The Optimization of Thin-Film Ribbon Springs

Rachel Willick

Advisors: Stephen Nonnenmann and David Hanneke
May 8, 2025

Submitted to the
Department of Physics & Astronomy of Amherst College
in partial fulfilment of the
requirements for the degree of
Bachelors of Arts with honors

APPENDIX A

COVER SHEET TO BE APPENDED TO ALL THESES

By law, copyright in your thesis belongs to you, the author, including all rights of publication and reproduction. Only the copyright holder may determine what others may do with the thesis.

STEP 1. What should the library do with your thesis? (Choose one)

☒ Share my thesis

You authorize the library to share your thesis with others according to the Creative Commons license selected in step 2. (See Appendix B for an explanation of the three options.)

☐ Place an optional embargo on my thesis

The library should not allow anyone except college officials to see its copy of my thesis until January 1st of the year _____.

Under this option, you ask the library to prevent anyone else from accessing its copy of your thesis. At the end of the lockdown period, the library will distribute its copy of your thesis according to the license you choose below. (See Appendix B for an explanation of the three options.

STEP 2. Select a license for your thesis. (Choose one)

☐ CC BY-NC-ND

☐ CC BY-NC-SA

☒ CC BY

The type of Creative Commons license you choose determines what others may and may not do with your thesis. (See Appendix B for an explanation of the three options.)

STEP 3. Sign.

Please note that, by choosing to make your thesis available (whether now or at a future date), you are waiving any protections of the Family Educational Rights and Privacy Act (FERPA) that may apply with respect to your thesis as of the date specified. FERPA generally restricts disclosure by the college of records related to your education.

Rachel Willick	Rachel Willick	05/08/2025
Student Signature	Student Name	Date
David Hanneke	David Hanneke	2025-5-8
Thesis Advisor Signature	Thesis Advisor Name	Date

Abstract

Kirigami is a variation of origami, altering the appearance and behavior of a thin sheet of material using folds and cuts. Kirigami is used to create ultralightweight, bi-directional springs. The spring design is optimized by creating iterative patterns of springs using B-spline curves characterized by three control points and a ribbon-width scalar. Each design is then characterized using a numerical solver to determine the force necessary for a given displacement of the spring. Force and displacement are then measured experimentally for selected patterns to confirm model accuracy and develop numerical relationships between spring patterns. We then create a library of springs from which we can choose the desired amount of force for a given use case. The primary use case is tensioning to maintain the orientation of photovoltaic (PV) panels in the Space Solar Power Project.

Acknowledgments

Thank you to my advisors, Professors Stephen Nonnenmann at the University of Massachusetts, Amherst and David Hanneke at Amherst College for helping to provide me with the opportunity to do this research. Thank you to all the members of the Space Solar Power Lab at the California Institute of Technology for being so welcoming and helpful this past summer, especially my mentor, George Popov.

I would like to thank the Amherst College Department of Physics and Astronomy, the Resnick Sustainability Institute at the California Institute of Technology, and Professor Nonnenmann for providing the funding and equipment access necessary to complete this project.

I would also like to thank my friends, family, and partner for being willing to read endless iterations of this thesis and supporting me through the entire process.

Contents

Chapter 1

Introduction

Somewhere, something incredible is waiting to be known.

Carl Sagan

The Space Solar Power Project (SSPP) is a collaboration between three labs at the California Institute of Technology (Caltech), which are headed by Harry Atwater, Ali Hajimiri, and Sergio Pellegrino. The goal of the SSPP is to create spacecraft that collect solar energy using photovoltaics (PV) and radiate the energy using radio-frequency (RF) transmitters to receivers on Earth at microwave frequencies, which avoids large losses of solar energy when passing through the atmosphere, as the atmosphere is mostly transparent to microwave radiation. The energy flux from the Sun at the top of the atmosphere is about 1313 W/m^2 , whereas the

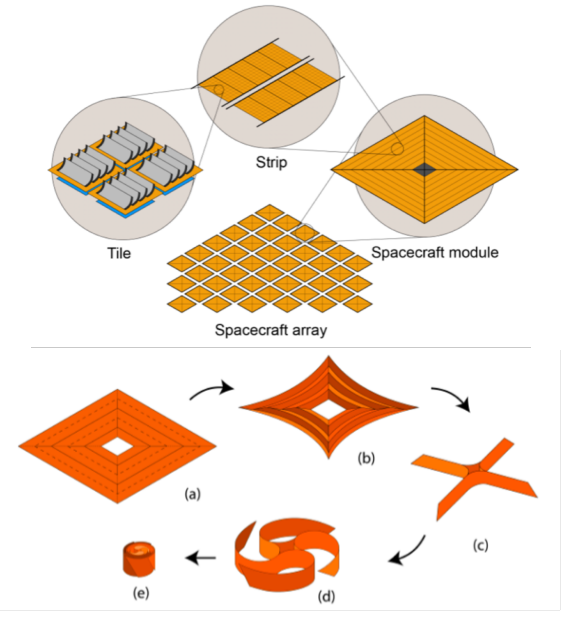

Figure 1.1 Modularity and folding and deployment method of Caltech Space Solar Power Project. Reproduced with permission from (Abiri et al., 2022)

flux at the surface of the Earth is only about 343 W/m^2 (Frame & Revell, 2023). This loss across the atmosphere occurs due to a combination of reflection of energy off of clouds and the atmosphere, and absorption and re-radiation of energy at all layers of the atmosphere. Re-radiated energy is emitted evenly in all directions, so approximately half of all absorbed energy is radiated back into space. Further, PV panels, colloquially known as solar panels, on the surface can only collect energy while in direct sunlight. This means solar panels are ineffective at night and during cloudy or stormy weather. This is particularly challenging in areas with frequently inclement weather, and in regions in the far north and south, as these regions can have months with little to no sunlight. Space solar collection can generate energy 24 hours a day, regardless of surface weather. The combination of these factors leads to an approximate eight-fold increase in potential power from space solar in comparison to surface solar (Abiri et al., 2022).

	Launch Capacity (tonnes)	
Rocket	LEO	GTO
Starship		150
Falcon Heavy	64	27
Space Launch System		46
Antares	8	
Pegasus XL	0.5	
Atlas V	18	9
Electron	0.3	
Falcon 9	23	8

Figure 1.2 Table showing the payload capacities and launch altitudes of various rockets. (SpaceX, 2025a), (SpaceX, 2025b), (NASA, 2024a), (NASA, 2024b), (ULALaunch, 2025), (RocketLab, 2025), (SpaceX, 2025c)

The main obstacle to implementing space solar is in the difficulty and cost of launching spacecraft to collect solar energy in space. The two main factors that influence the cost of space launches are the weight of the spacecraft and whether the solar array can be launched in completed form or requires assembly in space. It is more expensive to launch larger and heavier payloads, and the types of rocket that can carry the largest payloads are launched less

frequently. Figure 1.2 shows launch capacities and what orbits various rockets can achieve. The two orbits considered here are low Earth orbit (LEO) and geotransfer orbit (GTO). LEOs are cheaper and easier than GTOs to launch to as they are much closer to the surface, but receive less consistent sunlight. GTOs are also a type of geostationary orbit, which means that a space solar system in GTO would require only one receiving station on Earth, as it would remain fixed over one location. A space solar system in LEO would require a chain of receiving stations, as this orbit moves relative to the surface of the Earth. Both of these orbits are currently in consideration for the SSPP. So for example, in practical applications, a payload weighing 8 tonnes launching to GTO could be carried on the Antares, Atlas V, or Falcon 9, or it could be one part of the whole payload on the Falcon Heavy or SLS. However, a 10 tonne payload going to GTO could only be carried on the Falcon Heavy or SLS.

For these reasons, the Caltech SSPP is developing ultralightweight, autonomously deployable solar spacecraft. These spacecrafts are highly modular and condense the PV panels and RF transmission elements into small 10cm x 10cm tiles, each weighing only a few grams. Each tile is flexible in order to be stowed for launch. The tiles are then assembled into 2m x 60m strips, which combine into a 60m x 60m module (Figure 1, top) weighing about 650 kg (Abiri et al., 2022)(CalTech, 2017). The modularity has multiple benefits: it allows for easy scalability of the design, simple construction, redundancy in the case of damage to any one tile, and more compact, and thus easier, packaging, storage, and deployment (Figure 1.1, bottom). This redundancy is very important because, due to the nature of space and the vast distances and high speeds of orbits, they will be almost impossible to service, and any possible service or repair would be prohibitively expensive. Each tile is made up of layers of PV panels, RF transmitters, and integrated electronic circuits that are each fixed into a carbon fiber frame. The frames are then connected to one another with longer carbon fiber fixtures.

However, the tiles and the carbon fiber frames have very different thermal expansion coefficients (CTE), which causes them to expand and contract at different rates as the tiles

change temperature. This discrepancy leads to warping and shifting of the tiles within their frames, which disrupts both the collection of solar energy and the transmission of the acquired energy back to Earth. In order to combat this warping, this research builds on the work of George Popov at Caltech to design and characterize thin-film springs based on kirigami designs that will provide even tensioning for the tiles even as the tiles and frame expand and contract. This is not a problem for ordinary space-based solar arrays, as historically PV panels have not been flexible, and so will not experience warping. Stiff solar panels also don't need frames for support, so the changes from differences in CTE can be accounted for in the connections between panels.

In Chapter 2 we review other applications of kirigami, as well as how kirigami has been used previously to create thin-film springs, and compare the characteristics of existing springs against our goals. We also characterize the material, Kapton®, that the springs are made out of. In Chapter 3, we explain the methodology by which the springs were designed, manufactured, and tested, as well as sources of error that arise through our design and manufacturing process. In Chapter 4, we detail the findings from both computational and experimental tests and discuss the results. We show that the springs have the desired three-phase force-displacement curve, provide forces and displacements on the correct scale for our usage, and provide a sufficient amount of variation between patterns. In Chapter 5, we assemble the spring library and detail an example use case: a fixed (CTE=0) carbon fiber frame, and a representative tile made entirely of Kapton®.

Chapter 2

Background

> The whole of science is nothing more than a refinement of everyday thinking.
>
> Albert Einstein

2.1 Space Solar Project Tile Connections

As discussed in the Introduction, the PV/RF panels are fixed into carbon fiber composite frames and are exposed to wide temperature fluctuations. Over the course of these fluctuations, both the frames and the panels themselves expand and contract. However, because they are made of very different materials, the expansion of the panels and the frames occurs at different rates. This poses a problem: if the panels are attached directly to the frame, then the mismatch of thermal expansion causes warping and stretching of the PV panels, which limits the efficiency of the solar collection, and can disrupt the transmission of the collected energy to the surface (see figure 2.1).

It is possible to avoid warping and stretching by adding springs between the PV/RF panels and the frames (Figure 2.2). The following criteria are necessary for the functionality of these springs. The springs should be small, as they take up some of the space that could otherwise be used for energy collection, reducing the efficacy of the panels. In our case,

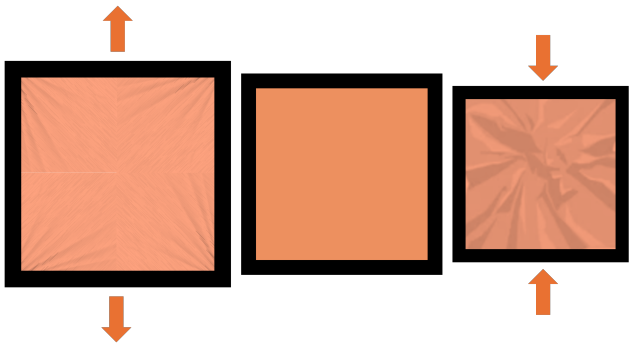

Figure 2.1 Depiction of how stretching and warping occurs as the solar panels and frames expand and contract as temperature changes. The left image shows how when the frame expands more than the panel, the panel is put under tension and develops stress folds. The middle image shows the panel and frame at original sizes. The right image shows how the panel crumples and buckles when the frame shrinks more than the panel.

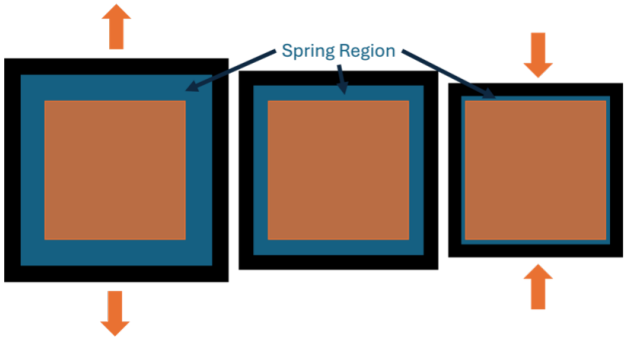

Figure 2.2 Depiction of how the introduction of springs keeps the panel flat and tensioned even as the frame changes size relative to the panel. It can be seen that the springs absorb the changes in size without passing the forces on to the panel.

we use strips of springs that are 1cm wide at rest to maximize energy collection area, but retain enough room for functionality and the elimination of warping and stretching. They also must have a large enough displacement range to cover the whole possible variance in the width of the gap between the panel and the frame. The springs should use as little force as possible to hold the panel flat so as to avoid damage to the panels. The springs also want to be bi-directionally stretchy in the plane of the panels to add stability. They should be thin and flexible so they can be rolled without damage and stowed in the compressed form of the larger space solar module for launch. The spring must also be tunable due to the fact that the exact material characteristics of the PV/RF films and carbon fiber frames are still in development, and the subsequent force and displacements are not yet known.

2.2 Kirigami

Kirigami is a cousin of origami that incorporates cuts as well as folds to create unique patterns. Perhaps the most commonly seen kirigami is the paper snowflakes which are made by folding a square into six equal parts and then cutting paper out of the sides and top to form a repeating pattern on each section when unfolded. Another common application is honeycomb packing paper, which is paper with small cuts that puffs when stretched and serves as a protective packing layer when shipping goods similar to bubble wrap. However, despite common encounters in everyday life, there has been limited academic study of the uses and characteristics of various kirigami patterns. The most frequently studied kirigami spring pattern is a "net-like multistable pattern" (Ai et al., 2021), which consists of a series of horizontal cuts staggered like bricks. This pattern is referred to in this thesis as "net pattern" for brevity and clarity. The net pattern has unidirectional stretch along the primary axis. The net pattern is useful because it is simple to cut with laser cutters (Taniyama & Iwase, 2019), optical lithography (Blees et al., 2015), or cutting plotters (Isobe & Okumura, 2016), and is relatively easy to characterize due to its simplicity and limited variables.

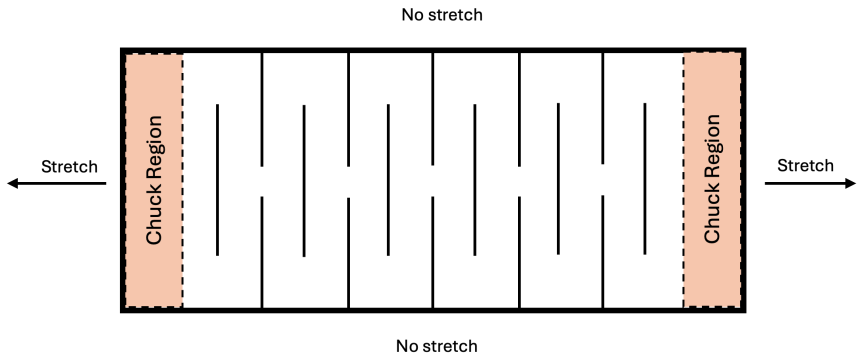

Figure 2.3 The Net Pattern consists of a series of staggered parallel cuts which allow the sample to be stretched along the in-plane axis perpendicular to the cuts. This stretching involves out-of-plane buckling and deformation.

Taniyama et. al. (Taniyama & Iwase, 2019) did an extensive characterization of the net pattern and determined that it could be modelled by a series of springs along the stretch axis. A series of springs is required for a couple reasons. First, the deformation along the

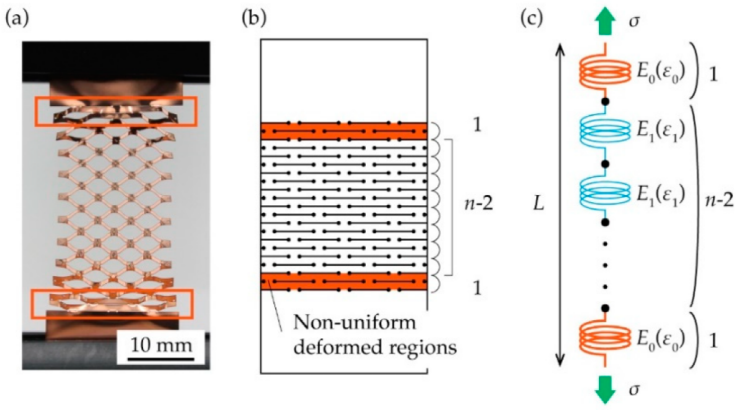

Figure 2.4 The out-of-plane buckling the the Net Pattern kirigami springs is non-uniform and can be modelled using a series of simple springs. Reproduced with permission from (Taniyama & Iwase, 2019)

cuts is not uniform along the length of the pattern; the cuts near the chuck region, where the sample is held for testing, experience less deformation than cuts near the center (see Figure 2.4). In other words, the sections of the spring near the middle stretch more than those at the ends, as can be seen in Figure 4.2a. Second, the patterns experience a three-phase stretching protocol, beginning with a linear "pre-buckling" phase in which the stretching is all in plane, and is strongly affected by the material characteristics. The pattern then goes through a buckle in which the segments between the cuts pop out of the plane to allow for greater stretching. The third phase, the "post-buckling" phase, is "softer" or "stretchier" than the pre-buckling phase and is also fairly linear. The third phase occurs when no more stretch can be gained through buckling, and the intrinsic characteristics of the material once again become dominant. This third phase can also be modelled linearly. The transition points between each phase can be sharp, with a distinct buckling point, or smooth, depending on the material and the style of cutting. (Taniyama & Iwase, 2019)(Isobe & Okumura, 2016). Rafsanjani et. al. (Rafsanjani & Bertoldi, 2017) studied a variation of the net pattern that uses an array of mutually orthogonal cuts (Figure 2.5), referred to henceforth as "square net pattern," to allow for stretching in multiple directions and more complicated buckling patterns. This square net pattern is very useful, as it can function as a spring in any in-plane

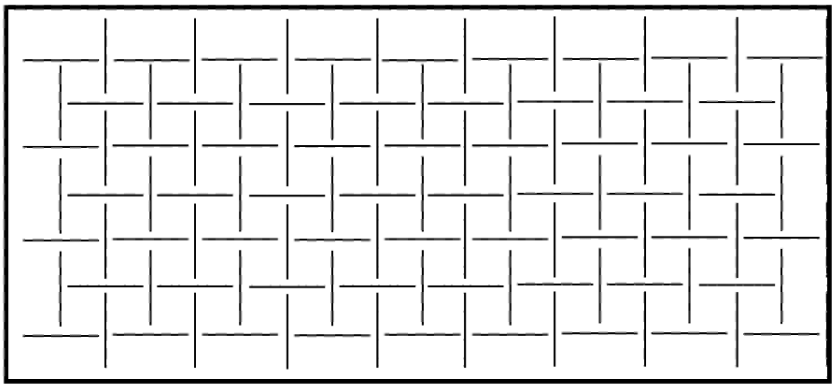

Figure 2.5 The Square Net pattern consists of mutually orthogonal cuts that allow the sample to be stretched in any direction, as well as to buckle into a stable, stiff configuration.

direction and can lock into a fully buckled state in which it gains significant stiffness. A 127 μm thick sheet of this pattern in the buckled state can hold a 20 g weight suspended between two pillars. This pattern seems to stretch most uniformly when pulled at a 45° angle to the cuts. While these patterns are all very useful, the constraints of our project have forced us

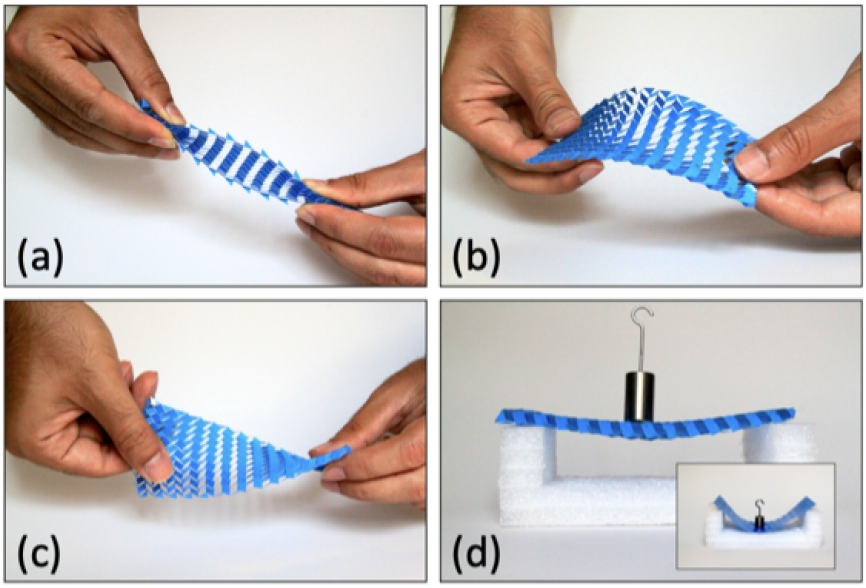

Figure 2.6 The Square Net pattern in the stable, stiff buckled configuration. Reproduced with permission from (Rafsanjani & Bertoldi, 2017)

to develop a new pattern specific to our criteria (see above in 2.1). The net pattern can only stretch in one direction, and the square net pattern is too stiff for our uses. For these reasons, we developed a modified cross pattern with a curve to allow for extra stretch. This

pattern is called the "Double S Pattern" because the space between the cuts resembles two perpendicular S shapes. This pattern meets most of our criteria but lacks tunability.

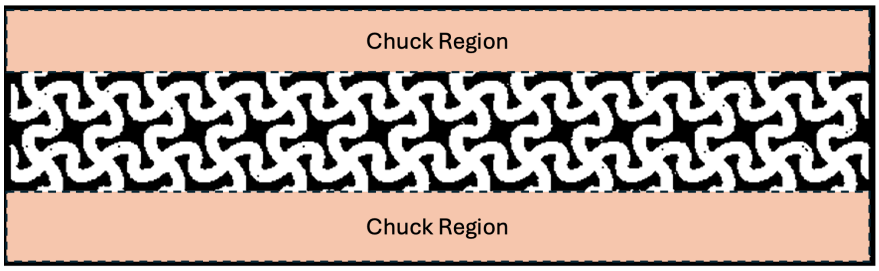

Figure 2.7 Image of the Double S Spring Pattern showing the chuck regions and the cross shaped, angled cuts that are reminiscent of the Square Net Pattern. The white parts are the remaining material and the black parts are where the material is removed.

In order to make this pattern tunable, we shift from looking at the cuts to the spaces between them, and we find that the general Double S shape can be modelled by four identical curves (see 3.1) each rotated 90° from one another. To model these curves, we had a couple options. In general curves can be modelled with polynomials, however, for our purposes, and due to the asymmetry of the curves we want, we would need to use high degree polynomials rotated at an angle. Instead, we can use splines, which are piecewise polynomial functions, each defined over a short region. Essentially, splines are multiple polynomial functions connected end to end to create more complex geometries. We used B-splines with two fixed endpoints and a movable "tuning" point to add tunability to this design, as will be discussed in Chapter 3.

2.3 Kapton®

When deciding on an ideal material from which to manufacture the springs, there are a number of criteria that must be met. First, the material should be lightweight to keep the overall system weight down. Second, the material should also be flexible out-of-plane to allow for rolling and buckling, while stiff in-plane to ensure that the stretching comes from the spring pattern and not the intrinsic material characteristics. Third, the material should

be relatively homogeneous for more consistency in stretching. Fourth, the material must retain its properties over a wide range of temperatures. Finally, the material must be able to survive in space long-term with minimal degradation.

The most difficult characteristics to achieve are retention of properties across temperatures and lack of degradation. Fortunately, these two characteristics often go together. Many of the materials which are sufficiently durable throughout temperature changes and long-term existence in space are metals and ceramics, which are not sufficiently light-weight or flexible for our uses. (HeegerMaterials, 2024) This leaves polymers, which are repeating chains or arrays of smaller molecules called monomers. The polymers with the highest stability across temperatures are polyimides, which are polymers of monomers that contain imide functional groups, a collection of oxygen, nitrogen, and carbon atoms (Greene, 2021). The repeating nature of polymers also creates high levels of homogeneity across a material. Polyimides can be made in films as thin as 25 μm, which provides the necessary flexibility out of plane.

For this project, we chose to use Kapton®, a 50 μm thin-film polyimide manufactured by DuPont. We chose to use a 50 μm polyimide rather than a 25 μm polyimide for added durability and stiffness in the springs. Kapton® is also relatively inexpensive and readily available, making it ideal for both prototyping and manufacturing. The availability and affordability of Kapton also enables the possibility of more widescale applications of the kirigami research beyond the scope of this paper.

Due to a variation in published values for the Young's Modulus of Kapton®, we decided to experimentally determine the Young's Modulus of the particular Kapton® (100 HN) we used for these experiments. The first step was to examine Kapton® under an optical microscope, which allowed for characterization of the thickness and homogeneity of the Kapton®. Next, the Young's Modulus of the Kapton® was measured using an Electroforce Universal Testing Machine to measure force and displacement. Force and displacement were then converted into stress-strain graphs based on the initial dimensions of the sample

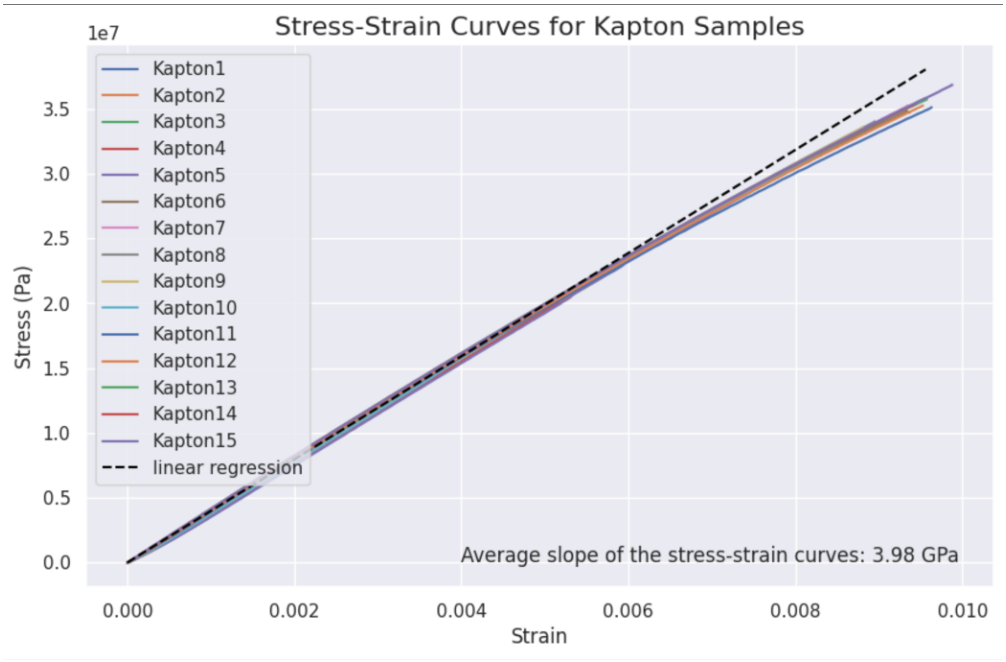

Figure 2.8 Plot of Stress-Strain Curves for solid Kapton® samples. The slope of the linear regression of the average of all curves is 3.98 GPa, which is in agreement with the literature values for the Young's Modulus of Kapton®.

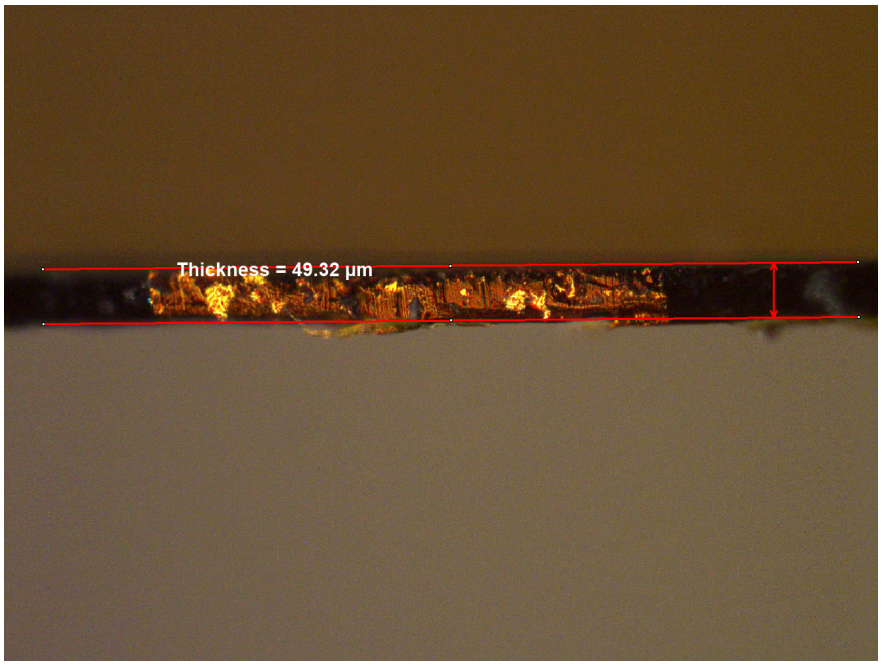

Figure 2.9 Annotated optical microscope image of the thickness of Kapton®.

measured. For these measurements, 15 tests were conducted on three different samples, each measuring approximately 20x100mm, with the force directed along the long axis. A

linear regression was performed on the average stress-strain graph compiled from all trials to find the Young's Modulus. We found the Young's Modulus to be 3.98 GPa based on this method (see Figure 2.7). This is not in agreement with the manufacturer's data from DuPont (DuPont, 2022), which states Kapton® has a Young's Modulus of 2.76 GPa, but is in agreement with the paper "Study on Young's modulus of thin films on Kapton® by microtensile testing combined with dual DIC system" by Wei He et. al. (He et al., 2016). It is possible that this difference arises due to the fact that it is not clear within the DuPont data sheet whether they are reporting a Young's modulus for 25μm thick or 50μm thick Kapton®. This would explain why the DuPont value is lower, as a thinner sheet of material would have a lower Young's Modulus.

Chapter 3

Methods

> Science is fun. Science is curiosity. We all have natural curiosity. Science is a process of investigating. It's posing questions and coming up with a method. It's delving in.
>
> Sally Ride

3.1 Developing Patterns

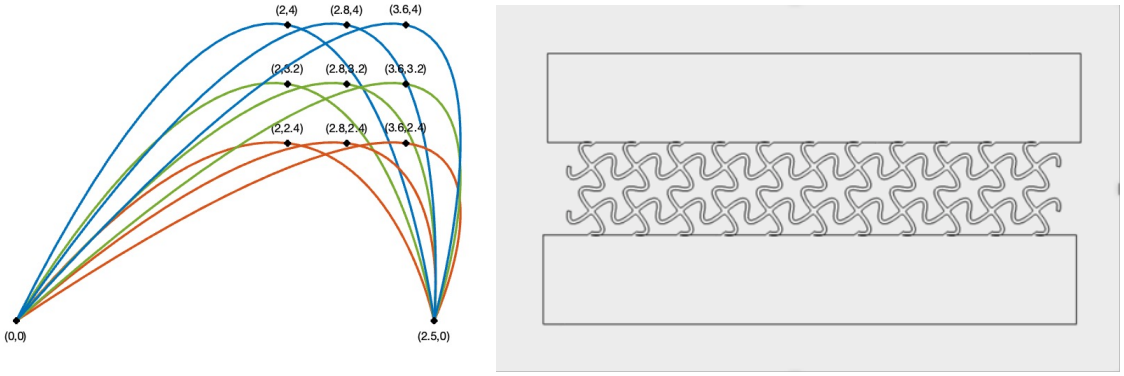

Figure 3.1 Left: The nine b-spline variations that were made into spring patterns for these experiments, all with fixed start and end points, and a variable middle control point. Right: an example of a b-spline made into a spring pattern with chuck regions. The spring region and each of the chuck regions are 1x6 cm, so the total pattern is 3x6 cm.

Once we decided to use splines, we had to choose what kind of spline to use. By definition, a spline is a piecewise polynomial function that is continuous across the function itself, as

well as the first and second derivatives. This provides a high degree of smoothness which is important for our project because it reduces the likelihood of weak points in the springs. Historically, the term spline arises from shipbuilding, in which long pieces of wood are bent, and will naturally form a smooth curve with the lowest potential energy. If the wood is only fixed at the ends, it will form a smooth, symmetrical curve. However, if you add a post two thirds of the way to the right side of the piece of wood, and bend the ends to the same points, the piece of wood will form a smooth curve that is skewed to the right. In mathematically generated splines, the control points serve much the same purpose as the post in the shipbuilding example.

There are a number of different types of splines, primarily defined by the degree of the polynomials that comprise them. The most common splines use cubic polynomials for simplicity and to reduce computational time. Within cubic splines, there is variation in how the control points are defined. Some splines, such as Hermite and Catmull-Rom splines, have control points defined by position and the derivative of the curve at the control point. This leads to continuity in the first derivative, but can sometimes have breakdown in the continuity of the second derivative. These splines are also more complicated to tailor to varying control points. Bezier splines are defined only by the position of the control points, and the curve is evaluated such that it fits completely within the polygon created by the control points, but will not pass through all the control points. However, Bezier splines can also have discontinuities in the second derivatives. This leaves natural cubic splines and B-splines. Both of these types of splines have continuous first and second derivatives, but have different control points. Natural cubic splines are more similar to Hermite and Catmull-Rom splines in that the generated curve will pass precisely through all points. However, changing any point can have significant ramifications on the curve as a whole. B-splines are like Bezier splines in that they are contained within the polygon defined by the splines, but they pass much closer to the control points, and the nearness can be increased by using repeated control points to weight the function. B-splines are also slightly more consistent in behavior

when moving the control points. (CMU, n.d.)

We chose B-splines over natural cubic splines because we are primarily concerned with the variation between similar spline patterns, rather than the exact values of the spline, and B-splines better serve that purpose. We generated the splines in MATLAB (see full code in Appendix B) using the bsplinepolytraj function, which accepts control points, a time interval, and time samples (Mathworks, 2024). This function then returns the trajectory positions, velocity, acceleration and polynomial coefficients of the B-spline. We only used the trajectory positions in our spline generation. We defined the splines using four control points, one fixed at (0,0), two fixed at (2.5,0), and one variable. The (2.5,0) control point is repeated to weight the spline slightly in the direction to improve tessellation of the individual springs into a grid. Weighting the endpoint helps the generated spline to end at precisely (2.5, 0). The third point is allowed to vary in both x and y to create variations in the depth of curvature and total ribbon length. Once the spline has been generated, we rotate the spline about the origin to create four evenly spaced arms. Next, we use the offsetCurve function in MATLAB to create width on each arm, based on a variable scalar. After each arm has width, the inside ends of each arm are connected using small symmetric splines to create a smooth fillet to prevent tearing at the joint. Once the individual spring has been created, it can be tessellated into the desired configuration, generally 2x12 spring instances, and chuck regions are added to provide a place for even application of force (see Appendix A for detailed images).

When deciding of possible spring arm ribbon widths, we were bounded on both sides by non-negotiable criteria. The minimum ribbon width, 0.6 mm was chosen because springs thinner than 0.6 mm became too fragile to work with effectively. The maximum ribbon width of 0.8 mm was chosen because that is the widest the spring arms can be without overlapping on the more densely tessellated patterns such as (3.6, 4.0) (see Figure 3.2).

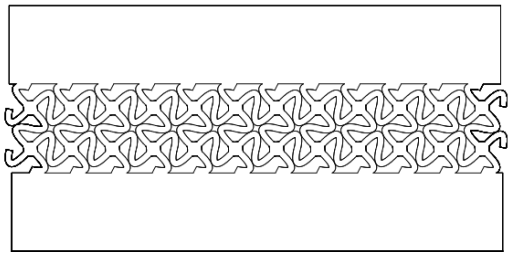

Figure 3.2 (3.6, 4.0) pattern with 0.8 mm spring arm width, showing that any increase in width would lead to overlap of lines.

3.2 Numerical Analysis Using Abaqus

Abaqus is a software suite used for finite element analysis (FEA) and computer aided engineering. FEA is a technique used for modelling complex geometries by breaking them down into smaller elements and applying mathematical equations based on the laws of physics to each part. Here we are using FEA to model the application of force to our samples. We can envision this working in its simplest method as a collection of strings tied end to end. When the first string is pulled, it pulls on the second string, which pulls on the third string, and so on until the end is reached. In this way, we can estimate the behavior of the combined string based on the behavior of each of the individual springs.

Our first step was to import the .dxf file created using Matlab into Abaqus as a sketch file, which can then be used to generate a part. Next, we used a Python code (see code in Appendix B) to generate the material and section parameters with Kapton® modelled as a homogeneous shell with a thickness of 50 μm, density of 1.42 g/cc, Young's Modulus of 3.98 GPa, and a Poisson's Ratio of 0.34 (DuPont, 2022). Next, the created part is assigned to the homogeneous section, and the chuck regions are partitioned from the springs for better meshing. Meshing is the process by which the larger part is divided into the finite elements that the software will use for computing. It is necessary to have small elements in order to accurately model the large-scale behavior, but the maximum number of elements is also defined by the limits of the available computing power. The "seed" is a metric of how large each element is, and the curvature parameter is essentially a measure of how long of an arc

is allowed in any given element. The chuck regions are meshed using a global seed of 0.8, a curvature parameter of 0.01, and a structured quad mesh. The springs are meshed using a local seed of 0.5 and curvature of 0.007, with a free quad-based mesh. The term "quad mesh" indicates that the elements are each quadrilaterals, rather than triangles. The chuck regions have regular quadrilateral elements (squares), while the springs use quadrilaterals of varying shapes and angles. These parameters are based on the paper "Wrinkled Membranes Part III: Numerical Simulations" by Wong and Pellegrino (Wong & Pellegrino, 2006), who also developed the protocol below for the buckling step, which was modified for this application. The part is then used to create an assembly on which the analysis will be run.

A second piece of Python code is used to generate three separate steps. The "Initial" step sets up the base conditions, in this case a set of boundary conditions that fix the right edge of the sample "encastre", or fixed in terms of both displacement and rotation. The next step is a "Buckle" step, in which we propagate the boundary conditions from the "Initial" step and add a shell edge load of -0.01 N to the left edge of the sample. We then use the Lanczos solver to request 10 eigenvalues greater than 0. The Lanczos solver is a method of generating the eigenmodes and eigenvalues of a system, which show us the natural vibration frequencies of a system. In this case, the eigenvalues show us the most likely places for buckling, and allow us to determine where to seed imperfections into the sample, as imperfections are necessary in order for buckling to occur. The eigenvalues from the Lanczos solver are imported into the model as a "File Imperfection" with an imperfection scalar of 0.01. This scalar determines the size of the imperfections relative to the sample itself. The third step is the "Load" step, where we once again propagate the "Initial" boundary conditions and add another boundary condition to pull the left edge of the sample 3-6mm to the left, depending on the sample. After the "Load" step, data is taken from all points along the left edge of the sample, and force is collected as a sum of the reaction force on all points, while displacement is collected as the average displacement for all points. It is important to note that the "Buckle" and "Load" steps cannot be run at the same time – the "Load" step must be suppressed to run

the buckling analysis, and the “Buckle” step must be suppressed to complete the “Load” step. (see video of ”Load” step in A).

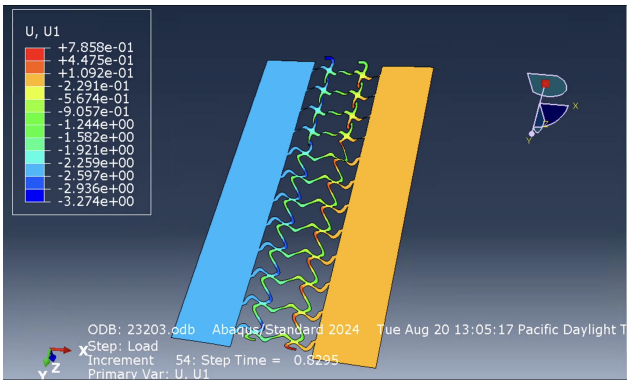

Figure 3.3 Image of the modelled sample in a fully buckled state, showing how the ribbon arms twist out of plane to facilitate buckling.

3.3 Experimental Testing

3.3.1 Laser Cut

Initially, our patterns were laser cut, as per previous work by Taniyama and Rafsanjani. Laser cutting is a very efficient way of cutting detailed patterns, as the laser head can move quickly and there is little risk of tearing or dragging the sample. Laser cutters also cut effectively through Kapton. Commercial laser cutters also accept .dxf file types which have high precision and high reproducibility. These patterns have been cut on multiple laser cutters (GCC LaserPro Spirit GLS and Universal Laser Systems Inc. Model: XLS10), and it was determined that it is ideal to cut Kapton at high speed, with the minimum amount of power necessary to cut cleanly through the sample. For the GCC LaserPro Spirit GLS, the optimal settings were 95% speed and 100% power. The high speed is important because it reduces energy dumping into the sample which causes melting and burning effects (see fig 3.4). Once the patterns are cut, acrylic or 3d printed “handles” are epoxied on to the chuck regions to provide a stiff surface to grip during experiments for an even distribution of force.

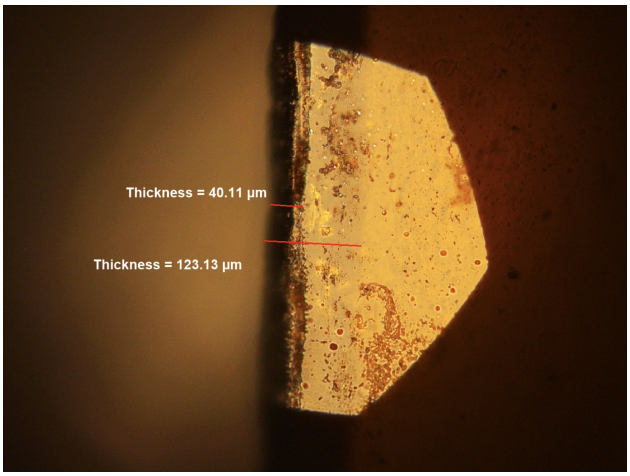

Figure 3.4 Optical Microscope image showing the burning and melting effects from the laser cutter, with melting spreading 123 µm into the sample, and burning of 40 µm.

These handles measure 1x6 cm to match the dimensions of the chuck region.

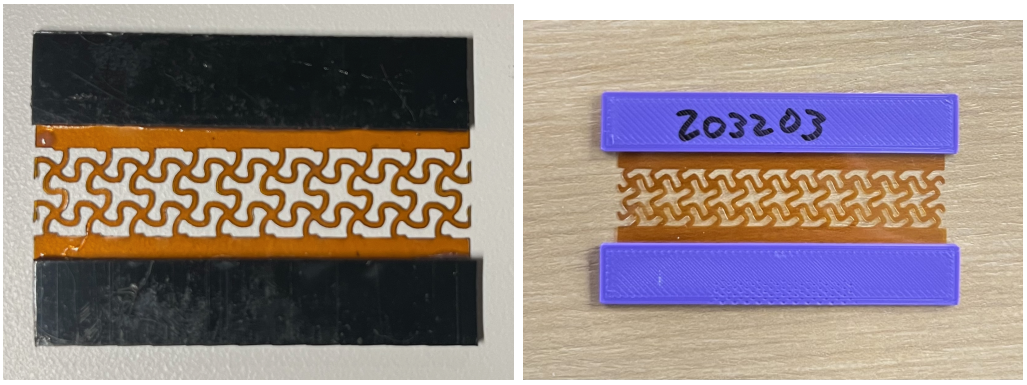

Figure 3.5 Left: Photograph of Double S Pattern with acrylic handles epoxied on to both chuck regions. Right: Photograph of B Spline Pattern with 3d printed handles epoxied on. These patterns are all 3x6 cm.

Burning

The burning and melting have a number of notable effects. First, almost 200 µm of material are burnt off the edges of the spring arm. This reduction in width leads to a reduction in the stiffness of the springs, in this case by almost 50%, as can be seen in Figure 3.7 and 3.8.

The burning also creates a "Dog Bone" effect, where the edges of the Kapton® become

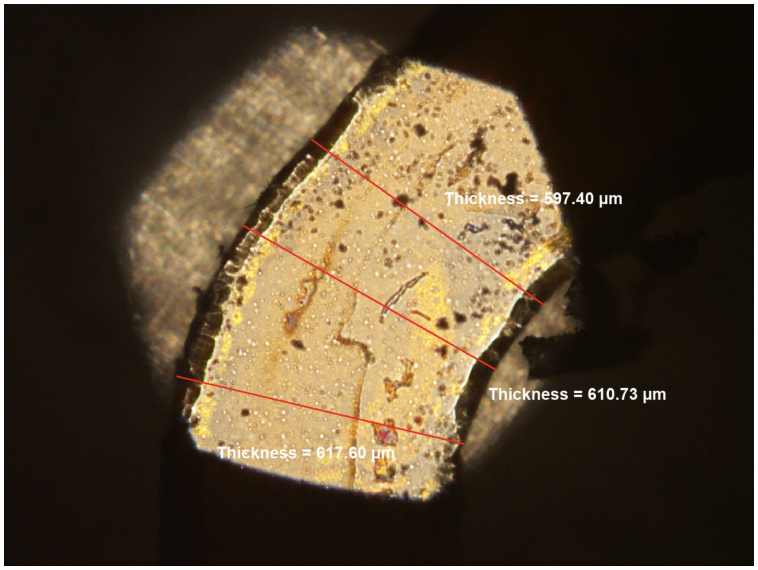

Figure 3.6 Optical microscope image of a 0.8 mm (800 µm) spring arm that has been burnt down to about 615 µm. This image also shows some microscale imperfections in the Kapton®.

Figure 3.7 Modelled (orange) and experimentally collected (black) data for the Double-S pattern with all parameters equal showing the initial buckle at about 0.01 cm, a stable region from 0.025cm to 0.1 cm, and a final exponential growth region.

thicker due to melting and a movement of material during laser cutting, similar to how candles become thicker as wax melts and resolidifies. This effect can be seen in Fig. 3.9. The Dog Bone effect leads to increased stiffness in the springs, as it increases the force needed for the springs to buckle because the added ribbon width resists buckling. Given that the Burned Model in Fig. 3.8 was modified such that the width was reduced but no

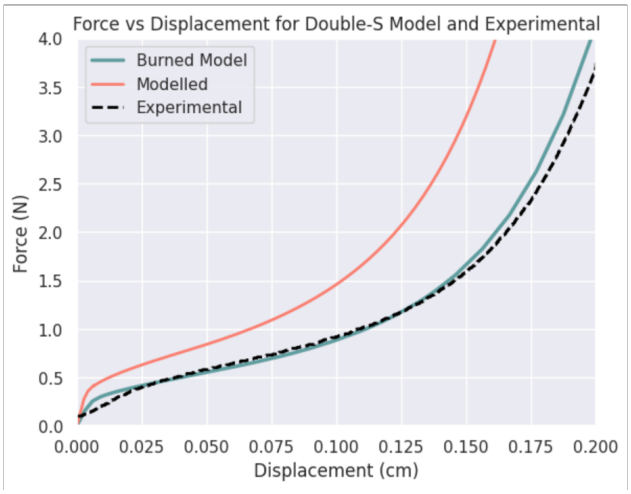

Figure 3.8 Effect of changing the modelled width to match the burned width of the experimental pattern, rather than the input to the laser cutter. Modelling the pattern with the burn effects leads to a model that has results within 4% of experimental values

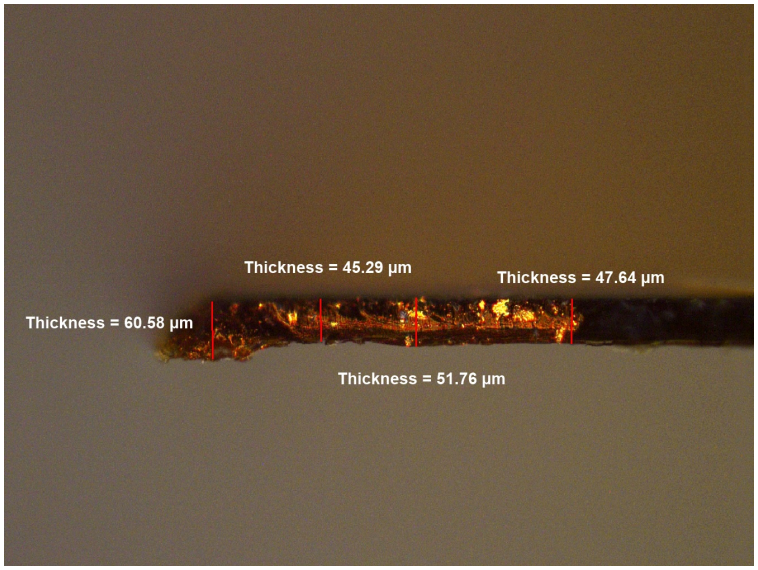

Figure 3.9 Optical microscope image of the "Dog Bone" effect that occurs during burning, in which the edge of the Kapton® becomes about 10 µm thicker than the rest of the sheet.

changes were made to account for the Dog Bone effect, it seems that the reduction in width is the dominant effect. The burning and melting has a final effect, which is to reduce the homogeneity of the Kapton®. This occurs to some degree throughout the entire pattern, as can be seen in Fig. 3.6, but is much more pronounced around the edges, as can be seen in Figures 3.4 and 3.10.

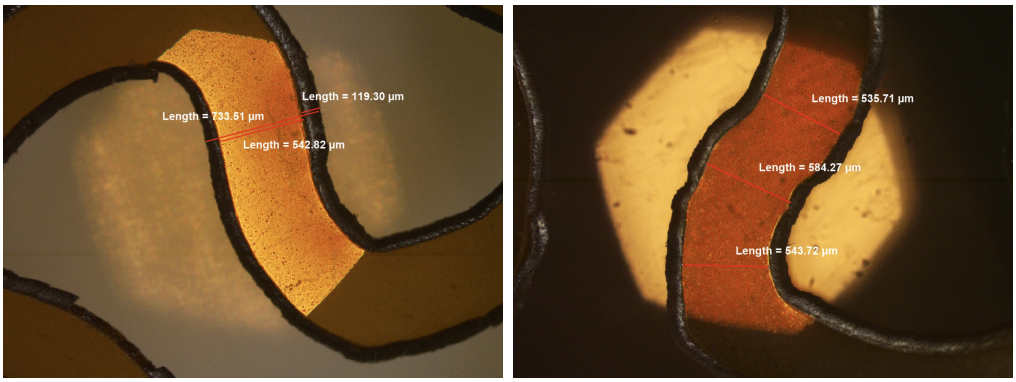

Figure 3.10 These images show how the burning and melting effects lead to a very irregular edge on the spring arms, with what appear to be entire chunks that have become brittle and broken off.

3.3.2 Knife Cut

In order to compensate for the burning that occurred while using the laser cutter, we also cut patterns using a Cricut Joy commercial cutting plotter. For this machine, the files were uploaded as .jpg files (see Figure 3.11), as the .dxf files were too large for the Cricut software. This importation process caused slight changes to the patterns, but these were minimized by filling in the image of the pattern before importing. This infill of the pattern was necessary, because the software is designed such that it will cut anywhere black and white are touching, so if the pattern is imported as lines, the plotter will cut on both sides of the lines, significantly reducing the width of the pattern and causing some warping of the pattern. The same 1x6cm epoxy handles were attached to these patterns, as can be seen in Fig. 3.4. There was an unexpected error that occurred during this process however, in that by uploading the files as .jpgs, the linewidth becomes relevant, and the knife cut patterns gained approximately 0.1 mm on either side of the arms; as a result, the patterns that were nominally 0.6 mm became 0.85 mm, and the 0.8 mm patterns became 1 mm. However, it was necessary to have some linewidth in order to create smooth infill and maintain the exact pattern edges. This led to an issue where two B-spline variations failed, as the increased width led to overlap of adjacent arms. In future work, there are a couple possible solutions to this. First, higher end commercial cutting plotters can accept larger .dxf file types, which

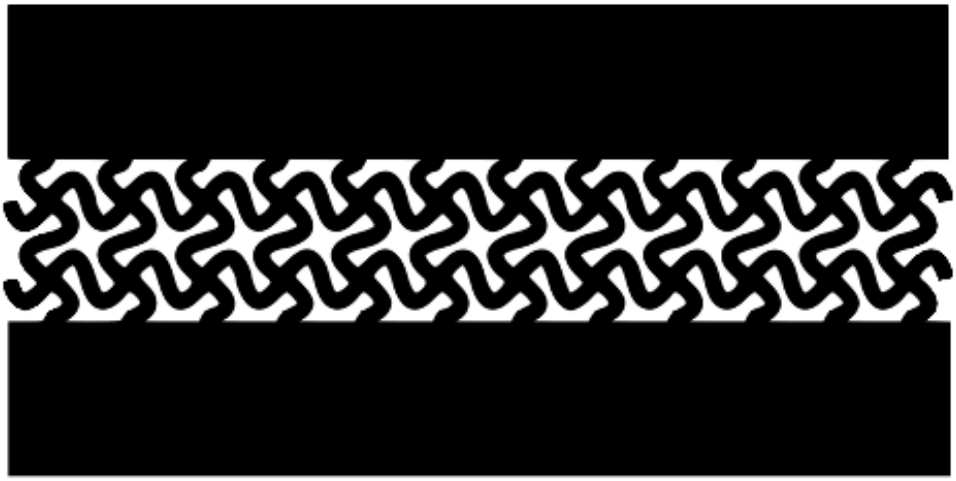

Figure 3.11 Example pattern with a black infill for importation to the Cricut cutter. In this case, the white will be cut away and the black will remain.

would increase precision and avoid this problem. Second, it would be possible to edit the designs to have reduced width such that they became the correct size with the inclusion of the linewidth. This could be somewhat tricky, as .jpg files do not have an intrinsic size, as .dxf files do, so it would take some experimentation to perfect the sizing. There is also an added element of inconsistency in that .jpg files must be resized to the correct 3x6 cm overall dimensions, and inconsistencies between the spring arm widths across patterns indicate that variations of up to 30 μm besides the 0.2 mm general increase occurred during the importation process.

Overall, however, the cutting plotter created patterns that are much more cleanly cut and retain much more of their homogeneity than those cut with the laser cutter. As can be seen in Fig. 3.10 (left), the edges are smooth and even, although there is a slight roll to the edges. However, the roll cannot be seen in an edge on view (Fig. 3.10 right).

3.3.3 Tensile Testing

For our tensile testing, we used an Instron Universal Testing System 34SC-5 (single column) with a 100 N load cell. For each test, we insert the sample into the custom top chuck (see appendix for details), and then clamp it into the bottom chuck with no tension. Next, we

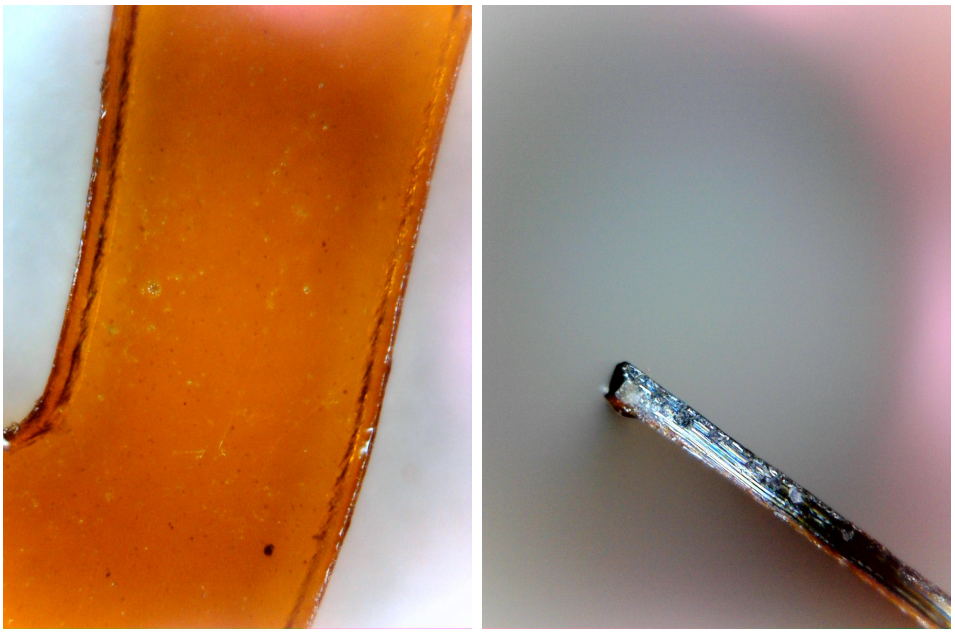

Figure 3.12 Optical Microscope images of patterns cut using the commercial cutting plotter, showing the cleanliness of the cuts (left) and the homogeneity of the edges (right). Note: there is some soot on these patterns as they were stored with the laser cut patterns.

instruct the Instron to slowly increase the displacement, and measure both the displacement and the accompanying force. For each sample, we continue to stretch until the pattern breaks or the force levels off, indicating imminent breakage.

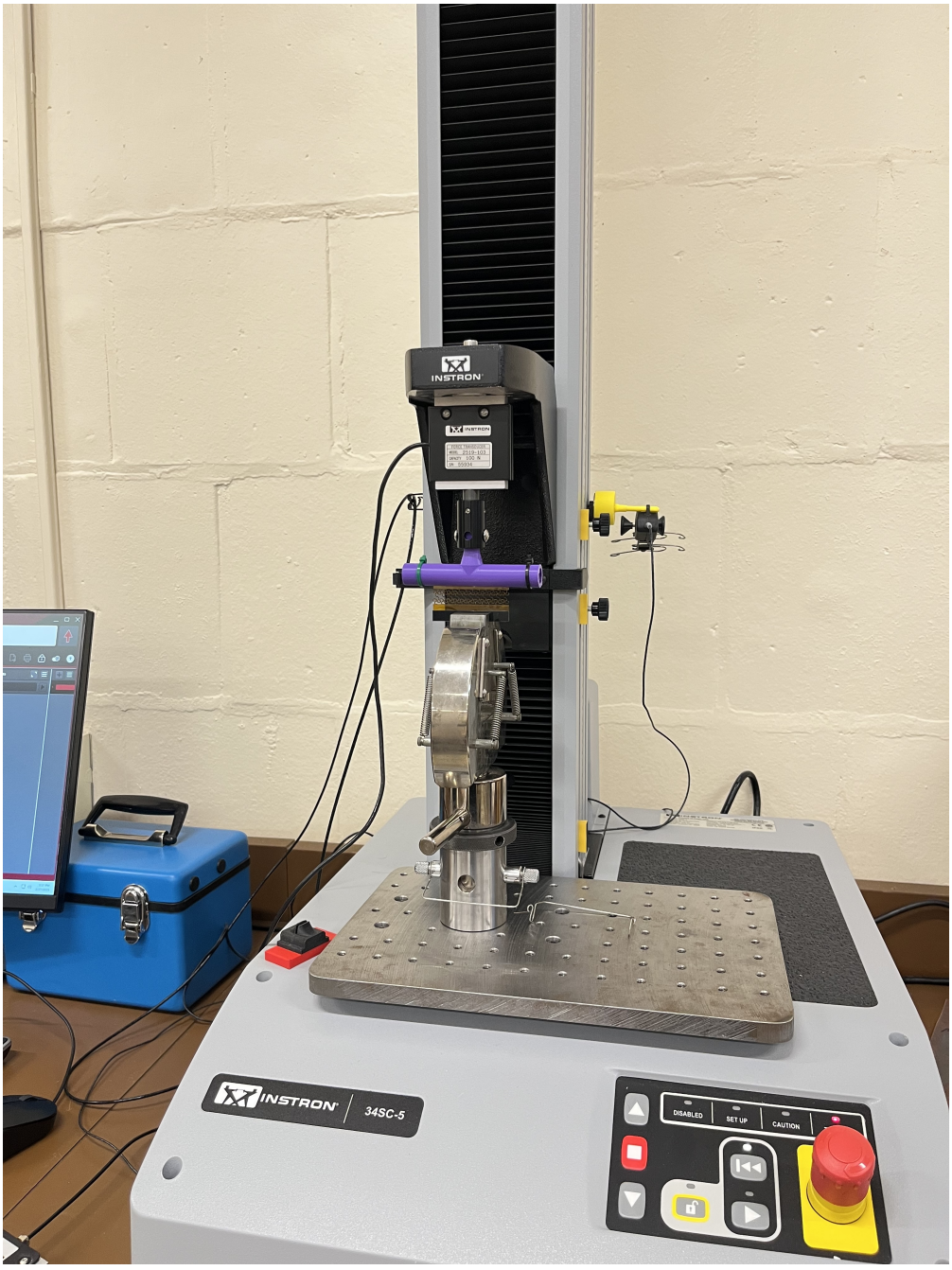

Figure 3.13 Image of the set-up for testing using the Instron Universal Testing System, showing the specialized top grip, sample, and bottom chuck.

Chapter 4

Results and Discussion

> Scientists do not discover in order to know, but rather, they know in order to discover.
>
> Alfred North Whitehead

4.1 Abaqus Results

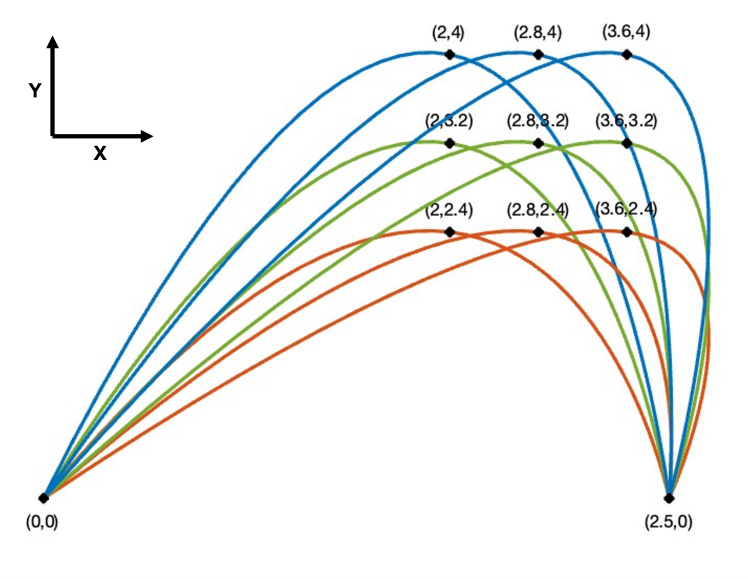

Figure 4.1 The nine B-spline variations that were made into spring patterns for these experiments, all with fixed start and end points, and a variable middle control point.

The results from the numerical modelling will be broken down into three subsections: an analysis of the effects of changing the ribbon width of the spring arms, an analysis of changing the x-coordinate of the splines, and an analysis of changing the y-coordinate of the splines. The x-coordinate represents the horizontal shift of the control point of the spline, as seen in Figure 4.1. This primarily has the effect of reducing the symmetry of the spline by skewing the peak of the curve to the right. Increasing the x-coordinate also leads to a shallower slope on the leading side of the curve, and a steeper slope on the trailing side. The y-coordinate represents the vertical shift of the control point. An increase in the y-coordinate leads to an increased depth of curvature and a longer arc length. The effects of each of these changes are intertwined but can be analyzed individually by carefully controlling the other variables.

4.1.1 Ribbon Width

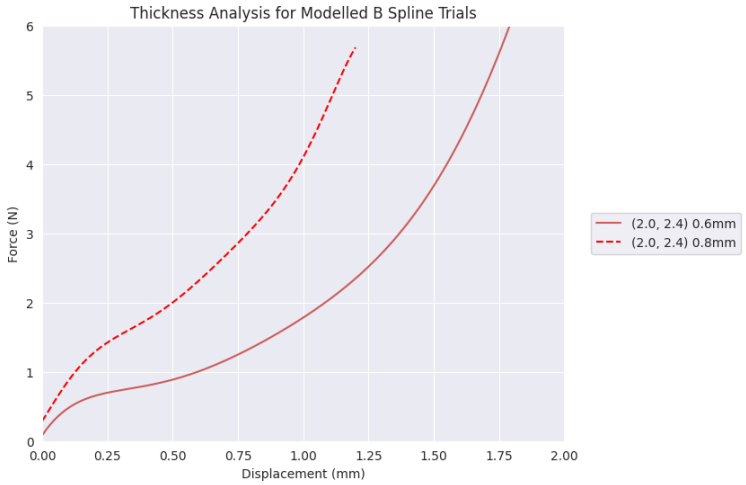

Figure 4.2 This graph shows the variations in spring behavior between the (2.0, 2.4) pattern spring at 0.6mm ribbon width and 0.8mm ribbon width.

We first analyzed the effects of changing the ribbon width by choosing one spline, in this case the spline defined by the control point (2.0, 2.4). It can clearly be seen in Figure 4.1 that an increase in ribbon width leads to an increase in the amount of force needed to displace the spring a given distance. For instance, to pull the 0.6 mm spring 0.5 mm

would require just under 1 N of force, whereas the 0.8 mm spring would require 2 N. This raises the question how the two curves are related. This was evaluated by fitting a 7th degree polynomial to each curve, and then optimizing a transformation from one to the other. We used a transformation of the form $P_B(x) = c * P_A(a * x + b) + d$ [Eqn. 4.1.1] and performed an error minimization in Python (see B). We used 7th degree polynomials because this was the degree that led to the best matching of the curves without overfitting. The transform was chosen because it provides the most information about the relationship between the two curves, as it accounts for changes in four dimensions: linear shifts in x and y, and scalar transforms in x and y. For these two patterns, we determined a relationship of $P_{0.6mm}(x) = 0.405 * P_{0.8mm}(0.999 * x - 0.001) + 0.113$ with a final mean squared error of $9.18 * 10^{-4}$ (Figure 4.2). This relationship is simpler than expected. The primary effect is

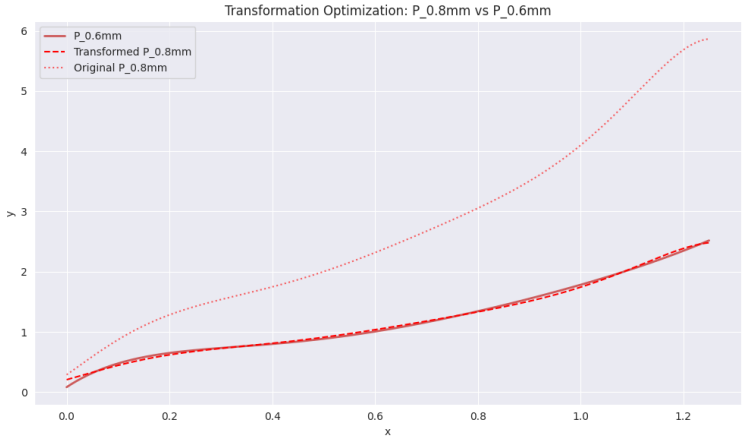

Figure 4.3 This graph shows the transformation optimization of the 0.8 mm (2.0, 2.4) spring pattern into the 0.6 mm version.

a reduction in necessary force by 60%, with an additional 0.113 vertical shift. The a and b parameters here have almost no effect on the transformation. In a sense, this means that this transform could be approximated by a simple scalar of 0.4, as can be seen in Fig. 4.3. This makes sense, as intuitively, an increase in ribbon width leads to more material that must be deformed in order to stretch, and therefore greater stiffness.

Next, we repeated this optimization process with the (2.0, 3.2) and (2.0, 4.0) spring patterns and found the following relationships:

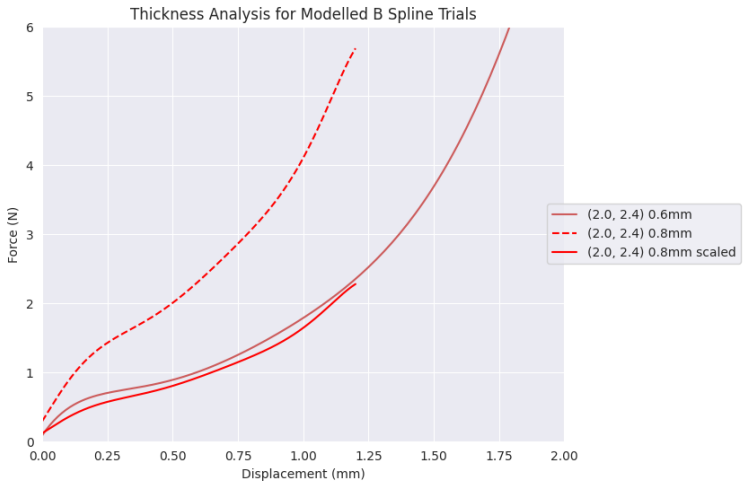

Figure 4.4 This graph shows a simplified transformation between the 0.6 and 0.8 mm (2.0, 2.4) springs using a 0.4x scalar.

$$P_{0.6mm,(2.0,3.2)}(x) = 0.502 * P_{0.8mm,(2.0,3.2)}(0.882 * x + 0.039) - 0.022$$

$$\text{Mean Squared Error} = 6.26 * 10^{-5}$$

$$P_{0.6mm,(2.0,4.0)}(x) = 0.528 * P_{0.8mm,(2.0,4.0)}(0.608 * x + 1.635) - 0.207$$

$$\text{Mean Squared Error} = 2.097 * 10^{-4}$$

Interestingly, these relationships seem to become less simple as the y-coordinate increases, with a, b, and d gaining more prominent effects. However, if we compare the mean squared errors for the relationship between the actual 0.6 mm springs and ones estimated with a transformation of the form $P_A = c * P_B$, [Eqn. 4.1.2] where c is the same as in the equations above, we get the these results:

$$\text{Error}_{(2.0,2.4)} = 0.0099,\ \text{Error}_{(2.0,3.2)} = 0.0877,\ \text{Error}_{(2.0,4.0)} = 0.0144$$

Unfortunately, these relationships do us little good in generalizing the relationships between ribbon widths, so we experimented with approximating all the 0.6 mm patterns as scaled by the average of our 3 c values, 0.478, compared to the 0.8 mm spring patterns. This gave us the following errors:

$$\text{Error}_{(2.0,2.4)} = 0.0211,\ \text{Error}_{(2.0,3.2)} = 0.0569,\ \text{Error}_{(2.0,4.0)} = 0.0105$$

Interestingly, this actually reduces the error for the (2.0, 3.2) and (2.0, 4.0) spring patterns. In later sections, we will use this averaged scalar for comparisons. It should be noted that

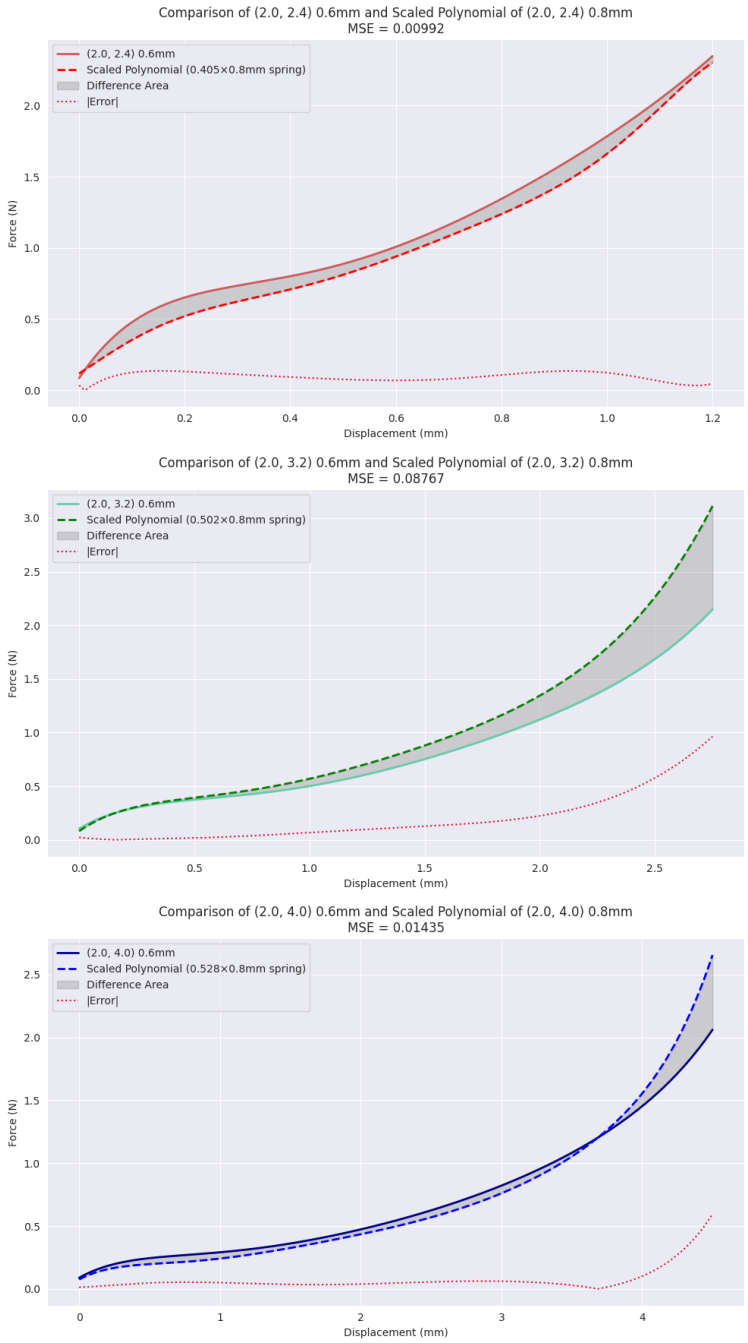

Figure 4.5 These graphs show the mean squared error between 0.6 mm modelled springs and estimations of the spring based on a scaled version of the 0.8 mm spring pattern using the relationship $P_A = c * P_B$. c values are taken from Equation 4.1.1.

this relationship was only established firmly for (2.0, y) patterns, but due to the smallness of changes when varying x, we will assume this relationship holds true for all patterns.

The ribbon width of the spring arms, for both modelled and experimental patterns,

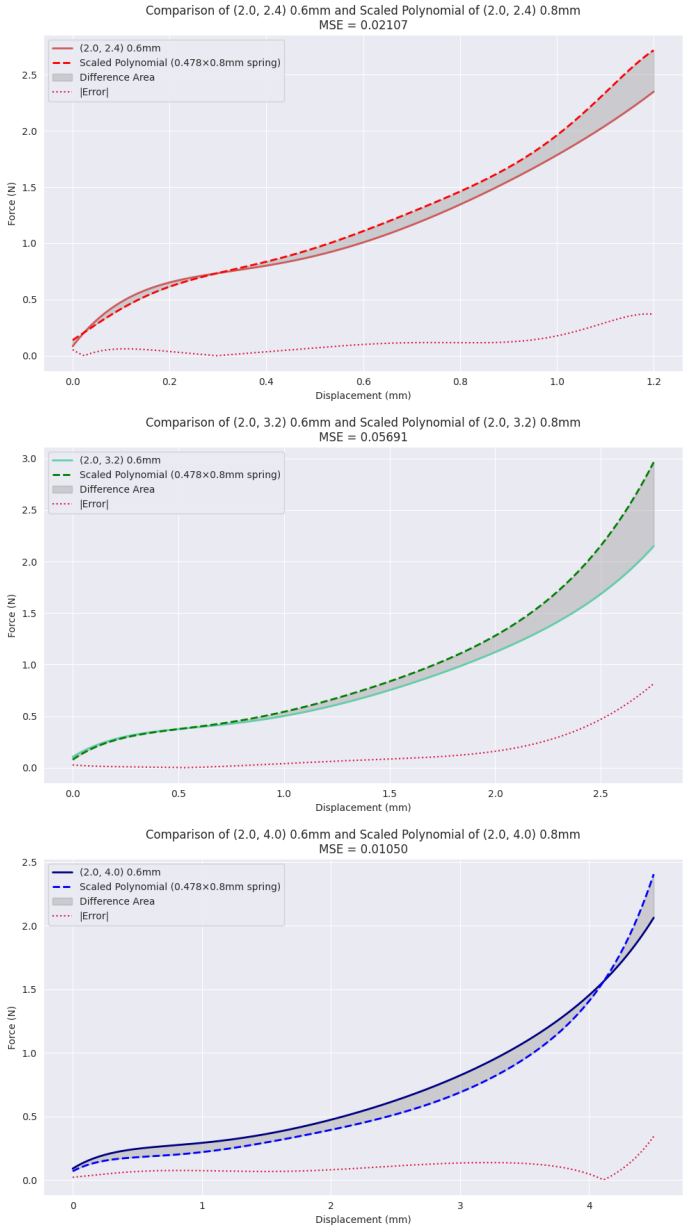

Figure 4.6 These graphs show the mean squared error between 0.6 mm modelled springs and estimations of the spring based on a scaled version of the 0.8 mm spring pattern using the relationship $P_A = c * P_B$. c values are averaged to get 0.478.

increase the stiffness of the springs. This makes sense in all three phases of the spring curve. In the pre-buckle phase, there is more material that needs to be moved and strained before the spring will buckle, leading to a higher force but similar displacements for the buckling point. This is seen in both the model and experimental data. In the working region, the average slope is steeper, indicating a stiffer spring, which again intuitively

makes sense, as there is more material resisting the stretching. In the third phase, all ribbon widths appear to asymptote towards the same slope, which makes sense because in that region the stretching is primarily dependent on material characteristics, which is the same across all patterns. Overall, a decrease in ribbon width from 0.8 mm to 0.6 mm leads to 0.478x the force required for stretching, which is in agreement with our intuitive hypothesis that wider springs require more force to stretch.

4.1.2 Spline Variations

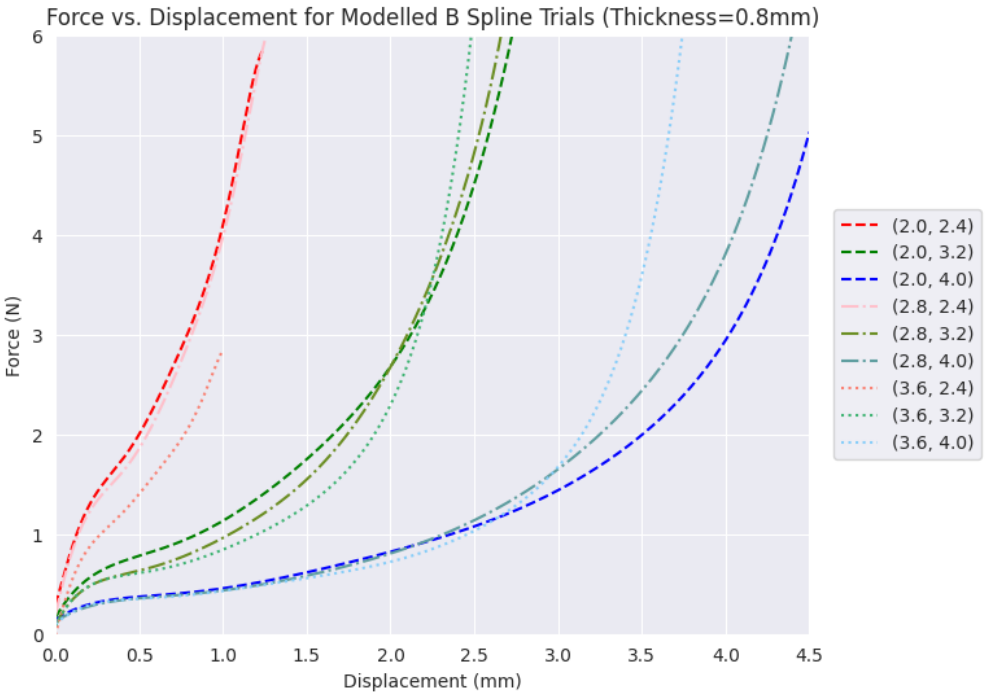

Figure 4.7 Modelled Force vs displacement graphs for 0.8 mm permutations of three variations each of the x and y coordinates of the B-spline control point.

For analyses of the x- and y- coordinates, we will focus primarily on what we will call the "working region". This refers to the second section of the spring curve, where the slope is shallower. This region is of the most interest, because it is the region in which the springs will be used. This region is defined based on the area from the point where the second derivative equals zero to the point with the largest value of x where the second derivative

equals 2.5. These parameters were developed in an attempt to mathematically define the region of the curves that have a positive curvature, but also still have a more shallow slope, indicating that the stretch is still occurring within the kirigami pattern, rather than the material itself. It is worth noting that the second parameter is slightly more arbitrary, and could be varied depending on the exact characteristics desired. This is important to reduce the material degradation that can occur through repeated stretching and compression.

X-Coordinate

Changing the x-coordinate of the splines shifts the peak of the curve horizontally, as can be seen in the bottom of Fig. 4.7. It can be easier to visualize this as a shift in the angle of the center-most line in the spring pattern, which has a negative slope for smaller x values, is vertical for medium values, and a positive slope for large values. As can be seen in

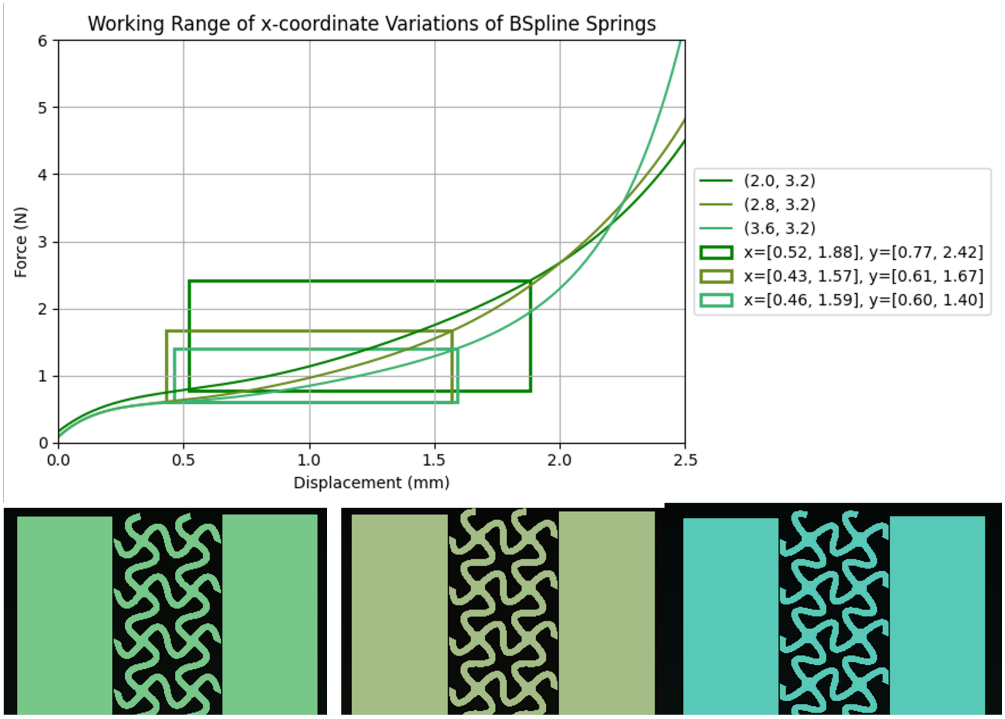

Figure 4.8 Variations in the "x" parameter of the B splines leads to slight changes in the curve shape. The boxes show the "working range" of the springs, or the region in which the springs behave most linearly.

Figures 4.6 and 4.7, varying the x-coordinate has only a small effect on the force required

to stretch the spring. The primary effect is a flattening of the curve in the working region, and a subsequent steepening in the third, post-buckling, phase. This has two ramifications for our project. First, a small reduction in the working region, in both force and displacement, which informs the selection of spring pattern for a given application. The second comes the flattening of the curve, in that, if emphasized, this has the potential to create a spring that could exert an almost constant force across a displacement range. This can be seen most clearly in the (3.6, 4.0) spring pattern which has a shallow slope from 0.5 to 1.75 mm. It is postulated, based on the data from Section 4.1.1 that this effect would be further pronounced in thinner springs.

Y-Coordinate

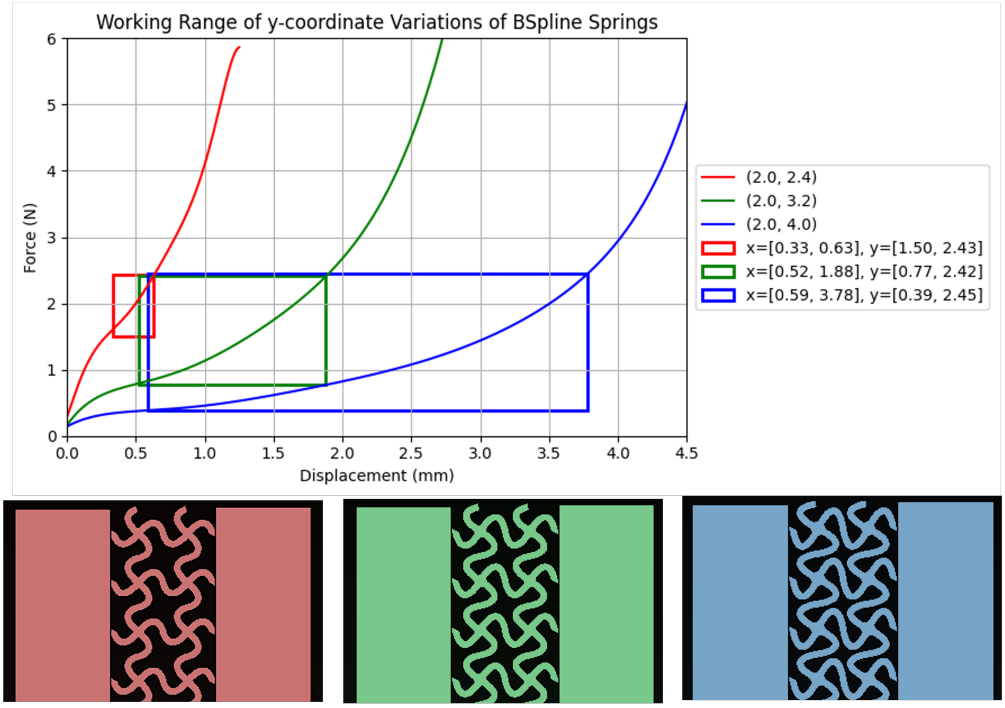

Figure 4.9 Variations in the "Y" parameter of the B splines leads to significant changes in the curve shape, the force required, and the distance the springs can be stretched. The boxes show the "working range" of the springs, or the region in which the springs behave most linearly.

Variations in the y-coordinate have a much more pronounced effect than variations in the

x-coordinate. We can see from the boxes in Figure 4.9 that having a smaller y-coordinate maintains a similar maximum force, but decreases the working range significantly in displacement. In other words, the spring requires more force to stretch, but can stretch less far. This means that while the spring can handle larger applied forces without breaking, the forces must be exerted over smaller distances relative to a spring with a larger y-coordinate. Increasing the y-coordinate at first glance seems to have a similar, if more

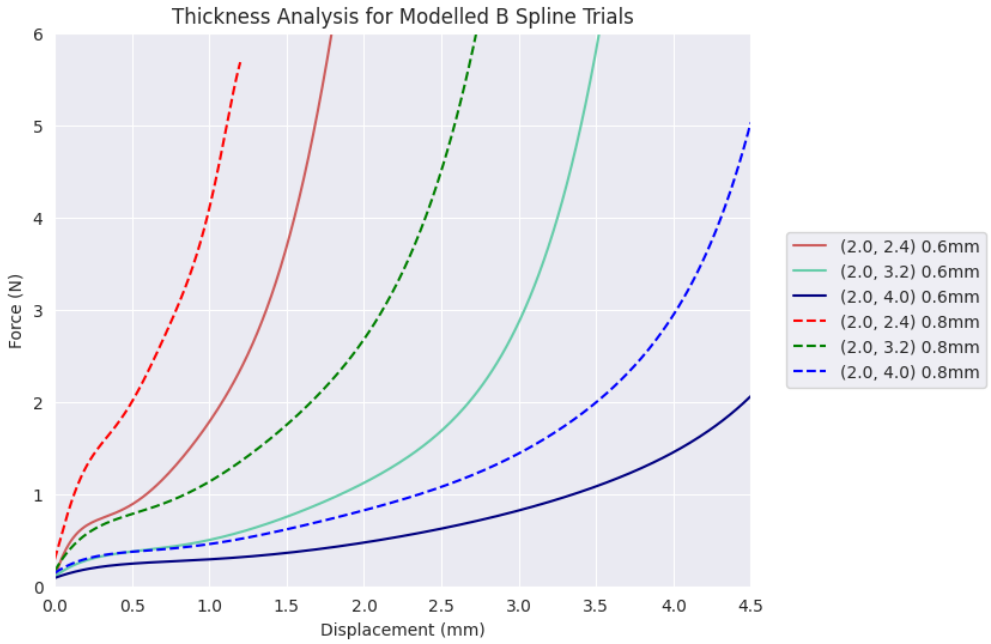

Figure 4.10 Graph showing the difference in effects between changing the ribbon width and varying the y-coordinate. It can be seen that, while the effects appear quite similar, varying the y-coordinate has a much larger effect than varying the spring ribbon width.

pronounced effect as decreasing the ribbon width, so we began our investigation of variations in the y-coordinate with the same analysis performed in Section 4.1.1. From this we got the following relationships:

$$P_{(2.0,3.2)}(x) = 38.94 * P_{(2.0,2.4)}(0.006 * x + 0.695) - 103.0$$

$$\text{Mean Squared Error} = 0.0021$$

$$P_{(2.0,4.0)}(x) = 304.0 * P_{(2.0,3.2)}(0.0001 * x - 0.255) + 354.0$$

$$\text{Mean Squared Error} = 0.0020$$

$$P_{(2.0,4.0)}(x) = 14.02 * P_{(2.0,2.4)}(0.004 * x + 0.718) - 38.15$$

Mean Squared Error = 0.0005

These equations cannot be estimated to be dependent only on c, due to the smallness of the a values, The small a value serves to elongate the curve in the x direction. For example, if we have a simple polynomial, x^2, and we transform it into $(0.5 * x)^2$, then each y-value will appear at twice the x-value it originally appeared at. This can be seen in how the x^2 curve intersects with y=4 at x=2, whereas the $(0.5 * x)^2$ curve intersects with y=4 at x=4. Likewise, the curve $(0.1 * x)^2$ would intersect with y=4 at x=20. We can then extend this relationship to our equations here to see that our transformed curves would intersect with a given y-value at an x-value of 400-1000 times larger that the original value. There is also not a clear relationship between the transformations between the spring patterns. For these reasons, we attempted to do a different fit, of the form $P_A = c * P_B$ [3] to test whether such a fit has small enough error to be useful. As can be seen in Figure 4.11, the errors in the simple fit are $< 3\%$, which means that this approximation is reasonably accurate.

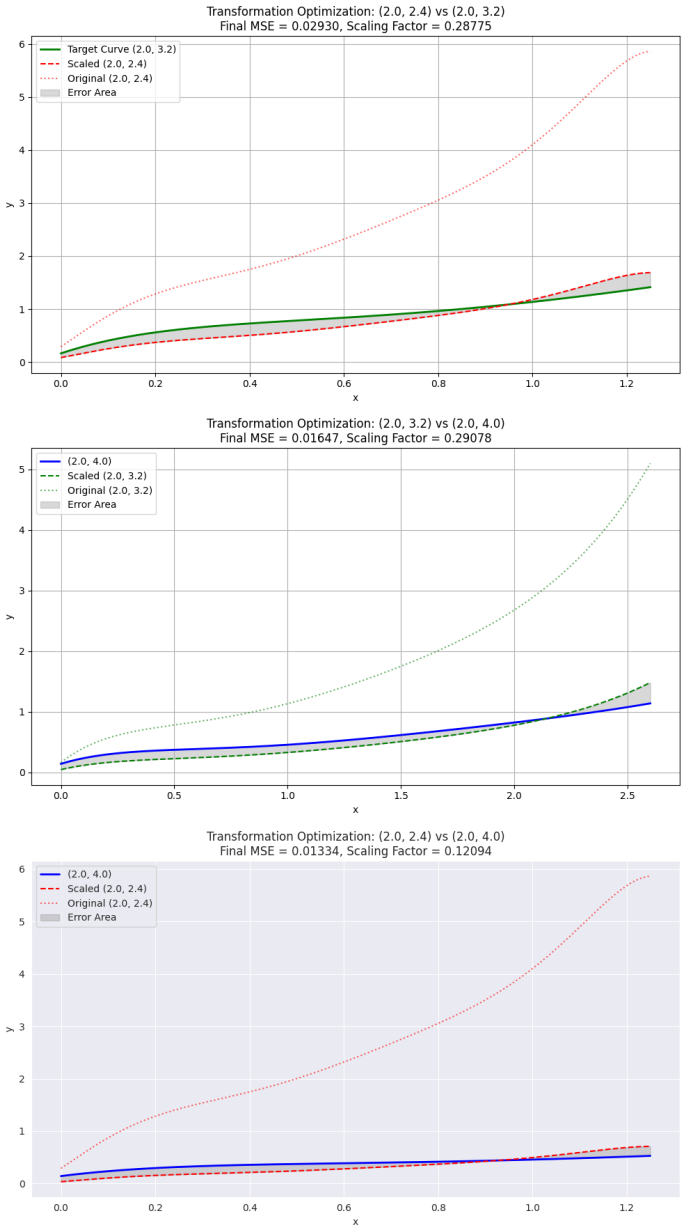

Figure 4.11 These graphs show the transformations between the three curves shown in Figure 4.9, using the fit in Equation 4.1.2.

4.2 Experimental Results

All experimental results in the following section are from knife cut springs. It is also important to note that the nominally 0.6 mm springs are in reality on average 0.85 mm, and the nominally 0.8 mm springs are actually 1.0 mm in width. This was discussed previously in depth in Section 3.3.2. For the remainder of this section, the springs will be referred to by their nominal width. While this does limit our ability to compare directly between the model and experimental data, we will make use of the previously defined ribbon width relationship to transform the springs for comparison. In order to do these transformations, we have fit 7th degree polynomials to each of the experimental curves. The original curves can be seen in Appendix A.

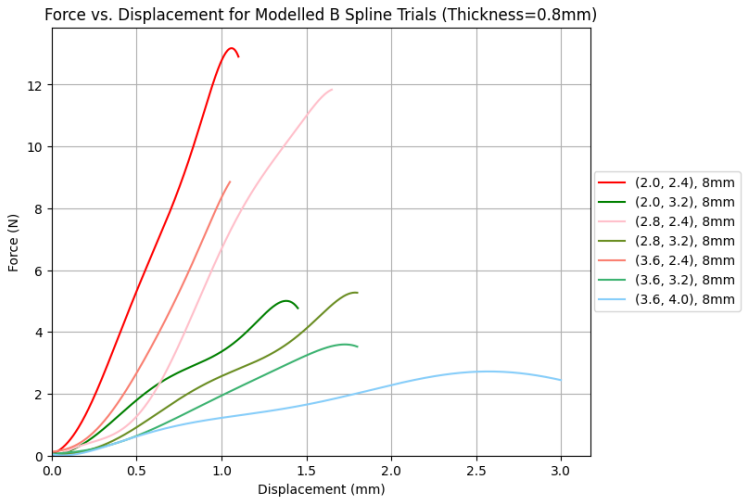

Figure 4.12 Experimental Force vs displacement graphs for 0.8 mm permutations of three variations each of the x and y coordinates of the b-spline control point.

An initial qualitative analysis of the experimental data reveals some interesting trends. The 0.6 mm springs appear to be mostly in agreement with the model, with the correct three-phase shape and clustering of patterns with the same y-coordinate. However, there are some discrepancies worth noting - primarily that the (2.0, y) patterns have a less defined shape than expected, and that the (2.8, 2.4) and (3.6, 2.4) patterns are flipped from where the model would expect them to be. We predict that the (2.0, y) patterns seem

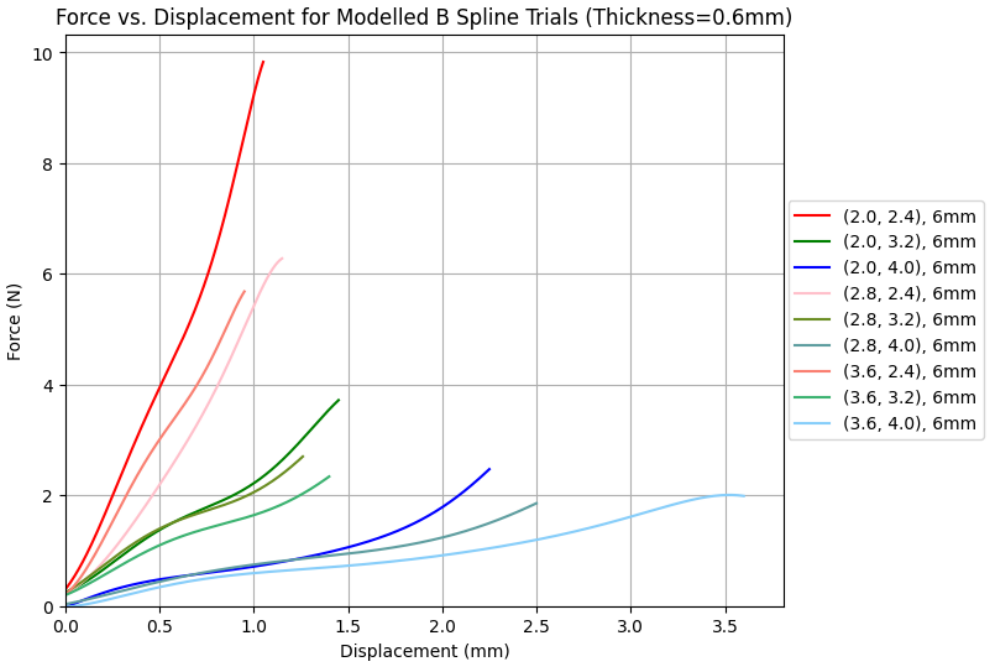

Figure 4.13 Experimental Force vs displacement graphs for 0.6 mm permutations of three variations each of the x and y coordinates of the b-spline control point.

to be more linear because the splines are getting straight enough that they are approaching a square lattice, which does not buckle, and would stretch linearly at the same rate as Kapton®. The second discrepancy, that the (2.8, 2.4) and (3.6, 2.4) patterns are flipped, we would expect is an artifact of data collection, that the sample was put into the Instron with more slack, and therefore the beginning of the data is the stretching occurring without tension. However, this flip also occurs in the 0.8 mm patterns, and is not satisfactorily resolved by a lateral shift of the entire curve, as the slope is also approximately the same between the two patterns. This feature requires further experimentation to explain. The 0.8 mm patterns are less in agreement with the modelled data, and show less distinct three-phase behavior. We believe this is because the width has been increased to a point where it begins to interfere with smooth buckling. This can be seen in how the buckling point is shifted significantly up and to the right, indicating that the sample requires both more force and more displacement in order to buckle. While this behavior could be useful in some scenarios, for the use case of the Space Solar Power

Project, this behavior is contrary to our goals. For this reason, we will mostly focus on the 0.6 mm spring patterns. The scaling in Figure 4.14 indicates that the scaling factor for

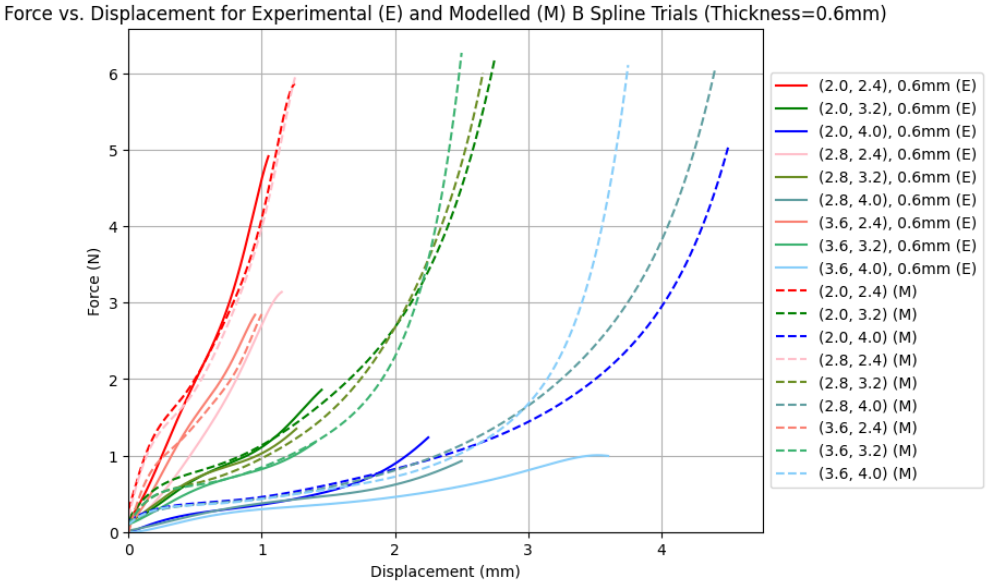

Figure 4.14 Force vs displacement graphs for 0.8 mm modelled and 0.6 mm experimental permutations of three variations each of the x and y coordinates of the b-spline control point. In this plot, the experimental curves are scaled down by 0.5x.

ribbon width is not perfectly linear based on difference in width, as there is a 0.478x difference between 0.8 mm and 0.6 mm springs, and another approximately 0.5x difference between ∼0.9 mm and 0.8 mm springs.

It is complicated to determine how in agreement the model and the experimental data is due to the high uncertainty created by a combination of generally overly wide springs and inconsistent spring widths. However, the analysis of the change in behavior of the springs with changing ribbon width indicates that the discrepancies between the modelled and experimental data are likely occurring from the larger ribbon width rather than an alternative complication. In support of this conclusion is the similar 3-phase curve shape between the models and the fact that a ribbon width transformation of the type described in Section 4.1.1, with a 0.478 scalar brings the 0.6 mm experimental and 0.8 mm modelled patterns into reasonable agreement. There is some disagreement remaining, particularly in the (x, 2.4) spring patterns, which lose some of the definition between the second and third

phases of the curves, likely due to the spring segments becoming stiffer, and therefor more similar to the Kapton® itself.

In some ways, this discrepancy is useful, because it informs us that there is a limit to the stiffness of the springs, at least while they remain at the 1 cm size. Intuitively, this makes sense as well, because splines with a small y-coordinate will approach straight lines, which cannot by nature buckle, and therefore will not display the spring-like behavior we are studying. There are also divergences between the buckling behavior between the model and experimental data. The experimental data has the buckle occurring significantly later than the model, which indicates some physical factor that resists buckling. This delayed buckle occurred only in knife cut spring patterns, which could arise from a number of possibilities, the two most prominent being that the ragged edges from the laser cutter create imperfections that incentivize buckling, or that the slight roll that appears in the knife cut patterns resists buckling. The rolling in the knife cut patterns appears in microscope images to be less pronounced than the dog bone effect in the laser cut patterns, which would indicate that the rolling is not the primary cause, however it is possible that increased fragility in the laser cut springs counteracts the added stability of the thickened edges.

Chapter 5

Library

> I am driven by two main philosophies: know more today about the world than I knew yesterday and lessen the suffering of others.
>
> Neil DeGrasse Tyson

The library can be used to select springs for given use cases. For example, let's say that we have a theoretical solar tile system in which the frame is made of a very special carbon fiber that has a thermal expansion coefficient (CTE) of zero, and the solar panel is made up entirely of Kapton®, which has a CTE of 20 ppm/K (DuPont, 2022). This CTE means that if you had a million meters of Kapton®, and you raised it by one degree Kelvin, it would become 1 million and 20 meters long. However, our tile isn't one million meters long, it's 10 cm[1], and our temperature range isn't one degree, it's about 400 K to 115 K. This is the temperature range experienced by the International Space Station, and is a reasonable approximation for what the Space Solar Project will experience (ESA, 2021). In this case, across the entire temperature range, the solar panel will expand linearly by 0.57 mm. Because there are springs on each side, each spring will have to absorb half that distance, or 0.285 mm. Let us further assume that our solar panel cannot handle more

[1]It is worth noting here that our experimental and modelled springs are only 6 cm, so a scaling factor should be implemented depending on the final size of the tiles. Preliminary research indicates that the force would increase linearly for longer springs.

Pattern (0.6 mm)	Force (N)	Force Range (N)	Displacement (mm)	Disp Range (mm)
(2.0, 2.4)	[0.75, 1.30]	0.55	[0.33, 0.78]	0.45
(2.0, 3.2)	[0.39, 1.76]	1.37	[0.57,2.55]	1.98
(2.0, 4.0)	[0.27, 2.10]	1.83	[0.77, 4.50]	3.73
Pattern (0.8 mm)	**Force (N)**	**Force Range (N)**	**Displacement (mm)**	**Disp Range (mm)**
(2.0, 2.4)	[1.50, 2.43]	0.93	[0.33, 0.63]	0.30
(2.0, 3.2)	[0.77, 2.42]	1.63	[0.52, 1.88]	1.36
(2.0, 4.0)	[0.39, 2.45]	2.06	[0.59, 3.78]	3.19
(2.8, 2.4)	[1.48, 2.28]	0.80	[0.33, 0.63]	0.30
(2.8, 3.2)	[0.61, 1.67]	1.06	[0.43, 1.57]	1.14
(2.8, 4.0)	[0.37, 2.60]	2.23	[0.62, 3.58]	2.96
(3.6, 2.4)	[1.07, 2.00]	0.93	[0.31, 0.80]	0.49
(3.6, 3.2)	[0.60, 1.40]	0.80	[0.46, 1.59]	1.13
(3.6, 4.0)	[0.36, 1.40]	1.05	[0.52, 2.83]	2.31

Figure 5.1 This table shows the locations and dimensions of the working region for each spring pattern, based on modelled data from Abaqus. It is easy to see in this table the following trends: force decreases and displacement increases as ribbon width increases, both the displacement and force ranges increase as the y-coordinate increases, and there is a slight decrease to both force and displacement as the x-coordinate increases.

than 2 N of force without breakage. This force is somewhat arbitrary, and will depend on the exact design and materials of the solar panels. If we consult our table, we can see that all of our springs can nominally handle a displacement range of 0.285 mm. However, the (x, 2.4) patterns have a displacement range that is only slightly greater than what we need, so to account for error or unforeseen circumstances, we should not choose those patterns. Next, we need to narrow the remaining patterns to those whose working ranges cap out below 2 N. This gives us the (2.0, 3.2) 0.6 mm pattern, and the (2.8, 3.2), (3.6, 3.2), and (3.6, 4.0) 0.8 mm patterns. Out of these patterns, we can select the (3.6, 3.2) 0.8 mm spring as the optimal pattern, because it has both the smallest maximum force, which reduces the risk of accidentally applying too much force, and the smallest displacement range, which limits unnecessary excess.

Chapter 6

Conclusion

> Every great advance in science has issued from a new audacity of imagination.
>
> John Dewey

Through the course of this project, we succeeded in our main goal – to create a library of spring patterns that could be used for a variety of use cases. We also established a set of relationships describing what happens when varying different parameters such as cutting method, ribbon width, and x- and y-coordinate positions.

In terms of cutting methods, we investigated using laser cutters and commercial plotting cutters (knife cutting). The laser cut patterns have losses due to burning of up to 200 µm of material. This loss can only be partially limited by purposefully cutting wider patterns, as the ribbon width is limited in order to avoid overlap of the spring arms. This method also introduces non-homogeneity into the Kapton® through melting and soot deposits, and creates a dog bone effect in which the edges of the Kapton® become thicker by about 10 µm. However, the laser cut patterns buckle very smoothly. The knife cut patterns were about 200-250 µm wider than intended, but this was due to software issues rather than an innate issue of the methodology and so could be fixed computationally. The knife cut patterns are much more homogenous, but do not buckle as smoothly as the laser cut

patterns. Further research in this area would involve testing laser cutters with different wavelengths, powers, and speeds to see if smaller losses can be achieved. It would also be worthwhile, especially if investigating a single pattern at length, to create a die to cut the pattern with efficiency and consistency.

For ribbon width, we confirmed the intuitive idea that wider springs require more force to stretch. We also discovered that an increase in spring arm ribbon width reduces the working region displacement range of the springs. Through numerical transforms, we established that 0.6 mm springs require 0.478 times the force as 0.8 mm springs, and 0.8 mm springs require approximately half the force as 0.9 mm springs. We further established that springs wider than 0.9 mm start to behave more like a solid material, and do not buckle smoothly, or have the three well-defined phases that we are looking for. Further research should be done on more widths of springs to develop a more clearly defined relationship between changes in ribbon width and required force.

Increasing the x-coordinate of the splines leads to a reduction in both the force and displacement ranges of the working region. It also creates a flatter working region, which has the potential to, if emphasized, create a spring that exerts an almost constant force regardless of displacement. This should be investigated further. Increasing the y-coordinate of the splines leads to an increase in the force and displacement ranges of the working region. This is a much more significant effect than those that arise from variations in the x-coordinate, especially in terms of the displacement range. In large part, this is due to the increase in the y-coordinate creating spring arms that are physically longer and so can stretch further. We were unable to define conclusive mathematical relationships for the differences in the curves upon varying the x- and y-coordinates. Further research would be benefited by investigating the mathematical relationships further, as well as by testing other pattern variations to confirm the qualitative relationships.

While this project has had many successes, and much has been learned about the behavior of these B spline springs, there is also much future work to be done. The spring patterns

need to be tested for durability to see how many times they can stretch within the working region before degrading to establish an estimated lifetime for the springs. This research could also help to develop a more well-defined upper boundary for the working region. Additionally, the behavior of the springs under shearing loads should be modelled to ensure that the springs are not exerting any torque on the solar panels. Finally, the spring behavior should be tested at a variety of temperatures to ensure that they function properly through the temperature changes they would experience in space.

Appendix A

Raw Data and Buckling Video

This appendix contains the raw experimental data plotted as force vs. displacement. It also contains an image series showing how the B-spline spring patterns are built from a single curve. There is also an attached video showing a detailed model of how the Double-S spring pattern buckles during stretching. Experimental Data Plots Image Series showing

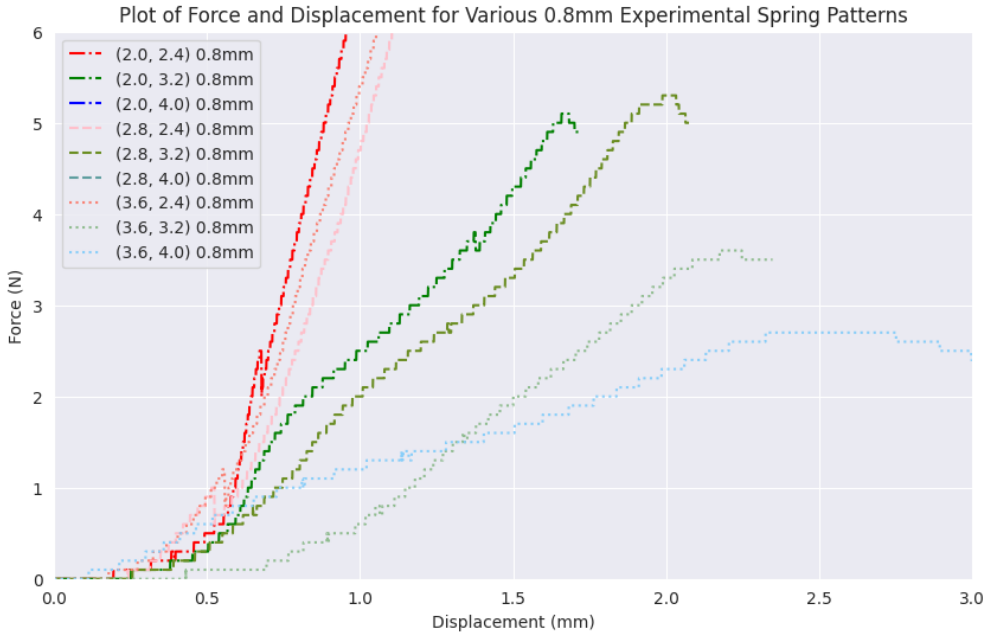

Figure A.1 Experimental Force vs displacement graphs for 0.8 mm permutations of three variations each of the x and y coordinates of the b-spline control point.

Figure A.2 Experimental Force vs displacement graphs for 0.6 mm permutations of three variations each of the x and y coordinates of the b-spline control point.

how the patterns are built from individual splines.

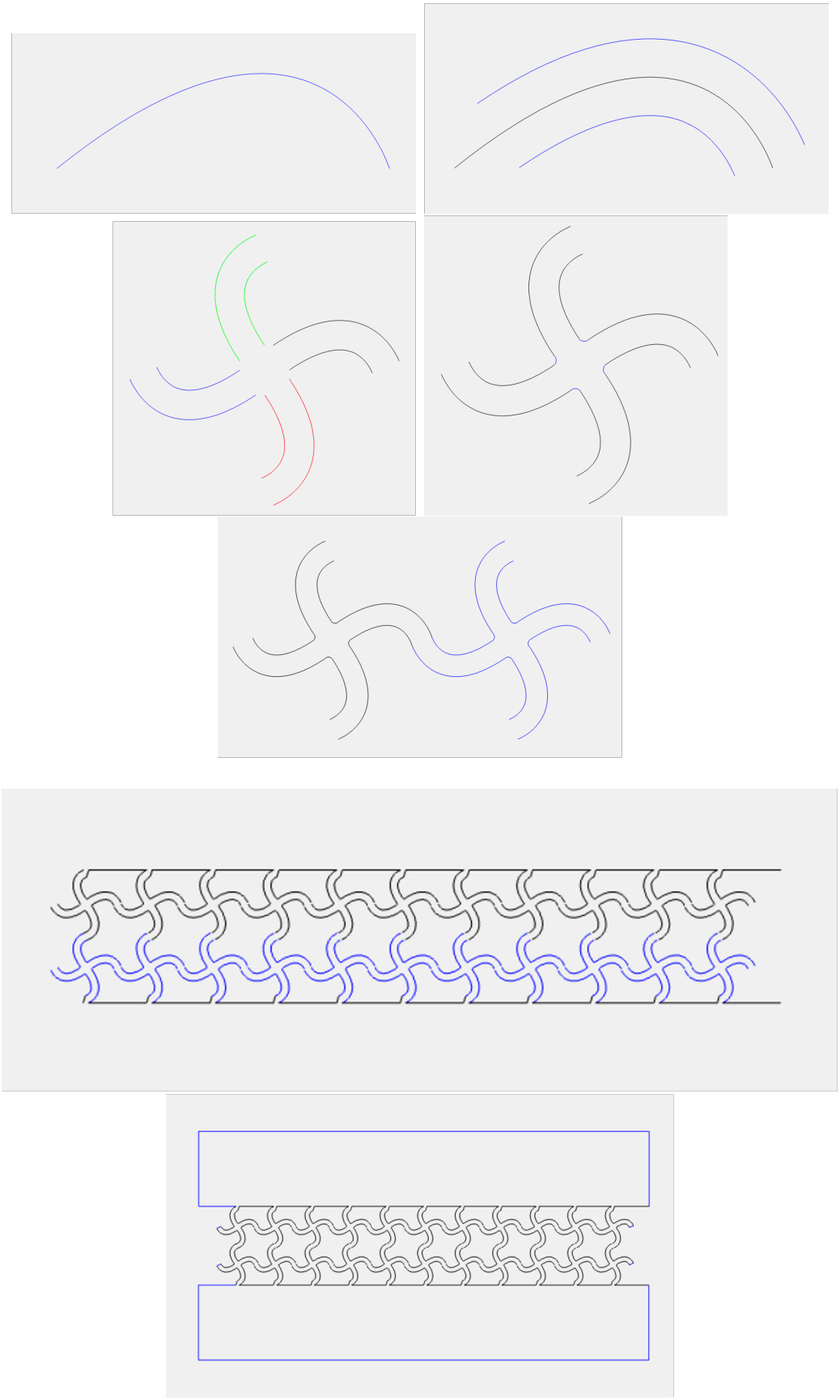

Figure A.3 A visual step-by-step of the pattern development outlined in 3.1

Link to Video of Abaqus "Load" Step: Abaqus Video

https://drive.google.com/file/d/1G5D8gPKNf9WzTnj3c3ifFGHYvLprCMiB/view?usp=sharing

Appendix B

Code and Files

This appendix contains the Python code for the Abaqus modelling and the Matlab code for the spline development under the "Code" subheading, and the code for data manipulation, plot-making, and polynomial fits are in Google Colaboratory files under the "Files" subheading. All the original data is also contained in a Google Sheet under the "Files" subheading.

B.1 Files

You will generate files for your thesis. It will behoove you to name and document them in a logical and consistent way as you go, as here you will provide a map of where everything is. For example:
The code to generate all of the figures used in this thesis are in the Google Colab notebooks linked here: ThesisWorkPt1, ThesisWorkPt2, Polynomial Fits
The raw data files are stored in the Google Sheet here: Original Data

B.2 Code

Code for Abaqus:

```
from abaqus import *
from abaqusConstants import *
import __main__
import section
import regionToolset
import displayGroupMdbToolset as dgm
import part
import material
import assembly
import step
import interaction
import load
import mesh
import optimization
import job
import sketch
import visualization
import xyPlot
import displayGroupOdbToolset as dgo
import connectorBehavior
#'2404','21604','362404','22404','23204','28404','36404','281604',
#'282404','283204','361604','363204'
names = ['2403','22403','23203']
#Determines which models the code will run on (theoretically,
#sometimes it just runs on whichever model is currently active)
for name in names:
    #Define Kapton
    mdb.models[name].Material(name='Kapton')
    mdb.models[name].materials['Kapton'].Density(table=((1.42e-03, ), ))
    mdb.models[name].materials['Kapton'].Elastic(table=((398000.0, 0.34),
   ))

    #Define Section
```

```
    mdb.models[name].HomogeneousShellSection(name='Section-1',
        preIntegrate=OFF, material='Kapton', thicknessType=UNIFORM,
        thickness=0.05, thicknessField='', nodalThicknessField='',
        idealization=NO_IDEALIZATION, poissonDefinition=DEFAULT,
        thicknessModulus=None, temperature=GRADIENT, useDensity=OFF,
        integrationRule=SIMPSON, numIntPts=5)
    #Section Assignment
    p = mdb.models[name].parts['Part-1']
    f = p.faces
    faces = f.getSequenceFromMask(mask=('[#1 ]', ), )
    region = p.Set(faces=faces, name='Set-1')
    p = mdb.models[name].parts['Part-1']
    p.SectionAssignment(region=region, sectionName='Section-1', offset
   =0.0,
        offsetType=MIDDLE_SURFACE, offsetField='',
        thicknessAssignment=FROM_SECTION)

    #MESH PART HERE (I do this manually because I couldn't successfully
   partition using python)
    #a. Global seed = 0.8, curvature param = 0.01
    #b. Create local seed for middle regions with seed = 0.5 and curvature
    = 0.007
    #c. Partition pull tabs and assign them as structured quad


    #Create steps
    mdb.models[name].BuckleStep(name='Buckle', previous='Initial',
        numEigen=10, eigensolver=LANCZOS, minEigen=0.0, blockSize=DEFAULT,
        maxBlocks=DEFAULT)

    mdb.models[name].StaticStep(name='Load', previous='Buckle',
        maxNumInc=10000000, initialInc=0.01, minInc=1e-20, maxInc=0.1,
        nlgeom=ON)
```

```
63
64      mdb.models[name].StaticStep(name='Load', previous='Buckle',
65          maxNumInc=10000000, initialInc=0.01, minInc=1e-20, maxInc=0.1,
66          nlgeom=ON)
67      a = mdb.models[name].rootAssembly
68
69      #Create BCs and Load (these may or may not fail sorry)
70      a = mdb.models[name].rootAssembly
71      e1 = a.instances['Part-1-1'].edges
72      edges1 = e1.getSequenceFromMask(mask=('[#2000000 ]', ), )
73      region = a.Set(edges=edges1, name='Set-1')
74      mdb.models[name].EncastreBC(name='BC-1', createStepName='Initial',
75          region=region, localCsys=None)
76
77      a = mdb.models[name].rootAssembly
78      s1 = a.instances['Part-1-1'].edges
79      side1Edges1 = s1.getSequenceFromMask(mask=('[#0:400 #400 ]', ), )
80      region = a.Surface(side1Edges=side1Edges1, name='Surf-1')
81      mdb.models[name].ShellEdgeLoad(name='Load-1', createStepName='Buckle',
82          region=region, magnitude=-0.01, distributionType=UNIFORM, field=''
    ,
83          localCsys=None)
84
85      a = mdb.models[name].rootAssembly
86      e1 = a.instances['Part-1-1'].edges
87      edges1 = e1.getSequenceFromMask(mask=('[#0:400 #400 ]', ), )
88      region = a.Set(edges=edges1, name='Set-2')
89      mdb.models[name].DisplacementBC(name='BC-2', createStepName='Load',
90          region=region, u1=-3.0, u2=0.0, u3=0.0, ur1=UNSET, ur2=UNSET,
91          ur3=UNSET, amplitude=UNSET, fixed=OFF, distributionType=UNIFORM,
92          fieldName='', localCsys=None)
93
```

```
94    #Create Imperfection (note: the filename here is specific to my
      nomenclature)
95    mdb.models[name].rootAssembly.engineeringFeatures.FileImperfection(
96        name='Imperfection-1', file='E:\\Rachel\\abaqus\\'+name+'buckle.
      odb',
97        step=1, increment=1, linearSuperpositions=((1, 0.01), (2, 0.01),
      (3,
98        0.01), (4, 0.01), (5, 0.01), (6, 0.01), (7, 0.01), (8, 0.01), (9,
99        0.01), (10, 0.01)))
100
101   #Create Jobs
102   mdb.Job(name=name+'Buckle', model=name, description='', type=ANALYSIS,
103       atTime=None, waitMinutes=0, waitHours=0, queue=None, memory=90,
104       memoryUnits=PERCENTAGE, getMemoryFromAnalysis=True,
105       explicitPrecision=SINGLE, nodalOutputPrecision=SINGLE, echoPrint=
      OFF,
106       modelPrint=OFF, contactPrint=OFF, historyPrint=OFF, userSubroutine
      ='',
107       scratch='', resultsFormat=ODB, numThreadsPerMpiProcess=1,
108       multiprocessingMode=DEFAULT, numCpus=1, numGPUs=1)
109   mdb.Job(name=name, model=name, description='', type=ANALYSIS,
110       atTime=None, waitMinutes=0, waitHours=0, queue=None, memory=90,
111       memoryUnits=PERCENTAGE, getMemoryFromAnalysis=True,
112       explicitPrecision=SINGLE, nodalOutputPrecision=SINGLE, echoPrint=
      OFF,
113       modelPrint=OFF, contactPrint=OFF, historyPrint=OFF, userSubroutine
      ='',
114       scratch='', resultsFormat=ODB, numThreadsPerMpiProcess=1,
115       multiprocessingMode=DEFAULT, numCpus=1, numGPUs=1)
116
117   #Suppress Load step and imperfection file
118   mdb.models[name].steps['Load'].suppress()
```

```
119     mdb.models[name].rootAssembly.engineeringFeatures.imperfections['
    Imperfection-1'].suppress()
120
121     #Submit buckle job
122     mdb.jobs[name+'Buckle'].submit(consistencyChecking=OFF)
123
124     #Resume load and imperfection and suppress buckle
125     mdb.models[name].steps['Load'].resume()
126     mdb.models[name].steps['Buckle'].suppress()
127     mdb.models[name].rootAssembly.engineeringFeatures.imperfections['
    Imperfection-1'].resume()
128
129     #Submit model job
130     mdb.jobs[name].submit(consistencyChecking=OFF)
```

Code for Matlab Pattern Building:

```
xo =2;
yo=1.6;
os = 0.3; %ribbon thickness scalar (final ribbon width is twice this
    scalar)
for xo = 2:0.8:4 %Selects x positions for BSpline control point
    for yo = 2.4:0.8:4 %Selects y positions for BSpline control point
        filename = [num2str(xo), '.', num2str(yo),num2str(os),'.dxf'];
        n = figure;
        set(0, 'DefaultLineLineWidth', 0.1);
        axis padded
        axis equal
        axis off

        %% Create first arm
        cpts = [0 xo 2.5 2.5; 0 yo 0 0];
        tpts = [0 5];

        tvec = 0:0.1:5;
        [q, qd, qdd, pp] = bsplinepolytraj(cpts,tpts,tvec);

        z = zeros(size(q));
        X = q(1,:);
        Y = q(2,:);
        Z = z(1,:);
        %% Rotate original arm to create 3 more instances 90 degrees apart
        ang2 = 180;  % degrees
        X1 =  (X)*cosd(ang2) + (Y)*sind(ang2);
        Y1 = -(X)*sind(ang2) + (Y)*cosd(ang2);
        ang = 90;  % degrees
        X2 =  (X)*cosd(ang) + (Y)*sind(ang);
        Y2 = -(X)*sind(ang) + (Y)*cosd(ang);
```

```
31        ang1 = -90;  % degrees
32        X3 =  (X)*cosd(ang1) + (Y)*sind(ang1);
33        Y3 = -(X)*sind(ang1) + (Y)*cosd(ang1);
34        %% Create two offset curves from each arm to create ribbon
    thickness
35        [x, y] = offsetCurve(X(4:45), Y(4:45), os);
36        [x1, y1] = offsetCurve(X(4:45), Y(4:45), -os);
37        [x2, y2] = offsetCurve(X1(4:45), Y1(4:45), os);
38        [x3, y3] = offsetCurve(X1(4:45), Y1(4:45), -os);
39        [x4, y4] = offsetCurve(X2(4:45), Y2(4:45), os);
40        [x5, y5] = offsetCurve(X2(4:45), Y2(4:45), -os);
41        [x6, y6] = offsetCurve(X3(4:45), Y3(4:45), os);
42        [x7, y7] = offsetCurve(X3(4:45), Y3(4:45), -os);
43
44        %% Create fillets
45        cpts2 = [x1(1,1) x1(1,1)-0.05 x4(1,1)-0.05 x4(1,1); y1(1,1) y1
    (1,1)-0.05 y4(1,1)+0.05 y4(1,1)];
46        tpts2 = [0 5];
47
48        tvec2 = 0:0.1:5;
49        [q2, qd2, qdd2, pp2] = bsplinepolytraj(cpts2,tpts2,tvec2);
50        a = q2(1,:);
51        b = q2(2,:);
52        a1 =  (a)*cosd(ang2) + (b)*sind(ang2);
53        b1 = -(a)*sind(ang2) + (b)*cosd(ang2);
54        a2 =  (a)*cosd(ang) + (b)*sind(ang);
55        b2 = -(a)*sind(ang) + (b)*cosd(ang);
56        a3 =  (a)*cosd(ang1) + (b)*sind(ang1);
57        b3 = -(a)*sind(ang1) + (b)*cosd(ang1);
58
59        %% Draw the springs
60        hold on
61        os2 =0; %Vertical spring instance offset
```

```
62        for os2 = 0:5:50
63            %Draw center hole
64            x_hole = [x flip(x3+5) a1+5 x6+5 flip(x5+5) a2+5 x2+5 flip(x1)
 a x4 flip(x7) a3];
65            y_hole = [y, flip(y3) b1 y6 flip(y5+5) b2+5 y2+5 flip(y1+5) b
+5 y4+5 flip(y7) b3]+os2;
66            %Draw edge holes
67            x_rs = [flip(x+5) flip(a3+5) x7+5 flip(x4+5) flip(a+5) x1+5 x(
end)+5 x(end)+5];
68            y_rs = [flip(y) flip(b3) y7 flip(y4+5) flip(b+5) y1+5 y1(end)
+4.6 y(end)]+os2;
69            x_ls = [flip(x3) a1 x6, flip(x5) a2 x2, x2(end) x3(end)];
70            y_ls = [flip(y3) b1 y6, flip(y5+5) b2+5 y2+5 y3(end)+0.4 y3(
end)]+os2;
71            %Draw edges and pull tabs
72                %There is sometimes an error that occurs here where a
small
73                %loop is drawn where the springs meet the pull tabs that
74                %will eventually need to be fixed.
75            x_edge = [flip(a2) x5 flip(x4) flip(a) x1 flip(x2+5) flip(a2
+5) x5+5 flip(x4+5) flip(a+5) x1+5 x(end)+5    x(end)+5 ...
76                17.5        17.5        x(end)+5    flip(x+5)  flip(a3+5)
 x7+5  flip(x6+5)  flip(a1)+5  x3+5  flip(x)    flip(a3)    x7    flip(
x6)   flip(a1)    x3    x3(end)    x2(end)     x2(end)      -12.5
   -12.5 x2(end) flip(x2)];
77            y_edge = [flip(b2) y5 flip(y4) flip(b) y1 flip(y2)   flip(b2)
  y5   flip(y4)   flip(b)   y1   y1(end)-0.4 y1(end)-2.5 ...
78                y1(end)-2.5 y(end)+57.5 y(end)+57.5 flip(y+55) flip(b3+55)
 y7+55 flip(y6+55) flip(b1)+55 y3+55 flip(y)+55 flip(b3)+55 y7+55 flip(
y6+55) flip(b1)+55 y3+55 y3(end)+55 y3(end)+55.4 y3(end)+57.5 y3(end)
+57.5 -2.5 -2.5 flip(y2)];
79            plot(x_hole, y_hole, 'k', x_rs, y_rs, 'k', x_ls, y_ls, 'k')
80            plot(x_edge, y_edge, 'k')
```

```
        end

%% Save as DXF
    FID = dxf_open(filename);
        dxf_polyline(FID,transpose(x_hole),transpose(y_hole),zeros(size(
    transpose(x_hole),1)));
        dxf_polyline(FID,transpose(x_rs),transpose(y_rs),zeros(size(
    transpose(x_rs),1)));
        dxf_polyline(FID,transpose(x_ls),transpose(y_ls),zeros(size(
    transpose(x_ls),1)));
        dxf_polyline(FID,transpose(x_edge),transpose(y_edge),zeros(size(
    transpose(x_edge),1)));
    dxf_close(FID);
        set(gcf, 'Renderer', 'painters');
    end
end
```

```
xo =2;
yo=1.6;
os = 0.4; %thickness scalar
for xo = 2:0.8:4
    set(0, 'DefaultLineLineWidth', 2);
    for yo = 2.4:0.8:4
        if yo==2.4
            color = '#D95319';
        elseif yo==3.2
            color='#77AC30';
        else
            color='#0072BD';
    end
        filename = [num2str(xo), '.', num2str(yo),'0.4','.dxf'];
        axis padded
        axis equal
```

```
17          axis off
18
19
20          %%Create first arm
21          cpts = [0 xo 2.5 2.5; 0 yo 0 0];
22          tpts = [0 5];
23
24          tvec = 0:0.1:5;
25          [q, qd, qdd, pp] = bsplinepolytraj(cpts,tpts,tvec);
26
27          z = zeros(size(q));
28          X = q(1,:);
29          Y = q(2,:);
30          Z = z(1,:);
31          hold on;
32          plot(X,Y, 'color', color);
33          plot(X(1,19),Y(1,19),'*k','markersize',5);
34          label = '('+string(xo)+','+string(yo)+')';
35          labelpoints(X(19),Y(19),label,'N',0.2,1);
36          hold on;
37      end
38  end
39  hold on;
40  plot(0,0, '*k', 'MarkerSize', 5);
41  labelpoints(0,0, '(0,0)','S',0.2,1);
42  plot(2.5,0, '*k', 'MarkerSize', 5);
43  labelpoints(2.5,0, '(2.5,0)','S',0.2,1);
44  hold off;
```

Bibliography

Abiri, B., Arya, M., Bohn, F., et al., A lightweight space-based solar power generation and transmission satellite arXiv preprint arXiv:2206.08373

Ai, C., Chen, Y., Xu, L., et al., Current Development on Origami/Kirigami-Inspired Structure of Creased Patterns toward Robotics Advanced Engineering Materials, 23, 2100473, doi: https://doi.org/10.1002/adem.202100473

Blees, M. K., Barnard, A. W., Rose, P. A., et al., Graphene kirigami Nature, 524, 204, doi: 10.1038/nature14588

CalTech. 2017, Space Solar Power Project, https://youtu.be/KtNwYweL6hY

CMU. n.d. https://www.cs.cmu.edu/afs/cs/academic/class/15462-s10/www/lec-slides/lec06.pdf

DuPont. 2022, Kapton® Summary of Properties, https://www.dupont.com/content/dam/dupont/amer/us/en/ei-transformation/public/documents/en/EI-10142_Kapton-Summary-of-Properties.pdf

ESA. 2021, Temperatures on the Space Station. https://www.esa.int/ESA_Multimedia/Images/2021/08/Temperatures_on_the_Space_Station

Frame, D., & Revell, L. 2023, Atmospheric, Oceanic, and Climate Dynamics

Greene, J. P. 2021, 11 - Thermoset Polymers, ed. J. P. Greene, Plastics Design Library (William Andrew Publishing), 175–190, doi: 10.1016/B978-0-12-818008-2.00002-7

He, W., Goudeau, P., Le Bourhis, E., et al., Study on Young's modulus of thin films on Kapton by microtensile testing combined with dual DIC system Surface and Coatings Technology, 308, 273, doi: https://doi.org/10.1016/j.surfcoat.2016.07.114

HeegerMaterials. 2024, List of materials that can withstand high temperatures. https://heegermaterials.com/blog/73_list-of-materials-that-can-withstand-high-tem.html

Isobe, M., & Okumura, K., Initial rigid response and softening transition of highly stretchable kirigami sheet materials Scientific Reports, 6, 24758

Mathworks. 2024, bsplinepolytraj. https://www.mathworks.com/help/robotics/ref/bsplinepolytraj.html

NASA. 2024a, SLS Overview Fact Sheet, National Aeronautical and Space Administration. https://www.nasa.gov/wp-content/uploads/2024/10/sls-4960-sls-fact-sheet-oct2024-508.pdf?emrc=7e096d

—. 2024b, Launch Services Program Rockets - NASA. https://www.nasa.gov/launch-services-program-rockets/

Rafsanjani, A., & Bertoldi, K., Buckling-induced kirigami Physical Review Letters, 118

RocketLab. 2025, Electron. https://www.rocketlabusa.com/launch/electron/

SpaceX. 2025a, SpaceX - Starship. https://www.spacex.com/vehicles/starship/

—. 2025b, SpaceX - Falcon Heavy. https://www.spacex.com/vehicles/falcon-heavy/

—. 2025c, SpaceX - Falcon 9. https://www.spacex.com/vehicles/falcon-9/

Taniyama, H., & Iwase, E., Design of Rigidity and Breaking Strain for a Kirigami Structure with Non-Uniform Deformed Regions Micromachines, 10, doi: 10.3390/mi10060395

ULALaunch. 2025, Atlas V. https://www.ulalaunch.com/rockets/atlas-v

Wong, Y. W., & Pellegrino, S., Wrinkled Membranes Part III: Numerical Simulations Journal of Mechanics of Materials and Structures, 1