Theoretical, computational, and experimental characterization of nematic elastomers

Thesis by
Victoria Lee

In Partial Fulfillment of the Requirements for the
Degree of
Doctor of Philosophy

CALIFORNIA INSTITUTE OF TECHNOLOGY
Pasadena, California

2021
Defended May 24, 2021

Victoria Lee
ORCID: 0000-0002-2748-0089

ACKNOWLEDGMENTS

I'd like to first thank my mentor and advisor, Kaushik Bhattacharya. Kaushik has been a wonderful guide throughout my entire grad career, from academics and research to career advising and mentorship. He has gifted me with the space and time to grow as a researcher here at Caltech, providing me with ample opportunities for enrichment through research collaborations, teaching assistantships, and conference presentations. I would not have been able to accomplish all of this without his valuable insight, and I'm truly grateful for all the knowledge that I've gained while exploring my research under Kaushik's advisement.

I'd also like to thank my thesis committee members, two of whom also served on my candidacy committee a few years ago: G. Ravichandran, Chiara Daraio, and Sergio Pellegrino. Ravi has always taken the time to check in with me and offer his help whenever I needed it. He is an inspiring teacher, and I value his advice greatly. Chiara is wonderfully supportive and inspiring, and I admire her enthusiasm and devotion. I'm grateful for the cheerful collaboration that exists between Chiara's lab and Kaushik's, which are just next door to each other. I'd like to thank Sergio for leading thought-provoking discussions and providing helpful feedback during our MURI meetings.

I'd like to extend a huge thank you to the inspiring professors I had at Brown, who provided me with a solid foundation for engineering and advised me throughout my undergraduate career: Haneesh Kesari, Karen-Marie Haberstroh, Allan Bower, Jennifer Franck, Christian Franck, David Henann, Pradeep Guduru, and many more.

I have been fortunate to work with great collaborators while exploring various research directions of nematic elastomers: Kenji Urayama, Pierluigi Cesana, Tim White and his group at Wright-Patterson AFRL, Taylor Ware, Cedric Ambulo, and Connor McMahan. I am grateful for the financial support of the US Air Force Office of Scientific Research through the MURI Grant No. FA9550-16-1-0566.

I have really enjoyed being a part of Kaushik's research group and getting to know its members. I thank them for fun conversations within our office and around Gates-Thomas: Ying Shi, Paul Plucinsky, Paul Mazur, Jin, Ruobing, Stella, Burigede, Kevin, Hao, Sharan, Eric, Lincoln, Louisa, Dingyi, Andy, and many more. I am deeply grateful to Jenni Campbell for helping me with any kind of administrative and logistical issue I have; you always take care of us with a smile! I'm also glad to have

gotten to know many of the members of Ravi, Chiara, Michael, and Sergio's groups through group meetings and lab collaborations, especially Erika, Will, Trent, Amir, Xingsheng, Manav, Eleftheria, Connor, Suraj, Vatsa, Jack, Fabien, and Charlie.

I want to thank various members of the MCE department for making my past five to six years an inclusive and enjoyable place to work. I feel very lucky to have had a great first-year cohort (including our counterparts in GALCIT!) to collaborate with on classwork and to study for quals together. I'm really grateful to all of my fellow SOPS leaders for devoting endless hours towards making the department a community, especially Will, Marcus, Leah, Alex, Kevin, Prithvi, Maegan, Natalie, and Dingyi. It was a pleasure working with Nadia, José, Melany, Tim, Domniki, Guillaume, and many other professors in the context of SOPS initiatives. The MCE staff are absolutely amazing and such instrumental members of the community: Jenni, Lynn, Holly, Sonya, Carolina, and Mikaela. I'm also extremely grateful to John, Bruce, and Paul for all their help in the machine shop.

I'm so thankful for all of the staff at the CCID for their tireless DEI work, especially Hanna Song and Erin-Kate for helping us out with the DEI database and organizing Women in MCE events! I also want to thank Grace Ho, Maggie Ateia, and Liz Jackman for their absolutely wonderful support!

I want to thank all of my wonderfully supportive friends. Amanda, thank you for being the best roommate and friend ever since we met on those long-ago Visit Days, and throughout all the ups and downs we've shared since. Kavya, Ying Shi, Erika, and Becky, thank you for bringing me joy and a deep sense of belonging both on and off campus. Nick and Eric, thank you especially for all of the fun times exploring Pasadena and SoCal.

Finally, I want to thank my incredible family, who have been with me every step of the way. Mom, Dad, and Oppa: you guys are simply the best, and I'm so fortunate to have your support and love. Stephen, thank you for absolutely everything, but especially in the context of this thesis, for the times you helped me debug my code or troubleshoot something electrical engineering-y in the lab!

ABSTRACT

Nematic elastomers are programmable soft materials that display large, reversible, and predictable deformation under an external stimulus such as a change in temperature or light. They are composed of a lightly crosslinked polymer network with stiff, rod-like liquid crystal molecules incorporated within the polymer chains. In thermotropic nematic elastomers, the liquid crystals undergo a continuous and reversible phase transition between the randomly oriented isotropic state and the highly oriented nematic state. Further, there is a direct thermo-mechanical coupling between the underlying temperature-responsive orientational order of the liquid crystal molecules and the macroscopic shape change of the surrounding elastomer chains. Finally, these materials display an unusually soft behavior. These remarkable properties make them promising materials for applications in aerospace as deployable structures and skins, in biomedical engineering as a soft pump, and in communications as the actuation mechanism in a reconfigurable antenna. Motivated by these applications, this thesis discusses the theoretical, computational, and experimental characterization of nematic elastomers.

We begin by investigating an example of actuation that takes advantage of the programmable, soft nature of these materials as well as instabilities associated with large deformation. We outline the multi-stable equilibrium solutions to a cylindrical balloon subjected to internal inflation, the material's microstructure formation due to this deformation, and its use as a soft pump with large ejection fraction, which involves a snap-through instability. Then we extend the Agostiniani-DeSimone-Dolzmann relaxed energy to a generalized Mooney-Rivlin constitutive relation and study four examples of Ericksen's universal deformations—the inflation of cylindrical and spherical balloons, the cavitation of a disk, and the bending of a block.

We then move beyond the modeling of ideal materials and present a new constitutive relation for isotropic-genesis polydomain nematic elastomers. It is based on internal variables that describe the fine-scale domain patterns and evolve according to a kinetic process with dissipation. We discuss the model's implementation in the commercial finite-element software, `ABAQUS`, and study the problem of torsion of a cylinder. We identify an interesting instability at large torsional strains as a result of the Poynting effect. Finally, we present the design of a thermo-

mechanical tensile setup and the experimental results for strain-rate dependence and temperature-dependence of samples that we synthesize in-house.

PUBLISHED CONTENT AND CONTRIBUTIONS

Victoria Lee and Kaushik Bhattacharya. Actuation of cylindrical nematic elastomer balloons. *Journal of Applied Physics*, 129(114701), 2021. doi: 10.1063/5.0041288. URL https://doi.org/10.1063/5.0041288.

V. Lee performed all calculations, created all plots, and co-wrote the manuscript.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

Chapter 1

INTRODUCTION

1.1 Nematic elastomers

Liquid crystal elastomers (LCEs) are remarkable stimuli-responsive materials that have recently been explored for their use in actuation [77]. LCEs are lightly crosslinked elastomer networks with liquid crystal molecules incorporated into the underlying polymer chains. P. G. de Gennes first envisioned the coupling between the orientational order of liquid crystals with the macroscopic shape change of a crosslinked elastomer network in 1975 [21]. There are different types of liquid crystal elastomers, including nematic, cholesteric, and smectic, and they can be responsive to different types of stimuli, including a change in temperature, pH, electric field, and light.

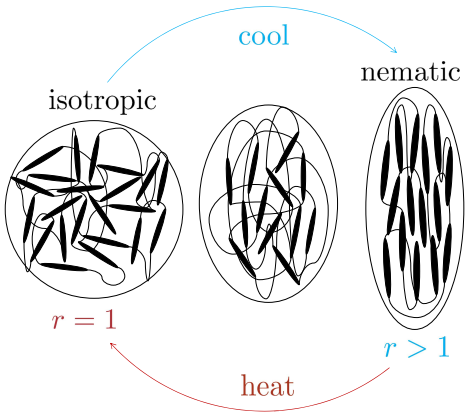


Figure 1.1: Isotropic-nematic phase transition in nematic elastomers with the isotropic with $r = 1$ at high temperatures and nematic phase with $r > 1$ at low temperatures.

Of particular interest are nematic elastomers, which are liquid crystal elastomers with stiff, rod-like liquid crystal molecules. They undergo an isotropic-to-nematic transition accompanied by a significant stretch (by as much as a factor of two or more) along the nematic director (in the direction parallel to a unit vector $\boldsymbol{n}$) and lateral contraction, as shown schematically in Figure 1.1. In this thesis, we focus on thermotropic nematic elastomers, which undergo the phase transition at a characteristic nematic-isotropic temperature T_{ni}. At temperatures above T_{ni}, the

material is in the isotropic state, where the liquid crystal molecules are randomly oriented, and at temperatures below T_{ni}, the material is in the nematic state, where the liquid crystal molecules are aligned along a preferred direction, denoted by $\boldsymbol{n}$. Because of the direct thermo-mechanical coupling between the nematic orientation and the surrounding polymer network, this phase transition induces a deformation of the underlying polymer chains in the nematic elastomer, so a macroscopic shape change is observed, in which the polymer chains stretch parallel to the mesogen alignment and contract perpendicular to it. The phase transition, and consequently the change of shape, is continuous and completely reversible. The degree of order observed in the liquid crystal molecules determines the degree of anisotropy, denoted by r throughout this thesis.

These materials can be synthesized as a monodomain or polydomain. Monodomains can be thought of as the elastomer equivalent of a single crystal, in which the entire sample is aligned along one direction in the nematic state. A polydomain sample is macroscopically randomly oriented in the nematic state, but at a mesoscale, there is nematic alignment within each domain (see Figure 1.2 for a schematic).

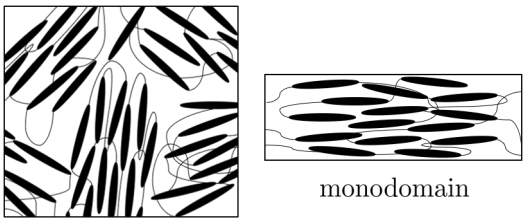


Figure 1.2: Schematics of polydomain and monodomain nematic elastomer.

1.2 Actuation

The study of materials for actuation applications traditionally focused on systems driven by mechanisms such as pneumatics, hydraulics, or motors. Active materials such as piezoelectrics, dielectric elastomers, a type of electroactive polymer (EAP), and shape memory alloys which undergo the austenite-martensite phase transition, offer much in the way of actuation without complex hardware and moving parts. Using the material as the machine [6] is a powerful idea motivating the study of active materials.

Nematic elastomers are advantageous choices in the field of active materials because they are soft, they can deliver a large energy per unit volume, their shape change is completely reversible in the presence or absence of the stimuli, and their director field

is programmable [2, 48, 49, 57, 76]. Further, it is possible to incorporate photo-active molecules into nematic elastomers, giving rise to photo-active materials that are actuated by light. Some compelling examples include refreshable Braille displays for reading instruments for the visually impaired [14], untethered robotic matter [41], and shape morphing [26]. Additionally, nematic elastomers can be programmed for wrinkle control in thin sheets due to microstructure development, for possible applications in deployable structures for aerospace applications [56].

1.3 Beyond actuation from flat sheets

Nematic elastomers have been exploited for programmable actuation and shape-morphing of thin sheets. Modes, Bhattacharya, and Warner [48, 49] suggested that if sheets of nematic elastomer with prescribed director patterns were fabricated in the nematic state and subsequently heated, they could deform out of plane into three-dimensional shapes. For example, a +1 disclination with an azimuthal director distribution would deform into a cone. This was demonstrated in nematic glasses by de Haan *et al.* [22]. Ware *et al.* [76] developed a method of synthesizing nematic elastomers where the director pattern could be written pixel by pixel on flat sheets and demonstrated the formation of these cones. Moreover, they showed that this actuation was extremely robust, as the cone-lifting weights were many hundreds of times larger than the structure itself. Since then, there have been a number of other studies on nematic elastomers [2, 47, 50, 57, 58], including the inverse problem of identifying the director pattern that would lead to a given actuated shape [3, 57]. All of these works address the programming and actuation of initially flat sheets.

Recent advances in 3D printing and other methods of directed synthesis have enabled the synthesis of curved shells [4, 32], and such structures change shape upon heating. In particular, Ambulo *et al.* demonstrated dramatic snap-through buckling of structures involving regions of positive Gauss curvature and regions of negative Gauss curvature [4]. More recently, magnetic fields have been used to independently control director orientation during 3D printing [68]. These developments in synthesis techniques motivate the work in Chapter 2, where we analyze a balloon made of nematic elastomer, subjected to internal inflation. The study of these materials in different configurations beyond flat sheets opens the doors to applications with more complex boundary conditions and loading configurations.

1.4 Microstructure formation

The interesting material properties of nematic elastomers are due to the interplay amongst features at distinct length scales. The macroscopic shape of the lightly crosslinked polymer network, which can be on the order of centimeters, is determined by the nematic mesogens, which can have a length on the order of nanometers, and domains of nematic alignment can exist in some mesoscale with a characteristic length of micrometers.

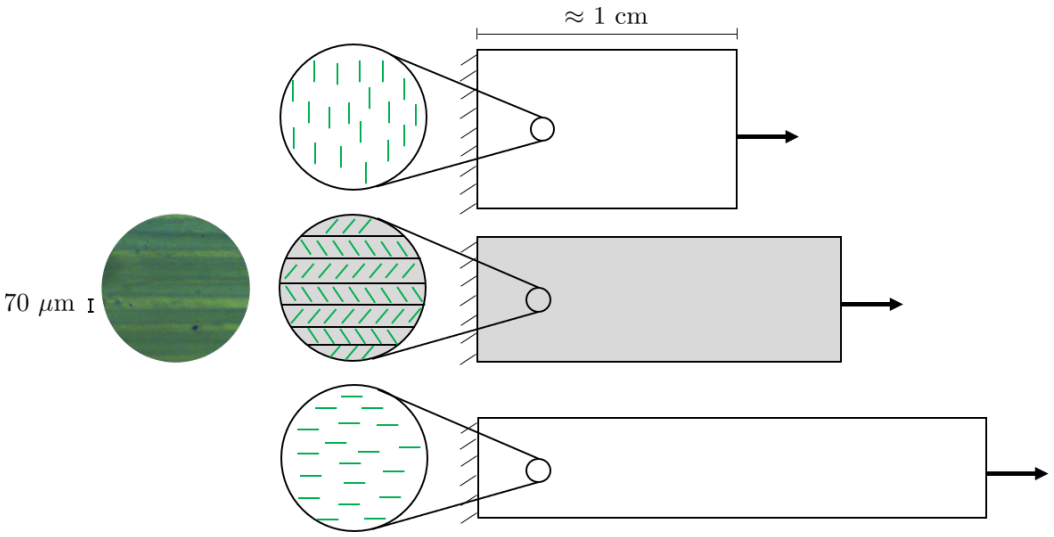


Figure 1.3: Schematic of the experiment in which a monodomain sample is pulled perpendicular to its nematic director, giving rise to stripe domains, which can be seen under polarized light microscopy (in the left-most circle).

The stripe-domain formation in monodomains pulled perpendicular to their nematic alignment exhibit the classical example of fine-scale microstructure formation (see Figure 4 of [42], also shown schematically in Figure 1.3). In this experiment, a monodomain sample with a uniform nematic alignment is clamped on one end and pulled uniaxially in the direction perpendicular to the nematic director. The liquid crystal molecules rotate to align themselves parallel to the direction of stretch, and they do so by forming alternating stripes in which the molecules rotate at opposite angles. When the liquid crystal molecules complete this reorientation process, the sample becomes a monodomain with the nematic director parallel to the direction of stretch. The stripe domains can be seen using polarized light microscopy, an example of which is shown in the left-most circle of Figure 1.3. The stripes are relatively uniform, and the width can vary depending on the specific synthesis methods and possibly also the geometry of the sample. The stripes exhibited by the

samples in our lab are about 70 micrometers in width, whereas the stripes in Figure 4 of [42] have a width closer to 15 micrometers.

At the beginning of the experiment, the clamped monodomain sample appears transparent because all of the liquid crystal molecules are aligned uniformly, but as the experiment progresses, the sample becomes opaque as the microstructure develops, and the liquid crystal molecules are no longer uniformly aligned. Eventually, when the liquid crystal molecules have finished reorienting and are again uniformly aligned, the sample has become a monodomain and therefore appears transparent.

The microstructure formation gives rise to the phenomenon of soft elasticity—the reorientation of the mesogens happens at zero stress, which is depicted by the stress plateau in stress-strain curves of the uniaxial extension of monodomains pulled perpendicular to their nematic alignment (see Figure 9 of [43]). Stripe domains are discussed in the context of the cylindrical balloon explored in Chapter 2.

A related experiment is the uniaxial stretch of a polydomain sample, which undergoes a transition known as the polydomain-to-monodomain transition. The associated stress-strain curve also exhibits soft elasticity, as mesogens within each mesoscale domain reorient towards the direction of most stretch. The soft behavior has largely been studied in uniaxial deformation, and more recently in biaxial deformation [71]. This motivates Chapter 3, where we study the consequences of softness on complex deformations.

1.5 Viscoelasticity and damping in nematic elastomers

It is important to understand and characterize the effect of viscoelasticity, or time dependence, in nematic elastomers if they are to be used in actuation applications, which are typically cyclic in nature. Existing work in this field includes stress-relaxation experiments on these materials, as well as uniaxial experiments at varying strain rates, e.g. through the polydomain-monodomain (P-M) transition [5, 16, 36, 54, 67, 69]. Some of these authors have also used dynamic mechanical analysis (DMA) of monodomain samples under simple shear. Hotta and Terentjev also applied the principle of time-temperature superposition to build a master curve of the material response over a range of frequencies using data from tests at different temperatures, e.g. [5, 35].

The large hysteresis between the loading and unloading stress-strain curves in uniaxial experiments on polydomain samples, e.g. in Figure 5 of [5], is indicative of the material's ability to dissipate a large amount of energy. Merkel *et al.* [46]

investigated the effect of temperature, and therefore degree of anisotropy, in the dynamic loading of polydomain nematic elastomers. A combination of the reorientation of the liquid crystal molecules and the viscosity of the polymer chains provides a mechanism to dissipate and absorb applied mechanical energy. Clarke *et al.* discussed the potential for nematic elastomers in mechanical damping applications due to the internal relaxation of the liquid crystal molecules, independent of the polymer backbone [17]. Examples of applications where efficient dissipation of mechanical energy are desirable include the automotive, aerospace, and white goods (e.g. washing machines, refrigerators) industries. One can imagine tuning the mesogen response for variable vibration and noise suppression.

We address the formulation of a finite-deformation constitutive relation that could model such viscoelastic effects in Chapter 4, which incorporates viscosity associated with the polymer network as well as evolution of some internal variables according to a dissipative kinetic process.

1.6 Nematic elastomers as an engineering material

Studies of monodomain nematic elastomers are extremely useful to characterize a structure or be able to predict material response, especially for actuation applications. However, polydomains, with their domains of fine-scale microstructure, are easier and less expensive to manufacture because they require no director alignment, and can be viewed as an engineering material.

Previous theories largely dealt with simple deformations or idealized materials. The goal of Chapter 4 is to formulate a model for nematic elastomers as an engineering material—with microstructure formation through the relaxed energy and viscoelasticity through the internal variables.

1.7 Thesis outline

This thesis is organized in the following way: First, in Chapter 2, we study the actuation of a cylindrical nematic elastomer balloon. This work is a step away from the established field of actuation from flat sheets, as we focus on actuation from an already three-dimensional reference state. We analyze the deformation of a cylindrical shell of a patterned nematic elastomer under pressure, show that it can undergo an enormous change of volume with changing temperature, and suggest its application as a pump with extremely high ejection fraction.

In Chapter 3, we build upon the framework of DeSimone and Dolzmann [23] and Agostiniani and DeSimone [1] for the relaxed energy of nematic elastomers. We again extend this energy into a generalized Mooney-Rivlin type energy density to capture correctly the elasticity of these materials at very high stretch, and we use this model to solve four examples of Ericksen's so-called universal deformations: expansion of a spherical balloon, expansion of a cylindrical balloon, cavitation of a disk, and bending of a block. The solutions are presented for varying anisotropy parameters, including the case where the material is isotropic, which corresponds to a rubber without liquid crystals.

Chapter 4 presents an engineering model involving internal variable evolution that captures the macroscopic behavior of isotropic-genesis polydomain nematic elastomers. We present the formulation for the model, its implementation in the commercial finite-element software `ABAQUS`, and some results under various deformations, including planar extension, biaxial stretch, and torsion. Our numerical results identify an unusual torsional instability driven by the Poynting effect.

In Chapter 5, we present the synthesis technique of thermotropic nematic elastomer samples, the design of our custom-built thermo-mechanical test setup, and the test results from uniaxial extension of these samples in the experimental setup at various crosslinked configurations and temperatures.

Finally, we summarize the findings of this thesis and conclude with comments on the future outlook of this exciting and expansive field of research in Chapter 6. Nematic elastomers play only a small role in the field of stimulus-responsive actuators. There are many promising avenues for further research and development of these materials, experimentally, theoretically, and computationally.

Chapter 2

ACTUATION OF CYLINDRICAL NEMATIC ELASTOMER BALLOONS

Victoria Lee and Kaushik Bhattacharya. Actuation of cylindrical nematic elastomer balloons. *Journal of Applied Physics*, 129(114701), 2021. doi: 10.1063/5.0041288. URL https://doi.org/10.1063/5.0041288.

2.1 Introduction

The goal of this work is two-fold. The first is to explore the combination of programmed synthesis of nematic shells and the geometric instabilities associated with the large deformation of slender structures. Similar instabilities have been exploited in other stimuli-responsive materials including electroactive materials [33, 63, 84]. In this work, we focus on the so-called aneurysm instability of pressurized cylinders [28]. As observed in long toy balloons, one observes a discontinuous change of radius (or volume) with an increase of pressure: typically the balloon inflates till it reaches a particular radius, beyond which point a bump (aneurysm) with a significantly larger radius appears in this region, and it propagates through the entire balloon before the radius further increases. We explore the response of a cylindrical shell made of a nematic elastomer and study how the isotropic-nematic phase transition affects this instability. Our work is closely related to that of Giudici and Biggins [31] who recently studied the ballooning instability in both nematic and isotropic LCEs using a Gent-style energy. He *et al.* [34] have studied the anomalous behavior of (isotropic-genesis polydomain) nematic balloons under tension. We then show how this instability can be used as a high ejection-fraction pump. The second goal is to study actuation and shape-morphing in the presence of mechanical loads. The prior literature has largely focused on free recovery.

We introduce the model of the nematic elastomers at large deformation in Section 2.2 and analyze the deformation of a nematic elastomer cylinder under internal pressure in Section 2.3. We then use the results to motivate a pump with extremely large ejection fraction in Section 2.4.

2.2 Large deformation model of nematic elastomers

We begin with the neo-classical theory of nematic elastomers following Bladon, Terentjev and Warner [10, 77]. The state of a liquid crystal elastomer is described by an anisotropy parameter r, a director $\boldsymbol{n}$ and the deformation gradient $\boldsymbol{F}$ relative to a stress-free reference configuration with anisotropy parameter r_0 and director $\boldsymbol{n}_0$. The anisotropy parameter is a function of temperature with $r = 1$ in the isotropic state above the transformation, and gradually increases with decreasing temperature so that $r > 1$ in the nematic state. We consider the material to be incompressible so that $\det \boldsymbol{F} = 1$. The neo-classical theory considers the entropy of the polymer chains in the Gaussian approximation, and the free energy density is given as

$$W_{\mathrm{WT}}(\boldsymbol{F}, \boldsymbol{n}, r) = \frac{\mu}{2}\left(\operatorname{tr}\left(\boldsymbol{\ell}_{n_0}\boldsymbol{F}^T\boldsymbol{\ell}_n^{-1}\boldsymbol{F}\right) - 3\right), \tag{2.1}$$

where μ is the shear modulus of the material, and

$$\boldsymbol{\ell}_n = r^{-1/3}\left(\boldsymbol{I} + (r-1)\boldsymbol{n}\otimes\boldsymbol{n}\right) \tag{2.2}$$

$$\boldsymbol{\ell}_{n_0} = r_0^{-1/3}\left(\boldsymbol{I} + (r_0-1)\boldsymbol{n}_0\otimes\boldsymbol{n}_0\right) \tag{2.3}$$

are the step-length tensors in the current and reference configurations that collect the anisotropy parameter and the director. It is easy to show that

$$W_{\mathrm{WT}}(\boldsymbol{F}, \boldsymbol{n}, r) = W_{\mathrm{NH}}\left(\boldsymbol{\ell}_n^{-1/2}\boldsymbol{F}\boldsymbol{\ell}_{n_0}^{1/2}\right), \tag{2.4}$$

where $W_{\mathrm{NH}}(\boldsymbol{F}) = \frac{\mu}{2}(\operatorname{tr}\boldsymbol{C}(\boldsymbol{F})) - 3)$, with $\boldsymbol{C}(\boldsymbol{F}) = \boldsymbol{F}^T\boldsymbol{F}$, is the neo-Hookean energy density which describes the entropy of polymer chains in ordinary rubber in the Gaussian approximation [72].

The neo-classical theory is known to describe complex features of nematic elastomers at finite, but moderate, deformation. However, at extremely large stretches, the Gaussian approximation does not hold, and this theory does not adequately describe the stiffening much like its neo-Hookean counterpart. Various constitutive relations are used to describe rubber in this high-stretch regime. A common feature of many of these models is that the energy density depends only on principle stretches λ_i of $\boldsymbol{F}$ (equivalently the eigenvalues λ_i^2 of $\boldsymbol{C}(\boldsymbol{F})$):

$$W_{\mathrm{E}}(\boldsymbol{F}) = f(\lambda_1, \lambda_2, \lambda_3). \tag{2.5}$$

For example, in the Ogden model [53] the energy density is

$$W_{\mathrm{O}}(\boldsymbol{F}) = \sum_{p=1}^{N}\frac{\mu_p}{\beta_p}\left(\lambda_1^{\beta_p} + \lambda_2^{\beta_p} + \lambda_3^{\beta_p} - 3\right), \tag{2.6}$$

where N, μ_p, and β_p are material constants. The shear modulus is $\mu = \frac{1}{2}\sum_{p=1}^{N} \mu_p\beta_p$. When $N = 1$ and $\beta_1 = 2$, the Ogden energy is the neo-Hookean energy, and when $N = 2$, $\beta_1 = 2$, and $\beta_2 = -2$, the Ogden energy is the Mooney-Rivlin energy. We use the Ogden energy to demonstrate our results following [63], though we can adapt them to any constitutive relation that describes the high stretch behavior. We adopt the elastic energy density (2.5) to nematic elastomers analogously to (2.4). See [1] for similar energies and their relaxation in the ideal case. Other approaches have been proposed to capture the high-stretch regime including the logarithmic correction by Gent [27] which was used by Giudici and Biggins [31] in their work.

Further, the cross-link density and the polymer network may carry an imprint of the initial director, and this leads to a breaking of symmetry (isotropy) leading to the preference of the director to remain in the original orientation. Such an interaction can be described using an additional non-ideal energy density [9]:

$$W_{\mathrm{NI}}(\boldsymbol{F}, \boldsymbol{n}) = \alpha\frac{\mu}{2}\mathrm{tr}\left(\boldsymbol{F}(\boldsymbol{I} - \boldsymbol{n}_0 \otimes \boldsymbol{n}_0)\boldsymbol{F}^T\boldsymbol{n} \otimes \boldsymbol{n}\right). \tag{2.7}$$

Note that this energy is minimized when $\boldsymbol{n} = \boldsymbol{n}_0$. Putting these together, we take the energy density of the nematic elastomer to be

$$W(\boldsymbol{F}, \boldsymbol{n}, r) = W_{\mathrm{E}}\left(\boldsymbol{\ell}_n^{-1/2}\boldsymbol{F}\boldsymbol{\ell}_{n_0}^{1/2}\right) + W_{\mathrm{NI}}(\boldsymbol{F}, \boldsymbol{n}). \tag{2.8}$$

For future use, we note a particular invariance of this energy density. Let $\boldsymbol{Q}$ be a rotation tensor that leaves the reference director invariant: $\boldsymbol{Q}\boldsymbol{n}_0 = \pm\boldsymbol{n}_0$. Then, we claim that

$$W(\boldsymbol{Q}\boldsymbol{F}\boldsymbol{Q}^T, \boldsymbol{Q}\boldsymbol{n}, r) = W(\boldsymbol{F}, \boldsymbol{n}, r). \tag{2.9}$$

Note that

$$\begin{aligned}
\boldsymbol{C}((\boldsymbol{Q}\boldsymbol{\ell}_n\boldsymbol{Q}^T)^{-1/2}(\boldsymbol{Q}\boldsymbol{F}\boldsymbol{Q}^T)\boldsymbol{\ell}_{n_0}^{1/2}) &= \boldsymbol{\ell}_{n_0}^{1/2}(\boldsymbol{Q}\boldsymbol{F}\boldsymbol{Q}^T)^T(\boldsymbol{Q}\boldsymbol{\ell}_n\boldsymbol{Q}^T)^{-1}(\boldsymbol{Q}\boldsymbol{F}\boldsymbol{Q}^T)\boldsymbol{\ell}_{n_0}^{1/2} \\
&= \boldsymbol{\ell}_{n_0}^{1/2}\boldsymbol{Q}\boldsymbol{F}^T\boldsymbol{\ell}_n^{-1}\boldsymbol{F}\boldsymbol{Q}^T\boldsymbol{\ell}_{n_0}^{1/2} \\
&= \boldsymbol{Q}\boldsymbol{\ell}_{n_0}^{1/2}\boldsymbol{F}^T\boldsymbol{\ell}_n^{-1}\boldsymbol{F}\boldsymbol{\ell}_{n_0}^{1/2}\boldsymbol{Q}^T \\
&= \boldsymbol{Q}\boldsymbol{C}(\boldsymbol{\ell}_n^{-1/2}\boldsymbol{F}\boldsymbol{\ell}_{n_0}^{1/2})\boldsymbol{Q}^T,
\end{aligned} \tag{2.10}$$

where we have used the invariance of $\boldsymbol{n}_0$ under $\boldsymbol{Q}$ in the third equality. It follows that both tensors have the same eigenvalues and thus the same Ogden energy density. A similar calculation holds for the non-ideal energy density as well, thereby establishing (2.9).

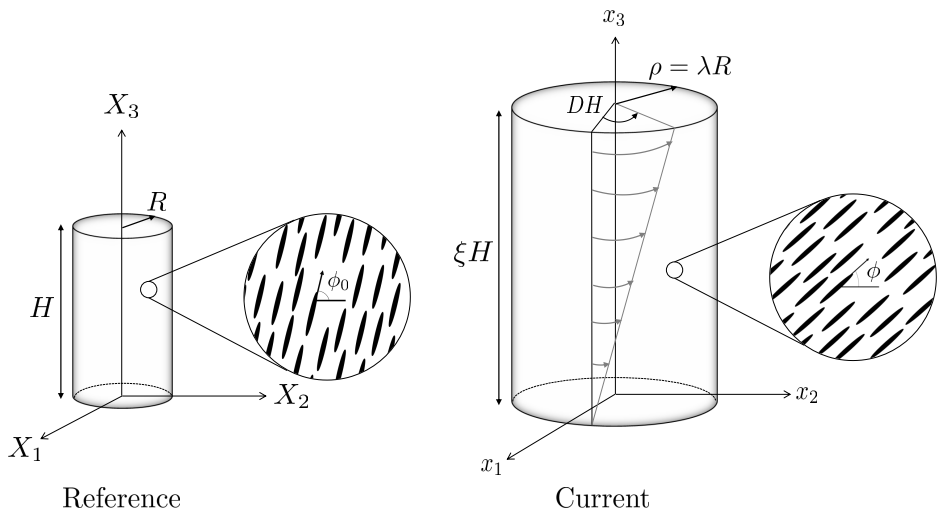


Figure 2.1: Inflation, extension, and torsion of a cylinder.

2.3 Inflation of a nematic cylinder

Consider a cylindrical shell of initial (reference) length H, inner radius R_i, and undeformed outer radius R_o subjected to an internal pressure p. Following Rivlin [61] and Ericksen [25], the deformation of the cylinder is described by the universal volume-preserving deformation involving radial expansion, axial extension and torsion (see Figure 2.1). The mapping is

$$\begin{cases} \rho = \rho(R) \\ \theta = \Theta + DZ \\ z = \xi Z \end{cases}, \tag{2.11}$$

where $\{R, \Theta, Z\}$ and $\{\rho, \theta, z\}$ denote the cylindrical coordinate system in the reference and deformed coordinate systems respectively. $\rho(R)$ describes the radial expansion, D the twist, and ξ the axial stretch. The deformation gradient in the cylindrical coordinate system is

$$\boldsymbol{F} = \begin{pmatrix} \rho' & 0 & 0 \\ 0 & \frac{\rho}{R} & \lambda R D \\ 0 & 0 & \xi \end{pmatrix} = \begin{pmatrix} \frac{1}{\lambda\xi} & 0 & 0 \\ 0 & \lambda & \lambda R D \\ 0 & 0 & \xi \end{pmatrix}, \tag{2.12}$$

where we have introduced the hoop stretch $\lambda(R) = \rho(R)/R$ and used the incompressibility to obtain the second equality. There is an off-diagonal term in the deformation gradient because we would like to allow shear or twist that may accompany director reorientation. We will see later that this indeed plays a role.

We assume that the director both in the reference and deformed configuration are tangential to the cylinder and make an angle ϕ_0 and ϕ, respectively, with the azimuthal coordinate. Thus, in cylindrical coordinates,

$$\boldsymbol{n}_0 = \begin{pmatrix} 0 \\ \cos\phi_0 \\ \sin\phi_0 \end{pmatrix} \text{ and } \boldsymbol{n} = \begin{pmatrix} 0 \\ \cos\phi \\ \sin\phi \end{pmatrix}. \tag{2.13}$$

The total potential energy of the system is

$$\Phi = \int_\Omega W(\boldsymbol{F}, \boldsymbol{n}, r) dV - p\Delta V, \tag{2.14}$$

where ΔV is the difference in the deformed and undeformed volumes. Applied to a balloon with height H, we obtain

$$\begin{aligned} \Phi &= \int_0^H \int_0^{2\pi} \int_{R_i}^{R_o} W(\boldsymbol{F}, \boldsymbol{n}, r) R dR d\Theta dZ - p\left(\pi\rho^2\xi H - \pi R^2 H\right)\Big|_{R=R_i} \\ &\approx 2\pi R_i H T W(\boldsymbol{F}, \boldsymbol{n}, r) - p\pi R_i^2 H\left(\xi\lambda^2 - 1\right). \end{aligned} \tag{2.15}$$

Above we have assumed that the shell is thin, $T := (R_o - R_i) << R_i$, to evaluate the integral.

For a given pressure p and anisotropy parameter r, we can now find the equilibrium as

$$\frac{\partial\Phi}{\partial\lambda} = \frac{\partial\Phi}{\partial\xi} = \frac{\partial\Phi}{\partial D} = \frac{\partial\Phi}{\partial\phi} = 0. \tag{2.16}$$

Physically, these equations describe the balance between the hoop stress in the cylinder and the internal pressure, the balance between the axial stress and the internal pressure, the balance of torque, and the balance of internal (material) torque on the director respectively.

To demonstrate the results, we consider a cylinder where the initial director is axial ($\phi_0 = 90°$) and which is mildly nematic with initial anisotropy parameter $r_0 = 2$. The rest of the parameters are shown in Table 2.1, and were chosen to be broadly consistent with an experiment conducted in our laboratory. We fix the current anisotropy parameter r and the hoop stretch λ and solve (2.16) for the pressure p, axial stretch ξ, the twist D, and the current director angle ϕ. We find that the system has two solutions, shown in Figure 2.2 for four different current anisotropy parameters r[1] and the classical neo-Hookean case, $r_0 = r = 1$. Note that the pressure has been normalized as $\hat{p} = \frac{pR_o}{\mu(R_o - R_i)}$, where $\mu = \frac{1}{2}\sum_{p=1}^{3}\mu_p\beta_p = 6.80 \cdot 10^4$ Pa represents the reference shear modulus.

[1]There is a third unstable solution where the director does not rotate, which we ignore.

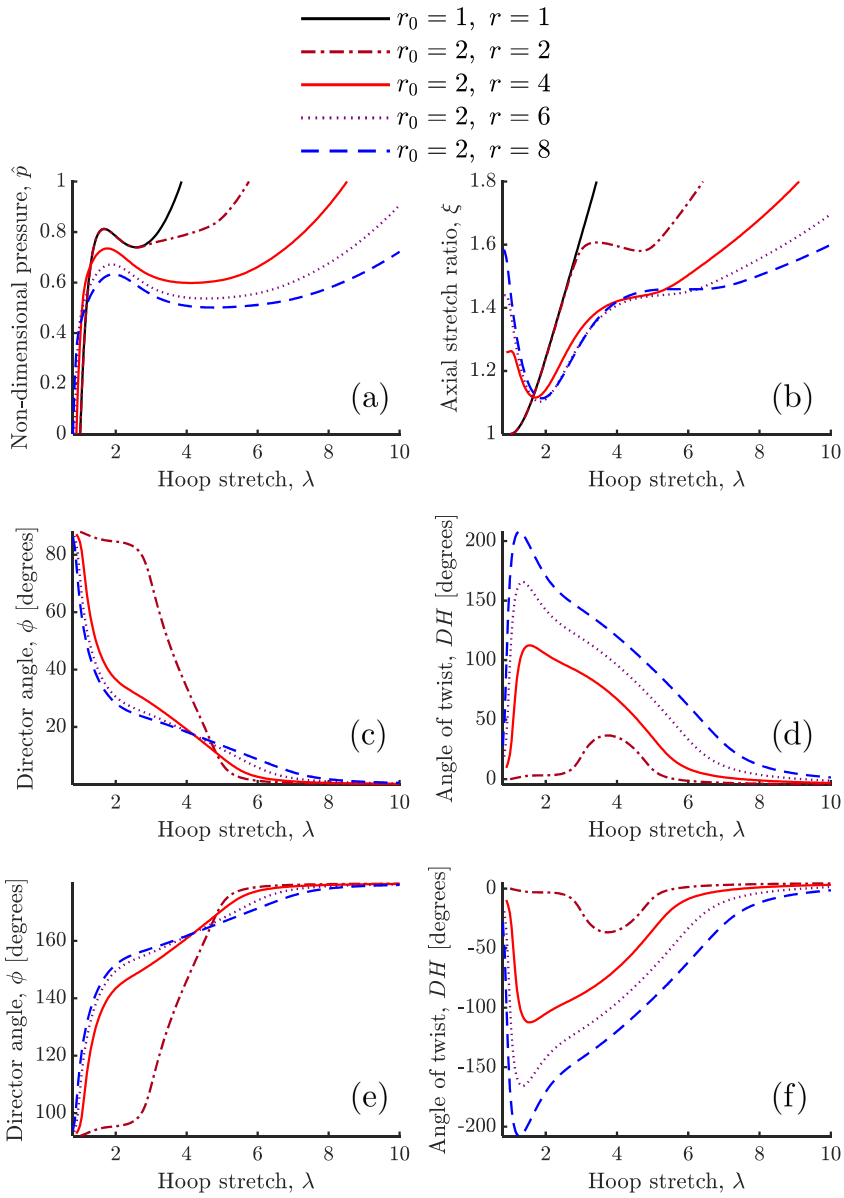


Figure 2.2: Inflation of a nematic cylinder: (a) Pressure vs. hoop stretch (solutions 1 and 2), (b) Axial stretch vs. hoop stretch (solutions 1 and 2), (c) Director angle vs. hoop stretch (solution 1), (d) Twist vs. hoop stretch (solution 1), (e) Director angle vs. hoop stretch (solution 2), (f) Twist vs. hoop stretch (solution 2).

Table 2.1: Table of parameters

Inner radius	R_i	1 cm
Outer radius	R_o	1.05 cm
Height of cylinder	H	5 cm
Initial director angle	ϕ_0	90°
Initial anisotropy parameter	r_0	2
Non-ideality parameter	α	0.3
Ogden model shear modulus	μ_1	$1.0 \cdot 10^5$ Pa
Ogden model shear modulus	μ_2	$1.904762 \cdot 10^2$ Pa
Ogden model shear modulus	μ_3	$-1.5873 \cdot 10^3$ Pa
Ogden model constant	β_1	1.3
Ogden model constant	β_2	6
Ogden model constant	β_3	-3

Consider the first solution, Figure 2.2(a)-(d). We observe that for $r_0 \neq r$, hoop stretch vs. pressure does not pass through $(1, 0)$ but through $((r/r_0)^{-1/6}, 0)$ since the change of the anisotropy parameter gives rise to a spontaneous deformation of the cylinder. The hoop stretch vs. pressure is non-monotone (Figure 2.2(a)): the pressure initially increases but then drops before increasing again with increasing hoop stretch. This reflects the well-known balloon instability: with increasing pressure, the radius increases till it reaches a critical pressure at which it jumps to a large radius. The onset and the extent of this instability is amplified in nematic elastomers due to a rotation of the director. Figure 2.2(c) shows that the director begins to rotate with inflation, reaching the hoop direction asymptotically. To understand this, an increase in radius increases the volume more than an increase in axial stretch, since the former leads to an increase of included area rather than length. Therefore, the pressure seeks to increase the circumference by reorienting the director. This reorientation also leads to a decrease of the axial stretch (Figure 2.2(b)). Consequently the axial stretch is also non-monotone: it decreases during reorientation but increases again as the director stabilizes. Finally, the reorientation leads to a twist in the cylinder (Figure 2.2(d)). The magnitude of all of these trends increases with increasing anisotropy parameter r. In particular, the critical pressure decreases and the change of hoop stretch increases with increasing anisotropy parameter r.

The reorientation, however, is resisted by the non-ideality as shown by varying the non-ideality parameter α in Figure 2.3. Note that the director rotation from the vicinity of the initial orientation $\phi_0 = \frac{\pi}{2}$ is increasingly delayed as the non-ideality

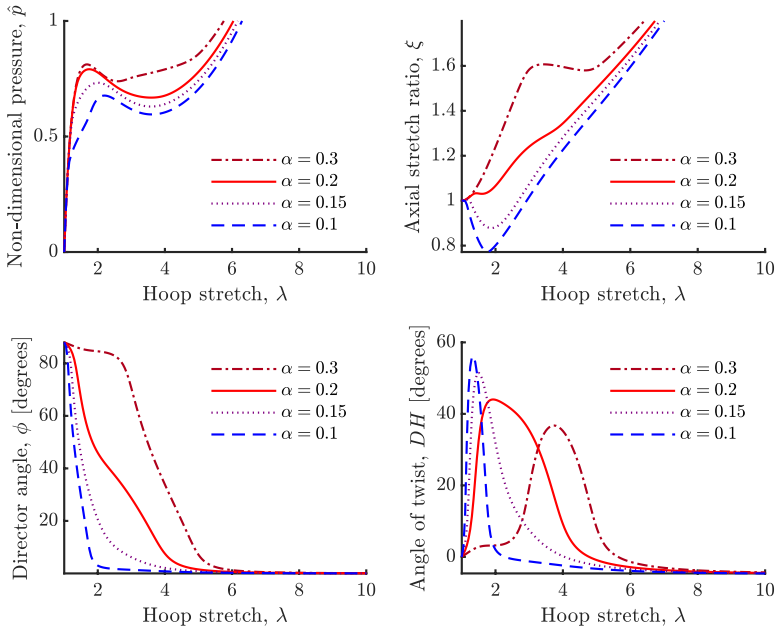


Figure 2.3: Effect of non-ideality parameter α upon pressure, axial stretch, director angle, and angle of twist for $r_0 = r = 2$.

parameter increases. The balance between the pressure-assisted reorientation and the non-ideality-mediated resistance leads to the observed behavior.

The second solution, Figure 2.2(a)-(b),(e)-(f), is very similar to the first, except that the reorientation and twist change sign. The pressure vs. hoop-stretch and the axial stretch vs. hoop-stretch curves remain unchanged. Consequently, both solutions have the same pressure vs. volume curves which are shown in Figure 2.4(a). The volume strain is plotted on a logarithmic scale due to the dramatic change of volume during the instability.

It is useful to understand the origin of the two solutions. The material is not chiral, and neither is the initial configuration. Therefore a breaking of the chiral symmetry by rotation of the director has to be accompanied by a symmetry-related counterpart. To elaborate on this, recall the invariance (2.9). Let $\boldsymbol{Q}$ be a 180° rotation about the azimuthal direction,

$$\boldsymbol{Q} = \begin{pmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{pmatrix}. \tag{2.17}$$

Note that $\boldsymbol{Q}\boldsymbol{n}_0 = -\boldsymbol{n}_0$ so that it satisfies the requirement for (2.9). It is easy to check that for $\boldsymbol{F}$ and $\boldsymbol{n}$ in (2.12) and (2.13),

$$\boldsymbol{Q}\boldsymbol{F}\boldsymbol{Q}^T = \begin{pmatrix} \frac{1}{\lambda\xi} & 0 & 0 \\ 0 & \lambda & -\lambda RD \\ 0 & 0 & \xi \end{pmatrix}, \quad \boldsymbol{Q}\boldsymbol{n} = \begin{pmatrix} 0 \\ \cos\phi \\ -\sin\phi \end{pmatrix}. \tag{2.18}$$

Thus, the invariance (2.9) implies that any solution with chirality has a symmetric counterpart with the same radial and azimuthal stretches.

The presence of the two symmetric solutions enables the formation of stripe domains that avoid overall torsion as shown in Figure 2.4(b). We divide the cylinder into short cylindrical rings and alternate between the two solutions. This leads to a continuous deformation, where one ring twists one way and the other the other way in an alternating pattern, but they meet continuously across the boundaries as indicated by the initially straight fiducial dashed line shown in the figure. The overall torsion is zero while the overall hoop and axial stretch are as before, leading to the pressure-volume curve shown in 2.4(a), where volume strain is defined as the current volume divided by the reference volume.

Stripe domains are widely observed in nematic elastomers, especially in uniaxial tension, where rigid grips prevent any shear [77]. In uniaxial tension of a nematic sheet along an axis that is perpendicular to the initial director orientation, director rotation accommodates stretch but causes shear. However, shear breaks the symmetry and therefore there are two solutions (rotation to the right or left), which alternate to form the stripe domains. The domains are fine, typically with the width of microns, and the interfaces are very sharp, with a width of nanometers. The stripe domains in Figure 2.4(b) are the exact analogs of those in uniaxial tension.

2.4 Pump

The pressure-volume curves in Figure 2.4(a) motivate the application of this cylindrical nematic elastomer balloon as a pump. Recall that the anisotropy parameter r depends on temperature, and therefore the four pressure-volume curves represent four distinct temperatures. In a typical monodomain nematic elastomer, $r = 2$ at a high temperature of about 85°C, while $r = 8$ at a low temperature of about 25°C [77]. These two pressure-volume curves are re-plotted in Figure 2.5 as the hot and cold nematic elastomers. An important observation is that the lower-critical pressure (point E) of the pressure-volume curve at the high temperature is higher than the upper-critical pressure (point B) of the pressure-volume curve at the low

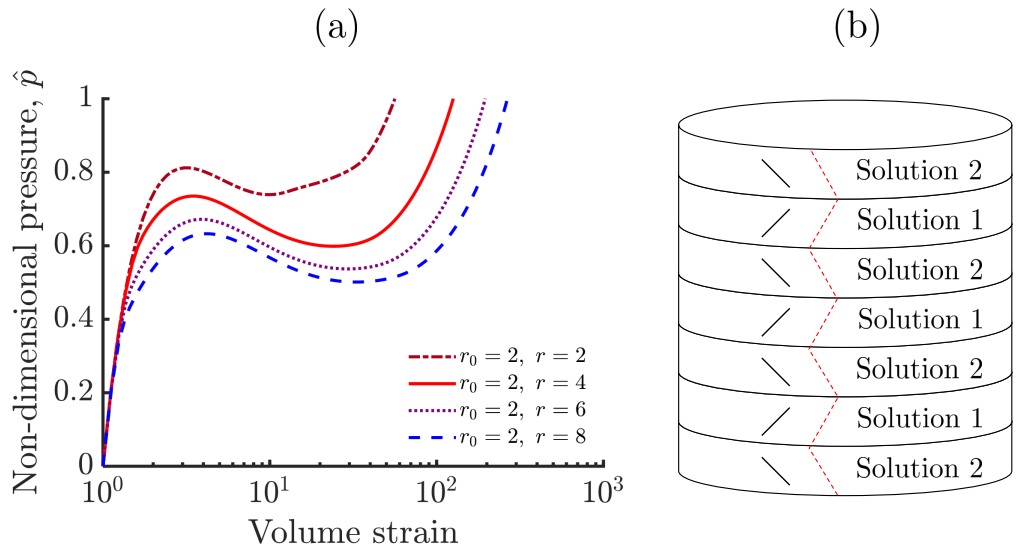


Figure 2.4: (a) Pressure-volume relation for the inflation of a nematic cylinder. (b) Formation of stripe domains that avoid any torsion—the lines describe the director, while the dashed line is the deformed shape of a fiducial line that is initially straight and axial.

temperature. This enables the operation as a pump between an inlet pressure p_i and outlet pressure p_o, where $p_B < p_i \leq p_o \leq p_E$.

Also shown in the figure are the isotherms (pressure-volume relation) of a fixed mass of fluid, in this case air, at the hot and the cold temperatures of 85°C and 25°C, respectively. These isotherms were calculated using the ideal gas law. The cold fluid isotherm is given by $p_{\text{cold}}(V) = n_C R T_{\text{cold}}/V$, where n_C is the number of moles of air at point C, $R = 8.3145$ is the ideal gas law constant, and T_{cold} is the cold fluid temperature. The hot fluid isotherm pressure is analogously given by the corresponding number of moles at point D and the hot fluid temperature: $p_{\text{hot}}(V) = n_D R T_{\text{hot}}/V$.

The pump operates as follows. Let us begin at the high temperature with the outlet closed and the inlet open so that the nematic pump is at the point A. Now, cool the pump with the inlet open so that the pressure remains at p_i. On cooling fully to the cold temperature when $r = 8$, the only equilibrium solution is point C, which has a very large volume. So, the balloon would draw in a large volume of air from A to C. We note that this process does not proceed smoothly. As the pump is cooled from the high temperature, the volume changes gradually till the temperature when the upper critical pressure (the point corresponding to B at the intermediate

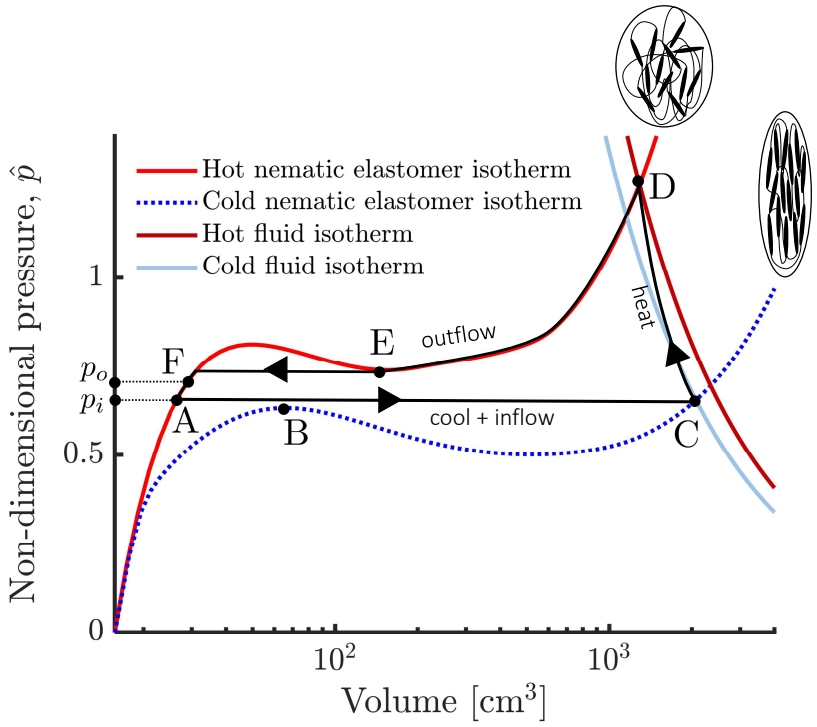


Figure 2.5: The operation of a pump with an input pressure p_i and output pressure p_o by heating and cooling a nematic cylinder.

temperature) equals p_i. At this point, there will be an instability (likely accompanied by an aneurysm), and the volume jumps to something close to C. This instability has been analyzed by Giudici and Biggins [31], and may be of interest in microfluidics. Subsequent cooling takes it to point C.

Now close the inlet and start heating the pump. The mass of fluid in the pump is fixed, and so its behavior shifts from that of the cold isotherm to that of the hot isotherm. In the interim, the pressure-volume curve of the pump also changes to that of the hot material. Therefore, the equilibrium shifts from C to D. Now, open the outlet so that the pressure decreases to p_o. The only available state in the hot pump is at F, and so the pump goes from point D, with very large volume, to point F, with small volume, expelling the fluid. This is again accompanied by an instability from E to F. Closing the outlet and opening the inlet takes us from F to A, resetting the pump.

A pump can be characterized by its ejection fraction. In this case, the ejection fraction is

$$\frac{\text{(filled volume)} - \text{(empty volume)}}{\text{filled volume}} = \frac{V_C - V_F}{V_C} = 98.6\%, \tag{2.19}$$

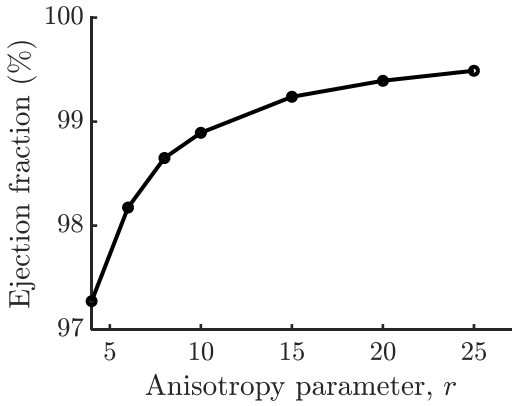


Figure 2.6: Ejection fraction as a function of anisotropy parameter, r.

which means that 98.6% of the fluid is pumped out of a filled balloon during each cycle. This is extremely high: a normal human heart has a left ventricular ejection fraction between 50% and 70%, according to the American College of Cardiology [39]. A plot of ejection fraction for fixed $r_0 = 2$ and varying r can be seen in Figure 2.6.

2.5 Conclusion

We have introduced a modified formulation of the standard Warner-Terentjev energy density incorporated into a higher-order Ogden model to more accurately describe the behavior of nematic elastomers at large deformation. Furthermore, this work has initiated the study of actuation from geometries beyond flat, two-dimensional sheets by exploring a curvilinear three-dimensional geometry. We have outlined the deformation of a nematic elastomer balloon under simple expansion and twist. The material is actuated remotely by changing the temperature to dictate the degree of anisotropy, and the response is tunable. The foundation for our understanding of nematic elastomer actuation from flat geometries has already been well established with respect to the design, optimization, manufacturing, and tuning (e.g. voxelated sheets [76], wrinkling-resistant membranes [56], and moving inchworm [81]). Future applications based on more complex geometries and loading conditions, for instance incorporation of disclination defects and gradients of director or temperature across the thickness, can build upon this framework. Finally, the actuation can be effected by light instead of temperature [79]. For example, the pump described here would function with a light-driven actuation from $r = 8$ to $r = 2$. A practical

difficulty to be overcome is that many photo-active materials have a low penetration depth and the actuation is in bending, rather than stretching.

Chapter 3

UNIVERSAL DEFORMATIONS OF NEMATIC ELASTOMERS

3.1 Introduction

The goal of this chapter is to understand how the soft elasticity of nematic elastomers affects complex inhomogeneous deformations. We have already seen in Chapter 2 that nematic elastomers can form fine-scale patterns called stripe domains. Stripe domains were first observed by Kundler and Finkelmann [42] in monodomain nematic sheets subjected to uniaxial tension in a direction normal to the original director. These domains were associated with soft elasticity. Mathematically, the Bladon-Terentjev-Warner theory [10] leads to an energy that is not convex, and its relaxation leads to fine-scale structure and soft-behavior [23]. This has been studied in homogeneous deformations (largely in uniaxial stretch [18, 19, 56], but more recently in biaxial stretch [85]), where we see different regimes depending on the imposed deformation. We expect these regimes to interact when the material is subjected to inhomogeneous deformation, and this motivates the current work.

We begin by describing the Bladon-Terentjev-Warner theory [10] and its relaxation due to Dolzmann and DeSimone [23]. The BTW theory uses a Gaussian approximation to treat the entropy of polymer chains. However, when polymer chains are subjected to large deformation, this approximation is no longer accurate as the polymer chains themselves are stretched, so we propose a generalization based on a Mooney-Rivlin energy. We then find its relaxation, which was independently done by Agosiniani and DeSimone [1]. We use this relaxed energy to study a series of problems—the spherical balloon expansion, cylindrical balloon expansion, cavitation, and bending of a beam. The key idea here is to exploit the notion of universal deformations proposed by Ericksen [25]. He showed that there are certain deformations that automatically satisfy the equations of elasticity in all isotropic incompressible elastic bodies, and this is the basis of much work in finite elasticity.

3.2 Energy

Bladon-Terentjev-Warner theory

We begin by recalling the Bladon-Terentjev-Warner (BTW) theory for ideal nematic elastomers, introduced in Chapter 2. We pick a reference configuration to be the stress-free isotropic state and lower the temperature so that the material is in a

nematic state. The BTW theory adapts the Gaussian chain model to this situation, and shows that the free energy density is given by

$$W^{BTW}(\boldsymbol{F}, \boldsymbol{n}) = \begin{cases} \frac{\mu}{2}\left[\operatorname{tr}\left(\boldsymbol{F}^{\top}\ell_n^{-1}\boldsymbol{F}\right) - 3\right] & \det \boldsymbol{F} = 1, |\boldsymbol{n}| = 1 \\ \infty & \text{else} \end{cases} \tag{3.1}$$

with step-length tensor $\ell_n = r^{-1/3}\left(\mathbf{I} + (r-1)\mathbf{n}\otimes\mathbf{n}\right)$. The anisotropy parameter r describes the mesogen ordering: $r = 1$ represents the isotropic state, and $r > 1$ represents the anisotropic nematic state. Note that $W^{BTW} = 0$ if and only if $\boldsymbol{F} = \boldsymbol{Q}\ell_0\boldsymbol{R}$, $\hat{\boldsymbol{n}} = \boldsymbol{Q}\hat{\boldsymbol{e}}_0$ for $\boldsymbol{Q}, \boldsymbol{R} \in SO(3)$, $\hat{\boldsymbol{e}}_0$ a fixed unit vector, and $\ell_0 = r^{1/3}\left(\boldsymbol{I} + (r-1)\hat{\boldsymbol{e}}_0\otimes\hat{\boldsymbol{e}}_0\right)$.

DeSimone-Dolzmann relaxation

The BTW free energy is not convex, and this leads to fine-scale microstructure. DeSimone and Dolzmann computed the relaxation [23]. To do so, minimize over the nematic director $\boldsymbol{n}$ to obtain

$$W(\boldsymbol{F}) = \min_{\boldsymbol{n}\text{ s.t. }|\boldsymbol{n}|=1} W^{BTW}(\boldsymbol{F}, \boldsymbol{n}) = \begin{cases} \frac{\mu}{2}\left[r^{1/3}\left(\frac{s^2}{r} + \frac{t^2}{s^2} + \frac{1}{t^2}\right) - 3\right] & \det \boldsymbol{F} = 1 \\ \infty & \text{else} \end{cases}, \tag{3.2}$$

where s is the largest singular value of $\boldsymbol{F}$, and t is the largest singular value of $\operatorname{cof}\boldsymbol{F}$. The relaxed energy is given by:

$$W^{qc}(\boldsymbol{F}) = \begin{cases} 0 & \boldsymbol{F} \in L \\ \frac{\mu}{2}\left(\frac{r^{1/3}}{t^2} + \frac{2t}{r^{1/6}} - 3\right) & \boldsymbol{F} \in M \\ W(\boldsymbol{F}) & \boldsymbol{F} \in S \\ \infty & \text{else} \end{cases}, \tag{3.3}$$

where L represents the liquid-like region, M represents the microstructure-formation region, and S represents the solid-like region:

$$\begin{aligned} L &: \{(s,t) : t \le s^2,\ t \ge \sqrt{s},\ t \le r^{1/6}\} \\ M &: \{(s,t) : t \ge r^{1/6},\ t \le s^2,\ t \ge r^{-1/2}s^2\} \\ S &: \{(s,t) : t \ge \sqrt{s},\ t \le r^{-1/2}s^2\} \end{aligned} \tag{3.4}$$

We can rewrite these constraints in terms of the principal stretches of $\boldsymbol{F}$ (details can be seen in the Appendix in Section A.4):

$$\begin{aligned} L &: \{\lambda_{max}\lambda_{mid} \leq r^{1/6}\} \\ M &: \{\lambda_{max}\lambda_{mid} \geq r^{1/6},\ \frac{\lambda_{max}}{\lambda_{mid}} \leq \sqrt{r}\}. \\ S &: \{\frac{\lambda_{max}}{\lambda_{mid}} \geq \sqrt{r}\} \end{aligned} \tag{3.5}$$

This describes the energy after the material has formed fine-scale microstructure. Figure 3.1 shows an illustration of the regions L, M, and S.

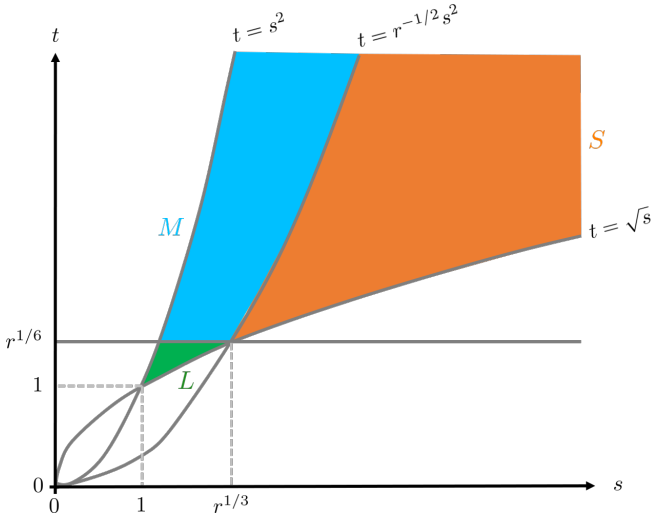


Figure 3.1: Regions of L, M, and S in the phase diagram of (s, t).

Physically, a liquid-like deformation in region L is accommodated by unstressed microstructure in which the nematic director is in three dimensions (not confined to a plane). In region M, the deformation is accommodated by stressed microstructure in which the nematic director is planar (confined to a two-dimensional plane). For example, stripe domains are examples of microstructure that form in region M. The solid-like region S corresponds to the stressed response of the polymer network without any liquid crystal molecule reorientation, for instance stretching a monodomain parallel to its nematic director.

One would expect that a nematic elastomer in region L to be opaque due to the scattering of light because the nematic directors are reorienting in 3D. A nematic elastomer undergoing planar microstructure in region M would be macroscopically opaque, but under polarized light microscopy there may be patterned features such as stripe domains detectable at the micrometer length scale. A sample in region S

would appear transparent, because the nematic mesogens would be fully aligned as a monodomain.

Generalized Mooney-Rivlin energy

This generalization and relaxation was independently proposed by Agostiniani and DeSimone [1]. Note that the trace formula is simply an extension of the neo-Hookean model for rubbers. This can be seen clearly when the energy is rewritten as such:

$$W^{BTW}(F) = \frac{\mu}{2}\left[\operatorname{tr}\left(\tilde{\boldsymbol{F}}^{\top}\tilde{\boldsymbol{F}}\right) - 3\right], \tag{3.6}$$

where

$$\tilde{\boldsymbol{F}} = \ell_n^{-1/2}\boldsymbol{F}. \tag{3.7}$$

Because the trace formula still relies on Gaussian chain modeling, it cannot model the stress build-up at large stretches. We will work to construct an energy density that is based on a generalized Mooney-Rivlin model.

Because a nematic elastomer is isotropic, our free energy density needs to be a function of the three invariants of the left Cauchy-Green tensor $\boldsymbol{b}$:

$$W = f(I_1, I_2, I_3). \tag{3.8}$$

The three invariants of the second-order tensor $\boldsymbol{b} = \boldsymbol{F}\boldsymbol{F}^{\top}$ are

$$I_1 = \operatorname{tr}\boldsymbol{b} = \operatorname{tr}\left(\boldsymbol{F}\boldsymbol{F}^{\top}\right) \tag{3.9}$$

$$I_2 = \frac{1}{2}\left[(\operatorname{tr}\boldsymbol{b})^2 - \operatorname{tr}\left(\boldsymbol{b}^2\right)\right] \tag{3.10}$$

$$I_3 = \det\boldsymbol{b} = \det\left(\boldsymbol{F}\boldsymbol{F}^{\top}\right) = (\det\boldsymbol{F})^2. \tag{3.11}$$

Due to incompressibility of the material, the third invariant $I_3 = 1$, so

$$W = g(I_1, I_2). \tag{3.12}$$

We notice that the first invariant is a function of $\boldsymbol{F}$ and the second invariant is a function of cof $\boldsymbol{F}$:

$$I_1 = \operatorname{tr}\boldsymbol{b} = \operatorname{tr}\left(\boldsymbol{F}\boldsymbol{F}^{\top}\right) \tag{3.13}$$

$$I_2 = \operatorname{tr}\boldsymbol{b}^{-1} = \operatorname{tr}(\operatorname{cof}\boldsymbol{b}) = \operatorname{tr}\left[(\operatorname{cof}\boldsymbol{F})\,(\operatorname{cof}\boldsymbol{F})^{\top}\right]. \tag{3.14}$$

Thus, we begin by looking at an energy based on the first invariant. For a rubber with deformation gradient $\boldsymbol{F}$,

$$W_{I_1}(\boldsymbol{F}) = c|I_1 - 3|^p \tag{3.15}$$

$$= c|\operatorname{tr}(\boldsymbol{F}\boldsymbol{F}^{\top}) - 3|^p. \tag{3.16}$$

For a nematic elastomer with deformation gradient $\boldsymbol{F}$,

$$W_{I_1}(\tilde{\boldsymbol{F}}) = W_{I_1}(\ell_n^{-1/2}\boldsymbol{F}) \tag{3.17}$$

$$= c\left|\mathrm{tr}\left(\ell_n^{-1}\boldsymbol{b}\right) - 3\right|^p. \tag{3.18}$$

After minimizing the energy over the nematic director $\boldsymbol{n}$, we obtain:

$$\min_{\boldsymbol{n}\text{ s.t. }|\boldsymbol{n}|=1} W_{I_1}\left(\ell_n^{-1/2}\boldsymbol{F}\right) = c\left|r^{1/3}\left(\frac{s^2}{r}+\frac{t^2}{s^2}+\frac{1}{t^2}\right) - 3\right|^p = c\left|W_1(s,t)\right|^p. \tag{3.19}$$

Details can be seen in the Appendix in Section A.2. Similarly, the energy based on the second invariant is as follows for rubbers:

$$W_{I_2}(\boldsymbol{F}) = d|I_2 - 3|^q \tag{3.20}$$

$$= d\left|\mathrm{tr}\left[(\mathrm{cof}\,\boldsymbol{F})\,(\mathrm{cof}\,\boldsymbol{F})^\top\right] - 3\right|^q. \tag{3.21}$$

For a nematic elastomer with deformation gradient $\boldsymbol{F}$,

$$W_{I_2}(\tilde{\boldsymbol{F}}) = W_{I_2}(\ell_n^{-1/2}\boldsymbol{F}) \tag{3.22}$$

$$= d\left|\mathrm{tr}\left[\left(\ell_n^{-1}\boldsymbol{b}\right)^{-1}\right] - 3\right|^q. \tag{3.23}$$

After minimizing the energy over the nematic director $\boldsymbol{n}$, we obtain:

$$\min_{\boldsymbol{n}\text{ s.t. }|\boldsymbol{n}|=1} W_{I_2}\left(\ell_n^{-1/2}\boldsymbol{F}\right) = d\left|r^{-1/3}\left(\frac{r}{s^2}+\frac{s^2}{t^2}+t^2\right) - 3\right|^q = d\left|W_2(s,t)\right|^q. \tag{3.24}$$

Details can be seen in the Appendix in Section A.3. Thus, we can create a generalized Mooney-Rivlin energy for an isotropic, incompressible nematic elastomer based on $W_1(s,t)$ and $W_2(s,t)$ as follows:

$$W(s,t) = \begin{cases} \displaystyle\sum_{i=1}^{M} c_i\left|r^{1/3}\left(\frac{s^2}{r}+\frac{t^2}{s^2}+\frac{1}{t^2}\right) - 3\right|^{p_i} & \\ \displaystyle+\sum_{j=1}^{N} d_j\left|r^{-1/3}\left(\frac{r}{s^2}+\frac{s^2}{t^2}+t^2\right) - 3\right|^{q_j} & \det\boldsymbol{F} = 1 \\ \infty & \text{else} \end{cases}, \tag{3.25}$$

where s is the largest singular value of $\boldsymbol{F}$, t is the largest singular value of cof $\boldsymbol{F}$, and c_i $(i = 1:M)$ and d_j $(j = 1:N)$ are constants.

Relaxation of the generalized Mooney-Rivlin energy

Based on previous work done in this field [1, 65], the free energy density in Equation 3.25 is not convex in s or t.

The relaxed form of this energy is:

$$W^{qc}(s,t) = \sum_{i=1}^{M} c_i \left(|W_1|^{p_i}\right)^{qc} + \sum_{j=1}^{N} d_j \left(|W_2|^{q_j}\right)^{qc}, \tag{3.26}$$

where

$$\left(|W_1|^{p_i}\right)^{qc}(s,t) = \begin{cases} 0 & \boldsymbol{F} \in L \\ \left|\frac{r^{1/3}}{t^2} + \frac{2t}{r^{1/6}} - 3\right|^{p_i} & \boldsymbol{F} \in M \\ |W_1|^{p_i} & \boldsymbol{F} \in S \\ \infty & \text{else} \end{cases} \tag{3.27}$$

and

$$\left(|W_2|^{q_j}\right)^{qc}(s,t) = \begin{cases} 0 & \boldsymbol{F} \in L \\ \left|r^{-1/3}t^2 + \frac{2r^{1/6}}{t} - 3\right|^{q_j} & \boldsymbol{F} \in M \\ |W_2|^{q_j} & \boldsymbol{F} \in S \\ \infty & \text{else} \end{cases}. \tag{3.28}$$

The regions L, M, and S are given by Equation 3.5, and the restrictions upon the exponents $p_i \geq 1$ $(i = 1 : M)$ and $q_j \geq 1$ $(j = 1 : N)$ must be satisfied.

3.3 Stress

Based on its principal values and directions, the left Cauchy-Green tensor is $\boldsymbol{b} = \sum_{i=1}^{3} \lambda_i^2 \hat{\boldsymbol{v}}_i \otimes \hat{\boldsymbol{v}}_i$. If the strain energy density of a material can be written in the form $W = W(\lambda_i)$, where $\lambda_1 \geq \lambda_2 \geq \lambda_3$ are the principal stretches, then the Cauchy stress of an incompressible, isotropic hyperelastic body is

$$\boldsymbol{\sigma} = -\eta \boldsymbol{I} + \sum_{i=1}^{3} \lambda_i \frac{\partial W}{\partial \lambda_i} \hat{\boldsymbol{v}}_i \otimes \hat{\boldsymbol{v}}_i. \tag{3.29}$$

Here, the vectors $\hat{\boldsymbol{v}}_i$ are normalized eigenvectors. We can derive the stresses for the relaxed generalized Mooney-Rivlin energy of Equation 3.26 in all three regions: $W = W^{qc}(s,t)$. In region L, the stresses are

$$\boldsymbol{\sigma}^L = -\eta^L \boldsymbol{I}. \tag{3.30}$$

In region M, $W^{qc}(s,t) = \sum_{i=1}^{M} c_i |A_M|^{p_i} + \sum_{j=1}^{N} d_j |B_M|^{q_j}$, where

$$A_M = r^{1/3}\lambda_3^2 + \frac{2\lambda_1\lambda_2}{r^{1/6}} - 3 \tag{3.31}$$

$$B_M = r^{-1/3}\lambda_1^2\lambda_2^2 + 2\lambda_3 r^{1/6} - 3. \tag{3.32}$$

The principal Cauchy stresses are

$$\boldsymbol{\sigma}^M = -\eta^M \boldsymbol{I} + \sum_{i=1}^{3} \lambda_i \frac{\partial W^{qc}}{\partial \lambda_i} \hat{\boldsymbol{v}}_i \otimes \hat{\boldsymbol{v}}_i, \tag{3.33}$$

where

$$\lambda_1 \frac{\partial W^{qc}}{\partial \lambda_1} = \sum_{i=1}^{M} c_i p_i |A_M|^{p_i - 1} 2r^{-1/6}\lambda_1\lambda_2 + \sum_{j=1}^{N} d_j q_j |B_M|^{q_j - 1} 2r^{-1/3}\lambda_1^2\lambda_2^2 \tag{3.34}$$

$$\lambda_2 \frac{\partial W^{qc}}{\partial \lambda_2} = \lambda_1 \frac{\partial W^{qc}}{\partial \lambda_1} \tag{3.35}$$

$$\lambda_3 \frac{\partial W^{qc}}{\partial \lambda_3} = \sum_{i=1}^{M} c_i p_i |A_M|^{p_i - 1} 2r^{1/3}\lambda_3^2 + \sum_{j=1}^{N} d_j q_j |B_M|^{q_j - 1} 2r^{1/6}\lambda_3. \tag{3.36}$$

In region S, $W^{qc}(s,t) = \sum_{i=1}^{M} c_i |A_S|^{p_i} + \sum_{j=1}^{N} d_j |B_S|^{q_j}$, where

$$A_S = r^{1/3}\lambda_3^2 + r^{1/3}\lambda_2^2 + r^{-2/3}\lambda_1^2 - 3 \tag{3.37}$$

$$B_S = r^{-1/3}\lambda_1^2\lambda_2^2 + r^{-1/3}\lambda_1^2\lambda_3^2 + r^{2/3}\lambda_2^2\lambda_3^2 - 3. \tag{3.38}$$

The principal Cauchy stresses are

$$\boldsymbol{\sigma}^S = -\eta^S \boldsymbol{I} + \sum_{i=1}^{3} \lambda_i \frac{\partial W^{qc}}{\partial \lambda_i} \hat{\boldsymbol{v}}_i \otimes \hat{\boldsymbol{v}}_i, \tag{3.39}$$

where

$$\lambda_1 \frac{\partial W^{qc}}{\partial \lambda_1} = \sum_{i=1}^{M} c_i p_i |A_S|^{p_i - 1} 2r^{-2/3}\lambda_1^2 + \sum_{j=1}^{N} d_j q_j |B_S|^{q_j - 1} 2r^{-1/3}\lambda_1^2(\lambda_2^2 + \lambda_3^2) \tag{3.40}$$

$$\lambda_2 \frac{\partial W^{qc}}{\partial \lambda_2} = \sum_{i=1}^{M} c_i p_i |A_S|^{p_i - 1} 2r^{1/3}\lambda_2^2 + \sum_{j=1}^{N} d_j q_j |B_S|^{q_j - 1} 2\lambda_2^2(r^{-1/3}\lambda_1^2 + r^{2/3}\lambda_3^2) \tag{3.41}$$

$$\lambda_3 \frac{\partial W^{qc}}{\partial \lambda_3} = \sum_{i=1}^{M} c_i p_i |A_S|^{p_i - 1} 2r^{1/3}\lambda_3^2 + \sum_{j=1}^{N} d_j q_j |B_S|^{q_j - 1} 2\lambda_3^2(r^{-1/3}\lambda_1^2 + r^{2/3}\lambda_2^2). \tag{3.42}$$

3.4 Ericksen's "universal deformations"

J. L. Ericksen established the problem of determining all deformations which can be produced in every isotropic, incompressible, hyperelastic body by the application of surface tractions alone (no body forces) [25]. Because these universal relations are independent of constitutive relation, they are a powerful tool in continuum mechanics.

Below are the solution families known thus far for a point (x, y, z), (ρ, θ, z), or (ρ, θ, ϕ) with respective material point (X, Y, Z), (R, Θ, Z), or (R, Θ, Φ), and constant a, b, c, d, e, f:

Family 0: Homogeneous deformations (for spatial point $\boldsymbol{x}$ with material point $\boldsymbol{X}$ and deformation gradient $\boldsymbol{F}$, constant vector $\boldsymbol{c}$)

$$\boldsymbol{x} = \boldsymbol{F}\boldsymbol{X} + \boldsymbol{c} \tag{3.43}$$

Family 1: Bending, stretching, and shearing of a rectangular block

$$\rho = \sqrt{2aX}, \theta = bY, z = \frac{Z}{ab} - bcY \tag{3.44}$$

Family 2: Straightening, stretching, and shearing of a sector of a tube

$$x = \frac{1}{2}ab^2R^2, y = \frac{\Theta}{ab}, z = \frac{Z}{b} - \frac{c\Theta}{ab} \tag{3.45}$$

Family 3: Inflation, bending, torsion, extension, and shearing of an annular wedge, with $a(cf - de) = 1$

$$\rho = \sqrt{aR^2 + b}, \theta = c\Theta + dZ, z = e\Theta + fZ \tag{3.46}$$

Family 4: Inflation or eversion of a sector of a spherical shell

$$\rho = [\pm R^3 + a]^{1/3}, \theta = \pm\Theta, \phi = \Phi \tag{3.47}$$

Family 5: Inflation, bending, extension, and azimuthal shearing of an annular wedge [38, 66]

$$\rho = a^{1/2}R, \theta = d\ln(bR) + c\Theta, z = eZ,\ ace = 1 \tag{3.48}$$

In this work, we address three of the families, as applied to nematic elastomers: the bending of a block is a Family 1 deformation, the inflation of a cylindrical balloon is a Family 3 deformation, and the inflation of a spherical balloon and the cavitation of a disk are classified as Family 4 deformations.

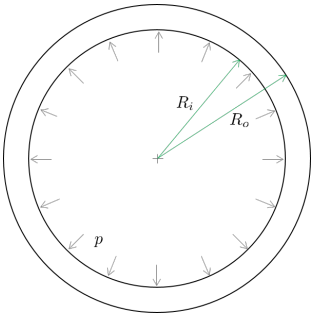


Figure 3.2: Cross-section of spherical balloon.

3.5 Spherical balloon

Deformation of spherical balloon expansion

We are interested in the deformation of a balloon, which can be modeled as a spherical shell subjected to an internal pressure. In the undeformed configuration, the balloon has inner radius R_i and outer radius R_o. The internal pressure is denoted by p. The spherical balloon is assumed to remain spherical throughout the deformation. The undeformed sphere has radial coordinate $R \in [R_i, R_o]$, azimuthal angle $\Theta \in [0, 2\pi)$, and polar angle $\Phi \in [0, \pi)$, while the deformed sphere has coordinate system $\rho \in [\rho_i, \rho_o]$, $\theta \in [0, 2\pi)$, and $\phi \in [0, \pi)$. Following Ericksen, we make the ansatz

$$\begin{cases} \rho = \rho(R) \\ \theta = \Theta \\ \phi = \Phi \end{cases}. \tag{3.49}$$

The deformation gradient in spherical coordinates is

$$\boldsymbol{F} = \begin{pmatrix} \frac{d\rho}{dR} & & \\ & \frac{\rho}{R} & \\ & & \frac{\rho}{R} \end{pmatrix}. \tag{3.50}$$

With incompressibility, $\det \boldsymbol{F} = 1$, we obtain this first-order differential equation

$$\frac{d\rho}{dR}\frac{\rho^2}{R^2} = 1. \tag{3.51}$$

Solving for the deformed radius ρ as a function of the undeformed radius R:

$$\rho = \left(R^3 + c\right)^{1/3}, \tag{3.52}$$

where c is a constant. Let λ denote the azimuthal stretch, $\lambda = \frac{\rho}{R}$, and let λ_o denote the azimuthal stretch at the outer radius, $\lambda_o = \frac{\rho(R_o)}{R_o}$. Then,

$$\lambda_o = \frac{\left(R_o^3 + c\right)^{1/3}}{R_o}. \tag{3.53}$$

Solving for c and plugging c back into Equation 3.52 yields

$$\rho = \left(R^3 + R_o^3\left(\lambda_o^3 - 1\right)\right)^{1/3}, \tag{3.54}$$

and the azimuthal stretch is

$$\lambda = \left(1 + \left(\frac{R_o}{R}\right)^3\left(\lambda_o^3 - 1\right)\right)^{1/3}. \tag{3.55}$$

Thus, the deformation gradient is

$$\boldsymbol{F} = \begin{pmatrix} \frac{1}{\lambda^2} & & \\ & \lambda & \\ & & \lambda \end{pmatrix}, \tag{3.56}$$

and the left Cauchy-Green tensor, $\boldsymbol{b} = \boldsymbol{F}\boldsymbol{F}^\top$ is

$$\boldsymbol{b} = \begin{pmatrix} \frac{1}{\lambda^4} & & \\ & \lambda^2 & \\ & & \lambda^2 \end{pmatrix}. \tag{3.57}$$

The principal stretches are $\lambda_1 = \lambda_2 = \lambda$ (corresponding to $\boldsymbol{e}_\theta$ and $\boldsymbol{e}_\phi$) and $\lambda_3 = \frac{1}{\lambda^2}$ (corresponding to $\boldsymbol{e}_\rho$). Thus, $s = \lambda$ and $t = \lambda^2$. The regions are:

$$L : \{R \geq R^*\} \tag{3.58}$$

$$M : \{R \leq R^*, r \geq 1\} \tag{3.59}$$

$$S : \{r \leq 1\}, \tag{3.60}$$

where

$$R^* = R_o\left(\frac{\lambda_o^3 - 1}{r^{1/4} - 1}\right)^{1/3}. \tag{3.61}$$

This leads to the possibility of three cases:

- Case 1: $r > 1$ and $R^* \leq R_i$: the entire balloon is in region L
- Case 2: $r > 1$ and $R_i < R^* < R_o$: the inner region of the balloon $R \in [R_i, R^*]$ is in M, and the outer region $R \in [R^*, R_o]$ is in L

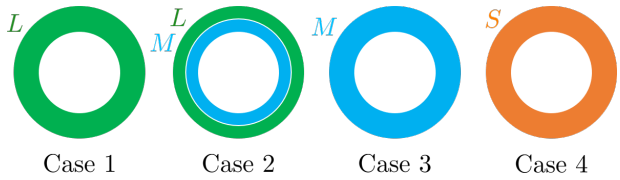


Figure 3.3: Diagram of all possible cases in the inflation of a nematic elastomer spherical balloon.

- Case 3: $r > 1$ and $R^* \geq R_o$: the entire balloon is in region M
- Case 4: $r = 1$ and the entire balloon is in region S

A diagram illustrating the various cases can be seen in Figure 3.3.

Note that Figure 3.1 can provide us with insight into the regions that this deformation will experience. The spherical balloon expansion is merely equibiaxial stretch (see the deformation gradient of Equation 3.56). Recalling that s is the largest singular value of $\boldsymbol{F}$ and t is the largest singular value of cof $\boldsymbol{F}$, this means that $t = s^2$ for this deformation. Following along the $t = s^2$ curve in Figure 3.1, we see that the deformation will move progressively through region L then M, never touching S for $r > 1$.

Stress

The expressions for stress are as follows: In region L,

$$\boldsymbol{\sigma}^L = -\eta^L \boldsymbol{I}. \tag{3.62}$$

In region M, the non-zero components of the stress are:

$$\begin{aligned}
\sigma^M_{\rho\rho} &= -\eta^M + \sum_{i=1}^{M} c_i p_i |A_M|^{p_i - 1} \frac{2r^{1/3}}{\lambda^4} + \sum_{j=1}^{N} d_j q_j |B_M|^{q_j - 1} \frac{2r^{1/6}}{\lambda^2} \\
\sigma^M_{\theta\theta} &= -\eta^M + \sum_{i=1}^{M} c_i p_i |A_M|^{p_i - 1} \frac{2\lambda^2}{r^{1/6}} + \sum_{j=1}^{N} d_j q_j |B_M|^{q_j - 1} 2r^{-1/3}\lambda^4 \\
\sigma^M_{\phi\phi} &= \sigma^M_{\theta\theta},
\end{aligned} \tag{3.63}$$

where

$$\begin{aligned}
A_M &= \frac{r^{1/3}}{\lambda^4} + \frac{2\lambda^2}{r^{1/6}} - 3 \\
B_M &= r^{-1/3}\lambda^4 + \frac{2r^{1/6}}{\lambda^2} - 3.
\end{aligned} \tag{3.64}$$

In region S, $r = 1$ and the non-zero components of the stress are

$$\begin{aligned}
\sigma^S_{\rho\rho} &= -\eta^S + \sum_{i=1}^{M} c_i p_i |A_S|^{p_i - 1} \frac{2}{\lambda^4} + \sum_{j=1}^{N} d_j q_j |B_S|^{q_j - 1} \frac{4}{\lambda^2} \\
\sigma^S_{\theta\theta} &= -\eta^S + \sum_{i=1}^{M} c_i p_i |A_S|^{p_i - 1} (2\lambda^2) + \sum_{j=1}^{N} d_j q_j |B_S|^{q_j - 1} (2\lambda^2) \left(\lambda^2 + \frac{1}{\lambda^4} \right) \\
\sigma^S_{\phi\phi} &= \sigma^S_{\theta\theta},
\end{aligned} \tag{3.65}$$

where

$$\begin{aligned}
A_S &= \frac{1}{\lambda^4} + 2\lambda^2 - 3 \\
B_S &= \lambda^4 + \frac{2}{\lambda^2} - 3.
\end{aligned} \tag{3.66}$$

Solving static equilibrium

The static equilibrium equations (in the absence of body forces) in spherical coordinates are

$$\rho : \frac{\partial \sigma_{\rho\rho}}{\partial \rho} + 2\frac{\sigma_{\rho\rho}}{\rho} + \frac{1}{\rho}\frac{\partial \sigma_{\phi\rho}}{\partial \phi} + \frac{\cot\phi}{\rho}\sigma_{\phi\rho} + \frac{1}{\rho \sin\phi}\frac{\partial \sigma_{\theta\rho}}{\partial \theta} - \frac{1}{\rho}\left(\sigma_{\theta\theta} + \sigma_{\phi\phi}\right) = 0 \tag{3.67}$$

$$\theta : \frac{\partial \sigma_{\rho\theta}}{\partial \rho} + 2\frac{\sigma_{\rho\theta}}{\rho} + \frac{1}{\rho}\frac{\partial \sigma_{\phi\theta}}{\partial \phi} + \frac{1}{\rho \sin\phi}\frac{\partial \sigma_{\theta\theta}}{\partial \theta} + \frac{\sigma_{\theta\rho}}{\rho} + \frac{\cot\phi}{\rho}\left(\sigma_{\phi\theta} + \sigma_{\theta\phi}\right) = 0 \tag{3.68}$$

$$\phi : \frac{\partial \sigma_{\rho\phi}}{\partial \rho} + 2\frac{\sigma_{\rho\phi}}{\rho} + \frac{1}{\rho}\frac{\partial \sigma_{\phi\phi}}{\partial \phi} + \frac{1}{\rho \sin\phi}\frac{\partial \sigma_{\theta\phi}}{\partial \theta} + \frac{\sigma_{\phi\rho}}{\rho} + \frac{\cot\phi}{\rho}\left(\sigma_{\phi\phi} - \sigma_{\theta\theta}\right) = 0, \tag{3.69}$$

and the boundary conditions for this problem are as follows:

$$\sigma_{\rho\rho}|_{\rho=\rho_i} = -p \tag{3.70}$$

$$\sigma_{\rho\rho}|_{\rho=\rho_o} = 0. \tag{3.71}$$

In all cases, solving the ϕ equation yields $\eta = \eta(\rho, \theta)$. Similarly, solving the θ equation, we find that $\eta = \eta(\rho)$. Finally, to solve the ρ equation, we will rewrite the stress expressions as

$$\sigma_{\rho\rho} = -\eta + \hat{\sigma}_{\rho\rho} \tag{3.72}$$

$$\sigma_{\theta\theta} = \sigma_{\phi\phi} = -\eta + \hat{\sigma}_{\theta\theta}, \tag{3.73}$$

and so the boundary conditions can be rewritten as follows

$$-\eta(\rho = \rho_i) + \hat{\sigma}_{\rho\rho}(\rho = \rho_i) = -p \tag{3.74}$$

$$-\eta(\rho = \rho_o) + \hat{\sigma}_{\rho\rho}(\rho = \rho_o) = 0. \tag{3.75}$$

Case 1

In Case 1, the entire balloon is in region L. The two boundary conditions for this case are:

$$-\eta^L(\rho = \rho_i) + \hat{\sigma}^L_{\rho\rho}(\rho = \rho_i) = -p \tag{3.76}$$

$$-\eta^L(\rho = \rho_o) + \hat{\sigma}^L_{\rho\rho}(\rho = \rho_o) = 0. \tag{3.77}$$

The stress in this region is found in Equation 3.62. Thus, the ρ equilibrium equation yields

$$\frac{d\sigma^L_{\rho\rho}}{d\rho} + 2\frac{\sigma^L_{\rho\rho}}{\rho} - \frac{1}{\rho}\left(\sigma^L_{\theta\theta} + \sigma^L_{\phi\phi}\right) = 0 \tag{3.78}$$

$$\frac{d(-\eta^L + \hat{\sigma}^L_{\rho\rho})}{d\rho} + 2\frac{(-\eta^L + \hat{\sigma}^L_{\rho\rho})}{\rho} - \frac{1}{\rho}\left(-\eta^L + \hat{\sigma}^L_{\theta\theta} - \eta^L + \hat{\sigma}^L_{\phi\phi}\right) = 0 \tag{3.79}$$

$$\int_{\rho_i}^{\rho_o} d\left(-\eta^L + \hat{\sigma}^L_{\rho\rho}\right) = \int_{\rho_i}^{\rho_o} 0 d\rho \tag{3.80}$$

$$p = 0 \tag{3.81}$$

Thus, for Case 1, the inner pressure is:

$$p = 0. \tag{3.82}$$

Case 2

In Case 2, $R \in [R_i, R^*]$ is in region M, and the $R \in [R^*, R_o]$ is in L. The two boundary conditions can be rewritten specific to the region:

$$-\eta^M(\rho = \rho_i) + \hat{\sigma}^M_{\rho\rho}(\rho = \rho_i) = -p \tag{3.83}$$

$$-\eta^L(\rho = \rho_o) + \hat{\sigma}^L_{\rho\rho}(\rho = \rho_o) = 0, \tag{3.84}$$

and there is an additional boundary condition for continuity between the two regions:

$$-\eta^L(\rho = \rho^*) + \hat{\sigma}^L_{\rho\rho}(\rho = \rho^*) = -\eta^M(\rho = \rho^*) + \hat{\sigma}^M_{\rho\rho}(\rho = \rho^*). \tag{3.85}$$

First, we solve the equilibrium equations in region L. Again, the stress in region L is found in Equation 3.62. The ρ equation yields:

$$\frac{d\sigma^L_{\rho\rho}}{d\rho} + 2\frac{\sigma^L_{\rho\rho}}{\rho} - \frac{1}{\rho}\left(\sigma^L_{\theta\theta} + \sigma^L_{\phi\phi}\right) = 0 \tag{3.86}$$

$$\frac{d(-\eta^L + \hat{\sigma}^L_{\rho\rho})}{d\rho} + 2\frac{(-\eta^L + \hat{\sigma}^L_{\rho\rho})}{\rho} - \frac{1}{\rho}\left(-\eta^L + \hat{\sigma}^L_{\theta\theta} - \eta^L + \hat{\sigma}^L_{\phi\phi}\right) = 0 \tag{3.87}$$

$$\frac{d\left(-\eta^L + \hat{\sigma}^L_{\rho\rho}\right)}{d\rho} = 0 \tag{3.88}$$

$$\left(-\eta^L + \hat{\sigma}^L_{\rho\rho}\right)\Big|^{\rho_o}_{\rho^*} = 0 \tag{3.89}$$

$$\left(-\eta^L(\rho_o) + \hat{\sigma}^L_{\rho\rho}(\rho_o)\right) - \left(-\eta^L(\rho^*) + \hat{\sigma}^L_{\rho\rho}(\rho^*)\right) = 0 \tag{3.90}$$

$$-\eta^L(\rho^*) + \hat{\sigma}^L_{\rho\rho}(\rho^*) = 0 \tag{3.91}$$

In region M, the stress is that of Eqns. 3.63 and 3.64. The ρ equation yields:

$$\frac{d\sigma^M_{\rho\rho}}{d\rho} + 2\frac{\sigma^M_{\rho\rho}}{\rho} - \frac{1}{\rho}\left(\sigma^M_{\theta\theta} + \sigma^M_{\phi\phi}\right) = 0 \tag{3.92}$$

$$\frac{d(-\eta^M + \hat{\sigma}^M_{\rho\rho})}{d\rho} + \frac{1}{\rho}\left(2\hat{\sigma}^M_{\rho\rho} - \hat{\sigma}^M_{\theta\theta} - \hat{\sigma}^M_{\phi\phi}\right) = 0 \tag{3.93}$$

$$\int_{\rho_i}^{\rho^*} d\left(-\eta^M + \hat{\sigma}^M_{\rho\rho}\right) = -\int_{\rho_i}^{\rho^*} \frac{2}{\rho}\left(\hat{\sigma}^M_{\rho\rho} - \hat{\sigma}^M_{\theta\theta}\right) d\rho \tag{3.94}$$

$$\left(-\eta^M(\rho^*) + \hat{\sigma}^M_{\rho\rho}(\rho^*)\right) - \left(-\eta^M(\rho_i) + \hat{\sigma}^M_{\rho\rho}(\rho_i)\right) = \int_{\rho_i}^{\rho^*} \frac{2}{\rho}\left(\hat{\sigma}^M_{\theta\theta} - \hat{\sigma}^M_{\rho\rho}\right) d\rho \tag{3.95}$$

$$p = \int_{\rho_i}^{\rho^*} \frac{2}{\rho}\left(\hat{\sigma}^M_{\theta\theta} - \hat{\sigma}^M_{\rho\rho}\right) d\rho. \tag{3.96}$$

The right-hand side requires a change of integration variable from ρ to R. Noting that $d\rho = \frac{1}{\lambda(R)^2} dR$, we obtain an expression for the inner pressure:

$$p = \int_{R_i}^{R^*} \frac{2}{\lambda(R)^3 R}\left[\hat{\sigma}^M_{\theta\theta}(\rho = \lambda(R)R) - \hat{\sigma}^M_{\rho\rho}(\rho = \lambda(R)R)\right] dR. \tag{3.97}$$

Case 3

In Case 3, all of the balloon is in region M. The two boundary conditions for this case are:

$$-\eta^M(\rho = \rho_i) + \hat{\sigma}^M_{\rho\rho}(\rho = \rho_i) = -p \tag{3.98}$$

$$-\eta^M(\rho = \rho_o) + \hat{\sigma}^M_{\rho\rho}(\rho = \rho_o) = 0. \tag{3.99}$$

The stress in region M can be found in Eqns. 3.63 and 3.64. The ρ equilibrium equation yields:

$$\frac{d\sigma^M_{\rho\rho}}{d\rho} + 2\frac{\sigma^M_{\rho\rho}}{\rho} - \frac{1}{\rho}\left(\sigma^M_{\theta\theta} + \sigma^M_{\phi\phi}\right) = 0 \tag{3.100}$$

$$\frac{d(-\eta^M + \hat{\sigma}^M_{\rho\rho})}{d\rho} + \frac{1}{\rho}\left(2\hat{\sigma}^M_{\rho\rho} - \hat{\sigma}^M_{\theta\theta} - \hat{\sigma}^M_{\phi\phi}\right) = 0 \tag{3.101}$$

$$\int_{\rho_i}^{\rho_o} d\left(-\eta^M + \hat{\sigma}^M_{\rho\rho}\right) = -\int_{\rho_i}^{\rho_o} \frac{2}{\rho}\left(\hat{\sigma}^M_{\rho\rho} - \hat{\sigma}^M_{\theta\theta}\right) d\rho \tag{3.102}$$

$$\left(-\eta^M(\rho_o) + \hat{\sigma}^M_{\rho\rho}(\rho_o)\right) - \left(-\eta^M(\rho_i) + \hat{\sigma}^M_{\rho\rho}(\rho_i)\right) = \int_{\rho_i}^{\rho_o} \frac{2}{\rho}\left(\hat{\sigma}^M_{\theta\theta} - \hat{\sigma}^M_{\rho\rho}\right) d\rho \tag{3.103}$$

$$p = \int_{\rho_i}^{\rho_o} \frac{2}{\rho}\left(\hat{\sigma}^M_{\theta\theta} - \hat{\sigma}^M_{\rho\rho}\right) d\rho. \tag{3.104}$$

As in Case 2, we employ a change of integration variable from ρ to R and obtain the inner pressure as:

$$p = \int_{R_i}^{R_o} \frac{2}{\lambda(R)^3 R}\left[\hat{\sigma}^M_{\theta\theta}(\rho = \lambda(R)R) - \hat{\sigma}^M_{\rho\rho}(\rho = \lambda(R)R)\right] dR. \tag{3.105}$$

Case 4

In Case 4, all of the balloon is in region S. Note that only the isotropic state ($r = 1$) falls under this case. The two boundary conditions specific to this case are

$$-\eta^S(\rho = \rho_i) + \hat{\sigma}^S_{\rho\rho}(\rho = \rho_i) = -p \tag{3.106}$$

$$-\eta^S(\rho = \rho_o) + \hat{\sigma}^S_{\rho\rho}(\rho = \rho_o) = 0. \tag{3.107}$$

The stress in region S can be found in Eqns. 3.65 and 3.66. The ρ equilibrium equation yields:

$$\frac{d\sigma^M_{\rho\rho}}{d\rho} + 2\frac{\sigma^M_{\rho\rho}}{\rho} - \frac{1}{\rho}\left(\sigma^M_{\theta\theta} + \sigma^M_{\phi\phi}\right) = 0 \tag{3.108}$$

$$\frac{d(-\eta^M + \hat{\sigma}^M_{\rho\rho})}{d\rho} + \frac{1}{\rho}\left(2\hat{\sigma}^M_{\rho\rho} - \hat{\sigma}^M_{\theta\theta} - \hat{\sigma}^M_{\phi\phi}\right) = 0 \tag{3.109}$$

$$\int_{\rho_i}^{\rho_o} d\left(-\eta^M + \hat{\sigma}^M_{\rho\rho}\right) = -\int_{\rho_i}^{\rho_o} \frac{2}{\rho}\left(\hat{\sigma}^M_{\rho\rho} - \hat{\sigma}^M_{\theta\theta}\right) d\rho \tag{3.110}$$

$$\left(-\eta^M(\rho_o) + \hat{\sigma}^M_{\rho\rho}(\rho_o)\right) - \left(-\eta^M(\rho_i) + \hat{\sigma}^M_{\rho\rho}(\rho_i)\right) = \int_{\rho_i}^{\rho_o} \frac{2}{\rho}\left(\hat{\sigma}^M_{\theta\theta} - \hat{\sigma}^M_{\rho\rho}\right) d\rho \tag{3.111}$$

$$p = \int_{\rho_i}^{\rho_o} \frac{2}{\rho}\left(\hat{\sigma}^M_{\theta\theta} - \hat{\sigma}^M_{\rho\rho}\right) d\rho. \tag{3.112}$$

As in Case 2, we employ a change of integration variable from ρ to R and obtain the inner pressure as:

$$p = \int_{R_i}^{R_o} \frac{2}{\lambda(R)^3 R}\left[\hat{\sigma}^M_{\theta\theta}(\rho = \lambda(R)R) - \hat{\sigma}^M_{\rho\rho}(\rho = \lambda(R)R)\right] dR. \tag{3.113}$$

Results

The calculations were performed in `MATLAB`. The solutions are plotted in Figures 3.4–3.5b with the inner radius $R_i = 1$ cm and outer radius $R_o = 1.1$ cm. The following parameters for the generalized Mooney-Rivlin model were used: $M = 2$, $N = 1$, $c_1 = 1.0 \cdot 10^5$ Pa, $c_2 = 1.90 \cdot 10^2$ Pa, $d_1 = 1.59 \cdot 10^{-2}$ Pa, $p_1 = 1.3$, $p_2 = 5$, and $q_1 = 2$. This yields an effective shear modulus $\mu = \frac{1}{2}\left(\sum_{i=1}^{N} c_i p_i + \sum_{j=1}^{M} d_j q_j\right) = 6.52 \cdot 10^4$ Pa.

The pressure p, normalized by the effective shear modulus μ, is plotted as a function of the azimuthal stretch at the outer radius λ_o. Figure 3.4 shows the comparison between the generalized Mooney-Rivlin model of this work and the BTW (neo-Hookean) model of previous work in the field [23] for an anisotropy parameter of $r = 8$. The neo-Hookean model is unable to capture the correct effect of elasticity in the material at very high values of stretch (e.g. for λ_o greater than about 6). As seen in experiments of rubber balloons [73], the balloon experiences a subsequent

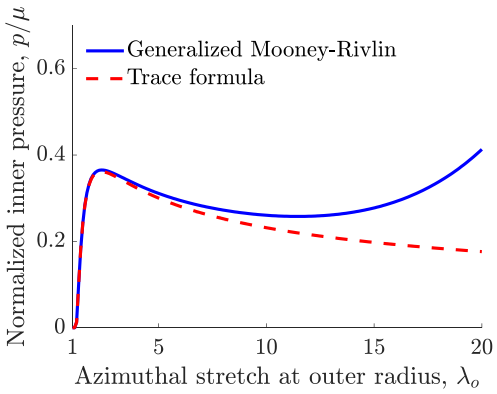
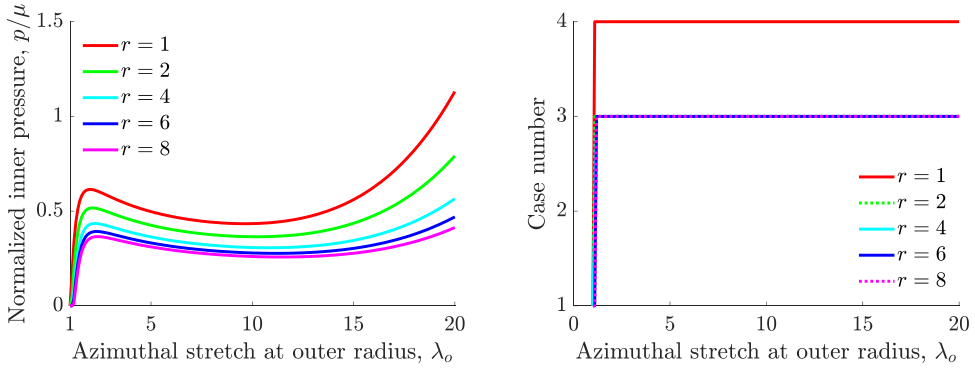


Figure 3.4: Comparison of this work's generalized Mooney-Rivlin model with the trace formula model of [23] for inflation pressure of a spherical balloon.

stiffening due to the further stretching of the polymer, or "effects of the limited extensibility of the network", as indicated by the increase in pressure at high stretch, which is correctly captured by our generalized Mooney-Rivlin model.



(a) (b)

Figure 3.5: Spherical balloon results: (a) Pressure-stretch curves at varying anisotropy parameter. (b) Progression of the spherical balloon solution through individual case numbers.

Figure 3.5a shows the results for varying anisotropy parameter r. The response of the balloon is stiffest in the isotropic state ($r = 1$), and gets correspondingly softer as r increases, as expected. As the pressure p increases, the balloon undergoes deformation according to the various cases, as seen in Figure 3.5b. For $r = 1$ (rubber), the balloon is in Case 1 (region L) at $\lambda_o = 1$ when there is no deformation, but for the rest of the deformation the balloon is entirely in region S, corresponding to a solid-like response with no microstructure formation, since the material is a rubber with no liquid crystals. For $r > 1$, the solutions move progressively from Case 1

to 2 to 3 throughout the deformation. At a stretch of $\lambda_o = 1$, the balloon is again in Case 1, corresponding to being entirely in the L region because no deformation has occurred. Immediately upon inflation of the balloon, the balloon jumps to Case 2, where the inner part of the balloon experiences region M, developing fine-scale microstructure. Then shortly after, the balloon will become entirely in region M, and the rest of the balloon will develop fine-scale microstructure in response to the pressure. This formation of microstructure by the liquid crystal molecules creates a softer response than the rubber without liquid crystals.

3.6 Cylindrical balloon

Deformation of cylindrical balloon expansion

We are interested in the deformation of a cylindrical shell composed of the nematic elastomer, capped off at both ends, subjected to an internal pressure. The shell has undeformed height H, inner radius R_i, and undeformed outer radius R_o. Note that there is no thin-wall approximation used in this formulation. The internal pressure is denoted by p, as seen in Figure 3.6a. The cylindrical shell is assumed to deform uniformly, i.e. the cylinder remains a cylinder throughout the deformation. The undeformed cylinder has coordinate system R, Θ, and Z, while the deformed cylinder has coordinate system ρ, θ, and z, as seen in Figure 3.6b. The liquid crystals are in an isotropic reference state and free to move throughout the deformation. The nematic elastomer is assumed to be incompressible.

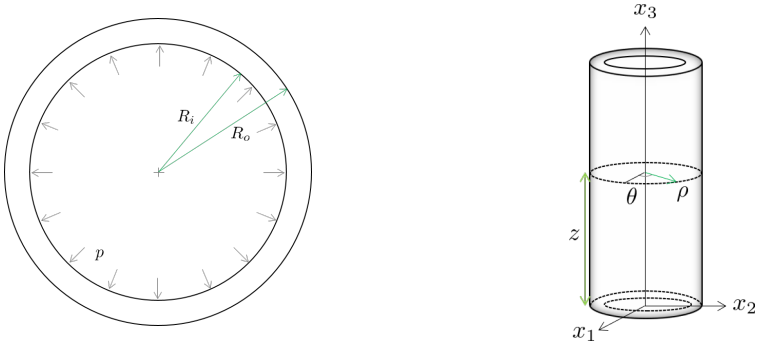


Figure 3.6: (a) Cross-section of cylindrical shell. (b) Schematic showing cylindrical coordinates.

We assume that the cylinder remains a cylinder throughout the deformation, so the mapping that describes this progression is $\varphi = \rho \mathbf{e}_\rho + z\mathbf{e}_z$. Following Ericksen, we

look for a solution of the form

$$\begin{cases} \rho = \rho(R) \\ \theta = \Theta \\ z = \xi Z \end{cases}. \tag{3.114}$$

Following the convention found in the paper by Rudykh *et al.* on electroactive balloons [63], let $\lambda = r/R$ be the hoop stretch ratio. Let ξ denote the axial stretch ratio. Thus the associated deformation gradient is:

$$\mathbf{F} = \begin{pmatrix} \frac{1}{\lambda\xi} & 0 & 0 \\ 0 & \lambda & \\ 0 & 0 & \xi \end{pmatrix}, \tag{3.115}$$

where

$$\lambda = \sqrt{\frac{1}{\xi} + \frac{R_o^2}{R^2}\left(\lambda_o^2 - \frac{1}{\xi}\right)}. \tag{3.116}$$

For fixed axial ratio $\xi = 1$, the principal stretches $\lambda_1 \geq \lambda_2 \geq \lambda_3$ are $\lambda_1 = \lambda$, corresponding to $\hat{\boldsymbol{e}}_\theta$, $\lambda_2 = 1$, corresponding to $\hat{\boldsymbol{e}}_z$, and $\lambda_3 = \frac{1}{\lambda}$, corresponding to $\hat{\boldsymbol{e}}_\rho$. Simplifying the regions, we find:

$$L : R \geq R_2 \tag{3.117}$$

$$M : R_1 \leq R \leq R_2 \tag{3.118}$$

$$S : R \leq R_1, \tag{3.119}$$

where $R_1 = \sqrt{\frac{R_o^2(\lambda_o^2-1)}{r-1}}$ and $R_2 = \sqrt{\frac{R_o^2(\lambda_o^2-1)}{r^{1/3}-1}}$.

This leads to the possibility of the following cases:

- Case 0: $r > 1$ and $R_1 \leq R_2 \leq R_i$: the entire balloon is in region L
- Case 1: $r > 1$ and $R_1 \leq R_i < R_2 < R_o$: the inner portion of the balloon is in region M and the outer portion is in L
- Case 2: $r > 1$ and $R_1 \leq R_i < R_o \leq R_2$: the entire balloon is in region M
- Case 3: $r > 1$ and $R_i < R_1 < R_o \leq R_2$: the inner portion of the balloon is in region S and the outer portion is in M
- Case 4: $r > 1$ and $R_i < R_1 < R_2 < R_o$: the balloon is in regions S, then M, then L from inside to outside
- Case 5: $r \geq 1$ and $R_o \leq R_1 \leq R_2$: the entire balloon is in region S

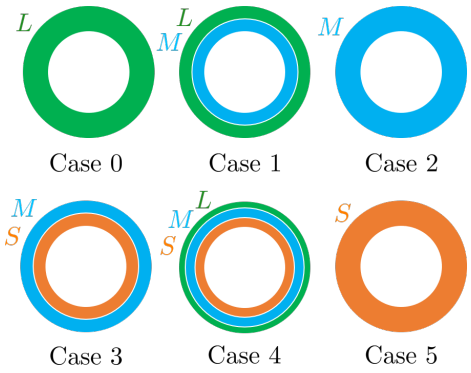


Figure 3.7: Diagram of all possible cases in the inflation of a nematic elastomer cylindrical balloon.

A diagram illustrating the various cases can be seen in Figure 3.7.

As in the spherical balloon deformation, we can gain insight from Figure 3.1 regarding the regions that this deformation will experience. The cylindrical balloon expansion with fixed axial ration $\xi = 1$ is the same as a planar extension deformation (the deformation gradient of Equation 3.115 is $\boldsymbol{F} = \mathrm{diag}(1/\lambda, \lambda, 1)$). Recalling that s is the largest singular value of $\boldsymbol{F}$ and t is the largest singular value of cof $\boldsymbol{F}$, this means that $t = s$ for this deformation. If we were to follow along the $t = s$ line in Figure 3.1, we see that the deformation will move progressively through region L then M then S.

Stress

The stress in region L is

$$\boldsymbol{\sigma}^L = -\eta^L \boldsymbol{I}. \tag{3.120}$$

The non-zero components of stress in region M are

$$\begin{aligned}
\sigma^M_{\rho\rho} &= -\eta^M + \sum_{i=1}^{M} c_i p_i |A_M|^{p_i - 1} 2\frac{r^{1/3}}{\lambda^2} + \sum_{j=1}^{N} d_j q_j |B_M|^{q_j - 1} \frac{2r^{1/6}}{\lambda} \\
\sigma^M_{\theta\theta} &= -\eta^M + \sum_{i=1}^{M} c_i p_i |A_M|^{p_i - 1} \frac{2\lambda}{r^{1/6}} + \sum_{j=1}^{N} d_j q_j |B_M|^{q_j - 1} 2r^{-1/3}\lambda^2 \\
\sigma^M_{zz} &= \sigma^M_{\theta\theta},
\end{aligned} \tag{3.121}$$

where

$$\begin{aligned}
A_M &= \frac{r^{1/3}}{\lambda^2} + \frac{2\lambda}{r^{1/6}} - 3 \\
B_M &= r^{-1/3}\lambda^2 + \frac{2r^{1/3}}{\lambda} - 3.
\end{aligned} \tag{3.122}$$

The non-zero components of stress in region S are

$$\begin{aligned}
\sigma^S_{\rho\rho} &= -\eta^S + \sum_{i=1}^{M} c_i p_i |A_S|^{p_i-1} 2\frac{r^{1/3}}{\lambda^2} + \sum_{j=1}^{N} d_j q_j |B_S|^{q_j-1} \frac{2}{\lambda^2}(r^{-1/3}\lambda^2 + r^{2/3}) \\
\sigma^S_{\theta\theta} &= -\eta^S + \sum_{i=1}^{M} c_i p_i |A_S|^{p_i-1} 2\lambda^2 r^{-2/3} + \sum_{j=1}^{N} d_j q_j |B_S|^{q_j-1} (2r^{-1/3}\lambda^2) \left(1 + \frac{1}{\lambda^2}\right) \\
\sigma^S_{zz} &= -\eta^S + \sum_{i=1}^{M} c_i p_i |A_S|^{p_i-1} 2r^{1/3} + \sum_{j=1}^{N} d_j q_j |B_S|^{q_j-1} \left(2r^{-1/3}\lambda^2 + 2\frac{r^{2/3}}{\lambda^2}\right),
\end{aligned} \tag{3.123}$$

where

$$\begin{aligned}
A_S &= \frac{r^{1/3}}{\lambda^2} + r^{1/3} + r^{-2/3}\lambda^2 - 3 \\
B_S &= r^{-1/3}\lambda^2 + r^{-1/3} + \frac{r^{2/3}}{\lambda^2} - 3.
\end{aligned} \tag{3.124}$$

Solving static equilibrium

Static equilibrium (in the absence of body forces) is obtained when div $\boldsymbol{\sigma} = \mathbf{0}$. In cylindrical coordinates and for a symmetric tensor $\boldsymbol{\sigma}$,

$$\begin{aligned}
\frac{\partial \sigma_{\rho\rho}}{\partial \rho} + \frac{1}{\rho}\frac{\partial \sigma_{\rho\theta}}{\partial \theta} + \frac{\sigma_{\rho\rho} - \sigma_{\theta\theta}}{\rho} + \frac{\partial \sigma_{\rho z}}{\partial z} &= 0, \\
\frac{\partial \sigma_{\rho\theta}}{\partial \rho} + \frac{1}{\rho}\frac{\partial \sigma_{\theta\theta}}{\partial \theta} + \frac{2\sigma_{\rho\theta}}{\rho} + \frac{\partial \sigma_{\theta z}}{\partial z} &= 0, \\
\frac{\partial \sigma_{\rho z}}{\partial \rho} + \frac{1}{\rho}\frac{\partial \sigma_{\theta z}}{\partial \theta} + \frac{\sigma_{\rho z}}{\rho} + \frac{\partial \sigma_{zz}}{\partial z} &= 0.
\end{aligned} \tag{3.125}$$

The boundary conditions for this problem are as follows:

$$\sigma_{\rho\rho}|_{\rho=\rho_i} = -p \tag{3.126}$$

$$\sigma_{\rho\rho}|_{\rho=\rho_o} = 0. \tag{3.127}$$

For every case, the θ and z equilibrium equations yield that $\eta = \eta(\rho)$. This leaves only the ρ equilibrium equation to be solved. We will rewrite the stress expressions as

$$\sigma_{\rho\rho} = -\eta + \hat{\sigma}_{\rho\rho} \tag{3.128}$$

$$\sigma_{\theta\theta} = -\eta + \hat{\sigma}_{\theta\theta}. \tag{3.129}$$

Case 0

In Case 0, the entire balloon is in region L. The two boundary conditions for this case are:

$$-\eta^L(\rho = \rho_i) + \hat{\sigma}^L_{\rho\rho}(\rho = \rho_i) = -p \tag{3.130}$$

$$-\eta^L(\rho = \rho_o) + \hat{\sigma}^L_{\rho\rho}(\rho = \rho_o) = 0. \tag{3.131}$$

The ρ equilibrium equation yields

$$\frac{d\sigma^L_{\rho\rho}}{d\rho} + \frac{1}{\rho}\left(\sigma^L_{\rho\rho} - \sigma^L_{\theta\theta}\right) = 0 \tag{3.132}$$

$$\frac{d(-\eta^L + \hat{\sigma}^L_{\rho\rho})}{d\rho} + \frac{1}{\rho}\left(\hat{\sigma}^L_{\rho\rho} - \hat{\sigma}^L_{\theta\theta}\right) = 0 \tag{3.133}$$

$$\int_{\rho_i}^{\rho_o} d\left(-\eta^L + \hat{\sigma}^L_{\rho\rho}\right) = \int_{\rho_i}^{\rho_o} 0 d\rho \tag{3.134}$$

$$p = 0, \tag{3.135}$$

and thus, for Case 1, the inner pressure is:

$$p = 0. \tag{3.136}$$

Case 1

In Case 1, $R \in [R_i, R_2]$ is in region M, and the $R \in [R_2, R_o]$ is in L. The two boundary conditions can be rewritten specific to the region:

$$-\eta^M(\rho = \rho_i) + \hat{\sigma}^M_{\rho\rho}(\rho = \rho_i) = -p \tag{3.137}$$

$$-\eta^L(\rho = \rho_o) + \hat{\sigma}^L_{\rho\rho}(\rho = \rho_o) = 0, \tag{3.138}$$

and there is an additional boundary condition for continuity between the two regions:

$$-\eta^L(\rho = \rho_2) + \hat{\sigma}^L_{\rho\rho}(\rho = \rho_2) = -\eta^M(\rho = \rho_2) + \hat{\sigma}^M_{\rho\rho}(\rho = \rho_2). \tag{3.139}$$

First, we solve the equilibrium equations in region L. The ρ equation yields:

$$\frac{d\sigma^L_{\rho\rho}}{d\rho} + \frac{1}{\rho}\left(\sigma^L_{\rho\rho} - \sigma^L_{\theta\theta}\right) = 0 \tag{3.140}$$

$$\frac{d(-\eta^L + \hat{\sigma}^L_{\rho\rho})}{d\rho} = 0 \tag{3.141}$$

$$\int_{\rho_2}^{\rho_o} d\left(-\eta^L + \hat{\sigma}^L_{\rho\rho}\right) = \int_{\rho_2}^{\rho_o} 0 d\rho \tag{3.142}$$

$$\left(-\eta^L + \hat{\sigma}^L_{\rho\rho}\right)|^{\rho_o}_{\rho_2} = 0 \tag{3.143}$$

$$\eta^M(\rho_2) - \hat{\sigma}^M_{\rho\rho}(\rho_2) = 0. \tag{3.144}$$

In region M, the ρ equation yields:

$$\frac{d\sigma^M_{\rho\rho}}{d\rho} + \frac{1}{\rho}\left(\sigma^M_{\rho\rho} - \sigma^M_{\theta\theta}\right) = 0 \tag{3.145}$$

$$\frac{d(-\eta^M + \hat{\sigma}^M_{\rho\rho})}{d\rho} + \frac{1}{\rho}\left(\hat{\sigma}^M_{\rho\rho} - \hat{\sigma}^M_{\theta\theta}\right) = 0 \tag{3.146}$$

$$\int_{\rho_i}^{\rho_2} d\left(-\eta^M + \hat{\sigma}^M_{\rho\rho}\right) = \int_{\rho_i}^{\rho_2} \frac{1}{\rho}\left(\hat{\sigma}^M_{\theta\theta} - \hat{\sigma}^M_{\rho\rho}\right) d\rho \tag{3.147}$$

$$p = \int_{\rho_i}^{\rho_2} \frac{1}{\rho}\left(\hat{\sigma}^M_{\theta\theta} - \hat{\sigma}^M_{\rho\rho}\right) d\rho. \tag{3.148}$$

The right-hand side requires a change of integration variable from ρ to R. Noting that $d\rho = \frac{1}{\lambda(R)^2} dR$, we obtain an expression for the inner pressure:

$$p = \int_{R_i}^{R_2} \frac{1}{\lambda(R)^3 R}\left[\hat{\sigma}^M_{\theta\theta}(\rho = \lambda(R)R) - \hat{\sigma}^M_{\rho\rho}(\rho = \lambda(R)R)\right] dR. \tag{3.149}$$

Case 2

In Case 2, $R \in [R_i, R_o]$ is in region M. The boundary conditions are:

$$-\eta(\rho_i) + \hat{\sigma}^M_{\rho\rho}(\rho_i) = -p \tag{3.150}$$

$$-\eta(\rho_o) + \hat{\sigma}^M_{\rho\rho}(\rho_o) = 0. \tag{3.151}$$

Similar to the previous cases, the ρ equilibrium equation yields the pressure:

$$p = \int_{R_i}^{R_o} \frac{1}{\lambda(R)^3 R}\left[\hat{\sigma}^M_{\theta\theta}(\rho = \lambda(R)R) - \hat{\sigma}^M_{\rho\rho}(\rho = \lambda(R)R)\right] dR. \tag{3.152}$$

Case 3

In Case 3, $R \in [R_i, R_1]$ is in region S and $R \in [R_1, R_o]$ is in region M. The boundary conditions are as follows:

$$-\eta^M(\rho = \rho_o) + \hat{\sigma}^M_{\rho\rho}(\rho = \rho_o) = 0 \tag{3.153}$$

$$-\eta^S(\rho = \rho_i) + \hat{\sigma}^S_{\rho\rho}(\rho = \rho_i) = -p, \tag{3.154}$$

and there is an additional boundary condition for continuity between the two regions:

$$-\eta^M(\rho = \rho_1) + \hat{\sigma}^M_{\rho\rho}(\rho = \rho_1) = -\eta^S(\rho = \rho_1) + \hat{\sigma}^S_{\rho\rho}(\rho = \rho_1). \tag{3.155}$$

In region M, the ρ equilibrium equation yields

$$\eta^S(\rho_1) - \hat{\sigma}^S_{\rho\rho}(\rho_1) = \text{int}_1 \tag{3.156}$$

$$= \int_{R_1}^{R_o} \frac{1}{\lambda(R)^3 R}\left[\hat{\sigma}^M_{\theta\theta}(\rho = \lambda(R)R) - \hat{\sigma}^M_{\rho\rho}(\rho = \lambda(R)R)\right] dR. \tag{3.157}$$

In region S, we have

$$\int_{\rho_i}^{\rho_1} d\left(-\eta^S + \hat{\sigma}^S_{\rho\rho}\right) = \int_{\rho_i}^{\rho_1} \frac{1}{\rho}\left(\hat{\sigma}^S_{\theta\theta} - \hat{\sigma}^S_{\rho\rho}\right) d\rho \tag{3.158}$$

$$-\eta^S(\rho_1) + \hat{\sigma}^S_{\rho\rho}(\rho_1) - \hat{\sigma}^S_{\rho\rho}(\rho_i) + \eta^S(\rho_i) = \int_{\rho_i}^{\rho_1} \frac{1}{\rho}\left(\hat{\sigma}^S_{\theta\theta} - \hat{\sigma}^S_{\rho\rho}\right) d\rho \tag{3.159}$$

$$-\text{int}_1 + p = \text{int}_2, \tag{3.160}$$

where

$$\text{int}_2 = \hat{\sigma}^S_{\rho\rho}(\rho_1) = \int_{R_i}^{R_1} \frac{1}{\lambda(R)^3 R}\left[\hat{\sigma}^S_{\theta\theta}(\rho = \lambda(R)R) - \hat{\sigma}^S_{\rho\rho}(\rho = \lambda(R)R)\right] dR. \tag{3.161}$$

This yields the result

$$p = \text{int}_1 + \text{int}_2, \tag{3.162}$$

or

$$\begin{aligned} p = &\int_{R_1}^{R_o} \frac{1}{\lambda(R)^3 R}\left[\hat{\sigma}^M_{\theta\theta}(\rho = \lambda(R)R) - \hat{\sigma}^M_{\rho\rho}(\rho = \lambda(R)R)\right] dR \\ &+ \int_{R_i}^{R_1} \frac{1}{\lambda(R)^3 R}\left[\hat{\sigma}^S_{\theta\theta}(\rho = \lambda(R)R) - \hat{\sigma}^S_{\rho\rho}(\rho = \lambda(R)R)\right] dR. \end{aligned} \tag{3.163}$$

Case 4

In Case 4, $R \in [R_i, R_1]$ is in region S, $R \in [R_1, R_2]$ is in region M, and $R \in [R_2, R_o]$ is in region L. The boundary conditions are

$$-\eta^L(\rho = \rho_o) + \hat{\sigma}^L_{\rho\rho}(\rho = \rho_o) = 0 \tag{3.164}$$

$$-\eta^S(\rho = \rho_i) + \hat{\sigma}^S_{\rho\rho}(\rho = \rho_i) = -p, \tag{3.165}$$

and there are additional boundary conditions for continuity between the regions:

$$-\eta^M(\rho = \rho_1) + \hat{\sigma}^M_{\rho\rho}(\rho = \rho_1) = -\eta^S(\rho = \rho_1) + \hat{\sigma}^S_{\rho\rho}(\rho = \rho_1) \tag{3.166}$$

$$-\eta^M(\rho = \rho_2) + \hat{\sigma}^M_{\rho\rho}(\rho = \rho_2) = -\eta^L(\rho = \rho_2) + \hat{\sigma}^L_{\rho\rho}(\rho = \rho_2). \tag{3.167}$$

Region L yields the result

$$\eta^L(\rho_2) - \hat{\sigma}^L_{\rho\rho}(\rho_2) = 0. \tag{3.168}$$

Region M yields

$$-\eta^M(\rho_2) + \hat{\sigma}^M_{\rho\rho}(\rho_2) - \hat{\sigma}^M_{\rho\rho}(\rho_1) + \eta^M(\rho_1) = \int_{\rho_1}^{\rho_2} \frac{1}{\rho}\left(\hat{\sigma}^M_{\theta\theta} - \hat{\sigma}^M_{\rho\rho}\right) d\rho \tag{3.169}$$

$$-\eta^L(\rho_2) + \hat{\sigma}^L_{\rho\rho}(\rho_2) - \hat{\sigma}^S_{\rho\rho}(\rho_1) + \eta^S(\rho_1) = \int_{\rho_1}^{\rho_2} \frac{1}{\rho}\left(\hat{\sigma}^M_{\theta\theta} - \hat{\sigma}^M_{\rho\rho}\right) d\rho \tag{3.170}$$

$$\eta^S(\rho_1) - \hat{\sigma}^S_{\rho\rho}(\rho_1) = \int_{\rho_1}^{\rho_2} \frac{1}{\rho}\left(\hat{\sigma}^M_{\theta\theta} - \hat{\sigma}^M_{\rho\rho}\right) d\rho, \tag{3.171}$$

so

$$\eta^S(\rho_1) - \hat{\sigma}^S_{\rho\rho}(\rho_1) = \text{int}_1 \tag{3.172}$$

$$= \int_{R_1}^{R_2} \frac{1}{\lambda(R)^3 R}\left[\hat{\sigma}^M_{\theta\theta}(\rho = \lambda(R)R) - \hat{\sigma}^M_{\rho\rho}(\rho = \lambda(R)R)\right] dR. \tag{3.173}$$

Region S yields

$$-\eta^S(\rho_1) + \hat{\sigma}^S_{\rho\rho}(\rho_1) - \hat{\sigma}^S_{\rho\rho}(\rho_i) + \eta^S(\rho_i) = \int_{\rho_i}^{\rho_1} \frac{1}{\rho}\left(\hat{\sigma}^M_{\theta\theta} - \hat{\sigma}^M_{\rho\rho}\right) d\rho \tag{3.174}$$

$$-\text{int}_1 + p = \int_{\rho_i}^{\rho_1} \frac{1}{\rho}\left(\hat{\sigma}^M_{\theta\theta} - \hat{\sigma}^M_{\rho\rho}\right) d\rho \tag{3.175}$$

$$-\text{int}_1 + p = \text{int}_2, \tag{3.176}$$

where

$$\text{int}_2 = \int_{R_i}^{R_1} \frac{1}{\lambda(R)^3 R}\left[\hat{\sigma}^S_{\theta\theta}(\rho = \lambda(R)R) - \hat{\sigma}^S_{\rho\rho}(\rho = \lambda(R)R)\right] dR. \tag{3.177}$$

Thus,

$$p = \text{int}_1 + \text{int}_2, \tag{3.178}$$

or

$$\begin{aligned} p = &\int_{R_1}^{R_2} \frac{1}{\lambda(R)^3 R}\left[\hat{\sigma}^M_{\theta\theta}(\rho = \lambda(R)R) - \hat{\sigma}^M_{\rho\rho}(\rho = \lambda(R)R)\right] dR \\ &+ \int_{R_i}^{R_1} \frac{1}{\lambda(R)^3 R}\left[\hat{\sigma}^S_{\theta\theta}(\rho = \lambda(R)R) - \hat{\sigma}^S_{\rho\rho}(\rho = \lambda(R)R)\right] dR. \end{aligned} \tag{3.179}$$

Case 5

In Case 5, $R \in [R_i, R_o]$ is in region S. The boundary conditions are as follows:

$$-\eta^S(\rho = \rho_o) + \hat{\sigma}^S_{\rho\rho}(\rho = \rho_o) = 0 \tag{3.180}$$

$$-\eta^S(\rho = \rho_i) + \hat{\sigma}^S_{\rho\rho}(\rho = \rho_i) = -p. \tag{3.181}$$

The ρ equilibrium equation yields the result

$$p = \int_{R_i}^{R_o} \frac{1}{\lambda(R)^3 R}\left[\hat{\sigma}^S_{\theta\theta}(\rho = \lambda(R)R) - \hat{\sigma}^S_{\rho\rho}(\rho = \lambda(R)R)\right] dR. \tag{3.182}$$

Results

The calculations were performed in `MATLAB` with inner radius $R_i = 1$ cm and outer radius $R_o = 1.1$ cm. The parameters that were used for the generalized Mooney-Rivlin model are the same as for the spherical balloon except with $p_2 = 5$, which yields an effective shear modulus $\mu = \frac{1}{2}\left(\sum_{i=1}^{N} c_i p_1 + \sum_{j=1}^{M} d_j q_j\right) = 6.55 \cdot 10^4$ Pa. The results for the inflation a cylindrical balloon with fixed axial stretch can be seen in Figures 3.8–3.9b.

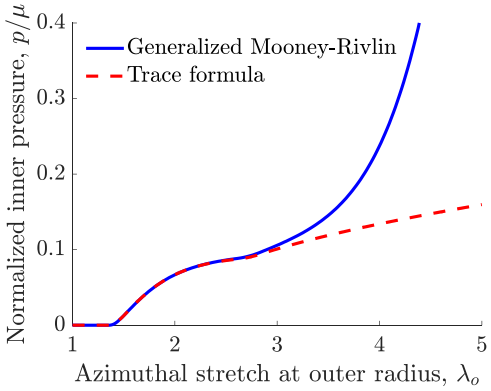
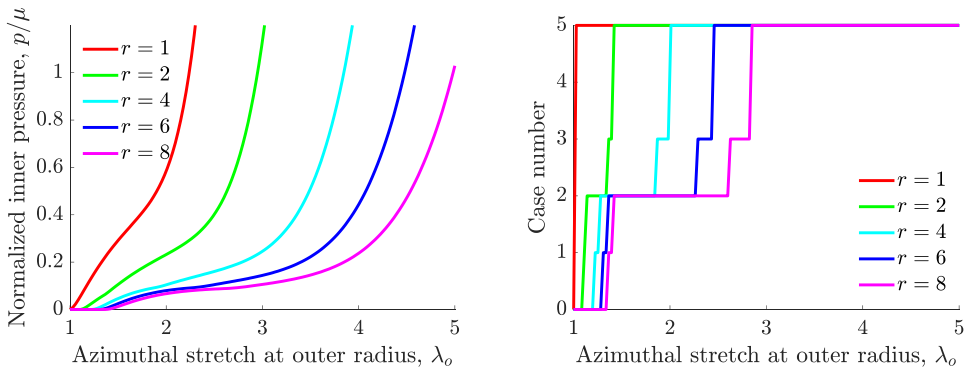


Figure 3.8: Comparison of this work's generalized Mooney-Rivlin model with the trace formula model of [23] for inflation pressure of a cylindrical balloon.

Figure 3.8 compares the results from the generalized Mooney-Rivlin model with that of the BTW theory (which features a neo-Hookean energy structure). Similarly to the spherical balloon, the BTW model is unable to capture the correct effect of elasticity in the material at very high values of stretch, so the two curves deviate starting around a stretch of $\lambda_o \approx 3$. This work's Mooney-Rivlin model captures the physics of the rubber extension at high stretch correctly.

Figure 3.9a and 3.9b shows the results for the balloon inflation at varying anisotropy parameter r. The response of the balloon is stiffest in the isotropic state ($r = 1$), and gets correspondingly softer as r increases, as expected. We also note that for $r = 1$, the deformation of the balloon passes through $(\lambda_o, p/\mu) = (1, 0)$ in its undeformed state and then the pressure begins immediately rising upon λ_o becoming greater than 1. However, for $r > 1$, the fact that the balloon moves through Case 0 for finite values of λ_o means that the balloon starts inflating at points that are not at $(\lambda_o, p/\mu) = (1, 0)$, as mentioned in Chapter 2. This is due to the fact that there is a spontaneous deformation associated with the nematic state, because the liquid crystal molecules spontaneously orient along a preferred direction and



(a) (b)

Figure 3.9: Cylindrical balloon results: (a) Results for cylindrical balloon expansion at varying anisotropy parameter. (b) Progression of the cylindrical balloon solution through individual case numbers.

the surrounding polymer network deforms accordingly, with a stretch along that preferred direction and contraction in the transverse axes. Macroscopically, this means that the cylindrical balloon experiences finite stretch with zero stress.

Figure 3.9b shows the solution's progression through Cases 0 through 5 throughout the deformation. For $r = 1$ (rubber), the balloon is in Case 0 (corresponding to region L) at $\lambda_o = 1$ because there is no deformation, but after that point the balloon is entirely in Case 5 (region S), corresponding to a solid-like response without microstructure formation. For $r > 1$, the balloons begin in Case 0 (entirely in region L). Then the balloons will develop an inner portion that lies in region M (Case 1), corresponding to fine-scale microstructure formation in the inner part of the balloon, due to the boundary condition that there is internal pressure at the inner radius of the balloon. Then the balloon will then become entirely in region M (Case 2) before the inner portion of the balloon will develop a solid-like response because it will be in region S (Case 3). Finally, the balloon will become entirely in region S, and the balloon will have a purely rubber response (Case 5). Note that we do not encounter the case in which L, M, and S are all present within the balloon (Case 4) in these calculations.

3.7 Cavitation

Deformation

The original experiments of cavitation by Gent and Lindley demonstrated that when rubber cylinders bonded to parallel plates are subjected to a tensile load, spherical

ruptures form and grow in radius [30]. A schematic for the deformation of cavitation is shown in Figure 3.10.

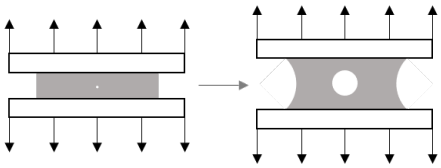


Figure 3.10: Schematic showing the cross-section of a disk of nematic elastomer bonded to parallel plates, which are stretched in uniaxial tension.

This experiment can be modeled as a spherical void inside of an infinite medium subjected to a state of triaxial extension, as seen in Figure 3.11. We do not consider nucleation (we assume that the spherical void already exists), and we assume that the cavity remains spherical throughout the deformation, which is consistent with the experiments. We are interested in the behavior as the spherical void becomes

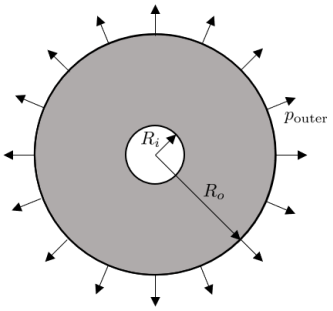


Figure 3.11: Nematic elastomeric sphere with radius R_o with spherical void of radius R_i, subjected to external pressure p_{outer}.

infinitesimally small (the limit as $R_i \to 0$) or, equivalently, when the sphere of nematic elastomer is infinitely large (the limit as $R_o \to \infty$). The deformation mapping for the growth of such a spherical void is as follows:

$$\begin{cases} \rho = \rho(R) \\ \theta = \Theta \\ \phi = \Phi \end{cases} . \tag{3.183}$$

The deformation gradient for this specific mapping then is

$$\mathbf{F} = \frac{\partial \rho}{\partial R}\mathbf{e}\langle\rho\rangle \otimes \mathbf{E}\langle R\rangle + \frac{\partial \theta}{\partial \Theta}\frac{\rho \sin\phi}{R \sin\Phi}\mathbf{e}\langle\theta\rangle \otimes \mathbf{E}\langle\Theta\rangle + \frac{\partial \phi}{\partial \Phi}\frac{\rho}{R}\mathbf{e}\langle\phi\rangle \otimes \mathbf{E}\langle\Phi\rangle \tag{3.184}$$

$$= \frac{d\rho}{dR}\mathbf{e}\langle\rho\rangle \otimes \mathbf{E}\langle R\rangle + \frac{\rho}{R}\mathbf{e}\langle\theta\rangle \otimes \mathbf{E}\langle\Theta\rangle + \frac{\rho}{R}\mathbf{e}\langle\phi\rangle \otimes \mathbf{E}\langle\Phi\rangle. \tag{3.185}$$

Thus,

$$\mathbf{F}\langle ij\rangle = \begin{pmatrix} \frac{d\rho}{dR} & & \\ & \frac{\rho}{R} & \\ & & \frac{\rho}{R} \end{pmatrix}. \tag{3.186}$$

Since nematic elastomers are incompressible, we constrain $\det \mathbf{F} = 1$:

$$\frac{d\rho}{dR}\frac{\rho^2}{R^2} = 1 \tag{3.187}$$

$$\rho^2 d\rho = R^2 dR \tag{3.188}$$

$$\rho^3 = R^3 + c. \tag{3.189}$$

We solve for the constant c in terms of ρ_i (the deformed radius at the undeformed radius $R = R_i$) and obtain

$$c = \rho_i^3 - R_i^3, \tag{3.190}$$

so we have a new expression for the deformed radius, rho:

$$\rho = \left[R^3 + \rho_i^3 - R_i^3\right]^{1/3}. \tag{3.191}$$

Let the non-dimensional azimuthal stretch at the inner radius be denoted by

$$\lambda_i = \frac{\rho_i}{R_i}. \tag{3.192}$$

Then in terms of λ_i, the deformed radius is

$$\rho = \left[R^3 + R_i^3\left(\lambda_i^3 - 1\right)\right]^{1/3}, \tag{3.193}$$

and the ratio of the deformed radius to undeformed radius is

$$\lambda = \frac{\rho}{R} = \left[1 + \frac{R_i^3}{R^3}\left(\lambda_i^3 - 1\right)\right]^{1/3}. \tag{3.194}$$

Thus, the deformation gradient is

$$\mathbf{F}\langle ij\rangle = \begin{pmatrix} \frac{1}{\lambda^2} & & \\ & \lambda & \\ & & \lambda \end{pmatrix}. \tag{3.195}$$

The principal values of $\mathbf{F}$ are $\lambda_1 \geq \lambda_2 \geq \lambda_3$: $\lambda_1 = \lambda_2 = \lambda$ (which correspond to $\hat{\boldsymbol{e}}_\theta$ and $\hat{\boldsymbol{e}}_\phi$) and $\lambda_3 = \frac{1}{\lambda^2}$ (corresponding to $\hat{\boldsymbol{e}}_\rho$).

There are the same possible cases as presented in the spherical balloon, except that for cavitation we have a new definition for R^*:

$$R^* = R_i\left(\frac{\lambda_i^3 - 1}{r^{1/4} - 1}\right)^{1/3}. \tag{3.196}$$

Stress

The stress state is the same as that for the spherical balloon, seen in Section 3.5.

Solving static equilibrium

This section is the same as that of the spherical balloon, but with the following boundary conditions imposed:

$$\sigma_{\rho\rho}(\rho =_o) = p \tag{3.197}$$

$$\sigma_{\rho\rho}(\rho =_i) = 0. \tag{3.198}$$

We solve static equilibrium in the absence of body forces.

Case 1

In Case 1, $R \in [R_i, R_o]$ is in region L. The outer pressure then is

$$p = 0. \tag{3.199}$$

Case 2

In Case 2, $R \in [R_i, R^*]$ is in region M and $R \in [R^*, R_o]$ is in region L. The two boundary conditions can be rewritten specific to the region:

$$-\eta^M(\rho = \rho_i) + \hat{\sigma}^M_{\rho\rho}(\rho = \rho_i) = 0 \tag{3.200}$$

$$-\eta^L(\rho = \rho_o) + \hat{\sigma}^L_{\rho\rho}(\rho = \rho_o) = p, \tag{3.201}$$

and there is an additional boundary condition for continuity between the two regions:

$$-\eta^L(\rho = \rho^*) + \hat{\sigma}^L_{\rho\rho}(\rho = \rho^*) = -\eta^M(\rho = \rho^*) + \hat{\sigma}^M_{\rho\rho}(\rho = \rho^*). \tag{3.202}$$

The result from region L is

$$-\eta^L(\rho^*) + \hat{\sigma}^L_{\rho\rho}(\rho^*) = p. \tag{3.203}$$

The result from region M is

$$-\eta^M(\rho^*) + \hat{\sigma}^M_{\rho\rho}(\rho^*) - \hat{\sigma}^M_{\rho\rho}(\rho_i) + \eta^M(\rho_i) = \int_{\rho_i}^{\rho^*} \frac{2}{\rho}\left(\hat{\sigma}^M_{\theta\theta} - \hat{\sigma}^M_{\rho\rho}\right) d\rho, \tag{3.204}$$

or

$$p = \int_{R_i}^{R^*} \frac{2}{\lambda(R)R}\left[\hat{\sigma}^M_{\theta\theta}(\rho = \lambda(R)R) - \hat{\sigma}^M_{\rho\rho}(\rho = \lambda(R)R)\right]\frac{1}{\lambda(R)^2} dR. \tag{3.205}$$

Case 3

In Case 3, all of the balloon is in region M. Note that the isotropic state ($r = 1$ and $\lambda \geq 1$) falls under this case. The two boundary conditions for this case are:

$$-\eta^M(\rho = \rho_i) + \hat{\sigma}^M_{\rho\rho}(\rho = \rho_i) = 0 \tag{3.206}$$

$$-\eta^M(\rho = \rho_o) + \hat{\sigma}^M_{\rho\rho}(\rho = \rho_o) = p. \tag{3.207}$$

The result of solving static equilibrium is

$$p = \int_{R_i}^{R_o} \frac{2}{\lambda(R)R}\left[\hat{\sigma}^M_{\theta\theta}(\rho = \lambda(R)R) - \hat{\sigma}^M_{\rho\rho}(\rho = \lambda(R)R)\right]\frac{1}{\lambda(R)^2}dR. \tag{3.208}$$

Case 4

In Case 4, all of the balloon is in region S. Note that the isotropic state ($r = 1$) falls under this case. The two boundary conditions for this case are:

$$-\eta^S(\rho = \rho_i) + \hat{\sigma}^S_{\rho\rho}(\rho = \rho_i) = 0 \tag{3.209}$$

$$-\eta^S(\rho = \rho_o) + \hat{\sigma}^S_{\rho\rho}(\rho = \rho_o) = p. \tag{3.210}$$

The result of solving static equilibrium is

$$p = \int_{R_i}^{R_o} \frac{2}{\lambda(R)R}\left[\hat{\sigma}^S_{\theta\theta}(\rho = \lambda(R)R) - \hat{\sigma}^S_{\rho\rho}(\rho = \lambda(R)R)\right]\frac{1}{\lambda(R)^2}dR. \tag{3.211}$$

Results

The calculations were performed in `MATLAB` and the solutions are plotted below. The outer radius is $R_o = 1$ cm, and we take the inner radius to be much smaller than the outer radius with a value of $R_i = 1 \cdot 10^{-8}$ m. The parameters used for the generalized Mooney-Rivlin model were the same as for the spherical balloon.

The pressure p, normalized by the shear modulus μ, is plotted as a function of the azimuthal stretch at the inner radius λ_i. Figure 3.12 shows the comparison between the generalized Mooney-Rivlin model of this work and the neo-Hookean model of previous work [23] for an anisotropy parameter of $r = 8$. The BTW theory and generalized Mooney-Rivlin theory deviate starting around a stretch of $\lambda_i \approx 15$. The BTW model predicts that the cavitation pressure plateaus, whereas the Extended Mooney-Rivlin model does not.

Figure 3.13a shows the cavitation results for varying anisotropy parameter r. As expected, the isotropic case, $r = 1$, is the stiffest response, and the response is softer as r increases and becomes more nematic.

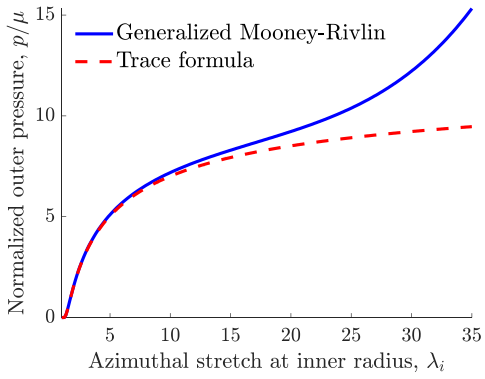
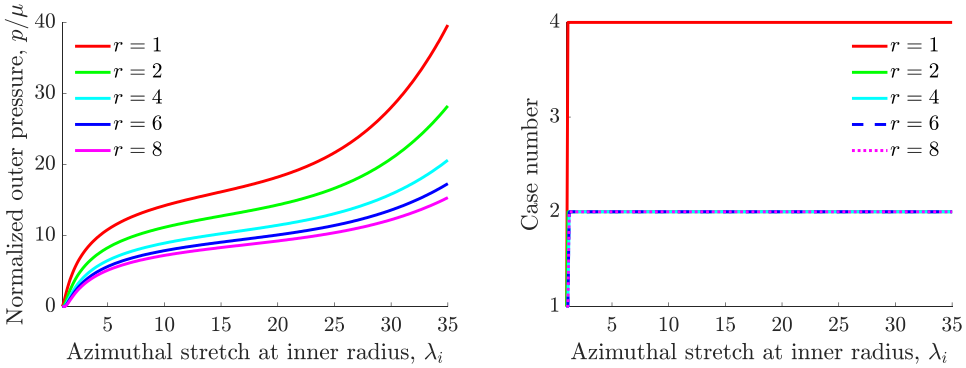


Figure 3.12: Comparison of this work's generalized Mooney-Rivlin model with the trace formula model of [23] for pressure of a growing spherical cavity inside a bulk disk.



(a) (b)

Figure 3.13: Cavitation results: (a) Results for cavitation at varying anisotropy parameter. (b) Progression of the cavitation solution through individual case numbers.

Figure 3.13b shows the solution's case at various values of λ_i. The disk is in Case 1 where the entire structure is undeformed and in region L. Then, for $r = 1$, the rest of the deformation belongs to Case 4, where the disk is entirely in region S, having a purely elastomer response with no liquid crystal effects. For $r > 1$, the disk develops microstructure in the area immediately surrounding the void that forms during cavitation. Because the void is so small compared to the length scale of the disk, the structure never moves into Case 3 (where the entire disk would be in region M).

3.8 Bending

Deformation of bending

The deformation analyzed in this section is the bending of a rectangular block into an arc of a circle, following Ericksen. A schematic of the two-dimensional cross-section of the body is shown in Figure 3.14. In the reference configuration, the block

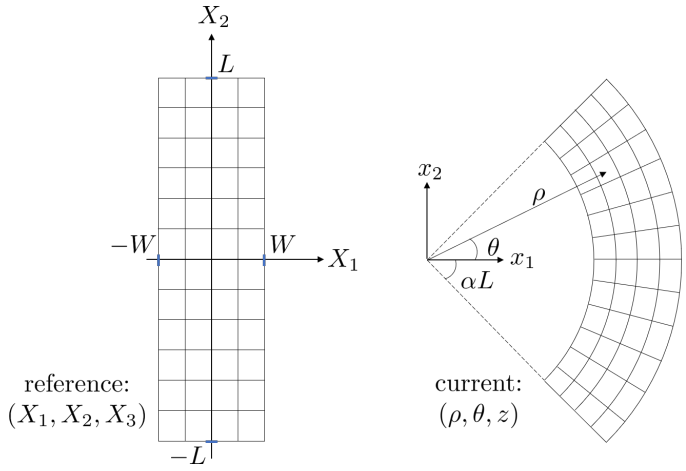


Figure 3.14: Schematic depicting the mid-plane of a rectangular block undergoing bending deformation.

has Cartesian coordinates (X_1, X_2, X_3), where

$$X_1 \in [-W, W]\,, X_2 \in [-L, L]\,, X_3 \in [-H, H]\,. \tag{3.212}$$

The planes $X_1 =$ constant become sectors of the cylindrical surface $\rho =$ constant, the planes $X_2 =$ constant become planes $\theta =$ constant, and the planes $X_3 =$ constant become planes $z =$ constant. We provide an ansatz for the deformation mapping:

$$\begin{cases} \rho = f(X_1) \\ \theta = g(X_2) \\ z = \lambda X_3 \end{cases}, \tag{3.213}$$

where f and g are functions only of X_1 and X_2 respectively. Therefore, the Cartesian coordinates in the current configuration become

$$\begin{cases} x_1 = \rho \cos\theta = f(X_1) \cos\left(g(X_2)\right) \\ x_2 = \rho \sin\theta = f(X_1) \sin\left(g(X_2)\right) \\ x_3 = z = \lambda X_3 \end{cases}. \tag{3.214}$$

The deformation gradient in Cartesian coordinates is then

$$\boldsymbol{F} = \boldsymbol{\nabla} \boldsymbol{x} \tag{3.215}$$

$$= \begin{pmatrix} f'(X_1)\cos\left(g(X_2)\right) & -f(X_1)g'(X_2)\sin\left(g(X_2)\right) & 0 \\ f'(X_1)\sin\left(g(X_2)\right) & f(X_1)g'(X_2)\cos\left(g(X_2)\right) & 0 \\ 0 & 0 & \lambda \end{pmatrix}. \tag{3.216}$$

Due to incompressibility, we must reinforce that $\det \boldsymbol{F} = 1$:

$$\lambda f'(X_1)f(X_1)g'(X_2)\cos^2\left(g(X_2)\right) + \lambda f(X_1)f'(X_1)g'(X_2)\sin^2\left(g(X_2)\right) = 1 \tag{3.217}$$

$$\lambda f(X_1)f'(X_1)g'(X_2) = 1 \tag{3.218}$$

$$g'(X_2) = \frac{1}{\lambda f(X_1)f'(X_1)}. \tag{3.219}$$

To solve for the functions f and g, we employ separation of variables, yielding two equations to be solved:

$$\frac{1}{\lambda f(X_1)f'(X_1)} = \alpha \tag{3.220}$$

$$g'(X_2) = \alpha. \tag{3.221}$$

Solving Equation 3.220, we obtain

$$\frac{1}{\alpha\lambda} = f\frac{df}{dX_1} \tag{3.222}$$

$$\int \frac{1}{\alpha\lambda} dX_1 = \int f df \tag{3.223}$$

$$\frac{1}{\alpha\lambda}X_1 + \hat{c} = \frac{f^2}{2} \tag{3.224}$$

$$f^2 = \frac{2}{\alpha\lambda}X_1 + \beta. \tag{3.225}$$

Since $\rho = f$, we have

$$\rho^2 = \frac{2}{\alpha\lambda}X_1 + \beta, \tag{3.226}$$

where β is a constant. Solving Equation 3.221 yields

$$\frac{dg}{dX_2} = \alpha \tag{3.227}$$

$$\int dg = \int \alpha dX_2 \tag{3.228}$$

$$g = \alpha X_2 + \tilde{c}, \tag{3.229}$$

where $\tilde{c}$ is a constant. Since $\theta = g$, we have

$$\theta = \alpha X_2 + \tilde{c}. \tag{3.230}$$

Assuming that the deformation is symmetric around the X_1-axis,

$$\theta(X_2) = -\theta(-X_2), \tag{3.231}$$

so $\tilde{c} = 0$. We then have

$$\theta = \alpha X_2. \tag{3.232}$$

Finally, our deformation mapping is

$$\begin{cases} \rho^2 = \frac{2}{\alpha\lambda} X_1 + \beta \\ \theta = \alpha X_2 \\ z = \lambda X_3 \end{cases}, \tag{3.233}$$

with $\rho \in \left[\sqrt{-\frac{2W}{\alpha\lambda} + \beta}, \sqrt{\frac{2W}{\alpha\lambda} + \beta}\right]$, $\theta \in [-\alpha L, \alpha L]$, and $z \in [-\lambda H, \lambda H]$. The deformation gradient will be written in mixed cylindrical/Cartesian coordinates. Thus, we have the reference configuration general coordinates as follows:

$$X^i = \{X_1, X_2, X_3\}, \tag{3.234}$$

and the current configuration general coordinates are

$$\xi^i = \{\rho, \theta, z\}, \tag{3.235}$$

with definitions from Equation 3.233. The position vector of a point in the current configuration is

$$\boldsymbol{x} = \rho \cos\theta \boldsymbol{E}_1 + \rho \sin\theta \boldsymbol{E}_2 + z\boldsymbol{E}_3. \tag{3.236}$$

The covariant basis vectors are defined as $\boldsymbol{e}_i = \frac{\partial \boldsymbol{x}}{\partial \xi^i}$, so we have

$$\boldsymbol{e}_\rho = \frac{\partial \boldsymbol{x}}{\partial \rho} = \cos\theta \boldsymbol{E}_1 + \sin\theta \boldsymbol{E}_2 \tag{3.237}$$

$$\boldsymbol{e}_\theta = \frac{\partial \boldsymbol{x}}{\partial \theta} = -\rho \sin\theta \boldsymbol{E}_1 + \rho \cos\theta \boldsymbol{E}_2 \tag{3.238}$$

$$\boldsymbol{e}_z = \frac{\partial \boldsymbol{x}}{\partial z} = \boldsymbol{E}_3 \tag{3.239}$$

with magnitudes

$$|\boldsymbol{e}_\rho| = 1 \tag{3.240}$$

$$|\boldsymbol{e}_\theta| = \rho \tag{3.241}$$

$$|\boldsymbol{e}_z| = 1. \tag{3.242}$$

We introduce the physical basis

$$\boldsymbol{e}\langle i\rangle = \frac{\boldsymbol{e}_i}{|\boldsymbol{e}_i|} = \frac{\boldsymbol{e}^i}{|\boldsymbol{e}^i|} \tag{3.243}$$

so that we can obtain our deformation gradient:

$$\begin{aligned}\boldsymbol{F} &= \underbrace{\frac{\partial \xi^i}{\partial X^j}|\boldsymbol{e}_i||\boldsymbol{E}_j|^{-1}}_{F\langle ij\rangle}\boldsymbol{e}\langle i\rangle \otimes \boldsymbol{E}\langle j\rangle \\ &= \frac{1}{\alpha\lambda\rho}\boldsymbol{e}\langle\rho\rangle \otimes \boldsymbol{E}\langle 1\rangle + \alpha\rho\boldsymbol{e}\langle\theta\rangle \otimes \boldsymbol{E}\langle 2\rangle + \lambda\boldsymbol{e}\langle z\rangle \otimes \boldsymbol{E}\langle 3\rangle.\end{aligned} \tag{3.244}$$

Note that $\frac{\partial\rho}{\partial X_1} = \frac{1}{\alpha\lambda\rho}$. The left Cauchy-Green tensor is

$$\begin{aligned}\boldsymbol{b} &= \underbrace{\frac{\partial \xi^i}{\partial X^j}\frac{\partial \xi^k}{\partial X^j}|\boldsymbol{e}_i||\boldsymbol{e}_k||\boldsymbol{E}_j|^{-2}}_{B\langle ik\rangle = F\langle ij\rangle F\langle kj\rangle}\boldsymbol{e}\langle i\rangle \otimes \boldsymbol{e}\langle k\rangle \\ &= \frac{1}{\alpha^2\lambda^2\rho^2}\boldsymbol{e}\langle\rho\rangle \otimes \boldsymbol{e}\langle\rho\rangle + \alpha^2\rho^2\boldsymbol{e}\langle\theta\rangle \otimes \boldsymbol{e}\langle\theta\rangle + \lambda^2\boldsymbol{e}\langle z\rangle \otimes \boldsymbol{e}\langle z\rangle,\end{aligned} \tag{3.245}$$

and the principal stretches are

$$\frac{1}{\alpha\lambda\rho}, \alpha\rho, \lambda. \tag{3.246}$$

If the strain energy density of a material can be written in the form $W = W(\lambda_i)$, where λ_i are the principal stretches, then the Cauchy stress of an incompressible, isotropic hyperelastic body is

$$\boldsymbol{\sigma} = -p\boldsymbol{I} + \sum_{i=1}^{3}\lambda_i\frac{\partial W}{\partial\lambda_i}\boldsymbol{e}\langle i\rangle \otimes \boldsymbol{e}\langle i\rangle. \tag{3.247}$$

Static equilibrium (in the absence of body forces) is obtained when $\operatorname{div}\boldsymbol{\sigma} = \boldsymbol{0}$. In cylindrical coordinates and for a symmetric tensor $\boldsymbol{\sigma}$, the equations we need to solve are:

$$\begin{aligned}(\operatorname{div}\boldsymbol{\sigma})\langle\rho\rangle &= \frac{\partial\sigma\langle\rho\rho\rangle}{\partial\rho} + \frac{1}{\rho}\frac{\partial\sigma\langle\rho\theta\rangle}{\partial\theta} + \frac{\sigma\langle\rho\rho\rangle - \sigma\langle\theta\theta\rangle}{\rho} + \frac{\partial\sigma\langle\rho z\rangle}{\partial z}, \\ (\operatorname{div}\boldsymbol{\sigma})\langle\theta\rangle &= \frac{\partial\sigma\langle\rho\theta\rangle}{\partial\rho} + \frac{1}{\rho}\frac{\partial\sigma\langle\theta\theta\rangle}{\partial\theta} + \frac{2\sigma\langle\rho\theta\rangle}{\rho} + \frac{\partial\sigma\langle\theta z\rangle}{\partial z}, \\ (\operatorname{div}\boldsymbol{\sigma})\langle z\rangle &= \frac{\partial\sigma\langle\rho z\rangle}{\partial\rho} + \frac{1}{\rho}\frac{\partial\sigma\langle\theta z\rangle}{\partial\theta} + \frac{\sigma\langle\rho z\rangle}{\rho} + \frac{\partial\sigma\langle zz\rangle}{\partial z}.\end{aligned} \tag{3.248}$$

Let $\rho = \kappa$, where $\kappa = \frac{1}{\alpha}$ is the radius of the neutral axis in the beam. This neutral axis is the axis that experiences zero extension because the arc length of the beam at

this radius does not change from the reference length throughout deformation. The length of the beam in the undeformed configuration is $2L$, as is the arc length of the neutral axis in the current configuration ($s = \rho\theta$), i.e. the product of the radius ($\kappa = \frac{1}{\alpha}$) and the angle ($2\alpha L$). See Figure 3.15 for details.

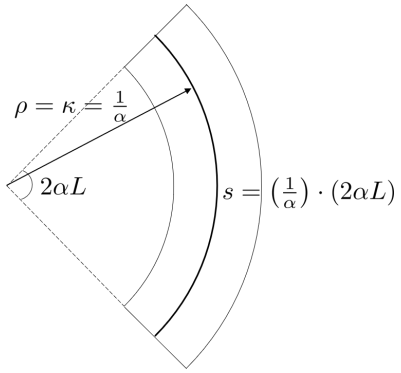


Figure 3.15: Schematic depicting the neutral axis of a rectangular block with arc length s equal to the length of the undeformed beam, $2L$.

For the case of plane strain (the stretch in the z direction is fixed at $\lambda = 1$), the principal stretches are $\frac{\rho}{\kappa}$ (corresponding to $\hat{\boldsymbol{e}}_\theta$), 1 (corresponding to $\hat{\boldsymbol{e}}_z$), and $\frac{\kappa}{\rho}$ (corresponding to $\hat{\boldsymbol{e}}_\rho$). The relative magnitudes of the principal stretches depends on the relationship between ρ and κ. This leads to a breakdown of the regions L, M, and S as in Figure 3.16.

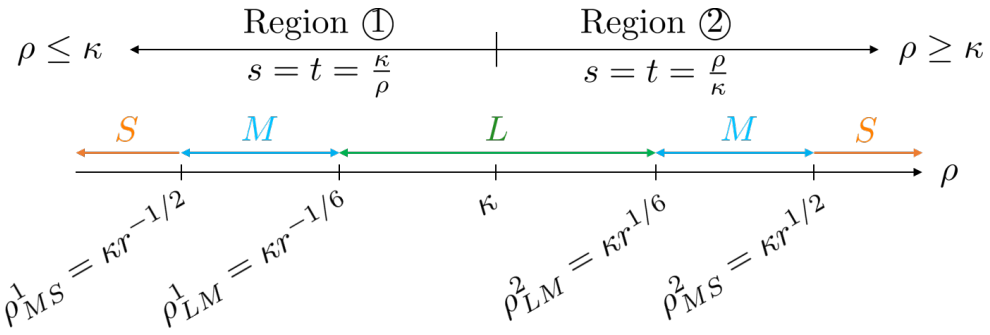


Figure 3.16: Breakdown of the regions L, M, and S along the radius ρ in the bending deformation.

The radius of the neutral axis, $\rho = \kappa$ lies in the region L. Let us denote the part of the beam where $\rho < \kappa$ by ①, and for $\rho > \kappa$ by ②. The ends of the beam are located at $\rho_1 = \sqrt{-\frac{2W}{\alpha\lambda} + \beta}$ and $\rho_2 = \sqrt{\frac{2W}{\alpha\lambda} + \beta}$. These radii are an unknown and will be solved for using the static equilibrium equation. We will define the following radii

for convenience:

$$\rho^1_{MS} = \frac{\kappa}{r^{1/2}} \tag{3.249}$$

$$\rho^1_{LM} = \frac{\kappa}{r^{1/6}} \tag{3.250}$$

$$\rho^2_{LM} = \kappa r^{1/6} \tag{3.251}$$

$$\rho^2_{MS} = \kappa r^{1/2}. \tag{3.252}$$

The boundary conditions in this problem are:

$$\sigma_{\rho\rho}|_{\rho=\rho_1} = \sigma_{\rho\rho}|_{\rho=\rho_2} = 0. \tag{3.253}$$

In addition, continuity in the radial stresses at the boundaries between regions must be enforced, i.e. at $\rho = \rho^1_{MS}$, $\rho = \rho^1_{LM}$, $\rho = \rho^2_{LM}$, $\rho = \rho^2_{MS}$. The possible cases that arise are:

- Case 1: $r > 1$ and $\rho^1_{LM} \leq \rho_1 \leq \kappa \leq \rho_2 \leq \rho^2_{LM}$: All of the beam lies in region L
- Case 2: $r > 1$ and $\rho^1_{MS} \leq \rho_1 \leq \rho^1_{LM} \leq \kappa \leq \rho^2_{LM} \leq \rho_2 \leq \rho^2_{MS}$: The inner portion of the beam lies in region L, and the outer regions lie in M
- Case 3a: $r > 1$ and $\rho_1 \leq \rho^1_{MS} \leq \rho^1_{LM} \leq \kappa \leq \rho^2_{LM} \leq \rho_2 \leq \rho^2_{MS}$: The beam, in order of increasing radius, lies in region S, then M, then L, then M
- Case 3b: $r > 1$ and $\rho^1_{MS} \leq \rho_1 \leq \rho^1_{LM} \leq \kappa \leq \rho^2_{LM} \leq \rho^2_{MS} \leq \rho_2$: The beam, in order of increasing radius, lies in region M, then L, then M, then S
- Case 4: $r > 1$ and $\rho_1 \leq \rho^1_{MS} \leq \rho^1_{LMS} \leq \kappa \leq \rho^2_{LM} \leq \rho^2_{MS} \leq \rho_2$: The beam, in order of increasing radius, lies in region S, then M, then L, then M, then S
- Case 5: $r = 1$ and $\rho_1 \leq \kappa \leq \rho_2$: The entire beam is in region S

A diagram illustrating the various cases can be seen in Figure 3.17. The heavy black line shows the beam with ends $\rho = \rho_1$ and $\rho = \rho_2$.

As in the previous deformations, we can gain insight from Figure 3.1 regarding the regions that this deformation will experience. The plane-strain ($\lambda = 1$) bending of a block is the same as a planar extension deformation (the deformation gradient of Equation 3.244 is $\boldsymbol{F} = \mathrm{diag}(1/(\alpha\rho), \alpha\rho, 1)$). Therefore, like the cylindrical balloon with $\xi = 1$, this means that $t = s$ for this deformation (recalling that s is the largest singular value of $\boldsymbol{F}$ and t is the largest singular value of $\mathrm{cof}\,\boldsymbol{F}$). If we were to follow along the $t = s$ line in Figure 3.1, we see that the deformation will move progressively through region L then M then S.

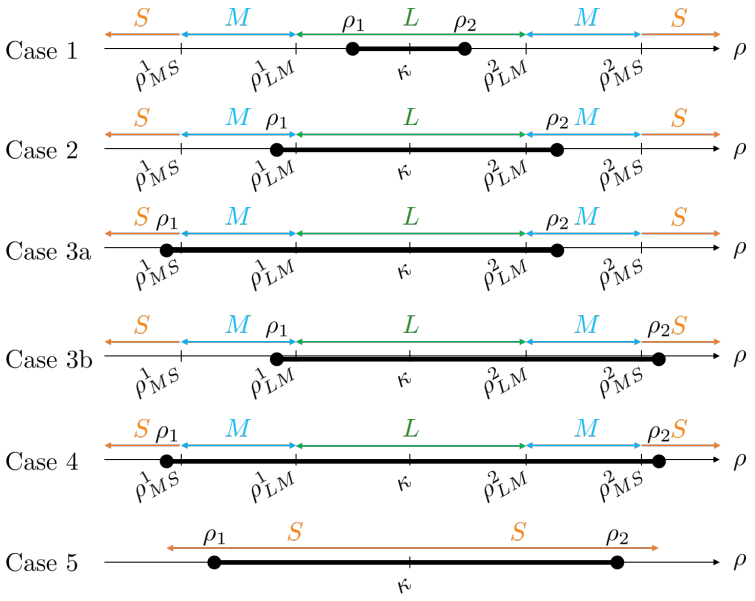


Figure 3.17: Diagram of all possible cases in the bending of a nematic elastomer block.

Stress, moment, and forces

Static equilibrium (in the absence of body forces) is obtained when $\operatorname{div} \boldsymbol{\sigma} = \mathbf{0}$. In cylindrical coordinates and for a symmetric tensor $\boldsymbol{\sigma}$,

$$\begin{aligned} \frac{\partial \sigma_{\rho\rho}}{\partial \rho} + \frac{1}{\rho}\frac{\partial \sigma_{\rho\theta}}{\partial \theta} + \frac{\sigma_{\rho\rho} - \sigma_{\theta\theta}}{\rho} + \frac{\partial \sigma_{\rho z}}{\partial z} &= 0, \\ \frac{\partial \sigma_{\rho\theta}}{\partial \rho} + \frac{1}{\rho}\frac{\partial \sigma_{\theta\theta}}{\partial \theta} + \frac{2\sigma_{\rho\theta}}{\rho} + \frac{\partial \sigma_{\theta z}}{\partial z} &= 0, \\ \frac{\partial \sigma_{\rho z}}{\partial \rho} + \frac{1}{\rho}\frac{\partial \sigma_{\theta z}}{\partial \theta} + \frac{\sigma_{\rho z}}{\rho} + \frac{\partial \sigma_{zz}}{\partial z} &= 0. \end{aligned} \tag{3.254}$$

The boundary conditions for this problem are as follows:

$$\sigma_{\rho\rho}|_{\rho=\rho_1} = 0 \tag{3.255}$$

$$\sigma_{\rho\rho}|_{\rho=\rho_2} = 0. \tag{3.256}$$

For every case, the θ and z equilibrium equations yield that $\eta = \eta(\rho)$. This leaves only the ρ equilibrium equation to be solved. After obtaining the stresses, we can calculate the following three quantities. First, the magnitude of the force in the θ-direction is

$$|\boldsymbol{F}_\theta| = \int_{-\lambda H}^{\lambda H}\int_{\rho_1}^{\rho_2} \sigma_{\theta\theta} d\rho dz. \tag{3.257}$$

(Note that $\lambda = 1$ here.) Second, the magnitude of the force in the z-direction is

$$|\boldsymbol{F}_z| = \int_{-\frac{L}{\kappa}}^{\frac{L}{\kappa}}\int_{\rho_1}^{\rho_2} \sigma_{zz} d\rho d\theta. \tag{3.258}$$

Finally, the moment is

$$M = \int_{-\lambda H}^{\lambda H} \int_{\rho_1}^{\rho_2} \sigma_{\theta\theta} \rho d\rho dz. \tag{3.259}$$

Solving static equilibrium

Case 1

In this trivial case, the beam lies entirely in the L region. That is, the radius $\rho \in [\rho_1, \rho_2]$ is such that

$$\rho_{LM}^1 \leq \rho_1 \leq \kappa \leq \rho_2 \leq \rho_{LM}^2. \tag{3.260}$$

Thus, the stress condition is simply the Lagrange multiplier:

$$\boldsymbol{\sigma}^L = -\eta^L \boldsymbol{I}. \tag{3.261}$$

We can solve for static equilibrium by solving div $\boldsymbol{\sigma} = \mathbf{0}$ in cylindrical coordinates, given by Equation 3.248:

$$z : \eta^L = \eta^L(\rho, \theta) \tag{3.262}$$

$$\theta : \eta^L = \eta^L(\rho) \tag{3.263}$$

$$\rho : \frac{\partial \sigma_{\rho\rho}^L}{\partial \rho} + \frac{1}{\rho}\left(\sigma_{\rho\rho}^L - \sigma_{\theta\theta}^L\right) = 0 \rightarrow \sigma_{\rho\rho}^L = \text{const} \rightarrow \eta^L = \text{const}. \tag{3.264}$$

We can use the traction boundary condition, $\sigma_{\rho\rho}^L|_{\rho=\rho_1} = \sigma_{\rho\rho}^L|_{\rho=\rho_2} = 0$. Thus, we find that the Lagrange multiplier $\eta^L = 0$. The magnitude of the force in the θ-direction is calculated from Equation 3.257:

$$|\boldsymbol{F}_\theta| = \int_{-\lambda H}^{\lambda H} \int_{\rho_1}^{\rho_2} \sigma_{\theta\theta}^L d\rho dz = 0. \tag{3.265}$$

The magnitude of the force in the z-direction is calculated from Equation 3.258:

$$|\boldsymbol{F}_z| = \int_{-\frac{L}{\kappa}}^{\frac{L}{\kappa}} \int_{\rho_1}^{\rho_2} \sigma_{zz}^L d\rho d\theta = 0. \tag{3.266}$$

Finally, the moment is calculated from Equation 3.259:

$$M = \int_{-\lambda H}^{\lambda H} \int_{\rho_1}^{\rho_2} \sigma_{\theta\theta}^L \rho d\rho dz = 0. \tag{3.267}$$

This case physically refers to a beam with $\kappa \rightarrow \infty$ (the radius of curvature is infinitely large, so the beam is flat). This refers to the undeformed case, which corresponds to zero stresses.

Case 2

In Case 2, the beam has outer regions M and inner regions L, and the radius $\rho \in [\rho_1, \rho_2]$ is such that

$$\rho^1_{MS} \leq \rho_1 \leq \rho^1_{LM} \leq \kappa \leq \rho^2_{LM} \leq \rho_2 \leq \rho^2_{MS}. \tag{3.268}$$

The boundary conditions (BCs) specific to this case are:

$$\sigma^{M①}_{\rho\rho}(\rho_1) = 0 \tag{3.269}$$
$$\sigma^{M①}_{\rho\rho}(\rho^1_{LM}) = \sigma^{L①}_{\rho\rho}(\rho^1_{LM}) \tag{3.270}$$
$$\sigma^{L①}_{\rho\rho}(\kappa) = \sigma^{L②}_{\rho\rho}(\kappa) \tag{3.271}$$
$$\sigma^{L②}_{\rho\rho}(\rho^2_{LM}) = \sigma^{M②}_{\rho\rho}(\rho^2_{LM}) \tag{3.272}$$
$$\sigma^{M②}_{\rho\rho}(\rho_2) = 0. \tag{3.273}$$

We solve the equations of static equilibrium using Equation 3.248, which are in cylindrical coordinates. For every region, we only need to solve the equation in ρ:

$$\frac{\partial \sigma_{\rho\rho}}{\partial \rho} + \frac{1}{\rho}\left(\sigma_{\rho\rho} - \sigma^L_{\theta\theta}\right) = 0. \tag{3.274}$$

Moving from left to right within the beam, we start at region $M①$:

$$-\frac{d\eta^{M①}}{d\rho} + \frac{d\hat{\sigma}^{M①}_{\rho\rho}}{d\rho} + \frac{1}{\rho}\left(\hat{\sigma}^{M①}_{\rho\rho} - \hat{\sigma}^{M①}_{\theta\theta}\right) = 0. \tag{3.275}$$

Solving for $\eta^{M①}$ yields

$$\eta^{M①} = \int \left[\frac{d\hat{\sigma}^{M①}_{\rho\rho}}{d\rho} + \frac{1}{\rho}\left(\hat{\sigma}^{M①}_{\rho\rho} - \hat{\sigma}^{M①}_{\theta\theta}\right)\right] d\rho + C_1. \tag{3.276}$$

Using the first BC, we can obtain C_1. Next, in region $L①$, we repeat the same procedure and obtain:

$$\eta^{L①} = \int \left[\frac{d\hat{\sigma}^{L①}_{\rho\rho}}{d\rho} + \frac{1}{\rho}\left(\hat{\sigma}^{L①}_{\rho\rho} - \hat{\sigma}^{L①}_{\theta\theta}\right)\right] d\rho + C_2. \tag{3.277}$$

Using the second BC, we can obtain C_2. Repeating the same procedure in $L②$, we have

$$\eta^{L②} = \int \left[\frac{d\hat{\sigma}^{L②}_{\rho\rho}}{d\rho} + \frac{1}{\rho}\left(\hat{\sigma}^{L②}_{\rho\rho} - \hat{\sigma}^{L②}_{\theta\theta}\right)\right] d\rho + C_3. \tag{3.278}$$

Using the third BC gives us C_3. Finally, in region $M②$, we have

$$\eta^{M②} = \int \left[\frac{d\hat{\sigma}^{M②}_{\rho\rho}}{d\rho} + \frac{1}{\rho}\left(\hat{\sigma}^{M②}_{\rho\rho} - \hat{\sigma}^{M②}_{\theta\theta}\right)\right] d\rho + C_4. \tag{3.279}$$

Using the fourth BC gives us C_4, and using the last BC gives us the unknown β, which gives us the deformed radii of the beam, ρ_1 and ρ_2. Thus, we now have all the stress values and can calculate $|\boldsymbol{F}_\theta|$, $|\boldsymbol{F}_z|$, and M from Eqns. 3.257, 3.258, and 3.259 respectively. The integral in ρ between ρ_1 and ρ_2 must be broken into individual regions with corresponding bounds:

$$|\boldsymbol{F}_\theta| = \int_{-\lambda H}^{\lambda H} \left[\int_{\rho_1}^{\rho^1 LM} \sigma_{\theta\theta}^{M①} d\rho + \int_{\rho_{LM}^1}^{\kappa} \sigma_{\theta\theta}^{L①} d\rho \right] dz \tag{3.280}$$

$$+ \int_{-\lambda H}^{\lambda H} \left[\int_{\kappa}^{\rho^2 LM} \sigma_{\theta\theta}^{L②} d\rho + \int_{\rho_{LM}^2}^{\rho_2} \sigma_{\theta\theta}^{M②} d\rho \right] dz \tag{3.281}$$

$$|\boldsymbol{F}_z| = \int_{-\frac{L}{\kappa}}^{\frac{L}{\kappa}} \left[\int_{\rho_1}^{\rho^1 LM} \sigma_{zz}^{M①} d\rho + \int_{\rho_{LM}^1}^{\kappa} \sigma_{zz}^{L①} d\rho \right] \tag{3.282}$$

$$+ \int_{-\frac{L}{\kappa}}^{\frac{L}{\kappa}} \left[\int_{\kappa}^{\rho^2 LM} \sigma_{zz}^{L②} d\rho + \int_{\rho_{LM}^2}^{\rho_2} \sigma_{zz}^{M②} d\rho \right] d\theta \tag{3.283}$$

$$M = \int_{-\lambda H}^{\lambda H} \left[\int_{\rho_1}^{\rho^1 LM} \sigma_{\theta\theta}^{M①} \rho d\rho + \int_{\rho_{LM}^1}^{\kappa} \sigma_{\theta\theta}^{L①} \rho d\rho \right] \tag{3.284}$$

$$+ \int_{-\lambda H}^{\lambda H} \left[\int_{\kappa}^{\rho^2 LM} \sigma_{\theta\theta}^{L②} \rho d\rho + \int_{\rho_{LM}^2}^{\rho_2} \sigma_{\theta\theta}^{M②} \rho d\rho \right] dz. \tag{3.285}$$

Case 3a

In Case 3a, the beam, in order of increasing radius, has portions that are in region M, L, M, and S. The radius $\rho \in [\rho_1, \rho_2]$ is such that

$$\rho_1 \leq \rho_{MS}^1 \leq \rho_{LM}^1 \leq \kappa \leq \rho_{LM}^2 \leq \rho_2 \leq \rho_{MS}^2, \tag{3.286}$$

and the boundary conditions specific to this case are:

$$\sigma_{\rho\rho}^{S①}(\rho_1) = 0 \tag{3.287}$$

$$\sigma_{\rho\rho}^{S①}(\rho_{MS}^1) = \sigma_{\rho\rho}^{M①}(\rho_{MS}^1) \tag{3.288}$$

$$\sigma_{\rho\rho}^{M①}(\rho_{LM}^1) = \sigma_{\rho\rho}^{L①}(\rho_{LM}^1) \tag{3.289}$$

$$\sigma_{\rho\rho}^{L①}(\kappa) = \sigma_{\rho\rho}^{L②}(\kappa) \tag{3.290}$$

$$\sigma_{\rho\rho}^{L②}(\rho_{LM}^2) = \sigma_{\rho\rho}^{M②}(\rho_{LM}^2) \tag{3.291}$$

$$\sigma_{\rho\rho}^{M②}(\rho_2) = 0. \tag{3.292}$$

Solving from left to right within the beam, we start at region $S①$:

$$\eta^{S①} = \int \left[\frac{d\hat{\sigma}_{\rho\rho}^{S①}}{d\rho} + \frac{1}{\rho} \left(\hat{\sigma}_{\rho\rho}^{S①} - \hat{\sigma}_{\theta\theta}^{S①} \right) \right] d\rho + C_1. \tag{3.293}$$

We can use the first BC to solve for C_1. In region $M_{①}$ we have

$$\eta^{M①} = \int \left[\frac{d\hat{\sigma}_{\rho\rho}^{M①}}{d\rho} + \frac{1}{\rho}\left(\hat{\sigma}_{\rho\rho}^{M①} - \hat{\sigma}_{\theta\theta}^{M①}\right)\right] d\rho + C_2. \tag{3.294}$$

Solving the second BC gives us C_2. In region $L_{①}$ we have

$$\eta^{L①} = \int \left[\frac{d\hat{\sigma}_{\rho\rho}^{L①}}{d\rho} + \frac{1}{\rho}\left(\hat{\sigma}_{\rho\rho}^{L①} - \hat{\sigma}_{\theta\theta}^{L①}\right)\right] d\rho + C_3. \tag{3.295}$$

Solving the third BC gives us C_3. In region $L_{②}$ we have

$$\eta^{L②} = \int \left[\frac{d\hat{\sigma}_{\rho\rho}^{L②}}{d\rho} + \frac{1}{\rho}\left(\hat{\sigma}_{\rho\rho}^{L②} - \hat{\sigma}_{\theta\theta}^{L②}\right)\right] d\rho + C_4. \tag{3.296}$$

Solving the fourth BC gives us C_4. Finally, in region $M_{②}$ we have

$$\eta^{M②} = \int \left[\frac{d\hat{\sigma}_{\rho\rho}^{M②}}{d\rho} + \frac{1}{\rho}\left(\hat{\sigma}_{\rho\rho}^{M②} - \hat{\sigma}_{\theta\theta}^{M②}\right)\right] d\rho + C_5. \tag{3.297}$$

Solving the fifth BC gives us C_5, and solving the last BC gives us β. We can calculate $|\boldsymbol{F}_\theta|$, $|\boldsymbol{F}_z|$, and M from Eqns. 3.257, 3.258, and 3.259 analogously as we did in Case 2, breaking the integrals down to their individual regions between ρ_1 and ρ_2.

Case 3b

Case 3b is very similar to Case 3a. In this case, the beam has portions that are in region S, M, L, and M, in order of increasing radius. The radius $\rho \in [\rho_1, \rho_2]$ is such that

$$\rho_{MS}^1 \leq \rho_1 \leq \rho_{LM}^1 \leq \kappa \leq \rho_{LM}^2 \leq \rho_{MS}^2 \leq \rho_2, \tag{3.298}$$

and the boundary conditions specific to this case are:

$$\sigma_{\rho\rho}^{M①}(\rho_1) = 0 \tag{3.299}$$

$$\sigma_{\rho\rho}^{M①}(\rho_{LM}^1) = \sigma_{\rho\rho}^{L①}(\rho_{LM}^1) \tag{3.300}$$

$$\sigma_{\rho\rho}^{L①}(\kappa) = \sigma_{\rho\rho}^{L②}(\kappa) \tag{3.301}$$

$$\sigma_{\rho\rho}^{L②}(\rho_{LM}^2) = \sigma_{\rho\rho}^{M②}(\rho_{LM}^2) \tag{3.302}$$

$$\sigma_{\rho\rho}^{M②}(\rho_{MS}^2) = \sigma_{\rho\rho}^{S②}(\rho_{MS}^2) \tag{3.303}$$

$$\sigma_{\rho\rho}^{S②}(\rho_2) = 0. \tag{3.304}$$

Solving from left to right within the beam, we start at region $M_{①}$:

$$\eta^{M①} = \int \left[\frac{d\hat{\sigma}_{\rho\rho}^{M①}}{d\rho} + \frac{1}{\rho}\left(\hat{\sigma}_{\rho\rho}^{M①} - \hat{\sigma}_{\theta\theta}^{M①}\right)\right] d\rho + C_1. \tag{3.305}$$

We can use the first BC to solve for C_1. In region $L_{①}$ we have

$$\eta^{L①} = \int \left[\frac{d\hat{\sigma}_{\rho\rho}^{L①}}{d\rho} + \frac{1}{\rho}\left(\hat{\sigma}_{\rho\rho}^{L①} - \hat{\sigma}_{\theta\theta}^{L①}\right)\right] d\rho + C_2. \tag{3.306}$$

Solving the second BC gives us C_2. In region $L_{②}$ we have

$$\eta^{L②} = \int \left[\frac{d\hat{\sigma}_{\rho\rho}^{L②}}{d\rho} + \frac{1}{\rho}\left(\hat{\sigma}_{\rho\rho}^{L②} - \hat{\sigma}_{\theta\theta}^{L②}\right)\right] d\rho + C_3. \tag{3.307}$$

Solving the third BC gives us C_3. In region $M_{②}$ we have

$$\eta^{M②} = \int \left[\frac{d\hat{\sigma}_{\rho\rho}^{M②}}{d\rho} + \frac{1}{\rho}\left(\hat{\sigma}_{\rho\rho}^{M②} - \hat{\sigma}_{\theta\theta}^{M②}\right)\right] d\rho + C_4. \tag{3.308}$$

Solving the fourth BC gives us C_4. Finally, in region $S_{②}$ we have

$$\eta^{S②} = \int \left[\frac{d\hat{\sigma}_{\rho\rho}^{S②}}{d\rho} + \frac{1}{\rho}\left(\hat{\sigma}_{\rho\rho}^{S②} - \hat{\sigma}_{\theta\theta}^{S②}\right)\right] d\rho + C_5. \tag{3.309}$$

Solving the fifth BC gives us C_5, and solving the last BC gives us β. We can calculate $|\boldsymbol{F}_\theta|$, $|\boldsymbol{F}_z|$, and M from Eqns. 3.257, 3.258, and 3.259 as in Case 3a.

Case 4

In Case 4, in order of increasing radius, the beam lies in regions S, M, L, M, and S. The radius $\rho \in [\rho_1, \rho_2]$ is such that

$$\rho_1 \leq \rho_{MS}^1 \leq \rho_{LMS}^1 \leq \kappa \leq \rho_{LM}^2 \leq \rho_{MS}^2 \leq \rho_2. \tag{3.310}$$

The boundary conditions specific to this case are:

$$\sigma_{\rho\rho}^{S①}(\rho_1) = 0 \tag{3.311}$$

$$\sigma_{\rho\rho}^{M①}(\rho_{MS}^1) = \sigma_{\rho\rho}^{S①}(\rho_{MS}^1) \tag{3.312}$$

$$\sigma_{\rho\rho}^{M①}(\rho_{LM}^1) = \sigma_{\rho\rho}^{L①}(\rho_{LM}^1) \tag{3.313}$$

$$\sigma_{\rho\rho}^{L①}(\kappa) = \sigma_{\rho\rho}^{L②}(\kappa) \tag{3.314}$$

$$\sigma_{\rho\rho}^{L②}(\rho_{LM}^2) = \sigma_{\rho\rho}^{M②}(\rho_{LM}^2) \tag{3.315}$$

$$\sigma_{\rho\rho}^{M②}(\rho_{MS}^2) = \sigma_{\rho\rho}^{S②}(\rho_{MS}^2) \tag{3.316}$$

$$\sigma_{\rho\rho}^{S②}(\rho_2) = 0. \tag{3.317}$$

Solving from left to right within the beam, we start at region $S\textcircled{1}$:

$$\eta^{S\textcircled{1}} = \int \left[\frac{d\hat{\sigma}_{\rho\rho}^{S\textcircled{1}}}{d\rho} + \frac{1}{\rho}\left(\hat{\sigma}_{\rho\rho}^{S\textcircled{1}} - \hat{\sigma}_{\theta\theta}^{S\textcircled{1}}\right)\right] d\rho + C_1. \tag{3.318}$$

We can use the first BC to solve for C_1. In region $M\textcircled{1}$ we have

$$\eta^{M\textcircled{1}} = \int \left[\frac{d\hat{\sigma}_{\rho\rho}^{M\textcircled{1}}}{d\rho} + \frac{1}{\rho}\left(\hat{\sigma}_{\rho\rho}^{M\textcircled{1}} - \hat{\sigma}_{\theta\theta}^{M\textcircled{1}}\right)\right] d\rho + C_2. \tag{3.319}$$

Solving the second BC gives us C_2. In region $L\textcircled{1}$ we have

$$\eta^{L\textcircled{1}} = \int \left[\frac{d\hat{\sigma}_{\rho\rho}^{L\textcircled{1}}}{d\rho} + \frac{1}{\rho}\left(\hat{\sigma}_{\rho\rho}^{L\textcircled{1}} - \hat{\sigma}_{\theta\theta}^{L\textcircled{1}}\right)\right] d\rho + C_3. \tag{3.320}$$

Solving the third BC gives us C_3. In region $L\textcircled{2}$ we have

$$\eta^{L\textcircled{2}} = \int \left[\frac{d\hat{\sigma}_{\rho\rho}^{L\textcircled{2}}}{d\rho} + \frac{1}{\rho}\left(\hat{\sigma}_{\rho\rho}^{L\textcircled{2}} - \hat{\sigma}_{\theta\theta}^{L\textcircled{2}}\right)\right] d\rho + C_4. \tag{3.321}$$

Solving the fourth BC gives us C_4. In region $M\textcircled{2}$ we have

$$\eta^{M\textcircled{2}} = \int \left[\frac{d\hat{\sigma}_{\rho\rho}^{M\textcircled{2}}}{d\rho} + \frac{1}{\rho}\left(\hat{\sigma}_{\rho\rho}^{M\textcircled{2}} - \hat{\sigma}_{\theta\theta}^{M\textcircled{2}}\right)\right] d\rho + C_5. \tag{3.322}$$

Solving the fifth BC gives us C_5. Finally, in region $S\textcircled{2}$ we have

$$\eta^{S\textcircled{2}} = \int \left[\frac{d\hat{\sigma}_{\rho\rho}^{S\textcircled{2}}}{d\rho} + \frac{1}{\rho}\left(\hat{\sigma}_{\rho\rho}^{S\textcircled{2}} - \hat{\sigma}_{\theta\theta}^{S\textcircled{2}}\right)\right] d\rho + C_6 \tag{3.323}$$

Solving the sixth BC gives us C_6, and solving the last BC gives us β. We can calculate $|\boldsymbol{F}_\theta|$, $|\boldsymbol{F}_z|$, and M from Eqns. 3.257, 3.258, and 3.259.

Case 5

In Case 5, the entire beam is in region S ($\rho_1 \leq \kappa \leq \rho_2$), and the anisotropy parameter $r = 1$, corresponding to a rubber material, and not a nematic elastomer. The radius $\rho \in [\rho_1, \rho_2]$ is such that

$$\rho_1 \leq \kappa \leq \rho_2. \tag{3.324}$$

The boundary conditions specific to this case are:

$$\sigma_{\rho}^{S\textcircled{1}}(\rho_1) = 0 \tag{3.325}$$

$$\sigma_{\rho\rho}^{S\textcircled{1}}(\kappa) = \sigma_{\rho\rho}^{S\textcircled{2}}(\kappa) \tag{3.326}$$

$$\sigma_{\rho\rho}^{S\textcircled{2}}(\rho_2) = 0. \tag{3.327}$$

Notice that at $\rho = \kappa$, the stress in the r-direction will be zero: $\sigma_{\rho\rho}^{S\textcircled{1}}(\kappa) = \sigma_{\rho\rho}^{S\textcircled{2}}(\kappa) = 0$, corresponding to the neutral axis at the radius of curvature. We begin by solving in the region $r \leq \kappa$ with $S\textcircled{1}$:

$$\eta^{S\textcircled{1}} = \int \left[\frac{d\hat{\sigma}_{\rho\rho}^{S\textcircled{1}}}{d\rho} + \frac{1}{\rho}\left(\hat{\sigma}_{\rho\rho}^{S\textcircled{1}} - \hat{\sigma}_{\theta\theta}^{S\textcircled{1}}\right)\right] d\rho + C_1. \tag{3.328}$$

We can use the first BC to get C_1, then move to the region $r \geq \kappa$ with $S\textcircled{2}$:

$$\eta^{S\textcircled{2}} = \int \left[\frac{d\hat{\sigma}_{\rho\rho}^{S\textcircled{2}}}{d\rho} + \frac{1}{\rho}\left(\hat{\sigma}_{\rho\rho}^{S\textcircled{2}} - \hat{\sigma}_{\theta\theta}^{S\textcircled{2}}\right)\right] d\rho + C_2. \tag{3.329}$$

We use the second BC to get C_2, and then the third BC to get β. With all the unknowns known, we again calculate $|\boldsymbol{F}_\theta|$, $|\boldsymbol{F}_z|$, and M from Eqns. 3.257, 3.258, and 3.259.

Results

The stresses, forces, and moments as well as the deformed radii were all solved for using `Mathematica` using the following parameters: $c_1 = 1.03 \cdot 10^5$ Pa, $c_2 = 1.96 \cdot 10^2$ Pa, $d_1 = 1.63 \cdot 10^{-2}$ Pa, $p_1 = 1$, $p_2 = 1$, and $q_1 = 2$. The beam's width is $W = 1$ cm, and its length is $L = 2W$. Figures 3.19 and 3.20 (with the legend in Figure 3.18) plot the bending moment and the cases as a function of $\alpha = 1/\kappa$ (the radius of curvature). A value of $\kappa \to \infty$ (or $\alpha = \frac{1}{\kappa} = 0$) corresponds to the undeformed beam (with an infinite radius of curvature), and as κ decreases (α increases), this corresponds to the beam bending more and more.

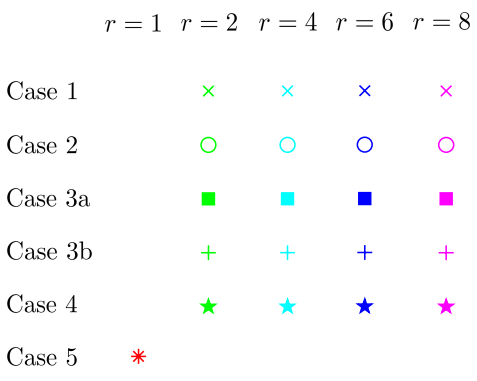


Figure 3.18: Legend for Figures 3.19 and 3.20.

The case $r = 1$ yields the highest bending moment, as expected for the isotropic material. The solution is entirely in Case 5, i.e. the beam is entirely in the S region,

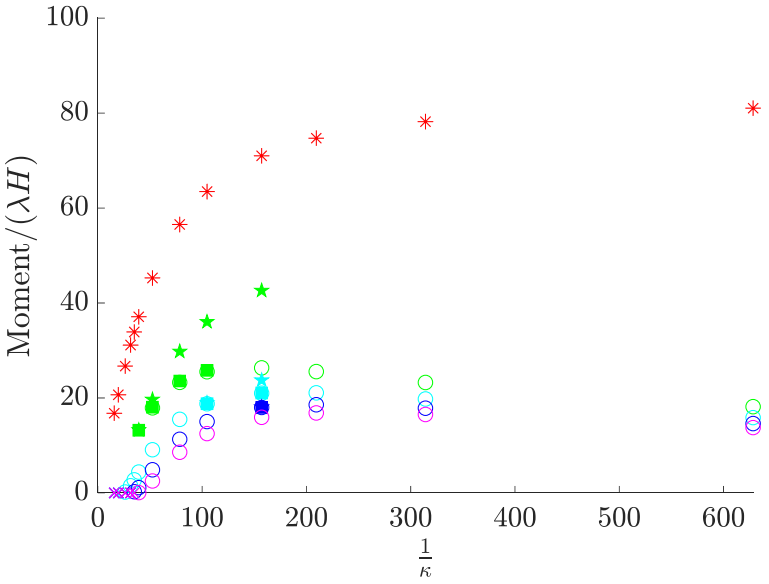


Figure 3.19: Results for bending moment at varying anisotropy parameter.

so it has a solid-like rubber response. For $r > 1$, we first note the spontaneous deformation associated with the nematic state, similar to what we observed in the balloon inflation deformations. Due to the change in temperature associated with an anisotropy parameter $r > 1$, the block will experience a spontaneous stretching along some preferred direction and contraction perpendicular in the transverse directions. There is a change in the block's curvature at zero moment associated with this spontaneous deformation.

Next, we turn our attention to the fact that there are sometimes multiple cases that yield an equilibrium solution for a given value of $\alpha = 1/\kappa$, so all possible solutions are plotted. For instance, for $r = 2$, the beam begins in Case 1 at $\alpha = 0$, where all of the beam lies in region L and there is no deformation. Then for every point $\alpha > 0$, the beam has equilibrium solutions in Case 2, where the inner part of the beam is in region L, and the outer parts of the beam lie in region M. Physically, the beam is developing fine-scale microstructure in the innermost region (which is experiencing compressive stress) and the outermost region (which is experiencing tensile stress). The region in between is still in region L, experiencing zero stress. Then for α approximately between 40 and 100, the beam has three possible equilibrium solutions, in Cases 2, 3a, and 4. Case 3a physically corresponds to some development of region S in the innermost part of the beam, and Case 4 corresponds to development of region S in both the innermost and outermost parts

of the beam. Then Case 3a stops being a viable equilibrium solution, then shortly after that Case 4 stops being a possible equilibrium solution, and for $\alpha > 160$, Case 2 is the only equilibrium solution, and the bending moment decreases.

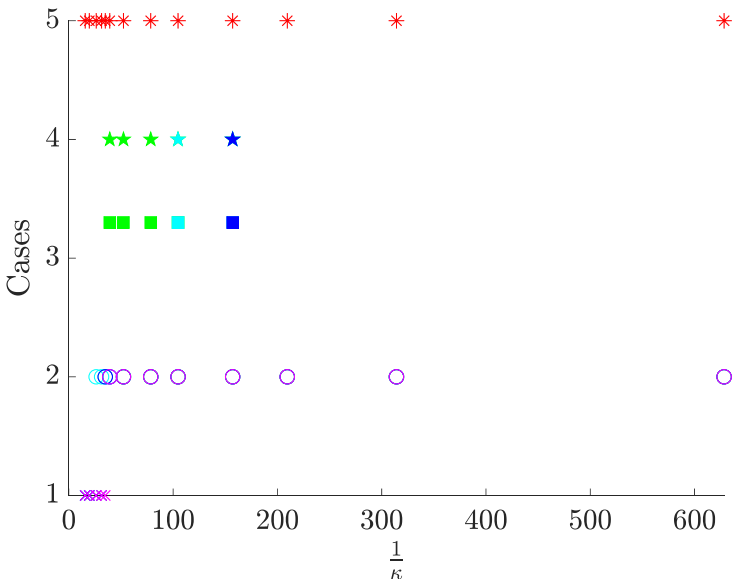


Figure 3.20: Progression of the bending solution through individual case numbers.

3.9 Conclusion

The remarkable softness that is characteristic of nematic elastomers has emphasized the need for an energy density that extends beyond the limitations of the neo-Hookean trace formula. In this work we have developed a generalized Mooney-Rivlin energy density and have used this model to solve some examples of universal deformations in nematic elastomers. Nematic elastomers have been studied extensively in classical geometries, such as the deformation of thin sheets, and traditional loading conditions, such as uniaxial tension, but they have yet to be explored in-depth as deformations applied to the bulk material, such as the bending of a block, cavitation of a disk, and internal pressure of a balloon. With the relatively recent developments in synthesizing these materials in the bulk with click chemistry and 3D printing, more complex geometries and loading configurations can be explored, and the solutions to the universal deformations for nematic elastomers is a significant step in this direction.

Chapter 4

A GENERAL CONSTITUTIVE MODEL FOR A NON-IDEAL ISOTROPIC-GENESIS POLYDOMAIN NEMATIC ELASTOMER

4.1 Introduction

The previous chapter addressed ideal nematic elastomers. In these ideal materials, the director can reorient freely. Consequently, the fine-scale domain patterns can form and change freely, and the overall behavior is described by the relaxed energy. However, in reality, there is a resistance to changing domain patterns. Specifically, as a result of the synthesis process, there may be a local preference to director alignment at each material point. This preference is extremely large in nematic-genesis elastomers (materials crosslinked in the nematic state), but relatively small in the isotropic-genesis elastomers (materials crosslinked in the isotropic state) [8, 74]. Consequently, isotropic-genesis elastomers still show soft behavior, though one needs to apply a small stress to change domain patterns—see Figure 4a from [74]. Further, isotropic-genesis elastomers have a fine polydomain structure as synthesized because the director preference is random—see Figure 2a from [74]. This preference can be modeled as non-ideality [7], a term that is small in isotropic-genesis nematic elastomers (also discussed in Chapter 2 of this thesis). Since the material is now possibly heterogenous, one has to relax and homogenize: this can be accomplished through bounds [9] or numerically [85].

In this chapter, we develop a constitutive model to describe the macroscopic response of nematic elastomers (on a scale large compared to domain patterns, but small compared to the scale of application). The idea is to use the relaxation and identify internal variables that describe the fine-scale domain patterns and then impose a kinetic process with dissipation on these internal variables. We validate the model against the experiments of Tokumoto, Takabe, and Urayama, as reported in Tokumoto et al. [71]. Finally, we implement the model in the finite element program `ABAQUS`. After verifying it using homogenous deformations, we study the problem of torsion of a cylinder. We identify an interesting instability at large torsional strains as a result of the Poynting effect.

This continuum-level model, and its implementation for finite element analysis, is the first of its kind for modeling non-ideal polydomain nematic elastomers in 3D.

It is a powerful tool that can be used to analyze nematic elastomers in arbitrarily complex deformations, which will contribute towards nematic elastomers becoming an accessible engineering material.

4.2 Formulation of the constitutive relation

Revisiting the relaxed energy of an ideal nematic elastomer

We follow the work of DeSimone and Dolzmann [23] using the same three regions of interest, a liquid-like region L, a solid-like region S, and a region M in which laminated microstructure occurs. Recall the generalized energy density of a nematic elastomer

$$W_{\mathrm{NE}}(\boldsymbol{F}, \boldsymbol{n}) = f(\boldsymbol{F}^T \boldsymbol{\ell}_{\boldsymbol{n}}^{-1} \boldsymbol{F}), \tag{4.1}$$

where $\boldsymbol{\ell}_{\boldsymbol{n}} = r^{-1/3}((r-1)\boldsymbol{n} \otimes \boldsymbol{n} + \boldsymbol{I})$ is the "step-length tensor" that describes the metric of the nematic elastomer in the "stress-free state". The corresponding elastic energy is

$$W(\boldsymbol{F}) = \min_{\boldsymbol{n}} W_{NE}(\boldsymbol{F}, \boldsymbol{n}) = \min_{\boldsymbol{Q} \in SO(3)} f(\boldsymbol{F}^\top \boldsymbol{\ell}_{\boldsymbol{Q}\boldsymbol{n}_0}^{-1} \boldsymbol{F}) = \min_{\boldsymbol{Q} \in SO(3)} f(\boldsymbol{F}^\top \boldsymbol{Q} \boldsymbol{\ell}_{\boldsymbol{n}_0}^{-1} \boldsymbol{Q}^\top \boldsymbol{F}). \tag{4.2}$$

Now, consider a situation where the nematic elastomer has formed a fine-scale domain pattern so that the resulting "stress-free" state is described by a metric $\boldsymbol{G}$. We know from previous chapters that

$$\boldsymbol{G} = \boldsymbol{Q} \boldsymbol{G}_0(\lambda, \delta) \boldsymbol{Q}^\top, \tag{4.3}$$

where

$$\boldsymbol{Q} \in SO(3), \tag{4.4}$$

$$\boldsymbol{G}_0(\lambda, \delta) = \begin{pmatrix} \lambda^2 & 0 & 0 \\ 0 & \frac{\delta^2}{\lambda^2} & 0 \\ 0 & 0 & \frac{1}{\delta^2} \end{pmatrix}, \text{ and} \tag{4.5}$$

$$\lambda, \delta \in \mathcal{T} := \{(s,t) : t \leq r^{1/6}, t \leq s^2, t \geq \sqrt{s}\}. \tag{4.6}$$

The triangular region $\mathcal{T}$ is depicted in Figure 4.1.

In analogy to (4.2), we may write the energy of a nematic elastomer with fine-scale domain pattern characterized by λ, δ to be

$$W_{\mathrm{PNE}}(\boldsymbol{F}, \lambda, \delta) = \min_{\boldsymbol{Q} \in SO(3)} f(\boldsymbol{F}^\top \boldsymbol{Q} \boldsymbol{G}_0^{-1} \boldsymbol{Q}^\top \boldsymbol{F}), \tag{4.7}$$

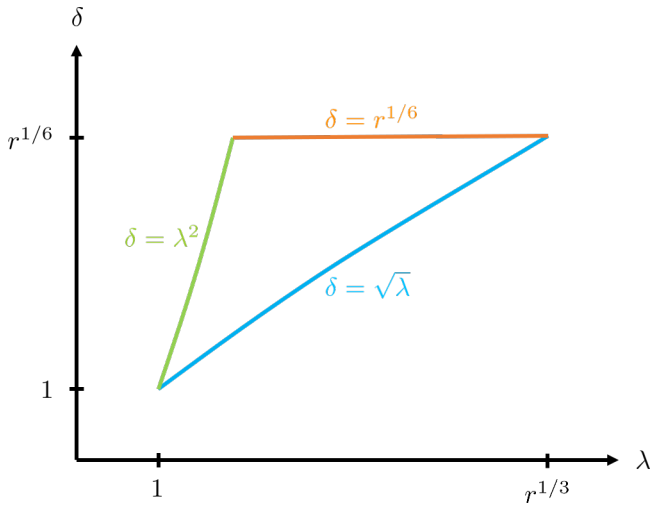


Figure 4.1: Triangular region $\mathcal{T}$ in (λ, δ) space enclosed by the three constraints: $\delta \leq r^{1/6}$, $\delta \leq \lambda^2$, and $\delta \geq \sqrt{\lambda}$.

so that the effective energy over arbitrary domain patterns is given by

$$W_{\text{eff}}(\boldsymbol{F}) = \min_{\lambda,\delta\in\mathcal{T}} \min_{\boldsymbol{Q}\in SO(3)} f(\boldsymbol{F}^\top \boldsymbol{Q}\boldsymbol{G}_0^{-1}\boldsymbol{Q}^\top \boldsymbol{F}). \tag{4.8}$$

Theorem 1. *With the definitions above,*

$$W_{\mathit{eff}}(\boldsymbol{F}) = W^{qc}(\boldsymbol{F}). \tag{4.9}$$

Proof. Let s be the largest singular value of $\boldsymbol{F}$ and t the product of the largest singular values of $\boldsymbol{F}$. Then,

$$W_{\text{eff}}(\boldsymbol{F}) = \min_{\lambda,\delta\in\mathcal{T}} f\begin{pmatrix} \left(\frac{s}{\lambda}\right)^2 & & \\ & \left(\frac{t\lambda}{\delta s}\right)^2 & \\ & & \left(\frac{\delta}{t}\right)^2 \end{pmatrix}. \tag{4.10}$$

Now, recall that

$$f(\boldsymbol{A}) = \sum_i c_i(\text{tr}\boldsymbol{A} - 3)^{p_i} + \sum_j d_j(\text{tr}(\text{cof}\boldsymbol{A}) - 3)^{q_j} \tag{4.11}$$

where $p_i, q_j \geq 1$ so that

$$W_{\text{PNE}}(\boldsymbol{F}, \lambda, \delta) = \sum_i c_i \left(\frac{s^2}{\lambda^2} + \frac{t^2\lambda^2}{\delta^2 s^2} + \frac{\delta^2}{t^2} - 3\right)^{p_i} + \sum_j d_j \left(\frac{\lambda^2}{s^2} + \frac{\delta^2 s^2}{t^2\lambda^2} + \frac{t^2}{\delta^2} - 3\right)^{q_j} \tag{4.12}$$

and

$$W_{\text{eff}}(\boldsymbol{F}) = \min_{\lambda,\delta\in\mathcal{T}} \left(\sum_i c_i \left(\frac{s^2}{\lambda^2} + \frac{t^2\lambda^2}{\delta^2 s^2} + \frac{\delta^2}{t^2} - 3\right)^{p_i} + \sum_j d_j \left(\frac{\lambda^2}{s^2} + \frac{\delta^2 s^2}{t^2\lambda^2} + \frac{t^2}{\delta^2} - 3\right)^{q_j}\right). \tag{4.13}$$

In light of the constraint $\lambda, \delta \in \mathcal{T}$, we have multiple cases.

Case 1: Attained minimum. We solve

$$\frac{\partial W_{\text{eff}}}{\partial \lambda} = \frac{\partial W_{\text{eff}}}{\partial \delta} = 0. \tag{4.14}$$

Since $c_i, d_j > 0, p_i, q_j \geq 1$, it is sufficient (and necessary in a generic sense) that

$$\frac{\partial}{\partial \lambda}\left(\frac{s^2}{\lambda^2} + \frac{t^2\lambda^2}{\delta^2 s^2} + \frac{\delta^2}{t^2}\right) = 0, \tag{4.15}$$

$$\frac{\partial}{\partial \delta}\left(\frac{s^2}{\lambda^2} + \frac{t^2\lambda^2}{\delta^2 s^2} + \frac{\delta^2}{t^2}\right) = 0, \tag{4.16}$$

$$\frac{\partial}{\partial \lambda}\left(\frac{\lambda^2}{s^2} + \frac{\delta^2 s^2}{t^2\lambda^2} + \frac{t^2}{\delta^2}\right) = 0, \tag{4.17}$$

$$\frac{\partial}{\partial \delta}\left(\frac{\lambda^2}{s^2} + \frac{\delta^2 s^2}{t^2\lambda^2} + \frac{t^2}{\delta^2}\right) = 0, \tag{4.18}$$

or

$$\frac{\partial W}{\partial \lambda} = 0 \to \frac{\lambda^4}{s^4} = \frac{\delta^2}{t^2}, \quad \frac{\partial W}{\partial \delta} = 0 \to \frac{\lambda^2}{s^2} = \frac{\delta^4}{t^4} \iff \lambda = s, \delta = t \implies W^{\text{eff}}(F) = 0. \tag{4.19}$$

This is possible if and only if $s, t \in \mathcal{T}$. Recalling that $W^{\text{qc}} = 0$ when $s, t \in \mathcal{T}$, we conclude

$$W_{\text{eff}}(F) = W^{\text{qc}}(F) \quad \text{in } L. \tag{4.20}$$

Case 2: $t > r^{1/6}$. We set $\delta = r^{1/6}$ and solve

$$\frac{\partial W_{\text{eff}}}{\partial \lambda} = 0. \tag{4.21}$$

Arguing as before, we conclude

$$\frac{\lambda}{s} = \frac{r^{1/12}}{t^{1/2}} \tag{4.22}$$

which implies

$$W_{\text{eff}}(F) = \sum_i c_i \left(2\frac{t}{r^{1/6}} + \frac{r^{1/3}}{t^2} - 3\right)^{p_i} + \sum_j d_j \left(2\frac{r^{1/6}}{t} + \frac{t^2}{r^{1/3}} - 3\right)^{q_j}. \tag{4.23}$$

Note that this coincides with the expression for W^{qc} in M. However, for $\delta = r^{1/6}$ and λ according to (4.22), $\lambda, \delta \in \mathcal{T}$ if and only if $1 \leq s/t^{1/2} \leq r^{1/4}$ or $t \leq s^2 \leq r^{1/2}t$. By assumption, $t > r^{1/6}$. So this is the region M. We conclude,

$$W_{\text{eff}}(F) = W^{\text{qc}}(F) \quad \text{in } M. \tag{4.24}$$

Case 3: $s^2 > r^{1/2}t,\ t > r^{1/6}$**.** Note that this is the region S. We set $\lambda = r^{1/3}, \delta = r^{1/6}$, and it is easy to verify that

$$W_{\text{eff}}(F) = W^{\text{qc}}(F) \quad \text{in } S. \tag{4.25}$$

□

Constitutive relation

The effective or relaxed energy (4.8) is obtained by assuming that the microstructure evolves instantaneously to minimize the energy. However, microstructure evolves according to a kinetic process which is dissipative. Further, some domains may be locally pinned, and this introduces a hardening energy. Finally, there is viscosity associated with the polymer network. All of these considerations motivate the following constitutive relation. We describe this for the isothermal situation where the temperature is fixed. However, this is easily generalized to a general temperature-dependent situation.

We assume that the state of a non-ideal isotropic-genesis polydomain nematic elastomer is described by the deformation gradient $\boldsymbol{F}$, internal variables λ, δ, and temperature T. We postulate that the stored energy density of a non-ideal isotropic-genesis polydomain nematic elastomer is given by

$$W(\boldsymbol{F}, \lambda, \delta, T) = W_{\text{PNE}}(\boldsymbol{F}, \lambda, \delta, T) + W_{\text{h}}(\lambda, \delta, T), \tag{4.26}$$

where

$$W_{\text{PNE}}(\boldsymbol{F}, \lambda, \delta, T) = \sum_i c_i \left(\frac{s^2}{\lambda^2} + \frac{t^2\lambda^2}{\delta^2 s^2} + \frac{\delta^2}{t^2} - 3 \right)^{p_i} + \sum_j d_j \left(\frac{\lambda^2}{s^2} + \frac{\delta^2 s^2}{t^2\lambda^2} + \frac{t^2}{\delta^2} - 3 \right)^{q_j} \tag{4.27}$$

as before, and

$$W_{\text{h}}(\lambda, \delta, T) = C \frac{\delta - 1}{(r^{1/6} - \delta)^k} \tag{4.28}$$

is the hardening energy. The form of the hardening is chosen to penalize $\delta \to r^{1/6}$, i.e. the completion of the polydomain-to-monodomain transition. The moduli c_i, d_i as well as the parameter r may depend on temperature.

The Cauchy stress is given by

$$\boldsymbol{\sigma}(\boldsymbol{F}, \lambda, \delta) = -p\boldsymbol{I} + \frac{\partial W_{\text{PNE}}}{\partial \boldsymbol{F}}(\boldsymbol{F}, \lambda, \delta)\boldsymbol{F}^{\top} + \beta \boldsymbol{d}, \tag{4.29}$$

where p is an unknown pressure, $\boldsymbol{d} = \frac{1}{2}(\dot{\boldsymbol{F}}\boldsymbol{F}^{-1} + \boldsymbol{F}^{-\top}\dot{\boldsymbol{F}}^{\top})$ is the rate-of-deformation tensor, and β is the viscosity. The microstructure parameters λ, δ evolve according to the equations

$$\begin{cases} \dot{\lambda} = -\alpha_\lambda \dfrac{\partial}{\partial \lambda}(W_{\mathrm{PNE}} + W_{\mathrm{h}}) \\ \qquad\qquad\qquad\qquad\qquad \text{subject to } \lambda, \delta \in \mathcal{T}. \\ \dot{\delta} = -\alpha_\delta \dfrac{\partial}{\partial \delta}(W_{\mathrm{PNE}} + W_{\mathrm{h}}) \end{cases} \tag{4.30}$$

Note that r, and hence $\mathcal{T}$, depends on temperature. It is convenient to introduce a rate-of-dissipation potential for the microstructure evolution

$$D(\dot{\lambda}, \dot{\delta}, d) = \frac{1}{2}\left(\alpha_\lambda |\dot{\lambda}|^2 + \alpha_\delta |\dot{\delta}|^2\right). \tag{4.31}$$

If we discretize the evolution equation by a backward Euler (implicit) time discretization, we can update the variables as

$$\begin{aligned} \lambda^{n+1}, \delta^{n+1} = \underset{\lambda^{n+1}, \delta^{n+1} \in \mathcal{T}}{\operatorname{argmin}} \Bigg[& W_{\mathrm{PNE}}(\boldsymbol{F}, \lambda^{n+1}, \delta^{n+1}, T) + W_{\mathrm{h}}(\lambda^{n+1}, \delta^{n+1}, T) \\ & + \Delta t D\left(\frac{\lambda^{n+1} - \lambda^n}{\Delta t}, \frac{\delta^{n+1} - \delta^n}{\Delta t}\right) \Bigg]. \end{aligned} \tag{4.32}$$

Useful calculation It is useful to compute the rotation associated with the minimization in (4.8). Let λ_i be the principal values of $\boldsymbol{F}$ with $\lambda_1 \geq \lambda_2 \geq \lambda_3$. Let

$$\boldsymbol{C} = \boldsymbol{F}^\top \boldsymbol{F} = \sum_{i=1}^{3} \lambda_i^2 \boldsymbol{N}_i \otimes \boldsymbol{N}_i, \quad \boldsymbol{b} = \boldsymbol{F}\boldsymbol{F}^\top = \sum_{i=1}^{3} \lambda_i^2 \boldsymbol{n}_i \otimes \boldsymbol{n}_i. \tag{4.33}$$

It follows that

$$F = \sum_{i=1}^{3} \lambda_i \boldsymbol{n}_i \otimes \boldsymbol{N}_i. \tag{4.34}$$

Set

$$\boldsymbol{G}_0 = \sum_{i=1}^{3} \xi_i \boldsymbol{e}_i \otimes \boldsymbol{e}_i, \tag{4.35}$$

where $\xi_1 \geq \xi_2 \geq \xi_3$. Therefore,

$$\boldsymbol{F}^\top \boldsymbol{Q} \boldsymbol{G}_0^{-1} \boldsymbol{Q}^\top \boldsymbol{F} = \sum_{i,j,k} \lambda_i \lambda_k \xi_j^{-1} (\boldsymbol{n}_i \cdot \boldsymbol{Q}\boldsymbol{e}_j)(\boldsymbol{n}_k \cdot \boldsymbol{Q}\boldsymbol{e}_j) \boldsymbol{N}_i \otimes \boldsymbol{N}_k, \tag{4.36}$$

$$(\boldsymbol{F}^\top \boldsymbol{Q} \boldsymbol{G}_0^{-1} \boldsymbol{Q}^\top \boldsymbol{F})^{-\top} = \sum_{i,j,k} \lambda_i^{-1} \lambda_k^{-1} \xi_j (\boldsymbol{n}_i \cdot \boldsymbol{Q}\boldsymbol{e}_j)(\boldsymbol{n}_k \cdot \boldsymbol{Q}\boldsymbol{e}_j) \boldsymbol{N}_i \otimes \boldsymbol{N}_k. \tag{4.37}$$

Now, in light of (4.11), maximizing $f(\boldsymbol{A})$ over $\boldsymbol{A}$ is equivalent to maximizing the trace of $\boldsymbol{A}$ and $\operatorname{cof}(\boldsymbol{A})$. Examining the above, we see that we maximize the trace of $\boldsymbol{F}^\top \boldsymbol{Q}\boldsymbol{G}_0^{-1}\boldsymbol{F}$, $\operatorname{cof}(\boldsymbol{F}^\top \boldsymbol{Q}\boldsymbol{G}_0^{-1}\boldsymbol{F}) = (\boldsymbol{F}^\top \boldsymbol{Q}\boldsymbol{G}_0^{-1}\boldsymbol{F})^{-\top}$ exactly when

$$\boldsymbol{n}_i = \boldsymbol{Q}\boldsymbol{e}_i, \quad i = 1, 2, 3. \tag{4.38}$$

Thus, the maximizing $\boldsymbol{Q}$ is

$$\boldsymbol{Q} = \sum_{i=1}^{3} \boldsymbol{n}_i \otimes \boldsymbol{e}_i \tag{4.39}$$

so that for the maximizing $\boldsymbol{Q}$,

$$\boldsymbol{G} = \sum_{i=1}^{3} \xi_i \boldsymbol{n}_i \otimes \boldsymbol{n}_i, \tag{4.40}$$

i.e. $\boldsymbol{G}$ shares an eigenbasis with $\boldsymbol{b}$ and

$$\boldsymbol{F}^\top \boldsymbol{Q}\boldsymbol{G}_0^{-1}\boldsymbol{Q}^\top \boldsymbol{F} = \sum_{i=1}^{3} \lambda_i^2 \xi_i^{-1} \boldsymbol{N}_i \otimes \boldsymbol{N}_i, \tag{4.41}$$

$$(\boldsymbol{F}^\top \boldsymbol{Q}\boldsymbol{G}_0^{-1}\boldsymbol{Q}^\top \boldsymbol{F})^{-\top} = \sum_{i=1}^{3} \lambda_i^{-2} \xi_i \boldsymbol{N}_i \otimes \boldsymbol{N}_i. \tag{4.42}$$

4.3 Validation of the model

We now validate the model against the experiments of Tokumoto, Takabe, and Urayama, as reported in Tokumoto *et al.* [71]. They subjected sheets of isotropic-genesis polydomain nematic elastomers to biaxial extension while leaving the faces of the sheet traction-free, i.e. deformations of the form

$$\boldsymbol{F} = \begin{pmatrix} \lambda_x & 0 & 0 \\ 0 & \lambda_y & 0 \\ 0 & 0 & (\lambda_x \lambda_y)^{-1} \end{pmatrix} \tag{4.43}$$

where λ_x, λ_y are imposed stretches. In all their experiments $\lambda_x \lambda_y > 1$, and we assume that $\lambda_x > \lambda_y$. Therefore,

$$s = \lambda_x, t = \lambda_x \lambda_y. \tag{4.44}$$

We neglect viscosity ($\beta = 0$), and so the Cauchy stress $\boldsymbol{\sigma}$ is

$$\begin{pmatrix} -p + \mu_1 \frac{s^2}{\lambda^2} + \mu_2 \left(\frac{s^2\delta^2}{t^2\lambda^2} + \frac{t^2}{\delta^2} \right) & 0 & 0 \\ 0 & -p + \mu_1 \frac{t^2\lambda^2}{s^2\delta^2} + \mu_2 \left(\frac{\lambda^2}{s^2} + \frac{t^2}{\delta^2} \right) & 0 \\ 0 & 0 & -p + \mu_1 \frac{\delta^2}{t^2} + \mu_2 \left(\frac{\lambda^2}{s^2} + \frac{s^2\delta^2}{t^2\lambda^2} \right) \end{pmatrix}.$$

Since the faces of the sheet are traction-free, $\sigma_{33} = 0$. It follows,

$$p = \mu_1 \frac{\delta^2}{t^2} + \mu_2 \left(\frac{\lambda^2}{s^2} + \frac{s^2\delta^2}{t^2\lambda^2} \right), \tag{4.45}$$

and the two non-zero components of stress are

$$\sigma_{11} = \mu_1 \left(\frac{s^2}{\lambda^2} - \frac{\delta^2}{t^2} \right) + \mu_2 \left(\frac{t^2}{\delta^2} - \frac{\lambda^2}{s^2} \right) \tag{4.46}$$

$$\sigma_{22} = \mu_1 \left(\frac{t^2\lambda^2}{s^2\delta^2} - \frac{\delta^2}{t^2} \right) + \mu_2 \left(\frac{t^2}{\delta^2} - \frac{s^2\delta^2}{t^2\lambda^2} \right). \tag{4.47}$$

It remains to solve for the evolution of the internal variables λ, δ. We do so by implementing (4.32) using `MATLAB`. We use the Bladon-Warner-Terentjev form (generalization of the neo-Hookean), where $c_1 = \mu_1/2, p_1 = 1$ and $c_i, d_j = 0$ for $i > 1, j \geq 1$. Table 4.1 summarizes the material parameters used. In `MATLAB`, a gradient descent code was implemented to fit the various parameters in the theoretical model to the experimental data.

Table 4.1: Material properties used in `MATLAB` simulations

Shear modulus	$\mu_1 = 4.93752 \cdot 10^4$ Pa
LCE anisotropy parameter	$r = 9.1393$
Hardening coefficient	$C = 298$ Pa
Hardening exponent	$k = 2$
Dissipation property	$\alpha_\delta = 2.1838 \cdot 10^7$ Pa
Dissipation property	$\alpha_\lambda = \frac{\alpha_\delta}{100} = 2.1838 \cdot 10^5$ Pa
Exponent in dissipation potential D	$p = 2$

We explore four specific deformations in the following sections: uniaxial (U) extension, planar extension (PE), equibiaxial (EB) extension, and unequal-biaxial (UB) extension. The general deformation gradient, following from Equation 4.43 and 4.44, is $\boldsymbol{F} = \mathrm{diag}(s, t/s, 1/t)$. Figs. 4.2 – 4.7 are plots for each individual deformation with more detail. In each figure, the top left subplot depicts the stress-stretch curve, the top right plot shows the (red) path that the internal variables take through the (black) triangular region $\mathcal{T}$, the bottom left plot shows the internal variables as a function of stretch, and the bottom right plot shows the energy density as a function of stretch.

Uniaxial extension (U) In the uniaxial case, the body is subjected to uniaxial stress in the x-direction and traction-free in the y- and z-directions. Thus, $\sigma_{22} = \sigma_{33} = 0$. The relationship between the stretches is $t = \sqrt{s}$. Thus, the deformation gradient is $\boldsymbol{F} = \mathrm{diag}(s, 1/\sqrt{s}, 1/\sqrt{s})$.

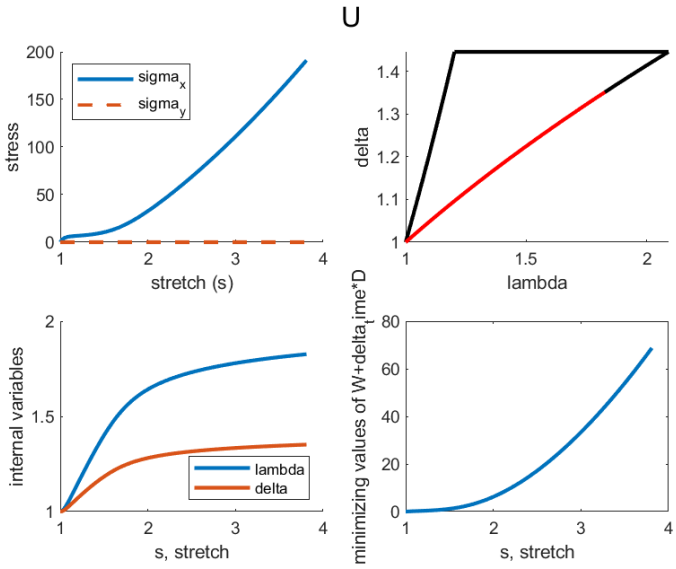


Figure 4.2: Uniaxial extension.

Planar extension (PE) In the planar extension deformation, the stretch in the y-direction is fixed at a ratio of 1, yielding $\lambda_y = 1$, and the body is traction-free in the z-direction. Thus, we have $t = s$, and the deformation gradient is $\boldsymbol{F} = \mathrm{diag}(s, 1, 1/s)$.

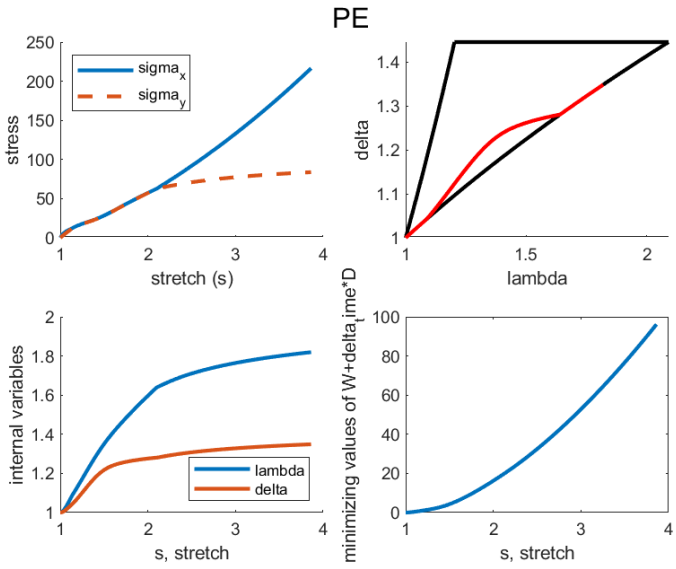


Figure 4.3: Planar extension.

Equibiaxial extension (EB) In the equibiaxial deformation, the stretch in the x-direction and y-direction are equal, and the body is traction-free in the z-direction. Thus, $s = \frac{t}{s}$ or $t = s^2$, and the deformation gradient is $\boldsymbol{F} = \mathrm{diag}(s, s, 1/s^2)$.

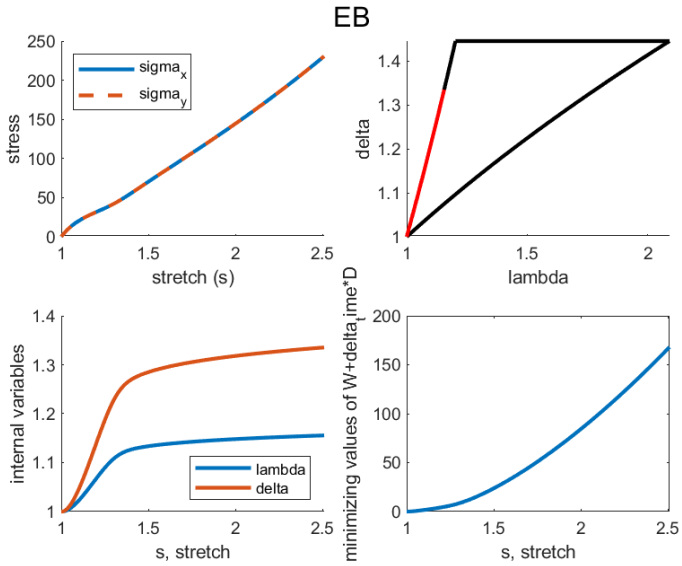


Figure 4.4: Equibiaxial extension.

Unequal biaxial extension (UB) In the case of the unequal-biaxial extension, the body is traction-free in the z-direction. The experiments are performed by fixing the ratio of $\frac{\varepsilon_x}{\varepsilon_y} = \frac{\lambda_x - 1}{\lambda_y - 1}$ to be a constant β, where β has values equal to $5/3, 5/2, 5/1$. (Note that $\beta = 1$ recovers the equibiaxial case.) This means that we have $\frac{s-1}{t/s-1} = \beta$, so $t = \frac{s}{\beta}(s - 1 + \beta)$. Thus, the deformation gradient is $\boldsymbol{F} = \mathrm{diag}\left(s, \frac{s-1+\beta}{\beta}, \frac{\beta}{s(s-1+\beta)}\right)$. The figures can be seen in Figures 4.5–4.7.

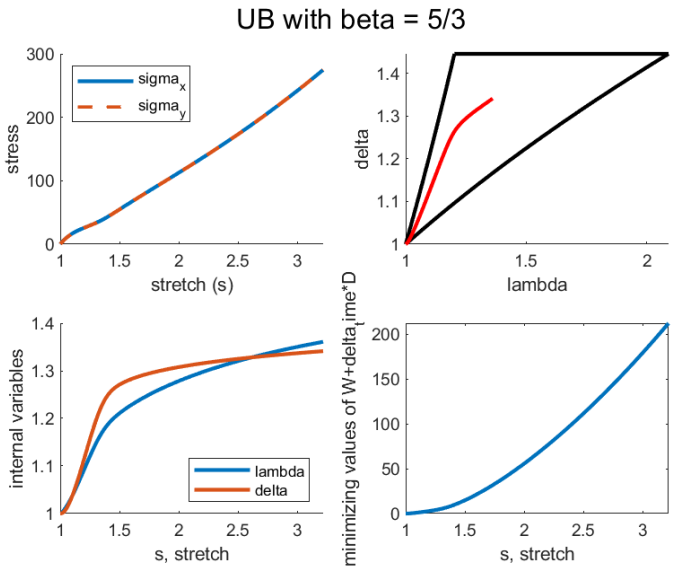


Figure 4.5: Unequal biaxial extension, $\beta = 5/3$.

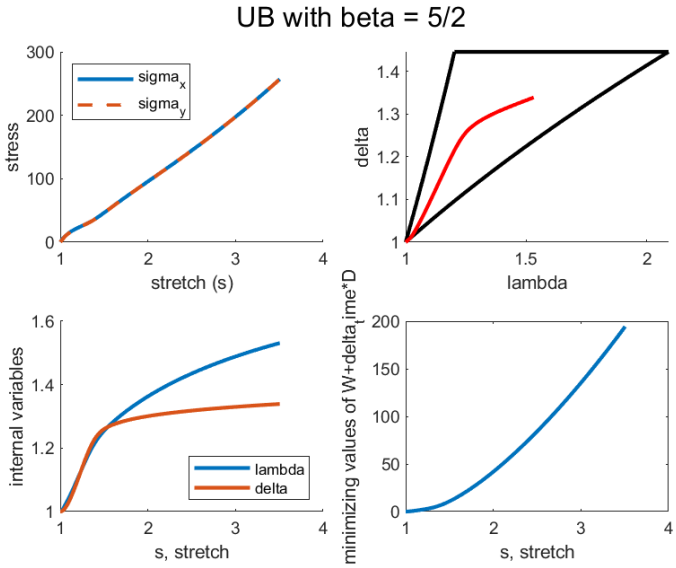


Figure 4.6: Unequal biaxial extension, $\beta = 5/2$.

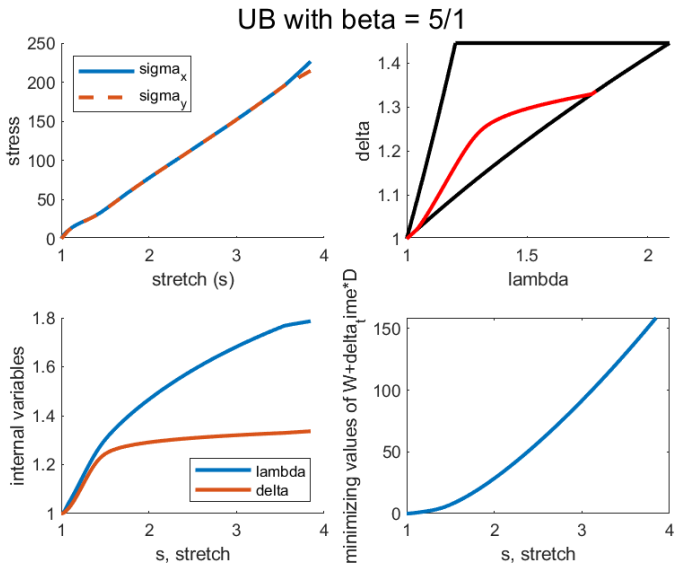


Figure 4.7: Unequal biaxial extension, $\beta = 5/1$.

Summary: liquid-like behavior Figures 4.8 and 4.9 show a comparison of the stress-stretch data, plotted for the experimental data from Urayama's research group [71] and the theoretical model for various deformations.

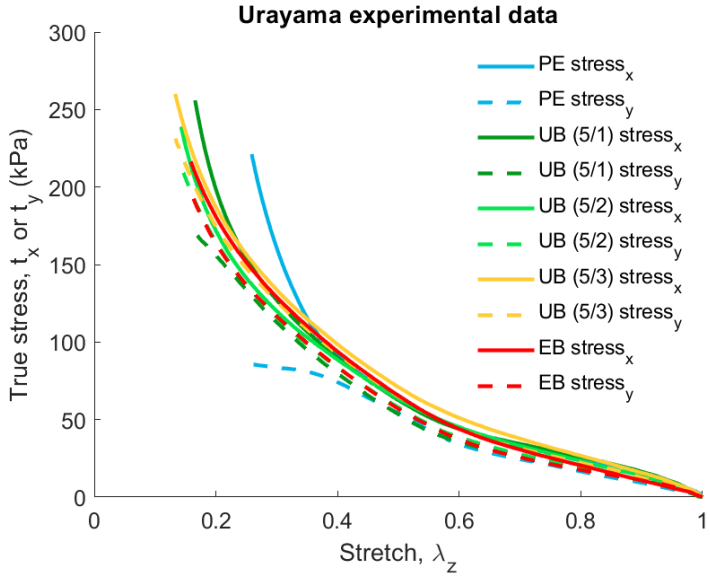


Figure 4.8: Experimental stress, plotted as a function of λ_z.

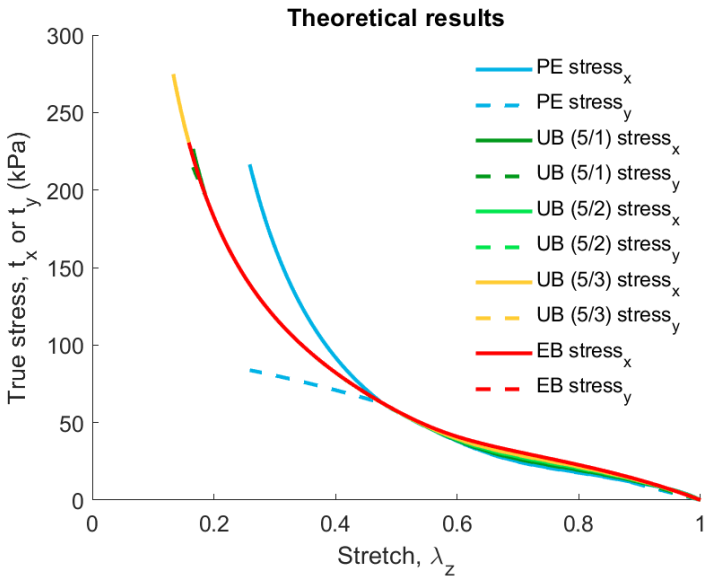
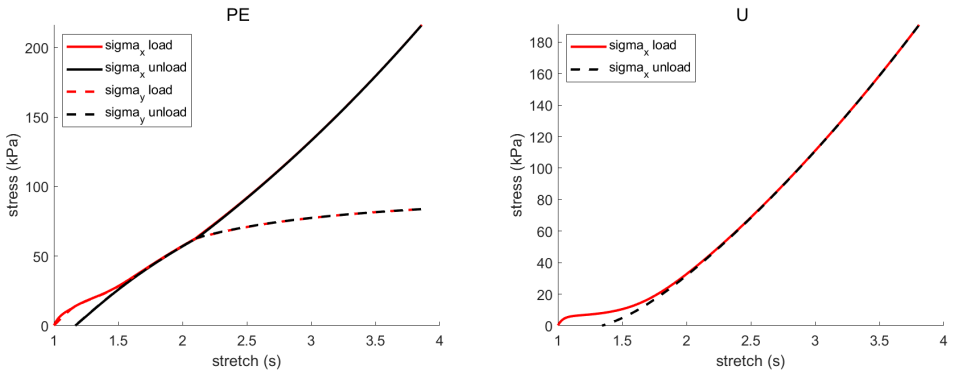


Figure 4.9: Theoretical stress, plotted as a function of λ_z.

Hysteresis Finally, Figures 4.10a and 4.10b show the load/unload curve for the U and PE deformations. The model is able to capture energy dissipation between the load and unload curves, depicted by the hysteresis between the dashed and solid curves.



(a) (b)

Figure 4.10: Load/unload curves for the (a) PE deformation and (b) U deformation.

4.4 Implementation in ABAQUS

We now discuss the model's implementation in a custom user-material, or UMAT, for use in ABAQUS. The code for the UMAT, written in FORTRAN, can be found in Appendix D.1. It is useful for numerical purposes to consider a compressible model. Further, our validation showed good agreement with the neo-Hookean. So we take the strain

energy density to be:

$$\bar{W} = \frac{\mu}{2}\left[J^{-2/3}\mathrm{tr}\,\left(\boldsymbol{F}^{\top}\boldsymbol{G}^{-1}\boldsymbol{F}\right) - 3\right] + \frac{\kappa}{2}\left(\ln J\right)^2, \tag{4.48}$$

where $\boldsymbol{F}$ is the deformation gradient, $J = \det \boldsymbol{F}$ is the determinant of the deformation gradient, and $\boldsymbol{b} = \boldsymbol{F}\boldsymbol{F}^{\top}$ is the left Cauchy-Green tensor. $\boldsymbol{G}$ is the tensor of internal variables that follow an evolution law and shares the same principal basis as $\boldsymbol{b}$. In the following sections, we show that the energy density can be rewritten in terms of a tensor $\tilde{\boldsymbol{G}}$ which shares the same principal basis as $\boldsymbol{C}$, the right Cauchy-Green tensor:

$$W = \frac{\mu}{2}\left[J^{-2/3}\mathrm{tr}\,\left(\boldsymbol{F}\tilde{\boldsymbol{G}}^{-1}\boldsymbol{F}^{\top}\right) - 3\right] + \frac{\kappa}{2}\left(\ln J\right)^2. \tag{4.49}$$

Notation

We have the right stretch tensor,

$$\boldsymbol{U} = \sum_{i=1}^{3} \lambda_i \boldsymbol{N}_i \otimes \boldsymbol{N}_i = \sqrt{\boldsymbol{C}} \tag{4.50}$$

and the right Cauchy-Green tensor

$$\boldsymbol{C} = \sum_{i=1}^{3} \lambda_i^2 \boldsymbol{N}_i \otimes \boldsymbol{N}_i. \tag{4.51}$$

From polar decomposition, we have

$$\boldsymbol{F} = \boldsymbol{R}\boldsymbol{U} = \boldsymbol{V}\boldsymbol{R}, \tag{4.52}$$

where $\boldsymbol{U}$ is the right stretch tensor as above, $\boldsymbol{R} \in SO(3)$, and $\boldsymbol{U} \in GL(3)$ symmetric and positive-definite. Because $\boldsymbol{U}$ is symmetric and real, we can write its eigendecomposition as

$$\boldsymbol{U} = \boldsymbol{Q}\boldsymbol{\Lambda}\boldsymbol{Q}^{\top}, \tag{4.53}$$

where the orthogonal matrix $\boldsymbol{Q}$ is

$$\boldsymbol{Q} = \left(\begin{array}{c|c|c} & & \\ \boldsymbol{N}_1 & \boldsymbol{N}_2 & \boldsymbol{N}_3 \\ & & \end{array}\right) \tag{4.54}$$

and

$$\boldsymbol{\Lambda} = \begin{pmatrix} \lambda_1 & & \\ & \lambda_2 & \\ & & \lambda_3 \end{pmatrix}. \tag{4.55}$$

The right Cauchy-Green tensor is

$$\boldsymbol{C} = \boldsymbol{U}^2 \tag{4.56}$$

$$= \boldsymbol{Q}\boldsymbol{\Lambda}^2\boldsymbol{Q}^\top \tag{4.57}$$

$$= \sum_{i=1}^{3} \lambda_i^2 \boldsymbol{N}_i \otimes \boldsymbol{N}_i. \tag{4.58}$$

The left Cauchy-Green tensor is

$$\boldsymbol{b} = \boldsymbol{V}^2 \tag{4.59}$$

$$= \boldsymbol{R}\boldsymbol{Q}\boldsymbol{\Lambda}^2\boldsymbol{Q}^\top\boldsymbol{R}^\top \tag{4.60}$$

$$= \sum_{i=1}^{3} \lambda_i^2 \boldsymbol{n}_i \otimes \boldsymbol{n}_i, \tag{4.61}$$

where the eigenvectors of $\boldsymbol{b}$ are related to the eigenvectors of $\boldsymbol{C}$ through $\boldsymbol{n} = \boldsymbol{R}\boldsymbol{N}$. The internal variable tensor $\boldsymbol{G}$ shares the same principal basis as $\boldsymbol{b}$:

$$\boldsymbol{G} = \boldsymbol{R}\boldsymbol{Q}\boldsymbol{\Xi}\boldsymbol{Q}^\top\boldsymbol{R}^\top \tag{4.62}$$

$$= \sum_{i=1}^{3} \xi_i^2 \boldsymbol{n}_i \otimes \boldsymbol{n}_i, \tag{4.63}$$

where

$$\boldsymbol{\Xi} = \begin{pmatrix} \xi_1 & & \\ & \xi_2 & \\ & & \xi_3 \end{pmatrix} = \begin{pmatrix} \lambda^2 & & \\ & \delta^2/\lambda^2 & \\ & & 1/\delta^2 \end{pmatrix}. \tag{4.64}$$

A related internal variable tensor $\tilde{\boldsymbol{G}}$ shares the same principal basis as $\boldsymbol{C}$:

$$\tilde{\boldsymbol{G}} = \boldsymbol{Q}\boldsymbol{\Xi}\boldsymbol{Q}^\top \tag{4.65}$$

$$= \sum_{i=1}^{3} \xi_i^2 \boldsymbol{N}_i \otimes \boldsymbol{N}_i. \tag{4.66}$$

Rewritten energy density

The original energy density for our constitutive model was

$$W = \frac{\mu}{2}\left[J^{-2/3}\mathrm{tr}\,\left(\boldsymbol{F}^\top\boldsymbol{G}^{-1}\boldsymbol{F}\right) - 3\right] + \frac{\kappa}{2}\left(\ln J\right)^2. \tag{4.67}$$

We can rewrite the energy as a function of $\tilde{\boldsymbol{G}}$:

$$W = \frac{\mu}{2}\left[J^{-2/3}\text{tr}\,\left(\boldsymbol{G}^{-1}\boldsymbol{b}\right) - 3\right] + \frac{\kappa}{2}\left(\ln J\right)^2 \tag{4.68}$$

$$= \frac{\mu}{2}\left[J^{-2/3}\text{tr}\,\left(\boldsymbol{R}\boldsymbol{Q}\boldsymbol{\Xi}^{-1}\boldsymbol{Q}^\top\boldsymbol{R}^\top\boldsymbol{R}\boldsymbol{Q}\boldsymbol{\Lambda}^2\boldsymbol{Q}^\top\boldsymbol{R}^\top\right) - 3\right] + \frac{\kappa}{2}\left(\ln J\right)^2 \tag{4.69}$$

$$= \frac{\mu}{2}\left[J^{-2/3}\text{tr}\,\left(\boldsymbol{Q}\boldsymbol{\Xi}^{-1}\boldsymbol{Q}^\top\boldsymbol{Q}\boldsymbol{\Lambda}^2\boldsymbol{Q}^\top\right) - 3\right] + \frac{\kappa}{2}\left(\ln J\right)^2 \tag{4.70}$$

$$= \frac{\mu}{2}\left[J^{-2/3}\text{tr}\,\left(\boldsymbol{F}\tilde{\boldsymbol{G}}^{-1}\boldsymbol{F}^\top\right) - 3\right] + \frac{\kappa}{2}\left(\ln J\right)^2. \tag{4.71}$$

We can also rewrite the energy as a function of the principal values:

$$W = \frac{\mu}{2}\left[J^{-2/3}\text{tr}\,\left(\boldsymbol{R}\boldsymbol{Q}\boldsymbol{\Xi}^{-1}\boldsymbol{Q}^\top\boldsymbol{R}^\top\boldsymbol{R}\boldsymbol{Q}\boldsymbol{\Lambda}^2\boldsymbol{Q}^\top\boldsymbol{R}^\top\right) - 3\right] + \frac{\kappa}{2}\left(\ln J\right)^2 \tag{4.72}$$

$$= \frac{\mu}{2}\left[J^{-2/3}\text{tr}\,\left(\boldsymbol{\Xi}^{-1}\boldsymbol{\Lambda}^2\right) - 3\right] + \frac{\kappa}{2}\left(\ln J\right)^2 \tag{4.73}$$

$$= \frac{\mu}{2}\left[J^{-2/3}\left(\frac{\lambda_1^2}{\xi_1^2} + \frac{\lambda_2^2}{\xi_2^2} + \frac{\lambda_3^2}{\xi_3^2}\right) - 3\right] + \frac{\kappa}{2}(\ln J)^2, \tag{4.74}$$

or

$$W = \frac{\mu}{2}\left[J^{-2/3}\left(\frac{\lambda_1^2}{\xi_1^2} + \frac{\lambda_2^2}{\xi_2^2} + \frac{\lambda_3^2}{\xi_3^2}\right) - 3\right] + \frac{\kappa}{2}(\ln J)^2, \tag{4.75}$$

where $J = \det F = \lambda_1\lambda_2\lambda_3$.

Adding in viscosity

Here, we note that we have added a viscous part of the stress. We assume an additive decomposition of the Piola-Kirchhoff stress $\boldsymbol{P}$ into an elastic portion and a viscosity portion:

$$\boldsymbol{P} = \boldsymbol{P}^e + \boldsymbol{P}^v, \tag{4.76}$$

where

$$\boldsymbol{P}^e = \frac{\partial W}{\partial \boldsymbol{F}} \tag{4.77}$$

and $\boldsymbol{P}^v = \beta J \boldsymbol{d}\boldsymbol{F}^{-\top}$ is the viscosity portion. Thus, we have the Cauchy stress:

$$\boldsymbol{\sigma} = \boldsymbol{\sigma}^e + \boldsymbol{\sigma}^v \tag{4.78}$$

$$= \frac{1}{J}\frac{\partial W}{\partial \boldsymbol{F}}\boldsymbol{F}^\top + \beta\boldsymbol{d}. \tag{4.79}$$

Stress measures

Second Piola-Kirchhoff stress

Let us calculate the second Piola-Kirchhoff stress. Let a symmetric, 3x3 tensor $\boldsymbol{C}$ have eigenvalues μ_i and eigenvectors $\boldsymbol{N}_i$ so that

$$\boldsymbol{C}\boldsymbol{N}_i = \mu_i\boldsymbol{N}_i \text{ (no sum over } i). \tag{4.80}$$

Then, from [37],

$$\frac{\partial \mu_i}{\partial \boldsymbol{C}} = \boldsymbol{N}_i \otimes \boldsymbol{N}_i. \tag{4.81}$$

Here, the eigenvalues of the right Cauchy-Green tensor are $\mu_i = \lambda_i^2$. Thus, we obtain

$$\boldsymbol{S} = 2\frac{\partial W}{\partial \boldsymbol{C}} \tag{4.82}$$

$$= 2\left(\frac{\partial W}{\partial \lambda_1^2}\frac{\partial \lambda_1^2}{\partial \boldsymbol{C}} + \frac{\partial W}{\partial \lambda_2^2}\frac{\partial \lambda_2^2}{\partial \boldsymbol{C}} + \frac{\partial W}{\partial \lambda_3^2}\frac{\partial \lambda_3^2}{\partial \boldsymbol{C}}\right) \tag{4.83}$$

$$= 2\left(\frac{\partial W}{\partial \lambda_1^2}\boldsymbol{N}_1 \otimes \boldsymbol{N}_1 + \frac{\partial W}{\partial \lambda_2^2}\boldsymbol{N}_2 \otimes \boldsymbol{N}_2 + \frac{\partial W}{\partial \lambda_3^2}\boldsymbol{N}_3 \otimes \boldsymbol{N}_3\right) \tag{4.84}$$

$$= \mu J^{-2/3}\left(\frac{1}{\xi_1^2}\boldsymbol{N}_1 \otimes \boldsymbol{N}_1 + \frac{1}{\xi_2^2}\boldsymbol{N}_2 \otimes \boldsymbol{N}_2 + \frac{1}{\xi_3^2}\boldsymbol{N}_3 \otimes \boldsymbol{N}_3\right) \tag{4.85}$$

$$- \frac{\mu}{3}J^{-5/3}\left(\frac{\lambda_1^2}{\xi_1^2} + \frac{\lambda_2^2}{\xi_2^2} + \frac{\lambda_3^2}{\xi_3^2}\right)\left(\frac{\lambda_2\lambda_3}{\lambda_1}\boldsymbol{N}_1 \otimes \boldsymbol{N}_1 + \frac{\lambda_1\lambda_3}{\lambda_2}\boldsymbol{N}_2 \otimes \boldsymbol{N}_2 + \frac{\lambda_1\lambda_2}{\lambda_3}\boldsymbol{N}_3 \otimes \boldsymbol{N}_3\right) \tag{4.86}$$

$$+ \kappa\frac{\ln J}{J}\left(\frac{\lambda_2\lambda_3}{\lambda_1}\boldsymbol{N}_1 \otimes \boldsymbol{N}_1 + \frac{\lambda_1\lambda_3}{\lambda_2}\boldsymbol{N}_2 \otimes \boldsymbol{N}_2 + \frac{\lambda_1\lambda_2}{\lambda_3}\boldsymbol{N}_3 \otimes \boldsymbol{N}_3\right). \tag{4.87}$$

Recognizing that

$$\tilde{\boldsymbol{G}}^{-1} = \frac{1}{\xi_1^2}\boldsymbol{N}_1 \otimes \boldsymbol{N}_1 + \frac{1}{\xi_2^2}\boldsymbol{N}_2 \otimes \boldsymbol{N}_2 + \frac{1}{\xi_3^2}\boldsymbol{N}_3 \otimes \boldsymbol{N}_3 \tag{4.88}$$

$$\mathrm{tr}(\tilde{\boldsymbol{G}}^{-1}\boldsymbol{C}) = \frac{\lambda_1^2}{\xi_1^2} + \frac{\lambda_2^2}{\xi_2^2} + \frac{\lambda_3^2}{\xi_3^2} \tag{4.89}$$

$$J\boldsymbol{C}^{-1} = \frac{\lambda_2\lambda_3}{\lambda_1}\boldsymbol{N}_1 \otimes \boldsymbol{N}_1 + \frac{\lambda_1\lambda_3}{\lambda_2}\boldsymbol{N}_2 \otimes \boldsymbol{N}_2 + \frac{\lambda_1\lambda_2}{\lambda_3}\boldsymbol{N}_3 \otimes \boldsymbol{N}_3, \tag{4.90}$$

we are left with

$$\boldsymbol{S} = \mu J^{-2/3}\tilde{\boldsymbol{G}}^{-1} - \frac{\mu}{3}J^{-5/3}\mathrm{tr}(\tilde{\boldsymbol{G}}^{-1}\boldsymbol{C})J\boldsymbol{C}^{-1} + \kappa\frac{\ln J}{J}J\boldsymbol{C}^{-1}, \tag{4.91}$$

or

$$\boldsymbol{S} = \mu J^{-2/3}\left[\tilde{\boldsymbol{G}}^{-1} - \frac{1}{3}\mathrm{tr}(\tilde{\boldsymbol{G}}^{-1}\boldsymbol{C})\boldsymbol{C}^{-1}\right] + \kappa(\ln J)\boldsymbol{C}^{-1}. \tag{4.92}$$

Note that we also reached this stress expression by computing the tensor derivative as follows:

$$\boldsymbol{S} = 2\frac{\partial W(\boldsymbol{C})}{\partial \boldsymbol{C}}, \tag{4.93}$$

where

$$W(\boldsymbol{C}) = \frac{\mu}{2}\left[(\det \boldsymbol{C})^{-1/3}\mathrm{tr}(\tilde{\boldsymbol{G}}^{-1}\boldsymbol{C}) - 3\right] + \frac{\kappa}{8}\left[\ln(\det \boldsymbol{C})\right]^2 \tag{4.94}$$

with $\frac{\partial \tilde{\boldsymbol{G}}}{\partial \boldsymbol{C}} = \boldsymbol{0}$.

First Piola-Kirchhoff stress

The first Piola-Kirchhoff stress can be computed as follows:

$$\boldsymbol{P} = \boldsymbol{F}\boldsymbol{S} \tag{4.95}$$

$$\boldsymbol{P} = \mu J^{-2/3}\left[\boldsymbol{F}\tilde{\boldsymbol{G}}^{-1} - \frac{1}{3}\mathrm{tr}\left(\tilde{\boldsymbol{G}}^{-1}\boldsymbol{C}\right)\boldsymbol{F}^{-\top}\right] + \kappa\,(\ln J)\,\boldsymbol{F}^{-\top}. \tag{4.96}$$

Cauchy stress

The Cauchy stress expression is needed for `ABAQUS`. We find

$$\boldsymbol{\sigma} = \frac{1}{J}\boldsymbol{P}\boldsymbol{F}^{\top} = \frac{1}{J}\boldsymbol{F}\boldsymbol{S}\boldsymbol{F}^{\top} \tag{4.97}$$

$$\boldsymbol{\sigma} = \mu J^{-5/3}\left[\boldsymbol{F}\tilde{\boldsymbol{G}}^{-1}\boldsymbol{F}^{\top} - \frac{1}{3}\mathrm{tr}(\tilde{\boldsymbol{G}}^{-1}\boldsymbol{C})\boldsymbol{I}\right] + \kappa\frac{\ln J}{J}\boldsymbol{I}. \tag{4.98}$$

Note that when $\tilde{\boldsymbol{G}} = \boldsymbol{I}$, the Cauchy stress recovers the neo-Hookean stress, as expected.

Kirchhoff stress

The Kirchhoff stress is simply

$$\boldsymbol{\tau} = J\boldsymbol{\sigma} = \boldsymbol{F}\boldsymbol{S}\boldsymbol{F}^{\top} \tag{4.99}$$

$$\boldsymbol{\tau} = \mu J^{-2/3}\left[\boldsymbol{F}\tilde{\boldsymbol{G}}^{-1}\boldsymbol{F}^{\top} - \frac{1}{3}\mathrm{tr}(\tilde{\boldsymbol{G}}^{-1}\boldsymbol{C})\boldsymbol{I}\right] + \kappa\,(\ln J)\,\boldsymbol{I}. \tag{4.100}$$

Derivation of the material Jacobian, `DDSDDE`

Jaumann rate of the Kirchhoff stress

The Jaumann rate of the Kirchhoff stress is

$$\overset{\triangledown}{\boldsymbol{\tau}}{}^{(JK)} = \dot{\boldsymbol{\tau}} - \boldsymbol{w}\boldsymbol{\tau} + \boldsymbol{\tau}\boldsymbol{w}, \tag{4.101}$$

where $\boldsymbol{w} = \frac{1}{2}\left(\boldsymbol{\ell} - \boldsymbol{\ell}^\top\right)$ is the spin tensor. In symbolic form, we have

$$\begin{aligned}
\overset{\triangledown}{\boldsymbol{\tau}}{}^{(JK)} &= -\frac{2\mu}{3} J^{-2/3} (\operatorname{tr} \boldsymbol{\ell}) \left[\boldsymbol{F}\tilde{\boldsymbol{G}}^{-1}\boldsymbol{F}^\top - \frac{1}{3}\operatorname{tr}\left(\tilde{\boldsymbol{G}}^{-1}\boldsymbol{C}\right)\boldsymbol{I} \right] \\
&\quad - \frac{\mu}{3} J^{-2/3} \left[\operatorname{tr}\left(\tilde{\boldsymbol{G}}^{-1}\dot{\boldsymbol{C}}\right)\delta_{ij} \right] \\
&\quad + \mu J^{-2/3} \left[\boldsymbol{d}\boldsymbol{F}\tilde{\boldsymbol{G}}^{-1}\boldsymbol{F}^\top + \boldsymbol{F}\tilde{\boldsymbol{G}}^{-1}\boldsymbol{F}^\top\boldsymbol{d} \right] \\
&\quad + \kappa (\operatorname{tr} \boldsymbol{\ell})\boldsymbol{I} \\
&\quad + \mu J^{-2/3} \left[\boldsymbol{F}\dot{\tilde{\boldsymbol{G}}}^{-1}\boldsymbol{F}^\top - \frac{1}{3}\operatorname{tr}\left(\dot{\tilde{\boldsymbol{G}}}^{-1}\boldsymbol{C}\right)\boldsymbol{I} \right].
\end{aligned} \tag{4.102}$$

Consistent Jacobian, `DDSDDE`

The consistent Jacobian matrix (called `DDSDDE` in `ABAQUS`), is defined as

$$\overset{\triangledown}{\boldsymbol{\tau}}{}^{(JK)} = J(\mathbf{DDSDDE}) : \boldsymbol{d}, \tag{4.103}$$

or

$$\overset{\triangledown}{\tau}{}^{(JK)}_{ij} = J\ \mathtt{DDSDDE}_{ijkl} d_{kl}. \tag{4.104}$$

We can rewrite $\overset{\triangledown}{\boldsymbol{\tau}}{}^{(JK)}$ in the following way so that the `DDSDDE` expression is easily extricated. Note that we will make use of Section B.2.

$$\begin{aligned}
\overset{\triangledown}{\tau}{}^{(JK)}_{ij} &= -\frac{2\mu}{3} J^{-2/3} \left[\left(\boldsymbol{F}\tilde{\boldsymbol{G}}^{-1}\boldsymbol{F}^\top\right)_{ij} \delta_{kl} - \frac{1}{3}\operatorname{tr}\left(\tilde{\boldsymbol{G}}^{-1}\boldsymbol{C}\right)\delta_{ij}\delta_{kl} \right] d_{kl} \\
&\quad - \frac{2\mu}{3} J^{-2/3} \left[\left(\boldsymbol{F}\tilde{\boldsymbol{G}}^{-1}\boldsymbol{F}^\top\right)_{kl} \delta_{ij} \right] d_{kl} \\
&\quad + \frac{\mu}{2} J^{-2/3} \left[\left(\boldsymbol{F}\tilde{\boldsymbol{G}}^{-1}\boldsymbol{F}^\top\right)_{lj} \delta_{ik} + \left(\boldsymbol{F}\tilde{\boldsymbol{G}}^{-1}\boldsymbol{F}^\top\right)_{ik} \delta_{jl} + \left(\boldsymbol{F}\tilde{\boldsymbol{G}}^{-1}\boldsymbol{F}^\top\right)_{il} \delta_{jk} + \left(\boldsymbol{F}\tilde{\boldsymbol{G}}^{-1}\boldsymbol{F}^\top\right)_{jk} \delta_{il} \right] d_{kl} \\
&\quad + \kappa \left[\delta_{ij}\delta_{kl} \right] d_{kl} \\
&\quad + \mu J^{-2/3} \left[\left(\boldsymbol{F}\dot{\tilde{\boldsymbol{G}}}^{-1}\boldsymbol{F}^\top\right)_{ij} - \frac{1}{3}\operatorname{tr}\left(\dot{\tilde{\boldsymbol{G}}}^{-1}\boldsymbol{C}\right)\delta_{ij} \right].
\end{aligned} \tag{4.105}$$

For the last line, we have used the auxiliary Section B.3, where

$$A_{ij} = \left[\left(\boldsymbol{F}\dot{\tilde{\boldsymbol{G}}}^{-1}\boldsymbol{F}^\top\right)_{ij} - \frac{1}{3}\operatorname{tr}\left(\dot{\tilde{\boldsymbol{G}}}^{-1}\boldsymbol{C}\right)\delta_{ij} \right]. \tag{4.106}$$

Thus, the material Jacobian, from Equation 4.104, is

$$\begin{aligned}
\texttt{DDSDDE}_{ijkl} = &-\frac{2\mu}{3}J^{-5/3}\left[\left(\boldsymbol{F}\tilde{\boldsymbol{G}}^{-1}\boldsymbol{F}^{\top}\right)_{ij}\delta_{kl} + \left(\boldsymbol{F}\tilde{\boldsymbol{G}}^{-1}\boldsymbol{F}^{\top}\right)_{kl}\delta_{ij}\right] \\
&+ \frac{2\mu}{9}J^{-5/3}\mathrm{tr}\left(\tilde{\boldsymbol{G}}^{-1}\boldsymbol{C}\right)\delta_{ij}\delta_{kl} + \frac{\kappa}{J}\delta_{ij}\delta_{kl} \\
&+ \frac{\mu}{2}J^{-5/3}\left[\left(\boldsymbol{F}\tilde{\boldsymbol{G}}^{-1}\boldsymbol{F}^{\top}\right)_{lj}\delta_{ik} + \left(\boldsymbol{F}\tilde{\boldsymbol{G}}^{-1}\boldsymbol{F}^{\top}\right)_{ik}\delta_{jl}\right] \\
&+ \frac{\mu}{2}J^{-5/3}\left[\left(\boldsymbol{F}\tilde{\boldsymbol{G}}^{-1}\boldsymbol{F}^{\top}\right)_{il}\delta_{jk} + \left(\boldsymbol{F}\tilde{\boldsymbol{G}}^{-1}\boldsymbol{F}^{\top}\right)_{jk}\delta_{il}\right] \\
&+ \mu J^{-5/3}\left[\frac{1}{|\boldsymbol{d}|^2}\left(A_{ij}d_{kl} + d_{ij}A_{kl}\right) - \frac{1}{|\boldsymbol{d}|^4}A_{mn}d_{mn}d_{ij}d_{kl}\right].
\end{aligned} \tag{4.107}$$

Note that we can check that the material Jacobian satisfies the symmetries

$$\texttt{DDSDDE}_{ijkl} = \texttt{DDSDDE}_{klij} = \texttt{DDSDDE}_{ijlk}. \tag{4.108}$$

We can also confirm that when $\tilde{\boldsymbol{G}} = \boldsymbol{I}$ (and therefore $\dot{\tilde{\boldsymbol{G}}} = \boldsymbol{0}$), we recover the neo-Hookean material Jacobian.

Also note that in approximating the spatial velocity gradient $\boldsymbol{\ell}$ and consequently the rate-of-deformation tensor $\boldsymbol{d}$ and spin tensor $\boldsymbol{w}$, we use the approximation from Weber, Anand (1999) [78]:

$$\boldsymbol{\ell}_n = \frac{1}{\Delta t}\left(\boldsymbol{F}_n\boldsymbol{F}_{n-1}^{-1} - \boldsymbol{I}\right). \tag{4.109}$$

We considered other approximations; the first was a finite difference approximation:

$$\dot{\boldsymbol{F}}_n = \frac{\boldsymbol{F}_n - \boldsymbol{F}_{n-1}}{\Delta t}, \tag{4.110}$$

and the second used the increment `DSTRAN`, as suggested in Nguyen and Waas [52]:

$$\boldsymbol{d} = \frac{\texttt{DSTRAN}}{\Delta t}. \tag{4.111}$$

Optimization using `NLOpt`

Inside the `UMAT`, we will be using an external optimization algorithm called `NLOpt` (documentation **here**) to conduct the constrained optimization of the internal variables. For algorithms that use a gradient-based method, we need to compute the derivatives of the objective function with respect to the internal variables λ and δ.

Recall the minimization problem from Equation 4.32. Written explicitly, we have

$$(\lambda,\delta) = \underset{(\lambda,\delta)\in\mathcal{T}}{\arg\inf} \left\{ \frac{\mu}{2}\left[J^{-2/3}\left(\frac{\lambda_1^2}{\xi_1^2}+\frac{\lambda_2^2}{\xi_2^2}+\frac{\lambda_3^2}{\xi_3^2}\right)-3\right] + \frac{\kappa}{2}(\ln J)^2 \right. \\ \left. + C\frac{\delta-1}{(r^{1/6}-\delta)^k} + \Delta t\left(\frac{\alpha_\lambda}{2}\dot{\lambda}^p + \frac{\alpha_\delta}{2}\dot{\delta}^p\right)\right\}, \tag{4.112}$$

where λ_i $(i = 1:3)$ are the principal stretches of $\boldsymbol{F}$, $J = \det \boldsymbol{F} = \lambda_1\lambda_2\lambda_3$, $\tilde{\boldsymbol{G}} = \boldsymbol{Q}\boldsymbol{\Xi}\boldsymbol{Q}^\top$, and $\xi_1 = \lambda^2, \xi_2 = \frac{\delta^2}{\lambda^2}, \xi_3 = \frac{1}{\delta^2}$. We wish for the internal variables to evolve at different speeds, so the parameters of dissipation must satisfy the relation

$$\alpha_\lambda \ll \alpha_\delta. \tag{4.113}$$

Strain rate

This is the loading for a uniaxial block. We begin with the equation relating stretch and strain:

$$\varepsilon = \frac{L-L_0}{L_0} = \frac{\Delta L}{L_0} = \frac{L}{L_0} - 1 = \lambda - 1. \tag{4.114}$$

We choose a reference length for the model L_0, e.g. in some simulations we have $L_0 = 0.01$m, or 1 cm. We choose the desired final strain $\varepsilon_f = 3$, or a desired final stretch of $\lambda_f = \varepsilon_f + 1 = 4$. Then the distance that the block needs to stretch is $\Delta L = \varepsilon_f L_0$. The velocity that the block needs to move is determined by $v = \dot{\varepsilon} L_0$, or in other words the strain rate is determined by $\dot{\varepsilon} = \frac{v}{L_0}$. The total time for the simulation is $t^* = \frac{\Delta L}{v}$, and the stretch as a function of time $\lambda(t)$ is

$$\lambda(t) = \dot{\varepsilon} t + 1 = \frac{v}{L_0} t + 1 = \frac{\varepsilon_f}{t^*} t + 1. \tag{4.115}$$

Eigenvector/eigenvalue coding

We find the eigendecomposition of $\boldsymbol{C}$, the right Cauchy-Green tensor using a subroutine called `Jacobi`. We find the eigenvalues of $\boldsymbol{C}$ and then order them from greatest to least (using a bubble sort): $\lambda_1^2 \geq \lambda_2^2 \geq \lambda_3^2$, and the square root of the eigenvalues are collected into the principal stretch matrix is

$$\boldsymbol{\Lambda} = \begin{pmatrix} \lambda_1 & & \\ & \lambda_2 & \\ & & \lambda_3 \end{pmatrix}. \tag{4.116}$$

We also order the corresponding normalized eigenvectors $\boldsymbol{N}_1, \boldsymbol{N}_2, \boldsymbol{N}_3$. The eigenvectors are collected into the orthogonal eigenvector matrix $\boldsymbol{Q}$:

$$\boldsymbol{Q} = \left(\begin{array}{c|c|c} & & \\ \boldsymbol{N}_1 & \boldsymbol{N}_2 & \boldsymbol{N}_3 \\ & & \end{array} \right). \tag{4.117}$$

Then $\boldsymbol{C} = \boldsymbol{Q}\boldsymbol{\Lambda}\boldsymbol{Q}^\top$. Recall the energy density (where $J = \det F = \lambda_1\lambda_2\lambda_3$)

$$W = \frac{\mu}{2}\left[J^{-2/3}\left(\frac{\lambda_1^2}{\xi_1^2} + \frac{\lambda_2^2}{\xi_2^2} + \frac{\lambda_3^2}{\xi_3^2}\right) - 3\right] + \frac{\kappa}{2}(\ln J)^2. \tag{4.118}$$

So each λ_i correlates with ξ_i, $i = 1:3$. Thus we have the internal variable matrix $\tilde{\boldsymbol{G}} = \boldsymbol{Q}\boldsymbol{\Xi}\boldsymbol{Q}^\top$, where $\boldsymbol{Q}$ is the same as Equation 4.117 and the eigenvalues ξ_1, ξ_2, ξ_3 correlate to $\lambda_1 \geq \lambda_2 \geq \lambda_3$:

$$\boldsymbol{\Xi} = \begin{pmatrix} \xi_1 & & \\ & \xi_2 & \\ & & \xi_3 \end{pmatrix} = \begin{pmatrix} \lambda^2 & & \\ & \frac{\delta^2}{\lambda^2} & \\ & & \frac{1}{\delta^2} \end{pmatrix}. \tag{4.119}$$

This allows us to compute the quantity $\boldsymbol{F}\tilde{\boldsymbol{G}}^{-1}\boldsymbol{F}^\top$ and therefore the energy density.

4.5 Results from the ABAQUS implementation

Verification of the model

We compute the results for a single element undergoing the following deformations: uniaxial (U), planar extension (PE), equibiaxial (EB), and unequal biaxial (UB) with stretch ratios of 5/3, 5/2, and 5/1. In all of these cases, we have a match between the theoretical model, solved in MATLAB, and the ABAQUS simulation results. The parameters used in these results are the same as in Table 4.1, with additional parameters shown in Table 4.2. In each deformation, we see that the results match.

Table 4.2: Additional material properties used in ABAQUS simulations

Bulk modulus	$\kappa = 4.93752 \cdot 10^8$ Pa
Viscosity	$\beta = 1 \cdot 10^6$ Pa $\cdot$ s
Proportionality constant for $\dot{\tilde{\boldsymbol{G}}}^{-1} = \alpha\dot{\boldsymbol{C}}^{-1}$	$\alpha = 5 \cdot 10^-6$

We repeated the calculations with multiple elements (e.g. cubes meshed with $5\times5\times5$ and $10\times10\times10$ elements) with the same boundary conditions and loading conditions as above and confirmed that the results agreed perfectly with the $1\times1\times1$ simulations.

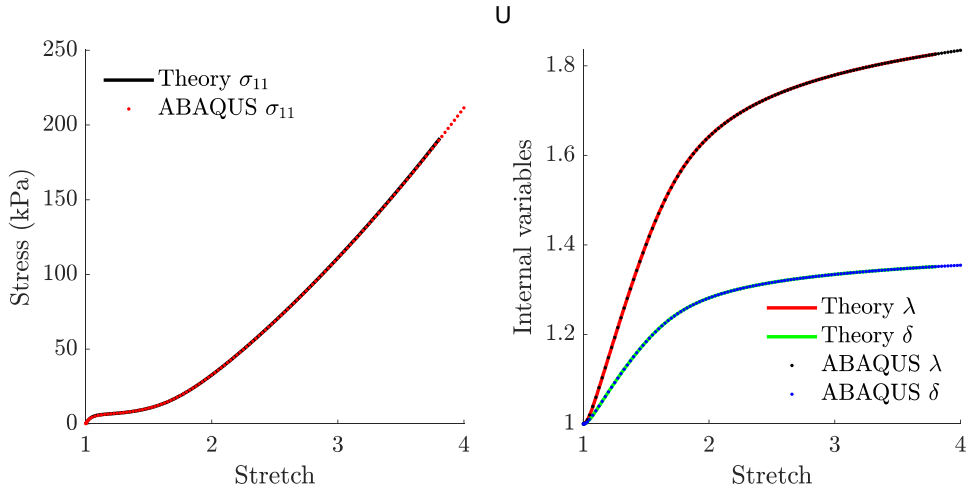


Figure 4.11: Uniaxial single-element simulation results plotted against theoretical results.

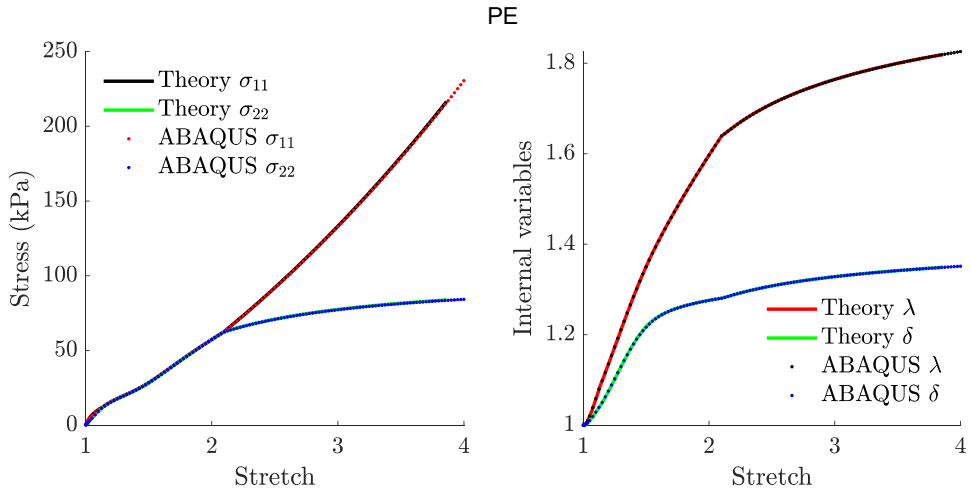


Figure 4.12: Planar extension single-element simulation results plotted against theoretical results.

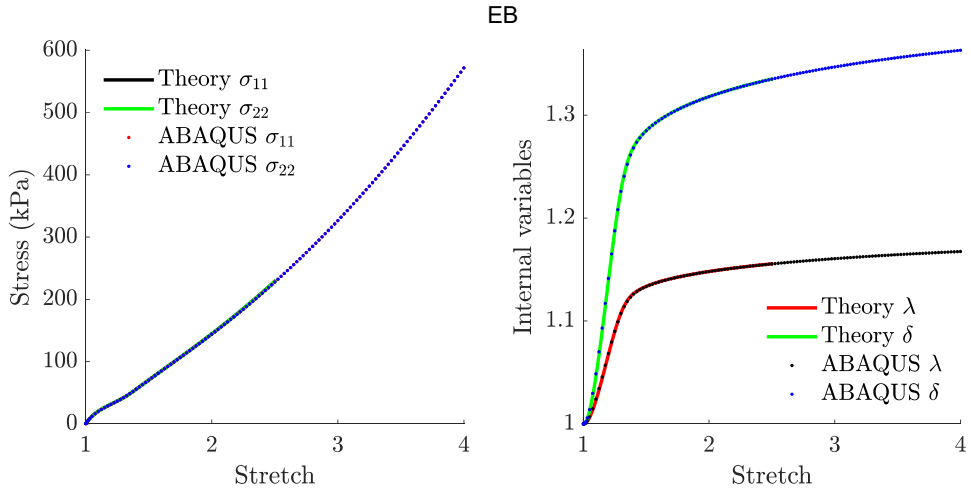


Figure 4.13: Equibiaxial single-element simulation results plotted against theoretical results.

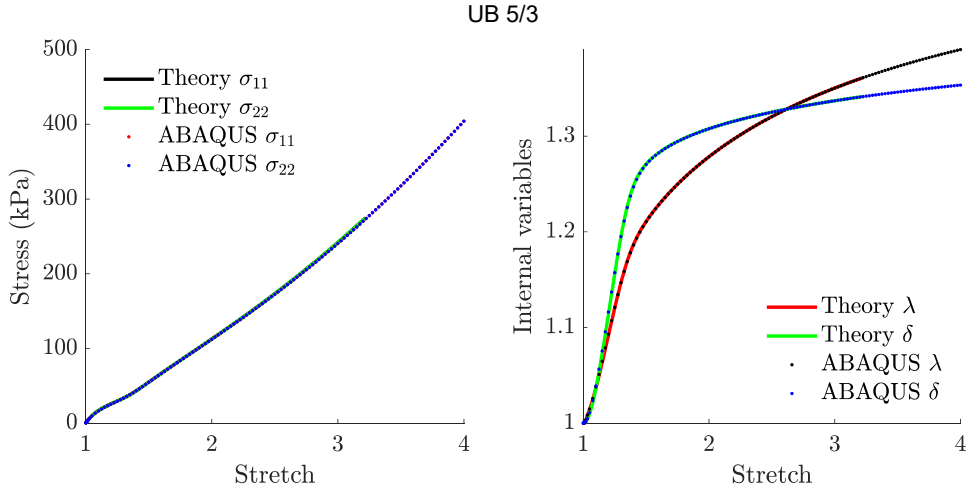


Figure 4.14: Unequal biaxial (with stretch ratio 5/3) single-element simulation results plotted against theoretical results.

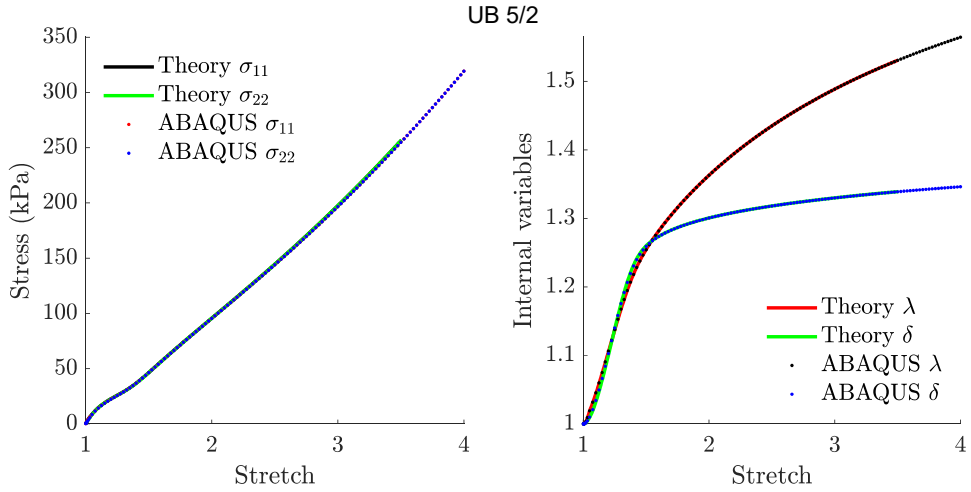


Figure 4.15: Unequal biaxial (with stretch ratio 5/2) single-element simulation results plotted against theoretical results.

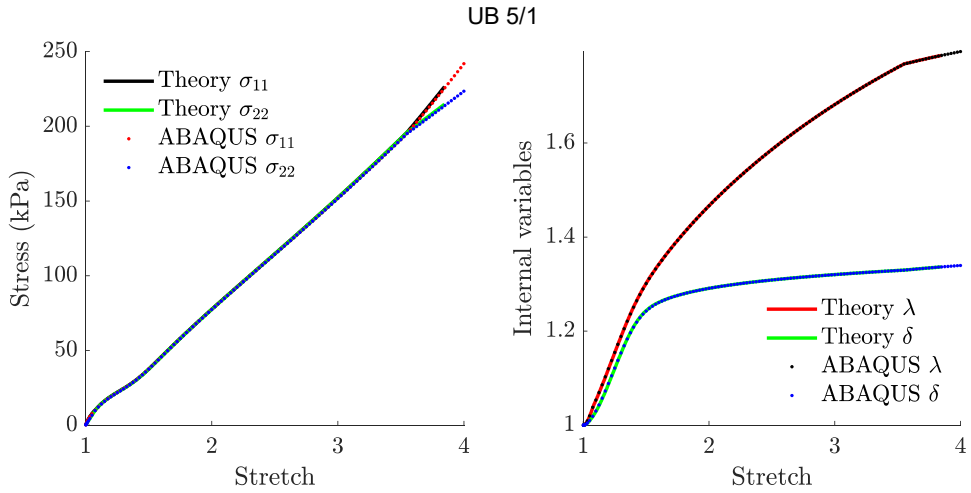


Figure 4.16: Unequal biaxial (with stretch ratio 5/1) single-element simulation results plotted against theoretical results.

Dependence upon strain rate and dissipation coefficients in uniaxial stretch

This section summarizes the results from single-element uniaxial stretch for three different strain rates—fast (1×10^{-2}/s), medium (1×10^{-3}/s), and slow (1×10^{-4}/s)—and three different values of α_δ—big (2.1838×10^8 Pa), medium (2.1838×10^7 Pa), and small (2.1838×10^6 Pa).

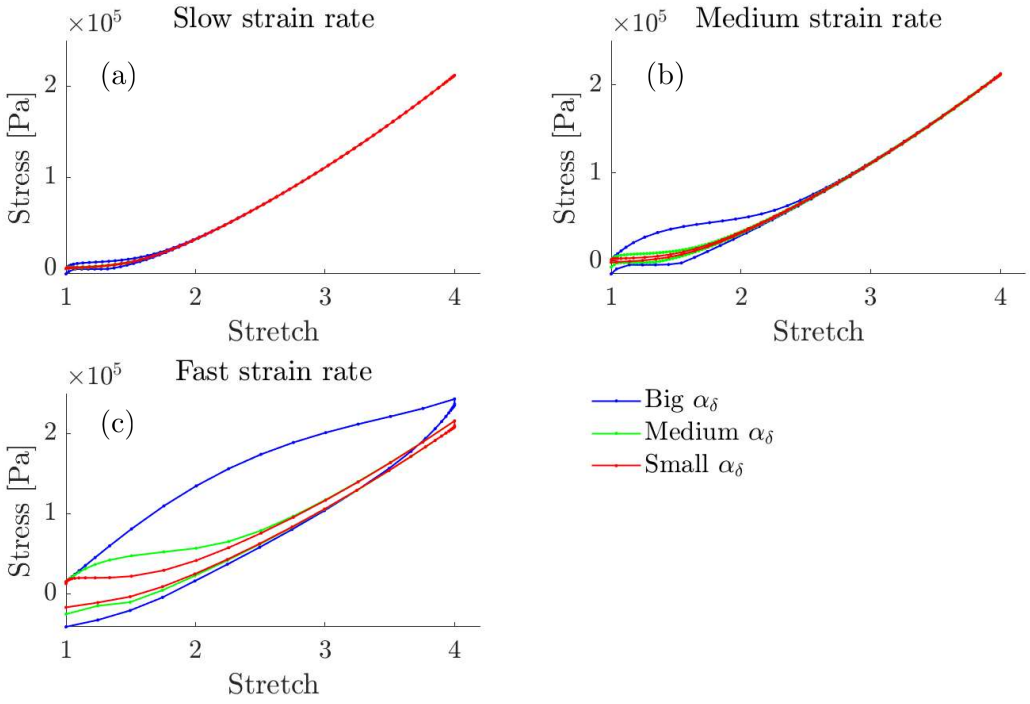


Figure 4.17: Uniaxial load and unload curves for varying α_δ and fixed strain rates: (a) slow (1×10^{-4}/s), (b) medium (1×10^{-3}/s), and (c) fast (1×10^{-2}/s).

As seen in Figure 4.18, the material response has correspondingly less hysteresis for slower strain rates, as is expected. We can also see that the smaller values of α_δ correspond to less hysteresis as well.

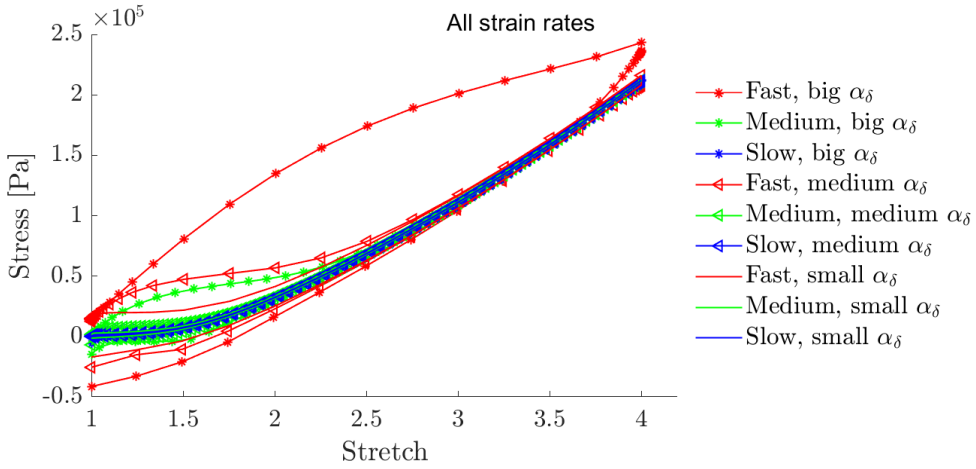


Figure 4.18: Uniaxial load and unload curves for all α_δ values and all strain rates.

Torsion of a cylinder

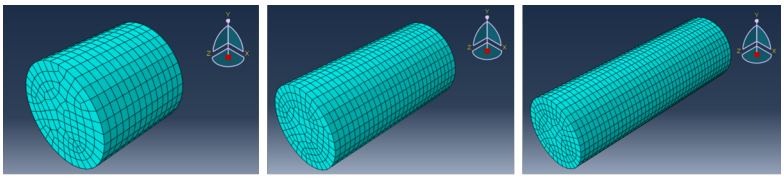
We move beyond the deformation of thin sheets or ribbons and study the bulk response of a solid cylinder of nematic elastomer under torsion, which have not been modeled before using FEA. The cylinder has height H, diameter D, and varying height-to-diameter ratios $H : D$, as shown in Table 4.3. We use a C3D8H element in `ABAQUS` with hybrid formulation, and the mesh is depicted in Figure 4.19.

Table 4.3: Dimensions of the cylinder under torsion

$H : D$ ratio	Height, H [m]	Diameter, D [m]	Number of elements
1:1	0.01	0.01	1652
2:1	0.02	0.01	3509
3:1	0.03	0.01	4859

The boundary conditions are such that the right face of the cylinder is entirely fixed in displacement and rotation in all three directions ($U_1 = U_2 = U_3 = 0$ and $VR_x = VR_y = VR_z = 0$ in `ABAQUS`) and the left face is zero entirely except for a prescribed angular velocity $\omega = 0.0026$ rad/s ($VR_z = 0.0026$ in `ABAQUS`). There is a tie constraint between the cylinder's face and a rigid plate at each end of the cylinder.

The angle DH describes the angle that one end of the cylinder twists with respect to the other end, as seen in Figure 4.20. We can define the strain rate at the outer rim as $\dot{D}R_o$, where R_o is the half the diameter. For example, for $\omega = \dot{D}H = 0.0026$ rad/s and $H : D = 1 : 1$, the strain rate at the outer rim is 0.0013 rad/s.



$H : D = 1 : 1$ $H : D = 2 : 1$ $H : D = 3 : 1$

Figure 4.19: Meshed cylindrical bodies for varying $H : D$ ratio.

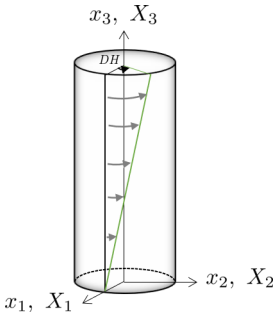


Figure 4.20: Schematic of the twist DH in the cylinder torsion deformation.

The moment and normal force obtained from the `ABAQUS` results are plotted in Figs. 4.21 and 4.22, separated by anisotropy parameter and $H : D$ ratio. The fact that the normal force F is nonzero and compressive is due to the Poynting effect [59]. Without F, the cylinder would elongate in the axial direction.

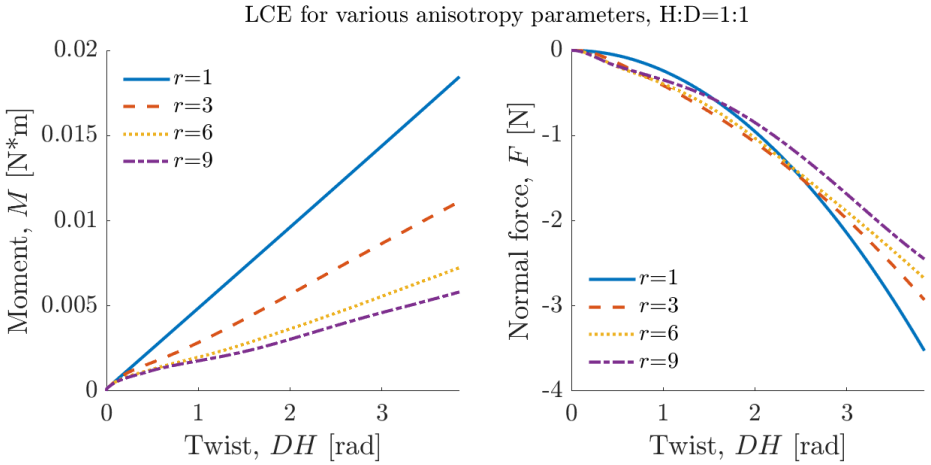


Figure 4.21: Moment M and normal force F of the cylinder under torsion for varying anisotropy parameters r. The $H : D$ ratio is fixed at $1 : 1$.

Based on the moment M and normal force F, we can also define two helpful quantities: torsional stress at the outer rim $\tau = \frac{2M}{\pi R_o^3}$ and normal stress $N = \frac{2F}{\pi R_o^2}$. Rivlin and Saunders [62] found that for rubbers, M was proportional to D and $|N|$

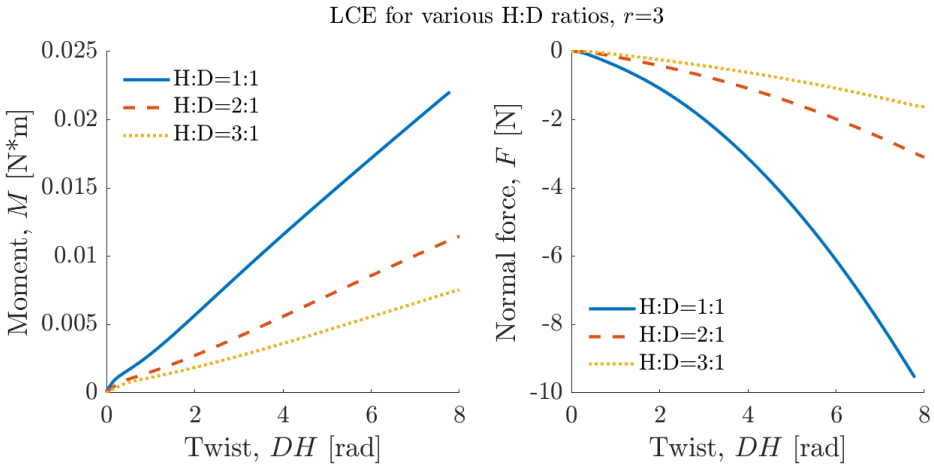


Figure 4.22: Moment M and normal force F of the cylinder under torsion for varying height-to-diameter ($H : D$) ratios. The anisotropy parameter is fixed at $r = 3$.

was proportional to D^2. So in Figure 4.23, we plot $\left|\frac{N}{\tau}\right|$ and indeed see that it is linear with DR_o for rubber (anisotropy parameter $r = 1$). For the nematic elastomers ($r > 1$), the behavior of $\left|\frac{N}{\tau}\right|$ is nonlinear. There are two distinct regimes of linear behavior with different slopes. The first, higher-slope regime corresponds to the reorientation of the liquid crystals, and the second, lower-slope regime corresponds to the response of the polymer chains.

For both the nematic elastomers ($r > 1$) and ordinary rubber ($r = 1$), the cylinders develop a kinking instability at certain critical values of twist, which corresponds to the abrupt change from positive to negative slope in the $|N/\tau|$ plots. The onset of kinking is a function of the $H : D$ ratios: the larger $H : D$ is, the easier it is for the structure to form a kink. Figs. 4.24a and 4.24b show views of the kink for anisotropy parameter $r = 3$ and height-to-diameter ratio $H : D = 3 : 1$. Note that the cross-section develops a pinch and no longer remains circular.

We also plot the evolution of the internal variables, λ and δ, throughout the deformation. Figures 4.26 and 4.27 show λ and δ for varying nodes of radii r_1 through r_5 in the cross-section halfway through the height of the cylinder (see Figure 4.25).

Due to the Poynting effect, twisting the incompressible cylinder will produce a lengthening in the axial direction, but since the cylinder is constrained in the length, the cylinder instead continuously develops an instability. Similar instabilities were observed expeirmentally in rubber cylinders that were subjected to twist, with and without axial stretch [29, 70]. Coyne [20] and Thompson and Champneys [70]

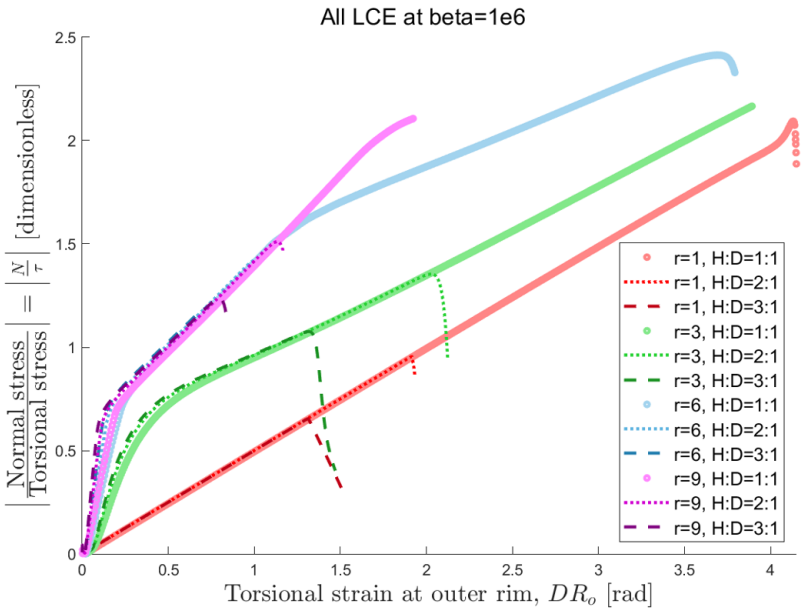
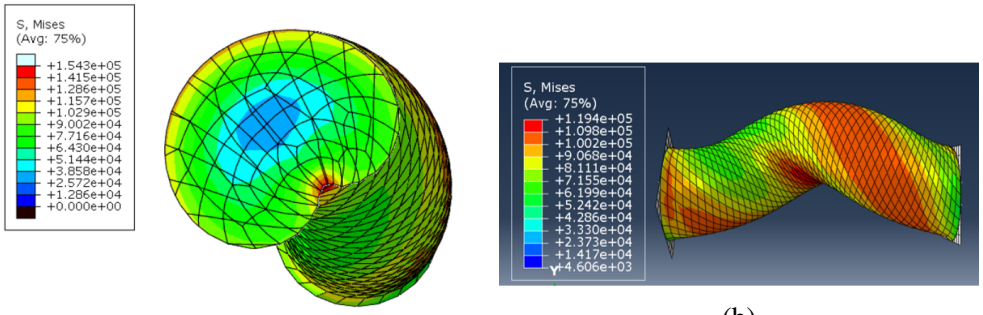


Figure 4.23: Plot of the absolute value of the ratio between torsional stress at the outer rim τ and normal stress N for varying anisotropy parameters r and height-to-diameter ratios $H : D$.



(a) (b)

Figure 4.24: Kinking instability for $r = 3$, $H : D = 3 : 1$: (a) Cross-sectional view, halfway through the height of the cylinder. (b) Side view.

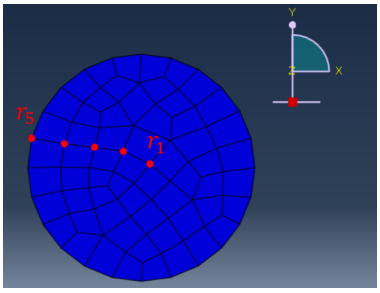


Figure 4.25: Nodes with increasing radii r_1 through r_5 in the cross-section halfway through the height of the cylinder.

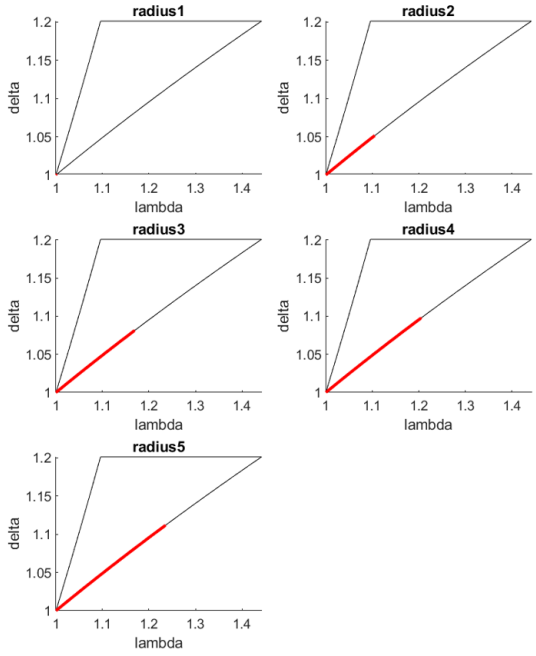


Figure 4.26: Evolution of the internal variables, depicted in the triangular region $\mathcal{T}$ for nodes of varying radii.

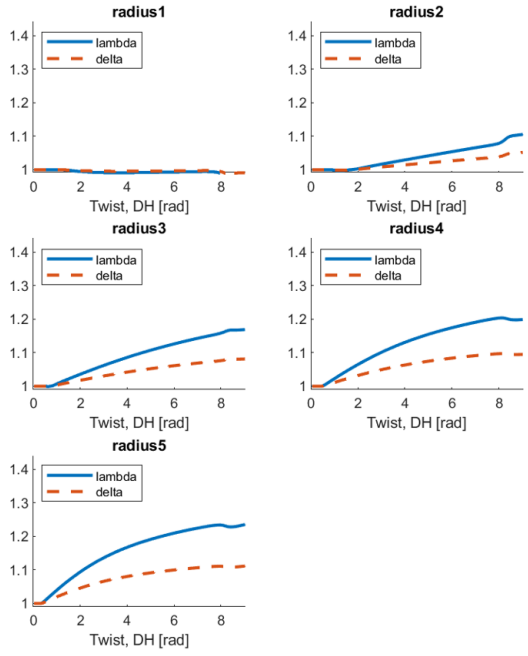


Figure 4.27: Evolution of the internal variables as a function of twist for nodes of varying radii.

describe the development of a quasi-static localization or helix similar to what we observe in Figure 4.24b. The kink has directionality based on the direction the cylinder is twisted; the helical shape develops as a continuation of the twist direction.

It is not surprising that the nematic elastomer cylinders kink, as the ordinary rubbers do so as well. If the height-to-diameter ratio were longer, we would expect that the cylinders would form loops and upon further loading, the loops would twist, similar to the experiments of [70]. However, the behavior of the normal stress and torsional stress from Figure 4.23 is a new finding.

To understand the instability further, we conducted an eigenvalue analysis in ABAQUS of the cylinder under torsion at various height-to-diameter ratios and anisotropy parameters. The cylinder is first preloaded under the torsion deformation described above, then a buckling step is conducted to gain the eigenvalues and eigenmodes. The viscosity is fixed at $\beta = 0$ throughout the simulation. The first five modes for $r = 3$ and $H : D = 2 : 1$ are shown in Figure 4.28. We observe shear banding as the deformation mode, which is very similar across the five modes. The corresponding eigenvalues, from Mode 1 to 5, are $(4.5688, 4.5689, 4.5711, 4.5715, 4.5746)$. The eigenvalues are very close together, which speaks to an imperfection-sensitive structure [24].

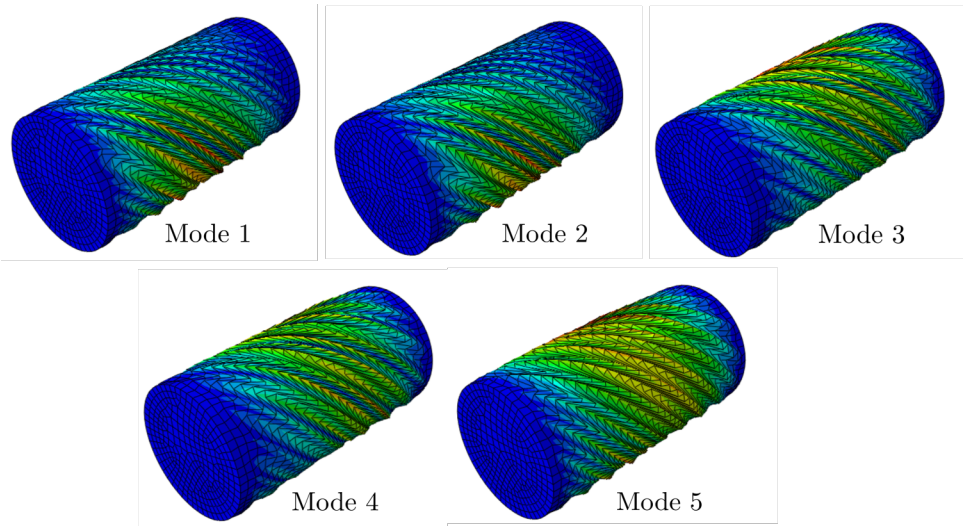


Figure 4.28: First five eigenmodes for $r = 3$ and $H : D = 2 : 1$. The colors represent the magnitude of displacement, from zero displacement (blue) to high displacement (red).

The eigenvalues from the first mode are plotted as a function of anisotropy parameter in Figure 4.29, where the eigenvalues for $r > 1$ start to move away from the isotropic ($r = 1$) eigenvalue.

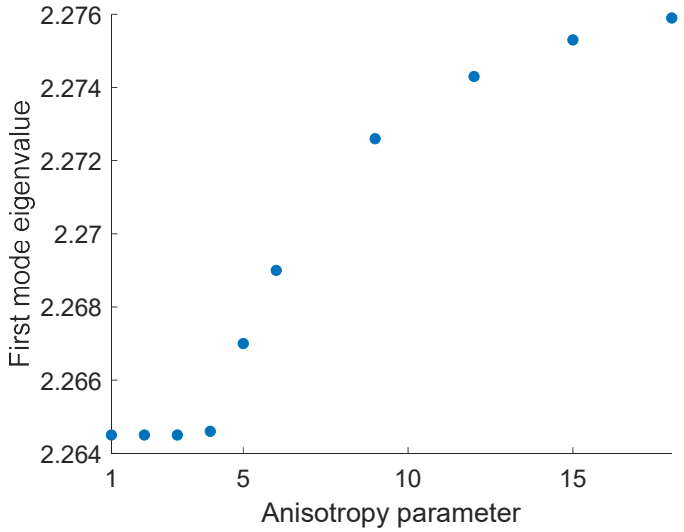


Figure 4.29: Eigenvalues from the first eigenmode plotted against anisotropy parameter for fixed $H : D = 1 : 1$.

4.6 Conclusion

In this chapter, we have formulated a constitutive relation to describe non-ideal isotropic-genesis polydomain nematic elastomers, which builds upon the work of DeSimone and Dolzmann [23]. We introduced internal variables that evolve according to a dissipative kinetic process that represent the material behavior throughout the classic regions of interest: a liquid-like region, a region in which fine-scale microstructure develops, and a solid-like region. We verify the model in `MATLAB`, performing the constrained optimization within the triangular region $\mathcal{T}$ using the `fmincon` function, then validate it with comparison against experimental results for various biaxial tension tests performed by Kenji Urayama's group. Finally, we adapt the constitutive relation for input as a user-defined material code in `ABAQUS` and study the deformation of torsion in a solid cylindrical body. We now have a tool to study these non-ideal polydomain nematic elastomers under arbitrarily complex loading configurations and boundary conditions in the future.

Chapter 5

EXPERIMENTAL CHARACTERIZATION OF NEMATIC ELASTOMERS

5.1 Introduction

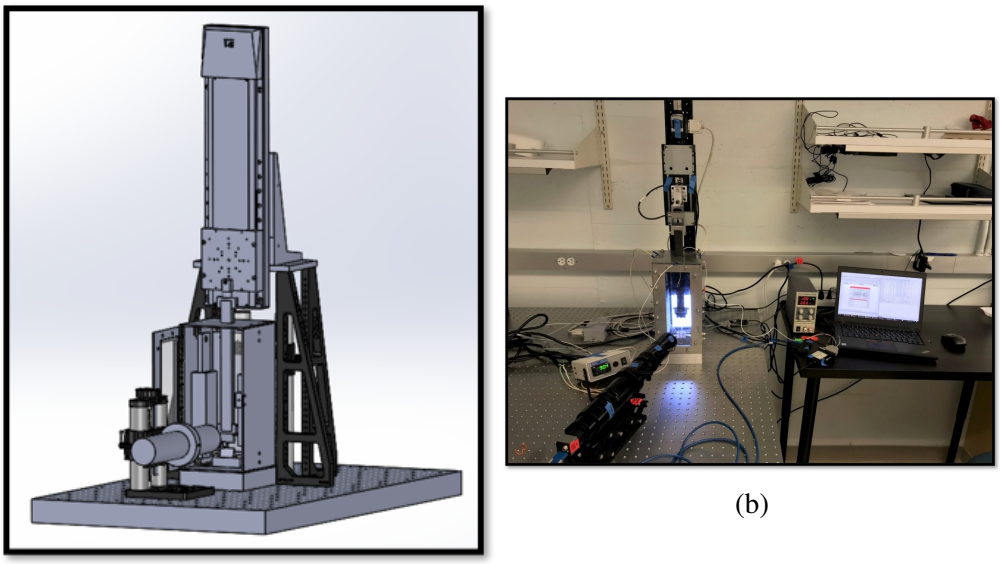
The previous chapters of this thesis have addressed the theoretical and computational characterization of nematic elastomers, discussing various deformations of both monodomains and polydomains. The goal of this chapter is to characterize the rate dependence and temperature dependence of these nematic elastomers experimentally.

The synthesis techniques of temperature-responsive nematic elastomers began with thin films. There has been much work in developing ways to align the nematic director field within the plane of the thin film of nematic elastomer, e.g. through applied magnetic field, mechanical strain, and textured surface patterning [12, 13, 45, 74–76, 81–83]. The advent of "click chemistry" allowed bulk nematic elastomers to be made partially crosslinked in molds, after which another crosslinking process could induce nematic alignment, without the limitations of thin films [55, 64, 80]. Further, the capability to 3D print of nematic elastomers with shear-alignment of the liquid crystal mesogens have been developed [4, 40].

In this thesis, we make samples of nematic elastomers in-house using the click-chemistry method of synthesizing nematic elastomers following [64], discussed in Section 5.2. This technique involves an easy-to-follow, one-pot chemistry recipe in which the polymer mixture is poured into open-faced molds, and after curing, polydomain nematic elastomers are formed. If a UV crosslinker is added to the mixture, then there exists the option to undergo a second crosslinking step to create a monodomain, using applied mechanical strain as the alignment technique.

In Chapter 5.3, we discuss the design and build of a thermo-mechanical tensile test setup capable of testing nematic elastomers at various temperatures while capturing strain, stress, temperature, and imaging data. Figure 5.1a shows the `SolidWorks` design of the experimental setup, and Figure 5.1b shows a picture of the actual setup in the lab. We have implemented an imaging setup using polarized light microscopy. Due to the underlying optical properties of the liquid crystals, useful information can be gained from the optical properties of nematic elastomers. Viewing a sample

of nematic elastomer between cross-polarizers can help determine the amount of ordering existing in the system due to the material's birefringence. Macroscopically, a monodomain sample in its nematic state appears transparent, and a polydomain sample appears cloudy and opaque. Using a microscope can help study a sample more closely and reveal information about anisotropy and domains.



(a) (b)

Figure 5.1: (a) `SolidWorks` model of a portion of the experimental setup. (b) Picture of the experimental setup.

Finally, Section 5.4 shows the stress-strain curves resulting from tests at varying temperatures, and the code used to run the experiments can be found in the Appendix D.2.

5.2 Sample preparation

Materials

For sample preparation, we synthesize main-chain polydomain nematic elastomers, following [64]. We include the photoinitiator HHMP to allow for a second crosslinking, used for director alignment into a polydomain. We use the following chemicals for the synthesis procedure:

Table 5.1: Table of chemicals

Chemical name and purpose	Full chemical formula	Manufacturer
RM257, di-acrylate mesogen	1,4-Bis-[4-(3-acryloyloxypropyloxy) benzoyloxy]-2-methylbenzene	Wilshire Technologies
EDDET, di-thiol spacer	2,2'-(ethylenedioxy) diethanethiol	Sigma Aldrich
PETMP, tetra-thiol crosslinker	Pentaerythritol tetrakis (3-mercaptopropionate)	Sigma Aldrich
DPA, catalyst	Dipropylamine	Sigma Aldrich
HHMP, photoinitiator	2-Hydroxy-4'-(2-hydroxyethoxy)-2-methylpropiophenone	Sigma Aldrich
Toluene, solvent	Toluene	Sigma Aldrich

Please see Section C.1 for further details. The following lab equipment was used in preparing the samples:

- Fume hood
- Vortex mixer (VWR 10153-834)
- Vacuum oven (VWR 89508-428) and vacuum pump (VWR 89209-762)
- HDPE molds (McMaster 8619K614, then machined by CNC)
- Hot plate (VWR 12620-978)
- Eppendorf micro-pipettes (Sigma Aldrich Z683884)
- Precision scale (VWR 75802-858)
- UV lamp (CureUV 191340), protective box, and UV-protecting glasses (Thor-Labs LG3)
- Compartmented sample boxes (McMaster 4629T15)

The molds that were used to make the samples were made from $6" \times 6" \times 1"$ HDPE blocks, and we CNCed them into custom shapes in the machine shop.

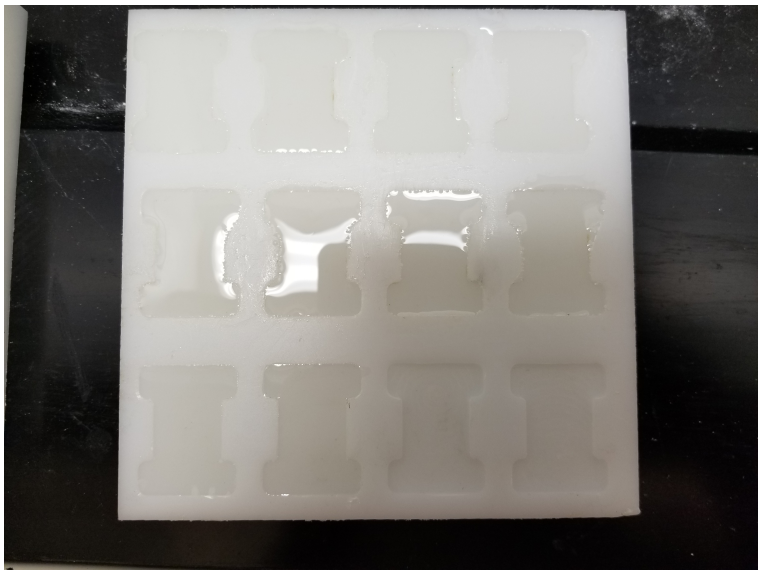


Figure 5.2: HDPE molds with CNCed pockets for samples to be poured into.

We made various molds for making polydomain dogbone samples, as well as larger polydomain samples that could be clamped and stretched into monodomains. The corners of pockets are filleted, rather than manufactured with a sharp corner. The sharp corners often lead to cracks in the finished sample, since removing the cured samples is a delicate process, and the polymer can easily get caught on snags and the sample may rip.

Synthesis procedure

The sample preparation has been modified slightly from the original source in [64]. Note that there is a template that helps to follow the procedure, found in Section C.2. Below are instructions for following the synthesis procedure.

Record your name, date, and the time you start the synthesis procedure. Make sure you have on the following personal protective equipment (PPE): lab coat, safety glasses, gloves, closed-toe shoes, and long hair tied back. Clean the mold that you will use with isopropyl alcohol, rinse it with deionized water, and wipe it down with a Kim wipe so that it is completely dry. Prepare two glass vials of 30mL capacity, one labeled as “LCE solution," and the other labeled as “DPA+toluene," which will hold the diluted catalyst solution.

Set the hot plate dial level to between 7 and 7.5. Place a beaker with water on top of the hot plate, and place the glass thermometer (held by the stand) inside the beaker of water. Allow the water to heat to $\approx 80^\circ$ C.

We will follow a triple batch recipe because making a larger batch attains better accuracy than the single batch recipe from [64]. [1] This is also the recipe for 50 mol% PETMP, but the crosslink density can be changed by modifying the amount of PETMP and EDDET accordingly. Where possible, fill out the template with the actual weight before and after transferring the chemical to the vial, and calculate the percent error between the actual weight and the expected weight.

Start by taring a weigh boat, measuring 12g of RM257 in the weigh boat, and transfer about half of the RM257 to the “LCE solution" vial. Set the rest of the RM257 aside. Add in 3000μL of toluene to the vial via micro-pipette. Place the vial on the hot plate, starting a timer. Occasionally swirl around the solution in the vial by hand.

While waiting for the solution to completely dissolve, prepare the HHMP. Fold a square piece of weigh paper in half across the diagonal, and weigh out 0.0257g of HHMP. Add the HHMP to the vial, and continue heating it on the hot plate. (Since you want to keep the scale tared to the RM257 weigh boat measurement, you’ll need to do the HHMP math by hand. Record the weight of the weigh paper, then the weight of the weigh paper plus HHMP, and finally the weight of the leftover HHMP on the weigh paper.) When the RM257+toluene mixture has dissolved enough that there is enough room in the vial to add more RM257 and toluene, finish adding the rest of the RM257 and 2549.1μL of toluene via micro-pipette. Record the weight of any leftover RM257 on the weigh boat. Place the vial back on the hot plate.

While waiting for the solution to fully dissolve again, prepare the catalyst solution in the separate vial. Make 1.5 times what the solution needs, measuring out 0.0549g of DPA (74.25μL via micro-pipette) and 2.5011g of toluene (2891.25μL via micro-pipette). Mix the catalyst solution on the Vortex mixer thoroughly.

When the LCE solution has fully dissolved and there are no solids left, record the time that it took to heat the solution. Without waiting, move forward with adding 2.1657g of PETMP (1691.64μL via micro-pipette). The PETMP is a very viscous liquid so if you choose to use the micro-pipette, you will need to wait for the solution to be fully drawn into the micro-pipette tip. Add 1.6158g of EDDET (1442.67μL via micro-pipette). Vortex mix the catalyst solution one last time and then add 1.7043g of the solution (1974μL via micro-pipette) to the LCE solution. Vortex mix the LCE solution for $\approx$ 20 seconds. Slightly loosen the cap on the vial and place the

[1] I thank Ruobing Bai for this suggestion.

vial in the vacuum oven and pull a vacuum of 20in Hg at room temperature. Keep a constant vacuum of 20in Hg for 45 seconds, then release the vacuum.

Look at the solution and see if there are any solids that have precipitated in the solution. Note the solution appearance: cloudy, clear, undissolved solids? Then if there are undissolved solids, put it back on the hot plate and record the time it takes to get a fully dissolved solution again. Take a picture of the solution and then immediately pour the solution into the mold. An example picture is shown in Figure 5.3a. Make sure the solutions are filled completely without any air bubbles and that each pocket is filled to approximately the same depth. Take a picture of the mold.

Allow the mold to rest in the fume hood at room temperature for about 12 hours before moving onto the next step, which is placing the samples into the vacuum oven.

Cleanup: Record the time that the sample preparation finished, dispose of any hazardous waste appropriately, clean utensils and dishes, turn off the scale, and turn off the hot plate.

Placing samples into vacuum oven. Roughly 12 hours later, put the samples into the vacuum oven. Record the date and time. Wearing gloves, place the mold into the center of the middle shelf of the vacuum oven. Pull the vacuum level to 20.5 in Hg by setting the handle to “evacuate," turning on the vacuum pump, and when the vacuum level reads 20.5 then simultaneously turn off the vacuum pump with one hand and setting the handle to “closed" with the other hand. (We overshoot the vacuum level because the vacuum level will settle to $\approx$ 20in Hg after a few hours.) Turn on the temperature switch, with the temperature dial set to 1.5. Take note of anything irregular. Record the temperature at a couple times throughout the 24 hour period.

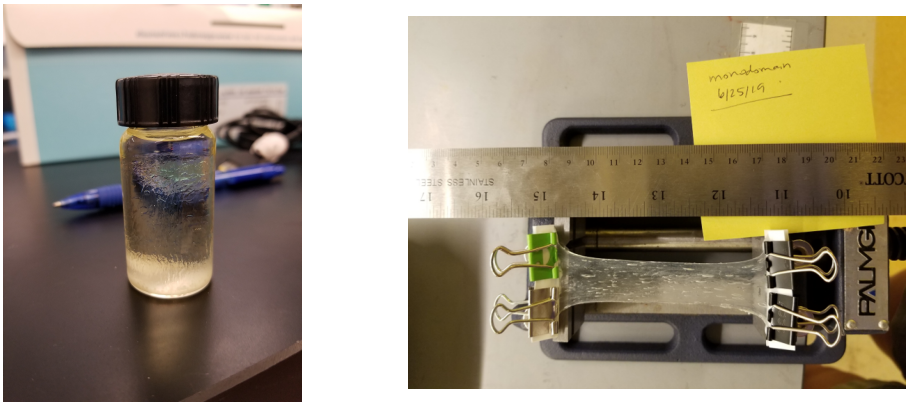
Taking samples out of vacuum oven. Roughly 24 hours later, take the samples out of the vacuum oven. Record the date. Turn off the temperature switch on the vacuum oven, and release the vacuum by setting the handle to read “vent." Take the samples out of the vacuum oven using a heat-protecting glove, and place them in the fume hood. Write down the time that the samples are taken out of the oven. Note anything irregular.

For sample storage, be sure to keep the samples stored inside a sealed box in a cool, dry cabinet, protected from ambient light and air. Whenever samples are not being prepared or used, store them in this way. You can also vacuum seal them.

About 30 minutes later, write down the time. Wear gloves and use a metal spatula to gently take each sample out of the mold. Heat each sample until it is in its isotropic state, then place it in a labeled sample box. Note any cracks, bubbles, or other irregularities in the samples.

UV crosslinking. Anytime after the samples have come out of the molds, you can UV crosslink them at room temperature (as long as the samples were made with HHMP, the photoinitiator). Record the date and time that you are UV crosslinking the samples, as well as the UV crosslinking equipment. You must wear UV-protecting glasses at all times during this process. Wearing gloves, clamp the sample in the desired configuration and place it inside the UV crosslinking box. Turn on the UV lamp and use a stopwatch to be able to record the time spent under the UV light. If necessary, flip sample over and repeat process until the sample is fully crosslinked in the desired configuration. Take pictures, and take note of the temperature of the box (the UV lamp may cause the box to heat up) and any other irregularities.

For instance, Figure 5.3b shows a polydomain sample that was clamped until it became optically clear, which forms a monodomain when crosslinked in that configuration.



(a) (b)

Figure 5.3: (a) LCE solution immediately before being poured into mold. (b) Polydomain sample clamped in a stretched monodomain state, ready for UV crosslinking.

Best practices

Below are a list of items to consider when making samples:

- When solution is inside the vial, keep the lid on the vial as much as possible because liquid can evaporate and disturb any weight measurements.
- When weighing on the scale, close all doors to the scale to limit air flow that will disturb the measurements, and wait for the numbers to settle before recording the measurement.
- Never exceed the maximum scale weight, or 60g.
- Always return micro-pipettes to their maximum stated capacity for storage. (e.g. If it is a 1000μL micro-pipette, turn the reading to 1000μL.)
- Always use a new tip when using micro-pipettes. Keep the tip of the micro-pipette submerged in the liquid you are retrieving at all times. Wipe off any excess liquid from the micro-pipette tips before transferring the solution to the vial. Ensure that there are no bubbles inside the micro-pipette tip.
- The synthesis should take place roughly 12 hours before the samples go into the vacuum oven (if necessary, more than 12 hours can pass, but not less). Then another 24 hours should pass before the samples are taken out of the vacuum oven (if necessary, slightly more than 24 hours can pass, but not less). Then you can carefully take the samples out of the molds about 30 minutes after they have been taken out of the vacuum oven. You can UV crosslink anytime after they have been taken out of the molds, but testing of these samples should ideally take place within 2 days of being taken out of the molds.

5.3 Experimental setup

Design, manufacturing, and assembly

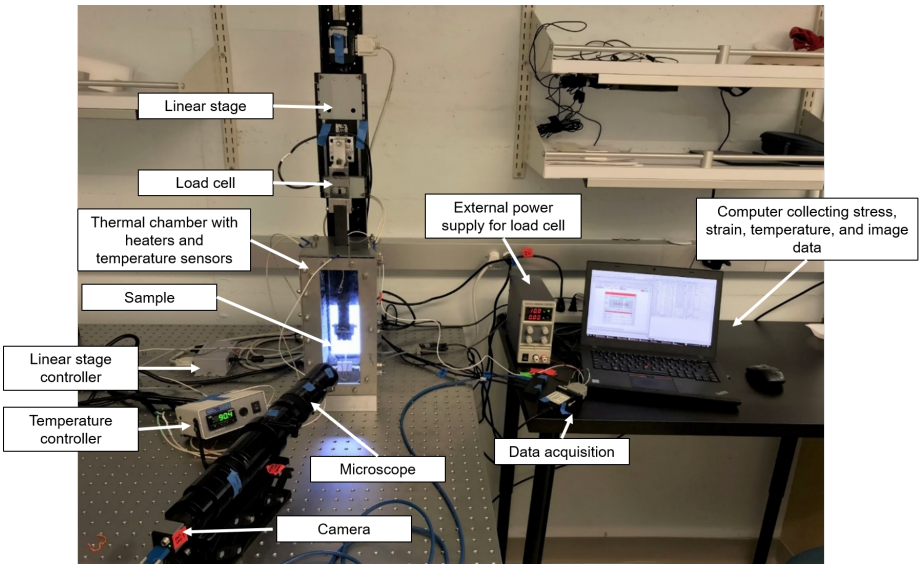


Figure 5.4: Picture of experimental setup.

Figure 5.4 shows the entire experimental system with labels, and the schematics in Figure 5.5 and 5.6 depict the connections in the test setup. The system is made up of the following subsystems: chamber assembly (chamber with windows, stationary bottom clamp, and moving pullrod with clamp), heating (heaters, RTD sensors, and temperature controller), extension (linear stage, and linear stage controller, and suspension assembly), load (load cell and external power supply), optics (optical table, lighting, microscope, camera, cross-polarizers), and data acquisition (computer and DAQ).

The setup was designed in `SolidWorks` (see Figure 5.1a), with metal adapters to connect the chamber to the optical table, the pullrod to the load cell, and the load cell to the linear stage. Glass windows were designed in the front and the back of the chamber to accommodate the optical setup. The pullrod and one of the adapters was originally manufactured in metal, but conversations with Sam Daly led to them being remade in machinable ceramic to prevent the load cell from overheating. We used the Jim Hall Design and Prototyping Lab (the machine shop in the Mechanical and Civil Engineering department of Caltech) to manufacture all of the metal and machinable ceramic pieces.

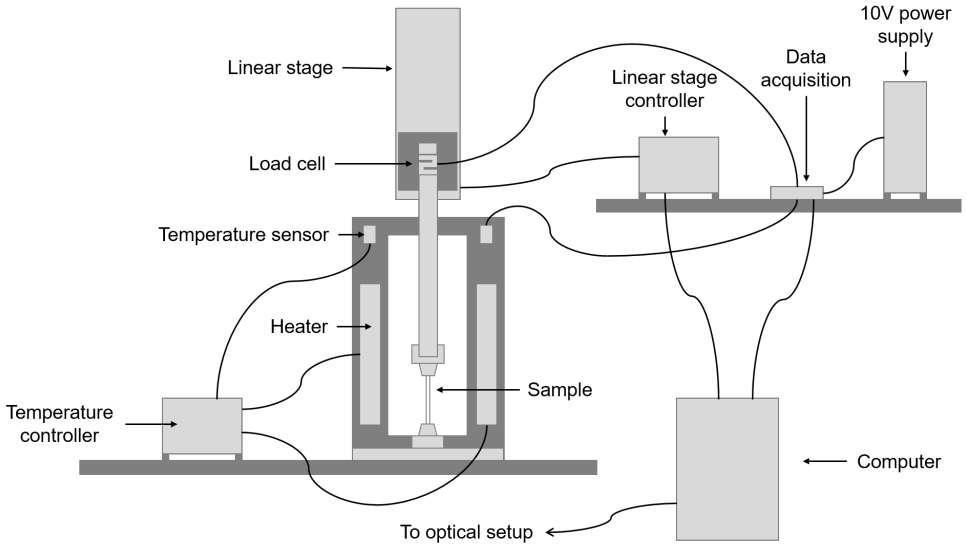


Figure 5.5: Schematic of experimental setup.

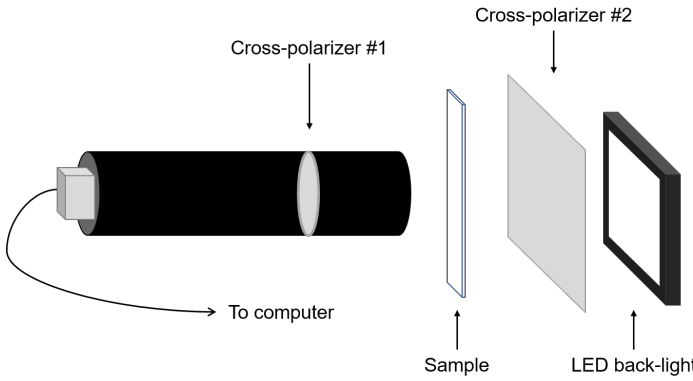


Figure 5.6: Schematic of the optical portion of the experimental setup.

The data acquisition system involved soldering the wires of one of the RTDs and the load cell to the DAQ, and coding in `MATLAB` to obtain the temperature and load data. It also required further coding in `MATLAB` to acquire the extension data from the linear stage. Julia Combs, a student who worked in the lab for a summer via the Summer Undergraduate Research Fellowships (SURF) program, figured out how to obtain imaging data from the camera in `MATLAB`.

Clamping the sample can be difficult, depending on the stiffness and texture of the sample. We chose to use binder clips to clamp the samples because they are self-tightening. Binder clips of different sizes and materials were used depending on the sample. Clamping at high temperature is especially challenging because the

samples became susceptible to breakage at the clamping point. To avoid the sample slipping in the grips, we handle the test samples only with gloves, so that the sample does not retain any oils from skin. Additionally, sanding the insides of the binder clip can help to create friction between the sample and the clamp, or adding a piece of rubber (e.g. from a rubber glove) can also create more friction so that the samples do not slip out.

Materials

The various components of the setup, listed by subsystem, are:

- Custom-made chamber assembly
  - Chamber components (McMaster 1658A12, 8983K128, 8983K118, 2313N23)
  - Various hardware (nuts, bolts, washers, etc.)
  - Various adapters (McMaster 8489K44, 4416T47)
  - Windows and sealing (McMaster 8476K999, 92320A662)
  - Stationary bottom clamp and moving clamp on pullrod (McMaster 12755T82)
  - Moving pullrod (McMaster 8489K46)
- Heating
  - Two heaters (Omega OTF-102/120V)
  - Two RTD air temperature sensors (Omega RTD-805-B)
  - Temperature controller (Omega CSI32RTD-C24)
- Extension
  - Linear stage (Physik Instrumente M-531.EC)
  - Linear stage controller (Physik Instrumente C-863.11)
  - Linear stage z-axis mounting bracket (Physik Instrumente M-592.10)
  - Suspension system (ThorLabs VB01B)
- Load
  - Load cell (Omega LC101-50)
  - External 10V power supply
- Optics
  - Vibration-isolation table (ThorLabs T46J)
  - LED backlight (Edmund Optics 83-873)
  - Long working-distance microscope (Infinity K2/SC)
  - Camera (Edmund Optics 86-770)
  - Cross-polarizers (Infinity 991167, Edmund Optics 45-669)
  - Calibration slide (ThorLabs R1L3S5P)
- Data acquisition

Lab computer (Lenovo ThinkStation P330)
DAQ (Omega INET-600)
`MATLAB` software

Tensile test procedure

Below are instructions for the tensile test procedure:

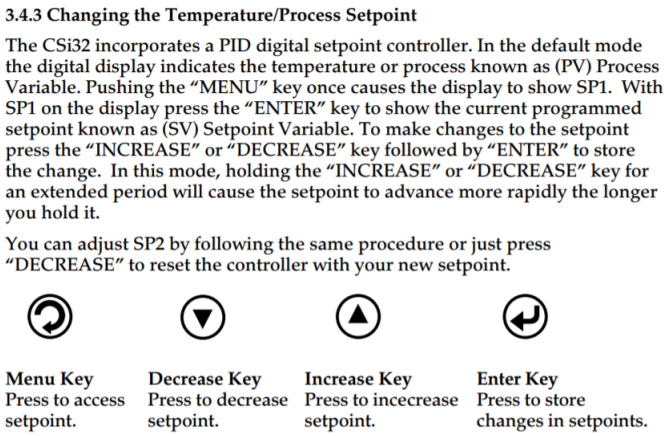
Start by donning the proper PPE: put on safety glasses, gloves when handling samples, and heat-protection gloves when you will be in contact with the chamber when it is hot. On the lab computer, make a new file folder that will hold the test results (naming it with the date of the test, sample information, etc.) and also start a new file in OneNote using the test results template found in Section C.3. Copy in the `MATLAB` code to the folder (see the Appendix in Section D.2).

Visually check that the linear stage has no obstructions in its path, and make sure that anywhere the load cell moves will not crush it or cause damage to it. A suggestion is to put up a sign on the door to the lab saying that a test is in progress, and make sure that no wires or other objects will touch the optical table throughout the test.

Refer to the schematic in Figure 5.5 for all relevant connections. Turn on the external 10V power supply, linear stage controller, and temperature controller. Plug in the 3 USB connectors to the computer: the linear stage controller, the iNet600 data acquisition (DAQ), and the camera. For optical data, turn on the LED backlight and position the cross-polarizers. There is an in-line cross-polarizer inside the microscope, and another cross-polarizer should be between the sample and the LED backlight, as seen in Figure 5.6. Then open `MATLAB`, set the working directory to the current test's folder, and open the main .m file (see Section D.2).

Wearing gloves, heat the sample you wish to test and let it cool completely. Then cut the sample into a strip using a new, sharp razor blade. With calipers, measure the width and thickness of the sample. Record these measurements in the OneNote template and in the `MATLAB` code, taking care to keep the units consistent.

If collecting imaging data, open the `uEye Cockpit` software. Make sure that the camera is facing in the correct orientation (see the red sticker on the camera). Take and save a picture of the ThorLabs calibration slide so that you can create a size scale. Ensure that polarizing filters are in place. You can set up a sequence of images to be taken, or you can capture a video. Be sure to take a picture that allows you to correlate the image to the timing of the test.



3.4.3 Changing the Temperature/Process Setpoint

The CSi32 incorporates a PID digital setpoint controller. In the default mode the digital display indicates the temperature or process known as (PV) Process Variable. Pushing the "MENU" key once causes the display to show SP1. With SP1 on the display press the "ENTER" key to show the current programmed setpoint known as (SV) Setpoint Variable. To make changes to the setpoint press the "INCREASE" or "DECREASE" key followed by "ENTER" to store the change. In this mode, holding the "INCREASE" or "DECREASE" key for an extended period will cause the setpoint to advance more rapidly the longer you hold it.

You can adjust SP2 by following the same procedure or just press "DECREASE" to reset the controller with your new setpoint.

Menu Key Press to access setpoint.

Decrease Key Press to decrease setpoint.

Increase Key Press to incecrease setpoint.

Enter Key Press to store changes in setpoints.

Figure 5.7: Instructions for setting the setpoint temperature in the temperature controller.

If the test is performed at a temperature higher than room temperature, then use the Omega temperature controller to set the setpoint temperature according to the instructions manual (see Figure 5.7). Note that the temperature is in Farenheit, not Celsius. Preheat the test setup to the desired temperature and then clamp the samples in to begin the test, using the preheat section of the code to record the preheat data. Put the insulation on the top of the chamber, to keep the load cell cool. Close the chamber door, and point the floor fan (on a low speed) at the load cell to keep it cool as well[2].

In the code, enter in the desired strain and strain rate, according to Table 5.2. Also set the position where the test will start using `startCoarsePos`. Ensure that this position will never cause the pullrod to crash into anything, which would damage the linear stage and the load cell.

Table 5.2: Table of testing strain rates

Fast strain rate	10^{-2}/sec
Medium strain rate	10^{-3}/sec
Slow strain rate	10^{-4}/sec

Double-check that the linear stage is free to move without obstruction. Run the block of code that connects `MATLAB` to the linear stage controller and references the linear stage. This code will cause the linear stage to move to its reference position at 153mm, which is in the exact middle of the 306mm-long stage.

[2]I thank Sam Daly for this tip about using a fan to cool the load cell.

Attach the pullrod to the load cell. With gloves on, carefully clamp the sample into the top clamp. You can finish clamping the sample using one of two methods. For the first method, if you know that the start position in `startCoarsePos` is a good location to hold the specimen in the clamps, then hold the bottom clamp open and execute the next block of code, which moves the linear stage to the start position. Skip the next block of code, which allows you to jog the linear stage up and down, without executing it. Or you may do the second method, which is to execute the block of code that moves the pullrod to `startCoarsePos`, and you can execute the block of code after that to jog the linear stage up and down to get the specimen clamped in position well. Type `u` or `d` to move the linear stage up or down in small increments until you are satisfied with the position, and you can clamp the sample in this way. Do not introduce any stresses to the specimen, and make sure the specimen is aligned well. When the specimen is fully clamped, close the chamber door.

The next block of code queries the linear stage for its position and then calculates the gauge length automatically. There is no action needed here.

The next block of code calculates the extension of the linear stage, the velocity, and the time that the test will take. This code assumes a full cycle through loading and unloading. The code also calculates the cross-sectional area. Modify the step size accordingly for various strain rates, ensuring that the linear stage will not move too fast or too slow for its specifications.

After that, the next block of code will configure the linear stage, calculating the starting position, the stopping position, and the array called `steps`. Also execute the initialization of the data acquisition system, using the correct .prf file that matches the correct load cell and temperature sensors signals. There is no action required from you here.

Finally, the `move` function is called in the next block of code, and the position, temperature, and load data are saved in their corresponding structures. Be sure not to disturb the test in any way by touching the optical table. After the test is finished, the last blocks of code can be executed to plot and save the data, which includes the stress-strain curve.

Save the files and data, complete the test template as you go, and remove the specimen from the clamps. Create a new folder for each test performed, and repeat this testing process as needed. After you complete testing, then there are cleanup procedures to follow. Save all data, exit `MATLAB`, and unplug the 3 USB connectors from the

computer (linear stage controller, camera, and data acquisition). Turn the power off to the linear stage, the temperature controller, and the external power supply. Cover up the camera lens and store it so that no light or dust enters the sensor. Take the pullrod off the load cell for storage so that the load cell is unloaded when not in use. Be sure to store the specimens properly (in containers, inside closed cabinets) so that they do not age as quickly in the ambient air or light.

Best practices

Below is a list of items to consider when using the tensile test setup:

- Don't touch the vibration isolation table while test is running!
- It might be helpful to put a sign on the lab door saying "testing in progress" so people don't come into the lab and disrupt the test.
- Wear safety glasses (in case of fracture).
- Make sure that when the linear stage moves, the wires and everything are not going to get caught, that the load cell won't be jammed or overloaded.
- Shut the door of the chamber during the test, because we don't want ambient air in the lab to affect it.
- Handle samples with gloves so that your hand's oils don't transfer to the sample, making it harder for the clamp to hold onto the sample.
- Use a new sharp razor blade to cut samples
- Store any samples inside the box in darkness when not being tested.
- Test samples starting on the day that the samples came out of the oven, and finish tests within 2 days.
- Use virgin samples for all the tests, unless stated otherwise. Make a note of the sample's loading and heating history.
- I am able to leave the experiment unattended when testing is done at room temperature, with the permission from the Safety Office, who has inspected the experimental setup. However, I do not leave the experiment unattended when testing is done at high temperature because the heaters present a fire risk.
- When testing a sample in the load/unload cycle, ensure that you choose a strain that will be safe enough for the sample not to fail prematurely during the test.

Test matrix

Here we have the completed test matrix, with each cell denoting the date that the samples are synthesized (not tested). The “polydomain" samples are not treated to a second crosslinking and are therefore in the polydomain state at nematic temperatures, while the “monodomain ⊥" samples are pulled in a uniform direction and treated to a second crosslinking so that they are in the monodomain state at nematic temperatures. The monodomain samples are pulled in tension perpendicular to the direction of the mesogen orientation.

Table 5.3: Tests at nematic temperature (room temperature ≈ 22°C)

	Fast strain rate	Medium strain rate	Slow strain rate
Polydomain pull til failure	7/29/20	8/12/20	11/4/20
Polydomain load/unload sample 1	7/29/20	7/29/20	7/29/20
Polydomain load/unload sample 2	7/29/20	7/29/20	7/29/20
Polydomain load/unload sample 3	7/29/20	7/29/20	7/29/20
Monodomain ⊥ pull til failure	8/12/20	8/12/20	8/12/20
Monodomain ⊥ load/unload sample 1	8/12/20	8/12/20	8/12/20
Monodomain ⊥ load/unload sample 2	8/12/20	8/12/20	8/12/20
Monodomain ⊥ load/unload sample 3	8/12/20	–	–

Table 5.4: Tests at isotropic temperature (≈ 130°C)

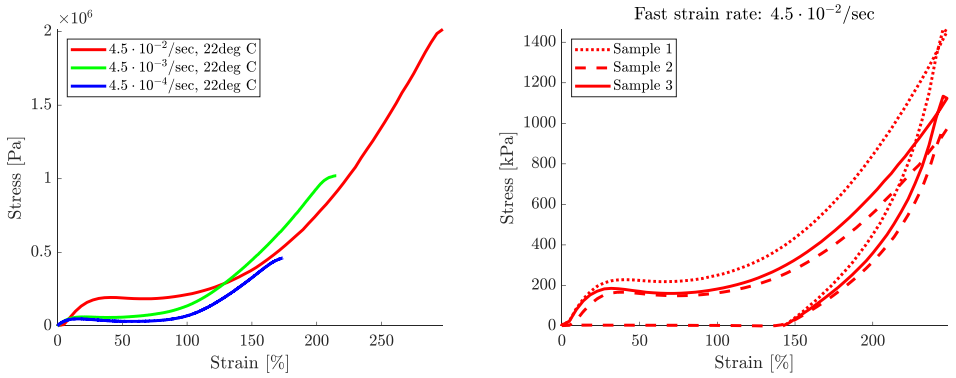
	Fast strain rate	Medium strain rate	Slow strain rate
Polydomain pull til failure	4/8/2021	4/8/2021	4/8/2021

5.4 Experimental results

In this section, all samples have 50 mol% PETMP. Each curve corresponds to distinct samples (no samples were reused). The stress is defined as force divided by the undeformed cross-sectional area.

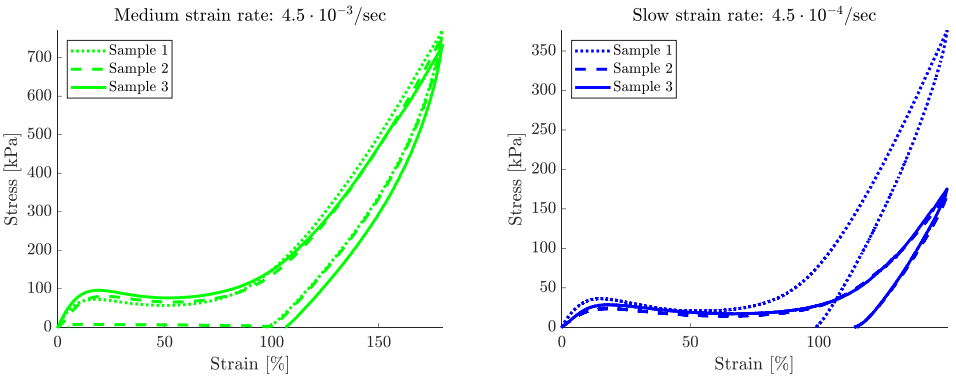
Results for polydomain samples at room temperature

Figure 5.8a shows the results for polydomain samples pulled til failure at three different strain rates, all at room temperature. The fastest strain rate corresponds to the stiffest response, as expected. Figure 5.8b plots 3 samples tested at the fastest strain rate with their load and unload curves. Figure 5.9a shows the same for the medium strain rate, and Figure 5.9b shows the same for the slowest strain rate.



(a) (b)

Figure 5.8: (a) Stress-strain curve of polydomain samples that were pulled til break. (b) Load and unload stress-strain curves for polydomain samples at fast strain rate.



(a) (b)

Figure 5.9: (a) Load and unload stress-strain curves for polydomain samples at medium strain rate. (b) Load and unload stress-strain curves for polydomain samples at slow strain rate.

Figure 5.13 shows the stress-strain curves for all the polydomain samples at all strain rates.

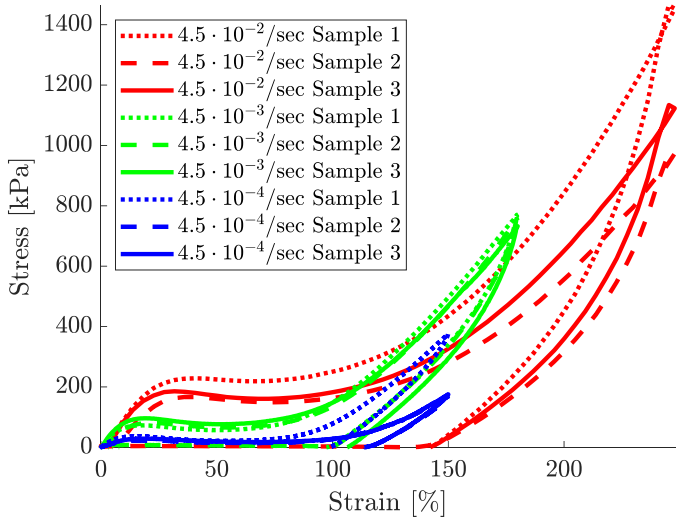
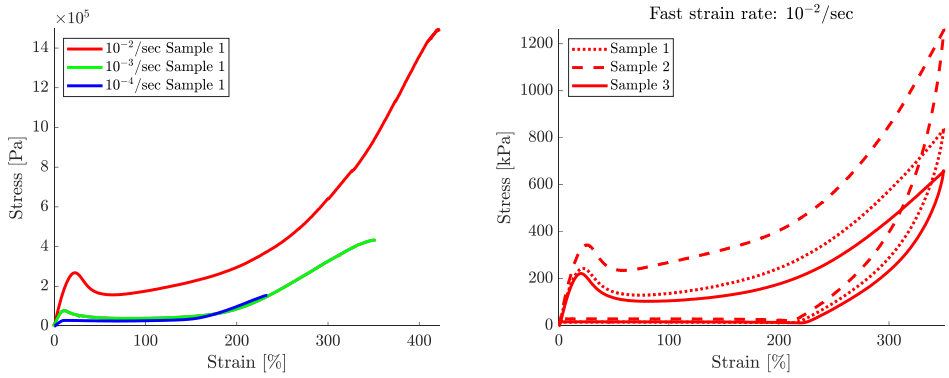


Figure 5.10: Stress-strain loading curves for all room-temperature polydomain samples at all strain rates.

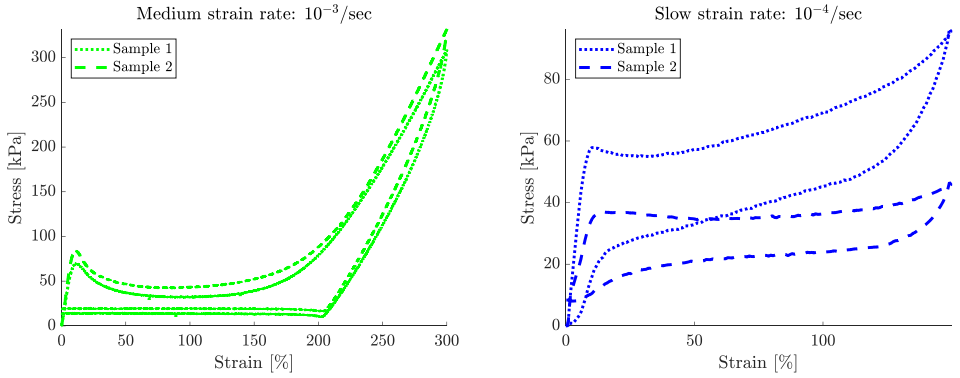
Results for monodomain samples at room temperature

Figure 5.11a shows the results from monodomain samples that were pulled perpendicular to their crosslinked nematic pattern until failure (all at room temperature). Figures 5.11b, 5.12a, and 5.12b show the load and unload curves for samples that were pulled at the fastest, medium, and slowest strain rates, respectively.



(a) (b)

Figure 5.11: (a) Stress-strain curve of monodomain samples that were pulled til break. (b) Load and unload stress-strain curves for monodomain samples at fast strain rate.



(a) (b)

Figure 5.12: (a) Load and unload stress-strain curves for monodomain samples at medium strain rate. (b) Load and unload stress-strain curves for monodomain samples at slow strain rate.

Figure 5.13 show the stress-strain response for all the monodomain samples at all strain rates at room temperature.

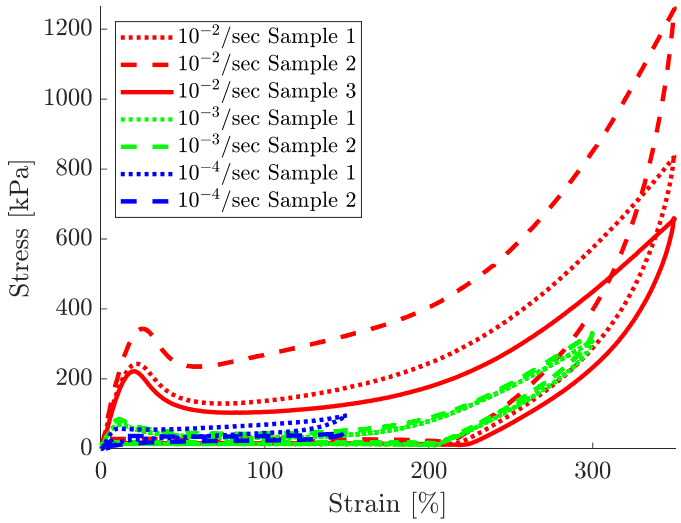


Figure 5.13: Stress-strain loading curves for all monodomain samples at all strain rates.

Figure 5.14 shows the image of stripe domains that formed in the uniaxial stretch of a monodomain sample. The stretch direction is vertical, and the stripes are parallel to the direction of stretch. The width of a stripe is about 75μm.

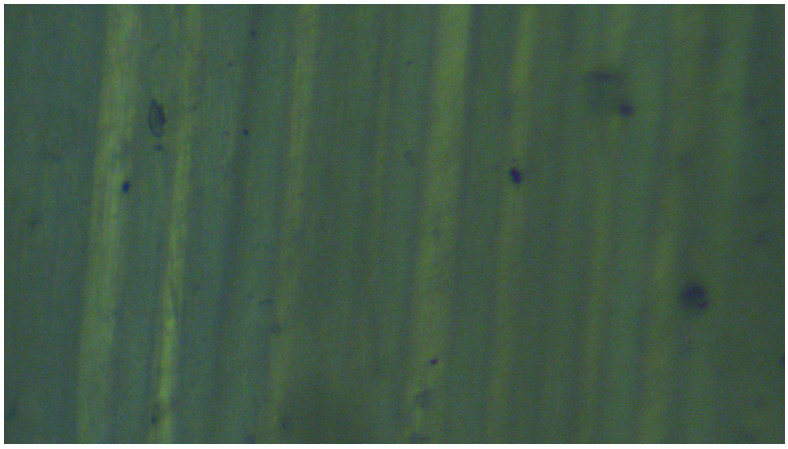
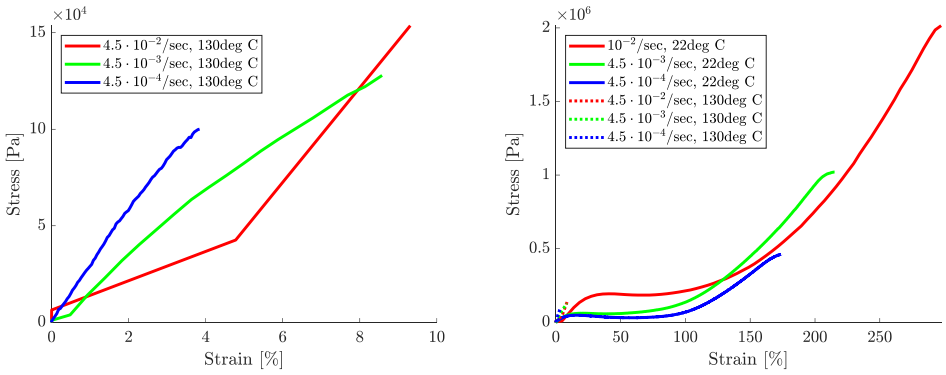


Figure 5.14: Image of stripe domains that appeared in the uniaxial stretch of a monodomain sample pulled perpendicular to its nematic alignment direction (tested at room temperature, imaged with cross-polarizers).

Results for polydomain samples at high temperature

In this section, polydomain samples were tested at high temperature, ≈ 130 degC. The entire testing chamber was preheated to the high temperature and then the samples were clamped in to begin the test. Figure 5.15a shows the stress-strain curves of the polydomain samples at all strain rates at high temperature, and Figure 5.15b shows the stress-strain curves of the polydomain samples at room temperature and high temperature for comparison.



(a) (b)

Figure 5.15: (a) Stress-strain loading curves for samples at high temperature at varying strain rates. (b) Stress-strain loading curves for all samples at low and high temperatures and varying strain rates. Note that the high-temperature curves appear very close to the origin.

The samples broke in the clamps extremely quickly at high temperature, so there was not much stress-strain data gathered in these tests. Strain rate does not have the expected effect upon the stress-strain curves in Figure 5.15a; the fastest strain rate corresponds to the softest sample, and the slowest strain rate corresponds to the stiffest sample. However, when analyzing the effect of temperature in Figure 5.15b, the three samples at high temperature are all stiffer than the three samples at low temperature, which is the expected effect.

Cyclic testing results

This test was conducted by Julia Combs. She tested a polydomain sample with 15 mol% PETMP at a strain rate of 10^{-2}/sec at room temperature. She used one sample throughout the twenty cycles. For a given cycle, she clamped it, loaded it, unloaded it, heated it up in the clamp (without unclamping it) using the heat gun, waited for it to cool ($\approx$ 3 min), then repeated. The results can be seen in Figure 5.16. The stress at 200% strain decreases with the number of cycles, while the stress at which mesogen reorientation occurs increases with the number of cycles. We suspect that some cycles have a higher stress plateau than others because the sample wasn't fully cooled down to room temperature, so the sample would be stiffer due to being in a more isotropic state.

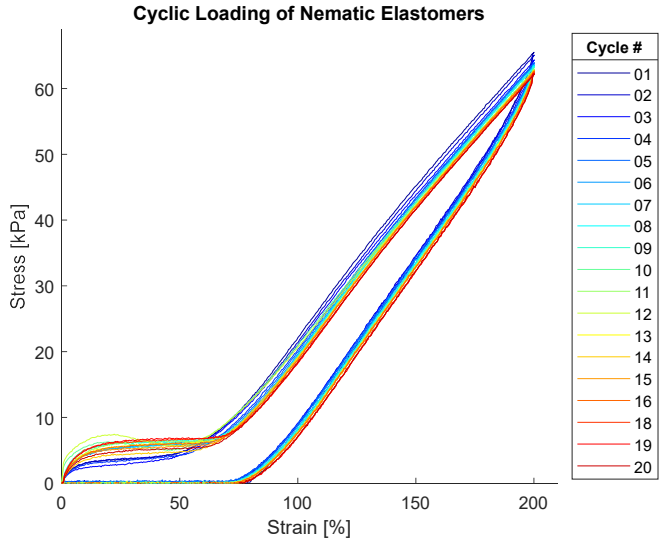


Figure 5.16: Cyclic stress-strain curves for a single sample over 20 cycles.

Repeatedly testing the same sample, or different samples from the same synthesis batch, would provide insight into the aging of the samples over time. Aging in polymers is well understood, but not well-studied in nematic elastomers. Aging

can depend upon the particular chemistry of synthesis, temperature, loading history, radiation, and more, and aging would affect nematic elastomers' use in various long-term applications.

5.5 Conclusion

In this chapter, we have described the synthesis of nematic elastomer samples that were tested in our experimental tensile test apparatus, which features a temperature-controlled chamber and polarized light microscopy capabilities. We presented the results for the uniaxial stretch of polydomain and monodomain nematic elastomers in their isotropic and nematic temperatures. We observed stripe domains in monodomain samples pulled perpendicular to their nematic alignment, and we have also conducted cyclic loading tests.

There exists much potential to build upon the current experimental setup for future research directions. For example, a future avenue of research could be obtaining full strain fields through digital image correlation (DIC). There are also some improvements that can be made to the experimental setup. The chamber design could be improved for easier clamping of samples when the setup is pre-heated, and the clamping mechanism could be improved beyond simple butterfly clamps.

Future experiments that could be conducted include uniaxial stretch of monodomain samples at varying angles with respect to director alignment, in-depth characterization of aging in samples over time, cavitation of a disk (for comparison with results from Chapter 3), and cyclic testing of samples at varying temperatures, strain rates, and director orientation.

Chapter 6

CONCLUSION AND FUTURE OUTLOOK

6.1 Summary and impact of the findings

Liquid crystal elastomers = liquid crystals + elastomer

At the heart of liquid crystal elastomers lies the interplay between two aspects of its microscopic makeup, i.e. the liquid crystal molecules embedded within a lightly crosslinked polymer backbone, and the macroscopic shape change. The microscale affects the macroscale, and vice versa. In response to an external stimulus such as change in temperature, nematic elastomers continuously deform between a high-temperature isotropic state, in which the liquid crystals are randomly oriented and there is no orientational order, and a low-temperature nematic state, in which it is energetically favorable for the rigid, rod-like liquid crystal molecules to align along a preferential direction. In return, external stretch of the material at the macroscale induces a change in the microscale, where the liquid crystal molecules reorient to align along the direction of most stretch. Furthermore, they have both liquid-like behavior (during mesogen reorientation) and solid-like behavior (during the elastic response of the elastomer chains). Finally, this unique microscopic structure leads to the ability for the material to develop fine-scale microstructure, and consequently exhibit their unusually soft behavior. These characteristics continue to inspire researchers in this field since the foundational stripe domains from the uniaxial stretch experiments of Kundler and Finkelmann [42].

Modeling nematic elastomers

This thesis utilized three different approaches to the continuum-level modeling of nematic elastomers. In Chapter 2, we introduced the classical model to characterize nematic elastomers, which is the Bladon-Terentjev-Warner (BTW) model, or “trace formula" [10]. However, because this model is an extension of the neo-Hookean model for rubbers and is therefore derived from Gaussian chain modeling, it works well to capture the material behavior at small to moderate stretches, but it fails to predict the stiffer behavior when the polymer chains are stretched sufficiently far such that Gaussian statistics are violated. For instance, the BTW model cannot capture the correct behavior at large stretch in the case of the expansion of a balloon subjected to internal pressure. Thus, we used the Ogden extension of the BTW

theory in Chapter 2, which allowed us to study the large-deformation behavior of a cylindrical nematic elastomer balloon and its application as a soft pump.

In Chapter 3, we introduced DeSimone and Dolzmann's theoretical model that captures fine-scale microstructure through relaxation of the BTW theory [23]. In the model, nematic elastomers can form domains of region L, denoting a liquid-like behavior, region M, denoting the formation of fine-scale microstructure, and region S, denoting the solid-like behavior where there is no nematic mesogen reorientation. Again, the BTW theory is unable to capture the correct physics of nematic elastomers at large stretch, so we employed a generalized Mooney-Rivlin model, following Silhavy, Agostiniani, and DeSimone [1, 65].

Finally, Chapter 4 discussed the new constitutive relation that we developed to describe non-ideal isotropic-genesis polydomain nematic elastomers. The model featured internal variables that corresponded to the formation of microstructure, which evolved according to a dissipative relation, and viscosity associated with the polymer network.

Cylindrical balloon and corresponding pump

In Chapter 2, we analyzed a monodomain cylindrical balloon in which the nematic director was axially oriented in its undeformed state. We found multiple equilibrium solutions. One solution was metastable, where the nematic mesogens would not rotate from their original axial orientation. The other two solutions featured fine-scale microstructure development, where the nematic director rotated symmetrically at a positive or negative angle in opposite directions to accommodate the deformation. The balloon would form alternating domains of the two symmetric solutions, and the macroscopic behavior of the twist would also alternate between the two solutions.

Then we described how such a cylindrical balloon could form a pumping cycle through a combination of inflow, outflow, and changing temperature. We measured the ejection fraction of the pump, which described how much fluid the balloon could pump in a given cycle, and found that it was extremely high due to the softness of the material and thus the nature of the pressure-volume curve.

This work was one of the first examples of actuation from geometries beyond flat, two-dimensional sheets, and analyzed nematic elastomers under a more complex loading and condition than previously considered.

Universal deformations of nematic elastomers

In Chapter 3, we studied the four deformations of cylindrical balloon expansion, spherical balloon expansion, cavitation of a disk, and bending of a block. We discussed how the materials can form regions of L, M, and S in response to the boundary conditions (e.g. for the spherical and cylindrical balloons, see Figures 3.3, 3.7 respectively). For the balloons, due to the boundary condition of internal pressure, the area of most stress was in the inner radius of the balloon, which was where the balloon formed microstructure first, before eventually the entire cross-section became enveloped by region M. We discussed how the deformation of the spherical balloon inflation was essentially the deformation of equibiaxial stretch, so it followed the EB curve of Figure 6.1, while the cylindrical balloon inflation with fixed axial stretch was a manifestation of planar extension, so its deformation path was the PE curve. Consequently, the spherical balloon remained in the microstructure-formation region for the rest of its deformation, whereas the cylindrical balloon stopped microstructure formation at a certain point and had a solid-like response at high stretch.

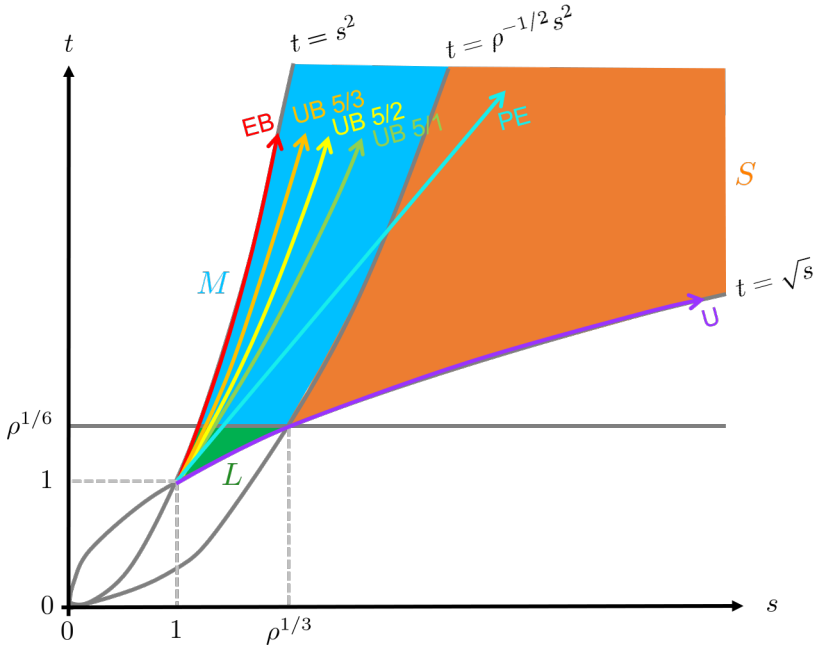


Figure 6.1: Regions of L, M, and S in the phase diagram of (s,t). Also shown are common deformations and their paths through the three regions: equibiaxial stretch (EB), unequal biaxial stretch (UB) with various strain ratios (5/3, 5/2, 5/1), planar extension (PE), and uniaxial stretch (U).

Furthermore, there was a spontaneous deformation associated with the temperature-induced phase transition, in which the surrounding polymer chains stretch parallel to the preferred direction and contract in the transverse directions. This phenomenon, which occurs at zero stress, was observed in both Chapters 2 and 3. In the case of the balloons subjected to internal pressure, for anisotropy parameters greater than 1, the pressure-stretch curves did not begin inflation at the point $(\lambda_o, p) = (1, 0)$. It was also seen clearly in the plot of moment vs. stretch for the deformation of bending. These spontaneous deformations associated with the unique thermo-mechanical coupling in nematic elastomers exemplify why these materials are so compelling to study.

We also discussed the cavitation of a disk, which was originally studied for ordinary rubber by Gent and Lindley [30], and the bending of a block. We compared the results between ordinary rubber response and the nematic elastomer response. Interestingly, in the bending deformation, we found that there were multiple equilibrium solutions as the deformation progressed, similar to the multiple equilibrium solutions found in Chapter 2.

Previous work in the field of nematic elastomers has focused on studying homogeneous deformations, so this chapter is a significant step towards understanding nematic elastomers under various inhomogeneous deformations.

Computational characterization of non-ideal, isotropic-genesis polydomain nematic elastomers

In Chapter 4, we developed an entirely new constitutive model to describe non-ideal, isotropic-genesis polydomain nematic elastomers. We implemented the model in the commercial finite-element software `ABAQUS` by writing a custom `UMAT`. We verified the `ABAQUS` code against the one-dimensional theory in `MATLAB` for various biaxial extension tests (uniaxial extension, planar extension, equibiaxial extension, and unequal biaxial extension), then validated the results from the model against experimental results for the same biaxial extension tests. We found great agreement between the simulation and the experiments, which were conducted by Tokumoto *et al.* [71]. We then studied cylinder torsion at various height-to-diameter ratios and anisotropy parameters. We compared the results between ordinary rubber and nematic elastomer and discussed the kinking instability.

Though nematic elastomers have been studied through multiple decades, there are still barriers in realizing these materials in engineering applications, unlike materials

such as shape memory alloys, which are commercially available and accepted in industry [11, 51]. The finite element model using our constitutive relation from Chapter 4 is a significant step towards a tool that can study nematic elastomers as an engineering material.

Experimental characterization of rate dependence and temperature dependence in polydomain and monodomain nematic elastomers

Finally, Chapter 5 described the synthesis of nematic elastomer samples that were tested in our experimental tensile test apparatus, which featured a temperature-controlled chamber. We presented the results for the uniaxial stretch of polydomain and monodomain nematic elastomers tested at their isotropic and nematic temperatures. The thermo-mechanical coupling inherent in nematic elastomers is responsible for the characteristic stress-strain curves presented in this chapter, for example in the polydomain-to-monodomain transition and in the monodomain-pull-perpendicular experiments that yielded the stripe domain patterns. We could clearly identify the distinct regimes in the stress-strain plots where the liquid crystal mesogen reorientation dominates vs. where the elasticity of the polymer backbone dominates.

The experimental setup also featured polarized light microscopy capabilities. Because of nematic elastomers' optical properties derived from the underlying liquid crystals, we were able to gain valuable information from viewing samples under cross-polarizers. We observed the formation of fine-scale microstructure, exhibited by stripe domains in monodomain samples pulled perpendicular to their nematic alignment. Finally, we also studied viscoelastic effects by studying the hysteresis at various strain rates and conducting a cyclic loading test.

6.2 Future outlook

Following the work presented in this thesis, there are many exciting future avenues of research within the field of nematic elastomers, and more broadly, active materials.

Theoretical characterization

The natural next steps in characterizing these materials theoretically are to use the relaxed generalized Mooney-Rivlin model to obtain analytical solutions to the remaining families of universal deformations that we did not cover in Chapter 3. Additionally, the cylindrical balloon deformation in Chapter 2 had multiple stable solutions, as did the bending deformation in Chapter 3. There is a lot of potential

for designing structures with multiple stable solutions for energy-efficiency reasons, so that the only energy expended is moving between the various stable states. The snap-through instability of Chapter 2 and the kinking instability of Chapter 4 are both interesting phenomena that result from finite elasticity at large deformation, and it would be interesting to continue studying other such instabilities in nematic elastomers.

Computational characterization

To build upon the computational work of Chapter 4, one could perform simulations of the universal deformations in `ABAQUS` using the constitutive relation for non-ideal polydomains, and investigating the effects of viscoelasticity, for instance plotting the hysteresis between loading and unloading. We also saw that interesting instabilities can arise, for instance in the torsion of a cylinder, opening the door to further investigation regarding the onset of such instabilities in nematic elastomers, the dependence of the instability upon material parameters and geometry, and more. With the `UMAT` built, further finite element simulations answering such questions are straightforward to run and analyze.

Experimental characterization

The natural next step in experimentally characterizing these materials is to expand upon the experimental results of Chapter 5 to build a complete set of material parameters to match an Ogden model for nematic elastomers for the ideal Bladon-Terentjev-Warner model with the non-ideality in Chapter 2, the relaxed generalized Mooney-Rivlin model of Chapter 3, and the constitutive relation for non-ideal polydomains for Chapter 4. Then, one can perform experiments on the expansion of a monodomain nematic elastomer balloon and quantitatively compare the expansion and twist parameters from Section 2.3, as well as manufacturing and testing a pump made from this monodomain balloon to construct pressure-volume curves at different anisotropy parameters, as described in Section 2.4. The physical size of such a pump could be on the order of centimeters, such as the balloon found in [34].

Furthermore, one can perform experiments within various classes of universal deformations, e.g. the bending of a polydomain block, inflation of polydomain balloons (spherical and cylindrical), and cavitation of a polydomain disk, and match the experimental results with the theoretical results of Chapter 3. Other experiments that were traditionally performed on thin films, such as the bulge and blister tests, would also be useful avenues of exploration to characterize the material.

Nematic elastomers and other active materials

Within the field of active materials, there are exciting paths forward leading towards the multifunctional, the adaptable, and the autonomous. We can think of the integration of active materials with origami/kirigami for shape-morphing applications, as well as designing adaptable features such as roughness and stickiness for bio-inspired soft robotics applications.

Thermotropic nematic elastomers are quick to heat, but the cooling time can be slow in ambient air based on the temperature differential and the geometry of the sample. Actuating within a bath increases response times but can be limiting depending on the application. Phototropic nematic elastomers, for instance, have better response times, although there are other issues associated with penetration depth of the light. The combination of such nematic elastomers responding to multiple stimuli, or the combination of various active materials responding to multiple stimuli, can create multifunctional structures in which the order and extent of the responses can be controlled and tuned for the desired actuation. Additionally, composites of nematic elastomers can be designed for one or more desired properties, e.g. fiber-reinforced elastomers for augmented mechanical behavior, stretchable wiring for augmented electrical capabilities. Composites of active materials can be optimized for various loading configurations using topology optimization.

As mentioned previously, we observe three distinct length scales in nematic elastomers: the nematic mesogens (order of nanometers), domains of nematic alignment (order of microns), and the macroscale (on the order of centimeters). However, if a desired application is of a different macroscopic length scale than this, perhaps designing artificial nematic elastomers, featuring a fundamental phase transition occurring at the smallest length scale fully coupled with shape change at the macroscale, might be a fruitful area of exploration.

BIBLIOGRAPHY

[1] V. Agostiniani and A. DeSimone. Ogden-type energies for nematic elastomers. *International Journal of Non-Linear Mechanics*, 47(2):402–412, 2012. ISSN 0020-7462. doi: 10.1016/J.IJNONLINMEC.2011.10.001.

[2] H. Aharoni, E. Sharon, and R. Kupferman. Geometry of Thin Nematic Elastomer Sheets. *Physical Review Letters*, 113(257801):257801, 2014. doi: 10.1103/PhysRevLett.113.257801.

[3] H. Aharoni, Y. Xia, X. Zhang, R. D. Kamien, and S. Yang. Universal inverse design of surfaces with thin nematic elastomer sheets. *Proc. Nat. Adad. Sci.*, 115(28):7201–7211, 2018. ISSN 10916490. doi: 10.1073/pnas.1804702115.

[4] C. P. Ambulo, J. J. Burroughs, J. M. Boothby, H. Kim, M. R. Shankar, and T. H. Ware. Four-dimensional Printing of Liquid Crystal Elastomers. *ACS Applied Materials and Interfaces*, 9(42):37332–37339, 2017. ISSN 1944-8244. doi: 10.1021/acsami.7b11851.

[5] A. Azoug, V. Vasconcellos, J. Dooling, M. Saed, C. Yakacki, and T. Nguyen. Viscoelasticity of the polydomain-monodomain transition in main-chain liquid crystal elastomers. *Polymer*, 98(98):165–171, 2016. ISSN 1359-0286. doi: 10.1016/j.cossms.2010.07.001.

[6] K. Bhattacharya and R. D. James. The material is the machine. *Science*, 307 (5706):53–54, 2005. ISSN 00368075. doi: 10.1126/science.1100892.

[7] J. S. Biggins, E. M. Terentjev, and M. Warner. Semisoft elastic response of nematic elastomers to complex deformations. *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics*, 78(4):1–9, 2008. ISSN 15393755. doi: 10.1103/PhysRevE.78.041704.

[8] J. S. Biggins, M. Warner, and K. Bhattacharya. Supersoft elasticity in polydomain nematic elastomers. *Physical Review Letters*, 103(3), 2009. ISSN 00319007. doi: 10.1103/PhysRevLett.103.037802.

[9] J. S. Biggins, M. Warner, and K. Bhattacharya. Elasticity of polydomain liquid crystal elastomers. *Journal of the Mechanics and Physics of Solids*, 60 (4):573–590, 2012. ISSN 00225096. doi: 10.1016/j.jmps.2012.01.008.

[10] P. Bladon, M. Terentjev, and M. Warner. Transitions and instabilities in liquid-crystal elastomers. *Physical Review E*, 47(6):3838–3840, 1993.

[11] Boeing. Boeing Frontiers, 2006. URL `https://www.boeing.com/news/frontiers/archive/2006/march/i_tt.html`.

[12] J. M. Boothby and T. H. Ware. Dual-responsive, shape-switching bilayers enabled by liquid crystal elastomers. *Soft Matter*, 13(24):4349–4356, 2017. ISSN 1744-683X. doi: 10.1039/C7SM00541E.

[13] M. Camacho-Lopez, H. Finkelmann, P. Palffy-Muhoray, and M. Shelley. Fast liquid-crystal elastomer swims into the dark. *Nature Materials*, 3(5):307–310, 2004. ISSN 1476-1122. doi: 10.1038/nmat1118.

[14] C. J. Camargo, H. Campanella, J. E. Marshall, N. Torras, K. Zinoviev, E. M. Terentjev, and J. Esteve. Batch fabrication of optical actuators using nanotube-elastomer composites towards refreshable Braille displays. *Journal of Micromechanics and Microengineering*, 22(7), 2012. ISSN 09601317. doi: 10.1088/0960-1317/22/7/075009.

[15] S. A. Chester. Abaqus Subroutines. URL `https://web.njit.edu/$\sim$sac3/Software.html`.

[16] S. M. Clarke and E. M. Terentjev. Slow stress relaxation in liquid crystal elastomers and gels. *Faraday Discussions*, 112:352–333, 1999.

[17] S. M. Clarke, A. R. Tajbakhsh, E. M. Terentjev, C. Remillat, G. R. Tomlinson, and J. R. House. Soft elasticity and mechanical damping in liquid crystalline elastomers. *Journal of Applied Physics*, 89(11 I):6530–6535, 2001. ISSN 00218979. doi: 10.1063/1.1368177.

[18] S. Conti, A. DeSimone, and G. Dolzmann. Soft elastic response of stretched sheets of nematic elastomers: a numerical study. *Journal of the Mechanics and Physics of Solids*, 50(7):1431–1451, 2002. ISSN 0022-5096. doi: 10.1016/S0022-5096(01)00120-X.

[19] S. Conti, A. Desimone, and G. Dolzmann. Semisoft elasticity and director reorientation in stretched sheets of nematic elastomers. *Physical Review E*, 66 (061710), 2002. doi: 10.1103/PhysRevE.66.061710.

[20] J. Coyne. Analysis of the Formation and Elimination of Loops in Twisted Cable. *IEEE Journal of Oceanic Engineering*, 15(2):72–83, 1990. ISSN 15581691. doi: 10.1109/48.50692.

[21] P.-G. de Gennes. Réflexions sur un type de polymères nématiques. *Comptes rendus de l'Académie des Sciences, Série B*, 281:101–103, 1975.

[22] L. T. de Haan, C. Sánchez Somolinos, C. M. W. Bastiaansen, A. P. H. J. Schenning, and D. J. Broer. Engineering of Complex Order and the Macroscopic Deformation of Liquid Crystal Polymer Networks. *Angewandte Chemie International Edition*, 51(50):12469–12472, 2012.

[23] A. DeSimone and G. Dolzmann. Macroscopic Response of Nematic Elastomers via Relaxation of a Class of SO(3)-Invariant Energies. *Archive for*

Rational Mechanics and Analysis, 161(3):181–204, 2002. ISSN 0003-9527. doi: 10.1007/s002050100174.

[24] A. Documentation. Eigenvalue buckling prediction, 2020.

[25] J. L. Ericksen. Deformations possible in every isotropic, incompressible, perfectly elastic body. *Zeitschrift für angewandte Mathematik und Physik ZAMP*, 5(6):466–489, 1954. ISSN 00442275. doi: 10.1007/BF01601214.

[26] M. J. Ford, C. P. Ambulo, T. A. Kent, E. J. Markvicka, C. Pan, J. Malen, T. H. Ware, and C. Majidi. A multifunctional shape-morphing elastomer with liquid metal inclusions. *Proceedings of the National Academy of Sciences of the United States of America*, 116(43):21438–21444, 2019. ISSN 10916490. doi: 10.1073/pnas.1911021116.

[27] A. N. Gent. A new constitutive relation for rubber. *Rubber Chemistry and Technology*, 69(1):59–61, 1996. ISSN 00359475. doi: 10.5254/1.3538357.

[28] A. N. Gent. Elastic instabilities of inflated rubber shells. *Rubber Chemistry and Technology*, 72(2):263–268, 1999. doi: 10.5254/1.3538799.

[29] A. N. Gent and K. C. Hua. Torsional instability of stretched rubber cylinders. *International Journal of Non-Linear Mechanics*, 39(3):483–489, 2004. ISSN 00207462. doi: 10.1016/S0020-7462(02)00217-2.

[30] A. N. Gent and P. B. Lindley. Internal rupture of bonded rubber cylinders in tension. *Proceedings of the Royal Society A*, 249(1257):195–205, 1959. ISSN 2053-9169. doi: 10.1098/rspa.1959.0016.

[31] A. Giudici and J. S. Biggins. Giant deformations and soft-inflation in LCE balloons. *EPL*, 132(3):1–7, 2020. ISSN 12864854. doi: 10.1209/0295-5075/132/36001.

[32] A. S. Gladman, E. A. Matsumoto, R. G. Nuzzo, L. Mahadevan, J. A. Lewis, A. Sydney Gladman, E. A. Matsumoto, R. G. Nuzzo, L. Mahadevan, and J. A. Lewis. Biomimetic 4D Printing. *Nat. Mater.*, 15(4):413?418, 2016. ISSN 14764660. doi: 10.1038/nmat4544.

[33] N. Goulbourne, E. Mockensturm, and M. Frecker. A nonlinear model for dielectric elastomer membranes. *Journal of Applied Mechanics*, 72(6):899–906, 2005.

[34] Q. He, Y. Zheng, Z. Wang, X. He, and S. Cai. Anomalous inflation of a nematic balloon. *J Mech Phys Solids*, 142(104013):104013, 2020. ISSN 00225096. doi: 10.1016/j.jmps.2020.104013.

[35] A. Hotta and E. Terentjev. Dynamic soft elasticity in monodomain nematic elastomers. *The European Physical Journal E*, 10(4):291–301, 2003. ISSN 1292-8941. doi: 10.1140/epje/i2002-10005-5.

[36] A. Hotta and E. M. Terentjev. Long-time stress relaxation in polyacrylate nematic liquid crystalline elastomers. *Journal of Physics: Condensed Matter*, 13(50):11453–11464, 2001. ISSN 0953-8984. doi: 10.1088/0953-8984/13/50/305.

[37] A. Kelly, A. P. Stebner, and K. Bhattacharya. A micromechanics-inspired constitutive model for shape-memory alloys that accounts for initiation and saturation of phase transformation. *Journal of the Mechanics and Physics of Solids*, 97:197–224, 2016. ISSN 00225096. doi: 10.1016/j.jmps.2016.02.007.

[38] W. W. Klingbeil and R. T. Shield. On a class of solutions in plane finite elasticity. *Zeitschrift für angewandte Mathematik und Physik ZAMP*, 17(4):489–511, 1966. ISSN 00442275. doi: 10.1007/BF01595984.

[39] A. Kosaraju, A. Goyal, Y. Grigorova, and A. N. Makaryus. Left Ventricular Ejection Fraction. [Updated 2020 May 5]. In *StatPearls [Internet]*. StatPearls Publishing, Treasure Island, 2020.

[40] A. Kotikian, R. L. Truby, J. W. Boley, T. J. White, and J. A. Lewis. 3D Printing of Liquid Crystal Elastomeric Actuators with Spatially Programed Nematic Order. *Advanced Materials*, 30(10):1706164, 2018. ISSN 09359648. doi: 10.1002/adma.201706164.

[41] A. Kotikian, C. McMahan, E. C. Davidson, J. M. Muhammad, R. D. Weeks, C. Daraio, and J. A. Lewis. Untethered soft robotic matter with passive control of shape morphing and propulsion. *Science Robotics*, 4(33):7044, 2019. ISSN 24709476. doi: 10.1126/scirobotics.aax7044.

[42] I. Kundler and H. Finkelmam. Strain-induced director reorientation in nematic liquid single crystal elastomers. *Macromol. Rapid Commun*, 16:679–686, 1995. ISSN 1022-1336.

[43] J. Küpfer and H. Finkelmann. Liquid crystal elastomers: Influence of the orientational distribution of the crosslinks on the phase behaviour and reorientation processes. *Macromolecular Chemistry and Physics*, 195(4):1353–1367, 1994. ISSN 10221352. doi: 10.1002/macp.1994.021950419.

[44] A. K. Landauer, X. Li, C. Franck, and D. L. Henann. Experimental characterization and hyperelastic constitutive modeling of open-cell elastomeric foams. *Journal of the Mechanics and Physics of Solids*, 133:103701, 2019. ISSN 00225096. doi: 10.1016/j.jmps.2019.103701.

[45] C. Legge, F. Davis, and G. Mitchell. Memory effects in liquid crystal elastomers. *Journal de Physique II*, 1(10):1253–1261, 1991.

[46] D. R. Merkel, R. K. Shaha, C. M. Yakacki, and C. P. Frick. Mechanical energy dissipation in polydomain nematic liquid crystal elastomers in response to oscillating loading. *Polymer*, 166:148–154, 2019. ISSN 0032-3861. doi: 10.1016/J.POLYMER.2019.01.042.

[47] C. Modes and M. Warner. Shape-programmable materials. *Physics Today*, 69 (1):32–38, 2016. doi: 10.1063/PT.3.3051.

[48] C. D. Modes, K. Bhattacharya, and M. Warner. Disclination-mediated thermo-optical response in nematic glass sheets. *Physical Review E*, 81(60701), 2010.

[49] C. D. Modes, K. Bhattacharya, and M. Warner. Gaussian curvature from flat elastica sheets. *Proceedings of the Royal Society A*, 467:1121–1140, 2011. ISSN 1364-5021. doi: 10.1098/rspa.2010.0352.

[50] C. Mostajeran, M. Warner, T. H. Ware, and T. J. White. Encoding Gaussian curvature in glassy and elastomeric liquid crystal solids. *Proceedings of the Royal Society A*, 472(20160112), 2016.

[51] NASA. Metal with Memory: Shaping the Future of Aviation, 2017. URL `https://www.nasa.gov/feature/metal-with-memory-shaping-the-future-of-aviation`.

[52] N. Nguyen and A. M. Waas. Nonlinear, finite deformation, finite element analysis. *Zeitschrift für angewandte Mathematik und Physik ZAMP*, 67(3): 1–24, 2016. ISSN 00442275. doi: 10.1007/s00033-016-0623-5.

[53] R. W. Ogden. *Non-linear elastic deformations*. Ellis Harwood Ltd., Chichester, 1984. ISBN 0486696480.

[54] C. Ortiz, C. K. Ober, and E. Kramer. Stress relaxation of a main-chain, smectic, polydomain liquid crystalline elastomer. *Polymer*, 39(16):3713–3718, 1998. ISSN 0032-3861. doi: 10.1016/S0032-3861(97)10321-4.

[55] Z. Pei, Y. Yang, Q. Chen, E. M. Terentjev, Y. Wei, and Y. Ji. Mouldable liquid-crystalline elastomer actuators with exchangeable covalent bonds. *Nature Materials*, 13(1):36–41, 2013. ISSN 1476-1122. doi: 10.1038/nmat3812.

[56] P. Plucinsky and K. Bhattacharya. Microstructure-enabled control of wrinkling in nematic elastomer sheets. *Journal of the Mechanics and Physics of Solids*, 102:125–150, 2017. ISSN 0022-5096. doi: 10.1016/J.JMPS.2017.02.009.

[57] P. Plucinsky, M. Lemm, and K. Bhattacharya. Programming complex shapes in thin nematic elastomer and glass sheets. *Physical Review E*, 94(1), 2016. doi: 10.1103/PhysRevE.94.010701.

[58] P. Plucinsky, B. A. Kowalski, T. J. White, and K. Bhattacharya. Patterning nonisometric origami in nematic elastomer sheets. *Soft Matter*, 14:3127–3134, 2018. ISSN 1744-683X. doi: 10.1039/C8SM00103K.

[59] J. H. Poynting. On the Changes in the Dimensions of a Steel Wire when Twisted, and on the Pressure of Distortional Waves in Steel. *Proceedings of the Royal Society A*, 86:534–561, 1912.

[60] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in F77*. Cambridge University Press, 1992. ISBN 052143064X.

[61] R. S. Rivlin. Large elastic deformations of isotropic materials VI. Further results in the theory of torsion, shear and flexure. *Philosophical Transactions of the Royal Society A*, 845(242):173–195, 1949.

[62] R. S. Rivlin and D. W. Saunders. Large Elastic Deformations of Isotropic Materials VII. Experiments on the Deformation of Rubber. *Philosophical Transactions of the Royal Society of London. Series A, Mathematical and Physical Sciences*, 243(865), 1951. doi: 10.1007/978-1-4612-2416-7_8.

[63] S. Rudykh, K. Bhattacharya, and G. Debotton. Snap-through actuation of thick-wall electroactive balloons. *International Journal of Non-Linear Mechanics*, 47(2):206–209, 2012. ISSN 00207462. doi: 10.1016/j.ijnonlinmec.2011.05.006.

[64] M. O. Saed, A. H. Torbati, D. P. Nair, and C. M. Yakacki. Synthesis of Programmable Main-chain Liquid-crystalline Elastomers Using a Two-stage Thiol-acrylate Reaction. *Journal of Visualized Experiments*, 107:1–10, 2016. ISSN 1940-087X. doi: 10.3791/53546.

[65] M. Silhavý. Ideally soft nematic elastomers. *Networks and Heterogeneous Media*, 2(2):279–311, 2007. ISSN 1556-1801. doi: 10.3934/nhm.2007.2.279.

[66] M. Singh and A. C. Pipkin. Note on Ericksen's problem. *Zeitschrift für angewandte Mathematik und Physik ZAMP*, 16(5):706–709, 1965. ISSN 00442275. doi: 10.1007/BF01590971.

[67] P. Stein, N. Aßfalg, H. Finkelmann, and P. Martinoty. Shear modulus of polydomain, mono-domain and non-mesomorphic side-chain elastomers: Influence of the nematic order. *European Physical Journal E*, 4(3):255–262, 2001. ISSN 12928941. doi: 10.1007/s101890170107.

[68] M. Tabrizi, T. H. Ware, and M. R. Shankar. Voxelated Molecular Patterning in Three-Dimensional Freeforms. *ACS Applied Materials and Interfaces*, 11 (31):28236–28245, 2019.

[69] E. M. Terentjev, A. Hotta, S. M. Clarke, and M. Warner. Liquid Crystalline Elastomers: Dynamics and Relaxation of Microstructure. *Philosophical Transactions of the Royal Society A*, 361(1805):653–664, 2003. ISSN 1364-503X. doi: 10.1098/rsta.2002.1155.

[70] J. M. Thompson and A. R. Champneys. From helix to localized writhing in the torsional post-buckling of elastic rods. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 452(1944):117–138, 1996. ISSN 13645021. doi: 10.1098/rspa.1996.0007.

[71] H. Tokumoto, H. Zhou, A. Takebe, K. Kamitani, K. Kojio, A. Takahara, K. Bhattacharya, and K. Urayama. Probing the in-plane liquid-like behavior of liquid crystal elastomers. To appear. *Science Advances*, 2021.

[72] L. R. G. Treloar. *The Physics of Rubber Elasticity*. Oxford University Press, Oxford, 1975.

[73] L. R. G. Treloar. *The Physics of Rubber Elasticity*. Oxford University Press, Oxford, 3rd edition, 2005. ISBN 9780198570257.

[74] K. Urayama, E. Kohmon, M. Kojima, and T. Takigawa. Polydomain–Monodomain Transition of Randomly Disordered Nematic Elastomers with Different Cross-Linking Histories. *Macromolecules*, 42(12):4084–4089, 2009. ISSN 0024-9297. doi: 10.1021/ma9004692.

[75] T. H. Ware and T. J. White. Programmed liquid crystal elastomers with tunable actuation strain. *Polymer Chemistry*, 6:4835–4844, 2015. doi: 10.1039/c5py00640f.

[76] T. H. Ware, M. E. McConney, J. J. Wie, V. P. Tondiglia, and T. J. White. Voxelated liquid crystal elastomers. *Science*, 347(6225):982–984, 2015. ISSN 0036-8075. doi: 10.1126/science.1261019.

[77] M. Warner and E. M. Terentjev. *Liquid Crystal Elastomers*. Oxford University Press, Oxford, 2003. ISBN 0198527675.

[78] G. Weber and L. Anand. Finite deformation constitutive equations and a time integration procedure for isotropic, hyperelastic-viscoplastic solids. *Computer Methods in Applied Mechanics and Engineering*, 79(2):173–202, 1990. ISSN 00457825. doi: 10.1016/0045-7825(90)90131-5.

[79] T. White. *Photomechanical Materials, Composites, and Systems: Wireless Transduction of Light into Work*. John Wiley and Sons, 2017. ISBN 978-1-119-12330-9.

[80] C. M. Yakacki, M. Saed, D. P. Nair, T. Gong, S. M. Reed, and C. N. Bowman. Tailorable and programmable liquid-crystalline elastomers using a two-stage thiol–acrylate reaction. *RSC Advances*, 5:18997, 2015. doi: 10.1039/c5ra01039j.

[81] M. Yamada, M. Kondo, R. Miyasato, Y. Naka, J.-i. Mamiya, M. Kinoshita, A. Shishido, Y. Yu, C. J. Barrett, and T. Ikeda. Photomobile polymer materials—various three-dimensional movements. *Journal of Materials Chemistry*, 19(1):60, 2009. ISSN 0959-9428. doi: 10.1039/b815289f.

[82] H. Yang, A. Buguin, J.-M. Taulemesse, K. Kaneko, S. Méry, A. Bergeret, and P. Keller. Micron-sized main-chain liquid crystalline elastomer actuators with ultralarge amplitude contractions. *Journal of the American Chemical Society*, 131(41):15000–15004, 2009. ISSN 00027863. doi: 10.1021/ja905363f.

[83] C. Yuan, D. J. Roach, C. K. Dunn, Q. Mu, X. Kuang, C. M. Yakacki, T. J. Wang, K. Yu, and H. J. Qi. 3D printed reversible shape changing soft actuators assisted by liquid crystal elastomers. *Soft Matter*, 13(33):5558–5568, 2017. ISSN 1744-683X. doi: 10.1039/C7SM00759K.

[84] X. Zhao and Z. Suo. Method to analyze electromechanical instability of dielectric elastomers. *Applied Physics Letters*, 91:61921, 2007.

[85] H. Zhou and K. Bhattacharya. Accelerated computational micromechanics and its applications to nematic elastomers. To appear. *Journal of the Mechanics and Physics of Solids*, 2021.

Appendix A

SUPPLEMENTARY INFORMATION IN DEVELOPING THE GENERALIZED MOONEY-RIVLIN MODEL

A.1 Principal stretches

Recall that the cofactor of a second-order tensor $\boldsymbol{F}$ is $\operatorname{cof}\boldsymbol{F} = (\det\boldsymbol{F})\,\boldsymbol{F}^{-\top}$ when $\det\boldsymbol{F} \neq 0$. Let $\lambda_{max} \geq \lambda_{mid} \geq \lambda_{min}$ be the singular values or principal stretches of $\boldsymbol{F}$, and let $\lambda_{max}^2 \geq \lambda_{mid}^2 \geq \lambda_{min}^2$ be the eigenvalues of $\boldsymbol{F}\boldsymbol{F}^{\top}$. Let $s = \lambda_{max}$ be the largest singular value of $\boldsymbol{F}$, and let $t = \lambda_{max}\lambda_{mid}$ be the largest singular value of $\operatorname{cof}\boldsymbol{F}$. We can summarize the principal stretches in terms of s and t:

$$\lambda_{max} = s,\ \lambda_{mid} = \frac{t}{s},\ \lambda_{min} = \frac{1}{t}, \tag{A.1}$$

where we have used incompressibility for the last relation ($\lambda_{max}\lambda_{mid}\lambda_{min} = 1$).

A.2 Minimization of the energy with respect to the nematic director

Energy based on the first invariant

The energy for nematic elastomers based on $\boldsymbol{F}$ is as follows:

$$\tilde{W}_1 = \min_{\boldsymbol{n}\ \text{s.t.}\ |\boldsymbol{n}|=1} \left\{ c \left| \operatorname{tr}\left(\boldsymbol{\ell}_n^{-1}\boldsymbol{b}\right) - 3 \right|^p \right\}. \tag{A.2}$$

Because I_1 is an invariant, its value is the same no matter what basis is chosen. Therefore, let us choose the principal basis:

$$\boldsymbol{b} = \begin{pmatrix} \lambda_{max}^2 & & \\ & \lambda_{mid}^2 & \\ & & \lambda_{min}^2 \end{pmatrix}. \tag{A.3}$$

Minimizing the energy over the director $\boldsymbol{n}$ yields the result that $\boldsymbol{n}$ is parallel to the eigenvector corresponding to the largest eigenvalue, λ_{max}^2. That is, $\boldsymbol{n} = (1,0,0)^{\top}$. Thus, the step-length tensor should be as follows:

$$\boldsymbol{\ell}_n^{-1} = r^{1/3}\left(\boldsymbol{I} + \left(\frac{1}{r} - 1\right)\boldsymbol{n}\otimes\boldsymbol{n}\right) \tag{A.4}$$

$$= \begin{pmatrix} r^{-2/3} & & \\ & r^{1/3} & \\ & & r^{1/3} \end{pmatrix}. \tag{A.5}$$

Therefore, we have

$$\tilde{W}_1(s,t) = c\left|\mathrm{tr}\left(\ell_n^{-1}\boldsymbol{b}\right) - 3\right|^p \tag{A.6}$$

$$= c\left|r^{1/3}\left(\frac{\lambda_{max}^2}{r} + \lambda_{mid}^2 + \lambda_{min}^2\right) - 3\right|^p \tag{A.7}$$

$$= c\left|r^{1/3}\left(\frac{s^2}{r} + \frac{t^2}{s^2} + \frac{1}{t^2}\right) - 3\right|^p, \tag{A.8}$$

where in the last equality we have used the fact that $s = \lambda_{max}$ and $t = \lambda_{max}\lambda_{mid}$.

A.3 Energy based on the second invariant

The energy for nematic elastomers based on cof $\boldsymbol{F}$ is as follows:

$$\tilde{W}_2 = \min_{\boldsymbol{n}\text{ s.t. }|\boldsymbol{n}|=1}\left\{d\left|\mathrm{tr}\left[\left(\boldsymbol{F}^\top\ell_n^{-1}\boldsymbol{F}\right)^{-1}\right] - 3\right|^q\right\}. \tag{A.9}$$

As in Section A.2, we will choose to work in the principal basis because I_2 is an invariant, which is independent of basis:

$$\boldsymbol{b} = \begin{pmatrix}\lambda_{max}^2 & & \\ & \lambda_{mid}^2 & \\ & & \lambda_{min}^2\end{pmatrix}. \tag{A.10}$$

Similarly to the case of W_{I_1}, minimizing the energy over the director $\boldsymbol{n}$ yields the result that $\boldsymbol{n}$ is parallel to the eigenvector corresponding to the largest eigenvalue, λ_{max}^2, making $\boldsymbol{n} = (1,0,0)^\top$. Thus, the step-length tensor should be as follows:

$$\ell_n^{-1} = r^{1/3}\left(\boldsymbol{I} + \left(\frac{1}{r} - 1\right)\boldsymbol{n}\otimes\boldsymbol{n}\right) \tag{A.11}$$

$$= \begin{pmatrix}r^{-2/3} & & \\ & r^{1/3} & \\ & & r^{1/3}\end{pmatrix}. \tag{A.12}$$

Therefore, we have

$$\tilde{W}_2(s,t) = d\left|\mathrm{tr}\left[\left(\ell_n^{-1}\boldsymbol{b}\right)^{-1}\right] - 3\right|^q \tag{A.13}$$

$$= d|r^{-1/3}\left(\frac{r}{\lambda_{max}^2} + \frac{1}{\lambda_{mid}^2} + \frac{1}{\lambda_{min}^2}\right) - 3|^q \tag{A.14}$$

$$= d\left|r^{-1/3}\left(\frac{r}{s^2} + \frac{s^2}{t^2} + t^2\right) - 3\right|^q. \tag{A.15}$$

Again, in the last equality we have used the fact that $s = \lambda_{max}$ and $t = \lambda_{max}\lambda_{mid}$.

A.4 Simplification of the regions

The regions are defined by the following constraints:

$$\begin{aligned} L &: \{(s,t) : t \leq s^2,\ t \geq \sqrt{s},\ t \leq r^{1/6}\} \\ M &: \{(s,t) : t \geq r^{1/6},\ t \leq s^2,\ t \geq r^{-1/2}s^2\} \\ S &: \{(s,t) : t \geq \sqrt{s},\ t \leq r^{-1/2}s^2\} \end{aligned} \tag{A.16}$$

We can rewrite these constraints in terms of the principal stretches of $\boldsymbol{F}$, $\lambda_{max} \geq \lambda_{mid} \geq \lambda_{min}$, by using the relations found in Equation A.1. For region L, the first constraint is:

$$t \leq s^2 \tag{A.17}$$

$$\lambda_{max}\lambda_{mid} \leq \lambda_{max}^2 \tag{A.18}$$

$$\lambda_{mid} \leq \lambda_{max} \tag{A.19}$$

This is true always, so it is not necessary to state this as a constraint. Similarly, the second constraint simplifies to:

$$t \geq \sqrt{s} \tag{A.20}$$

$$\lambda_{max}\lambda_{mid} \geq \sqrt{\lambda_{max}} \tag{A.21}$$

$$\lambda_{max}^2\lambda_{mid}^2 \geq \lambda_{max} \tag{A.22}$$

$$\lambda_{max}\lambda_{mid}^2 \geq 1 \tag{A.23}$$

$$\frac{1}{\lambda_{min}}\lambda_{mid} \geq 1 \tag{A.24}$$

$$\lambda_{mid} \geq \lambda_{min} \tag{A.25}$$

Again, this relation is true always. Thus, the last constraint in region L becomes

$$t \leq r^{1/6} \tag{A.26}$$

$$\lambda_{max}\lambda_{mid} \leq r^{1/6} \tag{A.27}$$

In region M, the first constraint is

$$t \geq r^{1/6} \tag{A.28}$$

$$\lambda_{max}\lambda_{mid} \geq r^{1/6} \tag{A.29}$$

The second constraint in M is the same as the first constraint in region L. The third constraint in M is

$$t \geq r^{-1/2}s^2 \tag{A.30}$$

$$\lambda_{max}\lambda_{mid} \geq r^{-1/2}\lambda_{max}^2 \tag{A.31}$$

$$\frac{\lambda_{max}}{\lambda_{mid}} \leq r^{1/2} \tag{A.32}$$

In region S, the first constraint is the same as the second constraint in region L. The second constraint in S is similar to the third constraint in M:

$$t \leq r^{-1/2} s^2 \tag{A.33}$$

$$\frac{\lambda_{max}}{\lambda_{mid}} \geq r^{1/2} \tag{A.34}$$

Thus, we have:

$$\begin{aligned} L &: \{\lambda_{max}\lambda_{mid} \leq r^{1/6}\} \\ M &: \{\lambda_{max}\lambda_{mid} \geq r^{1/6},\ \frac{\lambda_{max}}{\lambda_{mid}} \leq \sqrt{r}\} \\ S &: \{\frac{\lambda_{max}}{\lambda_{mid}} \geq \sqrt{r}\} \end{aligned} \tag{A.35}$$

Appendix B

DERIVING DDSDDE FOR THE UMAT

B.1 Useful items for deriving the material Jacobian

Here is a useful derivative:

$$\dot{J} = \frac{\partial(\det \boldsymbol{F})}{\partial F_{iJ}} \frac{\partial F_{iJ}}{\partial t} \tag{B.1}$$

$$= J F^{-\top}_{iJ} \dot{F}_{iJ} \tag{B.2}$$

$$= J \dot{F}_{iJ} F^{-1}_{Ji} \tag{B.3}$$

$$= J \mathrm{tr}\, \boldsymbol{\ell} \tag{B.4}$$

Additionally, we can prove that

$$\mathrm{tr}\, \boldsymbol{\ell} = \mathrm{tr}\, \boldsymbol{d} \tag{B.5}$$

by

$$\mathrm{tr}\, \boldsymbol{d} = \mathrm{tr} \left[\frac{1}{2} \left(\boldsymbol{\ell} + \boldsymbol{\ell}^{\top} \right) \right] \tag{B.6}$$

$$= \frac{1}{2} \left[\mathrm{tr}(\boldsymbol{\ell}) + \mathrm{tr}(\boldsymbol{\ell}^{\top}) \right] \tag{B.7}$$

$$= \frac{1}{2} \left[\mathrm{tr}(\boldsymbol{\ell}) + \mathrm{tr}(\boldsymbol{\ell}) \right] \tag{B.8}$$

$$= \mathrm{tr}\, \boldsymbol{\ell} \tag{B.9}$$

or the fact that

$$\mathrm{tr}(\boldsymbol{\ell}) = \mathrm{tr}(\boldsymbol{d} + \boldsymbol{w}) = \mathrm{tr}(\boldsymbol{d}) \tag{B.10}$$

since $\mathrm{tr}(\boldsymbol{w}) = 0$.

B.2 Auxiliary remark for the material Jacobian

The time derivative of the right Cauchy-Green tensor can be related to the rate of deformation tensor:

$$\dot{\boldsymbol{C}} = \overline{\dot{\boldsymbol{F}^{\top} \boldsymbol{F}}} \tag{B.11}$$

$$= \dot{\boldsymbol{F}}^{\top} \boldsymbol{F} + \boldsymbol{F}^{\top} \dot{\boldsymbol{F}} \tag{B.12}$$

$$= \boldsymbol{F}^{\top} \boldsymbol{F}^{-\top} \dot{\boldsymbol{F}}^{\top} \boldsymbol{F} + \boldsymbol{F}^{\top} \dot{\boldsymbol{F}} \boldsymbol{F}^{-1} \boldsymbol{F} \tag{B.13}$$

$$= \boldsymbol{F}^{\top} \boldsymbol{\ell}^{\top} \boldsymbol{F} + \boldsymbol{F}^{\top} \boldsymbol{\ell} \boldsymbol{F} \tag{B.14}$$

$$= 2 \boldsymbol{F}^{\top} \boldsymbol{d} \boldsymbol{F} \tag{B.15}$$

Thus,

$$\mathrm{tr}\left(\tilde{\boldsymbol{G}}^{-1}\dot{\boldsymbol{C}}\right) = \mathrm{tr}\left(2\tilde{\boldsymbol{G}}^{-1}\boldsymbol{F}^{\top}\boldsymbol{dF}\right) \tag{B.16}$$

$$= 2\mathrm{tr}\left(\boldsymbol{F}\tilde{\boldsymbol{G}}^{-1}\boldsymbol{F}^{\top}\boldsymbol{d}\right) \tag{B.17}$$

$$= 2\left(\boldsymbol{F}\tilde{\boldsymbol{G}}^{-1}\boldsymbol{F}^{\top}\right)_{ij} d_{ji} \tag{B.18}$$

B.3 Auxiliary remark for the material Jacobian

Given $A_{ij} = A_{ji}$, $d_{ij} = d_{ji}$, find $\mathbb{C}_{ijkl} = \mathbb{C}_{klij} = \mathbb{C}_{ijlk}$ s.t. $A_{ij} = \mathbb{C}_{ijkl}d_{kl}$.

Remark: There are an infinity of solutions. Add any symmetric $\mathbb{C}$ with eigenvectors perpendicular to d.

One possible solution is

$$\mathbb{C}_{ijkl} = a_1(A_{ij}d_{kl} + d_{ij}A_{kl}) + a_2 d_{ij}d_{kl}. \tag{B.19}$$

Then,

$$\mathbb{C}_{ijkl}d_{kl} = a_1 A_{ij}|\boldsymbol{d}|^2 + d_{ij}(a_1 A_{kl}d_{kl} + a_2|\boldsymbol{d}|^2). \tag{B.20}$$

Pick: $a_1 = \frac{1}{|\boldsymbol{d}|^2}$ and $a_2 = -a_1\frac{A_{kl}d_{kl}}{|\boldsymbol{d}|^2} = -\frac{A_{kl}d_{kl}}{|\boldsymbol{d}|^4}$ so that we have

$$\mathbb{C}_{ijkl}d_{kl} = A_{ij}. \tag{B.21}$$

This means that

$$\mathbb{C}_{ijkl} = \frac{1}{|\boldsymbol{d}|^2}\left(A_{ij}d_{kl} + d_{ij}A_{kl}\right) - \frac{1}{|\boldsymbol{d}|^4}A_{mn}d_{mn}d_{ij}d_{kl}. \tag{B.22}$$

Appendix C

SYNTHESIS AND TESTING

C.1 Chemical details

Item	Density (g/mL)	Molar weight (g/mol)	Molecular formula
RM257	N/A	588.6	$C_{33}H_{32}O_{10}$
Toluene	0.865	92.14	C_7H_8
HHMP	N/A	224.25	$C_{12}H_{16}O_4$
PETMP	1.28	488.66	$C_{17}H_{28}O_8S_4$
EDDET	1.12	182.3	$C_6H_{14}O_2S_2$
DPA	0.738	101.19	$C_6H_{15}N$

C.2 Synthesis template

Synthesis

Name: _____

Date: _____

Time start: _____

Time finish: _____

Sample type: _____

Molds used: _____

mol% PETMP: _____

To do:

- ☐ Wear PPE
  - ☐ Lab coat
  - ☐ Safety glasses
  - ☐ Gloves
  - ☐ Closed-toe shoes
  - ☐ Tie long hair back
- ☐ Clean mold with IPA, DI water, Kim wipe
- ☐ Prepare 2 vials, labeled as “LCE solution" and “DPA+toluene"
- ☐ Prepare any glass pipettes

Hot plate dial level: _____

Hot plate temperature: _____

Note: This is the triple batch recipe (a larger batch attains better accuracy than the single batch recipe from [64]).

Item	**Expected weight (g)**	**Micropipette volume (μL)**	**Actual weight, subtotal (g)**	**Leftover on weigh boat or paper (g)**	**Actual weight, final (g)**	**% error**
RM257	12 g	—				
Toluene	4.8g	5549.1μL				
HHMP	0.0257g	—				
Time on the hot plate:						
Make catalyst solution (DPA+toluene) in separate vial:						
DPA	0.0549 g	74.25 μ L				
Toluene	0.0549 g	74.25 μ L				
Mix catalyst solution on vortex mixer						
PETMP (50 mol%)	2.1657 g	1691.64 μ L				
EDDET	1.6158g	1442.67μL				
Catalyst solution	1.7043g	1974μL				
Time on vortex mixer: 20 sec						
Time in vacuum oven at 20in Hg at room temperature: 45 sec						

Did sample recrystallize and need a 2nd time on hot plate? _____

Solution appearance (cloudy, clear, undissolved solids?): _____

Picture of sample in mold:

To do:

- ☐ Dispose of all hazardous waste inside plastic bag
- ☐ Clean beakers
- ☐ Clean utensils
- ☐ Turn off scale, hot plate, lights

Notes:

Placing samples into vacuum oven

Date: _____

Time into oven: _____

To do:

- □ Pull vacuum level to 20.5 in Hg
- □ Set temperature dial to 1.5

Notes:

Taking samples out of vacuum oven

Date: _____

Time out of oven: _____

Time of transferring samples from mold to sample boxes (should be ≈ 30 min later):

_____

Notes:

UV Crosslinking

Date: _____

Time: _____

UV crosslinking equipment used: _____

Amount of time hit by UV light per side: _____

Pictures:

Notes:

C.3 Tensile test template

Name: _____

Date of test:_____

Time of test: _____

Test type (load only, lo ad/unload, pull til failure): _____

Temperature: _____

Strain rate: _____

Velocity: _____

Linear stage extension: _____

Total time of test: _____

Sample details:

Sample type (polydomain/monodomain): _____

Sample identifier: _____

Date of synthesis: _____

Amount of time after taken out of oven: _____

Dimensions of sample:

Width: _____

Thickness: _____

Gauge length: _____

Notes on heating/cooling history: _____

Notes on loading history: _____

Microscope/video?: _____

How did the sample break: _____

Pictures and data (stress-strain curve, temperature data, images of sample before/after test, etc.):

Notes:

Appendix D

CODE

D.1 UMAT code for simulations of nematic elastomers in ABAQUS

This section contains the FORTRAN code for the user material defined by Chapter 4, implemented as a UMAT in ABAQUS. Some portions of code are reproduced from [15, 44, 60], explicitly denoted within comments in the code.

```
1       subroutine umat(stress,statev,ddsdde,sse,spd,scd,
2      + rpl,ddsddt,drplde,drpldt,
3      + stran,dstran,time,dtime,temp,dtemp,predef,dpred,cmname,
4      + ndi,nshr,ntens,nstatv,props,nprops,coords,drot,pnewdt,
5      + celent,dfgrd0,dfgrd1,noel,npt,layer,kspt,kstep,kinc)
6       !
7       include 'aba_param.inc'
8       !
9       character*80 cmname
10      dimension stress(ntens),statev(nstatv),
11     + ddsdde(ntens,ntens),ddsddt(ntens),drplde(ntens),
12     + stran(ntens),dstran(ntens),time(2),predef(1),dpred(1),
13     + props(nprops),coords(3),drot(3,3),dfgrd0(3,3),dfgrd1(3,3)
14      !
15      ! variables defined and used in the umat:
16      !
17      external myfunc, myconstraint1, myconstraint2
18      integer i,j,k,l,mm,nn,aa,bb,cc, nrot,np
19      real*8 iden(3,3), material_jacobian(3,3,3,3), stress_3x3(3,3), dPdF(3,3,3,3)
20      real*8 F_3x3(3,3), detF, trb, b_3x3(3,3), Finv(3,3), C_3x3(3,3), detC
21      real*8 mu1, mu2, p_prop, r_prop, c_prop, k_prop, alpha_delta, alpha_lambda,
    kappa, lambda, delta, new_lambda, new_delta
22      real*8 G_diag(3,3), G_3x3(3,3), Ginv(3,3), Finvtranspose(3,3),trGinvb,
    PKstress(3,3)
23      real*8 eigvalC(3),eigvec_mat(3,3),tol_val,alpha_const
24      real*8 eigvalb(3),eigvec_mat_b(3,3),dstran_3x3(3,3),prev_F(3,3)
25      real*8 prevFinv(3,3),l_3x3(3,3),d_3x3(3,3),trd
26      real*8 Gtilde_3x3(3,3),Gtildeinv(3,3),trGtildeinvC,testscalar, dnorm,
    Gtildeinvdot(3,3), prev_Gtildeinv(3,3)
27      real*8 A_3x3(3,3),AA_3x3(3,3),trA,trAA,trAAd,Gtildedot(3,3),Fdot(3,3),
    prev_Gtilde(3,3)
28      real*8 Lambda_3x3(3,3), U_3x3(3,3), U_3x3_inv(3,3), R_3x3(3,3),F_3x3_check
    (3,3),Gdiaginv(3,3),helper(3,3),oldA_3x3(3,3)
29      real*8 beta, w_3x3(3,3), temp_mat(6)
30      real*8 zero,one,two,three,four,five, six, nine,third,half
31      parameter(zero=0.d0,one=1.d0,two=2.d0,three=3.d0,four=4.d0,five=5.d0,six =6.
    d0,nine=9.d0,third=1.d0/3.d0,half=1.d0/2.d0)
32      real*8 x(2), minf, lb(2), ub(2), c1data, c2data, fdata(22)
```

```
33      real*8 ps(3), anps(3,3)
34      integer*8 opt
35      integer ires ! ires is an integer return value which is positive on success
      and negative on failure
36      include 'nlopt.f'
37      !
38      mu1 = props(1)
39      mu2 = props(2)
40      kappa = props(3)
41      r_prop = props(4)
42      c_prop = props(5)
43      k_prop = props(6)
44      alpha_delta = props(7)
45      alpha_lambda = alpha_delta/100.d0
46      p_prop = props(8)
47      !
48      tol_val = 5.d-5
49      alpha_const = 5.d-6
50      beta = 1.d6
51      !
52      F_3x3 = dfgrd1
53      statev(3) = F_3x3(1,1)
54      statev(4) = F_3x3(1,2)
55      statev(5) = F_3x3(1,3)
56      statev(6) = F_3x3(2,1)
57      statev(7) = F_3x3(2,2)
58      statev(8) = F_3x3(2,3)
59      statev(9) = F_3x3(3,1)
60      statev(10) = F_3x3(3,2)
61      statev(11) = F_3x3(3,3)
62      ! identity matrix
63      call onem(iden)
64      !
65      ! Compute the relative volume change
66      call mdet(F_3x3,detF)
67      !
68      ! Compute the inverse of the deformation gradient
69      call m3inv(F_3x3,Finv)
70      !
71      Finvtranspose = transpose(Finv)
72      !
73      ! compute the left & right Cauchy-Green tensor
74      b_3x3 = matmul(F_3x3, transpose(F_3x3)) ! matrix multiplication
75      C_3x3 = matmul(transpose(F_3x3),F_3x3) ! matrix multiplication
76      call mdet(C_3x3,detC)
77      !
78      ! get the eigenvalues and eigenvectors of C_3x3 (in descending order)
79      !write(*,*) 'in main code'
80      call zerom(temp_mat)
81      temp_mat(1) = C_3x3(1,1)
82      temp_mat(2) = C_3x3(2,2)
83      temp_mat(3) = C_3x3(3,3)
84      temp_mat(4) = C_3x3(1,2)
85      temp_mat(5) = C_3x3(1,3)
86      temp_mat(6) = C_3x3(2,3)
```

```
87        np = 3
88        call sprind(temp_mat,eigvalC,eigvec_mat,1,ndi,nshr)
89        eigvec_mat = transpose(eigvec_mat)
90        call eigsrt(eigvalC, eigvec_mat, 3, np)
91        call m3inv(dfgrd0,prevFinv)
92        l_3x3 = one/dtime*(matmul(F_3x3,prevFinv)-iden)
93        d_3x3 = half*(l_3x3+transpose(l_3x3))
94        w_3x3 = half*(l_3x3-transpose(l_3x3))
95        call tracem(d_3x3,trd)
96        call tracem(matmul(d_3x3,transpose(d_3x3)),dnorm)
97        !
98        !
99        ! at the start of an abaqus calculation, the state
100       !  variables are passed into umat with zero values.
101       !  initialize the state variables. at this point,
102       !  the time total_time and step_time both have a value
103       !  equal to zero and the step counter, kstep, is
104       !  equal to 1.
105       if ((time(1).eq.zero).and.(kstep.eq.1)) then
106         lambda = one
107         statev(1) = lambda
108         delta = one
109         statev(2) = delta
110         stress_3x3 = zero
111         !
112       end if
113       !
114       !
115       !
116       !
117       ! store the values of the state variables at the beginning of the time step
118       lambda = statev(1) ! at the previous time step
119       delta = statev(2) ! at the previous time step
120       !
121       !! UPDATE LAMBDA AND DELTA USING NLopt
122       opt = 0.d0
123       !call nlo_create(opt, NLOPT_LN_COBYLA, 2)
124       call nlo_create(opt, NLOPT_LD_MMA, 2) ! 2nd argument is the name of the
      optimization algorithm
125       lb(1) = 1.d0
126       lb(2) = 1.d0
127       ub(1) = r_prop**(1.d0/3.d0)
128       ub(2) = r_prop**(1.d0/6.d0)
129       call nlo_set_lower_bounds(ires, opt, lb)
130       call nlo_set_upper_bounds(ires, opt, ub)
131       fdata(1) = mu1
132       fdata(2) = mu2
133       fdata(3) = detF
134       fdata(4) = kappa
135       fdata(5) = c_prop
136       fdata(6) = r_prop
137       fdata(7) = k_prop
138       fdata(8) = dtime
139       fdata(9) = alpha_lambda
140       fdata(10) = alpha_delta
```

```
141       fdata(11) = lambda
142       fdata(12) = delta
143       fdata(13) = p_prop
144       fdata(14) = F_3x3(1,1)
145       fdata(15) = F_3x3(1,2)
146       fdata(16) = F_3x3(1,3)
147       fdata(17) = F_3x3(2,1)
148       fdata(18) = F_3x3(2,2)
149       fdata(19) = F_3x3(2,3)
150       fdata(20) = F_3x3(3,1)
151       fdata(21) = F_3x3(3,2)
152       fdata(22) = F_3x3(3,3)
153       call nlo_set_min_objective(ires, opt, myfunc, fdata)
154       call nlo_add_inequality_constraint(ires, opt, myconstraint1, c1data, 1.d-8)
155       call nlo_add_inequality_constraint(ires, opt, myconstraint2, c2data, 1.d-8)
156       call nlo_set_xtol_rel(ires, opt, 1.d-8)
157       ! initialize w/ previous values
158       x(1) = lambda
159       x(2) = delta
160       !
161       call nlo_optimize(ires, opt, x, minf)
162       !if (ires.lt.0) then
163       !   write(*,*) 'kstep =', kstep, 'increment num = ', kinc, 'total time = ',
      time(2)
164       !   write(*,*) 'nlopt failed! ires = ', ires
165       !else
166       !   write(*,*) 'kstep =', kstep, 'increment num = ', kinc, 'total time = ',
      time(2)
167       !   write(*,*) x(1), x(2), F_3x3(1,1)
168       !   write(*,*) 'min val = ', minf, 'and ires =', ires
169       !endif
170       call nlo_destroy(opt)
171       ! end of the NLopt scheme
172       !
173       new_lambda = x(1)
174       new_delta = x(2)
175       !
176       statev(1) = new_lambda
177       statev(2) = new_delta
178       !
179       ! calculate G_diag
180       call zerom(G_diag)
181       G_diag(1,1) = new_lambda**two ! double asterisk ** is for raising to a power
182       G_diag(2,2) = new_delta**two/(new_lambda**two)
183       G_diag(3,3) = one/(new_delta**two)
184       call m3inv(G_diag,Gdiaginv)
185       !
186       Gtilde_3x3 = matmul(eigvec_mat,matmul(G_diag,transpose(eigvec_mat)))
187       call m3inv(Gtilde_3x3,Gtildeinv)
188       A_3x3 = matmul(F_3x3,matmul(Gtildeinv,transpose(F_3x3)))
189       call tracem(A_3x3, trA)
190       !
191       ! 3x3 cauchy stress
192       stress_3x3 = (mu1*detF**(-two/three)*(A_3x3-(one/three)*trA*iden)
193      +           + kappa*dlog(detF)*iden)/detF + beta*d_3x3
```

```
194       ! transform 3x3 stress into voigt notation
195       stress(1) = stress_3x3(1,1)
196       stress(2) = stress_3x3(2,2)
197       stress(3) = stress_3x3(3,3)
198       stress(4) = stress_3x3(1,2)
199       stress(5) = stress_3x3(1,3)
200       stress(6) = stress_3x3(2,3)
201       !
202       !
203       !
204       !
205       ! calculate the material jacobian
206       !
207       material_jacobian = zero
208       if (dnorm.lt.tol_val) then
209       do i=1,3
210        do j=1,3
211         do k=1,3
212          do l=1,3
213            material_jacobian(i,j,k,l) = material_jacobian(i,j,k,l)
214      +     - (two/three)*mu1*detF**(-five/three)*(A_3x3(i,j)*iden(k,l)+A_3x3(k,l)*
         iden(i,j))
215      +     + (two/nine)*mu1*detF**(-five/three)*trA*iden(i,j)*iden(k,l)
216      +     + kappa/detF*iden(i,j)*iden(k,l)
217      +     + mu1/two*detF**(-five/three)*(A_3x3(j,l)*iden(i,k)+A_3x3(i,k)*iden(j,l
         ))
218      +     + mu1/two*detF**(-five/three)*(A_3x3(i,l)*iden(j,k)+A_3x3(j,k)*iden(i,l
         ))
219      +     + beta/two*trd*(iden(i,k)*iden(j,l)+iden(j,k)*iden(i,l))
220      +     + beta/two*(iden(i,k)*w_3x3(l,j)+iden(i,l)*w_3x3(k,j)+w_3x3(k,i)*iden(l
         ,j)+w_3x3(l,i)*iden(j,k))
221      +     + beta/(two*dtime)*(iden(i,k)*iden(j,l)+iden(j,k)*iden(i,l))
222           enddo
223          enddo
224         enddo
225        enddo
226        else
227        do i=1,3
228         do j=1,3
229          do k=1,3
230           do l=1,3
231            material_jacobian(i,j,k,l) = material_jacobian(i,j,k,l)
232      +     - (two/three)*mu1*detF**(-five/three)*(A_3x3(i,j)*iden(k,l)+A_3x3(k,l)*
         iden(i,j))
233      +     + (two/nine)*mu1*detF**(-five/three)*trA*iden(i,j)*iden(k,l)
234      +     + kappa/detF*iden(i,j)*iden(k,l)
235      +     + mu1/two*detF**(-five/three)*(A_3x3(j,l)*iden(i,k)+A_3x3(i,k)*iden(j,l
         ))
236      +     + mu1/two*detF**(-five/three)*(A_3x3(i,l)*iden(j,k)+A_3x3(j,k)*iden(i,l
         ))
237      +     -two*mu1*alpha_const*detF**(-five/three)*(
238      +     one/(dnorm**two)*(d_3x3(i,j)*d_3x3(k,l)-third*trd*(iden(i,j)*d_3x3(k,l)
         +d_3x3(i,j)*iden(k,l)))
239      +     +one/(dnorm**four)*third*(trd**two)*d_3x3(i,j)*d_3x3(k,l))
240      +     + beta/two*trd*(iden(i,k)*iden(j,l)+iden(j,k)*iden(i,l))
```

```
241      +     + beta/two*(iden(i,k)*w_3x3(l,j)+iden(i,l)*w_3x3(k,j)+w_3x3(k,i)*iden(l
         ,j)+w_3x3(l,i)*iden(j,k))
242      +     + beta/(two*dtime)*(iden(i,k)*iden(j,l)+iden(j,k)*iden(i,l))
243           enddo
244         enddo
245        enddo
246       enddo
247       end if
248       !*********************************************************************
249       ! transform the tangent matrix into voigt notation. code from
250       ! Alexander K. Landauer, Xiuqi Li, Christian Franck, and David L. Henann.
251       ! Experimental characterization and hyperelastic constitutive modeling of
252       ! open-cell elastomeric foams. Journal of the Mechanics and Physics of
         Solids
253       ! doi:10.1016/j.jmps.2019.103701.
254       !*********************************************************************
255       ddsdde = zero
256       !
257       do i=1,ndi
258        do j=1,ndi
259           ddsdde(i,j) = material_jacobian(i,i,j,j)
260        enddo
261       enddo
262       !
263       if (nshr.ne.0) then
264       do i=1,ndi
265           ddsdde(i,ndi+1) = material_jacobian(i,i,1,2)
266           ddsdde(ndi+1,i) = material_jacobian(1,2,i,i)
267       enddo
268       ddsdde(ndi+1,ndi+1) = material_jacobian(1,2,1,2)
269       if (nshr.ne.1) then
270        do i=1,ndi
271        ddsdde(i,ndi+2) = material_jacobian(i,i,1,3)
272        ddsdde(ndi+2,i) = material_jacobian(1,3,i,i)
273        enddo
274        ddsdde(ndi+2,ndi+2) = material_jacobian(1,3,1,3)
275        ddsdde(ndi+1,ndi+2) = material_jacobian(1,2,1,3)
276        ddsdde(ndi+2,ndi+1) = material_jacobian(1,3,1,2)
277        if (nshr.ne.2) then
278         do i=1,ndi
279             ddsdde(i,ndi+3) = material_jacobian(i,i,2,3)
280             ddsdde(ndi+3,i) = material_jacobian(2,3,i,i)
281         enddo
282         ddsdde(ndi+3,ndi+3) = material_jacobian(2,3,2,3)
283         ddsdde(ndi+1,ndi+3) = material_jacobian(1,2,2,3)
284         ddsdde(ndi+3,ndi+1) = material_jacobian(2,3,1,2)
285         ddsdde(ndi+2,ndi+3) = material_jacobian(1,3,2,3)
286         ddsdde(ndi+3,ndi+2) = material_jacobian(2,3,1,3)
287         end if
288        end if
289       end if
290       return
291       end subroutine umat
292       !
293       !*************************************************************
```

```
294       !
295       subroutine myfunc(func_val, n, x, grad, need_gradient, fdata)
296       double precision x(n), grad(n), fdata(22)
297       real*8 func_val
298       integer i,j, n, need_gradient,nrot,np, ndi, nshr
299       real*8 mu1, mu2, detF, kappa, c_prop, r_prop, k_prop, dtime, alpha_lambda,
      alpha_delta, lambda, delta, p_prop
300       real*8 F_3x3(3,3), temp1(3,3), temp2(3,3), trtemp1, trtemp2
301       real*8 C_3x3(3,3), eigvalC(3), eigvec_mat(3,3),G_diag(3,3),Gtilde_3x3(3,3),
      Gtildeinv(3,3),A_3x3(3,3),trA
302       real*8 dGtildedlambda(3,3), dGtildeddelta(3,3),term1(3,3),trterm1,term2(3,3)
      ,trterm2,dDdlambda,dDddelta,dfddelta
303       real*8 W_val, f_val,diss_val, temp_mat(6)
304       real*8 zero,one,two,three,four,five, six, nine,third,half
305       parameter(zero=0.d0,one=1.d0,two=2.d0,three=3.d0,four=4.d0,five=5.d0,six =6.
      d0,nine=9.d0,third=1.d0/3.d0,half=1.d0/2.d0)
306       mu1 = fdata(1)
307       mu2 = fdata(2)
308       detF = fdata(3)
309       kappa = fdata(4)
310       c_prop = fdata(5)
311       r_prop = fdata(6)
312       k_prop = fdata(7)
313       dtime = fdata(8)
314       alpha_lambda = fdata(9)
315       alpha_delta = fdata(10)
316       lambda = fdata(11)
317       delta = fdata(12)
318       p_prop = fdata(13)
319       F_3x3(1,1) = fdata(14)
320       F_3x3(1,2) = fdata(15)
321       F_3x3(1,3) = fdata(16)
322       F_3x3(2,1) = fdata(17)
323       F_3x3(2,2) = fdata(18)
324       F_3x3(2,3) = fdata(19)
325       F_3x3(3,1) = fdata(20)
326       F_3x3(3,2) = fdata(21)
327       F_3x3(3,3) = fdata(22)
328       !
329       !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! func_val
      !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
330       np = 3
331       ndi = 3
332       nshr = 3
333       C_3x3 = matmul(transpose(F_3x3),F_3x3)
334       call zerom(temp_mat)
335       temp_mat(1) = C_3x3(1,1)
336       temp_mat(2) = C_3x3(2,2)
337       temp_mat(3) = C_3x3(3,3)
338       temp_mat(4) = C_3x3(1,2)
339       temp_mat(5) = C_3x3(1,3)
340       temp_mat(6) = C_3x3(2,3)
341       call sprind(temp_mat,eigvalC,eigvec_mat,1,ndi,nshr)
342       eigvec_mat = transpose(eigvec_mat)
343       call eigsrt(eigvalC, eigvec_mat, 3, np)
```

```
344       call zerom(G_diag)
345       G_diag(1,1) = x(1)**two  ! double asterisk ** is for raising to a power
346       G_diag(2,2) = x(2)**two/(x(1)**two)
347       G_diag(3,3) = one/(x(2)**two)
348       Gtilde_3x3 = matmul(eigvec_mat,matmul(G_diag,transpose(eigvec_mat)))
349       call m3inv(Gtilde_3x3,Gtildeinv)
350       A_3x3 = matmul(F_3x3,matmul(Gtildeinv,transpose(F_3x3)))
351       call tracem(A_3x3, trA)
352       W_val = (mu1/two)*(detF**(-two/three)*trA-three) + (kappa/two)*(dlog(detF))
      **two
353       f_val = c_prop*(x(2)-one)/((r_prop**(one/six)-x(2))**k_prop)
354       diss_val = (alpha_lambda/two)*((x(1)-lambda)/dtime)**p_prop + (alpha_delta/
      two)*((x(2)-delta)/dtime)**p_prop
355       !
356       func_val = W_val + f_val + dtime*diss_val
357       !
358       !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! gradient
      !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
359       !
360       call zerom(dGtildedlambda)
361       dGtildedlambda(1,1) = two*x(1)
362       dGtildedlambda(2,2) = -two*x(2)**two/(x(1)**three)
363       dGtildedlambda = matmul(eigvec_mat,matmul(dGtildedlambda,transpose(
      eigvec_mat)))
364       call zerom(dGtildeddelta)
365       dGtildeddelta(2,2) = two*x(2)/(x(1)**two)
366       dGtildeddelta(3,3) = -two/(x(2)**three)
367       dGtildeddelta = matmul(eigvec_mat,matmul(dGtildeddelta,transpose(eigvec_mat)
      ))
368       !
369       term1 = matmul(Gtildeinv,matmul(dGtildedlambda,matmul(Gtildeinv,C_3x3)))
370       call tracem(term1,trterm1)
371       term2 = matmul(Gtildeinv,matmul(dGtildeddelta,matmul(Gtildeinv,C_3x3)))
372       call tracem(term2,trterm2)
373       !
374       dWdlambda = -(mu1/two)*detF**(-two/three)*trterm1
375       dWddelta = -(mu1/two)*detF**(-two/three)*trterm2
376       !
377       dDdlambda = (alpha_lambda/two)*(p_prop/dtime)*((x(1)-lambda)/dtime)**(p_prop
      -one)
378       dDddelta = (alpha_delta/two)*(p_prop/dtime)*((x(2)-delta)/dtime)**(p_prop-
      one)
379       !
380       dfddelta = c_prop*((r_prop**(one/six)-x(2)+(x(2)-one)*k_prop)/((r_prop**(one
      /six)-x(2))**(k_prop+one)))
381       !
382       if (need_gradient.ne.0) then
383          grad(1) = dWdlambda + dtime*dDdlambda
384          grad(2) = dWddelta + dfddelta + dtime*dDddelta
385       endif
386       end
387       !
388       !************************************************************
389       !
390       subroutine myconstraint1(func_val, n, x, grad, need_gradient, d)
```

```
391       integer need_gradient
392       real*8 func_val, x(n), grad(n)
393       if (need_gradient.ne.0) then
394          grad(1) = 0.5d0/dsqrt(x(1))
395          grad(2) = -1.d0
396       endif
397       func_val = dsqrt(x(1)) - x(2)
398       end
399       !
400       !*************************************************************
401       !
402       subroutine myconstraint2(func_val, n, x, grad, need_gradient, d)
403       integer need_gradient
404       real*8 func_val, x(n), grad(n)
405       if (need_gradient.ne.0) then
406          grad(1) = -2.d0*x(1)
407          grad(2) = 1.d0
408       endif
409       func_val = x(2) - x(1)**2.d0
410       end
411 !*****************************************************************************
412 !     utility subroutines
413 !*****************************************************************************
414 !   from shawn chester, https://web.njit.edu/~sac3/Software.html
415 !*****************************************************************************
416       subroutine onem(a)
417       !
418       ! this subroutine initializes the 3x3 matrix a as the identity matrix
419       !
420       implicit none
421       !
422       integer i,j
423       !
424       real*8 a(3,3)
425       !
426       do i=1,3
427          do j=1,3
428             if (i .eq. j) then
429               a(i,j) = 1.d0
430             else
431               a(i,j) = 0.d0
432             end if
433          end do
434       end do
435       !
436       return
437       end subroutine onem
438
439 !*********************************************************************
440 !   this subroutine sets all entries of a 3 by 3 matrix to 0.d0.
441 !   from shawn chester, https://web.njit.edu/~sac3/Software.html
442 !*********************************************************************
443       subroutine zerom(a)
444       real*8 a(3,3)
445
```

```
446       do 1 i=1,3
447         do 1 j=1,3
448           a(i,j) = 0.d0
449 1     continue
450 !
451       return
452       end
453
454 !*****************************************************************
455 !   this subroutine calculates the trace of a 3 by 3 matrix [a]
456 !   and stores the result in the scalar tra
457 !   from shawn chester, https://web.njit.edu/~sac3/Software.html
458 !*****************************************************************
459       subroutine tracem(a,tra)
460
461       real*8 a(3,3),tra
462
463       tra = a(1,1) + a(2,2) + a(3,3)
464
465       return
466       end
467 !*****************************************************************
468 !   this subroutine calculates the determinant
469 !   of a 3 by 3 matrix [a].
470 !   from shawn chester, https://web.njit.edu/~sac3/Software.html
471 !*****************************************************************
472       subroutine mdet(a,det)
473
474       real*8  a(3,3), det
475
476       det =   a(1,1)*a(2,2)*a(3,3)
477      +          + a(1,2)*a(2,3)*a(3,1)
478      +          + a(1,3)*a(2,1)*a(3,2)
479      +      - a(3,1)*a(2,2)*a(1,3)
480      +      - a(3,2)*a(2,3)*a(1,1)
481      +      - a(3,3)*a(2,1)*a(1,2)
482
483       return
484       end
485 !*****************************************************************
486 !   this subroutine calculates the transpose of a 3 by 3
487 !   matrix [a], and places the result in atrans.
488 !   from shawn chester, https://web.njit.edu/~sac3/Software.html
489 !*****************************************************************
490       subroutine mtrans(a,atrans)
491
492       real*8 a(3,3),atrans(3,3)
493
494       do 1 i=1,3
495         do 1 j=1,3
496           atrans(j,i) = a(i,j)
497 1     continue
498
499       return
500       end
```

```
!*****************************************************************
!   this subroutine calculates the the inverse of a 3 by 3 matrix
!   [a] and places the result in [ainv].
!   if det(a) is zero, the calculation
!   is terminated and a diagnostic statement is printed.
!*****************************************************************
!   a(3,3)      -- the matrix whose inverse is desired.
!   det         -- the computed determinant of [a].
!   acofac(3,3) -- the matrix of cofactors of a(i,j).
!              the signed minor (-1)**(i+j)*m_ij
!              is called the cofactor of a(i,j).
!   aadj(3,3)   -- the adjoint of [a]. it is the matrix
!              obtained by replacing each element of
!              [a] by its cofactor, and then taking
!              transpose of the resulting matrix.
!   ainv(3,3)   -- returned as inverse of [a].
!              [ainv] = [aadj]/det.
!-----------------------------------------------------------------
!   from shawn chester, https://web.njit.edu/~sac3/Software.html
      subroutine m3inv(a,ainv)
      real*8  a(3,3), ainv(3,3), det, acofac(3,3), aadj(3,3)
      call mdet(a,det)
      if ( det .eq. 0.d0 ) then
        write(*,10)
        stop
      endif
      call mcofac(a,acofac)
      call mtrans(acofac,aadj)
      do 1 i = 1,3
      do 1 j = 1,3
           ainv(i,j) = aadj(i,j)/det
1     continue
10    format(5x,'--error in m3inv--- the matrix is singular',/,
     +         10x,'program terminated')

      return
      end
!*****************************************************************
!   this subroutine calculates the cofactor of a 3 by 3 matrix [a],
!   and places the result in [acofac].
!   from shawn chester, https://web.njit.edu/~sac3/Software.html
!*****************************************************************
      subroutine mcofac(a,acofac)

      real*8  a(3,3), acofac(3,3)

      acofac(1,1) = a(2,2)*a(3,3) - a(3,2)*a(2,3)
      acofac(1,2) = -(a(2,1)*a(3,3) - a(3,1)*a(2,3))
      acofac(1,3) = a(2,1)*a(3,2) - a(3,1)*a(2,2)
      acofac(2,1) = -(a(1,2)*a(3,3) - a(3,2)*a(1,3))
      acofac(2,2) = a(1,1)*a(3,3) - a(3,1)*a(1,3)
      acofac(2,3) = -(a(1,1)*a(3,2) - a(3,1)*a(1,2))
      acofac(3,1) = a(1,2)*a(2,3)  - a(2,2)*a(1,3)
      acofac(3,2) = -(a(1,1)*a(2,3) - a(2,1)*a(1,3))
      acofac(3,3) = a(1,1)*a(2,2) - a(2,1)*a(1,2)
```

```
556
557       return
558       end
559 !*********************************************************************
560       subroutine Jacobi(a,n,np,d,v,nrot)
561       !============================================================
562       ! Computes all eigenvalues and eigenvectors of a real symmetric matrix a,
      which is of size n
563       ! by n, stored in a physical np by np array. On output, elements of a above
      the diagonal are
564       ! destroyed. d returns the eigenvalues of a in its first n elements. v is a
      matrix with the same
565       ! logical and physical dimensions as a, whose columns contain, on output,
      the normalized
566       ! eigenvectors of a. nrot returns the number of Jacobi rotations that were
      required.
567       !------------------------------------------------------------
568       ! input ...
569       ! a(n,n): real symmetric matrix
570       ! np:
571       ! d(n): eigenvalues of a
572       ! v(n,n): columns contain normalized eigenvectors of a
573       ! nrot: # jacobi rotations that were required
574       !
575       ! Press, William H, Saul A Teukolsky, William T Vetterling, and Brian P
      Flannery. Numerical
576       ! Recipes in F77. Numerical Methods for Fortran 77: The Art of Scientific
      Computing, 1992.
577       !============================================================
578       implicit none  ! there's no "implicit none" in the numerical recipes
579       INTEGER n,np,nrot,NMAX
580       REAL*8 a(np,np),d(np),v(np,np)
581       PARAMETER (NMAX=500)
582       INTEGER i,ip,iq,j
583       REAL c,g,h,s,sm,t,tau,theta,tresh,b(NMAX),z(NMAX)
584       real*8 zero,one,two,three,four,five, six, nine,third,half
585       parameter(zero=0.d0,one=1.d0,two=2.d0,three=3.d0,four=4.d0,five=5.d0,six =6.
      d0,nine=9.d0,third=1.d0/3.d0,half=1.d0/2.d0)
586
587       do ip=1,n ! initialize to the identity matrix
588          do iq = 1,n
589             v(ip,iq) = 0.d0
590          end do
591          v(ip,ip) = 1.d0
592       end do
593       !
594       do ip=1,n
595          b(ip)=a(ip,ip) !Initialize b and d to the diagonal of a.
596          d(ip)=b(ip)
597          z(ip)=zero
598       enddo
599       !
600       nrot=0
601       do i = 1,50
602        sm=0.d0
```

```
603       do ip=1,n-1 !Sum off-diagonal elements.
604          do iq=ip+1,n
605             sm=sm+abs(a(ip,iq))
606          enddo
607       enddo
608       if(sm.eq.0.d0)return     ! The normal return, which relies on quadratic
      conver
609       if(i.lt.4)then           ! gence to machine underflow.
610         tresh=0.2d0*sm/n**2.d0 !...on the first three sweeps.
611       else
612         tresh=0.d0             !...thereafter.
613       endif
614       !
615       do ip=1,n-1
616         do iq=ip+1,n
617            g=100.d0*abs(a(ip,iq)) !After four sweeps, skip the rotation if the
      off-diagonal element is small.
618           if((i.gt.4).and.(abs(d(ip))+g.eq.abs(d(ip))).and.(abs(d(iq))+g.eq.abs(
      d(iq))))then
619              a(ip,iq)=0.d0
620           else if(abs(a(ip,iq)).gt.tresh)then
621             h=d(iq)-d(ip)
622             if(abs(h)+g.eq.abs(h))then
623                t=a(ip,iq)/h     ! t = 1/(2theta)
624             else
625                theta=0.5d0*h/a(ip,iq) !Equation (11.1.10).
626                t=1.d0/(abs(theta)+sqrt(1.d0+theta**2.d0))
627                if(theta.lt.0.d0)t=-t
628             endif
629             c=1.d0/sqrt(1.d0+t**2.d0)
630             s=t*c
631             tau=s/(1.d0+c)
632             h=t*a(ip,iq)
633             z(ip)=z(ip)-h
634             z(iq)=z(iq)+h
635             d(ip)=d(ip)-h
636             d(iq)=d(iq)+h
637             a(ip,iq)=0.d0
638             do j=1,ip-1 !Case of rotations 1 \leq j<p.
639                g=a(j,ip)
640                h=a(j,iq)
641                a(j,ip)=g-s*(h+g*tau)
642                a(j,iq)=h+s*(g-h*tau)
643             enddo
644             do j=ip+1,iq-1 !Case of rotations p<j<q.
645                g=a(ip,j)
646                h=a(j,iq)
647                a(ip,j)=g-s*(h+g*tau)
648                a(j,iq)=h+s*(g-h*tau)
649             enddo
650             do j=iq+1,n !Case of rotations q<j \leq n.
651                g=a(ip,j)
652                h=a(iq,j)
653                a(ip,j)=g-s*(h+g*tau)
654                a(iq,j)=h+s*(g-h*tau)
```

```
655               enddo
656               do j=1,n
657                  g=v(j,ip)
658                  h=v(j,iq)
659                  v(j,ip)=g-s*(h+g*tau)
660                  v(j,iq)=h+s*(g-h*tau)
661               enddo
662               nrot=nrot+1
663             endif
664           enddo
665         enddo
666         !
667         do ip=1,n
668            b(ip)=b(ip)+z(ip)
669            d(ip)=b(ip)  ! Update d with the sum of tapq,
670            z(ip)=0.d0   ! and reinitialize z.
671         enddo
672       enddo
673       pause 'too many iterations in jacobi'
674       return
675       end
676 !**********************************************************************
677       subroutine eigsrt(d,v,n,np)
678       ! Given the eigenvalues d and eigenvectors v as output from jacobi (11.1) or
       tqli (11.3),
679       ! this routine sorts the eigenvalues into descending order, and rearranges
      the columns of v correspondingly.
680       ! The method is straight insertion.
681       !
682       ! Press, William H, Saul A Teukolsky, William T Vetterling, and Brian P
      Flannery. Numerical
683       ! Recipes in F77. Numerical Methods for Fortran 77: The Art of Scientific
      Computing, 1992.
684       implicit none
685       INTEGER n,np
686       REAL*8 d(np),v(np,np)
687       INTEGER i,j,k
688       REAL p
689       do  i=1,n-1
690         k=i
691         p=d(i)
692         do j=i+1,n
693           if(d(j).ge.p)then
694             k=j
695             p=d(j)
696           endif
697         enddo
698         if(k.ne.i)then
699           d(k)=d(i)
700           d(i)=p
701           do j=1,n
702             p=v(j,i)
703             v(j,i)=v(j,k)
704             v(j,k)=p
705           enddo
```

```
706         endif
707       enddo
708       !
709       return
710       end
711
712 !*********************************************************************
713 ! this is a helper function. uncomment as needed to see eigenvalues, eigenvectors
714 !*********************************************************************
715       subroutine eighelp(A_3x3,eigvalA,eigvec_mat_A,n,np,nrot)
716       INTEGER n,np,nrot
717       REAL*8 A_3x3(np,np),eigvalA(np),eigvec_mat_A(np,np)
718       real*8 temp_mat(3,3),LHS(3),RHS(3),vec1(3),vec2(3),vec3(3)
719       real*8 tempeigvec(3)
720       INTEGER i,j,k
721
722       !write(*,*) 'original matrix'
723       !write(*,*) A_3x3(1,1), A_3x3(1,2), A_3x3(1,3)
724       !write(*,*) A_3x3(2,1), A_3x3(2,2), A_3x3(2,3)
725       !write(*,*) A_3x3(3,1), A_3x3(3,2), A_3x3(3,3)
726       !write (*,202) ! eigenvalues
727       !write(*,*) eigvalA(1), eigvalA(2), eigvalA(3)
728       !write (*,203) ! eigenvectors
729       !write (*,*) eigvec_mat_A(1,1), eigvec_mat_A(1,2), eigvec_mat_A(1,3)
730       !write (*,*) eigvec_mat_A(2,1), eigvec_mat_A(2,2), eigvec_mat_A(2,3)
731       !write (*,*) eigvec_mat_A(3,1), eigvec_mat_A(3,2), eigvec_mat_A(3,3)
732       ! rename the eigenvectors to vector vec1, vec2, and vec3
733       do j = 1,3
734           vec1(j) = eigvec_mat_A(j,1)
735           vec2(j) = eigvec_mat_A(j,2)
736           vec3(j) = eigvec_mat_A(j,3)
737       end do
738       !write (*,*) vec1(1), vec2(1), vec3(1)
739       !write (*,*) vec1(2), vec2(2), vec3(2)
740       !write (*,*) vec1(3), vec2(3), vec3(3)
741       call zerom(temp_mat)
742       temp_mat(1,1) = eigvalA(1)
743       temp_mat(2,2) = eigvalA(2)
744       temp_mat(3,3) = eigvalA(3)
745       temp_mat = A_3x3 - matmul(eigvec_mat_A,matmul(temp_mat,transpose(
      eigvec_mat_A)))
746       !write(*,*) 'original matrix minus reconstructed matrix'
747       !write(*,*) temp_mat(1,1), temp_mat(1,2), temp_mat(1,3)
748       !write(*,*) temp_mat(2,1), temp_mat(2,2), temp_mat(2,3)
749       !write(*,*) temp_mat(3,1), temp_mat(3,2), temp_mat(3,3)
750 201   format (6f12.6)
751 202   format (/,' Eigenvalues')
752 203   format (/,' Eigenvectors')
753       !
754       !write(*,*) 'lets check A_{ij}*v_j = eigval_i*v_i'
755       !write(*,*) 'eigvalA(1)'
756       LHS = 0.D0
757       do i=1,3
758           do j = 1,3
759           LHS(i) = LHS(i) + A_3x3(i,j)*vec1(j)
```

```
760           enddo
761           RHS(i) = eigvalA(1)*vec1(i)
762           tempeigvec(i) = LHS(i) - RHS(i)
763       end do
764       !write(*,*) 'LHS - RHS vector =', (tempeigvec(i),i=1,3)
765       !
766       !write(*,*) 'eigvalA(2)'
767       LHS = 0.D0
768       do i=1,3
769           do j = 1,3
770              LHS(i) = LHS(i) + A_3x3(i,j)*vec2(j)
771           end do
772           RHS(i) = eigvalA(2)*vec2(i)
773           tempeigvec(i) = LHS(i) - RHS(i)
774       end do
775       !write(*,*) 'LHS - RHS vector =', (tempeigvec(i),i=1,3)
776       !
777       !write(*,*) 'eigvalA(3)'
778       LHS = 0.D0
779       do i=1,3
780           do j = 1,3
781           LHS(i) = LHS(i) + A_3x3(i,j)*vec3(j)
782           end do
783           RHS(i) = eigvalA(3)*vec3(i)
784            tempeigvec(i) = LHS(i) - RHS(i)
785       end do
786       !write(*,*) 'LHS - RHS vector =', (tempeigvec(i),i=1,3)
787       !
788       return
789       end
```

D.2 MATLAB code for the thermo-mechanical characterization experiments

We include the code to run a uniaxial test using our custom setup, as discussed in Chapter 5. Note that the code is specific to the type of linear stage and controller, the load cell, data acquisition system, and camera. See Section 5.3 for the equipment list. We include the following code:

- `main.m`: the main code with instructions for setting up, executing, and cleaning up a test
- `test_parameters.m`: the function that defines test parameters
- `init_load.m`: the function that initiates the data acquisition
- `get_load.m`: the function that obtains data from the DAQ
- `PI_move.m`: the function that controls movement of the linear stage

The `main.m` code:

```
%% CONSTANTS

% imput sample geometry here
width = .280; % inches, sample width
thickness = .051; % inches, sample thickness
eps = 500; % [%] desired strain
epsdot = 4.5*10^-3; %[1/sec]

% global variables
global pos_data;
global temp_data;
global load_data;
global time_data;
global datetime_starttest;
startCoarsePos = 249;
jog_step = 1; %mm
jog_velocity = 4; % mm/sec

% plotting constants
linewidth=1.5;
fontsize=16;
fontsizemult=1.2;

%% Connect to controller and reference stage (without sample connected)
% the stage will move to reference position of 102mm (the middle of the stage)
% this is for a single axis that is connected via USB.
% if you are using a different controller & stage, please set up in
% MikroMove first, then edit accordingly in connect_and_reference_loaner().
if ~exist('PIdevice')
    [PIdevice, axis, Controller] = connect_and_reference_loaner();
end

disp('')
disp ('linear stage is done moving to reference position. position in mm =')
disp(PIdevice.qPOS(axis))


%% put sample into top clamp
    % Move to coarse start position and attach pullrod
    PIdevice.VEL(axis, jog_velocity);
    PIdevice.MOV(axis, startCoarsePos);
    while(abs(PIdevice.qPOS(axis) - startCoarsePos) > 0.001)
        continue
    end

%%     Jogging: move up/down to get specimen clamped in position
    PIdevice.VEL(axis, jog_velocity);
    u = 'u';
    d = 'd';
    while (1)
        dir = input('Move u or d? (hit enter when finished jogging.) ');
```

```
52          if (dir == 'u')
53              PIdevice.MOV(axis, PIdevice.qPOS(axis)-jog_step);
54          elseif (dir == 'd')
55              PIdevice.MOV(axis, PIdevice.qPOS(axis)+jog_step);
56          else
57              break
58          end
59      end
60
61      clear u d
62
63  %% calculate gauge length
64      currPos = PIdevice.qPOS(axis); % mm
65      gaugelength_theoretical_mm = 273.6978-currPos; %mm
66      gaugelength_theoretical_in = gaugelength_theoretical_mm*0.0393701;
67      % measure and input the gauge length!
68      gaugelength_measured_in = .375; % inches, sample gauge length
69      gaugelength_measured_mm = SI(gaugelength_measured_in, 'in')*10^3; %mm
70
71      abs(gaugelength_theoretical_in - gaugelength_measured_in)
72      gaugelength = gaugelength_theoretical_mm;
73      gaugelength_in = gaugelength_theoretical_in;
74
75   %% calculate test parameters
76      [extension, test_velocity, tmax, epsdot] = test_parameters(eps,
        gaugelength_theoretical_in, 'in',epsdot);
77      fprintf('epsdot = %.3e\n', epsdot)
78      fprintf('test_velocity = %.3e\n', test_velocity) % test_velocity in mm/sec
79       % extension:     mm, this is how much total distance you want stage to
80                      % move between startPos & stopPos
81                      % this is positive if you want stage to physically move up
82                      % (towards 0mm)
83      crosssecarea = width*thickness*(25.4)^2; %mm^2
84      sample_period = 1; % sec (must be >= 0.5)
85      step = test_velocity*sample_period; % mm
86
87      % Configure linear stage
88
89      startPos = PIdevice.qPOS(axis);
90      stopPos = startPos-extension;
91
92      if (startPos > stopPos)
93          steps = [startPos:-step:stopPos, (stopPos+step):step:startPos];
94      else
95          steps = [startPos:step:stopPos, (stopPos-step):-step:startPos];
96      end
97
98      max(steps)
99      min(steps)
100
101 %% Initialize inet + collect preheating data
102
103     if ~exist('inet')
104         inet = init_load(false, 'tori_RTD_louisaload_jun252020'); %.prf file
105     end
```

```
106
107      numb = 100000;
108      preheatload=NaN*zeros(size(numb));
109      preheattemp=NaN*zeros(size(numb));
110
111  for i = 1:numb
112      pause(2) % wait every 2 sec
113      [preheattemp(i), preheatload(i)] = get_load(inet);
114  end
115
116  % type ctrl+c when preheating is finished
117
118  %% save the preheat data
119  endnum = 410; % input the last index when you stop the preheating
120  preheattime = linspace(1,endnum*2,endnum);
121  figure;
122  subplot(2,1,1)
123  plot(preheattime,preheattemp(1:endnum))
124  xlabel('time (s)')
125  ylabel('temp (deg F)')
126  subplot(2,1,2)
127  plot(preheattime,preheatload)
128  xlabel('time (s)')
129  ylabel('load (lbs)')
130  %
131  save('preheat_data.mat','preheattime','preheatload','preheattemp')
132
133  %% MOVE
134      % Move routine
135
136      % extension data is saved in this variable pos_data
137      % even if you interrupt the script using ctrl+C, it will save the data in
138      % this variable, also saved in a .mat file
139
140      % create vector of zeros to fill to speed up data collection
141      pos_data = NaN*zeros(size(steps));
142      temp_data = NaN*zeros(size(steps));
143      load_data = NaN*zeros(size(steps));
144      time_data = NaN*zeros(size(steps));
145
146      PIdevice.VEL(axis, test_velocity);
147
148      move(startPos, stopPos, step, steps, PIdevice, axis, inet)
149
150  %%  calculate the actual test_velocity
151      actual_test_velocity = diff(pos_data)./diff(time_data);
152      output_data(1,1) = test_velocity;
153      output_data(1,2) = mean(actual_test_velocity(actual_test_velocity>0));
154      output_data(1,3) = std(actual_test_velocity(actual_test_velocity>0));
155
156      output_data(1,:) % print out these items in the command window
157
158  %% Plot raw data
159  N = length(pos_data);
160  new_pos_data = zeros(N,1);
```

```
161
162 for i=1:N
163     new_pos_data(i) = pos_data(1)-pos_data(i);
164 end
165
166 halfway=floor(N/2);
167
168 figure;
169 subplot(2,1,1)
170 hold on;
171 %plot(new_pos_data(1:halfway), load_data(1:halfway), '.-')
172 %plot(new_pos_data(halfway+1:end), load_data(halfway+1:end), '.-')
173 plot0a=plot(new_pos_data,load_data,'.-')
174 xlabel('extension, mm');
175 ylabel('load, lbs');
176 subplot(2,1,2)
177 plot0b=plot(pos_data, temp_data, '.-')
178 xlabel('extension, mm');
179 ylabel('temperature, degrees');
180 saveas(plot0a,'load_extension','png'); %saves it in the folder where this .m file
      is
181 saveas(plot0b,'temp_extension','png'); %saves it in the folder where this .m file
      is
182
183 %% Plot stress-strain
184 % this is for load_data being in lbs
185 stress_data = load_data*.4536*9.81/(crosssecarea/1000^2); %N/m^2=Pa
186 stress_data_zeroed = stress_data(:)-stress_data(1);
187 strain_data = new_pos_data/gaugelength;
188
189 figure;
190 plot1=plot(strain_data*100, stress_data_zeroed/1000,'.-');
191 ax=gca;
192 xlabel('Strain [%]')
193 ylabel('Stress [kPa]')
194 yt=ax.YAxis;
195 yt.FontSize=fontsize;
196 xt=ax.XAxis;
197 xt.FontSize=fontsize;
198 set(gca,'linewidth',linewidth);
199 set(plot1,'LineWidth',linewidth);
200
201 saveas(plot1,'stress_strain','emf'); %saves it in the folder where this .m file is
202 saveas(plot1,'stress_strain','png'); %saves it in the folder where this .m file is
203
204
205 save('stress_data.mat','strain_data','stress_data_zeroed','width','thickness','
      gaugelength_in','gaugelength','crosssecarea')
206
207 %% If you want to close the connection
208 % PIdevice.CloseConnection ();
209 %
210 % If you want to unload the dll and destroy the class object
211 % Controller.Destroy ();
212 % clear Controller;
```

```
213 % clear PIdevice;
```

The `test_parameters.m` code:

```
1  function [dl, ldot, t, epsdot_actual] = test_parameters(eps, lo, unit,epsdot)
2
3  eps = eps/100; % not percentage of strain
4
5  % make sure that everything is in the appropriate units
6  lo = 0.0254*lo*10^3; % [mm] convert the gauge length from inches into mm
7  dl = lo*eps; % [mm] total elongation needed for sample
8
9  %epsdot = 10e-2; % strain rate
10
11 ldot_desired = lo*epsdot; % [mm/s] velocity of sample
12 ldot = round(ldot_desired/.0005)*.0005;
13 epsdot_actual = ldot/lo;
14 epsdot_min = 0.0025/lo;
15 if ldot < .0025
16     warning(sprintf('Test velocity too small (must be >= 0.0025 mm/s). For given
       gauge length, epsdot must be >=%.3e',epsdot_min))
17 end
18 warning(sprintf('WARNING: Actual epsdot changed from %.3e to %.3e due to test
       velocity quantization.\n', epsdot, epsdot_actual))
19
20
21 t = 2*dl/ldot; % [s] time that the test will take to run
22
23 t = t/60; % [min] convert time to minutes
24
25 end
```

The `init_load.m` code:

```
1  function inet = init_load( varargin )
2  % -------------------------------------------------------------
3      % First optional argument 'simulate_i555_instead_of_connecting_to_hdwr':
4      %   if true, simulate i555 hardware instead of working with real hardware.
5
6      if (length(varargin) >= 1)
7          simulate_i555_instead_of_connecting_to_hdwr = varargin{1};
8      else
9          simulate_i555_instead_of_connecting_to_hdwr = false;
10     end
11
12     % -------------------------------------------------------------
```

```
13      % Second optional argument 'instrunet_prf_settings_filename':
14      %   name of .prf settings file to load, pass '' if you don't want to load one.
15
16      if (length(varargin) >= 2)
17          instrunet_prf_settings_filename = varargin{2};
18      else
19          instrunet_prf_settings_filename = '';
20      end
21
22      % -------------------------------------------------------
23      % Initialize instrunet and get one object that describes it called
24      % 'inet'. this is like running instrunet world software. you can only
25      % have one of these running at a time. and you cannot run instrunet
26      % world application software outside matlab while this object is alive.
27      % you need to dispose of it after you use it or stop running matlab and that
        will
28      % dispose of it. To dispose of it, call 'Delete(inet)'.
29
30      inet = instruNet_Class( ...
31          instrunet_prf_settings_filename, ...                % name of .prf
        settings file to load, pass '' if you don't want to load one
32          simulate_i555_instead_of_connecting_to_hdwr, ...    % if true, simulate
        i555 hardware instead of working with real hardware
33          false, ...      % show_alert_before_loading_settings_file, set to true if
        you want to show an alert before loading prf file
34          true, ...       % showAlertIfPrfIsNotFound, set to true if you want to
        show an alert if prf is not found
35          false, ...      % turn_off_all_alert_dialogs_when_initialize, set to true
        if you don't want any alerts during initialization
36          false, ...      % showNetworkPageAfterLoadDriver, set to true if you want
        to show instrunet world NETWORK page after loading
37          false);         % showRecordPageAfterLoadDriver, set to true if you want
        to show instrunet world RECORD page after loading
38
39      % -------------------------------------------------------
40      % Get the quantity of differential analog input channel connected to MATLAB.
41
42      gNumOfDiVinChannels = inet.ain_di.qty;
43
44      % -------------------------------------------------------
45      % Useful variables stored inside instruNet_Class object:
46      %
47      %   inet.system.initializedSuccessfully - true if instruNet initialized ok
48      %   inet.system.weAreSimulating_i555    - true if simulating i555
49      %   inet.system.loadedPrfOk             - true if loaded .prf settings file
        without error
50      %   inet.system.triedToLoadedPrfFile    - true if tried to loaded .prf
        settings file
51      %   inet.system.prf_fileandpath_name    - name of loaded .prf settings file,
        if one was loaded
52      %   inet.ain_di.qty                     - number of differential analog input
        channels
53      %   inet.ain_se.qty                     - number of single-ended analog input
        channels
54      %   inet.aout.qty                       - number of analog output channels
```

```
55
56      % ------------------------------------------------------------
57      % Run example code that reads/writes scalar values to/from instrunet
58      % hardware. This includes channels (e.g. analog input) and parameters
59      % (e.g. sensorType).
60
61      %instruNet_Class.ShowAlert('TORI: How to read a scalar (i.e. one value) read/
        write with instruNet hardware. Press OK to continue.');
62
63      if (inet.ain_di.qty <= 0)
64          %instruNet_Class.ShowAlert('TORI: We cannot run instrunet_scalar_io() due
        to not having hardware channels. Does Matlab have iNet driver open in another
        instance? If so, you need to close it (try Exiting Matlab and come back in).')
        ;
65          iNetErr = 1;
66          return;
67      end
68
69      %%scalar reading
70      % Load iNet channel address (numNum, devNum, modNum, chNum) into
71      % structs 'sAin1' and 'sDio30'.
72      loadcell1   = Load_Channel_Address(inet, 1,1,1, 13);
73      % Get loadcel1's channel's value given channel address stored in loadcell1
        struct
74      [ain_valueIn, iNetErr]      = GetChannelValue(inet, loadcell1);
75      ain_valueIn; %this should be the load in Newtons at the point when we run that
         line
76
77     % try reading more data points of the load!
78     tic
79      [ain_valueIn2, iNetErr]      = GetChannelValue(inet, loadcell1);
80      toc
81      ain_valueIn2;
82      [ain_valueIn3, iNetErr]      = GetChannelValue(inet, loadcell1);
83      ain_valueIn3;
84      [ain_valueIn4, iNetErr]      = GetChannelValue(inet, loadcell1);
85      ain_valueIn4;
86
87
88      %% digitizing
89
90      % ------------------------
91      % End user sets this to 'true' to open instrunet world RECORD page after we
        digitize,
92      % to see your data and check your set up.
93
94      open_record_page_to_see_data_after_we_finish_digitizing = true;
95
96      % ------------------------
97      % End user sets this to 'true' to plot digitized data in new window
98
99      plot_digitized_data_in_new_window = true;
100
101     % ------------------------
102     % End user sets this to 'true' to calculate average value of each
```

```
103     % digitized point for each channel.
104
105     calculate_average_value = false;
106
107     % ------------------------
108     % End user sets this to 'true' if you want to process complete buffers (scans)
         at a
109     % time. Alternatively, set to false to process little segments
110     % of data as they become available.
111
112     processOneScanAtATime = true;
113
114     % ------------------------
115     % End user sets this to 'true' if you want mouse down to stop digitization
116
117     allowMouseDownToStopDigitize = false;
118
119     % make sure we are not digitizing and get ready to start digitizing
120     Stop_iNet_Digitizing(inet);
121
122     % ------------------------
123     % Get 'numOfDigitizeChannels' and 'ptsPerScan' values from iNet32.dll
124     % and also create zero'ed ram buffer to hold digitize data in
125     % matlab 2x2 array: 'inet.digitize.data.value(1:ptsPerScan, 1:
        numOfDigitizeChannels)'.
126
127     Digitize__Allocate_2x2_Data_Matrix(inet);
128 end
```

The `get_load.m` code:

```
1  function [temp_val, load_val] = get_load(inet)
2      load_vals = [NaN, NaN];
3
4      % Collect Digitized Data
5      %
6      % It is ABSOLUTELY NECESSARY to let your computer's processor service
7      % instruNet digitizing buffers periodicaly 10 to .3 times per second)
8      % while digitizing.  This is done by continuously calling
9      % DigitizeListOfChannels() or Service_All_iNet_Digitize_Buffers() in a loop.
10
11     allowMouseDownToStopDigitize = false;
12     processOneScanAtATime = true;
13     check_for_controllerNotRespondingTimeout = true;
14     [   iNetErr, ...
15         last_scanNum_base1_tsfred_inFull_oneCtlr_int64, ...
16         we_stopped_digitizing_machine_or_it_finished_on_its_own, ...
17         we_started_digitizing_machine_since_it_was_not_on, ...
18         totalNumOfPtsAccessed_AllInputChannels_ThisRoutine_int64, ...
19         endUser_sampleRate_PointsPerSecondPerChannel_actual_double, ...
20         total_time_for_entire_digitization_Secs_actual_double, ...
```

```
21          total_time_for_each_scan_Secs_actual, ...
22          user_pressed_mouse_button_to_stop_early  ...
23      ] = DigitizeListOfChannels(inet, ...
24          allowMouseDownToStopDigitize, ...
25          processOneScanAtATime, ...
26          check_for_controllerNotRespondingTimeout);
27
28      if (iNetErr ~= 0)
29          return;
30      end
31
32      % Update these timing parameters (from internal inet32.dll memory):
33      % inet.digitize.p.ptsPerScan, inet.digitize.p.noOfScans, inet.digitize.p.
        endUser_sampleRate
34
35      if (1) %weAreDoingOurFirstAccessOfData == true)
36          [iNetErr] = Digitize__UpdateTimingParameters(inet);
37      end
38
39      % scan thru all channels enabled for high speed digitize
40      for digiChNum_base1 = 1 : inet.digitize.p.numOfChannels
41
42          % READ DIGITIZED DATA
43          %
44          % find out how much data is in digitize channel # 'digiChNum_base1'
45          % data buffer, and move it into matlab 2x2 data array
46          % 'inet.digitize.data.value(1:ptsPerScan, 1:numOfDigitizeChannels)'.
47          %
48          % first_PointOfSegment_Indexbase1 = pointToPullindex_base1
49          % last_PointOfSegment_Indexbase1  = pointToPullindex_base1 +
        numPointsToPull - 1
50          %
51          % first_PointOfSegment  = inet.digitize.data.value(
        first_PointOfSegment_Indexbase1, digiChNum_base1)
52          % last_PointOfSegment   = inet.digitize.data.value(
        last_PointOfSegment_Indexbase1,  digiChNum_base1)
53          %
54          % data segment          = inet.digitize.data.value(
        first_PointOfSegment_Indexbase1 : last_PointOfSegment_Indexbase1,
        digiChNum_base1)
55
56          [   iNetErr, ...
57              src_pointToPullindex_base1, ...             % {1... numPtsPerScan)
        index of the first point of the segment that we need to process (base 1 index)
58              src_numPointsToPull, ...                    % # of points to to pull
        out of the buffer
59              src_scanNumberToPull_base1 ...              % scan number {1...
        numScans} of the scan that we are currently pulling (base 1)
60          ] = GetDigitizedSegment(inet, digiChNum_base1);
61
62
63          if (iNetErr ~= 0)
64              return;
65          end
66
```

```
67          % if we have new data from this channel (i.e. number of new points
68          % is specified by variable 'src_numPointsToPull')...
69          if (src_numPointsToPull > 0)
70              % Calculate the total # of pts that has been digitized since we
    started digitizing
71  %              totalPtsPulledSoFar(digiChNum_base1) = totalPtsPulledSoFar(
    digiChNum_base1) + src_numPointsToPull;
72
73              % Calculate where segment is w.r.t. our ram buffer
74              % data segment = inet.digitize.data.value(
    first_PointOfSegment_Indexbase1 : last_PointOfSegment_Indexbase1,
    digiChNum_base1)
75              first_PointOfSegment_BufferIndexBase1 = src_pointToPullindex_base1;
76              last_PointOfSegment_BufferIndexBase1  = src_pointToPullindex_base1 +
    src_numPointsToPull - 1;
77
78              % Calculate the time of the first and last points of the segment,
    units Seconds, relative to when we started digitizing.
79              first_PointOfSegment_Secs       = ( double(
    first_PointOfSegment_BufferIndexBase1-1)   * inet.digitize.p.
    endUser_samplePeriodSecs ) ...
80                                              + ( double(src_scanNumberToPull_base1
    -1)              * inet.digitize.p.secsPerScan );
81
82              last_PointOfSegment_Secs        = ( double(first_PointOfSegment_Secs)
    ) ...
83                                              + ( double(src_numPointsToPull-1) *
    inet.digitize.p.endUser_samplePeriodSecs);
84
85              % true if segment includes the last point in the ram buffer
86              segmentIncludesLastPointInRamBuffer = (
    last_PointOfSegment_BufferIndexBase1 == inet.digitize.p.ptsPerScan);
87
88              % true if segment includes the last point in the ram buffer
89              segmentIncludesLastPointInEntireDigitization =
    segmentIncludesLastPointInRamBuffer && (src_scanNumberToPull_base1 == inet.
    digitize.p.noOfScans);
90
91
92              load_vals(digiChNum_base1) = inet.digitize.data.value(
    first_PointOfSegment_BufferIndexBase1 : last_PointOfSegment_BufferIndexBase1,
    digiChNum_base1);
93          end
94      end
95      temp_val = load_vals(1);
96      load_val = load_vals(2);
97  end
```

The `PI_move.m` code:

```
1  function PI_move(startPos, stopPos, step, steps, PIdevice, axis, inet)
```

```
2       global pos_data;
3       global temp_data;
4       global load_data;
5       global time_data;
6       global datetime_starttest;
7       fid = fopen('log.txt', 'a+');
8       finishup = onCleanup(@() myCleanUpFun(PIdevice, axis, fid));
9
10      logging = true;
11      %%% this following code is in the move function
12      % assume that referencing stuff has made it so that when this function is
        called, the PIdevice is located at startPos
13
14      pos_data(1) = PIdevice.qPOS(axis);
15      start = tic;
16      time_data(1) = toc(start);
17      [temp_data(1), load_data(1)] = get_load(inet);
18      datetime_starttest = datetime;
19
20      waiting = true;
21      for i = 2:length(steps)
22          % Move the device
23          step_size = steps(i) - steps(i-1);
24          PIdevice.MOV(axis, steps(i) + step_size/2);
25          % Record current data
26          pos_data(i) = PIdevice.qPOS(axis);
27          [temp_data(i), load_data(i)] = get_load(inet);
28          time_data(i) = toc(start);
29          v = [time_data(i), pos_data(i),load_data(i), temp_data(i)]
30          if logging
31              try
32                  fprintf(fid, '%f,%f,%f,%f\n', v);
33              catch
34                  logging = false;
35                  disp('logging failed')
36              end
37          end
38          if ~waiting
39              disp('logging took too long')
40          end
41          % Wait until it's moved to the new location
42          step_sign = sign(step_size);
43          waiting = false;
44          while(step_sign*PIdevice.qPOS(axis) < step_sign*steps(i))
45              waiting = true;
46              continue;
47          end
48
49      end
50
51  end
52
53  function myCleanUpFun(PIdevice, axis,fid)
54      PIdevice.MOV(axis, PIdevice.qPOS(axis));
55      global pos_data
```

```
56      global temp_data
57      global time_data
58      global load_data
59      global datetime_starttest
60      save('extension_data.mat','time_data','pos_data','temp_data','load_data','
        datetime_starttest')
61
62      try
63          fclose(fid)
64      catch
65      end
66  end
```