The University of Maine

DigitalCommons@UMaine

Honors College

Spring 5-2022

Investigating 3D-Printability of a Maine-Based Bio-Ink

Jordyn Judkins

INVESTIGATING 3D-PRINTABILITY OF A MAINE-BASED BIO-INK

by

Jordyn Judkins

A Thesis Submitted in Partial Fulfillment
of the Requirements for a Degree with Honors
(Biomedical Engineering)

The Honors College

University of Maine

April 2022

Advisory Committee:
    Bashir Khoda, Assistant Professor of Mechanical Engineering, Advisor
    Stephen Abbadessa, Crosby Lab Manager
    Julie DellaMattera, Associate Professor of Early Childhood Education and
    Development
    Michael Mason, Professor of Biomedical Engineering
    Karissa Tilbury, Assistant Professor of Biomedical Engineering

ABSTRACT

Biofabrication is the process of creating complex biologic products, such as artificial tissues, from raw materials such as living cells, biomaterials, and molecules. This can be done using 3D printed bio-ink, which is a combination of biomaterials and cells. However, the bio-ink must be a shear thinning fluid to allow for high-resolution and continuous printing, but also demonstrate post-printing mechanical integrity to self-support the structure, which is challenging to achieve. The research conducted here investigates how to improve the mechanical functionality of bio-ink using additives available in Maine. Chitosan, sodium alginate, and TEMPO nano fibrillated cellulose were chosen as the candidate biomaterials due to their biocompatibility. The printability of the bio-ink can be determined by considering the rheological properties and printing parameters for numerous mixtures. This research focuses on how the mixture ratio affects the printability of the bio-ink, while also investigating the individual material contributions.

Rheological data of four ink compositions were compared, and a "design of experiments" approach was used to determine which hydrogel ink produced the smallest filament width, and therefore best quality, when printed. The four ink compositions used were 2:1:0.1 w/v%, 2:1:0.5 w/v%, 2:1.5:0.1 w/v%, and 2:1.5:0.5 w/v% of Alginate:TEMPO-NFC:Chitosan. A flow curve, amplitude sweep, and thixotropy test were conducted for each ink to gather viscosity and modulus values data, and all tests indicated that the tested inks would be successful in printing. Each ink was then 3D printed to analyze filament width, which revealed the ink of highest solid content resulted

in the smallest width. Lastly, the design of experiments approach was applied to filament width and viscosities to reveal chitosan changes had the most effect on filament width, but T-NFC changes had the most effect on viscosity changes. Equations were also developed that can be used to predict the outcome variables of inks that could be tested in the future.

TABLE OF CONTENTS

LIST OF FIGURES AND TABLES

CHAPTER 1: INTRODUCTION

1.1: Introduction

Research of hydrogels is currently a very popular topic among scientists, especially their application to tissue engineering. Hydrogels are 3D gel-like structures with predesigned patterns that serve as a scaffold for cell growth [1]. These structures can mimic the extracellular matrix (ECM) of organisms, which is the naturally occurring structure in organisms that mechanically and biochemically supports cells [2]. This environment replication is what makes hydrogels so popular, as they have the ability to regulate cell fate [1]. Hydrogels can be made of chemical additives, though for applications in living organisms it is much more desirable to determine the right combination of naturally derived materials to include within the hydrogel solutions [3]. These naturally occurring materials should be chosen based on their abilities to provide structural and mechanical support of a scaffold, while also mimicking the ECM and providing appropriate gel porosity needed for cells [4]. Hydrogels can eventually be applied to human tissue regeneration through the use of medical imaging devices and computer-aided design, or CAD, programs to fabricate customizable implantable scaffolds to precisely fit to the affected region of a patient [5].

One of the most common ways to fabricate hydrogels is through bioprinting. Bioprinting is defined as a computer-controlled 3D printing process to produce a 3D functional living tissue scaffold through controlled layer-by-layer deposition of the hydrogel material [2], [6]. In this way, hydrogels can be called "ink" or "bio-ink" in their

preprinted forms. For the printing process, the 3D structures are pre-designed patterns

generated using CAD programs [1]. Bioprinting is desirable for fabrication of hydrogels

as it provides spatial control and repeatability of material deposition [2]. There are many

types of bioprinting, including inject bioprinting, electro-hydrodynamic jetting,

extrusion-based bioprinting, and laser-assisted bioprinting [2]. The most popular among

these is extrusion bioprinting. In this type of printing, the biomaterials that make up the

ink can be dispensed through nozzles or needles connected to the ink reservoir [3].

Printing parameters such as speed, dispensing pressure, print speed, and printing height

can easily then be changed to accommodate types of inks and different patterns [3]. This

type of printing also has the capacity to print inks of varying viscosities and varying cell

densities [1]. Knowing there are many factors that can be changed to affect the outcome

of the hydrogels, it is important to find and control the physical properties of these

materials to lead to more successful printability and shape fidelity [3]. This can be done

through assessing filaments of the printed hydrogels, and by studying the rheological

properties of the materials.

<div align="center">1.2: Literature Influences</div>

1.2.1: Chemistry

For the purpose of this study, a hydrogel ink composed of 3 components was

used. The first of these components was sodium alginate. Natural polymers such as

alginate and collagen have already been reported as good framework materials [3].

Sodium alginate is a sodium salt derived from brown algae or brown seaweed and is

biocompatible [2]. However, it may be desirable to have hydrogels contain crosslinking

<div align="center">2</div>

properties in addition to being biocompatible [1]. Luckily, sodium alginate can serve as a post-crosslinking agent in the presence of a calcium chloride solution [1]. The calcium ions from the calcium chloride solution direct cross-linking of the carboxylate groups of the sodium alginate, which allows gelation to be reached without negatively impacting the biocompatibility [2]. This cross-linking adds needed stability to structures that are liquid or semi-solid [6]. A 4 w/v% calcium chloride solution has been used in studies looking at alginate-carboxymethyl cellulose hydrogels [2], therefore this same concentration was used for this study. The sodium alginate used here was obtained from Sigma-Aldrich.

The second component used in this study was TEMPO nano-fibrillated cellulose. This substance has been added to hydrogels to improve mechanical stability and cell growth properties [4]. Nano-fibrillated cellulose, or NFC, can be derived from plants, and was obtained from The University of Maine (Orono, ME, USA) for this study. The fibers of NFC have a high aspect ratio, as the diameter is in the nano-meter range, while the length is in the micrometer range [4]. However, these fibers can coagulate or become nonuniform within solution, so they can be modified into tempo-NFC, or T-NFC to prevent this [4]. To complete this modification, the NFC material is oxidized with 2,2,6,6-tetramethyl-1-piperidinyloxy (TEMPO), which adds a negatively charged carboxylate ion that has the capacity to disperse into the water at the individual fiber level due to the electrostatic repulsion effect [7]. This improves uniformity, homogeneity, dispersibility, and printability [4].

The third and last substance used was chitosan. Chitosan hydrogels have been studied, but it was found they have low mechanical properties [6]. Only the highly viscous samples are able to hold their shape, but the shape only stays for a few hours [6]. Combining chitosan with these other components may solve this issue. It is a copolymer of β-[1→4]-linked 2-acetamido-2-deoxy-D-glucopyranose and 2-amino-2deoxy-D-glucopyranose, which is usually obtained by alkaline deacetylation of chitin [8]. Chitin is the main component found in the exoskeleton of crustaceans, such as shrimp [8]. The exact chemical interaction between chitosan and the other components of this study is unknown, though it was thought that the chitosan particles may simply be suspended within the solution. The chitosan for this study was obtained from MP Biomedicals, Inc.

1.2.2: Rheology

Rheology is the study of describing and assessing the deformation and flow behavior of materials [9]. Viscosity is one of the most common ways to describe the flow of a material, as it quantifies the flow resistance caused by internal friction from molecules sliding past each other [9]. High viscosity values mean more internal resistance and therefore more resistive flow, such as honey. Lower viscosity values mean less internal resistance and therefore less resistive flow, such as water. Rheometers are the devices that can be used to measure viscosity, as well as a number of other rheological parameters [9]. A rheometer works by placing a small amount of the substance on the base, then lowering a geometric attachment to "squish" the substance to

a desired height, which is called the gap [9]. For gel-like substances, parallel plate attachments are most often used [9]. To take measurements, the attachment is then continuously rotated or rotationally oscillated [9].

In the context of rheology of hydrogels, it has already been found that bio-inks need to exhibit shear thinning behavior, high recoverability, and adequate yield stress [1]. Recoverability is the ability of a hydrogel to obtain its mechanical strength post-printing. Yield stress is the stress at which a fluid begins to flow or deform [3], therefore it may be beneficial to compare these values between different inks. High yield stress materials are generally brittle and difficult to reshape, but materials with lower yield stress can be more easily remolded, such as printing of hydrogels [3]. Other rheological properties that are often measured for extrusion printing are viscosity, elasticity [5], loss modulus, and storage modulus [10]. Elasticity is the ability for a deformed material to return to its original shape when the forces causing the deformation are removed [9]. The loss modulus, G", reflects the viscous behavior of a fluid, which describes the liquid-like state of the fluid [9]. The storage modulus, G', reflects the elastic behavior of a fluid, which describes the solid-like state of a fluid [9].

Bio-inks are considered non-Newtonian fluids, as their viscosities are a function of the force creating the flow, which means there is a complex relationship between the force and the flow [10]. This behavior can be described as shear thickening or shear thinning. A shear thickening flow is when there is an increase in fluid viscosity with increasing shear rate. A shear thinning fluid, which is desirable for bio-inks, is when the

fluid experiences a reduction in viscosity with increasing shear rate [10]. This means with increasing force, the fluid viscosity decreases so it should become more liquid-like and therefore flow more easily.

Rheological studies have been applied to many types of hydrogels. In a study involving methylcellulose/gelatin-methacryloyl (MC/GelMA), the inks tested were characterized in terms of yield stress, complex modulus, shear thinning, self-healing, time, and temperature sweeps [1]. A rheometer with con-plate attachments with diameter of 50 mm was used for testing [1]. A frequency sweep was completed over a range of 0.01-100 Hz [1]. The shear thinning of the inks was investigated over a shear rate ramp from 0.01 to 1000 1/s [1]. Recoverability was measured in two ways. The first was done by applying repetitive low strain of 1% for 1 min, then by high strain of 100% for another minute at 1 Hz frequency [1]. The second way was by measuring viscosity in 3 steps; at low shear rate of 0.1 1/s for 1 minutes, at high shear rate of 100 or 500 1/s for 5 seconds, then at low shear rate of 0.1 1/s again for 2 minutes [1]. Lastly, yield stress was measured by ramping the shear stress from 0.01 to 10000 Pa at 1 Hz frequency [1].

Another study looking at atelocollagen (AC) and native collagen (NC) used a controlled-stress rheometer with parallel plate geometry at 25 mm in diameter [3]. A steady shear sweep analysis was used to measure the viscosity at varying temperatures [3]. A dynamic frequency sweep analysis was completed to measure the frequency dependent G' and G" of the different hydrogels from 0.1 to 100 rad/s at 0.1% strain [3]. These tests allowed the G' and G" values to be compared to each other, which revealed the liquid or solid-like behavior of the gels, and gave insight into the shear thinning behavior of the gels [3].

In a study looking at sodium alginate and carboxymethyl cellulose hydrogels, a rotational viscometer was used to measure rheological properties of the inks [2]. The shear rate, shear stress, viscosity and percentage of torque were measured at various rotational rpm [2]. Gathering this data was able to reveal the shear thinning behavior of the inks, which means printing may be more successful [2]. Similar behavior was seen of viscosities measured using a rotational viscometer for sodium alginate, T-NFC, and carboxymethyl cellulose hydrogels [4]. It also has been found that materials with viscosities ranging from 30 mPa•s to around 6×10^7 mPa•s have been shown to be compatible with extrusion bioprinters [6].

Lastly, a study looking at chitosan and hydroxyapatite hydrogels used parallel plate geometry of 25 mm in diameter on a rheometer to gather rheological information [6]. Dynamic frequency sweeps were completed to study the G' and G" values in the range of 1-100 rad/s at 0.2% strain under constant temperature [6]. Viscosity of the gels was determined by dynamic measurement with a rotary setup by varying frequency sweeps from 0.1 to 100 1/s for 200 seconds [6]. Some experiments within this study were carried out in triplicate, with these data sets being expressed as the average ± standard deviation [6]. For this study, magnitudes of G' were higher than G" in all groups, which means the hydrogels exhibited viscoelastic gel behavior, or more solid-like behavior [6].

1.2.3: The Bioprinting Process

As previously stated, the most desirable way of 3D printing hydrogels is through extrusion bioprinting. This way of printing allows biological structures with higher densities to be printed in comparison to other ways of printing, and therefore is more suitable for tissue fabrication [10]. Extrusion printers have been shown to produce the

best results for printing artificial tissues, multicellular systems, cell-laden materials, and other high viscosity materials [10]. Extrusion printers can dispense the materials in one of two ways. One way is to use mechanical force where a piston forces the ink out of the nozzle or needle at the bottom of the reservoir, and the other way is to use pneumatic forces where controlled air pressure forces the ink to be dispensed through the needle or nozzle [10]. While the ink is dispensing, the printer moves at continuous speed. It should be noted that the ink needs to form a cylindrical fiber rather than a droplet at the needle tip to show sufficient shape for structural integrity post printing [1].

One of the advantages of extrusion printing is that there are many different parameters that can be set to fine tune the printed structures. These parameters include air pressure, printing speed, printing height, and nozzle or needle size [1], [2]. For MC/GelMA inks, a 10 mL syringe reservoir with a 225 micron stainless steel needle was used for printing, with an ink flow rate of 0.5 mL/h when conducting printability tests [1]. To optimize the 3D printing conditions of the MC/GelMA inks, different deposition speeds of 1, 2, 4, 6, and 8 mm/s were tested with varying pressures and varying nozzle heights of 0.2, 0.3, and 0.4 mm [1]. The nozzle height should be adjusted to be close to the needle diameter to attain better resolution of the printed filaments [1].

Filament fusion tests have been completed on sodium alginate-carboxymethyl cellulose gels, where the filament-to-filament distance ranged from 1-5 mm, the pressure was set to 8 psi, the print speed was set to 5 mm/s, the nozzle diameter was 410 microns, and the printing height was 0.7 mm [2]. The effects of changing each parameter were also tested for the alginate-cellulose inks, where pressures of 5, 6, 8, 10, 12, and 15 psi were used, and printing heights of 0.4, 0.7, 0.9, 1.1, 1.3, and 1.5 mm were used [2]. It also was

found in this study that the filament widths were greater than the nozzle diameter due to material expansion after printing [2], which could be expected for other inks tested.

In a study of gelatin-cellulose-alginate hydrogels, a 3D printer was used to print continuous, single later strands of each ink using nozzle sizes of 600, 800, 910, and 1270 microns [5]. These filament widths were measured by taking width measurements in Image J at random locations along the printed filaments [5]. Using Image J to assess printability has been used in other instances, such as with AC and NC inks, as well [3]. In a study of sodium alginate, carboxymethyl cellulose, and T-NFC inks, a single filament was also deposited to measure filament width, and these filament widths were used to study the effect of air pressure on the widths [7]. Knowing this, filament width could also be used to study the effects of other parameters as well.

The printing pattern itself can also be changed within the extrusion printing process to achieve different results. For oxidized sodium alginate inks, lattice structures of 7 rows by 7 columns were created, with dimensions of 12.6x12.6 mm [11]. However, for sodium alginate-carboxymethyl cellulose inks, prints of 1D lines, 2D grids, and 3D scaffolds were all printed [2]. The filament widths and the pore sizes were measured in Image J, where the measurements were taken at random locations and expressed as average ± standard deviation [2].

1.2.4: Design of Experiments Analysis

Setting up the experimental process for developing hydrogels can be difficult, as there are numerous ways of testing multiple factors affecting the outcome of the ink. In reality, there are an infinite number of experiments that could be conducted, as many parameters can be changed, and many different types of outcomes and measurements can

be taken. It is impossible to perform experiments one-by-one, or to try and test everything possible, therefore some sort of experimental model must be set in place [3]. For this study, a design of experiments model was applied. This type of model uses factors, or variables, that are changed to influence the outcome of the experiment, which is a measurable quantity [12]. For this study, only numerical factors were changed. In this model, there is also a large importance on completing the tests in a random order to avoid any unmeasured or uncontrolled disturbances that could impact the system [12].

When applying the design of experiments method, or DOE method, it is important to note that this is a better approach than changing one single variable at a time, as that process leaves undiscovered values behind, and may deceitfully lead to what seems like the optimal outcome, when in reality may be a suboptimal outcome [12]. The outcomes are also different depending on the order at which variables are changed, which is undesirable [12].

For the purpose of this study, the DOE method for changing 2 variables was used. In order to change these two variables, a low and high value for each variable is chosen [12]. An outcome that can be quantified is then chosen to be measured for each experiment [12]. **Table 1** shows this layout and reveals that the total number of experiments for this structure is 4.

Table 1. DOE method setup for changing two variables.

Experiment	Variable 1	Variable 2	Outcome
1	-	-	A
2	+	-	B
3	-	+	C
4	+	+	D

This setup allows the effect variable 1 has on the outcome, and the effect variable 2 has on the outcome to be quantified [12]. Most often in real systems, the effect of one

variable is different at different levels of the other variable, which means there is an interaction effect happening between the variables themselves [12]. In order to quantify these effects, determine which has the greatest effect on the outcome, and to obtain an equation to predict other outcomes not tested, a least squares model can be applied to the data gathered from the experiments [12].

Once completing the experiments, the variable data can be input as vectors in a coding program such as R, along with a vector containing the corresponding outcome data, then the least squares model function can be run with these vectors. The least squares model is just a way of finding a prediction equation that best fits a set of data points by minimizing the distances from the data points to the prediction line. By running the least squares function in R, a four-term equation is generated where the first term represents the baseline or intercept, and is just the average of all of the data points [12]. The second term represents the main effect variable 1 has on the outcome [12]. The third term represents the main effect variable 2 has on the outcome [12]. The fourth and last term represents how the factor interaction between the two variables affects the outcome [12]. A pareto plot, which is a type of bar graph, can also be generated in R to visually display the levels of impact each term in the predictive equation has on the outcome [12].

1.2.5: Challenges in Bioprinting

Though extrusion bioprinting is the most commonly used way to fabricate hydrogels, there are still challenges with this process. In order to achieve the optimal hydrogel, the ink must have high shape fidelity, good resolution, be biocompatible, and have the ability to support cells [3]. There is a lack of suitable inks with all essential characteristics needed for optimal hydrogels, as improving fabrication properties

negatively impacts the biological requirements, and vice versa [1]. Increasing the shape fidelity and mechanical strength of a hydrogel leads to a more unstable biological environment, but increasing the biological stability decreases the mechanical strength of the hydrogel [1], [3].

Single component hydrogels lack mechanical strength, and have unsuitable degradation rates compared to native tissues, but adding other components to increase mechanical strength has the possibility to decrease the likelihood of cell survival [1]. Hydrogels also need to be designed in a way that lower levels of the scaffold are mechanically stable enough to support the higher layers once the entire structure has been printed [1]. Lastly, synthetic polymers allow for more control of the mechanical strength, but have poor biocompatibility, while hydrogels made from natural biomaterials provide optimal environments for cell survival, but have poor physical properties [3].

<p style="text-align:center">1.3: Objective</p>

Knowing the challenges of developing hydrogels, the objective for this study was to investigate the printability of a hydrogel composed of naturally occurring materials that could be obtained from the state of Maine. To do this, preliminary experiments were created to serve as a "baseline" for developing a more structured set of experiments. This structured set of experiments followed a DOE setup, where the concentrations of T-NFC and chitosan were the changing variables, and the concentration of sodium alginate and all other printing parameters were kept constant.

For each structured test, the ink was 3D printed so filament widths could be measured to quantify the effects of the changing concentrations. Rheological tests were

performed to gain insight into the flow behavior of the inks relative to each other, and to further quantify the effects of the changing concentrations. The objective of obtaining these results was also to determine which ink produced the smallest filament, and to aid in deciding on a direction for future experiments.

CHAPTER 2: METHODS

2.1: Choice of Materials

A successful hydrogel ink can only be formed with an appropriate selection of materials. These materials need to be chosen based on the desired mechanical and biological properties. As stated previously, sodium alginate, T-NFC, and chitosan were chosen for the hydrogel components. It is known that many successful hydrogels contain additives such as polymers, particles, fillers, and fibers [1], and the components chosen here fall into these categories. First and foremost, these three hydrogel components were chosen because they are all naturally occurring materials. Their locations in nature mean they theoretically could all be obtained within the state of Maine. Local obtainability would mean greater ease of future production.

Alginate was chosen due to its biocompatibility, viscosity properties, low price, and fast gelation rate through the almost instant sodium calcium ion exchange that takes place at room temperature [6]. Alginate also has the ability to be modified for a variety of tissue engineering applications, such as bone, adipose, and vascular tissues [11]. T-NFC has been used in instances with sodium alginate, and was chosen here knowing it would enhance the mechanical properties of the hydrogels for better printing, and can also aid in cell survival [5]. Chitosan was chosen as it has already been used in bone, skin, and cartilage tissue engineering [6]. Its ingredients also resemble the extracellular matrix components of native tissues [6]. Lastly, chitosan can also be metabolized by certain enzymes, meaning it can be considered biodegradable, can be bioadhesive, can promote wound healing, and has bacteriostatic effects [8].

2.2: Preliminary Experiments

The materials chosen for this study do not have a set mixing method, therefore a few preliminary hydrogel inks were formed. Based on the results of each preliminary experiment, the following experimental conditions were modified in attempt to find better experimental parameters for a set of structured tests. The inks formed in these preliminary experiments were based on keeping the solid contents low enough that printing may be possible, but also high enough so the mixture may remain sturdy after printing. To label each mixture, a ratio format was used where the first number represents the weight volume percentage of sodium alginate (Alg), the second number represents the weight volume percentage of tempo-NFC (T-NFC), and the third number represents the weight volume percentage of chitosan (Ch).

The first preliminary experiment was a 3:1:1 w/v% mixture. A premade 1 w/v% T-NFC slurry was used as the base for this mixture, as the T-NFC was already at the desired concentration. Twenty-five milliliters of T-NFC, 0.75 g of alginate, and 0.25 g of chitosan was measured out. The T-NFC slurry was set on a magnetic mixer at 120 rpm. The chitosan powder was added in very small amounts over 20 minutes, and a stir rod was used to push big clumps of chitosan into the slurry. With the speed still at 120 rpm, the sodium alginate powder was also added in very small amounts over 20 minutes, and a stir rod was used to push big clumps of alginate into the slurry. For overnight stirring, the speed was turned up to 140 rpm. After 18 hours of stirring, the ink mixture was removed from the magnetic stirrer and manually syringed by hand to test the viscosity.

To manually syringe the ink, a 3 mL syringe was used with a 610 micron needle. A 4 w/v% solution of calcium chloride was made with a base of distilled water to be used

15

as a sprayable crosslinker. To start, a single line of ink was syringed and sprayed with the crosslinker. As this line was successful without a huge amount of spreading, a single layer hashtag pattern was syringed, as well as a triple layer hashtag pattern to compare height differences. These hashtag patterns were also sprayed with the crosslinker immediately after the ink was dispensed.

The next preliminary experiment performed was done with a 3:1:2 w/v% ink mixture. Again, a premade 1 w/v% T-NFC slurry was used as the base for this mixture. Twenty-five milliliters of T-NFC, 0.75 g of alginate, and 0.5 grams of chitosan were measured out to obtain the desired concentrations. The T-NFC slurry was set on a magnetic stirrer at 120 rpms and the same adding process for the alginate and chitosan that was done for the previous ink was used. However, once all additions had been made, the increase in chitosan concentration made the solution visibly thicker, therefore having the stir bar turned too high resulted in the bar actually stopping its stirring motion, and instead just vibrated in place. Due to this, the solution was left at a different overnight stirring speed compared to the previous ink of 120 rpm.

After about 20.5 hours of stirring overnight, the solution was removed from the stir plate and manually syringed to ensure this was still possible even with a higher chitosan concentration. A 330 micron needle was used. The solution was only tested to see if it could be syringed into a straight line, just as proof that it moves through the needle easily. Once dispensed from the syringe, the line was sprayed with the same 4 w/v% calcium chloride solution to initiate crosslinking and prevent filament spreading.

A print test of the 3:1:1 w/v% ink was completed in preparation for rheological testing. The 410 micron needle was used for this test. Instead of manually syringing the

ink by hand, the ink was loaded into a syringe attached to the 3D printer. A hose was then attached to the syringe to supply pneumatic pressure in order to dispense the ink. Once the 3D printer started running, the pressure was continuously adjusted to allow ink to flow onto the printing base.

In preparation for creating a structured series of tests, the 3:1:1 w/v% ink was tested in the rheometer. All rheometer tests were done using a parallel plate attachment with a diameter of 50 mm. The first test run that could produce meaningful results was a flow curve. This was done with a gap of 0.5 mm at a constant 25°C. All other parameters of the test were left as the defaults. Next, a frequency sweep was completed. The gap used for this test was also set at 0.5 mm with a constant temperature of 25°C. All other values were left as their defaults. Lastly, an amplitude sweep was conducted with a gap of 0.5 mm at a constant temperature of 25°C, with all other values left as their defaults.

The last preliminary ink made was a 2:1:1 w/v% ink. The same premade 1 w/v% T-NFC slurry was used as the base for this ink mixture. Fifty milliliters of T-NFC, 1 g of sodium alginate, and 0.5 g of chitosan were measured out. The T-NFC slurry was set on the stir plate at 120 rpm. Small amounts of chitosan were added continuously over 20 minutes. After each addition, a stir rod was used to push the chitosan below the solution surface for better mixing. The sodium alginate was also added in small amounts over the 20 minutes, and again, the stir rod was used to push the alginate below the solution surface after each addition for better mixing. The solution was left to stir overnight at 120 rpm.

This 2:1:1 w/v% ink was also used to continue discovering what rheometer tests would produce meaningful results. A flow curve was conducted to look at the ink's

viscosity. The time was ramped logarithmically from 120 seconds to 1 second, and the

shear rates were set from 0.01 1/s to 1000 1/s. A gap of 0.5 mm was used, and 3 sets of

the flow curves were collected. A frequency sweep was also completed, with a strain rate

value of 0.5%, a 0.5 mm gap, and a range of 0.1 to 100 rad/s. An amplitude sweep was

conducted at a frequency of 10 rad/s, a gap of 0.5 mm, with a strain rate range of 0.1 to

100%. Lastly, a thixotropy test was conducted to gain insight into how the ink behaves

under printing conditions, and how it recovers after printing conditions. This test was

done with a 0.5 mm gap, but all other values were kept as their default values.

This same 2:1:1 w/v% ink was 3D printed to investigate how well this solid

content amount would print. The ink was loaded into the syringe attached to the printer.

The hose supplying pneumatic pressure was attached to the syringe, allowing for

continuous control of the air pressure. A 250 micron needle, a 330 micron needle, and a

410 micron needle were all tested with varying pressures to observe how well the ink

dispensed onto the syringe plate.

2.3: Ink Mixing

Based on the preliminary experimental results, a structured mixing and testing

protocol was developed to investigate printability and rheology of four ink mixtures.

These mixtures were 2:1:0.1 w/v%, 2:1:1.5 w/v%, 2:1.5:0.1 w/v%, and 2:1.5:0.5 w/v%.

Distilled water was used as the base for the preliminary experiments, therefore distilled

water was chosen for the base in the structured experiments as well. Distilled water has

also been used in other studies looking at sodium alginate hydrogels [11]. To ensure

homogeneous mixing, the same overnight mixing process was used, as this mixing time

produced homogeneous inks in the preliminary experiments. This overnight mixing process has also been used in studies of sodium alginate-carboxymethyl cellulose hydrogels and produced very homogeneous mixtures [2], further reinforcing the need for it to be applied to this study. Due to the T-NFC being in powder form, it was decided that it would be best to create a T-NFC slurry first to then add the other components to.

To begin, the desired amount of T-NFC was weighed out to create a 30 mL ink mixture. Thirty milliliters of distilled water was added to the T-NFC powder in a 100 mL beaker. The beaker was then set on a stir plate, covered in parafilm, then the stir plate speed was set to 1000 rpm. This slurry was then left to stir overnight for about 24 hours. Next, the desired amounts of alginate and chitosan were weighed out for the 30 mL mixture. Once turning the stir plate to 500 rpm, chitosan was added in small amounts over the span of 15 minutes. A stir rod was used after each addition to force the chitosan particles under the solution surface for better mixing. The sodium alginate was then also added in small amounts over the span of 15 minutes. A stir rod was again used after each addition to force the alginate particles under the solution surface for better mixing. Once all additions were made, the ink mixture was left stirring at 500 rpm for about 24 hours. This mixing process was repeated for each of the 4 inks. An example of an ink mixture on the stir plate is shown in **Figure 1**.
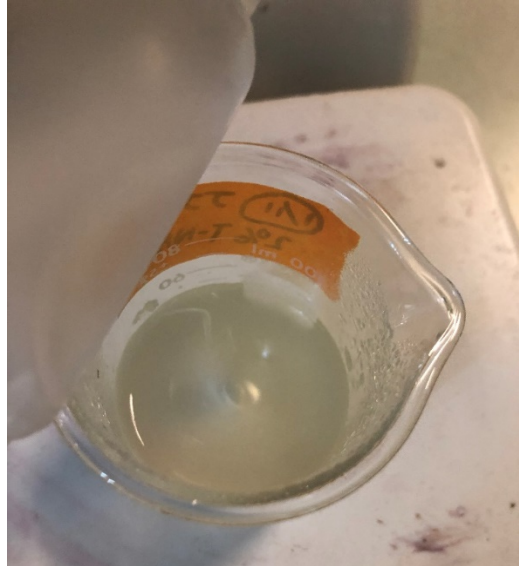
Figure 1. Example of ink on stir plate; the ink shown is the 2:1.5:0.5 w/v% ink.

2.4: Rheological Testing

For each of the four chosen inks, 3 different rheological tests were completed using an Anton Paar rheometer, and each test was completed 3 times to allow for average curves for each test to be generated. For each ink, a small amount was placed on the center of the rheometer base. An example of the ink placed on the rheometer and ready for testing is shown in **Figure 2**. A parallel plate attachment with a diameter of 50 mm was attached and used for all tests. First, a flow curve was conducted to gain insight into how the viscosity of each ink would change with increasing shear rate. This flow curve measured 21 points ramped logarithmically from a 120 second interval to a 1 second interval. The shear rate was ramped logarithmically from 0.01 1/s to 1000 1/s. To make sure the gap was larger than the size of the particles, a gap of 0.5 mm was used.

An amplitude sweep was conducted also with a 0.5 mm gap. This test was completed to gain insight into the storage and loss moduli of each ink to better

understand their solid or liquid-like behavior. The amplitude sweep measured a total of 25 points, and the shear strain was ramped logarithmically from 0.01 to 100%. A constant frequency of 10 rad/s was used. Lastly, a thixotropy test was conducted. This test was done over 3 different intervals, where each interval represented a part of the printing process. This would allow information to be gathered about how the ink might behave during the printing process, and how well it would recover after printing. The first interval measured 10 points, with each point taken at equal spacing of 6 seconds for a total interval time of 60 seconds. The shear rate used for this interval was 0.1 1/s. The second interval also measured 10 points, with each point taken every 1 second for a total interval time of 10 seconds. The shear rate used for this interval was 100 1/s. Lastly, the third interval measured 60 points, with each point taken every 2 seconds for a total interval time of 120 seconds. The shear rate used for this interval was 0.1 1/s.
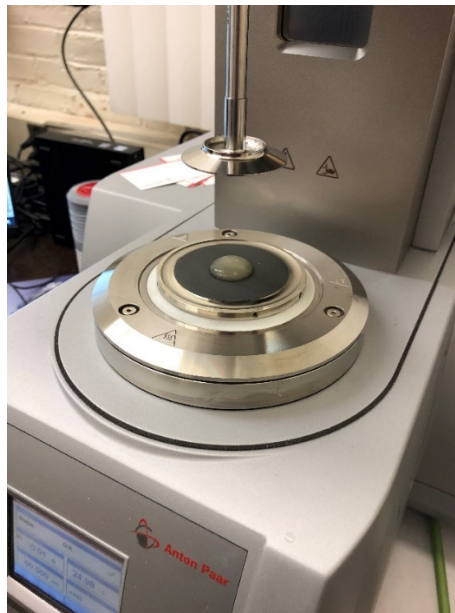
Figure 2. Setup of rheometer with ink placed on base ready for testing.

2.5: 3D Printing

Before beginning the printing process, a print pattern was created to allow the ink to print in a straight line for later filament width analysis. However, to allow the ink to reach a steady state flow while printing, the pattern created was a "back-and-forth" set of many lines. The pattern was created in a CAD software called Rhino. This "back-and-forth" pattern spanned an area of 5 cm by 5 cm, with a 5 mm distance between each of the center of the lines. Once created in Rhino, this pattern was converted to a vectorized code written in a basic scripting language. This code is called the g-code. This code could then be read by the printing software called Flashcut CNC. A three-axis 3D printer was used to print this line pattern. For each ink, a small amount was placed into the syringe attached to the printer, and a hose supplying pneumatic pressure was hooked to the end of the syringe. A 410 micron needle was used for all four inks tested. A side view of the printer setup is shown in **Figure 3**, and an angled view with an example of a needle attached to the syringe is shown in **Figure 4**.
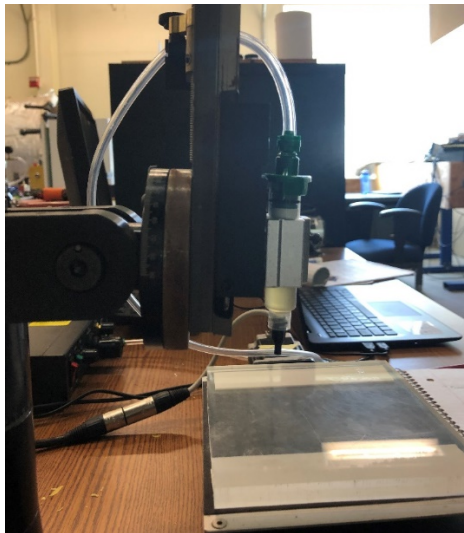
Figure 3. Setup of printer without a needle showing tubing for pneumatic pressure, syringe attachment to printer, and printing base.
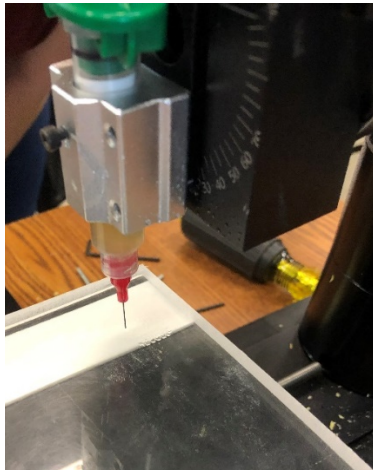
Figure 4. Setup of printer with needle.

A pressure of 10 psi was used during the printing process. The speed was left as the default in Flashcut CNC, which was at a feed rate of 1000. A printing height, which is the perpendicular distance from the printing base to the needle, was set to 0.3 mm for all prints. A labeled printer diagram showing this distance can be seen in **Figure 5**. Due to occasional clogging, the printing process was repeated until at least 3 good lines were produced to allow for later analysis. Once printed, each structure was lightly sprayed with a 4 w/v% calcium chloride crosslinker to strengthen the filaments and prevent spreading. The line structures were then imaged using a Sony 4K Handycam.
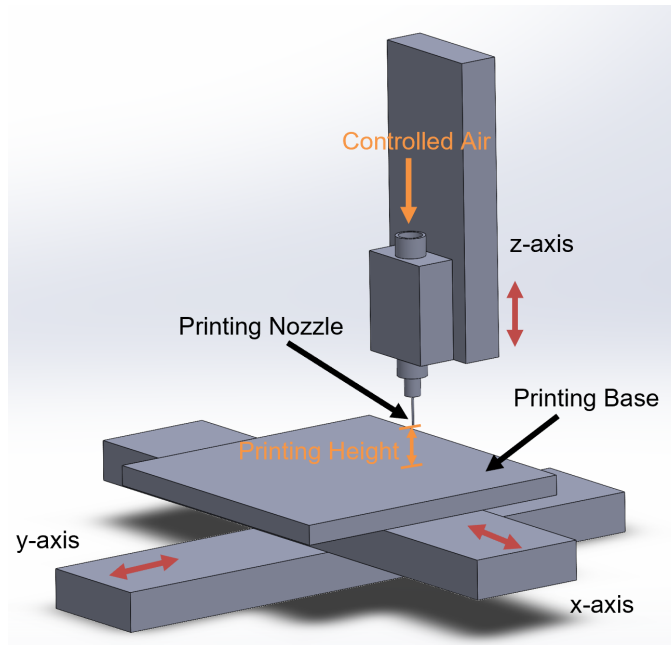
Figure 5. Labeled diagram of 3D printer setup.

2.6: ImageJ Analysis

To analyze the images, the open-source software Fiji Is Just ImageJ (Fiji) was used. This software is also referred to as Image J. By using this software, the macro recorder plugin could be used to generate the program script, and writing the program script allowed for an almost automated process to be created so future images could be analyzed in the exact same manner. To analyze data output from Image J, the program R was used. Using this program allowed CSV files to be read and data from these files to be manipulated to produce the desired end results.

Before any analysis was completed, the 2:1.5:0.5 and 2:1:0.1 images were rotated slightly to ensure the filaments were as straight as possible for later analysis. Before analyzing the filaments themselves, the ratio of pixels to millimeters needed to be determined. When printing the filaments, the 4 parent images were taken with a ruler

included in the picture. By doing this, an automated method could be created to determine how many pixels are in a millimeter, which allows the final filament widths to be expressed in terms of millimeters rather than pixels. To do this, a small line was first drawn across the ruler. The profile of this line was then plotted, and the "find peaks" function under the BAR plugin was used to find the locations of minimums in Image J. This data was then output to CSV files. The ruler tick marks are dark in color, therefore finding the distance in pixels between the tick marks allows the number of pixels in a millimeter to be found. An example of what the line, plotted profile, and find peaks functions look like is shown in **Figure 6**.
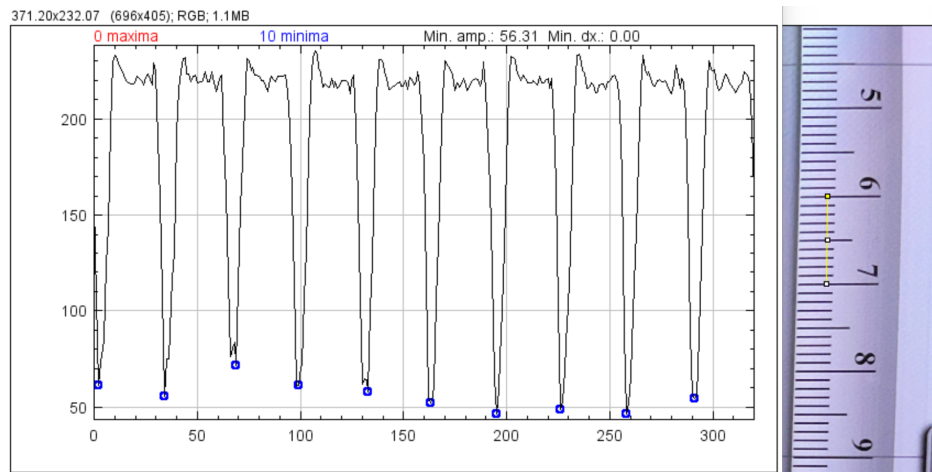
Figure 6. Example of ruler line and find peaks from the 2:1.5:0.1 image.

The CSV files were then opened and read in R. Because the line was drawn over many tick marks, the R program was written to only pull the locations of the minimums, organize them from lowest to highest, then find the average distance between the tick marks and print this as the pixels/mm value. The location of the ruler within each parent image is different, therefore this process was completed for each parent image separately, with the line location changing each time. The sections of the Image J and R codes that complete these tasks are commented on and can be seen in **Appendices A** and **B**.

Because the locations of the filaments are different for each parent image, the plotting lines process had to be altered slightly for each image. In order to plot lines over filaments in each image, rectangles were drawn to isolate the section of the filaments to be analyzed, and running these rectangles through the "internal clipboard" allowed them to each become their own image. A large for loop was created to run through the selection of these rectangles automatically. However, this for loop had to be created for each image due to the fact that the starting place of the rectangles changed between images, and the number of viable filaments to get measurements from also changed between images. The rectangles are all the same size for each image, with the length being 640 pixels and the height being 140 pixels. The placement of the rectangles was also determined based on where they could be drawn with the least amount of glare from the parent image included. Including glare in the images would then throw off the pixel values found later.

Once a rectangle is selected and made into its own image, it is converted to a black and white image, and another for loop allows for 31 lines spaced 20 pixels apart across the filament to be drawn, and the plotted profile values to be printed and saved to a CSV file for each line. These lines are drawn perpendicular to the direction of the filament, and span the full height of the image. An example of what a filament image with a line drawn on it is shown in **Figure 7**. The section of Fiji code that completes these 2 for loops for the rectangles and the line profiles for each image can be seen in **Appendix A**.
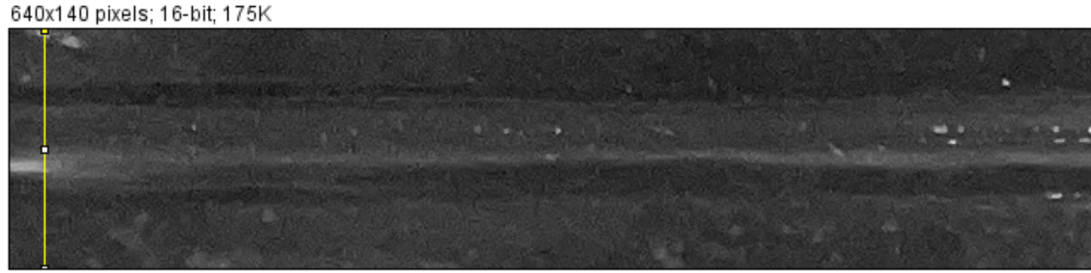
Figure 7. Example of the first filament and first line drawn from the 2:1.5:0.1 image.

Each parent image had a different number of viable filaments to measure from, among other unique properties, therefore the analysis of the plotted profiles in R needed to be done for each image, with certain values slightly altered for each image. A large for loop was created for each image to loop through the data for each filament. Within this large for loop, a smaller for loop was created to read the data from each of the 31 lines. The pixel values for each line were extracted vectors, therefore all 31 vectors were combined in a single large array. From here, the average pixel value across the array for each location was found using the "rowmeans" function.

The height of the rectangles may have been the same for all images, but this means the edges of the filaments were at different locations for each image. The bio-inks were printed on a dark background, therefore it was assumed that the edges of the filaments are the lowest pixel values in the area, as this is where the edges should be. To prevent catching low values from the crosslinker droplets, the ranges in which to search for the two lowest pixel values for each parent image were altered for each image by eyeballing the area on the filament image where the edge of the filament should be. In R, the lowest point in each range of the average pixel values was found, and the distance between these 2 lowest points for each filament was found. These values were also printed in R.

27

Finding the average of the filament distances found for each image then resulted in an overall average filament width given in pixels. The previously found pixels/mm value was then used in R to convert the overall average width in pixels to the overall width in mm. The sections of the R code that find these averages and standard deviations is commented on and can be seen in **Appendix B**.

2.7: Design of Experiments Analysis

To run the DOE analysis, R was used. The code version of the T-NFC values and the Ch values were input as vectors of -1, +1, -1, +1 and -1, -1, +1, +1, respectively. The -1 and +1 represent the concentrations of 1 w/v% and 1.5 w/v% of T-NFC in its vector, and the -1 and +1 represent the concentrations of 0.1 w/v% and 0.5 w/v% of Ch in its vector. An outcome vector containing the outcome values was created for each case. The first case was a vector of the filament widths. The second case was a vector of the initial viscosity values from the average flow curves. The third and final case was a vector of the final viscosity values from the average flow curves. These vectors were then input into the least squares model function in R to generate the least squares model equation values, then a pareto plot was generated for each case.

CHAPTER 3: RESULTS AND DISCUSSION

3.1: Preliminary Experiments

The first ink of the preliminary experiments to be tested was the 3:1:1 w/v% Alg:T-NFC:Ch ink. This ink was first syringed by hand in a single line. This produced a somewhat smooth flow with minimal clogging, though the needle size was large at 610 microns. Next, a single layer hashtag pattern was syringed. The center hole of the structure stayed open but did decrease greatly in size, and the flow of ink remained smooth. Lastly, a triple layer hashtag was completed. The center hole of this structure was very close to filling in completely. Looking at the heights of the 3 structures, the triple layer hashtag did have the largest height, which was encouraging that future inks would also have the ability to increase in height without large amounts of the material spreading out. **Figures 8** and **9** show the top and side view of these structures. Due to the smooth flow of these tests, the concentration of chitosan was increased for the next experiment.

Figure 8. Syringed structures of 3:1:1 w/v% ink; line is at the top, the single layer hashtag is in the center, and the triple layer hashtag is at the bottom.
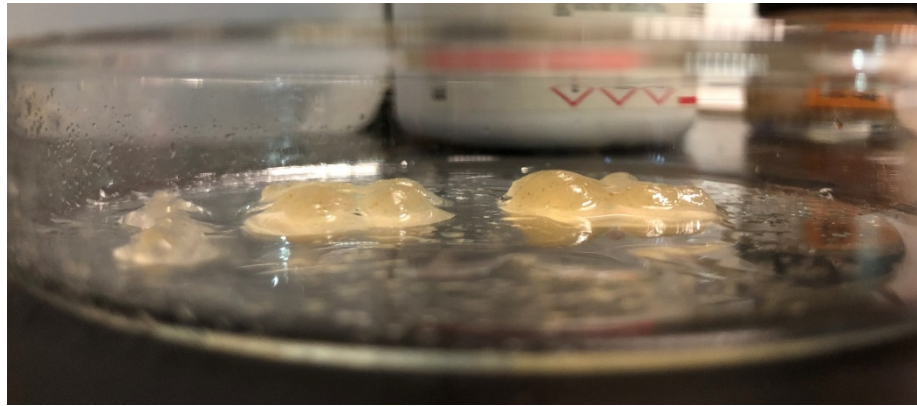
Figure 9. Syringed structures of 3:1:1 w/v% to compare heights; line is at the far left, the single layer hashtag is in the center, and the triple layer hashtag is at the far right.

The next ink syringed by hand was the 3:1:2 w/v% ink. When syringed by hand, clogging occurred much more often than with the previous ink, though the needle size was at a smaller 330 microns. Knowing that the filaments should be as small as possible once reaching the point of being ready to print, it was decided this really wouldn't be a useable ink since it clogged in a 330 micron needle, so no further testing was completed.

The 3:1:1 w/v% ink was tested in the 3D printer to see how the results improved from the syringed results. In order for any ink to flow through the needle, the pressure needed to be around 25 psi. However, clogging occurred very frequently. At every clogging point, the pressure was turned to about 25 psi to attempt to clear the clog, but this resulted in a large blob of material being dispensed until the pressure was turned back to 25 psi. Due to this, the next ink tested had a smaller solid content of sodium alginate in attempts to prevent clogging.

The 2:1:1 w/v% was print tested using multiple needle sizes. A 250 micron needle was used first, but barely any gel was dispensed, even when the pressure was turned up to 50 psi and higher. A 330 micron needle was tested, and a little bit more of the ink was dispensed, but clogging occurred so frequently that no smooth lines could be created. The

pressure also had to be turned up to 50 psi to dispense any ink. A 410 micron needle was tested last, but there wasn't much improvement from the 330 micron needle test. The pressure had to be between 25 and 50 psi to dispense any ink. All clogs that occurred in each of these tests could not be removed by increasing the printing pressure; they all had to be removed by hand. After seeing these results, it was deciding that future inks would need to be created at even lower solid contents or at different ratios to produce printable inks.

The 3:1:1 w/v% and 2:1:1 w/v% inks were used to aid in deciding on what rheological tests may be useful to complete along with knowing what tests have been used in past literature. A flow curve was conducted for both inks, which showed how the viscosity behaved with increasing shear rates. These inks showed shear thinning behavior as the viscosity decreased with increasing shear rates, therefore it was decided it would also be useful to obtain the flow curves of future inks. An example of the flow curve from the 2:1:1 w/v% ink is shown in **Figure 10**.
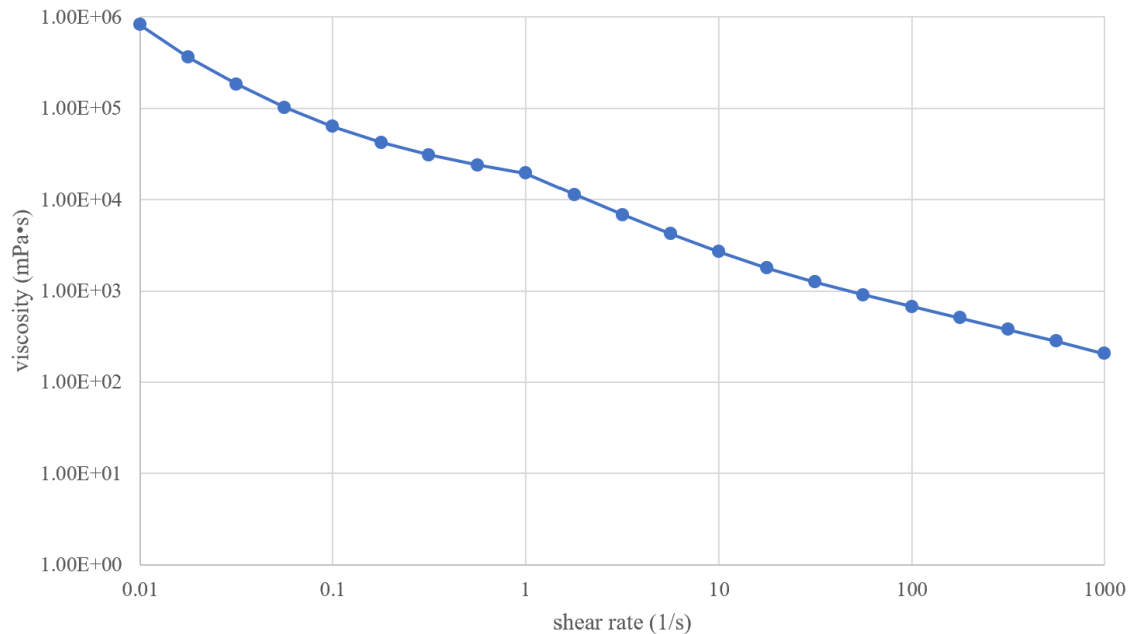
Figure 10. Flow curve of 2:1:1 w/v% ink with logarithmic axes.

Next, a frequency sweep and amplitude sweep were conducted for both the 3:1:1 w/v% and 2:1:1 w/v% inks. The amplitude sweep was very successful, as both the G' and G" values for the inks could be seen, and the point at which the flow went from solid-like behavior to liquid-like behavior could be seen as well, which was the point at which the G' and G" lines intersected. However, the frequency sweep showed the G' and G" values, but this intersection point could not be seen. Based on this, it was decided the amplitude sweep would be conducted for the future inks, but no further frequency sweeps would be collected. **Figure 11** shows as example of the amplitude sweep from the 2:1:1 w/v% ink, and **Figure 12** shows an example of the frequency sweep from the 2:1:1 w/v% ink.
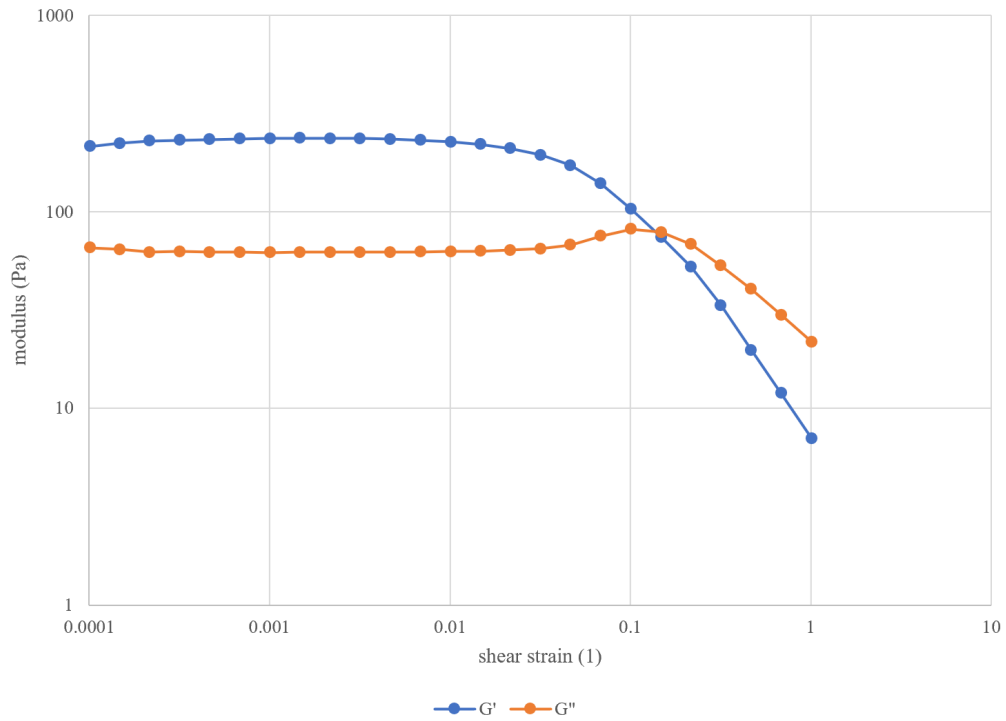
Figure 11. Amplitude sweep of 2:1:1 w/v% ink with logarithmic axes.
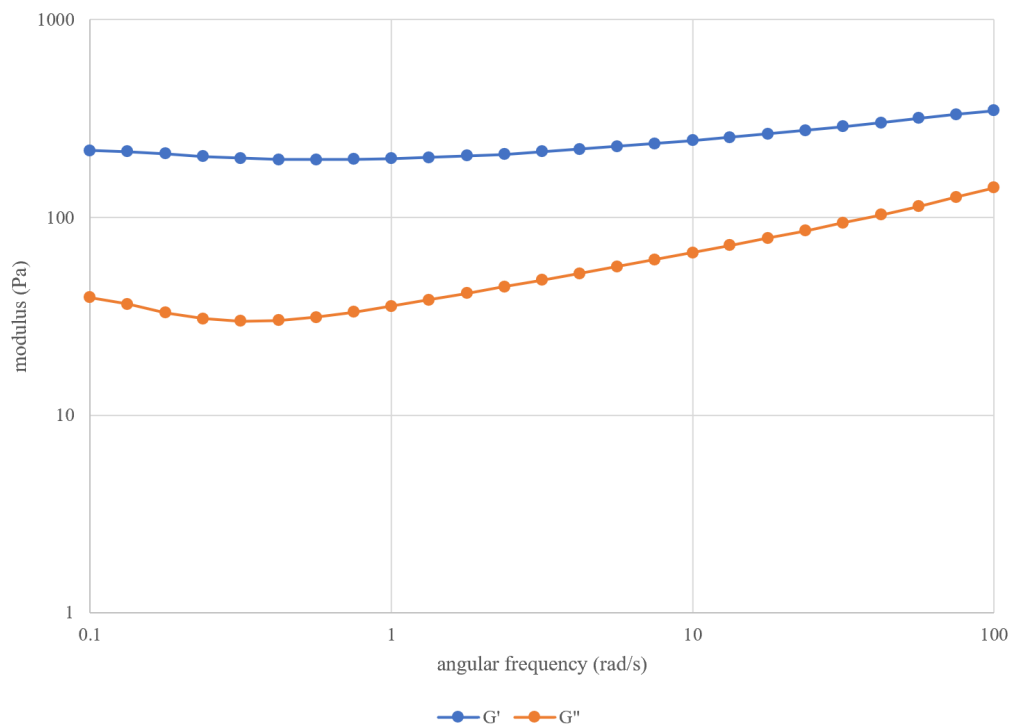
Figure 12. Frequency sweep of 2:1:1 w/v% ink with logarithmic axes.

33

Lastly, a thixotropy test was only completed for the 2:1:1 w/v% ink. The data showed viscosity behavior within each interval, and showed the viscosity slowing trying to reach its initial values in the third interval, or during the recovery period. It was decided this would also be useful data to gather for future inks. **Figure 13** shows the thixotropy data collected for this ink.
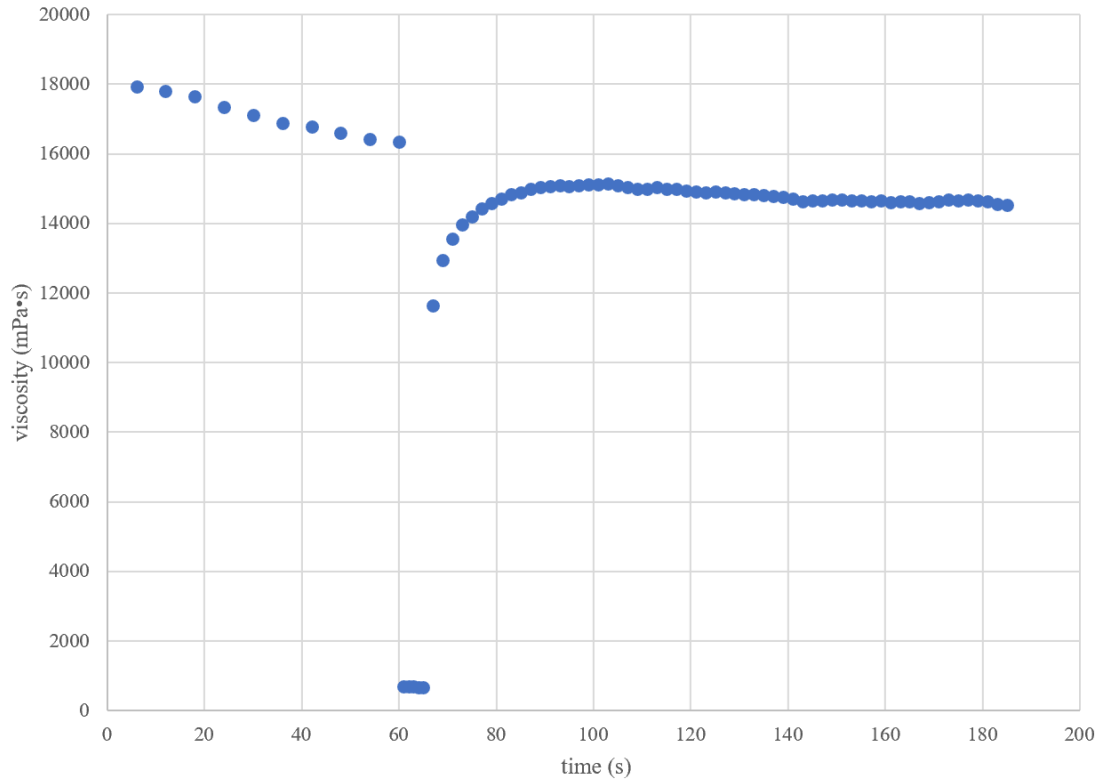
Figure 13. Thixotropy test of 2:1:1 w/v% ink with logarithmic y-axis.

3.2: Rheometer Results

For each of the inks tested in the structured set of experiments, 3 flow curves were generated and the data averaged together to get an average flow curve for each ink. **Figure 14** shows this data. It is seen that for all inks, the viscosity decreases with increasing shear rate. This is evidence of shear thinning fluids, which is desirable for

printing. During printing, the pressure places force on the ink in the syringe, and if the ink is shear thinning, its viscosity will decrease so as to allow it to flow and be dispensed through the nozzle. These curves also show the ink of highest solid content had the highest viscosity, which was expected. The inks with 1 w/v% of T-NFC seem to be close together, with a big jump to reach the inks with 1.5 w/v% of T-NFC, which are also close together. The chitosan concentration was only changed 0.1 w/v% less than the T-NFC was changed for this structured set of experiments, therefore this pattern in the curves suggests that the effect of T-NFC on the viscosity is higher than the effect of chitosan. The viscosities of these inks range from 125 mPa•s to 4.4×10^5 mPa•s, which falls in the range of 30 mPa•s to 6×10^7 mPa•s that is already known to be compatible with extrusion printers, further reinforcing these inks should print successfully.
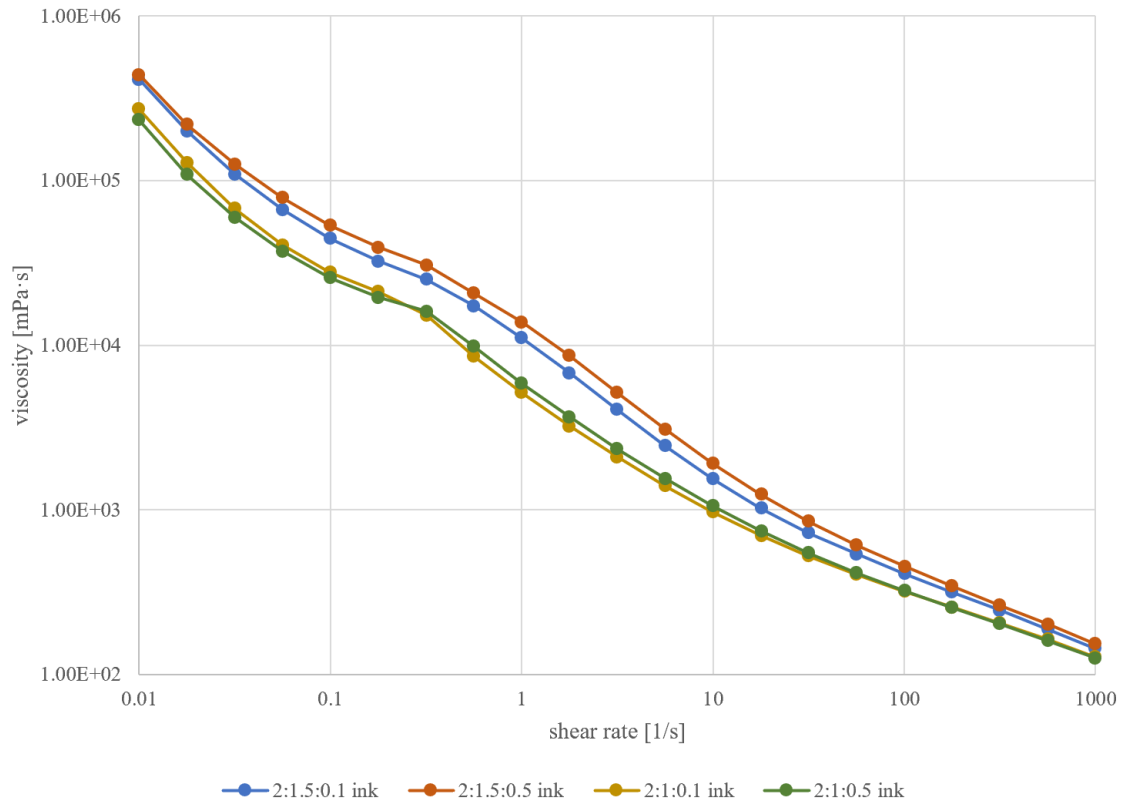
Figure 14. Average flow curves for the inks with logarithmic axes.

35

Next, three amplitude sweeps were conducted for each of the four inks, then these curves were averaged to obtain an average amplitude sweep for each ink. **Figure 15** shows this data. For all inks, at lower shear strains the G' value is higher than the G" value, indicating all inks exhibit more solid-like behavior at these values, and when untouched. As the shear strain increases, all inks have a crossover point between the values so the G" becomes larger than the G' value, meaning the inks have reached points at which they exhibit more liquid-like behavior. It is essential for these inks to exhibit liquid-like behavior, as this means they should be able to flow through a 3D printer needle at a certain point. Similar to the flow curves, the G' and G" curves are grouped based on the T-NFC percentages, suggesting that the T-NFC concentration changes affects these modulus values much more than the chitosan concentration changes.
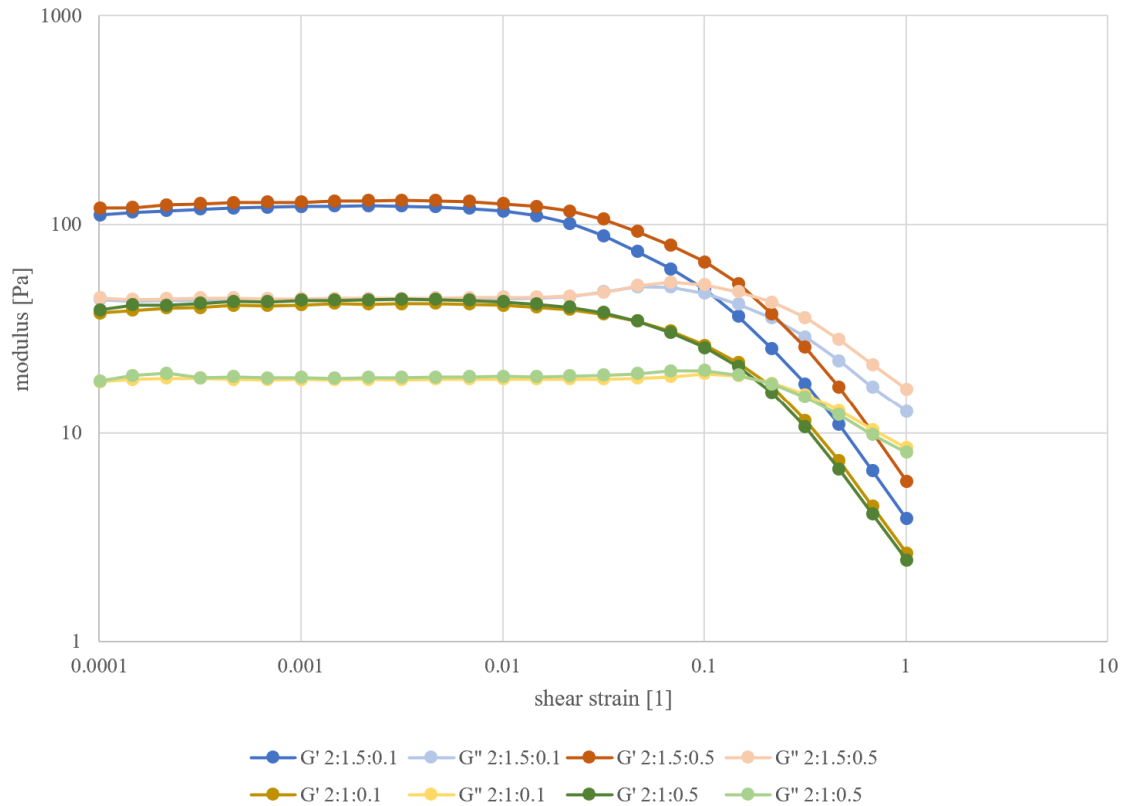
Figure 15. Average amplitude sweeps for the inks with logarithmic axes.

Lastly, three thixotropy tests were completed for each of the four inks, then this data was averaged to obtain an average thixotropy data set for each ink. **Figure 16** shows this data. The first interval was taken at a low constant shear rate, as this mimics the ink just sitting in the syringe without any applied pressure, waiting to be printed. The viscosities of all inks in this region are fairly constant, which was expected. The second interval was taken at a constant high shear rate, as this mimics the ink being pushed through the printer needle. All inks exhibit a large drop in viscosity, which suggests their flow properties are conducive for printing. The third and last interval was taken at a constant low shear rate, as this mimics the hydrogel sitting untouched on the printing base post-printing. All inks exhibit a large increase in viscosity at the start of this interval, meaning all inks are trying to move back to their original viscosity once the increased force has been removed. All inks also exhibit smaller increases in viscosity as they continue to move towards their original viscosities.
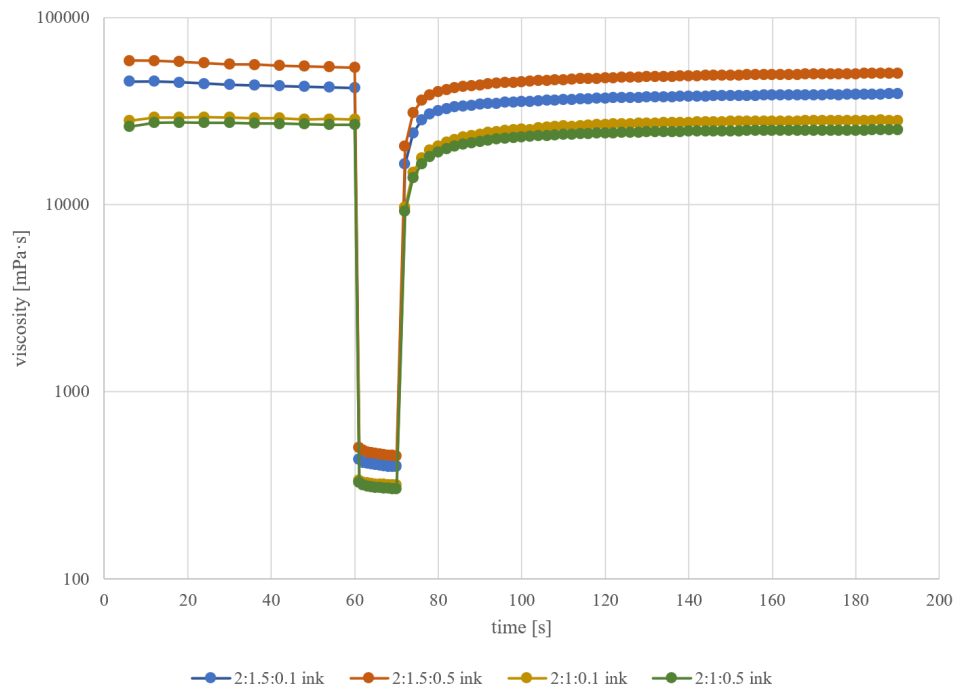
Figure 16. Average thixotropy data sets for the inks with logarithmic y-axis.

To see the percentage at which each ink was able to recover to the initial viscosity values within the 2 minutes of recovery time, the last recorded initial viscosity and the last recorded recovery viscosity were compared. **Table 2** shows this comparison, along with the percentage of the initial viscosity the recovery viscosity was able to reach. As shown in the table, all inks were able to recover more than 90% of the initial viscosity which means any fluid flow seen before printing would be very similar to how the filaments would spread after printing.

Table 2. Comparison of initial and recovery viscosities.

Ink	Initial Viscosity [mPa•s]	Recovery Viscosity [mPa•s]	Percentage of Initial Viscosity Recovery Viscosity Reached
2:1.5:0.1	41975.67	39029	93%
2:1.5:0.5	54150	50333.67	93%
2:1:0.1	28578.67	28133.67	98%
2:1:0.5	26758.33	25028	94%

Because each rheological test was taken in triplicate for each ink, all graphs are expressed as average curves. Due to the small standard deviations, the standard deviation bars are not visible within the graphs. To remedy this, the standard deviation data for each set of curves for the structured experiments are listed in **Tables 4-6** in **Appendices C-E**.

3.3: Filament Width Results

The four parent images taken of the 4 ink compositions are shown in **Figure 17**. Due to the fact that not all inks produced the same number of viable filaments, the process in ImageJ to measure the filaments was completed for as many solid lines as possible for each ink. Out of the 2:1:0.1 image, 6 filaments were viable to take

measurements from, with a resulting overall width of 2.48 mm. Out of the 2:1:0.5 image, only 4 filaments were viable to take measurements from, but the resulting overall width was 2.41 mm. The 2:1.5:0.1 image had 8 viable filaments, and an overall width of 2.54 mm. Lastly, the 2:1.5:0.5 image had 5 viable filaments and an overall width of 1.26 mm. A summary of the parent images and their filaments, filament measurements, and overall averages can be seen in **Table 3**.
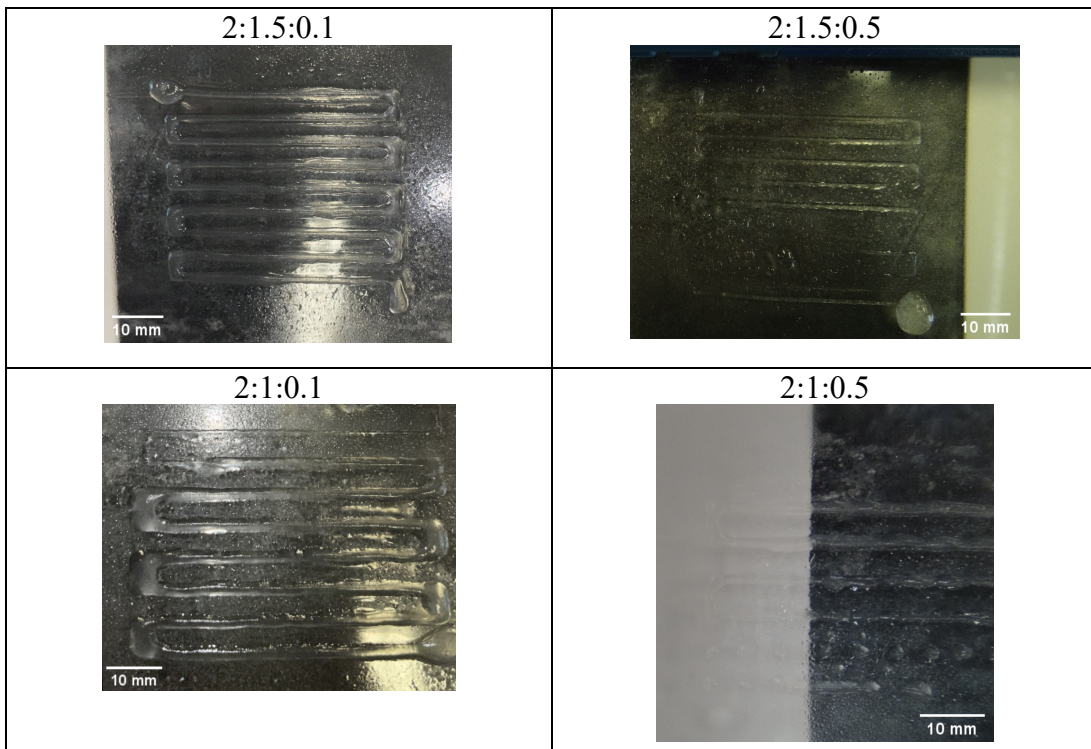
Figure 17. Parent images of the four chosen bio-ink compositions without rulers showing.

Table 3. Average and overall average widths of filaments for each image.

Image	Filament	Average Width (pixels)	Overall Average Width (pixels)	Overall Average Width (mm)
2:1:0.1	1	99	96.83	2.48
	2	95		
	3	85		
	4	103		
	5	101		
	6	98		
2:1:0.5	1	102	95.75	2.41
	2	104		
	3	81		
	4	96		
2:1.5:0.1	1	57	81.5	2.54
	2	64		
	3	65		
	4	92		
	5	95		
	6	96		
	7	90		
	8	93		
2:1.5:0.5	1	26	39.8	1.26
	2	46		
	3	46		
	4	40		
	5	41		

Looking at **Table 3**, the average filament width in pixels is given for each filament. It can be seen that with the exception of a few outliers, the measurements are fairly consistent within each parent image. Overall, the average filament width seems to

increase with increasing solid content, which is what was anticipated. The 2:1.5:0.1 bio-ink does not follow this trend, but this discrepancy could be due to a number of things. The pixel values for the filaments for this specific image could have been catching droplets from the sprayed crosslinker, creating false filament edges. The glare in the image could have also resulted in incorrect pixel values, which would result in incorrect widths. Lastly, the particles within this specific bio-ink could have settled in the container the ink was taken from for printing, or the solution may not have been stirred thoroughly enough, therefore the printed ink may have been at a lower solid content than what was intended, resulting in more spreading of the ink and a wider filament.

Based on the overall averages, the bio-ink with the best print resolution out of the tested inks would be the 2:1.5:0.5 bio-ink, as this had the smallest filament width of 1.26 mm, and therefore would be able to have greater pore definition than the other inks. However, the needle used for printing was only a 410 micron needle, therefore it would be desirable to create an ink with an even smaller filament closer to the needle size in the future, as this would mean even better pore definition.

3.4: Design of Experiments Results

To better understand the effects of the concentrations and their interactions with each other on the outcome of the inks, the DOE method was applied in three cases. In the first case, it was applied to the filament widths for each of the four parent images. Running the least squares model function in R yielded an equation seen in **Equation 1**.

$$\textbf{\textit{Eqn 1.}} \quad width = 2.1725 - 0.2727(TNFC) - 0.3375(Ch) - 0.3025(TNFC * Ch)$$

The generated pareto plot from this least squares model can be seen in **Figure 18**. From this plot and the equation, it can be seen that all effects are negative. This means that increasing either of the concentrations or both leads to a decreased filament width, which is desirable. The pareto plot also shows that based on the filament width outcomes, the changing concentration of the Ch has the most impact on the width, followed by the interaction between the T-NFC and Ch, then the T-NFC. Based on these results, to decrease the filament the most, the chitosan content should be increased first, and for smaller decreases in filament width, the T-NFC should be increased.
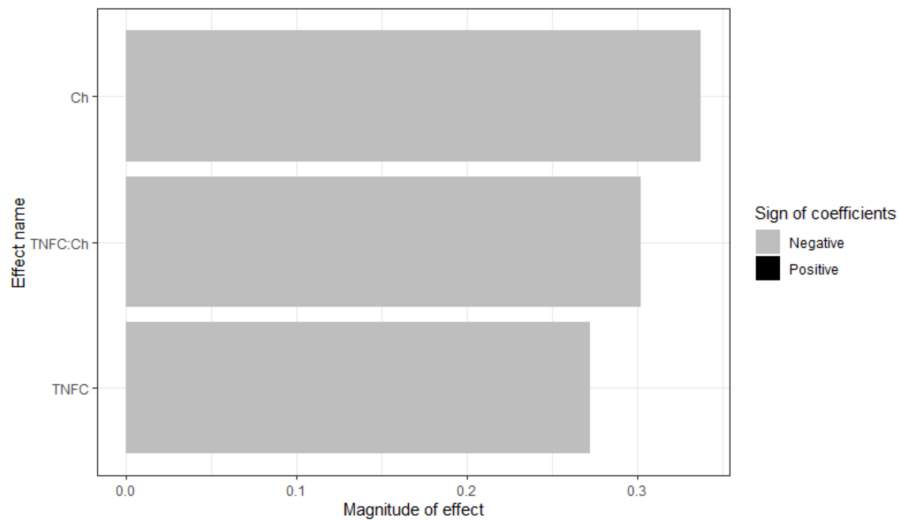
Figure 18. Pareto plot of effects on filament width.

In the second and third cases, the DOE method was applied to the initial and final viscosities of the average flow curves. Running the least squares model function in R for the initial viscosities yielded an equation seen in **Equation 2**. Running this same model function in R for the final viscosities yielded an equation seen in **Equation 3**.

$$\textbf{\textit{Eqn 2.}} \quad initial = 339750 + 86250(TNFC) - 2750(Ch) + 16750(TNFC * Ch)$$

$$\textbf{\textit{Eqn 3.}} \quad final = 137.5 + 11(TNFC) - 1.5(Ch) + 3(TNFC * Ch)$$

The generated pareto plot from using the initial viscosity values can be seen in

Figure 19, and the pareto plot generated from the final viscosity values can be seen in

Figure 20. The R code used to generate the least squares model equations and all pareto

plots can be seen in **Appendix F**. These pareto plots both show that the changing T-NFC

concentration has the largest effect on the viscosity, followed by the interaction of T-NFC

and Ch, then by the Ch. The effect of T-NFC and the interaction effects are both

positively related to the outcomes for both cases. This means that increasing the

concentration of T-NFC will result in an increase in viscosity. If the viscosity of an ink

needs to be increased by a large amount, based on these results, the T-NFC should be

increased first. The effects of Ch are very small for both cases, and negative for the initial

viscosities and positive for the final viscosities. This means Ch can be used to fine tune

ink viscosities in small amounts. This also means that increasing the Ch concentration

will slightly increase viscosity at high shear rates.
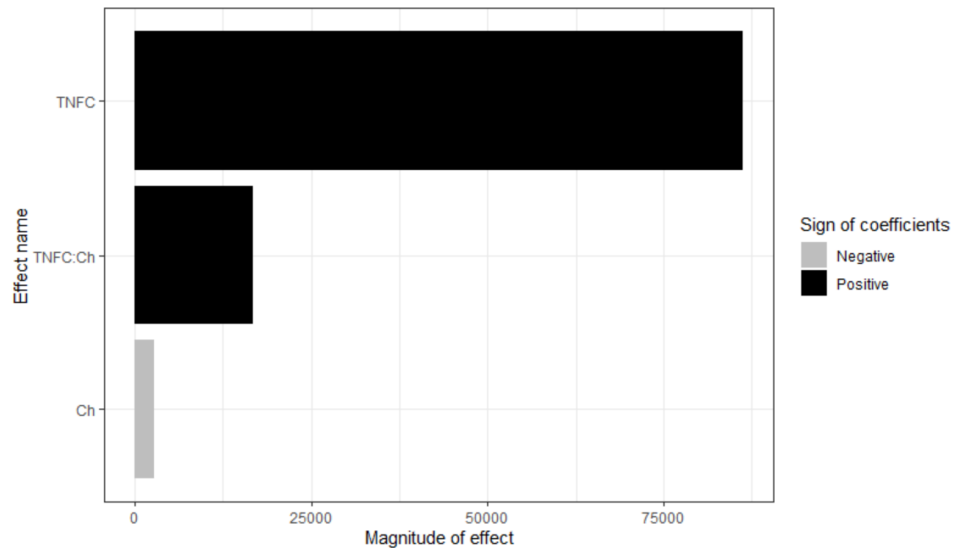
Figure 19. Pareto plot of effect on initial viscosity values.
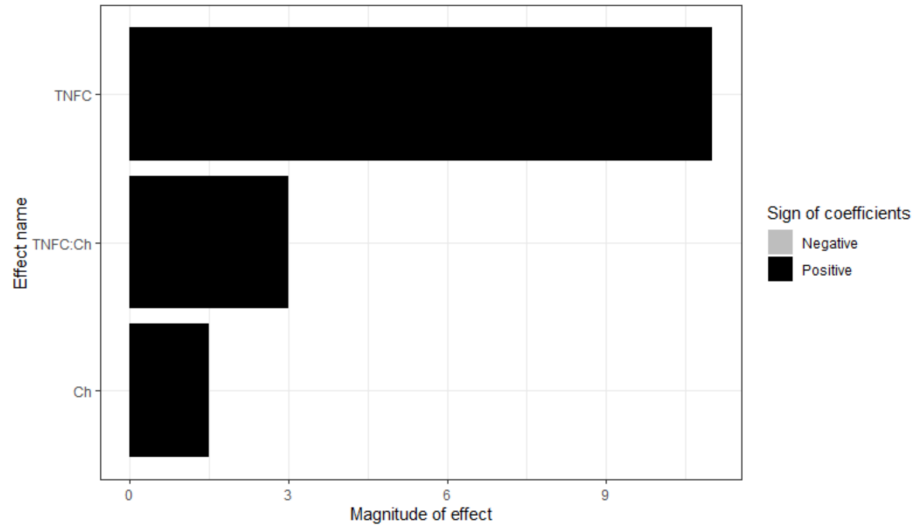
Figure 20. Pareto plot of effect on final viscosity values.

CHAPTER 4: CONCLUSION AND FUTURE WORK

This study investigated the printability of both preliminary experiments and a structured set of experiments by looking at different parameters. The preliminary experiments tested were 3:1:1 w/v%, 3:1:2 w/v%, and 2:1:1 w/v% of Alg:T-NFC:Ch. For each ink tested, the results of hand syringing or printing influenced the next experiment's concentration ratio. All preliminary experiments caused severe clogging in print tests, which is what ultimately led to changing of the concentration ratios for the structured experiments. Due to this clogging, none of these preliminary experiments are suggested for use in further testing unless the needle size is greatly increased, though doing so would result in larger filaments which is undesirable for creating well defined structures in 3D bioprinting.

The structured set of inks tested were 2:1:0.1 w/v%, 2:1:0.5 w/v%, 2:1.5:0.1 w/v%, and 2:1.5:0.5 w/v% of Alg:T-NFC:Ch. For each of these inks, average flow curves, amplitude sweep curves, and thixotropy data sets were generated. All viscosities from the flow curves fell within the range of hydrogel viscosities known to be compatible with bioprinting, therefore supporting the use of these inks in further printing. The amplitude sweep curves showed G' values higher than G" values for all inks at low shear strain indicating solid-like behavior. Within these curves, interception points of G' and G" are present, indicating the fluid shifts to liquid-like behavior at higher values of shear strain. This behavior is conducive to bioprinting, as the inks should be solid-like when left untouched to retain their mechanical stability. Lastly, thixotropy data showed all ink viscosities were able to return to more than 90% of the initial viscosities during the

recovery periods, reinforcing the success printing should have, as the recovered

mechanical strength of the inks will return more than 90% within a 2 minute time frame.

Running the DOE methodology and generating pareto plots for each of 3 cases

gave even more insight into how the outcome parameters are affected by the changing

concentrations of T-NFC and Ch. When using the filament widths at outcome variables,

the plot showed that increasing the concentration of Ch would decrease the filament

width the quickest, while increasing the concentration of T-NFC would only slightly

affect the width, and therefore could be changed for small adjustments. The plots using

initial and final velocities as outcome variables showed that increasing the concentration

of T-NFC would increase the viscosities the quickest, while increasing concentration of

Ch would only slightly change the viscosity values. The equations generated from the

least squares model can be used to predict inks that were not tested in this study, further

reinforcing the value of applying the DOE method here.

Moving forward, based on the idea that the highest solid content ink of 2:1.5:0.5

w/v% resulted in the highest viscosity and smallest filament width, it would make most

sense to test more inks at and just past the limits of this ink concentration. This would

allow for even smaller filament widths to be investigated, which could result in even

better pore definition. It would be beneficial to also apply the DOE method to new sets of

inks to aid in determining how the factors that are changing are affecting the outcome.

Other parameters could also be tested once finding a desirable concentration, such as

printing pressure or printing speed.

For Image J testing, better lighting could be implemented to create better quality images without glare. One way to do this would be to create a box to shield extra light from flooding across the printing base. More complex structures could be printed and analyzed as well. Grid patterns could be printed, and the theoretical pore areas could be compared to the actual analyzed areas seen in the images. One could also look at structure corners to determine how much spreading takes place there based on area filled in. Lastly, some sort of filtering or thresholding of the images could be investigated to determine if thresholding the images allows for better edge detection of the filaments, therefore measuring more accurate widths.

REFERENCES

[1]   H. Rastin, R. T. Ormsby, G. J. Atkins, and D. Losic, "3D Bioprinting of Methylcellulose/Gelatin-Methacryloyl (MC/GelMA) Bioink with High Shape Integrity," *ACS Appl. Bio Mater.*, vol. 3, no. 3, pp. 1815–1826, 2020, doi: 10.1021/acsabm.0c00169.

[2]   A. Habib, V. Sathish, S. Mallik, and B. Khoda, "3D printability of alginate-carboxymethyl cellulose hydrogel," *Materials (Basel).*, vol. 11, no. 3, 2018, doi: 10.3390/ma11030454.

[3]   J. Lee, S. J. Oh, S. An, W.-D. Kim, and S.-H. Kim, "Machine learning-based design strategy for 3D printable bioink: elastic modulus and yield stress determine printability," *Biofabrication*, vol. 12, no. 3, p. https://doi.org/10.1088/1361-6463/aad7de, 2020.

[4]   A. Habib, R. S. R. Kalidhindi, V. Sathish, and B. Khoda, "Comparative study on long and short cellulose fiber filled bio-ink," 2019.

[5]   P. Erkoc *et al.*, "3D Printing of Cytocompatible Gelatin-Cellulose-Alginate Blend Hydrogels," *Macromol. Biosci.*, vol. 20, no. 10, pp. 1–15, 2020, doi: 10.1002/mabi.202000106.

[6]   T. T. Demirtaş, G. Irmak, and M. Gümüşderelioğlu, "A bioprintable form of chitosan hydrogel for bone tissue engineering," *Biofabrication*, vol. 9, no. 3, 2017, doi: 10.1088/1758-5090/aa7b1d.

[7]   A. Habib and B. Khoda, "Effect of Process Parameters on Cellulose Fiber Alignment in Bio-Printing," 2019.

[8]   J. Berger, M. Reist, J. M. Mayer, O. Felt, and R. Gurny, "Structure and interactions in chitosan hydrogels formed by complexation or aggregation for biomedical applications," *Eur. J. Pharm. Biopharm.*, vol. 57, no. 1, pp. 35–52, 2004, doi: 10.1016/S0939-6411(03)00160-7.

[9]   T. G. Mezger, *Applied Rheology*. Anton Paar GmbH, 2019.

[10] A. Shafiee *et al.*, "Physics of bioprinting," *Appl. Phys. Rev.*, vol. 6, no. 2, 2019, doi: 10.1063/1.5087206.

[11] J. Jia *et al.*, "Engineering alginate as bioink for bioprinting," *Acta Biomater.*, vol. 10, no. 10, pp. 4323–4331, 2014, doi: 10.1016/j.actbio.2014.06.034.

[12] Kevin Dunn, *Process Improvement using Data*. 2022.

APPENDICES

APPENDIX A: FIJI CODE

Note: This code is intended to run in sections based on analyzing one image at a time, though it can be run as a whole, variables will just be replaced with new values.

```
//Create path to pull the images from.
image_path  = "C:/Users/jordy/OneDrive/Documents/HON_Files/Images/";
//Create path to general project folder.
project_path = "C:/Users/jordy/OneDrive/Documents/HON_Files/";


//Code for analyzing the 2:1.5:0.1 image.
//Open the desired image.
open(image_path+"2-1_5-0_1.pgm");
//Create "count" variable that starts at the value of 1.
count = 1;
//Draw line over ruler.
makeLine(3620, 616, 3616, 936);
//Plot the profile of the line.
run("Plot Profile");
//Find the minimums of the profile by setting the threshold for the maximums higher than
any of the values.
run("Find Peaks", "min._peak_amplitude=56.31 min._peak_distance=0 min._value=300
max._value=[] exclude list");
```

//Save the minimums found to a CSV file.

saveAs("Results", project_path+"2-15-01/Peak_Data.csv");

//Close the plotted profile.

close("Plot of 2-1_5-0_1");

//Close the plot generated from "Find Peaks".

close("Peaks in Plot of 2-1_5-0_1");

//Close the peak data.

close("Peak_Data.csv");

//Create for loop to cycle through drawing each rectangle.

for (k = 931; k < 1943; k+=142) {

      //Select the image window.

      selectWindow("2-1_5-0_1.pgm");

      //Draw the rectangle.

      makeRectangle(516,k,640,140);

      //Copy the rectangle.

      run("Copy");

      //Create image from drawn rectangle.

      run("Internal Clipboard");

      //Convert to a black and white image.

      run("16-bit");

```
//Create for loop to cycle through drawing each line.

for (j = 1; j < 32; j++) {

        //Create the variable for calculating the line location.

        line_location = j*20;

        //Select the filament image.

        selectWindow("Clipboard");

        //Draw the line.

        makeLine(line_location, 0, line_location, 140);

        //Plot the line profile.

        run("Plot Profile");

        //Get the values from the plot.

        Plot.getValues(x, y);

        //Create for loop to print each data point of plot.

        for (i=0; i<x.length; i++) {

                //Print data point.

                print(x[i]+","+y[i]);

        }

        //Close the plot.

        close("Plot of Clipboard");

        //Select the log window.

        selectWindow("Log");

        //Save the data from the log window as a CSV file for each line.
```

```
            saveAs("Text", project_path+"2-15-01/fil"+count+"/line"+j+".csv");

            //Close the log window.

            close("Log");

        }

        //Add 1 to the count variable so the filament number changes in the next loop.

        count++;

        //Close the filament image.

        close("Clipboard");

    }

    //Close the parent image.

    close("2-1_5-0_1.pgm");




    //Code for analyzing the 2:1.5:0.5 image.

    //Open the desired image.

    open(image_path+"2-1_5-0_5.pgm");


    run("Rotate... ", "angle=-2 grid=1 interpolation=Bilinear");

    //Create "count" variable that starts at the value of 1.

    count = 1;

    //Draw line over ruler.

    makeLine(2416, 152, 2732, 144);

    //Plot the profile of the line.
```

run("Plot Profile");

//Find the minimums of the profile by setting the threshold for the maximums higher than

any of the values.

run("Find Peaks", "min._peak_amplitude=54.64 min._peak_distance=0 min._value=300

max._value=[] exclude list");

//Save the minimums found to a CSV file.

saveAs("Results", project_path+"2-15-05/Peak_Data.csv");

//Close the plotted profile.

close("Plot of 2-1_5-0_5");

//Close the plot generated from "Find Peaks".

close("Peaks in Plot of 2-1_5-0_5");

//Close the peak data.

close("Peak_Data.csv");

//Create for loop to cycle through drawing each rectangle.

for (m = 900; m < 1478; m+=140) {

    //Select the image window.

    selectWindow("2-1_5-0_5.pgm");

    //Draw the rectangle.

    makeRectangle(2440,m,640,140);

    //Copy the rectangle.

    run("Copy");

    //Create image from drawn rectangle.

```
run("Internal Clipboard");

//Convert to a black and white image.

run("16-bit");




//Create for loop to cycle through drawing each line.

for (n = 1; n < 32; n++) {

        //Create the variable for calculating the line location.

        line_location = n*20;

        //Select the filament image.

        selectWindow("Clipboard");

        //Draw the line.

        makeLine(line_location, 0, line_location, 140);

        //Plot the line profile.

        run("Plot Profile");

        //Get the values from the plot.

        Plot.getValues(x, y);

        //Create for loop to print each data point of plot.

        for (o=0; o<x.length; o++) {

                //Print the data point.

                print(x[o]+","+y[o]);

        }

        //Close the plot.
```

```
            close("Plot of Clipboard");

            //Select the log window.

            selectWindow("Log");

            //Save the data from the log window as a CSV file for each line.

            saveAs("Text", project_path+"2-15-05/fil"+count+"/line"+n+".csv");

            //Close the log window.

            close("Log");

        }

        //Add 1 to the count variable so the filament number changes in the next loop.

        count++;

        //Close the filament image.

        close("Clipboard");

    }

//Close the parent image.

close("2-1_5-0_5.pgm");



//Code for analyzing the 2:1:0.1 image.

//Open the desired image.

open(image_path+"2-1-0_1.pgm");


run("Rotate 90 Degrees Right");

run("Rotate... ", "angle=-1 grid=1 interpolation=Bilinear");
```

```
//Create "count" variable that starts at the value of 1.

count = 1;

//Draw line over ruler.

makeLine(396, 630, 786, 630);

//Plot the profile of the line.

run("Plot Profile");

//Find the minimums of the profile by setting the threshold for the maximums higher than

any of the values.

run("Find Peaks", "min._peak_amplitude=70.28 min._peak_distance=0 min._value=300

max._value=[] exclude list");

//Save the minimums found to a CSV file.

saveAs("Results", project_path+"2-1-01/Peak_Data.csv");

//Close the plotted profile.

close("Plot of 2-1-0_1");

//Close the plot generated from "Find Peaks".

close("Peaks in Plot of 2-1-0_1");

//Close the peak data.

close("Peak_Data.csv");


//Create for loop to cycle through drawing each rectangle.

for (a = 2610; a < 3661; a+=210) {

        //Select the image window.

        selectWindow("2-1-0_1.pgm");
```

```
//Draw the rectangle.

makeRectangle(960,a,640,140);

//Copy the rectangle.

run("Copy");

//Create image from drawn rectangle.

run("Internal Clipboard");

//Convert to a black and white image.

run("16-bit");


//Create for loop to cycle through drawing each line.

for (l = 1; l < 32; l++) {

        //Create the variable for calculating the line location.

        line_location = l*20;

        //Select the filament image.

        selectWindow("Clipboard");

        //Draw the line.

        makeLine(line_location, 0, line_location, 140);

        //Plot the line profile.

        run("Plot Profile");

        //Get the values from the plot.

        Plot.getValues(x, y);


        //Create for loop to print each data point of plot.
```

```
        for (p=0; p<x.length; p++) {

                //Print the data point.

                print(x[p]+","+y[p]);

        }

        //Close the plot.

        close("Plot of Clipboard");

        //Select the log window.

        selectWindow("Log");

        //Save the data from the log window as a CSV file for each line.

        saveAs("Text", project_path+"2-1-01/fil"+count+"/line"+l+".csv");

        //Close the log window.

        close("Log");

    }

    //Add 1 to the count variable so the filament number changes in the next loop.

    count++;

    //Close the filament image.

    close("Clipboard");

}

//Close the parent image.

close("2-1-0_1.pgm");



//Code for analyzing the 2:1:0.5 image.
```

```
//Open the desired image.

open(image_path+"2-1-0_5.pgm");

//Create "count" variable that starts at the value of 1.

count = 1;

//Draw line over ruler.

makeLine(1636, 792, 1640, 1876);

//Plot the profile of the line.

run("Plot Profile");

//Find the minimums of the profile by setting the threshold for the maximums higher than

any of the values.

run("Find Peaks", "min._peak_amplitude=50 min._peak_distance=0 min._value=300

max._value=[] exclude list");

//Save the minimums found to a CSV file.

saveAs("Results", project_path+"2-1-05/Peak_Data.csv");

//Close the plotted profile.

close("Plot of 2-1-0_5");

//Close the plot generated from "Find Peaks".

close("Peaks in Plot of 2-1-0_5");


//Close the peak data.

close("Peak_Data.csv");


//Create for loop to cycle through drawing each rectangle.
```

```
for (q = 900; q < 1561; q+=220) {

        //Select the image window.

        selectWindow("2-1-0_5.pgm");

        //Draw the rectangle.

        makeRectangle(3200,q,640,140);

        //Copy the rectangle.

        run("Copy");

        //Create image from drawn rectangle.

        run("Internal Clipboard");

        //Convert to a black and white image.

        run("16-bit");


        //Create for loop to cycle through drawing each line.

        for (r = 1; r < 32; r++) {

                //Create the variable for calculating the line location.

                line_location = r*20;

                //Select the filament image.

                selectWindow("Clipboard");

                //Draw the line.

                makeLine(line_location, 0, line_location, 140);

                //Plot the line profile.

                run("Plot Profile");

                //Get the values from the plot.
```

```
            Plot.getValues(x, y);

            //Create for loop to print each data point of plot.

            for (s=0; s<x.length; s++) {

                    //Print the data point.

                    print(x[s]+","+y[s]);

            }

            //Close the plot.

            close("Plot of Clipboard");

            //Select the log window.

            selectWindow("Log");

            //Save the data from the log window as a CSV file for each line.

            saveAs("Text", project_path+"2-1-05/fil"+count+"/line"+r+".csv");

            //Close the log window.

            close("Log");

        }


        //Add 1 to the count variable so the filament number changes in the next loop.

        count++;

        //Close the filament image.

        close("Clipboard");

    }

    //Close the parent image.

    close("2-1-0_5.pgm");
```

APPENDIX B: FILAMENT WIDTH R CODE

Note: This code is intended to run in in sections based on analyzing one image at a time, though it can be run as a whole, variables will just be replaced with new values.

```
#Section of code for analyzing 2:1.5:0.1 ink.
#Create path the data.
path = 'C:\\Users\\jordy\\OneDrive\\Documents\\HON_Files\\2-15-01\\'
#Create data path for peak data.
data_path = paste0(path, 'Peak_Data.csv')
#Read the peak data.
read_data = read.csv(data_path)
#Extract just the valley locations.
xpoints = read_data$X2
#Take out any NA values.
real_xpoints = na.omit(xpoints)
#Sort the points in increasing order.
sorted_xpoints = sort(real_xpoints, decreasing=FALSE)
#Find the differences between the points.
differences = diff(sorted_xpoints)
#Find the average difference.
avg_difference = round(mean(differences), 2)
#Print the pixels/mm value found.
```

```r
print(paste0(avg_difference, " pixels for 2:1.5:0.1 image = 1 mm"))


#Create for loop to go through each filament.
for (j in 1:8) {


        #Create for loop to go through each line.
        for (i in 1:31) {
                #Create variable to store line names.
                nam = paste0('line',i)
                #Read the line CSV files and assign line name to the vector.
                assign(nam, read.csv(paste0(path, 'fil', j, '\\line', i, '.csv'), header=FALSE,
                colClasses = c("NULL",NA)))
        }


        #Create array of line data.
        pixel_fil =
        cbind(line1,line2,line3,line4,line5,line6,line7,line8,line9,line10,line11,line12,line
13,line1
        4,line15,line16,line17,line18,line19,line20,line21,line22,line23,line24,line25,line2
6,line2
        7,line28,line29,line30,line31)
        #Find averages of pixel values.
        avg_fil = rowMeans(pixel_fil)
```

```
#Find standard deviations of pixel values.

st = apply(pixel_fil, 1, sd)

#Save standard deviations to CSV files.

write.csv(st, paste0(path, 'Stdevs\\fil', j,'.csv'))


#Create range for first low point.

first_range = avg_fil[20:50]

#Find first minimum point.

first_point = min(first_range)

#Find the location of the first minimum point.

first_point_local = which.min(first_range)

#Find the actual minimum location.

first_local = 20+first_point_local


#Create range for second low point.

second_range = avg_fil[90:120]

#Find second minimum point.

second_point = min(second_range)

#Find the location of the second minimum point.

second_point_local = which.min(second_range)

#Find the actual minimum location.

second_local = 90+second_point_local
```

```
        #Create variable to store distance names.

        disnam = paste0('distance',j)

        #Calculate the distance and assign the distance name to the value.

        assign(disnam, second_local-first_local)


        #Print the average distance (width) for each filament.

        print(paste0("avg for fil", j, " ", second_local-first_local))

}


#Calculate overall average width.

overall_avg =

(distance1+distance2+distance3+distance4+distance5+distance6+distance7+distance8)/8

#Print the overall average width.

print(paste0('Overall average width for 2:1.5:0.1 ink is: ', overall_avg, ' pixels'))

#Convert the width in pixels to width in mm.

width = round(overall_avg/avg_difference, 2)

#Print the width in mm.

print(paste0('Width of 2:1.5:0.1 ink is: ', width, ' mm'))



#Section of code for analyzing 2:1.5:0.5 ink.

#Create path to the data.
```

```r
path = 'C:\\Users\\jordy\\OneDrive\\Documents\\HON_Files\\2-15-05\\'

#Create data path for peak data.

data_path = paste0(path, 'Peak_Data.csv')

#Read the peak data.

read_data = read.csv(data_path)

#Extract just the valley locations.

xpoints = read_data$X2

#Take out any NA values.

real_xpoints = na.omit(xpoints)

#Sort the points in increasing order.

sorted_xpoints = sort(real_xpoints, decreasing=FALSE)

#Find the differences between the points.

differences = diff(sorted_xpoints)

#Find the average difference.

avg_difference = round(mean(differences), 2)

#Print the pixels/mm value found.

print(paste0(avg_difference, " pixels for 2:1.5:0.5 image = 1 mm"))


#Create for loop to go through each filament.

for (j in 1:5) {


        #Create for loop to go through each line.

        for (i in 1:31) {
```

```r
          #Create variable to store line names.

          nam = paste0('line',i)

          #Read the line CSV files and assign line name to the vector.

          assign(nam, read.csv(paste0(path, 'fil', j, '\\line', i, '.csv'), header=FALSE,

          colClasses = c("NULL",NA)))

     }


     #Create array of line data.

     pixel_fil =

     cbind(line1,line2,line3,line4,line5,line6,line7,line8,line9,line10,line11,line12,line

13,line1

     4,line15,line16,line17,line18,line19,line20,line21,line22,line23,line24,line25,line2

6,line

     27,line28,line29,line30,line31)

     #Find averages of pixel values.

     avg_fil = rowMeans(pixel_fil)


     #Find standard deviations of pixel values.

     st = apply(pixel_fil, 1, sd)

     #Save standard deviations to CSV files.

     write.csv(st, paste0(path, 'Stdevs\\fil', j, '.csv'))
```

```r
#Create range for first low point.

first_range = avg_fil[30:60]

#Find first minimum point.

first_point = min(first_range)

#Find the location of the first minimum point.

first_point_local = which.min(first_range)

#Find the actual minimum location.

first_local = 30+first_point_local


#Create range for second low point.

second_range = avg_fil[61:100]

#Find second minimum point.

second_point = min(second_range)

#Find the location of the second minimum point.

second_point_local = which.min(second_range)

#Find the actual minimum location.

second_local = 61+second_point_local


#Create variable to store distance names.

disnam = paste0('distance', j)

#Calculate the distance and assign the distance name to the value.

assign(disnam, second_local-first_local)
```

#Print the average distance (width) for each filament.

print(paste0("avg for fil", j, " ", second_local-first_local))

}

#Calculate overall average width.

overall_avg = (distance1+distance2+distance3+distance4+distance5)/5

#Print the overall average width.

print(paste0('Overall average width for 2:1.5:0.5 ink is: ', overall_avg, ' pixels'))

#Convert the width in pixels to width in mm.

width = round(overall_avg/avg_difference, 2)

#Print the width in mm.

print(paste0('Width of 2:1.5:0.5 ink is: ', width, ' mm'))

#Section of code for analyzing 2:1:0.1 ink.

#Create path to the data.

path = 'C:\\Users\\jordy\\OneDrive\\Documents\\HON_Files\\2-1-01\\'

#Create path for peak data.

data_path = paste0(path, 'Peak_Data.csv')

#Read the peak data.

read_data = read.csv(data_path)

#Extract just the valley locations.

```r
xpoints = read_data$X2

#Take out any NA values.

real_xpoints = na.omit(xpoints)

#Sort the points in increasing order.

sorted_xpoints = sort(real_xpoints, decreasing=FALSE)

#Find the differences between the points.

differences = diff(sorted_xpoints)

#Find the average difference.

avg_difference = round(mean(differences), 2)

#Print the pixels/mm value found.

print(paste0(avg_difference, " pixels for 2:1:0.1 image = 1 mm"))


#Create for loop to go through each filament.
for (j in 1:6) {


        #Create for loop to go through each line.
        for (i in 1:31) {

                #Create variable to store line names.
                nam = paste0('line',i)

                #Read the line CSV files and assign line name to the vector.
                assign(nam, read.csv(paste0(path, 'fil', j, '\\line', i, '.csv'), header=FALSE,

                colClasses = c("NULL",NA)))

        }
```

```
#Create array of line data.

pixel_fil =

cbind(line1,line2,line3,line4,line5,line6,line7,line8,line9,line10,line11,line12,line

13,line1

4,line15,line16,line17,line18,line19,line20,line21,line22,line23,line24,line25,line2

6,line2

7,line28,line29,line30,line31)

#Find averages of pixel values.

avg_fil = rowMeans(pixel_fil)


#Find standard deviations of pixel values.

st = apply(pixel_fil, 1, sd)

#Save standard deviations to CSV files.

write.csv(st, paste0(path, 'Stdevs\\fil', j, '.csv'))


#Create range for first low point.

first_range = avg_fil[20:40]

#Find first minimum point.

first_point = min(first_range)

#Find the location of the first minimum point.

first_point_local = which.min(first_range)

#Find the actual minimum location.
```

```r
        first_local = 20+first_point_local


        #Create range for second low point.

        second_range = avg_fil[100:130]

        #Find second minimum point.

        second_point = min(second_range)

        #Find the location of the second minimum point.

        second_point_local = which.min(second_range)

        #Find the actual minimum location.

        second_local = 100+second_point_local


        #Create variable to store distance names.

        disnam = paste0('distance', j)

        #Calculate the distance and assign the distance name to the value.

        assign(disnam, second_local-first_local)


        #Print the average distance (width) for each filament.

        print(paste0("avg for fil", j, " ", second_local-first_local))
}


#Calculate overall average width.

overall_avg = (distance1+distance2+distance3+distance4+distance5+distance6)/6

#Print the overall average width.
```

```r
print(paste0('Overall average width for 2:1:0.1 ink is: ', overall_avg, ' pixels'))

#Convert the width in pixels to width in mm.

width = round(overall_avg/avg_difference, 2)

#Print the width in mm.

print(paste0('Width of 2:1:0.1 ink is: ', width, ' mm'))




#Section of code for analyzing 2:1:0.5 ink.

#Create path to the data.

path = 'C:\\Users\\jordy\\OneDrive\\Documents\\HON_Files\\2-1-05\\'

#Create data path for peak data.

data_path = paste0(path, 'Peak_Data.csv')

#Read the peak data.

read_data = read.csv(data_path)

#Extract just the valley locations.

xpoints = read_data$X2

#Take out any NA values.

real_xpoints = na.omit(xpoints)

#Sort the points in increasing order.

sorted_xpoints = sort(real_xpoints, decreasing=FALSE)

#Find the differences between the points.

differences = diff(sorted_xpoints)

#Find the average difference and divide by 5 to get pixels per mm.
```

```r
avg_difference = round(mean(differences/5), 2)

#Print the pixels/mm value found.

print(paste0(avg_difference, " pixels for 2:1:0.5 image = 1 mm"))


#Create for loop to go through each filament.

for (j in 1:4) {


        #Create for loop to go through each line.

        for (i in 1:31) {

                #Create variable to store line names.

                nam = paste0('line',i)

                #Read the line CSV files and assign line name to the vector.

                assign(nam, read.csv(paste0(path, 'fil', j, '\\line', i, '.csv'), header=FALSE,

                colClasses = c("NULL",NA)))

        }


        #Create array of line data.

        pixel_fil =

        cbind(line1,line2,line3,line4,line5,line6,line7,line8,line9,line10,line11,line12,line

13,line1

        4,line15,line16,line17,line18,line19,line20,line21,line22,line23,line24,line25,line2

6,line2

        7,line28,line29,line30,line31)
```

#Find averages of pixel values.

avg_fil = rowMeans(pixel_fil)

#Find standard deviations of pixel values.

st = apply(pixel_fil, 1, sd)

#Save standard deviations to CSV files.

write.csv(st, paste0(path, 'Stdevs\\fil', j, '.csv'))

#Create range for first low point.

first_range = avg_fil[15:40]

#Find first minimum point.

first_point = min(first_range)

#Find the location of the first minimum point.

first_point_local = which.min(first_range)

#Find the actual minimum location.

first_local = 15+first_point_local

#Create range for second low point.

second_range = avg_fil[90:120]

#Find second minimum point.

second_point = min(second_range)

```r
        #Find the location of the second minimum point.

        second_point_local = which.min(second_range)

        #Find the actual minimum location.

        second_local = 90+second_point_local


        #Create variable to store distance names.

        disnam = paste0('distance', j)

        #Calculate the distance and assign the distance name to the value.

        assign(disnam, second_local-first_local)


        #Print the average distance (width) for each filament.

        print(paste0("avg for fil", j, " ", second_local-first_local))
}


#Calculate overall average width.

overall_avg = (distance1+distance2+distance3+distance4)/4

#Print the overall average width.

print(paste0('Overall average width for 2:1:0.5 ink is: ', overall_avg, ' pixels'))

#Convert the width in pixels to width in mm.

width = round(overall_avg/avg_difference, 2)

#Print the width in mm.

print(paste0('Width of 2:1:0.5 ink is: ', width, ' mm'))
```

APPENDIX C: FLOW CURVE STANDARD DEVIATION DATA

Table 4. Standard deviation data for flow curves.

Data Point	2:1.5:0.1 [mPa·s]	2:1.5:0.5 [mPa·s]	2:1:0.1 [mPa·s]	2:1:1.5 [mPa·s]
1	1.83E+04	52180.94	3.03E+05	2.58E+05
2	5.46E+03	22090.3	1.43E+05	1.29E+05
3	2.80E+03	10411.5	71832	69184
4	1.83E+03	4168.616	41456	41888
5	1.17E+03	1771.23	27197	28821
6	9.75E+02	477.0608	19915	22023
7	1.28E+03	289.3769	14558	15641
8	1.20E+03	500.1223	8286.1	8786.9
9	2.17E+02	183.5356	5024.9	5208.4
10	2.88E+01	51.9749	3150.3	3252.7
11	2.31E+01	36.31217	2035.3	2106.7
12	3.57E+01	9.438397	1358.6	1404.5
13	2.81E+01	18.20082	942.11	972.48
14	2.00E+01	20.49057	680.37	702.45
15	1.39E+01	18.07794	509.85	526.7
16	9.88E+00	14.33759	394.93	407.35
17	6.63E+00	10.32041	313.01	322.37
18	4.58E+00	7.105789	250.96	257.83
19	2.94E+00	4.829341	201.59	206.6
20	1.90E+00	3.344792	160.48	164.08
21	1.27E+00	2.294915	125.82	128.21

Table 5. Standard deviation data for amplitude sweeps.

Data Point	2:1.5:0.1		2:1.5:0.5		2:1:0.1		2:1:1.5	
	G' [Pa]	G" [Pa]	G' [Pa]	G" [Pa]	G' [Pa]	G" [Pa]	G' [Pa]	G" [Pa]
1	7.111	2.860	1.955	0.826	2.029	1.354	1.039	1.560
2	8.187	2.633	3.017	0.927	0.434	0.929	1.755	1.122
3	8.024	2.356	1.623	0.842	0.696	0.683	1.791	0.728
4	7.709	2.156	1.775	1.147	0.912	0.055	1.577	0.466
5	7.670	2.091	1.408	0.231	0.951	0.097	1.187	0.297
6	7.047	1.968	1.744	0.514	1.026	0.145	1.889	0.462
7	6.809	1.780	0.285	0.327	0.342	0.196	1.635	0.633
8	6.230	1.752	0.341	0.241	0.687	0.214	1.907	0.567
9	6.177	1.797	0.484	0.113	0.123	0.064	2.132	0.562
10	5.976	1.727	0.449	0.197	0.599	0.078	1.844	0.534
11	5.757	1.689	0.591	0.213	0.480	0.102	1.817	0.505
12	5.368	1.657	0.712	0.215	0.624	0.068	1.757	0.508
13	4.925	1.690	0.410	0.264	0.854	0.193	2.024	0.450
14	3.975	1.923	1.179	0.373	0.904	0.201	1.786	0.521
15	3.149	1.963	1.493	0.483	0.791	0.201	1.441	0.493
16	2.141	2.014	1.505	0.669	0.749	0.211	1.343	0.470
17	1.721	1.802	1.462	0.774	0.583	0.226	1.226	0.498
18	1.400	1.548	1.170	0.714	0.409	0.308	0.987	0.558
19	0.912	1.270	1.139	0.626	0.388	0.307	0.772	0.536
20	0.522	1.129	1.315	0.458	0.320	0.303	0.482	0.437
21	0.388	0.933	0.733	0.178	0.161	0.255	0.114	0.394
22	0.368	0.748	0.449	0.295	0.161	0.240	0.070	0.442
23	0.199	0.520	0.334	0.360	0.088	0.170	0.123	0.372
24	0.111	0.323	0.244	0.328	0.044	0.122	0.137	0.329
25	0.083	0.276	0.131	0.180	0.029	0.096	0.069	0.213

APPENDIX E: THIXOTROPY TEST STANDARD DEVIATION DATA

Table 6. Standard deviation data for thixotropy tests.

Data Point	2:1.5:0.1 [mPa·s]	2:1.5:0.5 [mPa·s]	2:1:0.1 [mPa·s]	2:1:1.5 [mPa·s]
1	392.5086	1142.243	2372.574	992.9627
2	163.5553	1105.931	2277.452	914.2707
3	294.7864	921.6794	2221.852	881.6997
4	413.144	985.743	2160.526	855.5491
5	284.7759	248.1915	2143.535	839.4727
6	308.4088	1031.295	2199.174	824.4677
7	254.6848	992.7875	2019.727	873.8518
8	324.5012	983.4333	1972.729	894.4497
9	358.0116	983.8543	1984.945	932.1708
10	317.6702	896.6454	2010.588	886.3077
11	2.576179	2.192084	10.743	7.086976
12	1.751038	1.319747	9.983037	5.633483
13	2.165279	1.35611	9.905848	5.129779
14	1.624531	1.354339	9.408509	5.477101
15	2.225826	0.990808	9.507702	5.771623
16	1.300013	1.220341	9.360069	5.825224
17	0.435775	1.420293	9.282157	5.828834
18	2.710037	1.504527	9.438042	5.940816
19	2.666577	1.514034	9.247282	5.900257
20	2.925583	1.572429	9.217876	5.875554
21	255.2985	237.5633	618.2508	422.9253
22	499.7803	346.9308	879.6115	524.645
23	560.9929	459.4446	933.5922	527.73
24	605.0598	512.1644	952.1777	501.3525
25	565.691	520.297	806.5608	530.747
26	536.8839	557.4337	1002.318	538.6616
27	599.3255	565.033	1036.167	469.5043
28	534.661	577.313	1044.146	455.1058
29	536.1091	644.4913	1049.075	433.9151
30	497.8718	608.1129	1028.42	467.5322
31	508.3703	686.0425	1052.74	415.0614
32	706.4269	682.9468	1108.479	436.5047
33	590.6965	668.7578	1040.144	473.5846
34	674.1041	711.2709	1063.805	407.0692
35	600.1269	727.0532	1028.365	429.0097

36	521.3658	714.5803	1144.203	370.8697
37	564.1923	716.7956	1015.976	454.0297
38	635.6141	734.0227	1014.531	435.541
39	584.2799	714.6694	1163.547	427.6494
40	654.7368	757.1123	1076.165	484.9443
41	676.6161	728.5117	846.249	422.8806
42	653.245	756.9881	841.1256	452.1991
43	675.1721	772.5868	1065.463	416.3464
44	730.9562	692.6921	1127.644	430.4498
45	686.3121	690.1963	1087.348	431.2648
46	706.652	638.9752	1110.111	412.5385
47	697.8918	690.8881	1167.837	460.3176
48	764.555	702.1204	1158.409	423.9344
49	738.6603	694.7582	1145.596	487.7079
50	764.5556	724.1604	1164.882	462.0433
51	753.727	693.1972	1126.07	440.7565
52	808.837	690.8169	1079.683	447.2259
53	860.5736	674.6646	1169.106	398.3093
54	814.0076	671.8812	1106.052	448.3674
55	861.4043	734.7117	1056.8	455.2761
56	800.2389	755.6734	1161.878	468.2994
57	841.3741	827.8963	1085.327	496.8796
58	917.5916	886.3635	1133.151	478.3318
59	853.9245	846.4351	1132.685	507.4764
60	925.0032	843.7103	1111.649	488.9175
61	880.1536	783.4311	1123.514	529.5668
62	906.5534	828.6634	1071.2	536.6622
63	940.0236	798.9558	1128.422	502.9258
64	859.1742	764.2862	1127.094	563.5977
65	885.4425	816.4762	1115.243	520.4905
66	826.7263	776.5095	1136.652	578.0804
67	901.3237	842.5986	1140.508	578.4384
68	949.2188	776.361	1096.214	592.5766
69	930.5337	746.5161	1081.7	617.6026
70	998.6292	771.5156	1095.907	599.2673
71	912.4063	810.3446	1089.327	620.2107
72	991.6718	900.6566	1081.335	611.7658
73	764.3104	876.3044	1125.719	612.144
74	944.4164	904.4938	1108.616	642.5006
75	1018.724	882.6534	1126.914	597.3821

76	931.3152	823.6348	1088.007	651.2329
77	990.4677	833.4803	1102.125	650.1869
78	993.6158	833.8119	1127.907	692.6955
79	994.2822	839.6823	1109.786	670.6216
80	1040.898	873.4583	1017.637	665.1827

APPENDIX F: LEAST SQUARES MODEL R CODE

```
# Coded version of T-NFC, -1 = 1%, +1 = 1.5%

TNFC = c(-1, +1, -1, +1)

# Coded version of Ch, -1 = 0.1%, +1 = 0.5%

Ch = c(-1, -1, +1, +1)

# Filament widths corresponding to percentages.

width = c(2.48, 2.54, 2.41, 1.26)

# Used least squares model to find equation that can be used to predict other filament

widths.

model = lm(width ~ TNFC + Ch + TNFC*Ch)

# Show output of least squares model.

model

# Import pid library so pareto plot can be generated, plot reveals which factor affects the

outcome the most.

library(pid)

paretoPlot(model)

# Initial viscosities corresponding to percentages.

initial_viscosity = c(273000, 412000, 234000, 440000)

# Used least squares model to find equation that can be used to predict other viscosities.

model = lm(initial_viscosity ~ TNFC + Ch + TNFC*Ch)

# Show output of least squares model.

model
```

Import pid library so pareto plot can be generated, plot reveals which factor affects the

outcome the most.

library(pid)

paretoPlot(model)

Final viscosities corresponding to percentages.

final_viscosity = c(128, 144, 125, 153)

Used least squares model to find equation that can be used to predict other viscosities.

model = lm(final_viscosity ~ TNFC + Ch + TNFC*Ch)

Show output of least squares model.

model

Import pid library so pareto plot can be generated, plot reveals which factor affects the

outcome the most.

library(pid)

paretoPlot(model)

AUTHOR'S BIOGRAPHY

Jordyn Judkins grew up in Deer Isle, Maine, which is a small island on the coast known for its scenic views of the ocean and lobster fishing. Jordyn attended Deer Isle Stonington High School, and graduated from there in 2017 as valedictorian of her class. The University of Maine was her dream school, and though she began her college career as a mechanical engineering major and student of the Honors College, after her first year she decided she couldn't ignore her interests in human anatomy and biology, and therefore began her sophomore year as a biomedical engineering major with a mechanical engineering minor. During her five years at UMaine, Jordyn has been a part of the Biomedical Engineering Club, Society of Women Engineers, and joined Tau Beta Pi, a national engineering honors society. She also completed a Co-op at Northern Light Eastern Maine Medical Center, where she fell in love with working on medical equipment as a clinical engineering intern.

After her graduation in May 2022, Jordyn plans to enter straight into the workforce, with the hope of eventually becoming a clinical engineer, though she still has interests in other areas of biomedical engineering. Outside of academia, Jordyn is a Mainer at heart who loves to hike, hunt, and fish, but also loves to sit down with a good book.