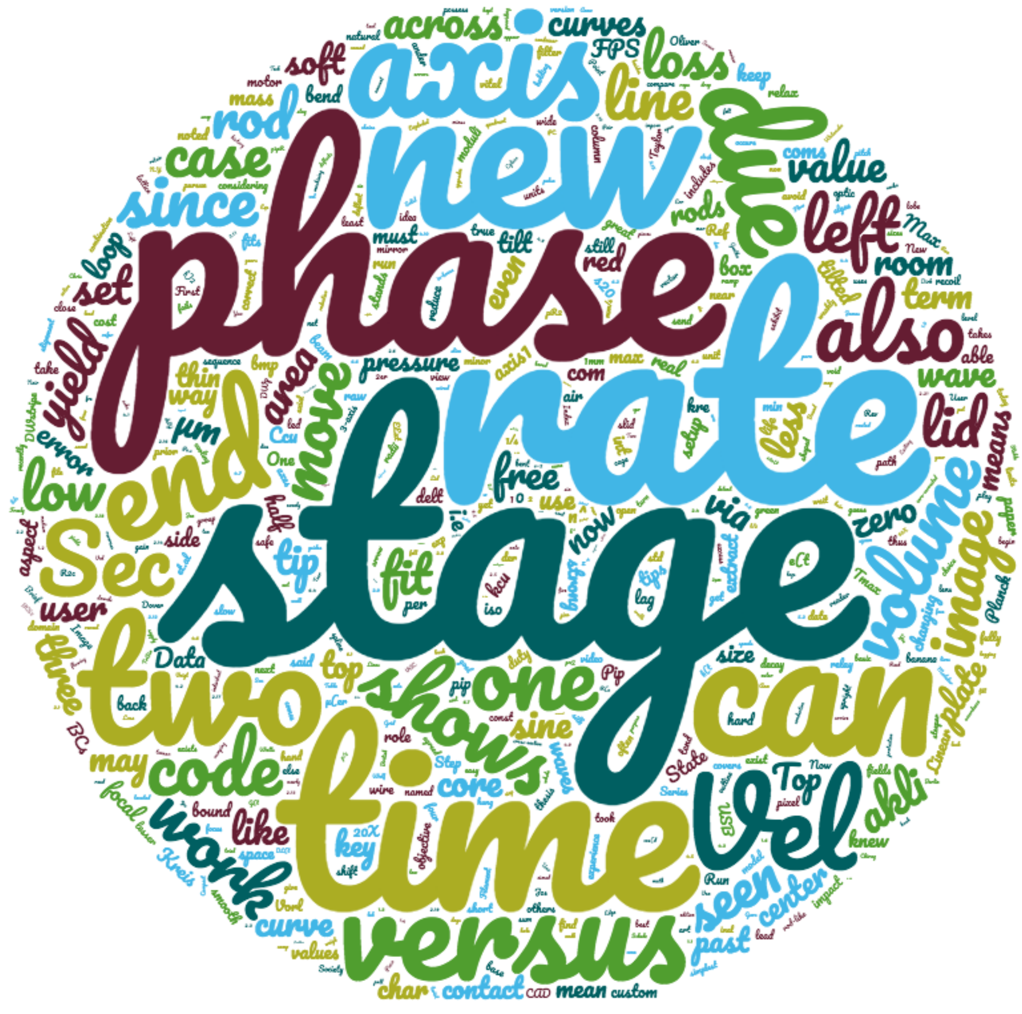
Table B.3: Positional Stage Components

B.4 Image Acquisition

Digital Image Recording			
Description	Manufacturer	Part Number	Purpose
4MP USB3 Camera	Point Grey Research*	GS3-U3-32S4M-C	Image recording

Table B.4: Image Acquisition Components

*Point Grey Research has since been acquired by FLIR


stage
rate
phase
time
new
two
can
versus
axis
end
due
vel
shows
one
code
also
Sec
work
image
volume
left
move
since
case
rod
loss
line
across
curves
value
yield
set
seen
area
less
akli
plate
wave
lid
means
low
tip
fit
three
user
like
may
top
sine
core
waves
center
past
curve
char
contact
mean
key
room
term
tilted
even
tilt
free
via
zero
now
use
error
µm
thin
way
pressure
soft
FPS
keep
rods
box
red
loop
must
com
max
kre
setup
pip
size
Data
focal
Kreis
smooth
space
bound
Top
Now
Linear
knew
State
BSN
Planck
Tmax
lag
lips
extract
changing
kcu
iso
per
Step
role
wire
objective
Vorl
values
custom
CAD
mass
bend
motor
natural
Oliver
modul
column
Taylor
coms
considering
noted
true
great
still
mirror
idea
Ref
Max
includes
optic
sequence
bmp
take
cost
close
First
air
real
unit
able
takes
min
slid
raw
Ccu
prior
One
half
side
aspect
safe
slow
lib
reduce
1mm
view
Dover
domain
Society
gain
shift
others
find
impact
Series
pixel
took
Run
hard
decay
relay
banana
paper
path
green
BCs
back
Pip
video
duty
said
next
fields
covers
axist
const
relax
recoil
send
filter
version
compare
3-axis
20X
10
i.e.
delt
ramp
near
lobe
pitch
avoid
occurs
wide
thesis
experience
short
Jcs
axis1
cage
guess
thus
begin
bend
lead
best
mode
Rev
User
level
New
sizes
dead

Oliver H. Kress

Private • Kramgasse 52 • 3011 Bern • Switzerland
+41 (0)76 377 20 76 (mobile)

Work • University of Bern • Fabrikstrasse 8 • 3012 Bern • Switzerland
oliver.kress@ikim.unibe.ch • +41 (0)31 631 8139 (work)

Nationalities • United States, Germany

Language Fluency • English, German

Employment History

June 2018 - Present
Postdoctoral Researcher at the Institute of Complementary and Integrative Medicine, University of Bern, Bern, Switzerland

Responsibilities: Leading a cross-institute, interdisciplinary project funded by an SNF Bridge Discovery grant. From research to final product, the project aims to design, build and market soft, wearable textile sensors using near-infrared spectroscopy to measure tissue oxygenation and pressure. The goal is to help limited mobility hospital patients and individuals such as paraplegics to avoid pressure ulcers by providing real-time blood oxygenation monitoring of known problem area such s the lower back, thighs and feet.

Nov. 2018 – June 2019
Team Leader - Systems Optimization at SWD AG Stator- und Rotortechnik, Densbüren, Switzerland

Responsibilities: Leading a team of operators in maintaining and optimizing a production line while developing new manufacturing techniques for thin film polymer adhesives for the application of stacked, laminated stators and rotors for electric motors in the automotive industry with emphasis on quality control and mass production.

Sept. 2015 - Oct. 2018 Doctoral Researcher at the Max Plank Institute for Dynamics and Self-Organization, Göttingen, Germany

Responsibilities: Designing and executing a three-year research project in the scope of a PhD in soft matter/liquid crystal physics. Design, optimization and manufacturing of experimental devices and setups including hardware/software interfacing, image analysis and closed-loop control for nano-scale dynamic stress and strain measurements for the viscoelastic analysis of active fibers formed by exotic liquid crystal phases. The resulting device, a nano-scale rheometer, recently received an EU patent while a US patent is pending. Teaching responsibilities were satisfied by graduate level recitation lectures in the topics of climate physics and wave mechanics.

April 2014 - June 2015 Graduate Student Researcher at the Liquid Crystal Institute at Kent State University, United States

Responsibilities: Conducting master's level research as a part of JákliLab, an active materials research lab run by Dr. Antal Jákli (http://jakligroup.com). Research topics have mainly included the investigation of liquid crystal fibers drawn from bent-core mesogens and the design and assembly of instrumentation and measurement devices for their properties. Working in JákliLab also introduces shared topics in bio-sensors and microfluidics.

June 2014 - July 2014 IRES (NSF) Research Intern at O.v.G. Universität in Magdeburg, Germany

Responsibilities: Five week, high-paced internship working on liquid crystal fiber research in the Abteilung Nichtlinieare Phänomene (non-linear phenomena lab) of Prof. Ralf Stannarius (http://www.uni-magdeburg.de/anp/) studying the temperature dependence of liquid crystal fibers both with and without the inclusion of carbon nanotubes.

2012 – 2013 Private Math and Physics Tutoring at the University Level

Responsibilities: 2 semesters of private tutoring of university students on topics in math and physics ranging from basic calculus and trigonometry to advanced undergraduate math and physics such as quantum mechanics, partial differential equations and complex analysis.

May 2011 – Aug. 2011 SULI (Department of Energy) Research Intern at the Thomas Jefferson National Accelerator Facility (JLab)

Responsibilities: Ten weeks of working with an experimental particle physics group on a project called the Heavy Photon Search (HPS) by building, calibrating and running a particle detector consisting of a PbWO4 (lead tungstate) scintillating crystals and APDs. This work was a necessary part of a facility-wide upgrade needed to improve performance and prepare data acquisition for the upcoming upgrade from a 6 GeV to a 12 GeV main beam line.

2010 - 2011 Math and Physics Tutor at Kent State University

Responsibilities: 2 semesters of group and private tutoring of university students in intermediate topics in math and physics.

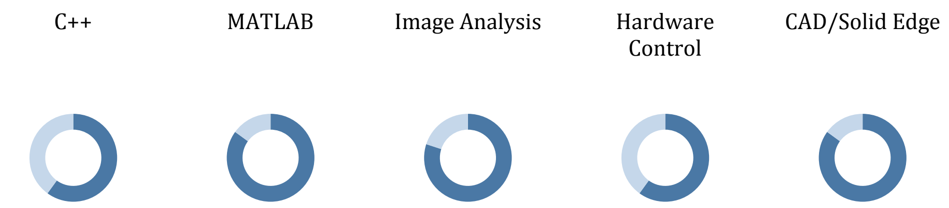
2005 - 2008 Automotive Repair Technician

Responsibilities: As a full-time mechanic, I performed all levels of vehicle diagnostics and repair on a wide range of early- and late-model vehicles. This position helped me to acquire and fine-tune the technical and mechanical intuition. Hands-on experience with a full range of modern engine and vehicle control sensors and systems and has strongly contributed to my abilities in the lab to diagnose, design and build instrumentation and develop sensor and measurement technology during my later research.

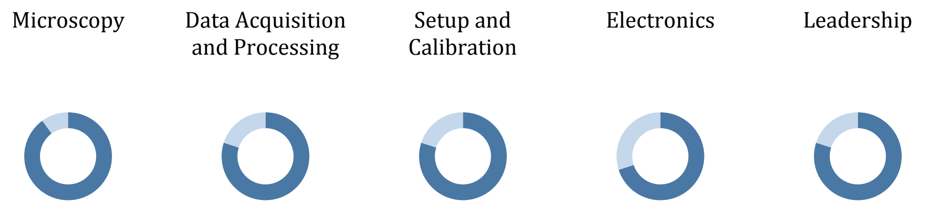
Education

2015 - 2018	University of Göttingen / Max Planck Institute (Germany) *Physics - PhD*
2013 - 2015	Liquid Crystal Institute at Kent State University (Ohio, USA) *Chemical Physics – MSc*
2008 - 2012	Kent State University (Ohio, USA) *Physics - BSc* *Mathematics - Minor*
2001 - 2005	Kent Roosevelt High School (Ohio, USA)

Programming Skills



Laboratory Experience



Publications

O. Kress, inventor; M. Makowski, T Ostapenko; coinventors. Nano-Extensional Rheometer. EU patent publication EP 3 431 960. *US patent pending US2019017931.*

S.M. Salili, T. Ostapenko, O. Kress, *et al.,* Rupture and Recoil of Bent-core Liquid Crystal Filaments, Soft Matter, 2016,12, 4725-4730

In Progress (pending approval from patent office):

O. Kress *et al.,* Structure and Elasticity of Single Filaments of Bent-core liquid crystals
O. Kress *et al.,* Viscoelastic Behavior of Bent-core liquid crystal fibers
O. Kress *et al.,* Extensional Rheology of Bent-core Smectic Fibers

Conference Contributions

O. Kress *et al.,* Extensional Rheology of Liquid Crystal Fibers, **Poster**, 10th Liquid Matter Conference, 2017, Ljubljana, Slovenia

Conference **Organizer and Editor** of the Third Infinity 2017 International Conference on the Physics of Biological and Complex Systems in Göttingen, Germany (http://www.thirdinfinity.mpg.de/)

O. Kress *et al.,* Tension and Elasticity of Smectic Liquid Crystal Fibers, **Poster**, International Liquid Crystal Conference, 2016, Kent, USA

O. Kress *et al.,* Tension of Self-assembled Bent-core Liquid Crystal Filaments with and without Carbon Nanotubes, **Talk**, Ferroelectric Liquid Crystal Conference, 2015, Prague, Czech Republik

O. Kress *et al.,* Bent-core Liquid Crystal Helical Filaments: Tension, Recoil and Rupture Behavior, **Talk**, American Physical Society Conference, 2015, San Antonio, USA

Hobbies + Personal Interests

Vehicle restoration, carpentry, travel, skiing and snowboarding, climbing, swimming, hiking, soccer, marksmanship, music, photography, technology, programming, maker.