2014

Shear Waves in Highly Concentrated Viscoelastic Wormlike Micellar Fluid

Rachel B. Crim
University of Mississippi. Sally McDonnell Barksdale Honors College

Shear Waves in Highly Concentrated Viscoelastic Wormlike Micellar Fluid

by
Rachel Barton Crim

A thesis submitted to the faculty of The University of Mississippi in partial fulfillments of the requirements of the Sally McDonnell Barksdale Honors College.

Oxford, MS
May 2014

Approved by

Advisor: Dr. Joseph Gladden

Reader: Dr. Joel Mobley

Reader: Dr. Lucien Cremaldi

ACKNOWLEDGEMENTS

I would like to give a special thank you to Dr. Josh Gladden, who advised me throughout the entire research and writing processes. Additionally, he wrote all of the Python codes used to analyze the data I gathered. None of this would have been possible without him. Qin Zhang also spent a good deal of time with me, helping me figure out how to build these enormous apparatuses. I am very grateful for both her and Dr. Gladden's patience with me. I would also like to thank my readers Dr. Joel Mobley and Dr. Lucien Cremaldi for taking time out of their busy schedules to critique my work. Finally, I am greatly indebted to the Sally McDonnell Barksdale Honors College for funding this research.

ABSTRACT

Wormlike micellar fluids form by the self-aggregation of surfactant molecules in aqueous solution. These non-Newtonian fluids have been well studied and are used in the oil industry, hydraulics, and medical research. However, little is known regarding the structure of the three-dimensional networks in which the "worms" become entangled and possibly branched, especially at high concentrations. What is known is that this composition results in two distinctive fluid characteristics: viscoelasticity and strain-birefringence. The latter is exploited in this work in order to study the shear wave speed and attenuation in 500/300mM CTAB/NaSal fluid. Three different experiments were conducted using either a laser/diode system or a camera/backlight system. The average speed of a shear wave in wormlike micellar fluid was determined to be 63.47cm/s for this concentration at room temperature with no consistent effect from aging. Temperature, however, had a significant impact. Around 35^{o}C, there was a dramatic drop in shear speed. The steady linear decrease in micelle length with increasing temperature does not account for this steep decline. A possible explanation is that the micelle conformation changes at this point—a topological phase transition. It is recommended that the rheology around this temperature be examined.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

CTAB – Cetyltrimethylammonium bromide

NaSal – Sodium salicylate

v – velocity

ƒ - frequency

λ – wavelength

C_s - speed of sound

G – shear modulus

ρ – mass density

INTRODUCTION

Wormlike micellar fluids result from amphiphilic surfactant molecules, meeting a minimum critical micellar concentration (CMC), being placed in an aqueous salt solution. The polarized surfactant molecules (Figure 1) self-aggregate into a conformation that minimizes Gibbs free energy. They will spontaneously arrange in a manner that exposes their hydrophilic head groups and shields the hydrophobic "tails" from the aqueous solution. This mechanism is most commonly associated with membranous phospholipid bilayers.

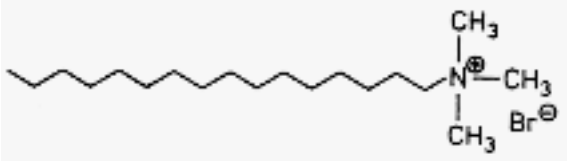

Figure 1: One molecule of CTAB, the surfactant used in this experiment. Seen is the polar head group and the nonpolar hydrocarbon chain.

The micelles that form at this point may take on many shapes, the simplest being spherical.[1] The salt ions act to mitigate the electrostatic repulsion between the polar head groups, screening the effect the charges have on one another. If the concentration of the counterion solvent is increased, the surfactant molecules will rearrange into a cylindrical tube that is usually several microns in length and roughly 15nm in diameter[2] (see Fig. 2). It should be noted that this geometry might vary depending on the surfactant itself, the aqueous solvent, temperature, and pressure.[1] This experiment will only vary temperature, and use a surfactant/solvent combination that is thermodynamically preferable in a cylindrical configuration.

The cetyl trimethylammonium bromide (CTAB) and sodium salicylate (NaSal) aqueous solution has been well-studied as a wormlike micellar fluid.[5] The CTAB acts as the long-tailed cationic surfactant while NaSal performs the role of counterion.

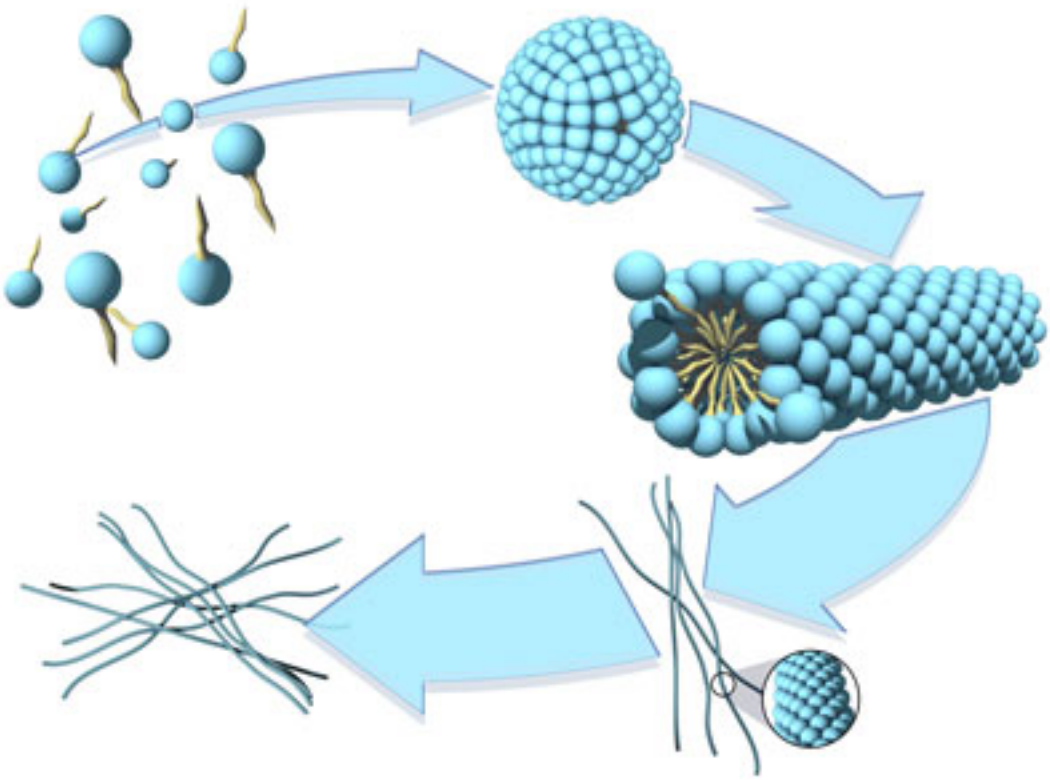

Figure 2: The conformation of surfactant molecules that arrange to form wormlike micelles. Additional salt ions screen the charged heads of the surfactant molecule from one another, leading to a cylindrical structure. (Image: Prof. Bjorn Lindman, University of Lund, Sweden).

So these wormlike micelles end up as long chains that may become entangled with one another in three-dimensional networks, much like polymers.[3] Pronounced viscoelasticity is a consequence of this complex rheology. Fluids deemed viscoelastic display both viscous and elastic properties when undergoing deformation. A fluid's viscosity is a measure of its resistance to deformation by stress. Elasticity describes

the tendency of a material to return to its original conformation after stress is removed. It is widely accepted that the viscoelasticity of wormlike micellar fluids is characterized by a single relaxation time.[4] They are thixotropic, linearly viscoelastic, and they display shear-thinning under rapid, steady shearing.[1] Another characteristic consequence of the wormlike micellar fluid's structure is its strain-birefringence. The randomly intertwined tangles of worms are optically isotropic in an equilibrium state. However, when sheared, the worms disengage and partially align with the direction of flow.[6] Thus, stress fields are able to be visually observed when the sample is placed between crossed polarizing filters. It is this property in particular that was utilized to measure the speed of shear waves in the fluid. The overall goal of this work was to observe speed and attenuation of shear waves in wormlike micellar fluid. A work published in 2012 also exploited the fluid's optical properties in order to study shear wave speeds.[7] Observation of shear wave speed in 20-500mM fluid indicated that three distinct scalings are present for different concentration ranges: square root at low concentrations, linear at medium concentrations, and linear with a steeper slope at high concentrations[8] (see Fig. 3):

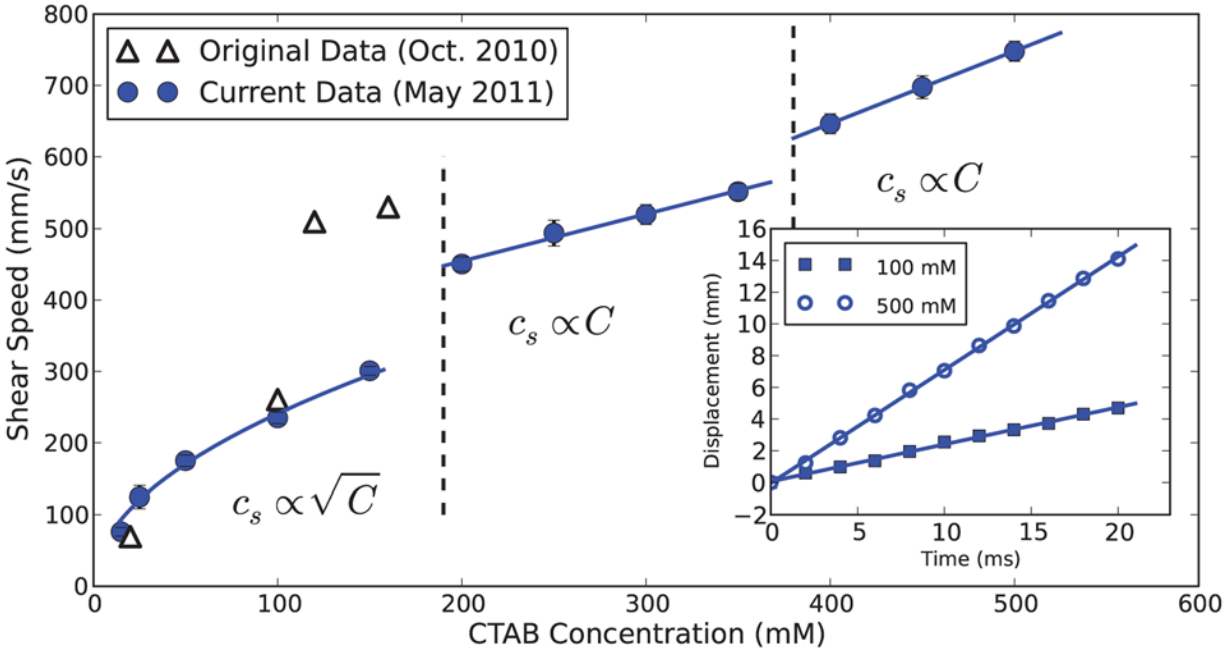

Figure 3: Data published in 2012 by Gladden, Mobley, Skelton, and Gamble. Shear speed changes with surfactant concentration, as does the pattern in which it changes.

No efforts were made to regulate temperature. In this work however, three different experiments were conducted with an emphasis on temperature, using only a 500/300mM CTAB/NaSal fluid (from this point on, simply referred to by its CTAB concentration, 500mM). The goal of Experiment 1 was to determine shear speed's dependence on temperature using a camera/backlight setup, while Experiment 2 sought to observe the speed of shear waves in room temperature fluid over an extended length of time, in order to study the effects of aging. Here, a laser/diode system was used as opposed to the camera/backlight system. Experiment 3 also used the laser/diode setup, but to study the effects of decreasing temperature on the speed of shear waves in the fluid.

EXPERIMENT 1: SPEED OF SHEAR WAVES AS A FUNCTION OF TEMPERATURE IN WORMLIKE MICELLAR FLUID

Introduction

The effect of temperature on the velocity of shear waves in wormlike micellar fluid was studied in what shall be referred to as Experiment 1 (see Fig. 4 for apparatus). It is generally accepted that micelle length shortens with increasing temperature at low concentrations[3], but little has been studied regarding the three-dimensional structure and temperature's impact on it at high concentrations (100-800 mM). Sound propagates more quickly through stiffer mediums:

$$C_s = \sqrt{(G / \rho)}$$

Dropping temperatures increase the stiffness of a wormlike micellar fluid, and the speed of sound increases.[2] Experiment 1 looks at what this relationship translates to at a high CTAB concentration. The stress field of a steady shear wave train in 500mM micellar fluid will be photographed as it cools from 70^{o}C to room temperature.

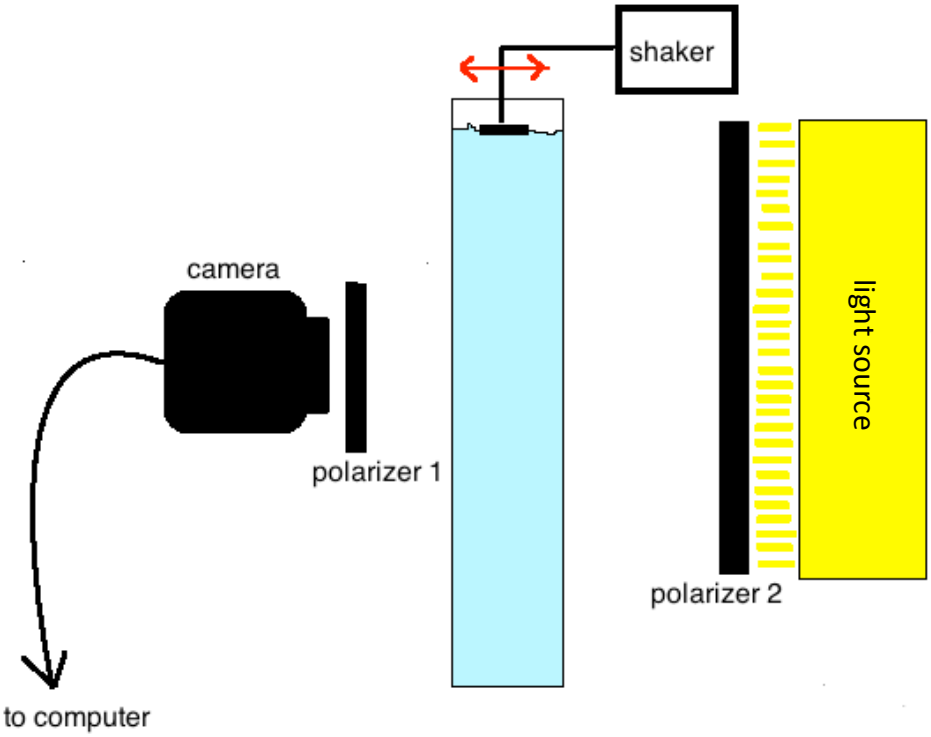

Figure 4: Schematic of Experiment 1 setup.

Methods

The fluid was prepared by combining 200.42g of cetyl trimethylammonium bromide (CTAB) with 52.84g of sodium salicylate (NaSal) in 1100mL of HPLC grade water. These values were calculated based on a 5:3 CTAB:NaSal ratio that had previously been determined as ideal for the formation of long micellar tubes. These masses were measured out and poured into two separate clean beakers, each containing half the 1100mL of water and heated to 60°C. Stirring continued for approximately one hour in both beakers until all particulate matter was dissolved. Next, the mixtures were combined, covered, and heated between 60-70°C with

intermittent stirring for five hours. The resulting 500mM mixture was poured into a McMaster-Carr glass tube that was 18" in length and had a 2" inner diameter. McMaster-Carr also provided a cap for the tube and a clamp to seal the attachment. The fluid was heated to 70°C by wrapping the glass cylinder in heater tape, then allowed to cool to room temperature by natural cooling (see Fig. 5).

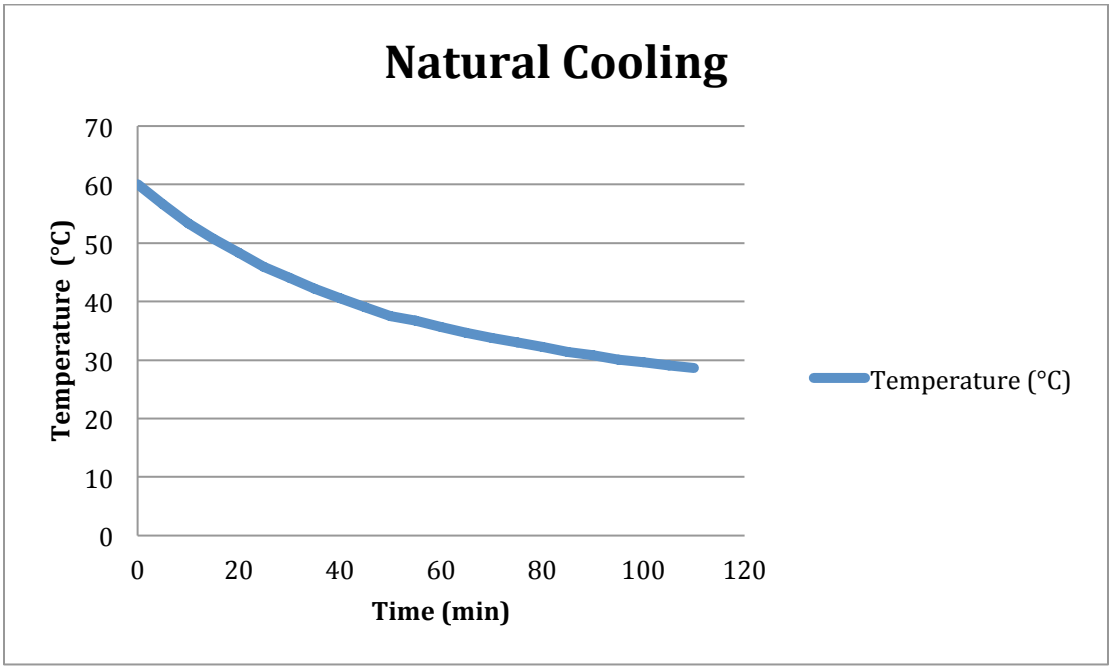

Figure 5: Temperature vs. time of a naturally cooling 500mM CTAB wormlike micellar fluid. As time passes, the fluid cools less rapidly.

As it was cooling, a photograph was taken every one minute, and the temperature of the fluid recorded using an Omega Engineering Type E thermocouple that was kept in place using electrical tape in order to prevent interference with the shearing plate. What the photograph shows is shear waves resulting from a shaker set atop the surface of the fluid, continuously shearing either at a frequency/amplitude of 50Hz/10Vpp, 30Hz/7Vpp, or 30Hz/4Vpp. These waves are visible due to a backlight

behind the tube covered with a polarizer, and a crossed polarizer attached to the camera lens.

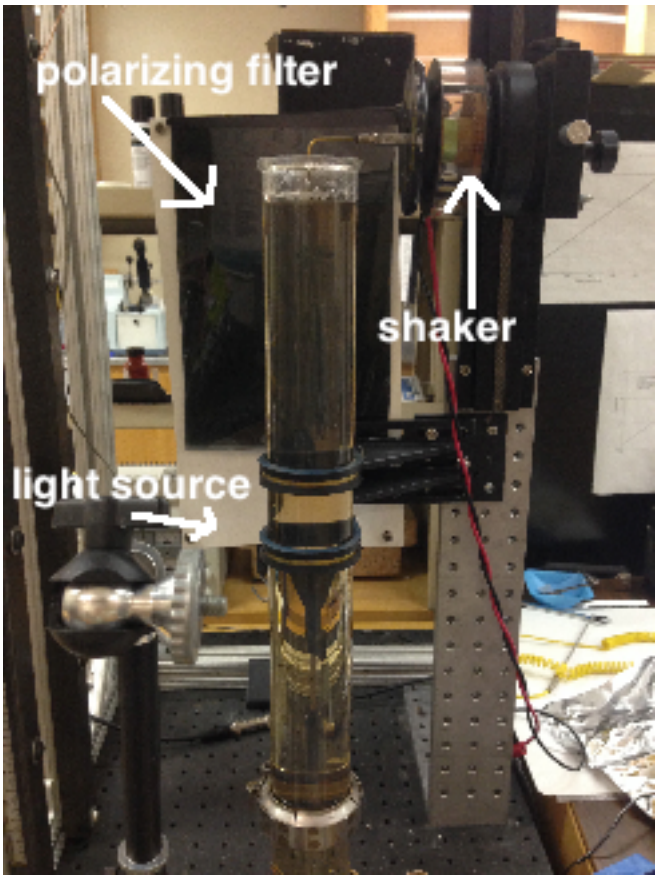

Figure 6: Zoomed out view from the camera without the crossed polarizing filter, which would cover the lens.

Thus, the inherent birefringence of the fluid, due to aligned micelles, allows light to still travel through to the camera. These photographs were run first through a program written in Python by Dr. Joseph Gladden that renamed each photograph as its corresponding temperature (see Appendix A for code). This particular script called for a text file that listed the temperatures with a count of the minutes at which they were recorded. Next, a second program (see Appendix B) rotated and

cropped the picture, then converted it to a .png file. In order for the code to accomplish these tasks, the dimensions of the desired crop must be manually entered. The dimensions were established using an open source image editing program called Gimp. Finally, a third script (see Appendix C) was written that produces a color map of each photograph to illustrate the intensity of the light transmitted. These values are quantified on a proximate plot that shows the intensity as a function of distance from the surface of the fluid (for examples of both images, see Fig. 7).

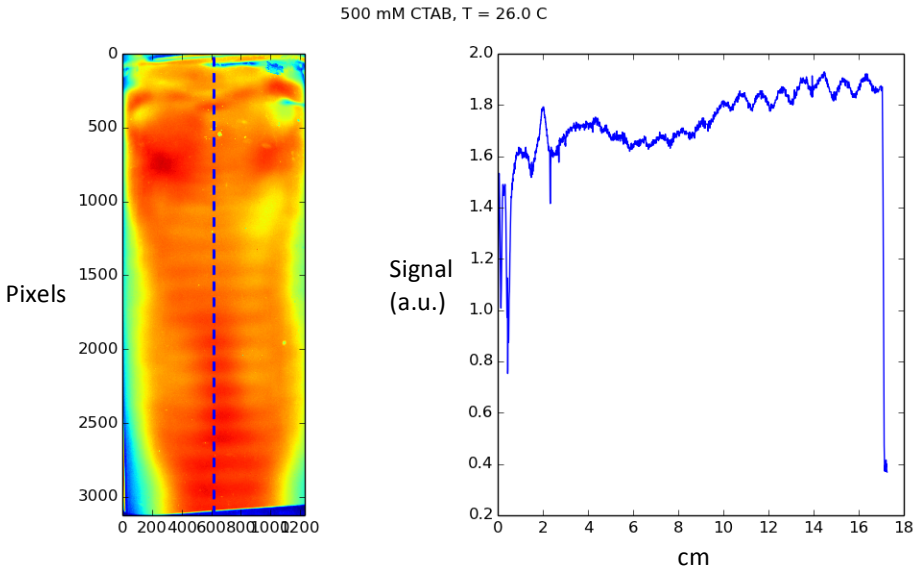

Figure 7: A color map illustrating the intensity of light transmitted through the fluid at 26°C (red being the highest). To the right, a plot of the intensity of that signal vs distance down the midline.

By using a centimeter/pixel ratio manually obtained from a calibration photograph, this script additionally produces a Fourier transform of power as a function of wavenumber (see Fig. 8).

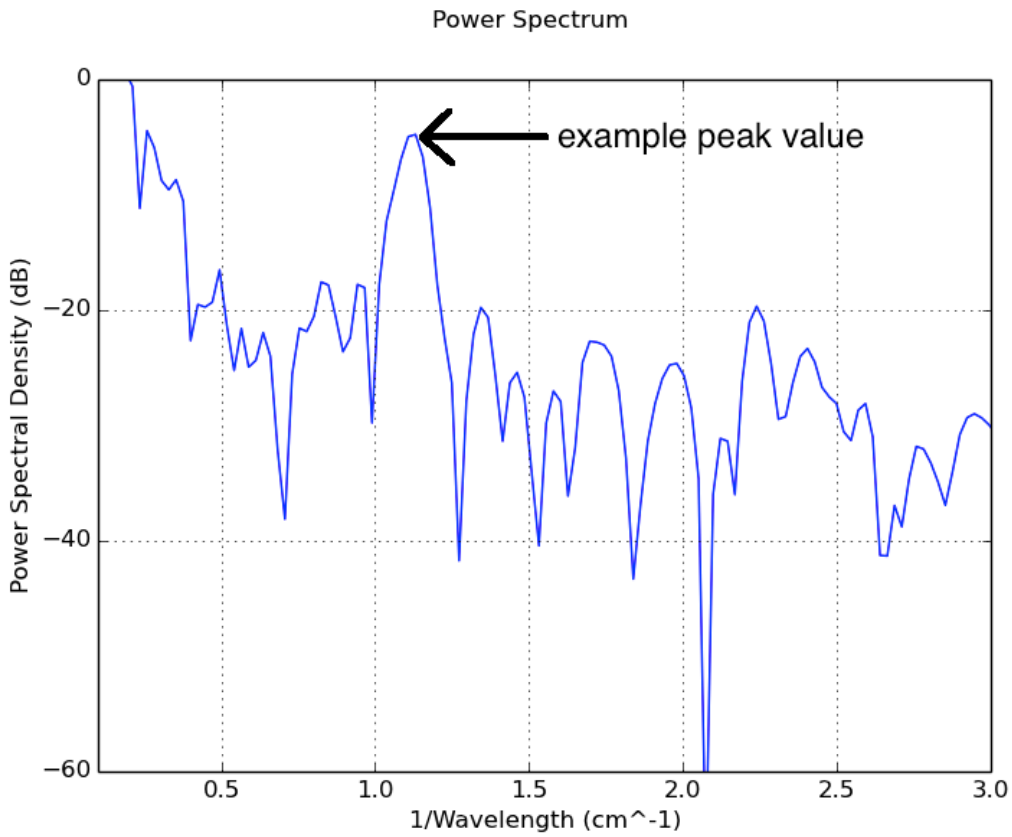

Figure 8: Example power spectrum Fourier Transform from the 26°C measurement

From this graph, the characteristic wavelength was determined by simply taking the inverse of the x-value, the wavenumber in cm^{-1}, of the most prominent peak. This value was then used to solve for the velocity of the shear wave in the fluid at certain temperatures:

$$v = f\lambda$$

Results and Discussion

In Experiment 1, success relied on bright fringes being visually distinguishable; for it is from the fringes that wavelength can be determined. However, a characteristic wavelength could not be identified for many temperature values because fringes were not visible until the fluid cooled to roughly 32^{o}C (see Fig. 9). Thus, analysis was only possible for photographs taken at temperatures lower than 32^{o}C. This was also true when the shaker was being driven at 4V rather than 7V. Resolution continued to be a problem even for cooler temperatures. For them, velocity values could be determined, but the contrast of the wave pattern was not high enough to reflect the predicted minor increases in speed. From 31.9^{o}C down to 26^{o}C, the wavelength values did not fall outside of the 0.88-0.92cm range. It should be noted that as the fluid cooled, the wavelength did increase in general. However, the value changes were minute and the fluctuations irregular, so the results of Experiment 1 were deemed inconclusive.

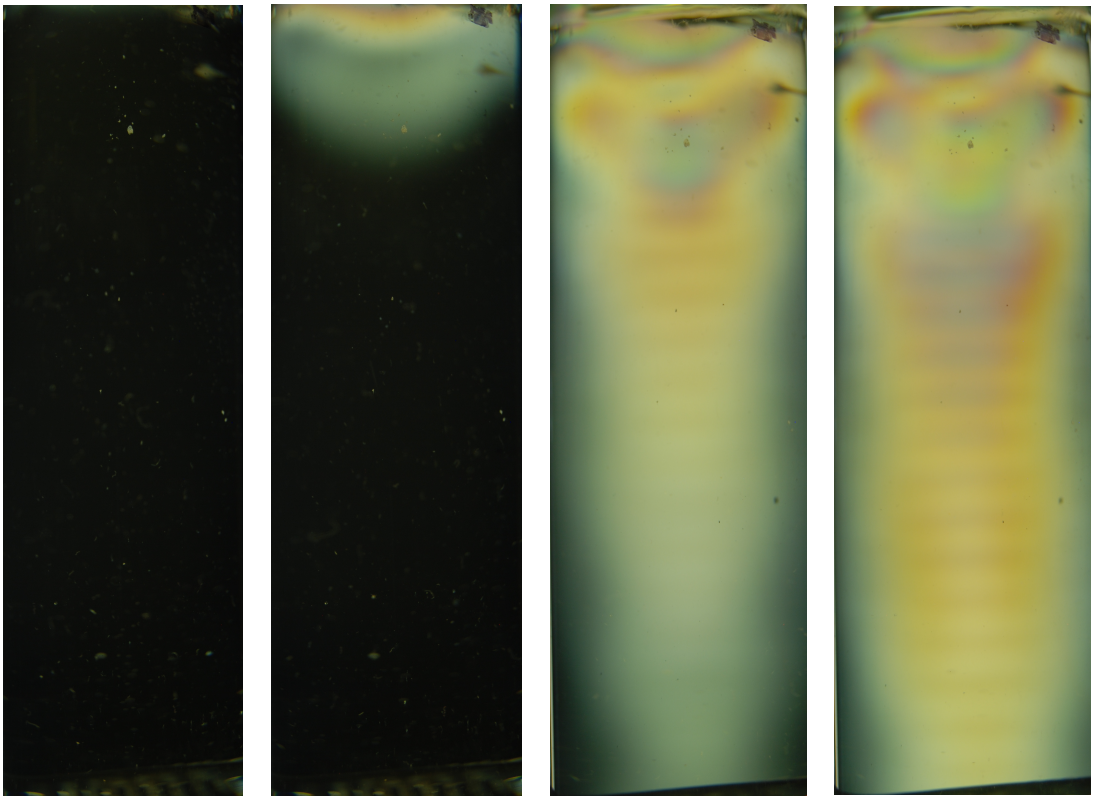

Figure 9: Photographs of the 500mM wormlike micellar fluids at 50°C, 40°C, 30°C, and 26°C (room temperature).

Conclusion

It is clear from naked observation, as well as the photographs, that the fluid stiffened as it cooled down. As temperature dropped, stress patterns became more optically apparent (refer to Fig. 9). Because of the increase in stiffness, shear speed grew faster and wavelength grew shorter. Unfortunately, the resolution of the photographs for all three frequency/amplitude combinations was flawed. The contrast was too low to be able to discern a wave pattern on a small enough scale to detect the changes in wavelength (and therefore speed). No relationship between temperature and shear speed could be quantified.

EXPERIMENT 2: EFFECT OF AGING ON SHEAR SPEED AND ATTENUATION IN WORMLIKE MICELLAR FLUID

Introduction

Experiment 2 used a laser/diode system to manipulate strain-birefringence and observe speed of the shear wave in a 500mM wormlike micellar fluid over the course of 51 days. Little effort has been made in the past to study the effects of aging on the rheological properties of the fluids. The purpose of Experiment 2 was to make multiple shear speed measurements and observe how the micelles react to aging.

Methods

For Experiment 2, the apparatus slightly differs from Experiment 1 (see Fig. 10). The tube remains in the same location (with the same fluid) while the backlight and camera are removed. A translational stage carrying the optical system was attached to a lead screw that allowed for vertical movement, controlled by an Applied Motion Products step motor.

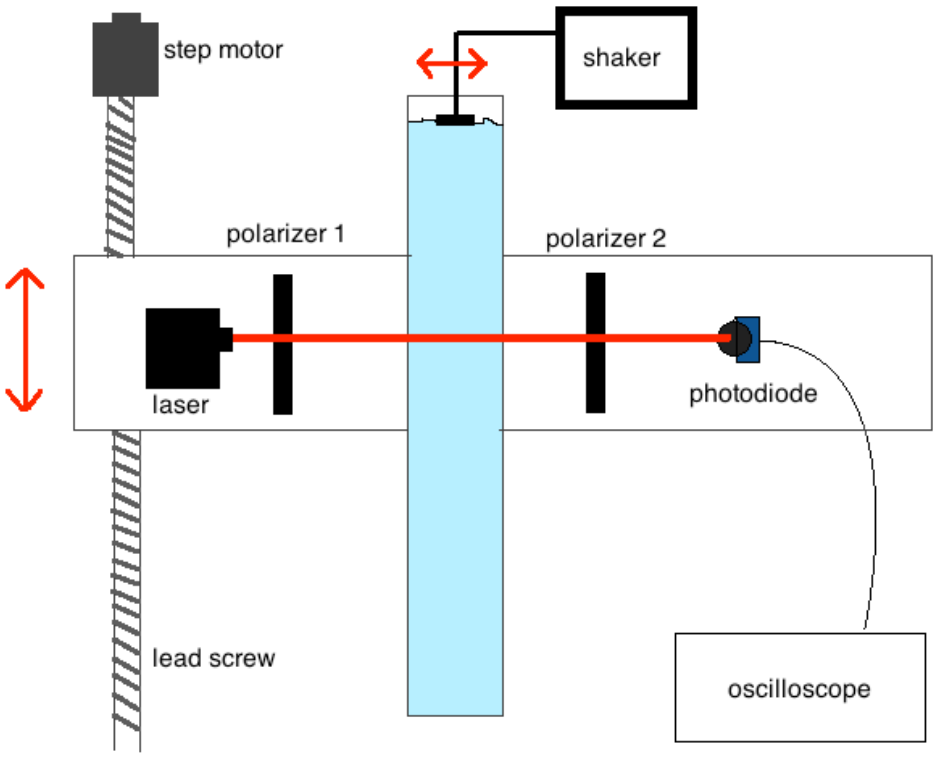

Figure 10: Schematic of setup for Experiments 2 and 3.

To operate the motor, a Labview program was created that runs the motor for a designated amount of time, turning the lead screw and either lowering or raising the optical system mounted on the translational stage. The time was deduced simply by trial and error. With this particular lead screw, the iteration needed to be set to 1600 in order for the stage to move 1cm vertically. A switch was connected in parallel with the motor that controlled the direction of vertical movement. The optical system consisted of a laser covered with a blue filter to reduce intensity, a polarizer in front of the tube, a crossed polarizer behind the tube, and a photodiode behind that (see Figures 11 and 12).

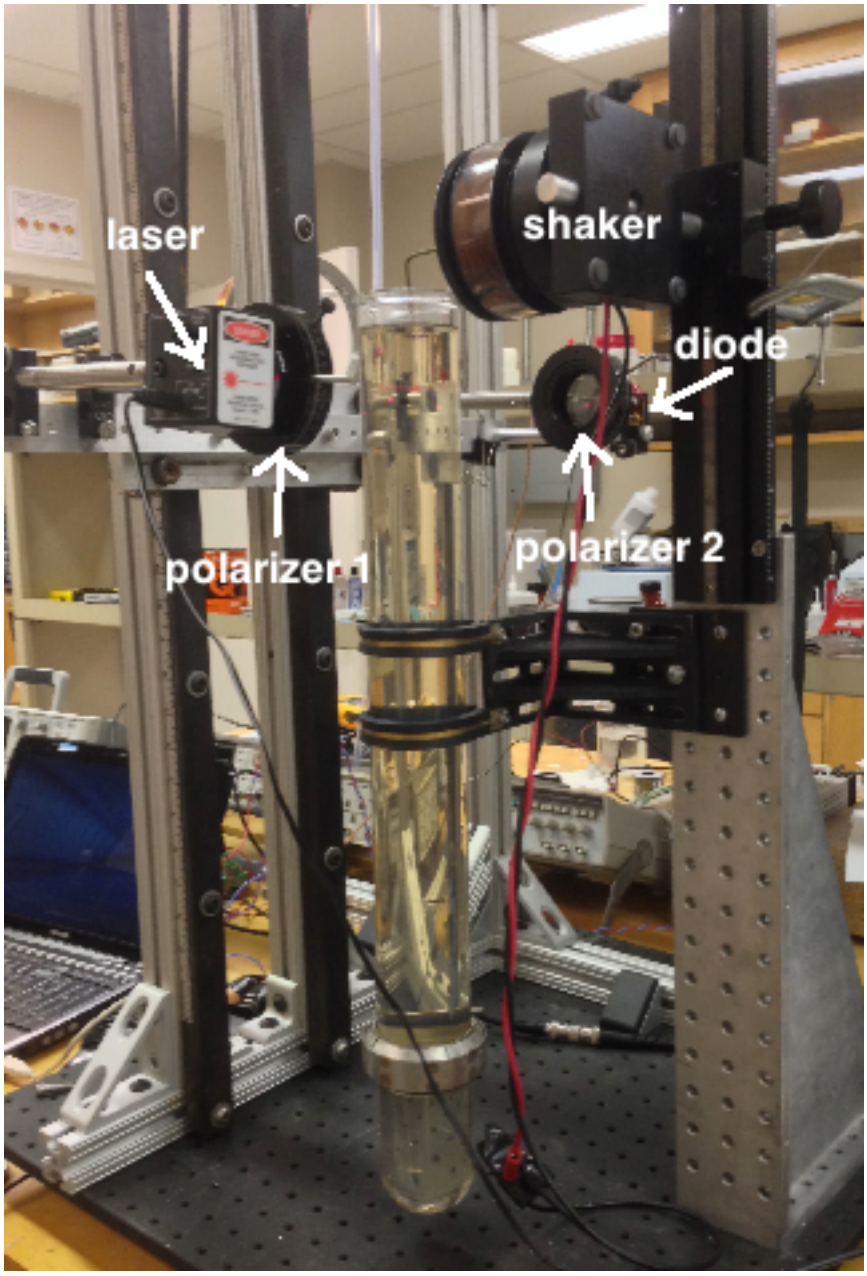

Figure 11: A side view of the setup for Experiments 2 and 3.

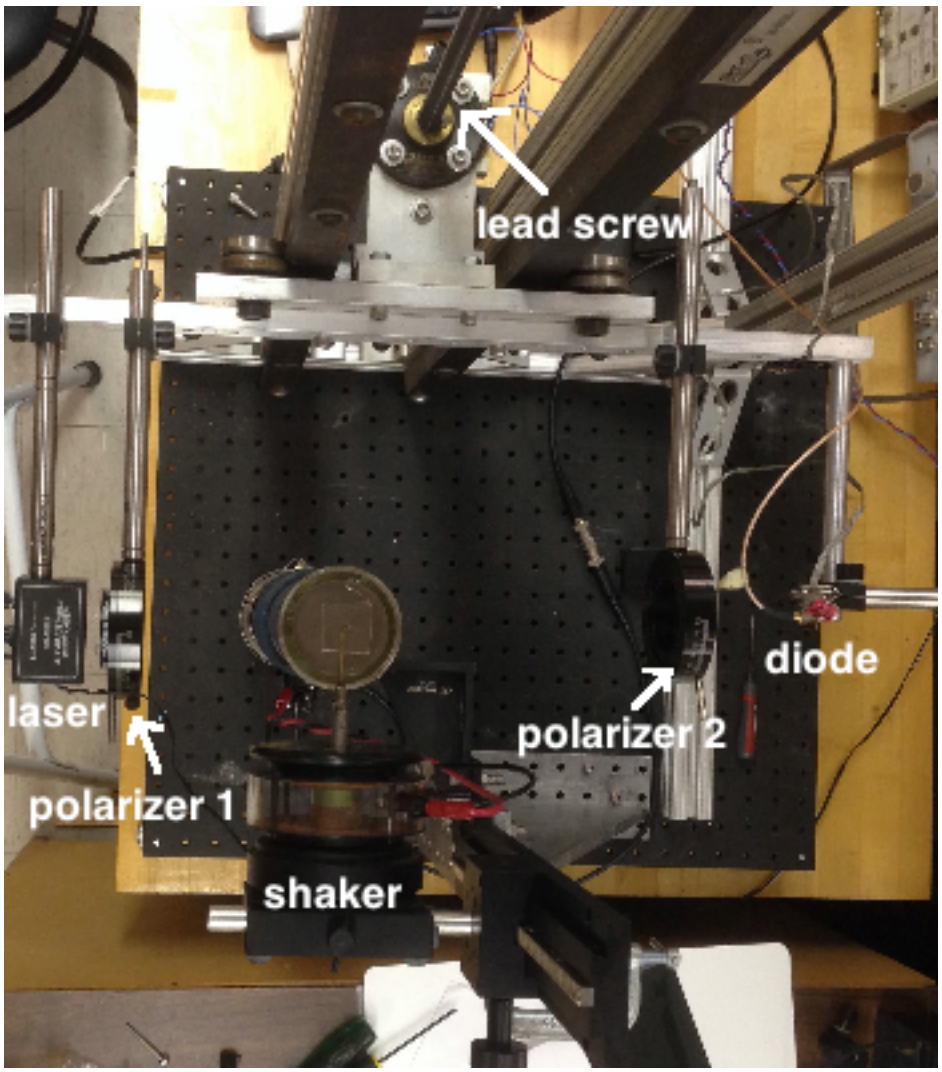

Figure 12: View from above of setup for Experiments 2 and 3.

The shaker was altered so that it executed a 45Hz pulse for 1s, and the changes in intensity of light was read by the photodiode, which relayed the information to an oscilloscope with preset parameters (see Appendix D for code). What the oscilloscope mapped was a spike in voltage, where the shear waves actually passed through the laser beam, and an exponential relaxation tail following (see Fig. 13).

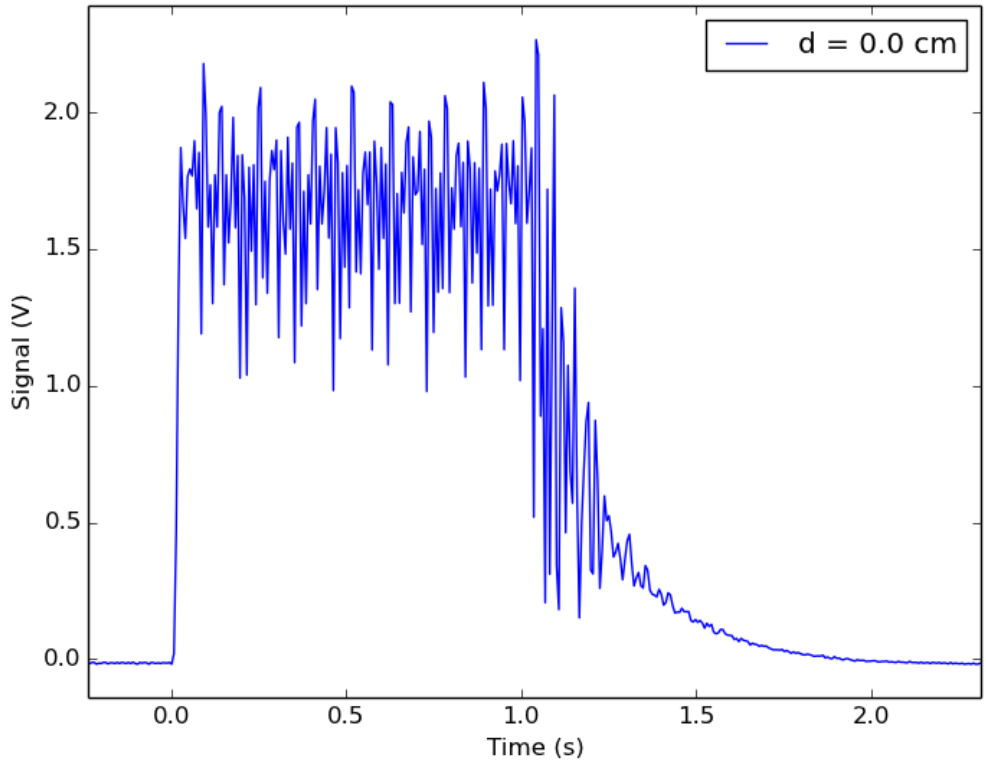

Figure 13: An example of the spike in light intensity (measured in volts) and subsequent relaxation tail that results from a 1s 45Hz shear pulse at room temperature.

The oscilloscope was connected to a computer that recorded this information as a series of text files. The first measurement (position00) was taken 1cm from the surface of the fluid, and then the translational stage would carry the optical system down 1cm, where the process would be repeated. A signal could be distinguished until approximately 32cm down the tube, and data was taken this far for the first leg of Experiment 2. Measurements were taken every three or four days starting 20 days after the creation of the mixture and ending 31 days later. In the second leg, the fluid was thermally reset (heated to 70^{o}C and stirred profusely) and data was

taken from 0-4 days following. Since the heater tape could not cover the entire tube, measurements were taken only for the first 25cm to optimize accuracy.

In order to process the intensities as a function of distance down the tube, a code was written (refer to Appendix E) that uses these .txt files to find the edge of the shear pulse, and measures the time it took for the pulse to reach the laser beam after the actual shearing of the plate (when measuring began). The oscilloscope triggers when the shaker shears, with an additional 2s delay, so each data set has a common starting point. The further the optical system travels down the tube, the longer it will take the shear pulse to reach it, as seen in Figure 14:

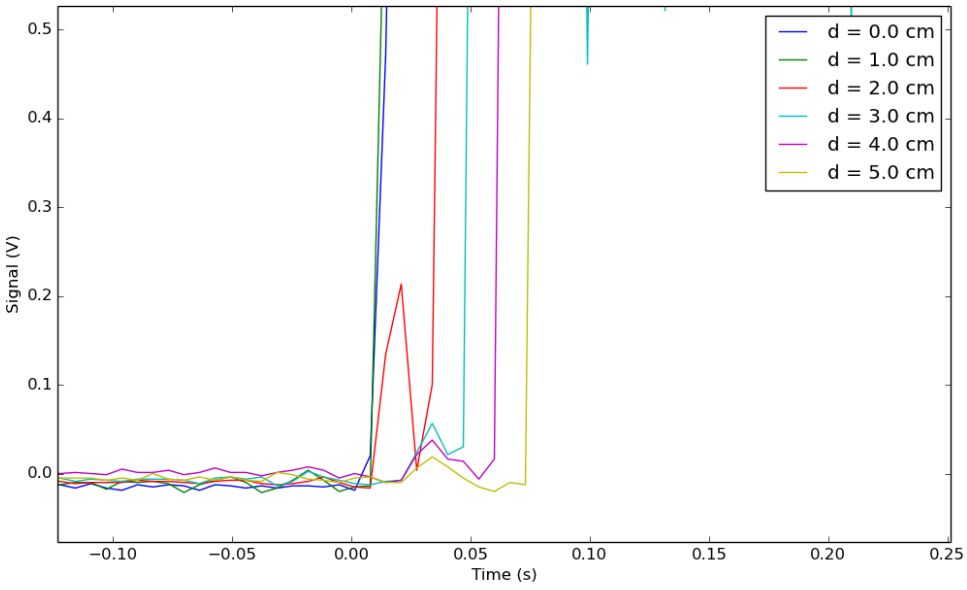

Figure 14: Multiple pulse signals (like the one seen in Fig. 13) laid side by side to demonstrate the increase in time delay. The time between the actual shear pulse and when that pulse reaches the laser/photodiode system increases as distance (d) increases.

The program takes this amount of time measured for each position along the tube, and simply plots position in centimeters versus time in seconds (for an example plot see Fig. 15) to produce the speed of the shear pulse through the wormlike micellar

fluid. This code also calculates the relaxation length by plotting amplitude during pulse in volts versus position in centimeters.

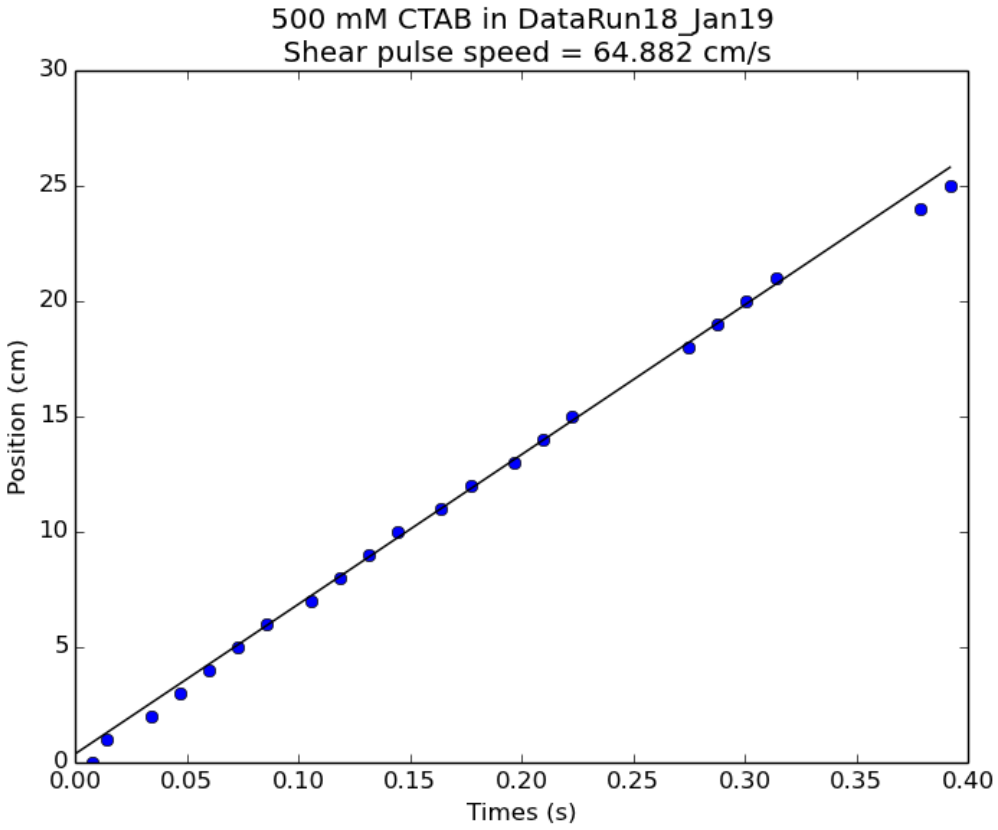

Figure 15: Velocity of the shear pulse 2 days after thermal reset.

Results and Discussion

The amplitude of the pulses, related to the attenuation, decreased in a reverse-exponential fashion as the sensor moved further from the surface of the fluid (see Fig. 16). If the pulse were equated to the ringing of a bell, the amplitude would be the loudness. As one moves down the tube, the intensity, or "loudness," decreases.

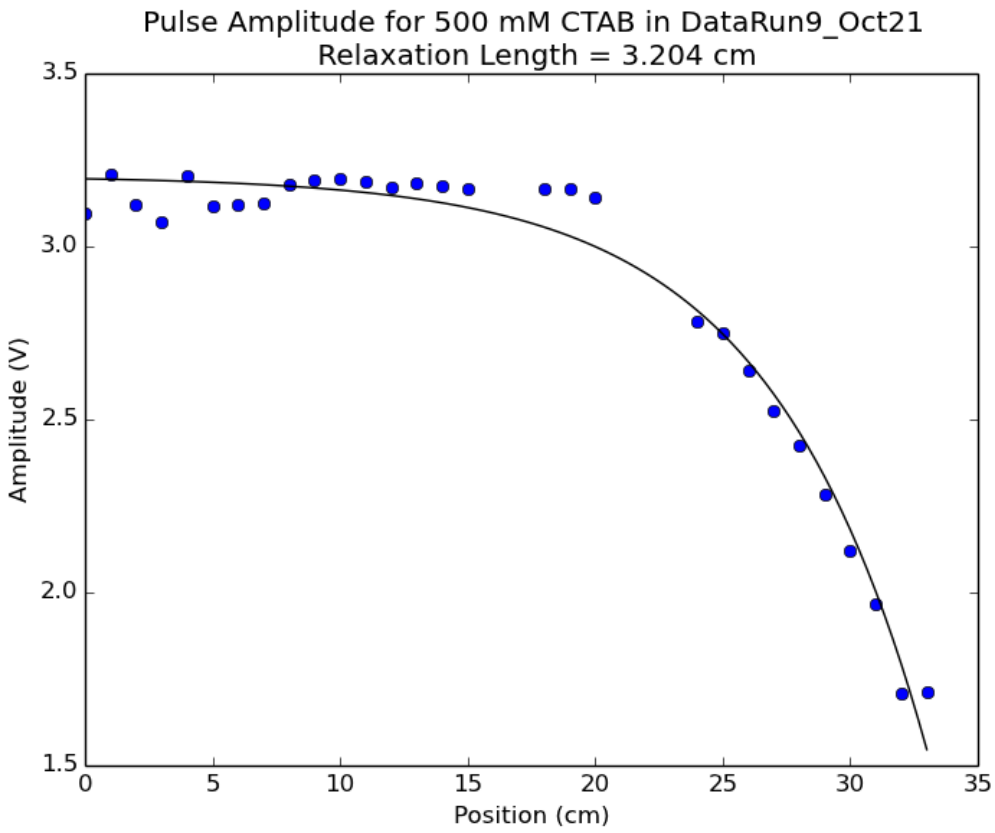

Figure 16: Amplitude of the pulse as a function of distance down the tube (away from the shear source).

Additionally, it was observed that as distance from the stress source increased, the exponential relaxation decreased, as seen in Figure 17. In the bell example, exponential relaxation would be the "ring down," or, the bell fading in volume as measured at a fixed position.[2] As you move down the tube, this "ring down" fades out more steeply. This curiosity is illustrated in Figure 17, where relaxation is equated with time constant tau (τ). It was found that an exponential model did not quite fit the relaxation curves, but rather, the over damped oscillation model fit well, according to $S(t)=Ae^{-t/\tau}.\cos(\omega t + \phi)$. Figure 18 illustrates the steepening of the exponential relaxation (decreasing τ) as the translational system

moves further from the shear source. Around 10cm down the tube, the pulse damps too quickly to measure.

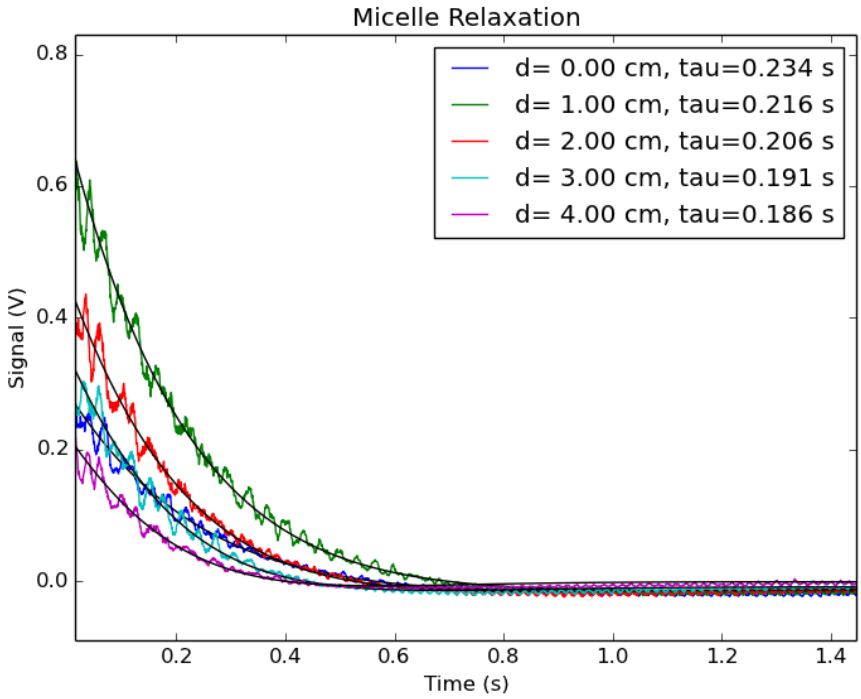

Figure 17: Relaxation after 5 different pulses. 5 different "ring downs."

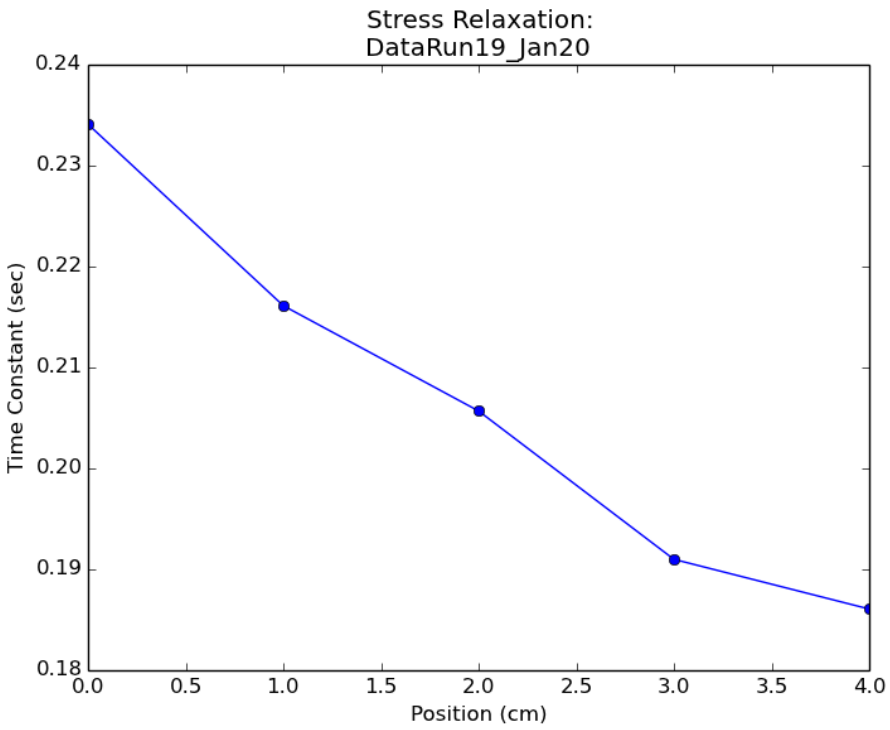

Figure 18: Tau as a function of position

The highest velocity calculated was 67.86cm/s and had a coefficient of determination (R^2) of 0.998. This value was measured the day of the thermal reset. Alternatively, the lowest velocity was measured 27 days after the thermal reset and equaled 60.64cm/s, with R^2=0.999. Unfortunately, the speed variation is large and erratic within this range over the course of the 51 days. For each measurement, based off of intensity data taken from the photodiode, R^2 was greater than 0.996. No consistent aging effect was seen. Shear waves in the wormlike micellar fluid at non-monitored varying room temperatures were calculated to have an average speed of 63.47cm/s. It is assumed the fluctuating ambient temperatures were not enough to cause the high variation in speed values (see Experiment 3 and Fig. 19). Between 26^{o}C and 33^{o}C, the shear speed does not vary more than 1cm/s.

Conclusion

As distance from the shear source increases, the relaxation constant (**τ**) and amplitude for identical pulses decrease. The average speed of shear waves in 500mM wormlike micellar fluid at room temperature is 63.47cm/s. On this value, aging has no distinct impact. No pattern was seen in shear speeds as number of days increased, but the irregular nature of the fluctuations in speed was cause for question.

EXPERIMENT 3: TEMPERATURE DEPENDENCE OF SHEAR SPEED IN WORMLIKE MICELLAR FLUIDS USING A LASER

Introduction

Since the relationship between shear speed and temperature was unable to be determined in Experiment 1, and this relationship could have affected Experiment 2, a third experiment was conducted, again using the laser/diode system. The purpose was to investigate the effect of ambient temperature on the fluid.

Methods

The fluid was again thermally reset (held around 50^{o}C for one hour while also being stirred), and data was taken continuously for the first six positions as the fluid cooled, employing the same parameters used in Experiment 2. The temperature was recorded using the same Type E thermocouple, which was inserted into the fluid at the onset of the experiment. Data was taken at position00-position05, and the ending temperature was recorded. The temperature values for the beginning and end points were averaged, and the trial was named for that temperature. Then the stage returned up to position00 and the process began again. Each run took roughly thirty seconds and the temperature did not change more than 0.3 ^{o}C. As the fluid cooled, variation decreased (refer to Figure 5). This procedure was repeated continuously until the fluid reached 24.0^{o}C. A pulse was distinguishable only at 39.45^{o}C and cooler. To analyze this mass of data, a code was written to go into the files designated for each temperature, pull out the speed, and

plot speed versus temperature (refer to Appendix F for code and to Fig. 19 for graph).

Results and Discussion

It was anticipated that as temperature increases, the stiffness of the fluid decreases. Since the speed of sound is proportional to the square root of stiffness (shear modulus), this would mean that the wave propagates more slowly as the fluid warms up. However, that was not exactly the case, as evidenced by Figure 19:

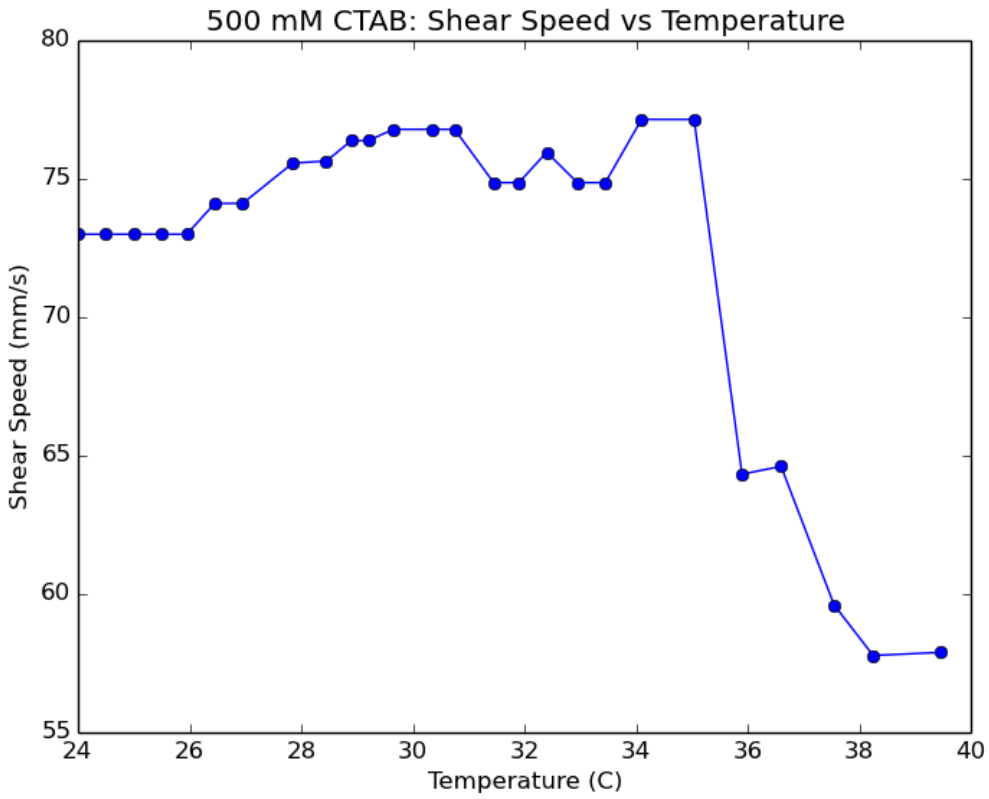

Figure 19: Speed of shear waves as a function of temperature for 500mM CTAB/NaSal micelle solutions. The speed at room temperature is greater here than was found in Experiment 2 because only the first six positions are used here, whereas in Experiment 2, intensity values were found for between 25 and 32 positions. It can be seen in Fig. 15 that the first six or seven positions alone yield a steeper slope (and thus higher speed) than the entire line. Therefore, the data is consistent.

The speed of the shear wave gradually increases (only slightly, the slope is not much greater than the error values) until about 35.5°C, where there is a steep drop-off, suggesting a dramatic softening. Observation of odd behavior near this temperature is not unprecedented in wormlike micellar fluids. Unpublished data taken by Dr. Joseph R. Gladden in 2004 indicated a similar anomaly in the relaxation activation energy (see Fig. 20). This information suggests that something dramatically changes in the topology of the micelles in this temperature range. It has been proven in low to medium surfactant concentrations (<100mM) that micelle length shortens linearly with increasing temperatures.[3] However, this gradual decrease in length does not account for the significant drop-off in speed that occurs around 35°C. Little information is known regarding the structure of the 3D networks in which the wormlike micelles are entangled, but it can be surmised that the dramatic structural alteration in question occurs on this level.

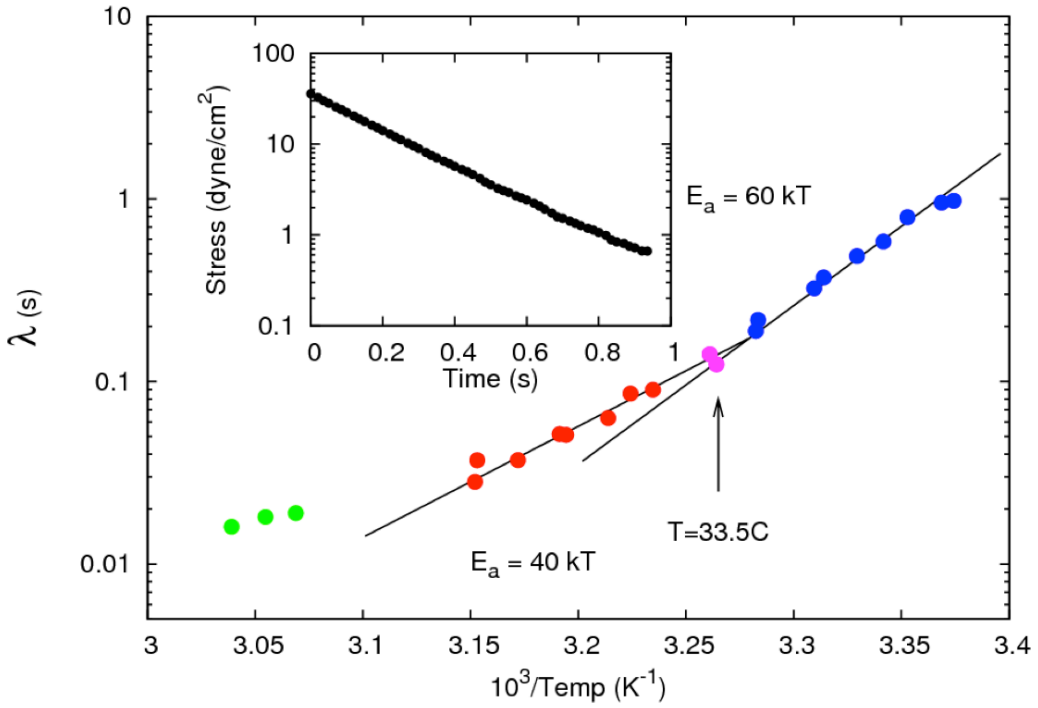

Figure 20: Unpublished data measured by Joseph R. Gladden shows relaxation time vs. 1/temperature. This plot illustrates that with decreasing temperature, the fluid's relaxation time increased according to an Arrhenius fashion by a certain factor. At 33.5°C, something in the fluid changes and the relaxation time begins to increase by a different factor.

Conclusion

Somewhere between 33-36°C, a dramatic change seems to occur in the morphology of worm networks at high concentrations. The relationship between a micellar solution of 500mM CTAB and temperature does not follow a simple linear proportionality. The dramatic softening would be consistent with a topological transition from a 3D branched micelle network at lower temperatures to a loose entanglement at high temperatures. This transition would be occurring at 33-36°C.

FINAL CONCLUSIONS

In summary, temperature effects on shear speed and attenuation in wormlike micellar fluids were optically apparent, but could not be quantified with the resolution level used in the camera/backlight setup of Experiment 1. Aging seems to have an effect as well, but that effect could not be determined quantatatively. Over the course of 51 days, the shear speed in room temperature 500mM wormlike micellar fluid varied over a range of 67.86 - 60.64cm/s, with an average of 63.47cm/s. Finally, there was a gradual increase in shear speed as the fluid was being heated. But when temperature reached about 35^{o}C, there was a dramatic softening in the fluid, and shear speed decreased significantly, which, based on past data, is not unprecedented. It is suggested that further research be done in order to investigate the morphology of high concentration wormlike micellar fluids and how it changes at 33-36^{o}C

REFERENCES

1. V.J. Anderson, E.S. Boek, J.R.A. Pearson, "The Rheology of Worm-like Micellar Fluids," *Rheology Reviews* (2006): 217-253.
2. Joseph R. Gladden in discussion with the author, October 2012.
3. Hu Cao, Narayan Ch. Das, Joseph R. Gladden, Helmut Kaiser, Paul E. Sokol, Garfield T. Warren, "Shape and Size of Highly Concentrated Micelles in CTAB/NaSal Solutions by Small Angle Neutron Scattering (SANS)," *Langmuir* 28 (2012): 11962-11968.
4. Jean-François Berret, "Rheology of Wormlike Micelles: Equilibrium Properties and Shear Banding Transition," in *Molecular Gels,* edited by Richard G. Weiss and Pierre Térech (Dordrecht: Springer, 2005): 235.
5. Joseph R. Gladden, Andrew Belmonte, "Motion of a Viscoelastic Micellar Fluid Around a Cylinder: Flow and Fracture," *Physical Review Letters* 98, no. 22 (2007): 4501.
6. Bani H. Cipriano, Dganit Danino, Bradley D. Frounfelker, Gokul C. Kalur, "Persistence of Birefringence in Sheared Solutions of Wormlike Micellar Fluids," *Langmuir* 25 (2009): 167-172.
7. J.R. Gladden, J. Mobley, C.E. Skelton, "Shear Waves in Viscoelastic Wormlike Micellar Fluids," *Journal of the Acoustical Society of America* 128, no. 5, (2010): 268-273.
8. A.M. Gamble, J.R. Gladden, J. Mobley, C.E. Skelton, "Shear Waves in Viscoelastic Wormlike Micellar Fluids Over a Broad Concentration Range," *Journal of the Acoustical Society of America* 131, no. 3, (2012): 2063-2067.

APPENDIX A

renamefiles.py

```
from glob import glob
from os import rename
from numpy import loadtxt

picnums,temps = loadtxt('pic_temp.txt',unpack=True, comments = '#')
files =glob('DSC*.JPG')
i=0
for file in files:
        newname = "image_T%2.1f.jpg"%temps[i]
        print file," to ",newname
        rename(file,newname)
        i+=1
```

APPENDIX B

RotateCrop.py

```
from subprocess import call
from glob import glob

files = glob('image_T*.jpg')
cropdims = '1432x2784+624+304'
imageTotal = len(files)
i=0
for file in files:
        outputFile = file[:-3]+'png'
        i+=1
        print 'Working on image: %s (%i/%i)' % (file,i,imageTotal)
        call(['convert',file,'-rotate','+90','temp.png'])
        call(['convert','temp.png','-crop',cropdims,outputFile])
```

APPENDIX C

speed_atten_photo_v0.5.py

```
from pylab import *
import sys

close('all')

if len(sys.argv) > 1:
        filename = sys.argv[1]
else: filename='image_T33.4.png'

temp = filename[7:11]

##############################
#Width of tube is 1380 pixels for a 3 (7.62 cm) inch tube
px2cm = 7.62/1380 #cm/pixel
freq = 30 #Hz
colorchannel = 3 # 0 = red, 1 = green, 2 = blue, 3 = all channels summed
close('all')
##############################


im = imread(filename)
hi=im.shape[0]
hcm = hi * px2cm
wi=im.shape[1]
harray = px2cm*linspace(0,hi,hi)


def findMinima(slice,window=(0.33,0.66)):
        minima = []

        return minima
wslice = wi//2
if colorchannel == 3:
        channel = im[:,wslice][:,0]+im[:,wslice][:,1]+im[:,wslice][:,2]
else: channel = im[:,wslice][:,colorchannel]

fftb = fft(channel)
MAG = sqrt(fftb.real**2 + fftb.imag**2)
sampleSize = len(channel)
spacing = harray[1]-harray[0]
fftK = fftfreq(len(fftb),d=spacing)


figure(figsize=(12,6))
suptitle("500 mM CTAB, T = %s C" % temp)
subplot(121)
if colorchannel == 3: imshow(im[:,:,0]+im[:,:,1]+im[:,:,2])
else: imshow(im[:,:,colorchannel])
arrow(wslice,0,0,hi,color="blue",lw=2,ls='dashed')
subplot(122)
plot(harray,channel)


figure()
suptitle('Power Spectrum')
subplot(111)
psd(channel,NFFT=len(channel),
        Fs=1/(harray[1]-harray[0]),
        pad_to=7680,
        scale_by_freq=False)
xlabel("1/Wavelength (cm^-1)")
xlim(0.1,3)
ylim(-60,0)

#subplot(212)
```

```
#semilogy(fftK,MAG)
#xlabel("Wavelength (cm)")
#xlim(0.1,3)

show()
```

APPENDIX D

AgilentControl_v1.0.py

```
import visa
import pylab as pl

timeScale = 0.5 # seconds per division
timeDelay = 2.000 # set delay time
voltOffset = +1500.0 #voltage offset in mV
voltScale = 500.0 #Volts / div in mV


instList = visa.get_instruments_list()
for inst in instList:
        usbscope = instList[1]
        scope = visa.instrument(usbscope)
        scopeid = scope.ask('*IDN?')
        if scopeid.count('2012') >0: break
        print "No Agilent DSO 2012 Oscope found in list of instrument:"
        print instList

#Set Waveform parameters
scope.write(':WAV:SOUR CHAN1') # Set source to Chan 1
scope.write(':WAV:FORM ASC') # Read data in ACII format
scope.write(':ACQ:TYPE HRES') # Set Acquisition mode to HRES
scope.write(':WAV:POIN 50000') # Acquire the maximum # of points

# Set Time base paprameters
scope.write(':TIM:SCAL %2.4f'%timeScale) #Set 500 ms/div
scope.write(':TIM:POS %2.4f'%timeDelay) #Set delay to 2.0 sec

#Set vertical parameters (voltage
scope.write(':CHAN1:SCAL %2.4f mV'%voltScale)
scope.write(':CHAN1:OFFSet %2.4f mV'%voltOffset)


def doRun():
        posn = raw_input("Position number ('done' to exit):")
        if posn == 'done':
                print "Exiting data acquisition..."
                return
        outfile = 'position'+posn+'.txt'
        pl.close('all')
        pl.figure()
        #Grab data from scope. 1st 10 characters are a preamble indicating length of string
        sdata = scope.ask(':WAV:DATA?')[10:].split(',')
        signal = pl.array(map(float,sdata))
        #Get the time base parameters
        timeOrig = float(scope.ask(':WAV:XOR?') )
        timeStep = float(scope.ask('WAV:XINC?'))
        timeInfo = scope.ask(':TIM?').split(';')
        timeDelay = float(timeInfo[-1].split(' ')[-1])
        timeRange = float(timeInfo[-2].split(' ')[-1])
        print "Start time: %2.2f sec, Time Delay: %2.3f sec, Sample Rate: %2.3f kSa/sec"%(timeOrig,
timeDelay,1./timeStep/1000.)
        print "Data points acquired: %2.4f"%len(signal)
        time = pl.linspace(timeOrig,timeStep*len(signal)+timeOrig,len(signal))

        #plot to check (only plot every 3rd point for speed)
        pl.plot(time[::3],signal[::3])
        pl.show()

#Save data
        data = zip(time,signal)
        pl.np.savetxt(outfile,data)
        return 0

def getParams():
```

```
        timeOrig = float(scope.ask(':WAV:XOR?') )
        timeStep = float(scope.ask('WAV:XINC?'))
        timeInfo = scope.ask(':TIM?').split(';')
        timeDelay = float(timeInfo[-1].split(' ')[-1])
        timeRange = float(timeInfo[-2].split(' ')[-1])
        print "Start time: %2.2f sec, Time Delay: %2.3f sec, Sample Rate: %2.3f kSa/sec"%(timeOrig,
timeDelay,1./timeStep/1000.)
```

APPENDIX E

speed_atten_v1.3.py

```
from pylab import *
import glob, scipy.stats
import os
from scipy.optimize import curve_fit

datadir = os.getcwd().split('/')[-1]
files = glob.glob('pos*.txt')

plotRawData = False
step = 1.0 # cm
stepThresh = 0.2
pulseTimes = []
pulsePositions = []
sigAmps = []
window = 10

def getOnset(time, signal):
        maxSig = max(signal)
        minSig = min(signal)
        for i in range(window,len(signal),window):

                currentAvg = average(signal[i-window:i])
                if i>window:
                        nextAvg = average(signal[i:i+window])
                        if nextAvg - currentAvg > (maxSig - minSig)*stepThresh:
                                pulseTime = time[i]
                                edgeIndex = i
                                return pulseTime, edgeIndex

        print "No edge detected with threshold of: ", stepThresh
        return None

def getAmp(signal, edgeIndex):
                maxAmps = []
                window = 1000
                for i in range(edgeIndex+window,edgeIndex+1500,window):
                        maxAmps.append(max(signal[i:i+window]))
                #print maxAmps
                avgAmp = average(maxAmps)
                return avgAmp

def getData(filename):
        time,signal = np.loadtxt(filename,unpack=True,skiprows=0)
        return time,signal

def plotData(time,signal,position):
        time = time[::10]
        signal = signal[::10]
        plot(time,signal, '-',label = "d = %2.1f cm"%position)

#function to fit relaxation data
def relax(x,A,d,So):
        return -A*exp(x/d) + So

close('all')


def plotSelect(positions):
        for position in positions:
                location=float(files[position][8:10])*step
                time,signal = getData(files[position])
                plot(time,signal,label = 'd = %2.1f cm'%location)

for file in files:
        position=float(file[8:10])*step
        pulsePositions.append(position)
```

```
        time,signal = getData(file)
        pulseTime,edgeIndex = getOnset(time,signal)
        avgAmp = getAmp(signal, edgeIndex)
        sigAmps.append(avgAmp)
        print "For file %s: Pulse edge found at %3.5f seconds, amplitude is %2.5f V" %
(file,pulseTime, avgAmp)
        if pulseTime: pulseTimes.append(pulseTime)
        if plotRawData:
                if position % 1 ==0 and position <1:
                        plotData(time,signal,position)
if plotRawData:
        xlabel("Time (s)")
        ylabel("Signal (V)")
        legend()

pulseTimes = array(pulseTimes)
pulsePositions = array(pulsePositions)

fitTimes=linspace(0,max(pulseTimes),100)
fitBracket = 5
p=polyfit(pulseTimes[fitBracket:-fitBracket],pulsePositions[fitBracket:-fitBracket],1)
slope, intercept, r_value, p_value, std_err = scipy.stats.linregress(pulseTimes[fitBracket:-
fitBracket],pulsePositions[fitBracket:-fitBracket])
fitPositions=polyval(p,fitTimes)



figure()
plot(pulseTimes,pulsePositions, 'bo')
plot(fitTimes,fitPositions,'k-')
xlabel('Times (s)')
ylabel('Position (cm)')
title("500 mM CTAB in %s \n Shear pulse speed = %2.3f cm/s"%(datadir,p[0]))

## Fit Relaxation Tail
par0=(2.,5.,5.)
fit = curve_fit(relax,pulsePositions, sigAmps,p0=par0)
optParams = fit[0]
A = optParams[0]
d = optParams[1]
S0 = optParams[2]
posnFit = linspace(min(pulsePositions),max(pulsePositions),100)

print "="*30
print "Pulse speed = %2.3f cm/sec"%p[0]
print "Correlation Coeff (R^2):", r_value**2
print '-'*10
print 'Pulse Attenuation Length = %2.2f cm'%d
print 'Attenuation/MaxSignal Ratio = %2.3f cm/V' %(d/S0)
print "="*30


figure()
plot(pulsePositions, sigAmps,'bo')
plot(posnFit,relax(posnFit,A,d,S0),'k-')
xlabel('Position (cm)')
ylabel('Amplitude (V)')
title("500 mM CTAB in %s \n Relaxation Length = %2.3f cm and d/S0 Ratio = %2.3f"%(datadir,d,d/S0))
show()
```

APPENDIX F

speed_atten_temp_v1.1.py

```
from pylab import *
import glob, scipy.stats
import os
from scipy.optimize import curve_fit

rootdir = os.getcwd()
datadir = rootdir.split('/')[-1]
dirs = glob.glob('T*')
temps=[]
for dir in dirs:
	temps.append(float(dir[1:]))

close('all')
doRelax = False
plotRawData = False
plotEachTemp = True
step = 1.0 # cm
stepThresh = 0.1
pulseTimes = []

window = 1
results = []

def getOnset(time, signal):
	maxSig = max(signal)
	minSig = min(signal)
	for i in range(window,len(signal),window):

		currentAvg = average(signal[i-window:i])
		if i>window:
			nextAvg = average(signal[i:i+window])
			if nextAvg - currentAvg > (maxSig - minSig)*stepThresh:
				pulseTime = time[i]
				edgeIndex = i
				return pulseTime, edgeIndex

	print "No edge detected with threshold of: ", stepThresh
	return None

def getAmp(signal, edgeIndex):
	maxAmps = []
	window = 1000
	for i in range(edgeIndex+window,edgeIndex+1500,window):
		maxAmps.append(max(signal[i:i+window]))
	#print maxAmps
	avgAmp = average(maxAmps)
	return avgAmp

def getData(filename):
	time,signal = np.loadtxt(filename,unpack=True,skiprows=0)
	return time,signal

def plotData(time,signal,position):
	time = time[::10]
	signal = signal[::10]
	plot(time,signal, '-',label = "d = %2.1f cm"%position)

#function to fit relaxation data
def relax(x,A,d,So):
	return -A*exp(x/d) + So


def doRun(dir):
	pulsePositions = []
	pulseTimes = []
	sigAmps = []
```

```
        files = glob.glob('pos*.txt')
        print '-'*60
        print "Working in directory: "+dir
        for file in files:
                position=float(file[8:10])*step
                pulsePositions.append(position)
                time,signal = getData(file)
                pulseTime,edgeIndex = getOnset(time,signal)
                avgAmp = getAmp(signal, edgeIndex)
                sigAmps.append(avgAmp)

                print "For file %s: Pulse edge found at %3.5f seconds, amplitude is %2.5f V" %
        (file,pulseTime, avgAmp)
                if pulseTime: pulseTimes.append(pulseTime)
                if plotRawData:
                        if position % 1 == 0:
                                plotData(time,signal,position)
        if plotRawData:
                xlabel("Time (s)")
                ylabel("Signal (V)")
                legend()

        pulseTimes = array(pulseTimes)
        pulsePositions = array(pulsePositions)

        fitTimes=linspace(0,max(pulseTimes),100)
        fitBracket = 1
        p=polyfit(pulseTimes[fitBracket:-fitBracket],pulsePositions[fitBracket:-fitBracket],1)
        slope, intercept, r_value, p_value, std_err = scipy.stats.linregress(pulseTimes[fitBracket:-
fitBracket],pulsePositions[fitBracket:-fitBracket])
        fitPositions=polyval(p,fitTimes)

        if plotEachTemp:
                plot(pulseTimes,pulsePositions, 'o',label=dir)
                plot(fitTimes,fitPositions,'k--')
                xlabel('Times (s)')
                ylabel('Position (cm)')
                title("500 mM CTAB: Shear Pulse Position vs Time")
        print "="*30
        print "Pulse speed = %2.3f cm/sec"%p[0]
        print "Correlation Coeff (R^2):", r_value**2
        print '-'*10

        ## Fit Relaxation Tail
        if doRelax:
                par0=(2.,5.,5.)
                fit = curve_fit(relax,pulsePositions, sigAmps,p0=par0)
                optParams = fit[0]
                A = optParams[0]
                d = optParams[1]
                S0 = optParams[2]
                posnFit = linspace(min(pulsePositions),max(pulsePositions),100)
                print 'Pulse Attenuation Length = %2.2f cm'%d
                print 'Attenuation/MaxSignal Ratio = %2.3f cm/V' %(d/S0)

        print "="*30

        #figure()
        #plot(pulsePositions, sigAmps,'bo')
        #plot(posnFit,relax(posnFit,A,d,S0),'k-')
        #xlabel('Position (cm)')
        #ylabel('Amplitude (V)')
        #title("500 mM CTAB in %s \n Relaxation Length = %2.3f cm and d/S0 Ratio =
%2.3f"%(datadir,d,d/S0))
        #show()
        return p[0], r_value

def runTemp(dir):
        os.chdir(dir)
        cs,R2 = doRun(dir)
        results.append([float(dir[1:]),cs,R2])
```

```
        os.chdir('../')

def cleanUp(results):
        os.chdir(rootdir)
        results = array(results)
        figure()
        temps = results[:,0]
        cs = results[:,1]
        plot(temps,cs,'-o')

figure()
for dir in dirs:
        runTemp(dir)
legend(loc=4)

os.chdir(rootdir)
results = array(results)
figure()
temps = results[:,0]
cs = results[:,1]
R2 = results[:,2]

#plot Cs
plot(temps,cs,'-o')
xlabel('Temperature (C)')
ylabel('Shear Speed (mm/s)')
title("500 mM CTAB: Shear Speed vs Temperature")

#R^2 plot
figure()
plot(temps,R2,'-o')
ylim([0.97,1.01])
xlabel("Temperature (C)")
ylabel("Correlation Coefficient (R^2)")
title("500 mM CTAB: Quality of Fit")
show()
```