Micromechanics of Anisotropic Partially Crystalline Emulsions

by

Alexandra Victoria Bayles

A thesis submitted to the Faculty of the University of Delaware in partial fulfillment of the requirements for the degree of Honors Degree in Chemical Engineering with Distinction

Spring 2013

MICROMECHANICS OF ANISOTROPIC PARTIALLY CRYSTALLINE EMULSIONS

by

Alexandra Victoria Bayles

Approved: _____
Eric M. Furst, PhD
Professor in charge of thesis on behalf of the Advisory Committee

Approved: _____
Abraham M. Lenhoff, PhD
Committee member from the Department of Chemical and Biomolecular Engineering

Approved: _____
Susan E. Groh, PhD
Committee member from the Board of Senior Thesis Readers

Approved: _____
Michael Arnold, Ph.D.
Director, University Honors Program

ACKNOWLEDGMENTS

There are many people that have enabled me to work on this project, and I want to acknowledge their support here. First and foremost, I would like to thank Professor Furst for acting as an ever patient and helpful mentor over the past 4 years. I will forever be grateful that he took a chance on a completely inexperienced 17 year old that asked to work in his lab without having taken any chemical engineering courses. I know that I will continue to draw on his advice and attempt to emulate his enthusiasm for science well beyond my undergraduate years. Professors Lenhoff and Groh have also served as excellent thesis advisors, and I appreciate the support and suggestions they've provided over the past year.

I would also like to thank all those who mentored me during my internship at Procter & Gamble and continue to serve as project collaborators. In particular, I would never have had the opportunity to work on this project without Patrick Spicer, who sat me down on the first day of my internship and told me about his idea for endoskeletal droplets. His idea turned into something tangible, incredibly interesting, and potentially useful, and I am so happy that I got to be a part of the entire evolution. Amar Pawar, Marco Caggioni, Seth Lindberg and Jessica Lenis have also all been fantastic teachers and collaborators. I am continually awed by your collective knowledge and resourcefulness.

The entire Furst Group has been incredibly supportive throughout all of my work. In particular, Kelly Schultz got me started in the group, and has continued to give me excellent research and career advice even after she earned her doctorate.

Peter Beltramo helped me order and configure equipment, and interpret results even though it had absolutely nothing to do with his work. I also credit Peter for keeping me sane while I attempted to put together this thesis during the past year. Hyelim and Lilian are saints for having me as a lab equipment trainer and not yet quitting the program. All of the rest of the Furst group members – Jim, Matt, Kathy, Jon and Mark – are all fantastic for letting the perpetually annoying undergrad crash in their offices and bounce her crazy ideas off of them.

Moving outside of the world of Colburn, I have to thank my family for being supportive of all that I've ever done. Throughout these past 4 years, I specifically want to thank my mom, Taryn, for teaching me how to crash research conferences; and I want to thank my dad, Gary, for being willing to proofread things that I sent him at 2 AM. My parents tend to say they should have never been allowed breed because any child two PhD Chemical Engineers would be worthy of institutionalization. They were right for 50% of the population. Though I was the one to get all the crazy genes, I've at least found a venue to further cultivate them. Meanwhile, my brother, Nathan will forever be our only tie to normalcy. Nathan, thank you for your perfect humor, infinite coolness, and willingness to claim Mom, Dad and I as family.

Here at UD, I've made the greatest friends that anyone could ask for. All of them have supported me through all of my successes and downfalls—both thesis and non-thesis related. Katie, Eric, Lindsey, Kadija, Kara, Christine, Johndavid, Matt, Phil, Christina and Jason – you guys specifically have made these past four years the best years of my life.

Most of all, I thank God, without Whom nothing would be possible.

TABLE OF CONTENTS

LIST OF FIGURES

ABSTRACT

Advancements in the synthesis of anisotropic colloids enable complex fluid systems to be designed with responsive properties. A capillary-based micromanipulation technique is developed to produce anisotropic droplets from the dispersed oil phase of partially crystalline emulsions. These particles have distinct non-spherical features with characteristic length to width aspect ratios ranging from 1 to 10. The particles demonstrate an ability to undergo sudden shape changes in response to changes in interfacial tension and temperature. During these deformation processes, the oil-water interface imposes a stress on the semi-crystalline network within each particle. When the imposed stress exceeds the critical stress of the crystalline material, the network yields and causes the particle to transition to a spherical shape. A model for morphological stability relating internal stress distributions to the interfacial Laplace pressure is developed to characterize this deformation. The analytical model captures qualitative trends observed in axisymmetric particle deformation, including the relationship between interfacial tension, crystalline content, characteristic size and regions prone to deformation. Through the development of a preliminary discrete modeling program, we lay the groundwork to quantitatively predict the deformation of experimental particles. Continued development of this modeling program will improve understanding of this system, and potentially provide product engineers with a tool to evaluate new particle shapes and emulsion environments. Continued experimental deformation, deposition, and material studies will reveal other modes of particle responsiveness and stimuli.

Together, the experimental, theoretical and computational approaches begun here will advance the understanding of anisotropic partially crystalline emulsion particles and their use in enhanced deposition applications.

Chapter 1

INTRODUCTION

1.1 Motivation and Goals

The overarching goal of this thesis is to develop means of producing and characterizing shape anisotropic particles from partially crystalline emulsions. Shape anisotropic particles are of particular interest due to their potential in enhanced delivery of suspended active ingredients to substrates, such as in commercial drug formulations, detergents, pesticides, cosmetics and foods [1, 2]. In these applications, the effectiveness of active delivery is linked to the morphology of the phase in which the ingredient is dispersed. The active morphology can affect the probability of substrate contact, the retention of phase after initial adhesion, and the area of mass transfer available to the active.

In the previously mentioned applications, the morphology of the active phase is limited by the nature of the dispersant. Historically, numerous pharmaceutical and commercial products have been emulsions where the active ingredient is suspended in the dispersed, liquid phase. Using emulsions offers many practical advantages in these products. Emulsions can enable the dissolution of multiple compounds, can be easy to prepare, and can wet and adhere to substrates like hair, skin, and other biological surfaces [1, 3]. However, the natural morphology of the dispersed phase in liquid-in-liquid emulsions often reduces active delivery effectiveness. The energetically favored morphology of a liquid dispersed in another of different hydrophobicity (assuming the viscosity, elasticity and density of the two liquids are

similar), is a sphere [4]. In this state, the interfacial area per unit of volume is at a minimum, and hence, the interfacial energy of the system is minimized. For the same reasons that a spherical morphology is energetically stable, a spherical droplet can act as a poor delivery vehicle. Having a 1:1 length to width aspect ratio reduces the probability that the dispersed phase will contact substrate [5]; there are multiple paths during flow that the droplet can take that do not intercept the substrate. If the droplet does contact the substrate, the low surface area of the droplet can reduce both its ability to adhere to the substrate and the time that the droplet remains in contact with the substrate. Furthermore, the low contact area between a spherical droplet and a substrate limits the area for mass transfer. Both of these scenarios are particularly true for droplets that exhibit poor wettability with the intended substrates.

As a result of the limited morphology of traditional emulsions, much of the active ingredient in emulsion-based products is not delivered to the substrate. Instead, the active ingredient remains in the dispersed phase of the emulsion. This dispersed phase either does not make it to the substrate, or is removed or rinsed from the substrate within short time scales. Thus, a high fraction of the active ingredient goes to waste. To compensate for the active lost, many products are designed with excess active so that the benefits of the active are still delivered. Active waste in products is detrimental due to its impact on economics and sustainability. The active ingredient is also typically the most expensive component of a product [6, 7]. Reducing the amount of active required to achieve the same product benefits could significantly reduce the cost of manufacture of emulsion-based products. Furthermore, efforts to improve the sustainability of products stress that limiting the use of material in products is desirable, as it reduces the environmental cost of obtaining and processing raw

components [8]. For products where the active ingredient poses some sort of environmental hazard, reducing the active wasted and subsequently released into the environment is also advantageous. For these reasons, there is a particular need to improve the efficacy of delivery of actives in emulsion-based products.

Altering the morphology of the active phase offers one potential means of increasing delivery efficiency. Non-spherical particles enhance delivery by improving deposition, retention and mass transfer. Figure 1.1 illustrates the potential benefits of using a non-spherical morphology, like a spherocylinder, over a spherical morphology to deliver actives.

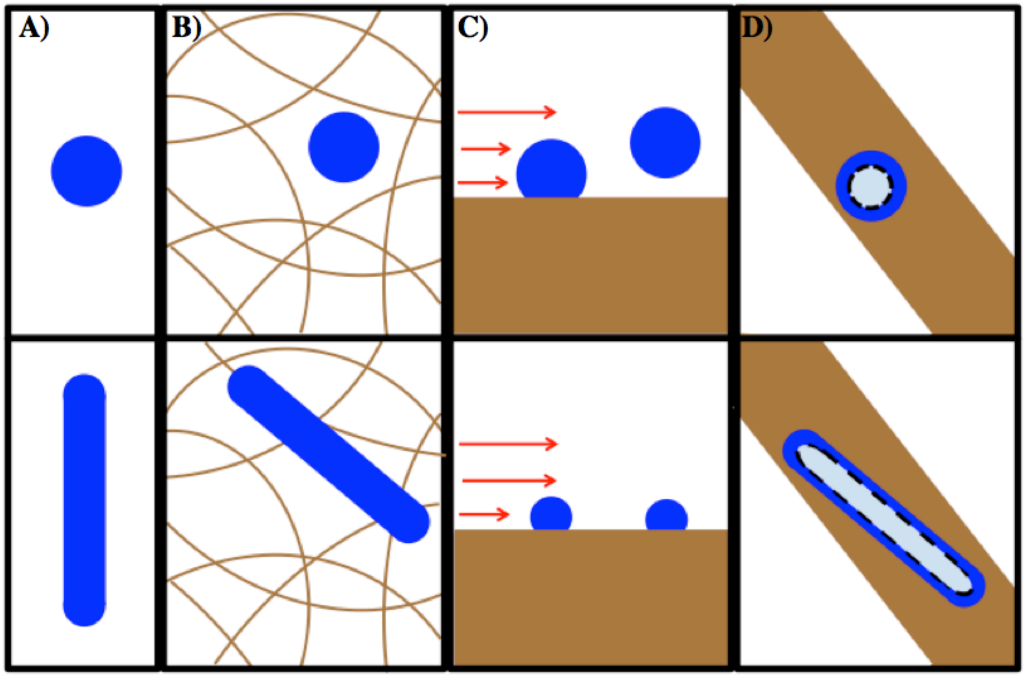


Figure 1.1: Deposition of spherical and non-spherical droplets onto various substrates. A) Cross section of spherical (upper) and nonspherical (lower) droplets. The droplets depicted have the same volume. B) Deposition onto fine fibrous substrate. The sphere has a lower probability of contact than the spherocylinder. C) Retention of droplets after deposition onto flat substrate subject to imposed flow (red arrows). Axis-cross section of spherocylinder illustrated in lower panel. The smaller height of the spherocylinder improves its retention on the substrate. D) Contact area on flat or coarse fibrous substrate. Spherocylinder has increased contact area (outlined by dashed line).

In Figure 1.1B, the probability of contact between the droplet and the substrate is greater for the spherocylinder. Though the droplets have the same total volume, the effective collision volume increases with length to width aspect ratio, thus increasing likelihood of contact. In Figure 1.1C, the increased height of the spherical droplet decreases the retention of the droplet on the substrate when an external flow is

imposed. In Figure 1.1D, the spherocylinder provides an increased contact area between the droplet and the substrate assuming both droplets exhibit the same wettability. Since the rate of mass transfer is fundamentally proportional to the area of contact, this also enhances delivery. Evidence of the phenomena depicted in Figure 1.1 is found in nature. Bacterial cells utilize non-spherical, anisotropic cell morphologies to improve deposition [2, 9].

As described, utilizing non-spherical droplets offers high potential for enhanced active delivery. For this reason, we defined the first goal at the outset of this work as follows:

> Goal 1: Develop a method to produce soft particles with non-spherical, anisotropic shapes to deliver actives in emulsion-based products.

In the previous goal, "soft" refers to particles that have non-rigid, liquid components. As described in Section 1.2, multiple methods have been developed to produce solid anisotropic colloids. These solid colloids, however, cannot be directly utilized in many products, since it is often difficult to entrain liquid active ingredients within solids, and because the materials used in laboratory studies are not optimal for consumer use. As an alternative to solid materials, partially crystalline emulsions (described in Section 1.3) were studied using micromanipulation, a technique previous used to study cell behavior [10, 11] and measure surface tension [12].

As described in the contents of this thesis, the first goal was accomplished by developing a micromanipulation method to constrain the crystallization of the dispersed phase of partially crystalline emulsions. Particles produced using this method not only have anisotropic morphologies, but also demonstrate the ability to undergo morphology variation in response to external stimuli. The dynamic

morphologies of the particles offer a new degree of functionality for emulsion products. Active deposition can not only be enhanced, but also controlled through external stimuli.

Control of droplet morphology requires a thorough, quantitative understanding of how different shapes and sizes respond to external stimuli. With this in mind, we defined the second goal of this work as follows:

> Goal 2: Develop a model to quantitatively describe the morphology of anisotropic partially crystalline emulsion droplets.

In pursuit of the second goal, theories of interfacial phenomena in emulsion systems (described in Section 1.5) were integrated to describe the PCE particles. Thus far, the model describes qualitative modes of morphological deformation.

In the context of the specified goals, we briefly describe previous work in anisotropic colloids and partially crystalline emulsions in the remaining sections of this chapter. We conclude the chapter by providing an overview of the methods utilized and results obtained in this work, thus outlining the thesis.

1.2 Anisotropic Colloids

Colloids, which are particles that range in size from tens of nanometers to microns [13], find application in countless fields. A colloid that has some property or characteristic that varies across its body is considered anisotropic. Glotzer defines eight common dimensions of anisotropy in colloidal particles: surface coverage, aspect ratio, faceting, pattern quantization, branching, chemical ordering, shape gradient, and roughness [14]. Advancements in the synthesis of colloidal particles have exploited these dimensions of anisotropy, and have allowed for particles to be tailor made for specific applications.

Of the anisotropy dimensions defined by Glotzer, aspect ratio, faceting, branching and shape gradient pertain to particle shape. Multiple methods, including droplet templating and assembly [15], and stop-flow lithography [16], micron-scale molds [17] have been used to synthesize a diverse spectra of complex-shaped solid colloids. These solid colloids find application in active ingredient delivery [18], self assembly [19], drug delivery [2, 20, 21], batteries [22], optics [23], and biomimicry [24], though this is hardly an all-inclusive list.

Common materials for solid colloids include metals, silica and polymers [14, 25 – 27]. While the solid character of shape anisotropic colloids does not inhibit their incorporation in the previous mentioned applications, it does limit their effectiveness in commercial products that utilize liquid droplets as delivery vehicles. As previously discussed, liquid droplets are specifically used in certain commercial applications because of their ability to wet and adhere to substrates [1, 3]. Furthermore, many of the exotic materials (such as alloys of rare metals) used to produce shape anisotropic solid colloids are not suitable in commercial applications like detergents, pesticides, and drug delivery. The potential chemical toxicity and the cost compared to liquid droplet materials used the same applications make many solid colloids incompatible with commercial applications. However, by examining the nature of underutilized potential base materials (like partially crystalline emulsions) and developing new synthesis routes (as developed in this work), we can expand the application space of shape anisotropic particles.

1.3 Partially Crystalline Emulsions

Partially crystalline emulsions (PCE) have specific characteristics that lend themselves to the formation of anisotropic particles with liquid characters. PCE are

oil-in-water emulsions that contain solid crystals within the dispersed phase. Common applications of partially crystalline emulsions include drug delivery [28], foods [29, 30], shampoos [31] and cosmetics [32].

Partially crystalline emulsions are prepared by first homogenizing the dispersed phase components at a temperature above the melting point of the crystal. This non-polar liquid is then emulsified in water in the presence of surfactant at a high temperature. The dispersed phase naturally forms spherical droplets as surface tension minimizes the interfacial area between the immiscible phases. After emulsification, the bulk is allowed to cool, initiating crystallization within the dispersed phase.

The crystalline network present within the oil droplets has an inherent elasticity that resists deformation from the shape in which it crystalizes. Generally, droplets assume a thermodynamically favorable state through minimization of the combined elastic energy embodied within the network and interfacial energy. The drive to minimize these combined elastic and interfacial energies promotes the formation of anisotropic particles in the form of partially coalesced droplets, which are the products of droplet collisions [31, 33]. The morphologies of such particles range from dimpled spheres to doublets to spheroid-cylinders, depending on the degree of coalescence. While this phenomenon produces non-spherical droplets readily in bulk (particularly in creaming, as illustrated in Figure 1.2), consistency in particle formation has been limited by poor control over droplet collisions [34]. Furthermore, the morphology of particles produced from arrested coalescence is limited in terms of dimension and shape. The collision of two droplets, for example, results exclusively in spherical derivatives (i.e. doublets and spheroid-cylinders) whose primary length-to-radius aspect ratio does not normally exceed two. However, by developing the

methods presented in this work, we demonstrate the ability to produce and control of anisotropic partially crystalline emulsion morphologies.

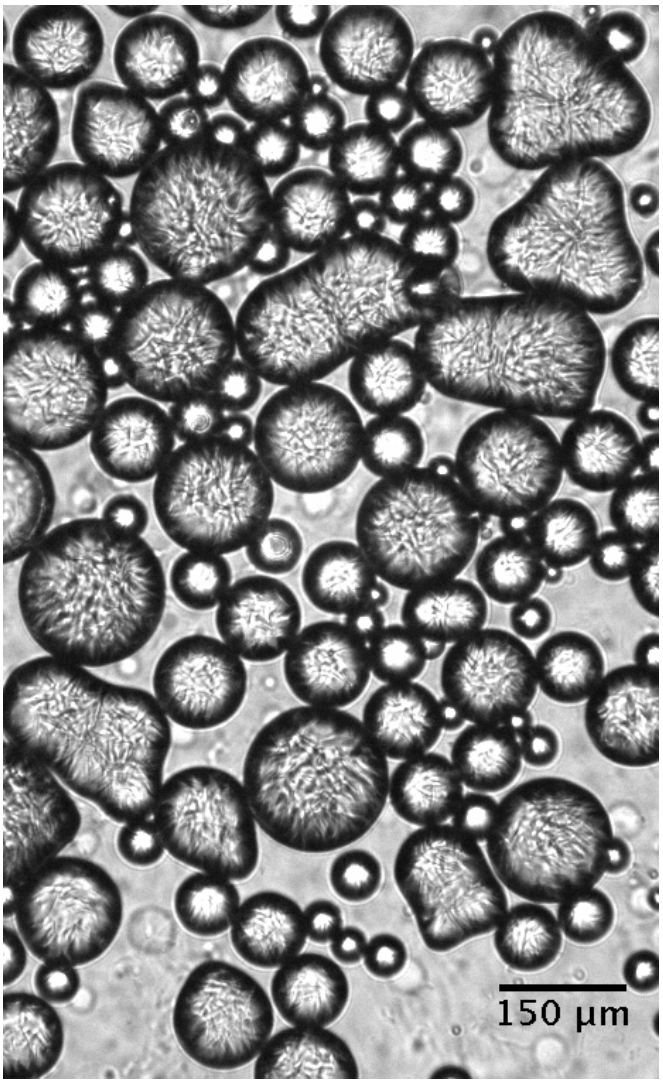


Figure 1.2: Spherical and non-spherical morphologies that result from creaming in partially crystalline emulsions.

1.4 Thesis Overview

In this thesis, we develop means of producing and characterizing shape anisotropic particles from partially crystalline emulsions. Chapter 2 describes the experimental techniques used and developed in this thesis. These include micromanipulation, capillary assisted molding, bright field microscopy and the application of thermal, interfacial tension and magnetic stimuli. Basic principles of these techniques are discussed along with the materials used and sample preparation methods.

Chapter 3 describes the different particle morphologies produced from partially crystalline emulsions. These include the spherocylinder, the "ball and stick", the "dual ball and stick" and the torus. Each of these morphologies exhibits some degree of dynamic behavior. Chapter 3 describes the different degrees of morphology stability, including a qualitative analysis of the principle factors that affect stability: solid content and interface curvature. Finally, the dynamic morphological response to thermal, interfacial tension, and magnetic stimuli is explored.

Chapter 4 discusses the models developed to describe the stability of different morphologies. Generally, these models were derived based on the balance between interfacial stresses and the resistance of the crystalline network to deformation. The simplest of the models is the spherocylinder model, where the internal stress is shown to be proportional to the Laplace pressure difference across the interface at the ends (along the axis) of the spherocylinder. To extend the fundamental Laplace pressure-stress relation to axisymmetric particles with non-uniform curvature (i.e. the ball and stick morphologies), a more complex model was developed. The three dimensional internal stress state at points within the droplet is shown to be a function of the variable Laplace pressure difference across the entire droplet interface. Notably, the

generalized axisymmetric model reduces to the same preliminary model in the spherocylinder case, indicating consistency between models. Internal stress distributions for idealized axisymmetric particles are presented, as well as their qualitative agreement with experimental observations. Finally, the entirely different approach used to model the dynamic deformation of torus shaped particles is described.

Chapter 5 details the preliminary discrete modeling program developed to characterize experimental particles through the lens of the axisymmetric model. The successes and limitations of this discrete modeling program are discussed, as well as suggestions for future program development. Finally, Chapter 6 summarizes our conclusions and proposes future direction for this work.

Chapter 2

MATERIALS AND METHODS

The aim of this thesis is to develop means of producing and characterizing shape anisotropic particles from partially crystalline emulsions. To accomplish this goal, a technique derived from basic micromanipulation, named capillary assisted molding, was developed. This chapter describes the apparatus constructed to perform the micromolding technique. Protocols used to subject anisotropic partially crystalline emulsion particles (APCE) to external stimuli are also described, along with bright field microscopy techniques used to image droplet morphology. Finally, the materials used in the study are detailed along with sample preparation techniques.

The materials and methods described in this chapter were first developed during a summer internship at Procter & Gamble Co. in Cincinnati, OH in 2011. When the fundamental study of APCE was relocated to the University of Delaware in the fall of 2011, these techniques were marginally adapted to utilize existing equipment. The equipment and material specifications provided here describe those used at the University of Delaware, although some of the results presented in this work were obtained using the equipment at Procter & Gamble. Differences between University of Delaware and Procter & Gamble equipment and protocols were designed to be minimal so that results obtained at the two locations could be compared. As such, we do not detail the similar apparatus and protocols used at Procter & Gamble here.

2.1 Experimental Methods

The following describes the experimental techniques employed to produce and study the anisotropic partially crystalline emulsion particles.

2.1.1 Capillary Assisted Molding Apparatus

The capillary assisted molding (CAM) apparatus was designed based on micromanipulation apparatuses used in biological cell and surface tension studies [10-12]. This apparatus was first designed and constructed at Procter & Gamble in Cincinnati, OH for the study of Pickering emulsions and partially crystalline emulsions [31]. A secondary apparatus was designed and constructed in Newark, DE when the fundamental study of APCE was moved to the University of Delaware. Figure 2.1 provides a schematic of the secondary apparatus.

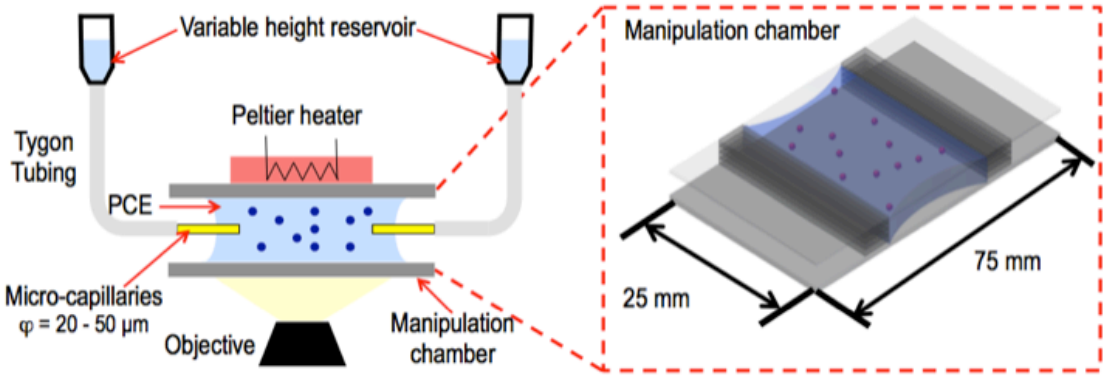


Figure 2.1: Schematic of capillary assisted molding apparatus with inset of micromanipulation chamber. Note that individual components are not depicted to scale.

The CAM apparatus consists of a glass chamber used to hold the emulsion sample; microcapillaries connected to fluid reservoirs used to manipulate and mold the

droplets; coarse micromanipulators (not depicted in Figure 2.1) used to adjust the position of the microcapillaries; a Peltier heater used to supply heat and melt the crystals in the emulsion; a microscope used to image the sample; and a custom fabricated stage (not depicted in Figure 2.1) used to support the chamber. Each of these components is described in the following subsections.

2.1.1.1 Chamber

The sample chamber (Figure 2.1 inset, Figure 2.2) is fabricated from glass slides and UV initiated thiolene resin (Norland Optical Adhesive 81). Eight glass spacers are first cut from a large 75 x 25 x 1mm microscope slide (Fischerbrand Plain Microscope Slide). The spacers are cut to be ~ 5 mm in width. Thiolene resin is then applied on one side of two separate spacers, and the spacers are set on a large 75 x 50 x 0.15 mm microscope slide (Thermo Fischer Scientific), which serves as the bottom of the chamber. The spacers are positioned such that there is approximately 25 mm between them. The spacers and glass slide are then exposed to an ultraviolet flood lamp (Spectronics Corp. 365 nm, 120 V, 1.05 Amps, Model SB-100P) approximately 12 cm above the sample for 30 s. This exposure cures the resin, bonding the spacers to the glass slide. The 6 remaining spacers are then equally distributed between the two spacer column bases and bonded sequentially in the same manner. The final result is a large base slide with two, 4 spacer high columns.

Two lines of thiolene resin are then applied between the spacer columns (the lines are applied perpendicular to the columns, at each column end). The lines are cured under the UV lamp for 30 s. The lines serve to help prevent the liquid sample (held within the chamber by capillary action) from leaking out the open sides of the chamber.

Finally, a 75 x 25 mm glass coverslip (Fischer Scientific) is bonded to the glass columns. Resin is applied to each of the top spacers, the glass coverslip is positioned on top of the spacers, and the entire chamber is cured for at least 5 minutes. The long exposure time ensures all the resin in the chamber completely cures.

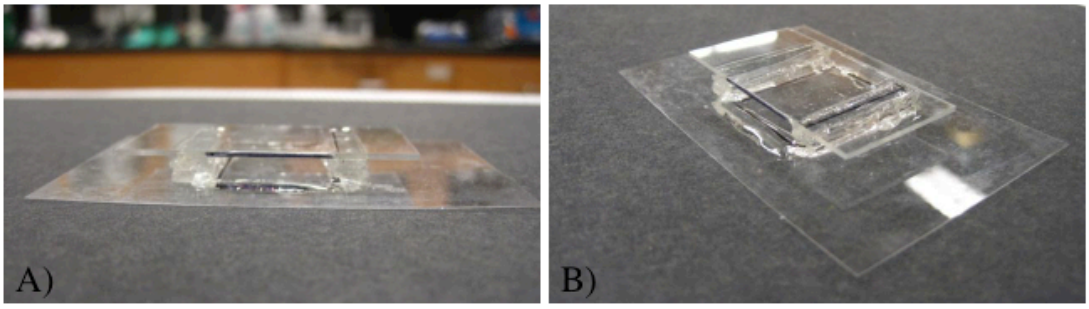


Figure 2.2: Glass micromanipulation chamber. A) Side View. B) Isotropic View

2.1.1.2 Microcapillaries

Tapered capillaries are fabricated from standard borosilicate glass capillaries (1 mm OD and 0.5 mm ID, Sutter Instruments) with a Micropipette Puller (P-97; Sutter Instruments, Taper settings: Prog. #2, Heat Ramp +15, Pull 0/55, Velocity 20/65, Time 200t/150t, Pressure 400). The tip of the pulled capillary is flattened using a Microforge (Model MF-830; Narishige Int'l. USA). Capillary tips range in diameter from 20 – 50 μm depending on the experiment. Microcapillaries are mounted on the micromanipulators, described in Section 2.1.1.5.

2.1.1.3 Hydrostatic Reservoir

The pressure exerted at the end of each microcapillary is controlled hydrostatically by varying the height of liquid reservoir attached to the

microcapillaries. Each capillary is attached to the reservoir with rubber tubing (Tygon ID 1/32 in). The rubber tubing is attached to the reservoir (10 mL syringe, BD 10 mL Luer-Lok) with a 16-gauge syringe needle (BD 16 G). The reservoir is mounted on a 450 mm long linear translation stage (Zaber, Model T-LSR450A-KT02) that controls the position of the reservoir within ±68 μm. The position of the stage can be varied manually using an integrated knob or digitally by specifying the position on a computer. The translation stage is oriented vertically next to the microscope on the microscope table, with the midpoint of the translation stage at the same height as the micromanipulation chamber. This positioning allows for both negative and positive pressure to be exerted at the end of the capillary when the reservoir is below and above the translation stage midpoint respectively.

Figure 2.3: Hydrostatic reservoir mounted on Zaber translation stage. Reservoir is connected to capillary with Tygon tubing.

2.1.1.4 Peltier Device

A Peltier device (Tellurex, Model C2-15-0404) supplies heat to the micromanipulation chamber. The Peltier device is a 15 x 15 x 3.6 mm thermoelectric module that transfers heat from one side of the module to the other by passing DC current across the device. In our apparatus, the hot side of the Peltier device is positioned so that it is flush with the top coverslip on the micromanipulation chamber. The device is attached to the coverslip using double-sided tape. The heat transferred with the device raises the bulk temperature of the emulsion above 60 °C, thereby melting the crystalline material in the dispersed phase. A DC power source (TE

Connectivity, 30 V, 5 Amps, Model HY2005-3) controls the voltage passed through the device, to control the rate of heat dissipation.

2.1.1.5 Microscope Stage and Micromanipulators

A custom microscope stage (Figure 2.4) was fabricated from aluminum to carry out CAM. The custom microscope stage has features that most traditional microscope stages do not have, including locations to mount micromanipulators and a smooth surface that enables the manual translation of the micromanipulation chamber during droplet capture. The custom stage was designed using Autodesk Inventor 3D CAD (computer aided design) Software. Appendix A contains technical drawings for the components of the stage. The stage was designed for use on a Zeiss inverted microscope (Carl Zeiss, Inc., Axiovert 200).

Two coarse, 3-axis micromanipulators (Narshige Int'l. USA, Model U-3C) are used to adjust the position of the microcapillaries. These micromanipulators are mounted on the microscope stage as shown in Figure 2.4.

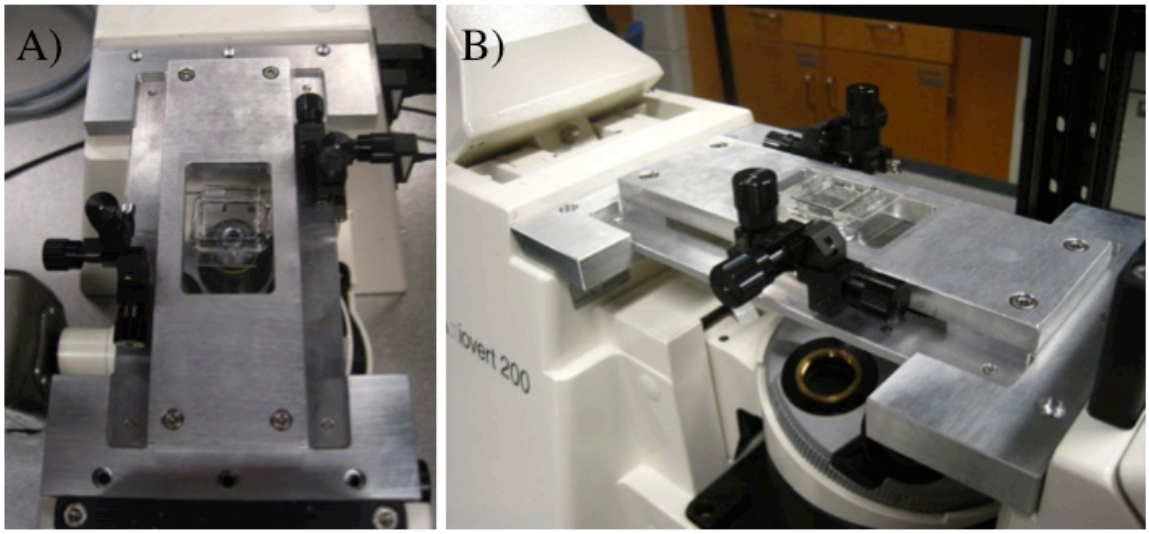


Figure 2.4: Custom fabricated micromanipulation stage mounted on Zeiss inverted microscope. Micromanipulators (black) and glass micromanipulation chamber also shown. A) Top view. B) Isotropic view.

2.1.1.6 Microscope

An inverted microscope (Carl Zeiss, Inc., Axiovert 200) is used to image APCE. The microscope is equipped with a metal oxide semiconductor (CMOS) camera (Vision Research Inc. Phantom V5.1). As described in Section 2.1.1.5, the microscope is outfitted with a custom stage for all CAM experiments.

2.1.2 Capillary Assisted Molding Technique

As discussed in Section 1.3, the dispersed phase of partially crystalline emulsions is spherical. Using the technique described in this section, the spherical droplets are made anisotropic by recrystallizing the wax network within a capillary.

First, the micromanipulation apparatus is prepared. The chamber is placed on the microscope stage and the leads of the Peltier device connected to the DC power source. The emulsion sample is then injected into the chamber using 3 mL syringes

(BD 3 mL Luer-Lok). The approximate sample volume for a single chamber is 2.5 mL. During injection, care must be taken to avoid injecting bubbles into the sample as they can obscure the view of droplets during micromanipulation. After the sample is injected, the reservoir and microcapillary are prepared. The reservoir is filled with 10 mL of ultrapure water (Millipore, resistivity >18.2 mΩ cm). The microcapillary is then inserted into the Tygon tubing connected to the reservoir. A syringe plunger is used to ensure that the microcapillary fills completely with water (air bubbles and other obstructions impede micromanipulation). Then, the microcapillary is mounted on the micromanipulator and inserted through the open sides of the chamber into the sample. Throughout this process, the reservoir is set to be at the same height as the chamber so that water is not inadvertently injected into the chamber (which dilutes the sample).

After the apparatus has been prepared, capillary assisted molding takes place. Figure 2.5 depicts the production of a partially crystalline "ball and stick" shaped particle through CAM. First, a capillary is used to capture a spherical droplet. The capillary is positioned next to a spherical droplet using the micromanipulators, and the reservoir is moved downward 3 to 11 cm. The motion induces a negative pressure difference of 290 – 1100 Pa between the droplet and the capillary tip, which draws the droplet toward the tip of the capillary. Next, the bulk of the continuous phase surrounding the droplet is heated to melt the solid crystals within the droplet. In our system, 0.9 A at 1.7 V is supplied to the Peltier device for at least 5 min to melt the crystals. While the crystals melt, the droplet is partially aspirated into the capillary by maintaining negative pressure at the capillary tip. Once the droplet is a homogeneous liquid and has been aspirated into the capillary to the desired length, power to the

Peltier device is turned off. The emulsion cools for 15 min, which initiates crystallization of the wax network in the shape of the aspirated droplet.

The anisotropic partially crystalline emulsion particle is ejected from the capillary by raising the reservoir between 15 and 22 cm, which induces positive pressure difference of 1470 - 2160 Pa between the end of the particle and the emulsion's continuous phase. Care is taken to eject the droplet slowly so that the droplet is not distorted during the process, and to avoid water from being inadvertently injected into the sample (which locally dilutes the stabilizing surfactant).

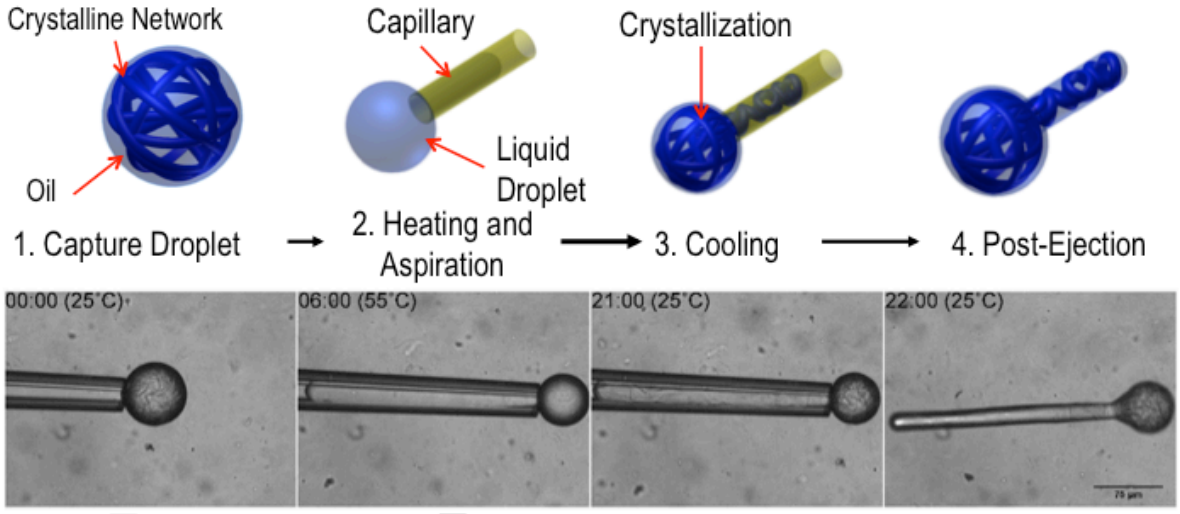


Figure 2.5: Capillary assisted molding.

2.1.3 Application of External Stimuli

Capillary assisted molding produces APCE of varying morphological stability (discussed in Section 3.2). In the cases where APCE is morphologically stable, applying external stimuli induces dynamic morphological response. The following describes how the two primary external stimuli, thermal and surfactant dilution, are

applied. Additionally, we describe the method of magnetic stimulation to APCE containing a ferrofluid.

2.1.3.1 Thermal

Raising the temperature of an emulsion containing an APCE induces morphological response by weakening the APCE internal crystalline structure. As in the CAM technique, 0.9 A at 1.7 V is supplied to the Peltier device to raise the temperature of the emulsion sample. This melts some of the crystals in the dispersed phase, thereby weakening the APCE structure. As the structure weakens, the structure can no longer resist the force that the interface applies to drives the droplet toward a spherical morphology. Typically, the 0.9A at 1.7V current is supplied to the Peltier device until the crystalline structure completely liquefies, at which point it returns to a spherical morphology.

2.1.3.2 Surfactant Dilution

Locally diluting the surfactant around an APCE induces morphological response by increasing the interfacial tension of the emulsion. This stimulus is applied by inserting a secondary microcapillary into the system. The secondary capillary is connected to a variable height reservoir (similar to the one used in CAM) filled with ultrapure water (Millipore, resistivity >18.2 mΩ cm). Using a second micromanipulator, the secondary capillary is positioned near an APCE produced via CAM. Raising the secondary reservoir injects water into the emulsion, thereby diluting the surfactant and raising the interfacial tension of droplets in the system. As the interfacial tension increases, the crystalline structure can no longer resist the interfacial force driving it to a spherical morphology. Consequently, the droplet

deforms. Typically, water is injected into the system until all of the surfactant laden microfibrous cellulose has been driven from the view frame.

2.1.3.3 Magnetic

A cylindrical magnet (1.25 cm diameter) was used to stimulate APCE particles containing ferrofluid. The magnet was manually brought near the micromanipulation chamber and the subsequent movement of the ferrofluid APCE observed using the microscope.

2.1.4 Microscopy

Standard bright field microscopy was used to image the particles. Using a metal oxide semiconductor (CMOS) camera (Vision Research Inc. Phantom V5.1), individual images were taken at 30 frames per second while videos were captured at 10 frames per second. Typical videos were 800 frames in length. All individual images and video frames were saved as 8-bit grayscale TIFFs (tagged image file formats).

2.2 Materials

The materials used in this work are discussed in this section.

2.2.1 Partially Crystalline Emulsions

Partially crystalline emulsions consist of two phases, a dispersed and a continuous phase. Here, preparation of the two phases and their combination are discussed.

2.2.1.1 Dispersed Phase

The dispersed phase consists of hexadecane (99.6%, Fischer Scientific) and wax (White Petrolatum, USP 100%, Vaseline ® by Unilever). Figure 2.6 shows an image of petrolatum dispersed in hexadecane. Petrolatum is a mixture of long chain hydrocarbons (> 25 carbons) that exists as a viscoelastic gel of waxy crystalline solids and fluids at room temperature, while hexadecane is a liquid. To homogenize the two components, they are first combined in a glass vial using a pipette. The glass vial is then immersed in a 65 °C hot water bath for at least 5 minutes. The hot bath immersion melts the petrolatum, and the hexadecane and petrolatum mix as liquids. Typical dispersed phase compositions range from 60 – 90 wt% petrolatum.

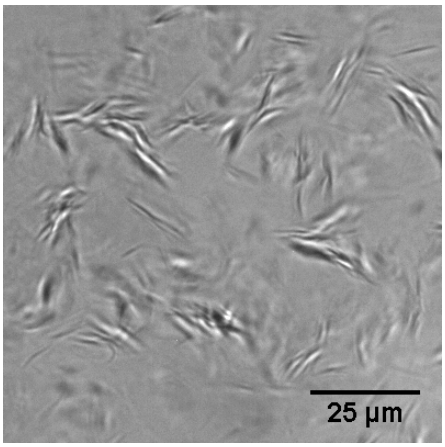


Figure 2.6: Dispersion of wax (Petrolatum) in hexadecane.

2.2.1.2 Continuous Phase

The continuous phase is an aqueous mixture of sodium dodecyl sulfate (SDS) and microfibrous cellulose. SDS (99.0%, Sigma-Aldrich) is an anionic surfactant that is frequently used to stabilize emulsions both in academic and industrial applications.

The critical micelle concentration of SDS is 8.2 mM at 25°C [35]. Microfibrous cellulose is a non-toxic additive typically used in industrial applications to thicken liquid products. In micromanipulation experiments, the MFC gives the continuous phase a small yield stress, which prevents the droplets from creaming out of the sample or moving uncontrollably during micromanipulation [31]. Based on the buoyant force acting on the dispersed phase, this yield stress is at least 0.11 Pa.

The continuous phase (0.5 wt% MFC, 10 mM SDS) is prepared by first massing ultrapure water (Millipore, resistivity >18.2 mΩ cm) into a glass beaker. The beaker is placed on a stir plate at 300 rpm, and the MFC added slowly. After all the powdered MFC has dissolved, the solution is transferred to a centrifuge tube (50 mL, BD) and the MFC is further mixed using a homogenizer (Biospect Products, Model number 985370, rotor speed 15) for 5 minutes. The solution is then centrifuged at 4000 rpm for 7 min to remove large cellulose aggregates from the solution. The supernatant (roughly 80 wt% of the original solution) is then transferred to another centrifuge tube. SDS is massed and added to the supernatant tube. The SDS/MFC solution is thoroughly vortexed, and air bubbles are allowed to cream out of the aqueous solution overnight. Typically, 50 mL of solution (prior to centrifugation) are prepared at once.

2.2.1.3 Emulsification of the Two Phases

The dispersed and continuous phases are prepared in 20/80 vol% ratio with a net volume of 10 mL. The two phases are combined and emulsified as follows. Appropriate aliquots of both phases are transferred to glass vials. Both vials are then added to a 65 °C hot water bath for at least 5 minutes. The hot bath immersion liquefies the dispersed phase and brings the continuous and dispersed phase to the

same temperature. After heating, the vials are removed from the bath and the dispersed phase is poured into the vial containing the continuous phase. This vial is inverted several times such that the dispersed phase is suspended in the continuous phase in the form of droplets. The vial is then placed under running tap water, which cools the bulk emulsion and induces crystallization in the dispersed phase. An image of an emulsion produced according to this procedure is given in Figure 2.7.

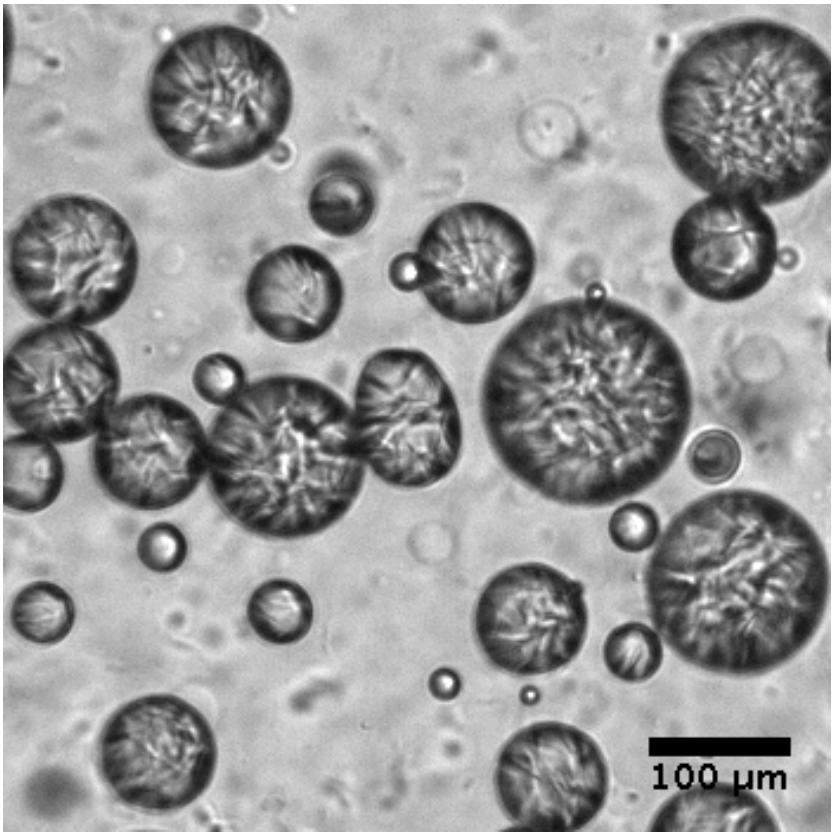


Figure 2.7: Partially crystalline emulsion. Dispersed phase is 70/30 wt% petrolatum/hexadecane.

2.2.2 Ferrofluid

An oil-based ferrofluid was added to the dispersed phase of hexadecane – petrolatum in water emulsions to determine if a magnetic stimulus could induce a morphological response in APCE. The ferrofluid (Ferrotec, EMG 905) consists of 10 nm particles suspended in light hydrocarbon oil with a density of 1.2 g/mL and viscosity of 3 cP. To combine the ferrofluid with the dispersed phase, the ferrofluid was first diluted in hexadecane (25 vol% ferrofluid) and vortexed. This ferrofluid – hexadecane mixture was then combined with petrolatum according to the method described in Section 2.2.1.1.

Chapter 3

ANISOTROPIC PARTIALY CRYSTALLINE EMULSION PARTICLES

In this chapter, we present the four different anisotropic partially crystalline emulsion particle morphologies produced to date. We additionally present qualitative characterization of morphology stability and dynamics. Chapter 4 presents a quantitative analysis of the observations described here.

3.1 Morphologies Produced

Of the four morphologies produced to date, three morphologies were produced using the capillary assisted molding technique described in Chapter 2. The forth morphology, a torus, was produced by collaborators at Procter & Gamble. We include the torus morphology here in preparation for the related mathematical analysis presented in Section 4.3.

3.1.1 Spherocylinder

Spherocylinders (Figure 3.1), also known as “rod” shaped particles, are formed by completely aspirating a droplet into a capillary during melting.

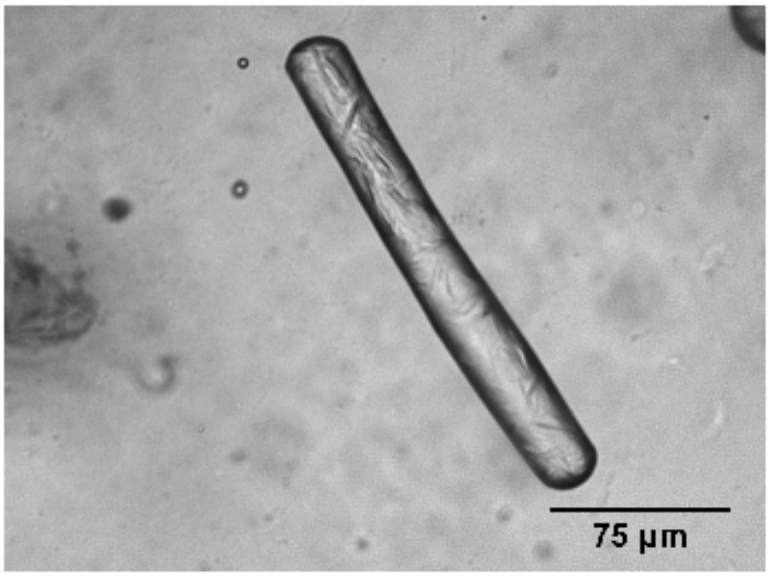


Figure 3.1: Spherocylinder anisotropic partially crystalline emulsion particle (80/20 petrolatum/hexadecane in 0.5 wt % MFC + 10 mM SDS).

3.1.2 Ball and Stick

A “ball and stick” (Figure 3.2) shaped particle is formed by partially aspirating a droplet into the capillary during molding.

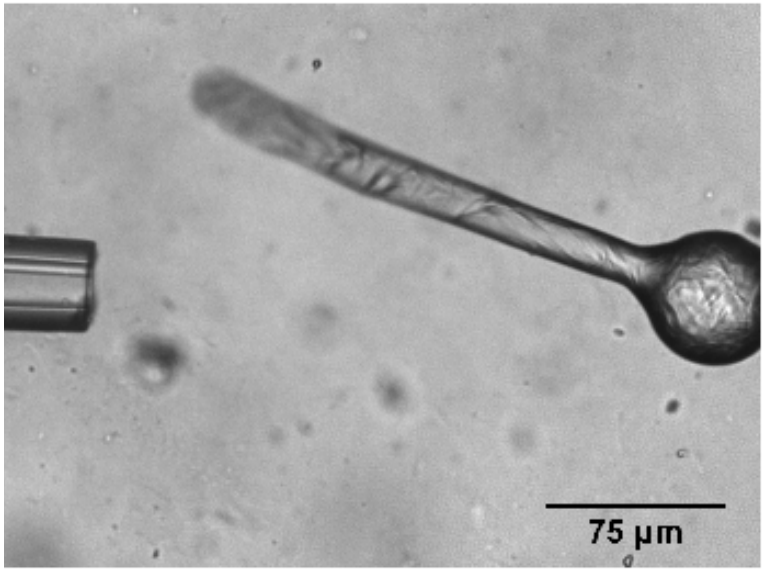


Figure 3.2: Ball and stick anisotropic partially crystalline emulsion particle (80/20 petrolatum/hexadecane in 0.5wt % MFC + 10 mM SDS).

3.1.3 Dual Ball and Stick

The "dual ball and stick" (Figure 3.3) is formed using two microcapillaries. The procedure for forming this morphology varies slightly from that described in Section 2.1.2. One capillary is first used to capture a droplet and form a singular ball and stick morphology. During melting, a secondary capillary is aligned coaxially with the primary capillary, and some of the droplet volume is aspirated in the secondary capillary. The droplet is then allowed to cool while constrained in the two capillaries. Since it is difficult to keep an entirely liquid droplet in this partially aspirated position between the two capillaries, the droplet is never completely melted during this process. As such, the crystalline structure is not assumed to be homogeneous as in the spherocylinder and singular ball and stick case.

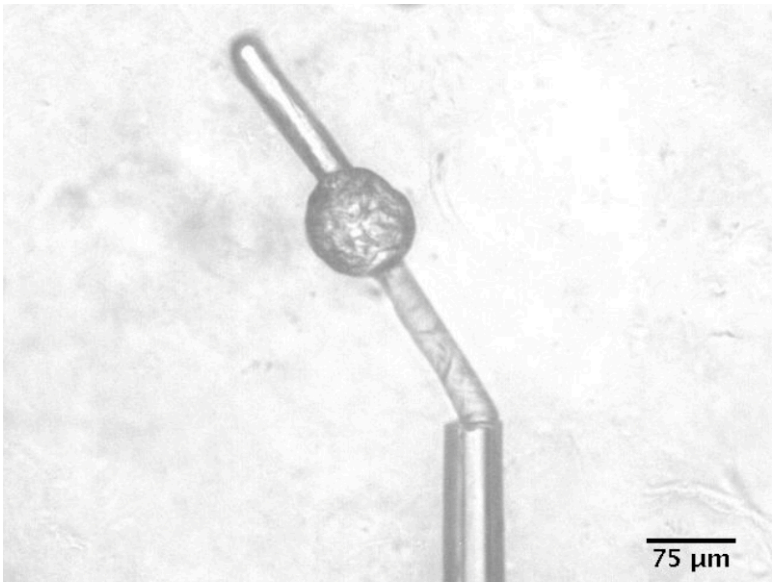


Figure 3.3: Dual ball and stick anisotropic partially crystalline emulsion particle (80/20 petrolatum/hexadecane in 0.5wt % MFC + 10 mM SDS). Note that the two spherocylinder portions are not in the same plane, which causes the upper spherocylinder portion to be out of focus.

3.1.4 Torus

A torus-shaped particle (Figure 3.4) was produced by our collaborators at Procter and Gamble. Through a variant of CAM (which we will not discuss here) was used to produce this shape, we assume that the crystalline structure within the particle is approximately homogenous for the purposes of the mathematical analysis presented in Section 4.3.

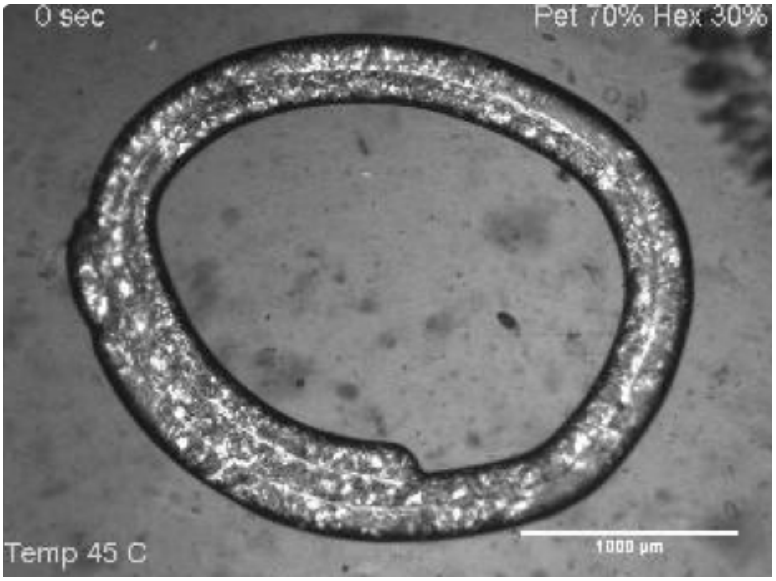


Figure 3.4: Torus anisotropic partially crystalline emulsion particle (70/30 petrolatum/hexadecane in 0.5wt % MFC + 10 mM SDS) prepared by Jessica Lewis at Procter & Gamble; image distributed by Marco Caggioni, also of Procter & Gamble.

3.2 Morphology Stability Factors

Morphological variation in APCE is characterized by distortion of the anisotropic particles from the original shape that they were cast in within the capillary mold. We theorize that this distortion is linked to opposing forces acting on the continuous-dispersed phase interface, namely compressive forces acting to reduce the

total interfacial area of the system, and resistance of the crystalline network to deformation. Distortion is observed in APCE in two principal instances: 1) APCE can distort after removal from the capillary and 2) APCE can distort when their local environment is altered in a way that affects the crystalline network or continuous-dispersed phase interface. The first distortion mechanism is discussed in this section. Section 3.3 describes the second distortion mechanism.

We have observed three degrees of morphological stability in axisymmetric APCE after capillary ejection: 1) Instantaneous instability, 2) long term instability and 3) long term stability. These three stability types are documented for three spherocylinder particles in Figures 3.5, 3.6, and 3.7 respectively. For both types of instability, APCE deform from their high aspect ratio morphologies to morphologies that have lower surface area to volume ratios. This transition is indicative of the drive to reduce the interfacial energy of the system. For the stable case, the strength of the internal crystalline network is sufficient to resist this drive. Note that throughout this work, a particle is considered statically stable when the particle retains its morphology for at least 30 minutes after ejection from the capillary.

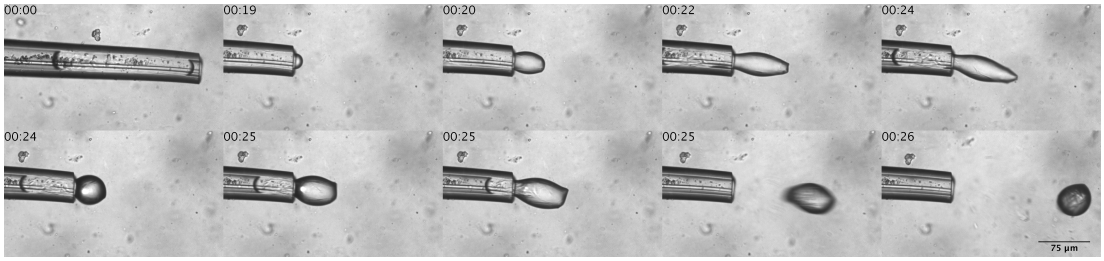


Figure 3.5: Spherocylinder APCE that deforms upon ejection (instantaneous instability). APCE is 60/40 wt% petrolatum/hexadecane.

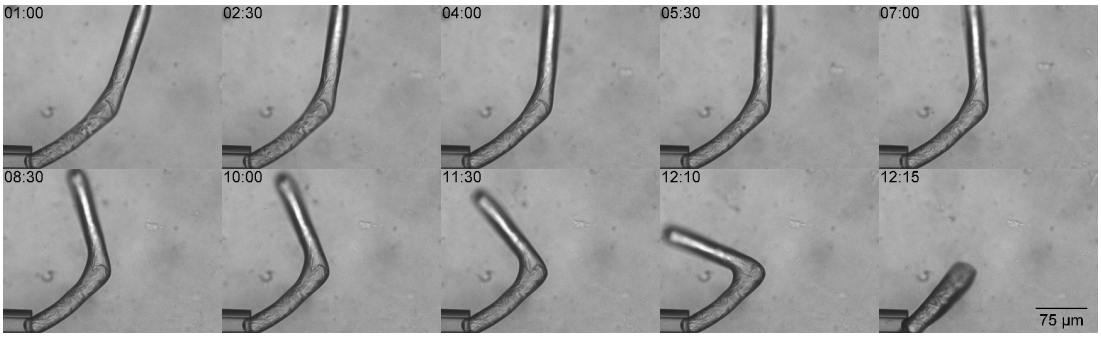


Figure 3.6: Spherocylinder APCE that deforms 12 minutes and 15 seconds after ejection (long term instability). APCE is 80/20 wt% petrolatum/hexadecane. Droplet ejection began at 00:00 min.

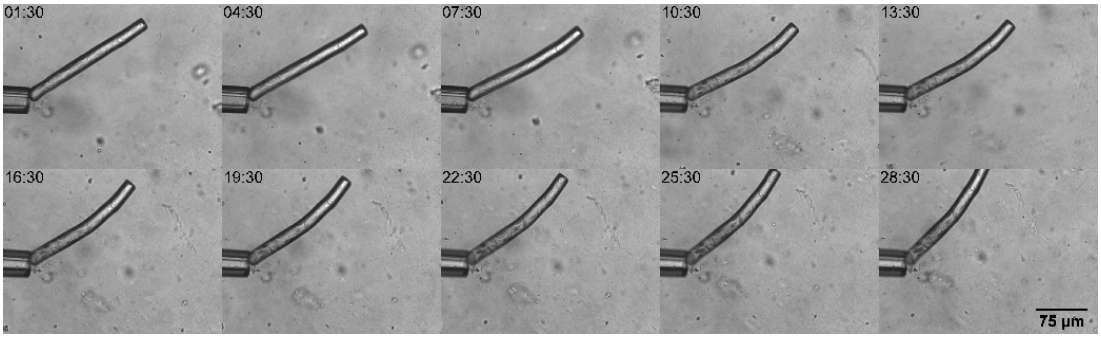


Figure 3.7: Spherocylinder APCE that approximately maintains its aspect ratio after ejection (long term stability). Deformation was not observed over 10 – 28 minutes; as such, the APCE was not imaged beyond 28 min. after ejection. APCE is 100 wt% petrolatum/hexadecane. Droplet ejection began at 00:00 min.

There are three principal factors that affect APCE stability. The first of these is droplet solid content. Holding temperature constant, the yield stress of the crystalline network increases with solid content (petrolatum composition). Consequently, droplets with higher solid content exhibit higher degrees of stability than those with lower solid content. This effect is illustrated in Figures 3.5 and 3.6, where the spherocylinder that is 60 wt% petrolatum immediately deforms while the 80

wt% petrolatum spherocylinder deforms over the course of 12 minutes and 15 seconds. Furthermore, the two droplets in Figures 3.5 and 3.6 do not assume the same deformed morphology. The 60 wt% petrolatum APCE attains nearly a perfect sphere, while the 80 wt% APCE attains a spherocylinder-like morphology that has a lower aspect ratio and interfacial area.

The second factor that influences APCE stability is the interfacial tension. As discussed in Chapter 4, the Laplace pressure exerted by the emulsion interface is proportional to the interfacial tension. Consequently, in systems with high interfacial tension (i.e. systems with low surfactant concentration), the compressive force acting on the internal crystalline network is higher, and APCE will deform to a spherical morphology more readily.

The third factor that influences APCE stability is the interfacial curvature. As will be discussed in Chapter 4, the Laplace pressure difference across the interface is proportional to the mean curvature of the interface. Anisotropic partially crystalline emulsion particles have interfaces that have variations in their curvature. At the ends of spherocylinder and ball and stick particles (parallel to the central axis), the interfacial curvature is high and positive, indicating that the interface imposes a large, inwardly directed local force on the crystalline structure. On the sides of the spherocylinder (perpendicular to the central axis), the interfacial curvature is positive but lower, indicating that the interface imposes a lower, inwardly directed local force on the crystalline structure. Within the ball and stick particles, at the connection between the sphere and the spherocylinder, there can even be a negative interfacial curvature. Here, the local force imposed by the interface is directed outward toward the continuous phase. These differences in interfacial curvature across the APCE

affect the location where deformation initiates. Specifically, ball and stick particles are predisposed to deform at the connection between the sphere and the spherocylinder. Fluid moves toward this region because there is a lower force exerted by the continuous phase. For the same reason, imperfect spherocylinder shaped particles that contain bends or kinks (which often form when an APCE is ejected too quickly during CAM) are predisposed to deform at the location of the bend. Figure 3.8 illustrates both of these phenomena.

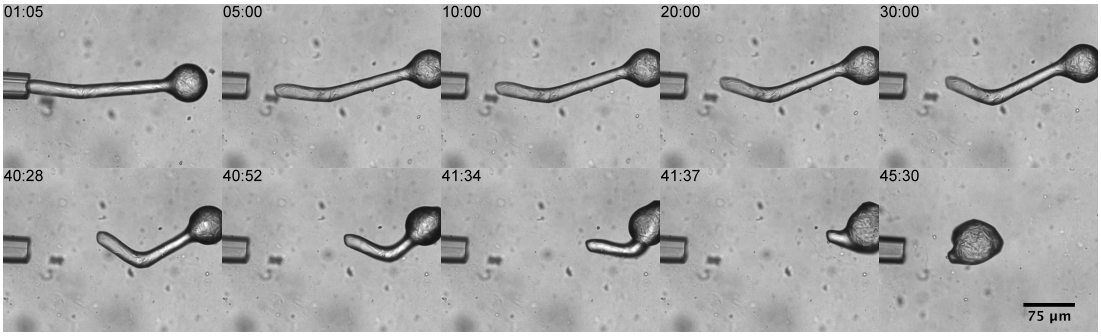


Figure 3.8: Long term instability in ball and stick APCE. Deformation initiates in two regions of low interfacial curvature: at the connection between the sphere and spherocylinder, and at the bend in the imperfect spherocylinder.

In addition to affecting the location where deformation initiates, the interfacial curvature affects the relative stability of the particles. APCE with lower characteristic dimensions (i.e. lower spherocylinder/ball and stick radius) have higher interfacial curvature across the entire interface. Since the Laplace pressure is proportional to the mean curvature, APCE that are thinner will deform more readily than those that are thicker, since the net compressive stress acting on the particle is higher.

3.3 Particle Response to External Stimuli

In this section, we discuss the second deformation mechanism, where an APCE's local environment is altered in a way that affects the crystalline network or continuous-dispersed phase interface.

3.3.1 Thermal

Changing the temperature of the continuous phase affects the crystalline content of the APCE. Increasing the temperature of the continuous phase through conductive heating decreases the fraction of wax crystallized. Since the yield stress of the network within each APCE is a direct function of the amount of wax crystallized, the yield stress decreases when heating is applied. If the continuous phase around a stable APCE is sufficiently heated the internal network begins to melt, the network may no longer be strong enough to resist the compressive force applied by the interface. Subsequently, the APCE deforms to a morphology with a lower interfacial area. This phenomenon is depicted in Figure 3.9.

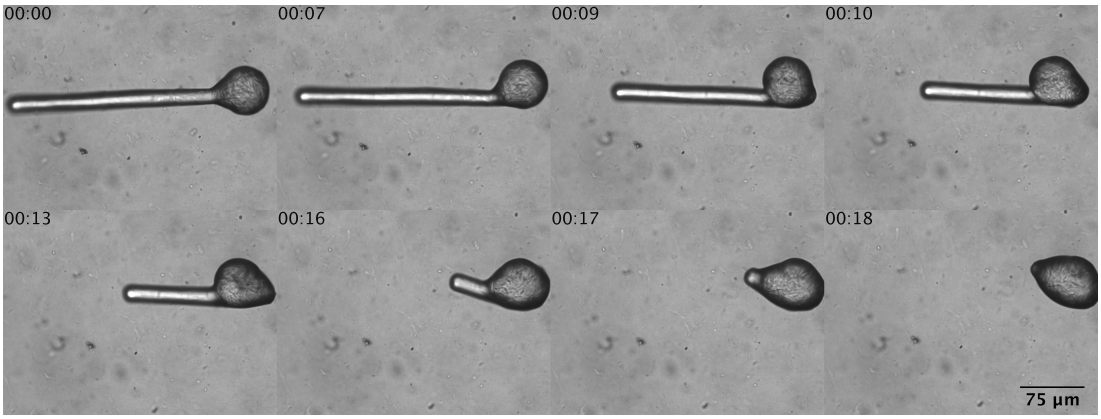


Figure 3.9: APCE ball and stick response to heating. As the bulk temperature of the emulsion rises, the crystalline network begins to melt. The reduced yield stress of the internal network results in morphological instability. The APCE deforms to a nearly spherical morphology via a rolling motion.

As in the singular ball and stick example, applying thermal stimulus to dual ball and stick shaped particles initiates rolling at the junction between the sphere and one of the spherocylinders. However, unlike the singular ball and stick example, the other spherocylinder feature maintains its shape throughout deformation (Figure 3.10). The two spherocylinders likely did not deform simultaneously because, as discussed in Section 3.1.3, the crystalline network was not completely homogenous throughout the droplet. Nevertheless, the retention of the secondary spherocylinder indicates that features of APCE may remain stable and intact while moving throughout the continuous phase. This is particularly important when considering the inevitable agitation and movement of APCE in emulsion products.

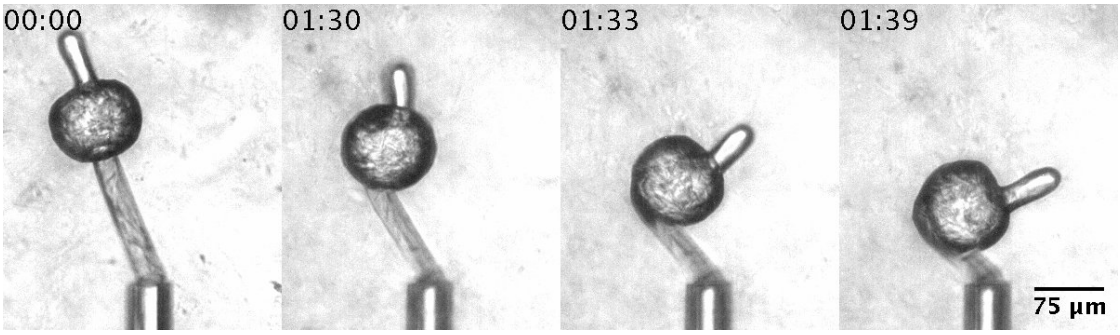


Figure 3.10: APCE dual ball and stick response to heating. One of the spherocylinder features retains its shape throughout the deformation. Note that the particle features and capillary are not in focus simultaneously since they are not all on the same plane.

For the torus APCE morphology, deformation induced by thermal stimuli occurs via a different mechanism than that observed for axisymmetric APCE. Since the torus particle is continuous, it cannot roll or buckle like the axisymmetric particles do. Instead, the torus thins in isolated regions, breaks, and then becomes spherical (as seen in Figure 3.11).

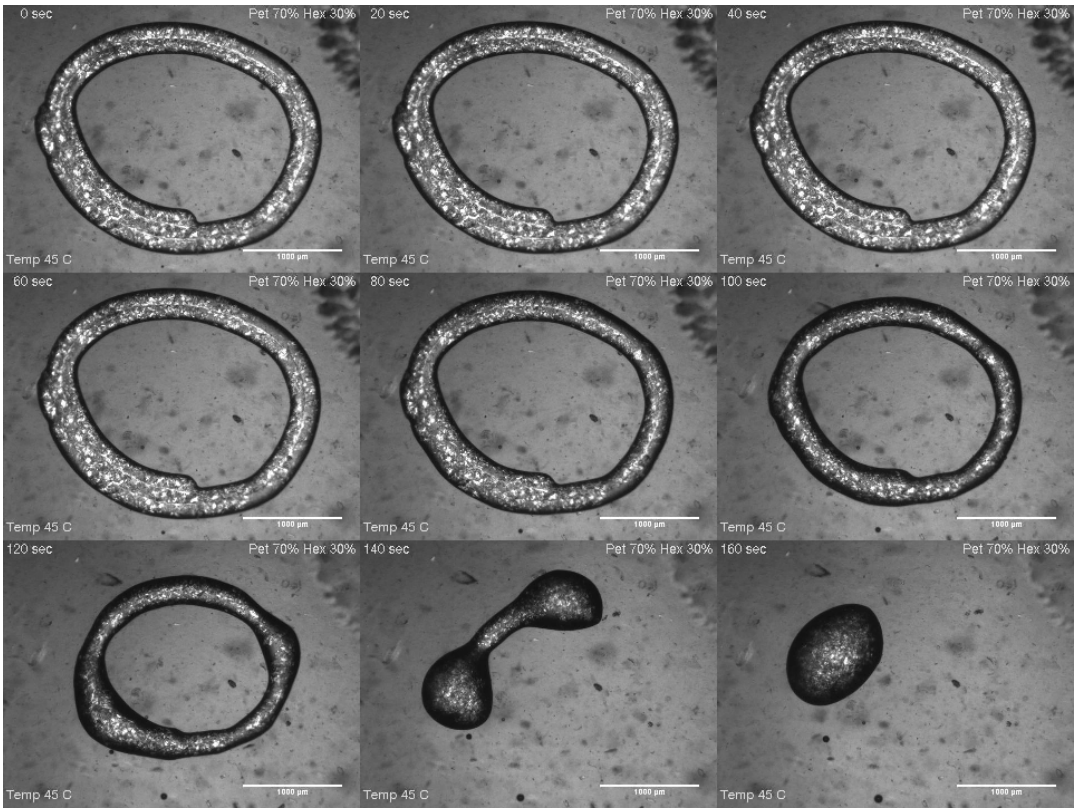


Figure 3.11: APCE torus response to heating. Experiment performed by Jessica Lenis; image distributed by Marco Caggioni, both of Procter & Gamble.

3.3.2 Dilution

Changing the local surfactant concentration of the continuous phase affects the forces acting on APCE interface. The local surfactant concentration in a surfactant-laden emulsion can be reduced through local dilution, as described in Section 2.1.3.2. Reducing the local surfactant concentration increases the interfacial tension and compressive force exerted by the APCE interface. If the interfacial tension is sufficiently increased, the network may no longer be strong enough to resist the compressive force applied by the interface. Subsequently, the APCE deforms to a morphology that has lower interfacial area. This phenomenon is depicted in Figure

3.12. In Figure 3.12, dilution is visible as the fibrous MFC moves out of the view frame, and the continuous phase becomes transparent.

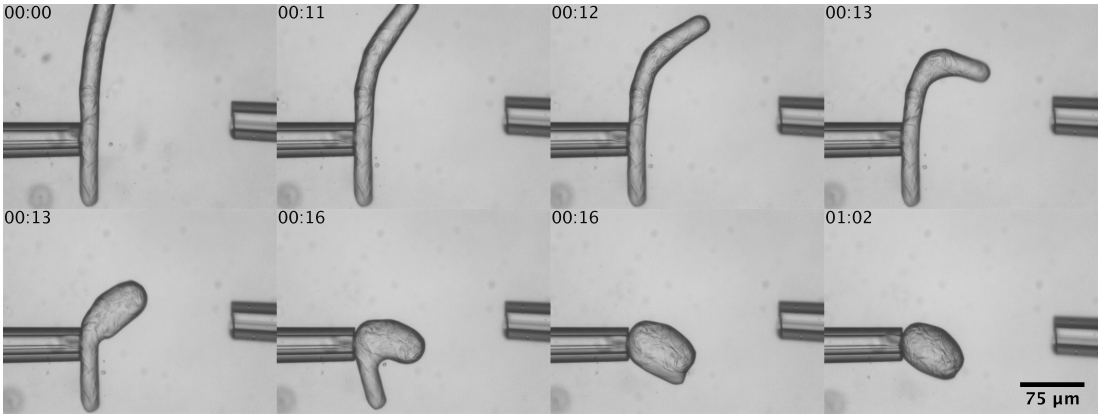


Figure 3.12: APCE spherocylinder response to local dilution (injection of water via capillary on the right). As local surfactant concentration decreases, the Laplace pressure increases. The increased compressive stress on the network results in morphological instability. The APCE deforms to a nearly spherical morphology via a buckling motion.

3.3.3 Magnetic

Unlike the response of thermal and dilution stimuli, APCE response to a magnetic stimulus does not result in dramatic morphology variation. Instead, when a magnetic field is introduced around a ferrofluid laden droplet, the droplet moves while maintaining its original morphology. This subtle movement is illustrated in Figure 3.13 for a pair of spherical droplet.

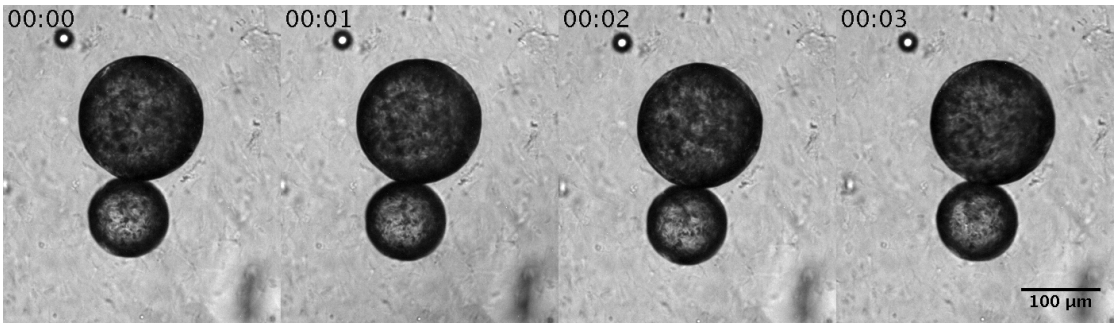


Figure 3.13: Rotation of ferrofluid laden spherical partially crystalline emulsion droplets during the application of a magnetic field. Comparing the location of dark, ferrofluid aggregates on the surface of each sphere between frames demonstrates that the droplets do move in response to magnetic stimuli; however, the droplets retain their original morphology.

Figure 3.14 shows the response of a ball and stick APCE laden with ferrofluid to magnetic stimulus. The dark ferrofluid is visibly concentrated at the spherical portion of the droplet in Figure 3.14. During the CAM process, the ferrofluid concentrates in this region because it is not aspirated into the capillary along with the rest of the fluid. Since the ferrofluid is not completely miscible with the hexadecane, it does not disperse uniformly in the hexadecane-petrolatum mixture. Instead, the ferrofluid migrates to the interface between the aqueous phase and the hydrophobic hexadecane-petrolatum mixture. Consequently, when the dispersed phase is aspirated into the capillary, the ferrofluid remains as a shell around the hexadecane-petrolatum outside of the capillary. Movement of the concentrated ferrofluid shell, rather than the entire droplet, makes the motion of the APCE droplet to magnetic stimulus more apparent than in the spherical example.

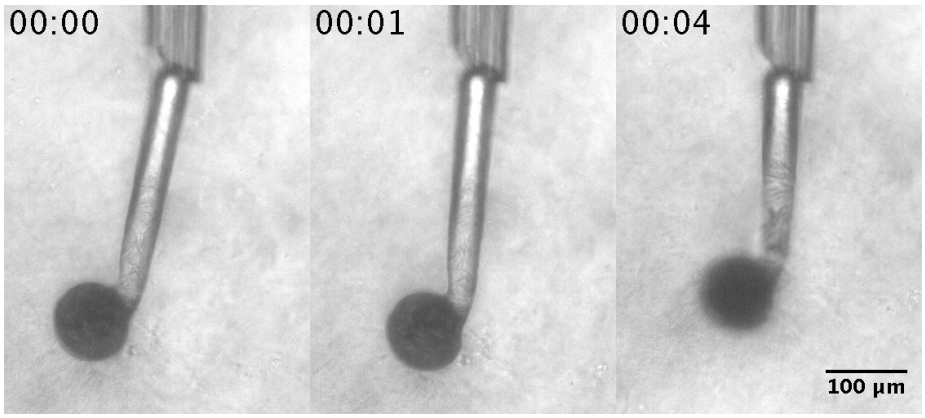


Figure 3.14: Movement of ferrofluid laden ball and stick APCE during the application of a magnetic field. The APCE morphology remains stable through the motion. At 00:04 seconds, the magnetic field draws the APCE out of the microscope's focal plane.

3.4 Substrate Deposition

The discovery that APCE morphology was responsive to thermal and dilution stimuli prompted us to consider a new potential functionality for APCE: controlled delivery of active ingredients in emulsion products.

Imagine a product that contains spherocylinder or ball and stick shaped APCE. When this product is applied to the target substrate of the product, say a network of fibers, the APCE collide with fibers with a higher frequency than spherical droplets. On its own, the increase in collision frequency should theoretically increase the amount of active ingredient transferred to the fibrous substrate. Depending on the collision, however, the entire APCE may not contact the fiber. Subsequently, the additional benefit gained from increased interfacial area during mass transfer in an individual APCE is lost. However, if the APCE is engineered so that it deforms around the fiber, the benefit associated with increased interfacial area is regained. Such deformation can be prompted by application of an external stimulus.

Specifically, when the emulsion product is diluted, an APCE droplet deforms to a more spherical morphology due to the increase in interfacial tension. If the APCE is in contact with the fiber prior to dilution, the fiber may impede this deformation, and the APCE will deform around the fiber. Figure 3.15 gives an idealized illustration of this mechanism.

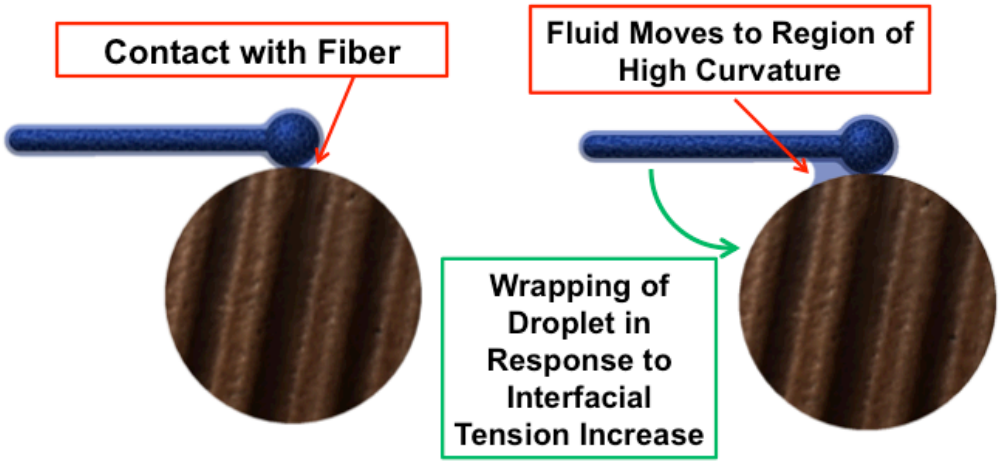


Figure 3.15: Idealized schematic of substrate deposition mechanism.

In an effort to explore this potential, we conducted a proof of concept deposition test. First, a spherical droplet (Figure 3.16) was made into an APCE through CAM. This APCE was then contacted with a hydrophobic Teflon fiber secured within the micromanipulation chamber (Figure 3.17).

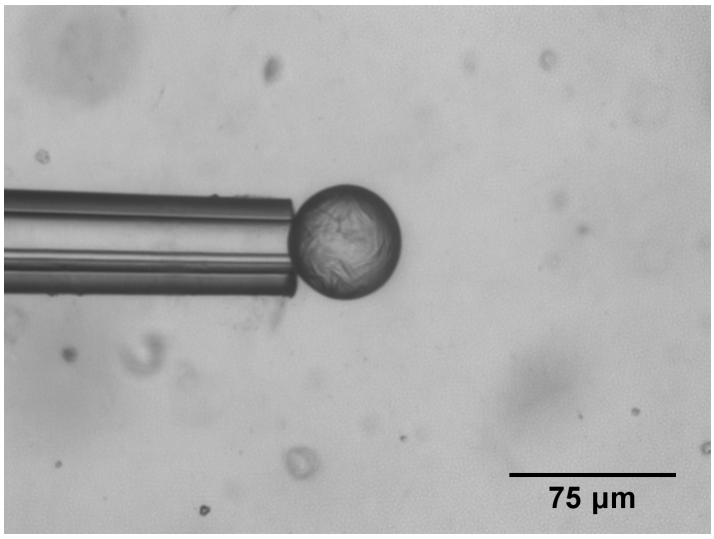


Figure 3.16: Original spherical droplet in substrate deposition test.

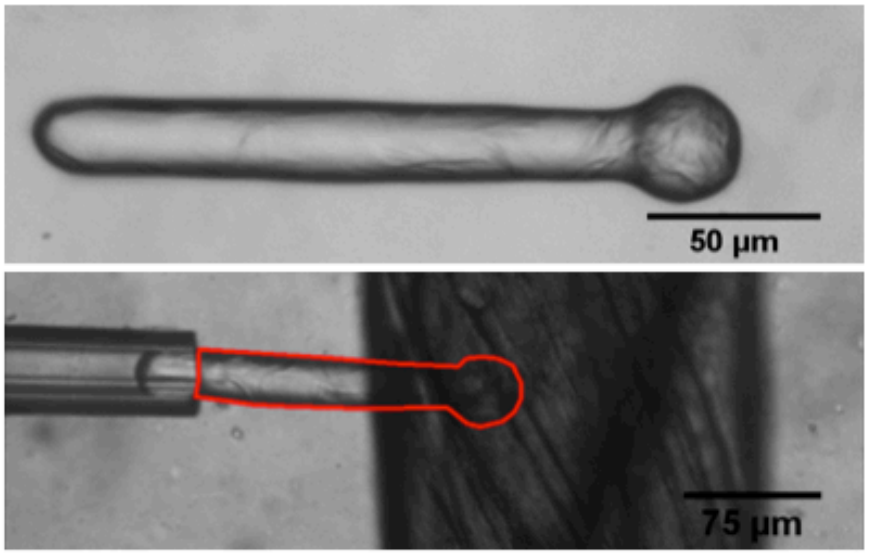


Figure 3.17: (Top) Anisotropic ball and stick partially crystalline emulsion particle (Bottom) The APCE contacted to the hydrophobic Teflon fiber using micromanipulation. The APCE interface is outlined in red to aid in visualization.

Following the initial contact and adhesion to the fiber, the APCE was removed from the capillary by applying a small positive pressure at the end of the capillary (Figure 3.18 between 0 and 7 seconds). Then, the surfactant was locally diluted by injecting water into the reservoir into the chamber around the fiber (in Figure 3.18, this can be seen as the fibrous MFC moves out of the view frame, and the continuous phase becomes transparent). Figure 3.19 is a contrast enhanced imaged of the final frame of Figure 3.18, which illustrates the complete wrapping of the droplet around the fiber.

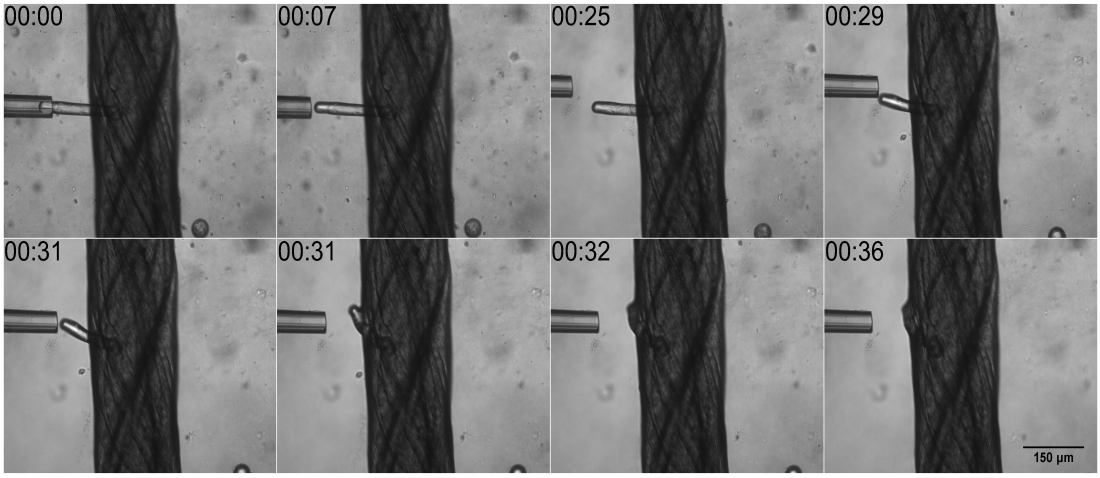


Figure 3.18: Directed deposition of ball and stick APCE onto Teflon fiber. APCE deformation was induced by locally diluting the surfactant.

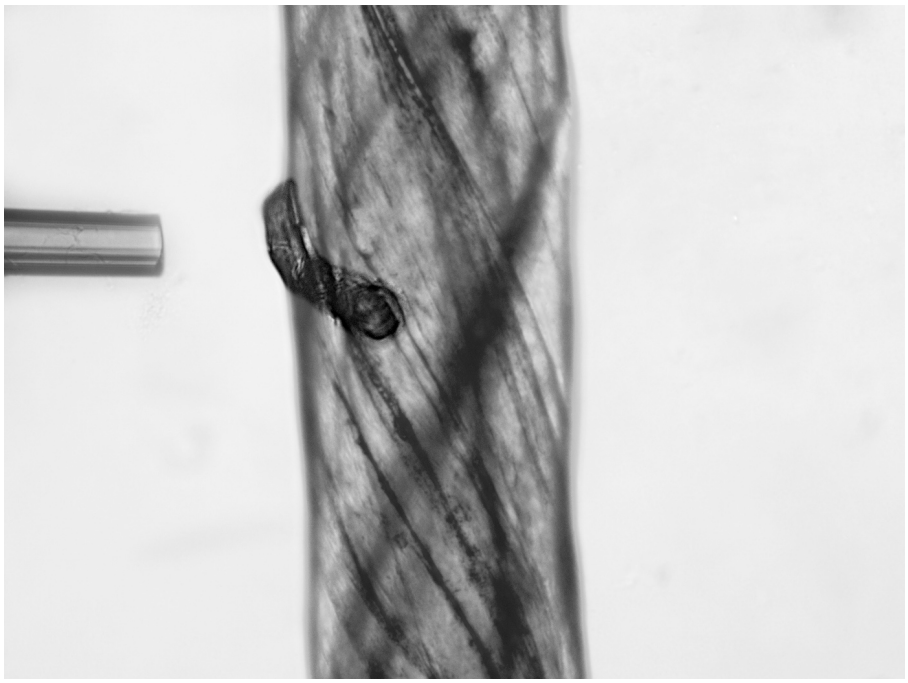


Figure 3.19: Contrast enhanced image of the ball and stick APCE wrapped around a Teflon fiber after local surfactant dilution.

3.5 Summary

Four anisotropic partially crystalline emulsion particle morphologies have been produced to date: 1) the spherocylinder, 2) the ball and stick, 3) the dual ball and stick and 4) the torus. Successful production of these morphologies represents significant progress to meeting the first goal of this work defined in Section 1.1. Morphological variation in these APCE is characterized by distortion of the anisotropic particles from the original shape that particles were cast in within the capillary. We hypothesize that this distortion is linked to opposing forces acting on the continuous-dispersed phase interface, namely compressive forces acting to reduce the total interfacial area of the system, and resistance of the crystalline network to deformation. Distortion has been observed the APCE in two principal instances: 1) APCE distort after ejection from the

capillary and 2) APCE distort when their local environment is altered in a way that affects the crystalline network or continuous-dispersed phase interface. In the first distortion mechanism, the time scale of distortion is influenced by three principal parameters: the crystalline solid content, the interfacial tension, and the interfacial curvature of the particle. In the second distortion mechanism, application of heating and local dilution causes the particles to collapse to morphologies with smaller interfacial areas (nearly spherical). Additionally, creating magnetic fields in ferrofluid laden APCE causes the droplets to move while retaining their morphology. The application potential of morphology response to external stimuli was successfully explored in a proof of concept substrate deposition test. While this proof of concept test illustrated the potential benefits of using APCE in emulsion products, quantitative understanding of morphology dynamics in this system was still poorly understood. As such, we defined the second goal of this work, and developed analytical models for APCE morphologies, all of which are described in Chapter 4.

Chapter 4

MODELING ANISOTROPIC PARTCLE STABILITY

The second goal of this work is to develop a quantitative understanding of morphology deformation in anisotropic partially crystalline emulsion particles. This understanding is crucial to the successful incorporation of anisotropic partially crystalline emulsion particles into emulsion products. Accurate descriptions of APCE stability will provide product engineers with limits on the temperature, formulation tolerances and processing parameters that each product can be subjected to before undergoing unintended deformation. In products that exploit the morphological response of APCE to external stimuli, quantitative descriptions will help define how much stimulus needs to be applied to induce response.

Explicit models of morphological stability in anisotropic partially crystalline emulsion particles did not exist at the outset of this work since the APCE system itself was recently created. However, by integrating fundamental, proven theories of emulsion, interface, stress and yielding phenomena, we were able to gain some understanding of the interplay between parameters in the APCE system. This chapter discusses three analytical models developed to describe the morphological stability of APCE. These models, developed through a mechanical, force balance approach, provide predictions of static APCE stability. A particle is considered statically stable when the particle retains its morphology for at least 30 minutes after ejection from the capillary holding all formulation (crystalline content) and external (temperature, surfactant, etc.) parameters constant.

Predicting dynamic morphological evolution during processes like heating is a problem with inherently greater complexity. Not only must one account for the balance between interfacial forces and crystalline network strength, but also for the shear forces imposed on the system throughout movement. With these added complexities, we did not attempt to model morphology dynamics in the scope of this thesis. However, development of dynamic models may also play a crucial role in emulsion product development. By attempting to develop static models that describe qualitative features of APCE deformation here, we hope to lay the groundwork for future studies that include dynamic analysis.

The three models presented in this chapter describe different APCE morphologies. Each model presentation is organized as follows. First, the morphologies covered by the model are listed. Second, the modeling objectives are defined. Agreement with the qualitative trends of morphological stability constitutes the core of the modeling objectives. Next, each model is derived from fundamental Laplace pressure, differential geometry and yield criteria laws. Finally, the predictions made by each analytical model for idealized particles are discussed.

4.1 Spherocylinder Model

The first model describes the stability of spherocylinder particles (Section 3.1.1).

4.1.1 Modeling Objectives

Empirically, we found that the stability of spherocylinders increased with:

- Increasing solid content (both a function of the original petrolatum composition and the emulsion temperature)

- Increasing surfactant concentration
- Increasing spherocylinder radius

The model described here seeks to explain each of these qualitative trends.

4.1.2 Model

The pressure difference between two fluids separated by an interface is a function of the interfacial tension and the shape of the interface. When the two fluids are static, the interface between them is approximated as infinitesimally thin, and the interfacial tension is constant, the Young Laplace equation describes the pressure difference between the fluids

$$\Delta P = \mathrm{P}_1 - \mathrm{P}_2 = 2\gamma H = \gamma\left(\frac{1}{R_1} + \frac{1}{R_2}\right) \qquad 4.1$$

where γ is the interfacial tension, H is the mean curvature of the interface, and R_1 and R_2 are the first and second principle radii of curvature of the interface.

For the purpose of relating physical properties of the APCE system to mathematical description, let us define fluid 1 as the hydrophobic, dispersed phase of an APCE, and fluid 2 as the hydrophilic, continuous phase surrounding the APCE as shown in Figure 4.1.

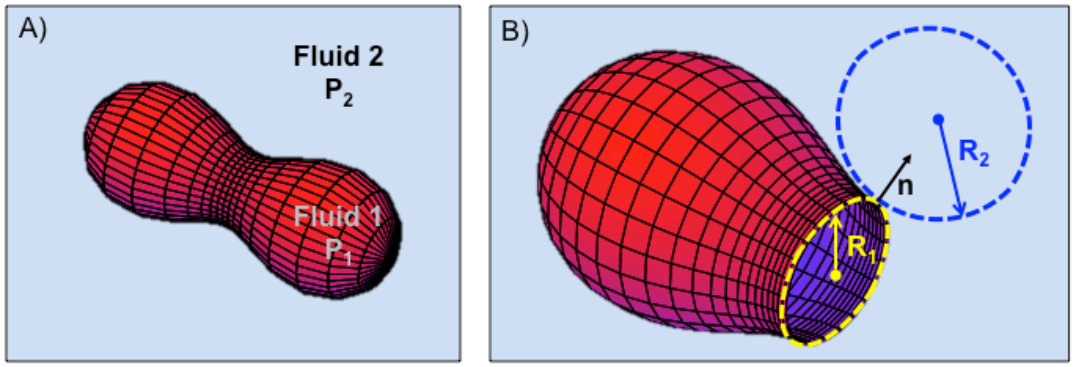


Figure 4.1: Defining notation in APCE system. A) For any anisotropic PCE particle, fluid 1 at P_1 is the hydrophobic continuous phase; fluid 2 at P_2 is hydrophilic dispersed phase. B) Cross section of APCE. $\hat{\boldsymbol{n}}$ is the outward facing normal, R_1 and R_2 are the principal radii of curvature.

If the mean curvature of the interface is positive, the pressure of the dispersed phase is greater than that of the continuous phase. This is because the interface exerts a compressive force on the dispersed phase as the interfacial area minimizes. In the APCE system, the force exerted by the interface is exerted on the semi-crystalline structure, rather than on the fluid alone. Since the shape of the interface varies with position along the anisotropic particle, the local force also varies.

To elucidate the difference between forces along the shape, we introduce a set of mathematical parameters to describe the APCE interface. We distinguish between vector and scalar quantities by bolding vectors. First, let us define the interface of an APCE as the surface S in $\Re^3$ parameterized by two variables, u and v. Let $\boldsymbol{r}$ denote the radius vector from the origin of the Cartesian coordinate system (x, y, z) with the unit vectors $(\boldsymbol{i}, \boldsymbol{j}, \boldsymbol{k})$ [36]. The surface S is defined by the vector equation

$$\boldsymbol{r} = f(u,v) = \phi(u,v)\boldsymbol{i} + \psi(u,v)\boldsymbol{j} + \chi(u,v)\boldsymbol{k} \qquad 4.2$$

or in the parameter form

$$x = \phi(u,v) \quad y = \psi(u,v) \quad z = \chi(u,v) \tag{4.3}$$

Let $\hat{\boldsymbol{n}}$ denote the outward facing unit normal of the surface *S*, defined by

$$\hat{\boldsymbol{n}} = \frac{\boldsymbol{r}_u \times \boldsymbol{r}_v}{|\boldsymbol{r}_u \times \boldsymbol{r}_v|} \tag{4.4}$$

where the subscript on the surface vector $\boldsymbol{r}$ denotes first derivative with respect to the independent parameter (Eqn 4.5),

$$\boldsymbol{r}_u = \frac{\delta \boldsymbol{r}}{\delta u} \tag{4.5}$$

In the APCE system, a vector that is "outward facing from the surface" points towards the continuous, hydrophobic phase.

The mean curvature of such a surface *S* is given by [36]

$$H = -\nabla \cdot \hat{\boldsymbol{n}} = \frac{LG - 2MF + NE}{2(EG - F^2)} \tag{4.6}$$

where

$$E = \boldsymbol{r}_u \cdot \boldsymbol{r}_u \tag{4.7a}$$

$$F = \boldsymbol{r}_u \cdot \boldsymbol{r}_v \tag{4.7b}$$

$$G = \boldsymbol{r}_v \cdot \boldsymbol{r}_v \tag{4.7c}$$

$$L = \boldsymbol{r}_{uu} \cdot \hat{\boldsymbol{n}} \tag{4.7d}$$

$$M = \boldsymbol{r}_{uv} \cdot \hat{\boldsymbol{n}} \tag{4.7e}$$

$$N = \boldsymbol{r}_{vv} \cdot \hat{\boldsymbol{n}} \tag{4.7f}$$

In the previous definitions, multiple subscripts of $\boldsymbol{r}$ denote mixed partial derivatives with respect to *u* and *v*:

$$\boldsymbol{r}_{uv} = \frac{\delta}{\delta v}\left(\frac{\delta \boldsymbol{r}}{\delta u}\right) \tag{4.8}$$

Readers familiar with differential geometry convention will recognize *E*, *F* and *G* as the coefficients of the first fundamental form of a surface and *L*, *M* and *N* as the

coefficients of the second fundamental form of a surface. However, understanding this distinction is not necessary to follow the remainder of the APCE model development.

With these definitions in place, we express the local force normalized by a differential area by the surface traction, $\boldsymbol{t}$, which is sometimes called the stress vector:

$$\boldsymbol{t}(u,v) = \frac{F}{\delta A} = -2\gamma H(u,v)\hat{\boldsymbol{n}}(u,v) \tag{4.9}$$

where $\boldsymbol{F}$ is the force generated by interfacial tension and A is a differential area element oriented in the same direction as $\hat{\boldsymbol{n}}$. As denoted by the presence of u and v in Equation 4.9, the surface traction, the mean curvature and the unit normal are all dependent on the position along the APCE interface. The dependence of $\hat{\boldsymbol{n}}$ on position is simple to visualize: $\hat{\boldsymbol{n}}$ is always perpendicular to the surface and pointed outwards toward the continuous phase. Since the surface traction, $\boldsymbol{t}$, is parallel to $\hat{\boldsymbol{n}}$, $\boldsymbol{t}$ is also always perpendicular to the surface. Whether $\boldsymbol{t}$ points towards the continuous or dispersed phase depends on the sign of H.

The magnitude of $\boldsymbol{t}$ is defined by the position dependent mean curvature H. The mean curvature of a spherocylinder differs in two places. At the hemispherical ends of the particle, mean curvature is

$$H_{hemisphere} = \frac{\mathbf{1}}{\boldsymbol{R}} \tag{4.10}$$

where R is the radius of the spherocylinder. Along the sides of the spherocylinder, the mean curvature is

$$H_{side} = \frac{\mathbf{1}}{\mathbf{2}\boldsymbol{R}} \tag{4.11}$$

Equations 4.10 and 4.11 are determined by first defining the surface $S_{spherocylinder}$ as a piecewise function that is parametric in u and v:

$$S_{spherocylinder}: \quad \mathbf{r}(u,v) = f(u)\cdot\cos(v)\,\boldsymbol{i} + u\cdot\boldsymbol{j} + f(u)\cdot\sin(v)\boldsymbol{k} \quad 4.12a$$

$$f(u) = \begin{cases} \sqrt{R^2-(u-R)^2} & 0 \le u < R \\ R & R \le u < R+L \\ \sqrt{R^2-\left(u-(R+L)\right)^2} & R+L \le u < 2R+L \end{cases} \quad 4.12b$$

where L is the length of the spherocylinder. Applying Equations 4.6 and 4.7 to Equation 4.12 gives Equations 4.10 and 4.11. The result of this derivation, which is the spatial distribution of traction forces for a spherocylinder, is illustrated in Figure 4.2b.

With the interfacial forces acting on spherocylinder defined, we seek to determine the how these forces stress the semi-crystalline network within the droplet. Here, we examine stress in the droplet using a simplified approach. The surface traction acting on an APCE droplet is the largest in magnitude at the hemispherical spherocylinder ends. As such, the traction forces acting parallel to the central axis collectively exert a compressive stress, σ_c, along the central axis. This compressive stress is illustrated in Figure 4.2c, and given by

$$\sigma_c = \frac{\boldsymbol{2\gamma}}{\boldsymbol{R}} \quad 4.13$$

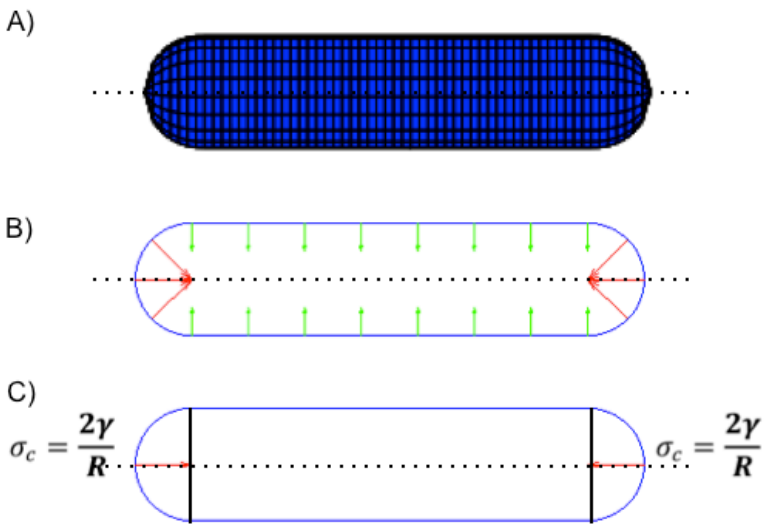


Figure 4.2: Interfacial forces acting on a spherocylinder of radius R. A) Three dimensional spherocylinder. B) Cross section of spherocylinder with surface traction displayed as vectors. Green vectors have a magnitude of 1/2R; red vectors have a magnitude of 1/R. C) Compressive stress along spherocylinder axis as defined by the spherocylinder model.

A material subjected to a single, uniaxial force will undergo plastic deformation when the compressive stress due to that force exceeds the critical uniaxial yield strength, Y_{1D}. For spherocylindrical particles, when σ_c as defined in Equation 4.13 exceeds Y_{1D} of the crystalline network, the APCE deforms. The equivalent stability expression for this is given by

$$Stable \rightarrow \quad \sigma_c < \mathrm{Y}_{1D} \tag{4.14a}$$

$$Stable \rightarrow \quad \frac{2\gamma}{\mathrm{Y}_{1D}R} < 1 \tag{4.14b}$$

The stability criterion given by Eqn 4.14b supports the qualitative observations discussed in Section 4.1.1. The yield stress is proportional to solid content; increasing the solid content increases the likelihood a particle will be stable. The interfacial

tension decreases with increasing surfactant concentration; consequently, increasing surfactant concentration increases the likelihood a particle will be stable. Finally, increasing the radius reduces the magnitude of the left hand side of equation 4.14b thereby increasing the likelihood a particle will be stable.

Agreement between the qualitative observations and the predictions by the model suggest that this simple model could be utilized practically to determine when spherocylinders deform. However, since the model relies on symmetric arguments to isolate surface tractions parallel to the central axis as dominating compressive forces, the model does not lend itself to morphologies that are antisymmetric across a droplet's semi-minor axis, as in the ball and stick type particles. In the development of the next model, we seek to develop an all-encompassing model for axisymmetric APCE that considers all of the forces exerted by the APCE interface.

4.2 Axisymmetric Model

The second model describes more generally the stability of axisymmetric particles. These include the spherocylinder, the ball and stick and the dual ball and stick described in Sections 3.1.1, 3.1.2, and 3.1.3 respectively. Additionally, this model should also describe other theoretical axisymmetric particles that have not yet been produced experimentally, like an ellipsoidal or Cassinian oval-shaped particle.

4.2.1 Modeling Objectives

Empirically, we found that the stability of axisymmetric particles increased with:

- Increasing solid content (both a function of the original petrolatum composition and the emulsion temperature)

- Increasing surfactant concentration
- Increasing characteristic radius
- Gradual transitions between features (i.e. ball and stick shaped particles with gradual slope between the sphere and the spherocylinder) rather than sharp transitions.

Additionally, we found that deformation appears to be predisposed to initiate in regions of low curvature as discussed in Section 3.2. The model described here seeks to explain both of these qualitative trends and the location predisposition.

4.2.2 Model

To develop the model for a generalized axisymmetric particle, we utilize the same notation and definition of interfacial surface forces as described in Section 4.1.2. Specifically, recall that the surface traction $\boldsymbol{t}$ is defined as

$$\boldsymbol{t}(u,v) = \frac{\boldsymbol{F}}{\delta A} = -2\gamma H(u,v)\hat{\boldsymbol{n}}(u,v) \tag{4.9}$$

The mean curvature and unit normal of a generalized axisymmetric particle varies with position, as indicated by the parameterization.

For a generalized axisymmetric particle, it is not possible to define H for entire surface regions as we did in Equations 4.10 and 4.11 for the spherocylinder; we must resort to Equations 4.6 and 4.7 to define H on a point-by-point basis over the surface. Since we cannot define a regional H, we cannot assume that tractions parallel to the central axis of the particle have the largest magnitude of all the tractions along the surface, nor can we isolate specific tractions that appear to impose a dominate stress acting on the network. Instead, we utilize surface integrals to account for all of the tractions acting on an APCE and infer the stress on the crystalline network rather than relying on simplifying arguments. We borrow this integration technique from the field

of material mechanics. Within this field, the Euler-Cauchy stress principle as defined by Truesdall and Toupin states that for any continuous material [37],

> *Upon any surface (real or imaginary) that divides the body, the action of one part of the body on the other is equivalent (equipollent) to the system of distributed forces and couples on the surface dividing the body.* [pg 702]

More generally, the force on any internal section (an internal force) balances the external forces on the connecting surface as illustrated by Figure 4.3.

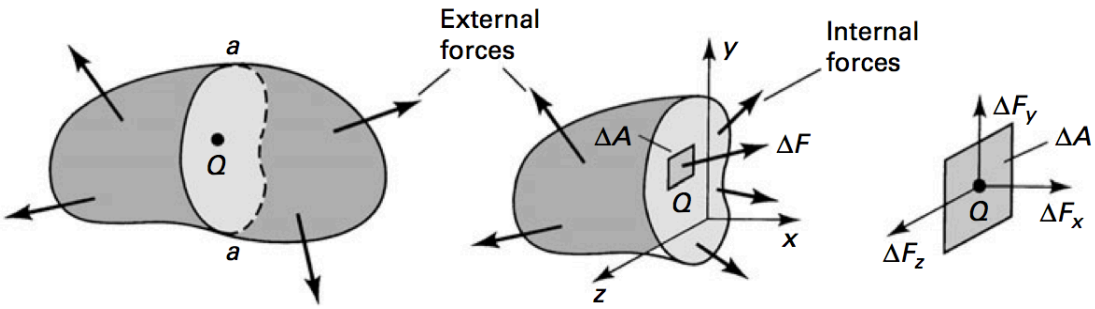


Figure 4.3: Internal forces as defined by the Euler-Cauchy stress principle. Figure taken from Ugural and Fenster [38]

When an internal force acts on a specific element of the continuous material, it subjects the element of material to stress. This stress can have any magnitude and direction depending on the external forces. Specifically, the stress of an internal element in some direction ***i*** is determined by integrating the surface traction component in the direction ***i*** exerted on a connected surface and then dividing the integrand by the segment area A:

$$\sigma_i = \frac{F_i}{A_i} = \frac{\int t_i dS}{A_i} \qquad 4.15$$

To develop a systematic method of expressing stresses and their directions, we again follow the mathematical descriptions used in the field of material mechanics. In this field, the magnitude and direction of stress on a section is defined by the nine component Cauchy Stress tensor:

$$[\sigma_{ij}] = \begin{bmatrix} \sigma_{xx} & \tau_{xy} & \tau_{xz} \\ \tau_{yx} & \sigma_{yy} & \tau_{yz} \\ \tau_{zx} & \tau_{zy} & \sigma_{zz} \end{bmatrix} \quad 4.16$$

In Equation 4.16, the first subscript denotes the direction of a normal to the plane on which the stress component acts and the second subscript denotes the direction of the stress itself [38]. Figure 4.4 depicts these stresses.

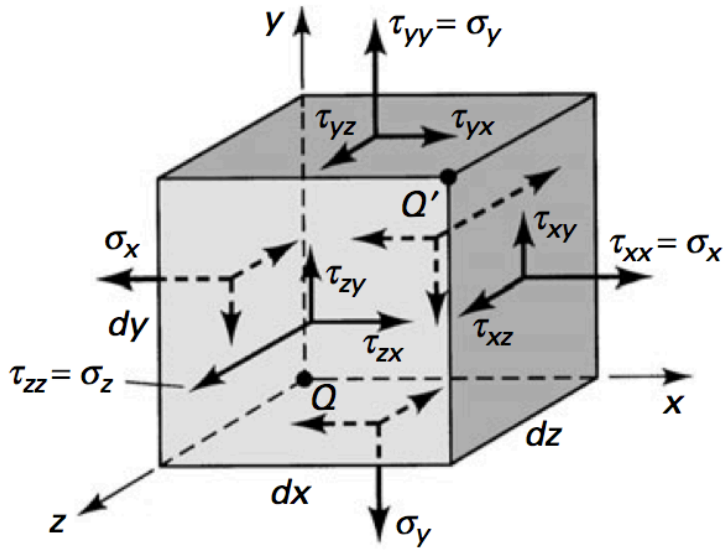


Figure 4.4: Location of Cauchy stress components acting on an element within a continuum. Figure taken from Ugural and Fenster [38].

The Cauchy Stress tensor can be simplified by determining the principal stresses of the tensor. Roark et al. gives a complete definition of this simplification [39]:

> *Through any point in a stressed body there pass three mutually perpendicular planes the stress on each of which is purely normal, tension or compression; these are the principal planes for that point. The stresses on these planes are the principal stresses; one of them is the maximum stress at the point, and one of theme is the minimum stress at the point. When one of the principal stresses is zero, the condition of is one of plane stress; when two of them are zero, the condition is one of uni-axial stress.* [pg 11]

By this definition, the principal stresses are the eigenvalues of the Cauchy Stress tensor (Eqn 4.17) and the principal planes are oriented by the associated eigenvectors.

$$[\sigma_i] = \begin{bmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & \sigma_3 \end{bmatrix} \tag{4.17}$$

For axisymmetric partially crystalline emulsion particles, one of the principal stresses is always oriented along the particle's axis of symmetry. As such, it is convenient to orient one of the coordinate axes in the APCE system parallel to the central axis of the particle. After doing this, we can take the components of the traction vector (Eqn 4.9), integrate them independently across a surface connected to an internal element, and obtain the net force acting in a principal direction, $\boldsymbol{F_i}$, on that internal element.

$$\boldsymbol{F_i}(\mathrm{w}) = \iint_{w_0(u_0,v_0,)}^{w(u,v)} \boldsymbol{t}_i dS \tag{4.18}$$

If we then divide the net force by the area of the element (which is simply bounded by the edge of the connected surface), we obtain the principal stress in the direction $\boldsymbol{i}$ on the internal element.

$$\sigma_i(\mathrm{w}) = \frac{\boldsymbol{F_i}(\mathrm{w})}{A_i(w)} = \frac{\iint_{w_0(u_0,v_0,)}^{w(u,v)} \boldsymbol{t}_i dS}{A_i(w)} \tag{4.19}$$

In Equations 4.18 and 4.19, we have introduced the variable *w* to denote a point along the principal axis $\boldsymbol{i}$. The surface connected to a segment at *w* spans the length of the

central axis from the point w_0 to w. The variables w_0 and w map to u,v space such that translating between u_0, v_0 to u,v defines the entire surface. This method of integration and relation to w is illustrated in Figure 4.5 for an idealized axisymmetric particle.

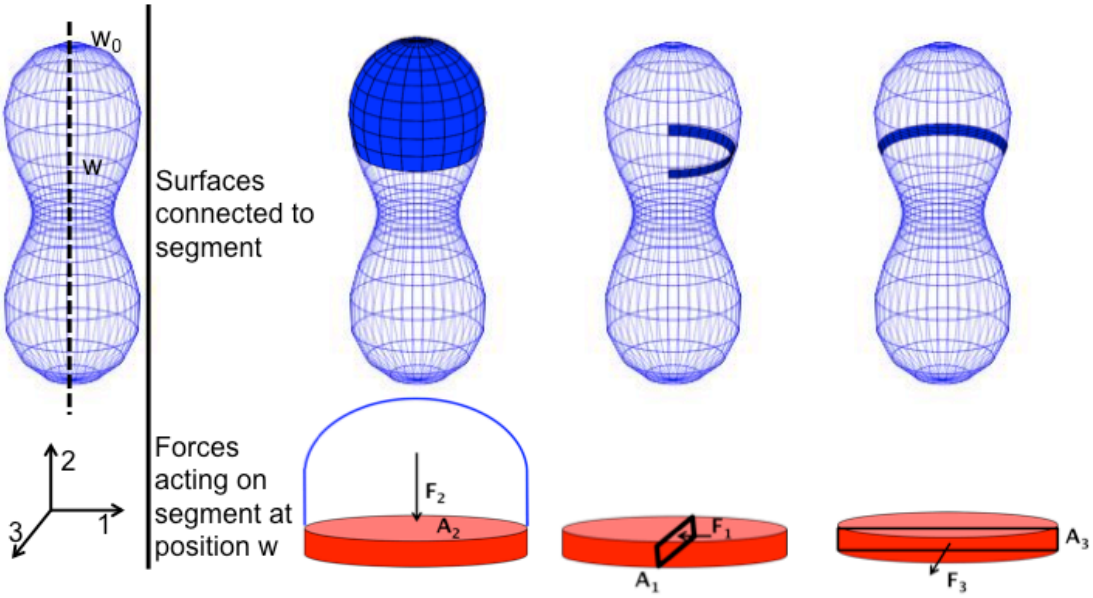


Figure 4.5: Integrating across connected surfaces to find the principal stresses on a segment (in red) at position w. For each of the principal stresses, the associated surfaces of integration are shown in dark blue. The areas A_i bounded by the connected surface are also marked. w_0 pertaining to the second principle axis is also marked.

The principal stress state is a useful way to characterize an isotropically yielding body. In the simplest sense, a body is said to yield plastically when it begins to deform irreversibly under an applied load. The magnitude of threshold load that marks the transition from elastic (reversible) to plastic (irreversible) deformation is the yield strength, Y, which is a scalar. Typically, the yield strength, Y, is equated to critical uniaxial strength, Y_{1D}. In order to compare the stress tensor (a field quantity

with magnitude and direction) with the yield strength (a scalar), we introduce a yield surface *X*, described by [39]

$$X = \Phi(\sigma_i) - K = 0 \qquad 4.20$$

where $\Phi(\sigma_i)$ is a function of the principal stress and *K* is a material constant related to Y_{1D}. Typically, *K* is proportional to Y_{1D} by a scaling factor. *X* is termed a yield surface because in the principal axes space, it represents a closed boundary that defines the transition between elastic and plastic deformation (Figure 4.6). If the function $\Phi(\sigma_i)$ is less than *K* or inside the yield surface, the material will deform elastically. If the function $\Phi(\sigma_i)$ is greater than *K* or outside the yield surface, the material will deform plastically [39]. The assertion that a specific functional $\Phi(\sigma_i)$ exists that describes deformation in this manner is defined as a yield criterion.

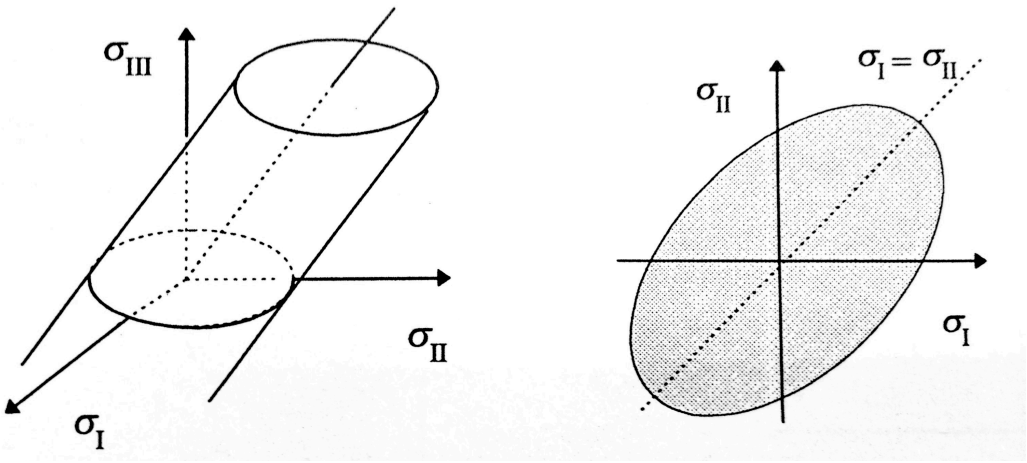


Figure 4.6: Representation of von Mises yield criterion in the principal stress space and the elliptical cross-section with the σ_1 and σ_2 plane. Figure taken from Roark [39].

In our study of APCE, we have explored applying the von Mises yield criterion (Eqn 4.19) to describe the deformation of APCE. The von Mises yield criterion is developed from the hypothesis that irreversible deformation in a material occurs when the strain energy density associated with shape modification exceeds a critical threshold value [39]

$$\Phi_{vM}([\sigma_i(w)]) = \sqrt{\frac{(\sigma_1-\sigma_2)^2+(\sigma_1-\sigma_3)^2+(\sigma_3-\sigma_2)^2}{2}} \qquad 4.21$$

Thus, if the effective von Mises stress, Φ_{vM}, exceeds K_{vM} for the semicrystaline network at any point within the APCE (denoted here by w), we expect the APCE to deform.

The equivalent stability expression for this model is given by

$$Stable \rightarrow \Phi_{vM}([\sigma_i(w)]) < \mathrm{K}_{vM} \quad for\ all\ values\ of\ w \qquad 4.22a$$

$$\text{Stable} \rightarrow \frac{\Phi_{vM}([\sigma_i(w)])}{\mathrm{K}_{vM}} < 1 \quad for\ all\ values\ of\ w \qquad 4.22b$$

As we did for the spherocylinder case, we now ask if the model given by Equation 4.22 supports the qualitative observation discussed in Section 4.2.1. As stated, K_{vm} is proportional to the yield stress of the material. Since the yield stress is proportional to the solid content, increasing the solid content increase the likelihood that a particle will be stable, which was one of the qualitative observations defined in 4.2.1. Furthermore, the effective von Mises stress is proportional to the interfacial tension since all of the principal stresses are proportional to the interfacial tension. The interfacial tension decreases with increasing surfactant concentration. Consequently, increasing surfactant concentration increases the likelihood a particle will be stable; this was another qualitative observation defined in Section 4.2.1.

The other qualitative observations defined in Section 4.2.1 relate to particle shape and particle size. With the stability criterion given in Equation 4.22 defined so generally, it is difficult to examine from the definition alone whether the model supports the observations. To aid in this examination, we have calculated and plotted the effective von Mises stress at different segments within idealized axisymmetric particles. Plotting the distribution of effective von Mises stress values illustrates which regions of a particle are under the highest stress, and subsequently which regions are more likely to deform. Here, we call these characteristic plots "internal stress distributions." Internal stress distributions for three different axisymmetric morphologies are provided in Figures 4.7, 4.8 and 4.9.

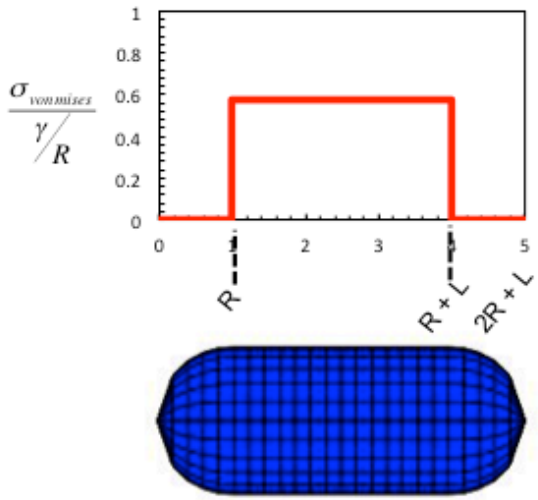


Figure 4.7: Internal stress distribution for an idealized spherocylinder APCE. The effective von Mises stress and the location are normalized by the spherocylinder radius, *R*.

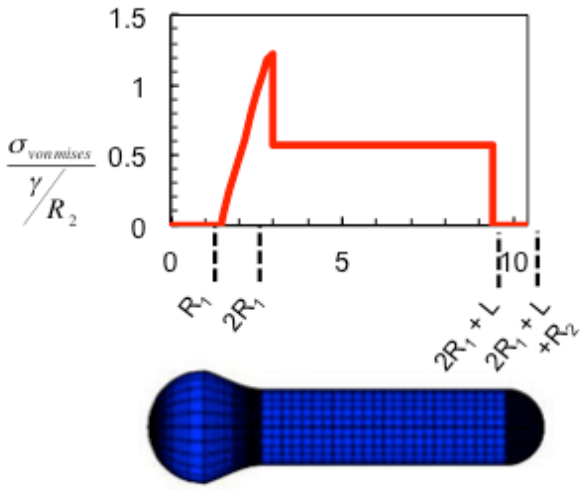


Figure 4.8: Internal stress distribution for idealized ball and stick APCE. The effective von Mises stress and the location are normalized by the radius of spherocylinder radius, R_2.

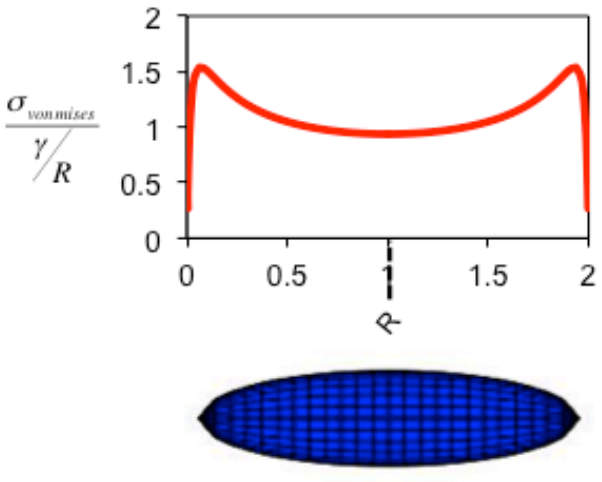


Figure 4.9: Internal distribution for idealized ellipsoidal APCE. The effective von Mises stress and the location are normalized by half of the length of the ellipsoid, R.

In Figures 4.7, 4.8 and 4.9, we have normalized the effective von Mises stress by the interfacial tension and the inverse of the characteristic radius for the specific morphology. The ability to perform this normalization further illustrates that the Φ_{vM} is proportional to the interfacial tension. Furthermore, Φ_{vM} is inversely proportional to the APCE characteristic radius. Since Φ_{vM} is inversely proportional to the characteristic radius, increasing the characteristic radius increases the likelihood that the APCE will be stable, which is another qualitative observation defined in Section 4.2.1. In spherocylinder particles, the effective von Mises stress distribution is uniform across the axis as shown in Figure 4.7. The uniform Φ_{vM} is again inversely proportional to the spherocylinder radius and proportional to the interfacial tension. With these proportionality relations, the axisymmetric model notably reduces to the same result as the spherocylinder model given in Equation 4.14.

Examining the distribution of the Φ_{vM} provides further insight into why that deformation appears to be predisposed to initiate in regions of low curvature. In Figure 4.8, the highest internal stress is located at the connection between the sphere and the cylindrical part of the idealized ball and stick (at position $2R_1$). Since the crystalline network in this segment is subjected to the highest internal stress, we expect it to undergo deformation prior to segments within the cylindrical or spherical system. This is consistent with the results presented in Section 3.2.

Agreement between the qualitative observations and the predictions by the model suggest that this generalized model could also be practically useful to determine when axisymmetric particles deform statically. Additionally, because the model is purposefully general, it can be used to analyze any axisymmetric shape. Product engineers interested in producing an APCE that deforms at one or more specific

locations can draft idealized shapes, compute internal stress distributions, and infer where their APCE is likely to deform. However, the potential utility of such a model depends on its accuracy. In Chapter 5, we describe our preliminary attempt at verifying the axisymmetric model from a quantitative standpoint by analyzing empirical results through a discrete modeling program.

4.3 Torus Model

The third model describes the stability of torus particles (Section 3.1.4).

4.3.1 Modeling Objectives

Empirically, our collaborators found that the stability of axisymmetric particles increased with:

- Increasing solid content (both a function of the original petrolatum composition and the emulsion temperature)
- Increasing surfactant concentration

Additionally, our collaborators found that the stability of axis symmetric particles was influenced by the characteristic aspect ratio (see following discussion). The model described here seeks to explain these qualitative trends.

4.3.2 Model

To develop the model for a generalized torus particle, we first define characteristic size parameters for the shape: R, the outer radius of the torus, and r, the inner radius of the torus (Figure 4.10).

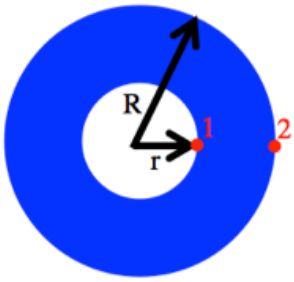


Figure 4.10: Cross-section of APCE torus cut along its equator. Pressure differences calculated at points 1 & 2 in red.

Again, following the discussion given in Section 4.1.2, the surface traction, $\boldsymbol{t}$, is

$$\mathbf{t}(\mathrm{u},\mathrm{v}) = -2\gamma H(u,v)\hat{\boldsymbol{n}}(u,v) \tag{4.23}$$

At points along the inner and outer equator of the torus, the mean curvature in Equation 4.23 reduces to simpler expressions for perfectly symmetric toroidal droplets.

$$\mathbf{t}(\mathrm{u}_1,\mathrm{v}_1) = -\gamma\left(\frac{1}{r_t} - \frac{1}{r}\right)\hat{\boldsymbol{n}}(\mathrm{u}_1,\mathrm{v}_1) \tag{4.24}$$

$$\mathbf{t}(\mathrm{u}_2,\mathrm{v}_2) = -\gamma\left(\frac{1}{r_t} + \frac{1}{R}\right)\hat{\boldsymbol{n}}(\mathrm{u}_2,\mathrm{v}_2) \tag{4.25}$$

where r_t is the radius of the cross section of the torus, equivalent to

$$r_t = \frac{1}{2}(R - r) \tag{4.26}$$

From these simplified traction equations, it is possible to calculate the pressure difference across the particle interface, since the pressure is simply the magnitude of the traction vector. The pressure gradient (i.e. the difference between the Laplace pressures) that exists between the points 1 and 2 in Figure 4.10 causes the torus to shrink over time while conserving its volume. With the torus shrinkage, the pressure

gradient between the two points also evolves with time, thereby affecting the dynamics of the morphology evolution.

Based on the traction equations, the pressure difference (Eqn 4.24 and 4.25) at the two points can be computed as a function of the torus aspect ratio for a particle with a specific initial aspect ratio. The torus aspect ratio is defined as the ratio between the outer ring radius, R, and the inner ring radius, r.

$$\frac{\Delta P_1}{\left(\gamma/V_0^{1/3}\right)} = \left(\frac{\pi^2}{4}\left[\left(\frac{R_0}{r_0}\right)^3 - \left(\frac{R_0}{r_0}\right)^2 - \left(\frac{R_0}{r_0}\right) + 1\right]\right)^{1/3} \cdot \left(\frac{2}{\frac{R}{r}-1} + \left(\frac{R}{r}\right)^{-1}\right) \tag{4.27}$$

$$\frac{\Delta P_2}{\left(\gamma/V_0^{1/3}\right)} = \left(\frac{\pi^2}{4}\left[\left(\frac{R_0}{r_0}\right)^3 - \left(\frac{R_0}{r_0}\right)^2 - \left(\frac{R_0}{r_0}\right) + 1\right]\right)^{1/3} \left(\frac{2}{\frac{R}{r}-1} - 1\right) \tag{4.28}$$

In Eqn 4.27 and 4.28, the subscript "0" denotes the characteristic torus dimensions at time 0. Plotting these pressure differences (Eqn 4.28) with respect to R normalized by the initial value of R shows when the pressure differences across the droplet become negative, and thus, when the ring becomes statically unstable. An example of one of these plots is given in Figure 4.11. The point at which the pressure differences across the droplet become negative defines the transition in static stability, and thus the stability criteria (Eqn 4.29):

$$\text{Stable} \rightarrow \frac{\Delta P_1}{\left(\gamma/V_0^{1/3}\right)} - \frac{\Delta P_2}{\left(\gamma/V_0^{1/3}\right)} > 0 \tag{4.29a}$$

$$\text{Stable} \rightarrow \frac{\left(\frac{2}{\frac{R}{r}-1}-1\right)}{\left(\frac{2}{\frac{R}{r}-1}+\left(\frac{R}{r}\right)^{-1}\right)} < 1 \quad for\ all\ values\ of\ u, v \tag{4.29b}$$

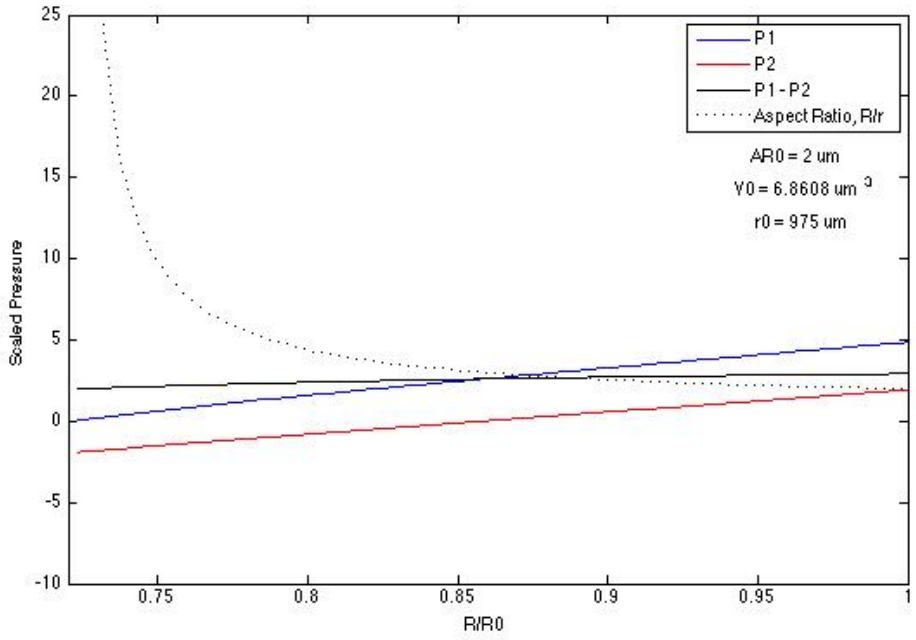


Figure 4.11: Evolution of pressure differences across droplet interface and ring cross-section for rings of initial aspect ratio 2. The pressure is scaled by $y/V_0^{1/3}$.

Notably, the stability criterion developed in this section and summarized in Eqn 4.29 are not dependent on the interfacial tension or the solid content of the particle. As such, this criterion does not capture the qualitative observations described in Section 4.3.1. It may, however, have greater applicability to the dynamics of torus evolution.

4.4 Summary

In this chapter, we developed three models from fundamental laws of interfacial phenomena and differential geometry to describe morphological stability in APCE. The spherocylinder and axisymmetric model appear to capture qualitative trends in static APCE stability well, while the torus model may be better suited to

capture morphology dynamics. The following chapter describes our preliminary attempt to validate the axisymmetric model (which reduces to the sphereocylinder model for rod shaped particles) experimentally.

Chapter 5

ANALYSIS OF PARTICLES IN CONTEXT OF DISCRETIZED AXISYMMETRIC MODEL

This chapter details our efforts to validate the analytical model for axisymmetric particle deformation presented in Chapter 4. These efforts were made in pursuit of the second goal of this work: to develop quantitative understanding of morphology deformation in anisotropic partially crystalline emulsion particles. Of the four different APCE morphologies described in Chapter 3, the three morphologies produced at the University of Delaware are axisymmetric. These particle morphologies are the spherocylinder (Section 3.1.1), ball and stick (3.1.2) and dual ball and stick (Section 3.1.3). Since we had the ability to produce these axisymmetric particles at the University of Delaware, we focused on validating the axisymmetric model rather than the preliminary torus model.

It should be noted that all experimental particles of the spherocylinder, ball and stick, and dual ball and stick types are only approximately axisymmetric due to the inevitable imperfections in the experimental systems. In the APCE system, imperfections in the microcapillaries and bending of the particles upon ejection particularly cause each of the "axisymmetric" particles to be asymmetric. Despite the imperfections, we seek to determine the perfectly axisymmetric model's robustness in predicting the stability of particles cast in axisymmetric molds.

To apply the axisymmetric model to droplet morphologies that were produced in the experiments, we developed a discrete modeling program in MATLAB that uses

images of axisymmetric particles to calculate the von Mises stress at different internal segments. We then compared the empirical internal stress distributions between particles that exhibited different degrees of stability to determine if there was any correlation between the inferred von Mises stress and static stability.

To date, the results of this validation attempt are inconclusive. As is discussed in this chapter, numerical artifacts of the discrete modeling program (namely the error associated with surface curvature calculations) limit our confidence in the internal stress distributions produced from experimental images. We do believe, however, that the accuracy of the discrete modeling program can be improved substantially by using more rigorous parameterization methods. We discuss these methods in brief in Section 5.4, which themselves lay the groundwork for future examination of axisymmetric APCE and the developed axisymmetric model.

5.1 Discrete Program Development Objectives

The objective of the discrete modeling program is to take an image of an axisymmetric APCE and calculate the magnitude and location of the segment with the maximum von Mises stress. This input output structure is depicted in Figure 5.1.

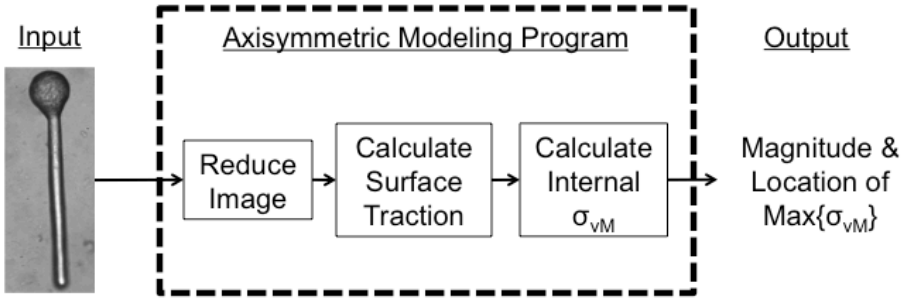


Figure 5.1: Input-output structure for three component axisymmetric discrete modeling program.

A program that accurately accomplishes this task would be useful in two applications. First, we could use the program to attempt to validate the axisymmetric model as previously discussed. Second, if the axisymmetric model shows utility in predicting APCE static stability, the program could be used to take any sketch of an APCE design and determine the locations where the APCE is most likely to undergo deformation. Product engineers could use the program to rapidly evaluate APCE morphologies, specifically those that are not easily parameterized and consequently not easily defined analytically. For these reasons, we felt it was worthwhile to begin developing such a program.

5.2 Working Program

Here, we first provide an overview of the three-component method developed to accomplish the objectives described in Section 5.1. We then detail each of the three components in the following subsections.

5.2.1 Overview

The discrete modeling program takes an image of an axisymmetric APCE and calculates the magnitude and location of the segment with the maximum von Mises stress. Prior to using this program, an axisymmetric APCE is first imaged through brightfield microscopy as described in Section 2.1.4. The surface profile of each particle is then extracted from the image by thresholding pixels of different intensity within ImageJ software (NIH Image). Complete extraction of the profile results in a text image that gives the location of pixels along the particle's profile. These locations are then input into the first MATLAB program (Appendix B). The first MATLAB program computes the instantaneous curvature of the surface by fitting multiple

parametric polynomials to the profile. The instantaneous curvature is proportional to the local surface traction by the Young-Laplace equation where interfacial tension is the proportionality constant. The output of the first MATLAB program is the surface tractions as a function of position.

The output of the first program serves as the input to the second MATLAB program (Appendix C), which computes the internal principal stress-state within segments perpendicular to the axis of symmetry. The second program effectively performs a discrete surface integral for the experimental particles by summing the tractions multiplied by the area along the profile connected to a specific segment, and then dividing this sum by the segment area. Inputting the stress-state into the von Mises failure criteria gives a normalized average stress for the segment. The program outputs the maximum average stress and the location of the maximum as a means of quantitatively predicting whether the particle will undergo deformation, and where the deformation will likely initiate.

5.2.2 Image Reduction

ImageJ (NIH Image) is used to extract the meridian profile of an APCE from a brightfield microscopy image. Prior to extraction, each original 8-bit grayscale Tiff (tagged image file format) image (Figure 3.8) is rotated (Figure 5.2A) so that the central axis of the APCE is oriented vertically. Rotating the image simplifies the process of importing the profile into the Matlab program. Note that we have specifically chosen to demonstrate the profile extraction and surface curvature calculation process using an APCE example that is not perfectly axisymmetric (i.e. it is a distorted ball and stick type droplet) to showcase the distribution of surface curvatures present in such distorted droplets.

After rotating the image, the *Find Edges* ImageJ feature is used to isolate the APCE profile (Figure 5.2B). *Find Edges* identifies sharp transitions in intensity between nearby pixels. Adjusting the pixel intensity threshold allows the user to manually select the pixels that demark the APCE boundary. Once satisfied with the selection, the user transforms the thresholded image into a binary image (Figure 5.2C) using ImageJ's *Make Binary* command. In the binary image, the black pixels were originally pixels that were below the intensity threshold. As seen in Figure 5.2C, several pixels that are not within the profile of the APCE (namely the crystalline structure pixels) are below the intensity threshold and show up in the binary image. These pixels are manually replaced with white pixels, resulting in Figure 5.2D.

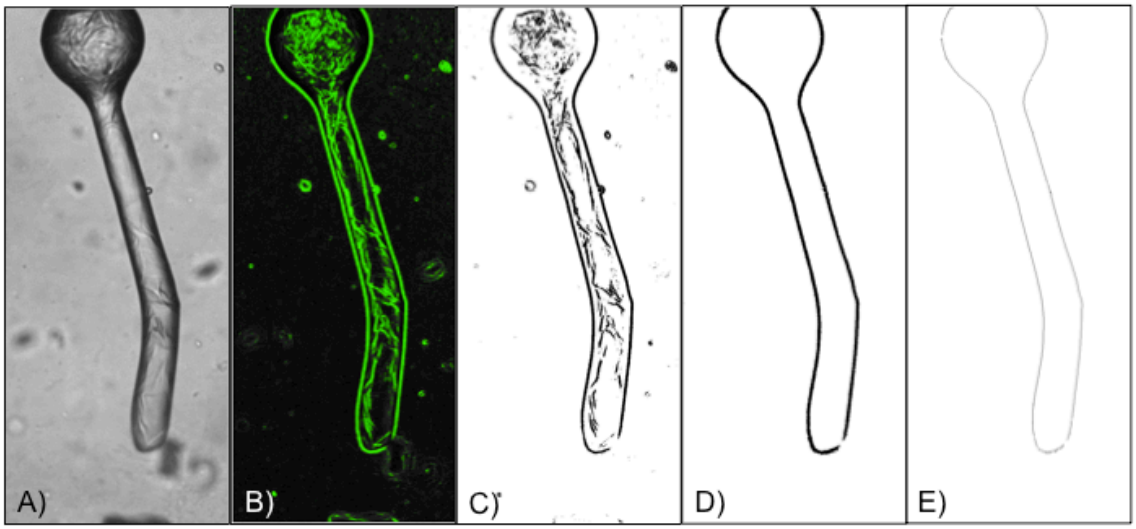


Figure 5.2: Extraction of droplet profile using Image J. A) Rotated bright field microcopy image. B) Image after *Find Edges* application. C) Binary Image. D) Binary image with all black pixels besides those demarking the profile removed. E) Skeletonized image.

In Figure 5.2D, it is apparent that the thickness of the profile is multiple pixels in width and that the thickness differs along profile locations. Variable profile thickness often occurs because the entire APCE is not in the same focal plane. From a data processing standpoint, having multiple pixels demark the location of the profile inhibits ordering the vectors when the pixel locations are imported to MATLAB. As such, it is easier to reduce the profile thickness using ImageJ's *Skeletonize* tool. *Skeletonize* finds the midpoint of a set of darkened pixels in the binary image, and then eliminates points other than the midpoint. Using *Skeletonize* on the binary image in Figure 5.2D results in the profile given in Figure 5.2E,which is one pixel in thickness.

After extracting the profile, the binary image seen in Figure 5.2E was saved as text file to later be used in MATLAB. In the text file, 1's denote black pixels while 0's denote white pixels.

5.2.3 Calculating Surface Traction

The first MATLAB program computes the local surface tractions at points along the APCE meridian profile. For APCE that are perfectly axisymmetric, rotating the meridian profile about the central axis defines the surface of the APCE. All points on the surface that are at the same axial position have the same surface curvature. Consequently, if the traction along the meridian profile is known, the tractions across the entire APCE surface are also known. The program described here relies on this fundamental mathematical argument to compute the surface tractions across three-dimensional APCE surface from a two-dimensional image of the meridian profile. For APCE that are not perfectly axisymmetric (which constitutes every experimental APCE), modeling the three-dimensional surface introduces some error in the

calculation of true surface traction. Nevertheless, we apply this simplification during this preliminary stage of model development.

Prior to describing how the program operates, we define a coordinate system for the two-dimensional image space (Figure 5.3). Let *x* and *y* be two perpendicular axes in the plane of the APCE meridian profile. Let the meridian profile be defined as a parametric two dimensional vector function, $\boldsymbol{f}$, where the components of $\boldsymbol{f}$ give the (x, y) coordinates of a point along the meridian profile:

$$\boldsymbol{f}(u) = \mathrm{x}(u) \cdot \boldsymbol{i} + y(u) \cdot \boldsymbol{j} \qquad 5.1$$

By defining the meridian profile in this way, we can define the surface of an axisymmetric APCE ($S_{axisymmetric}$) in $\Re^3$. Using the same notation as used in Chapter 4, we let $\boldsymbol{r}$ denote the radius vector from the origin of the Cartesian coordinate system (x, y, z) with the unit vectors ($\boldsymbol{i}$, $\boldsymbol{j}$, $\boldsymbol{k}$). The surface $S_{axisymmetric}$ is then defined by the vector equation

$$S_{axisymmetric}: \quad \mathbf{r}(u, v) = x(u) \cdot \cos(v)\, \boldsymbol{i} + y(u) \cdot \boldsymbol{j} + x(u) \cdot \sin(v) \boldsymbol{k} \qquad 5.2$$

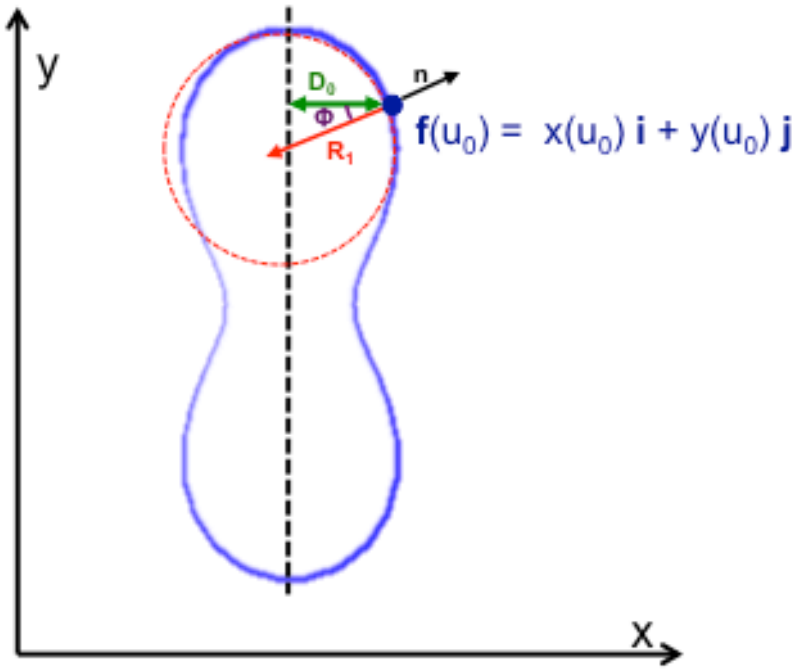


Figure 5.3: Coordinate system in discrete modeling program. Blue line demarks the meridian profile of an axisymmetric APCE. Rotating the meridian profile about the central axis (dashed black line) gives the surface of an axisymmetric APCE. u_0 denotes a specific point in the meridian profile.

Having defined this coordinate system and notation, we proceed to describe how the first MATLAB program computes the APCE surface traction program. The first section of the program (labeled in Appendix C) converts the text file into two vectors ***x(u)*** and ***y(u)***, which denote the coordinates of the meridian profile. Specifically, the binary text files described in Section 5.2.2 are read into MATLAB as a matrix of 1s and 0s with the position in the matrix corresponding to the pixel position in the x-y plane. Iterating through the matrix and determining the indexes associated with elements equal to 1 gives a set of x-y coordinates for the meridian profile. Since MATLAB reads in lines of the text file row by row, the set of x-y

coordinates are not ordered according to location along the meridian profile. Lines 33 to 71 are devoted to reordering the sets of (x, y) coordinates so that as the program indexes through the ***x*** and ***y*** vectors, it transverses counterclockwise about the meridian profile. The index of the ***x*** and ***y*** vectors serves as a convenient parameter *u* in the program.

After cataloging ***x(u)*** and ***y(u)***, we calculate the instantaneous mean curvature, *H(u,v)* at every *u* along the profile. From Chapter 4, recall that the surface traction on an APCE is related to the mean curvature by

$$\boldsymbol{t}(u,v) = \frac{\boldsymbol{F}}{\delta A} = -2\gamma H(u,v)\hat{\boldsymbol{n}}(u,v) \tag{5.3}$$

One definition of the mean curvature is given in terms of the two principal curvatures, R_1 and R_2:

$$H(u) = \frac{1}{2}\left(\frac{1}{R_1}(\mathrm{u}) + \frac{1}{R_2}(u)\right) \tag{5.4}$$

For an axisymmetric surface with a meridian profile defined parametrically, the first principal curvature (the curvature associated with the meridian profile), is given by: [41]

$$\frac{1}{R_1}(u) = \frac{|\boldsymbol{f}'(u) \times \boldsymbol{f}''(u)|}{|\boldsymbol{f}'(u)|^3} \tag{5.5}$$

where ***f'(u)*** and ***f''(u)*** denote the first and second derivatives with respect to *u* at a specific *u*:

$$\boldsymbol{f}'(u) = \frac{d\boldsymbol{f}}{du}|_{u=u} \tag{5.6a}$$

$$\boldsymbol{f}''(u) = \frac{d^2\boldsymbol{f}}{du^2}|_{u=u} \tag{5.6b}$$

The second principal curvature (the curvature associated with rotation about the central axis) is given by [41]

$$\frac{1}{R_2}(u) = \frac{cos(\emptyset(u))}{\mathrm{D(u)}} \tag{5.7}$$

where *D(u)* is the distance between the point at *u* on the meridian profile and the closest point on the central axis. $\phi(u)$ is the angle between the unit normal vector, $\hat{\boldsymbol{n}}$, of the surface and the line between the point at *u* on the meridian profile and the closest point on the central axis. Each of these variables are marked in Figure 5.1.

Section 2 of the MATLAB program computes the first principal curvature (Equation 5.5). To do this, the program first isolates each point along the meridian profile in terms of *u*. For convenience, we describe a single point on the meridian profile as u_0. At every u_0, the program fits two separate third order polynomials in x and y with respect to *u*. Points before, after, and including u_0 are included in the polynomial fit. The number of points included is left as a function input to be specified by the user. Typically 9 – 13 total points with u_0 as the middle point are included in the fit. After fitting the continuous polynomial *x(u)* and *y(u)*, the first and second derivatives are computed according to Equation 5.6. This section of the program also calculates the direction of $\hat{\boldsymbol{n}}$ which is parallel to the cross product of the first and second derivatives of $\boldsymbol{f}$ with respect to *u*:

$$\hat{\boldsymbol{n}}(u) = \frac{-\left(\boldsymbol{f}'(u) \times \boldsymbol{f}''(u)\right)}{|\boldsymbol{f}'(u) \times \boldsymbol{f}''(u)|} \tag{5.8}$$

Section 3 of the MATLAB program computes the second principal curvature (Equation 5.7). To accomplish this, the user first has to manually select points along the apparent axis of symmetry of an APCE. These points have to be manually selected because the axis of symmetry in an APCE image is not necessarily perfectly aligned with the y-axis as depicted in Figure 5.3. Reexamination of the particle images in Figures 5.1 and 5.2 show the misalignment of the central axis for real particle images. After selecting the axis of symmetry using the *ginput* function in

MATLAB, $D(u)$ and $\phi(u)$ are calculated. $D(u)$ is found for every u_0 using the Pythagorean Theorem between a point on the axis of symmetry and the point u_0 in question. $\phi(u)$ is determined using Equation 5.9

$$cos(\emptyset) = \frac{(-\hat{\boldsymbol{n}}(u) \cdot \boldsymbol{c}(u))}{|\hat{\boldsymbol{n}}(u)|\,|\boldsymbol{c}(u)|} \qquad 5.9$$

where $\hat{\boldsymbol{n}}$ is the unit normal vector of the surface, and $\boldsymbol{c}$ is the vector from the point $(x(u_0), y(u_0))$ on the meridian profile and the closest point on the central axis (parallel to the line marked D_0 in Figure 5.3).

In Section 4 of the MATLAB program, twice the value of the mean curvature at every u is found by summing the two principal curvatures (Equation 5.4).

In Section 5, the surface traction is calculated using Equation 5.3 after inputting a surface tension for the system. In our simulations, we have used 10 dyne/cm as a reasonable surface tension estimate at moderate surfactant concentrations in absence of a directly measured value. Finally, the program plots the surface traction along the meridian profile (see Figure 5.4). The program also outputs a matrix of the unit normal, the magnitude of the surface traction as a function of the (x, y) position on the surface.

5.2.4 Calculating Effective von Mises Stress for Different Segments.

The second MATLAB program computes the effective von Mises stress at different segments along the APCE central axis by performing a discrete integral of the tractions over the surface connected to each segment to determine the net force acting in a principal direction. Dividing by the area of the segment gives the principal stress. This method of determining stresses is described in Section 4.2.2 for surfaces defined by analytical parametric equations. Experimental particle surfaces are defined

by discrete points (pixels in a image) rather than by continuous analytical equations. Consequently, integration over the surface must be discretized in order to determine the local effective stress. Such discretization introduces error into the stress calculation. Despite this known error, we again proceed with the model development in attempt to validate the axisymmetric model.

Using the same coordinate system and notation described in Section 5.2.3, we describe how the second MATLAB program computes the principal stresses acting on a segment along an axisymmetric particles central axis. Note that the normal of such a segment is parallel to the central axis. The first section of the program (labeled in Appendix C) first imports the matrix output from the first MATLAB function. This matrix contains the unit normal and the magnitude of the surface traction as a function of the (x, y) position on the surface. The ***x(u)*** and ***y(u)*** vectors described in Section 5.2.3 are extracted from the columns of the matrix and redefined within the second program. Next, the user manually selects points along the apparent axis of symmetry of the APCE. After doing so, the program reorients the position of each point and the associated normal vector so that the center of each segments lies on the y-axis (x = 0) of the coordinate system. The reorientation aids in summation of the individual components of the traction vector during discrete surface integration. Additionally, defining the axis of symmetry separates the elements of the ***x(u)*** and ***y(u)*** vectors into two groups, one set of points which lie on one side of the central axis and the other set of points which lie on the other side of the central axis. We call these groups halves of the meridian profile.

The second section of the MATLAB program computes the effective von Mises stress from one half of the meridian profile. In a perfectly axisymmetric

particle, the two halves of the meridian profile are identical; consequently, the effective von Mises stress is the same regardless of which half the stress is computed using. In experimental particles, however, the two sides of the meridian profile are not identical; consequently, the effective von Mises stress differs depending on which half of the profile is used. In the program and analysis presented here, we average the effective von Mises stress computed for the two halves of experimental particles.

To compute the effective von Mises stress using one half on the meridian profile, we first recall that the principal stress in a direction $\boldsymbol{i}$, σ_i, is determined by integrating the i component of the traction over the connected surface and then dividing by the area of the segment A_i.

$$\sigma_i(\mathrm{w}) = \frac{\boldsymbol{F_i}(\mathrm{w})}{A_i(w)} = \frac{\iint_{w_0(u_0,v_0,)}^{w(u,v)} \boldsymbol{t}_i dS}{A_i(w)} \tag{5.10}$$

The principal stress in $\boldsymbol{i}$ varies with axial position, denoted by w. For a specific point u_0, the surface integral can be approximated as a line integral over the line s multiplied by a width δL (Eqn 5.11). In this case, s is the rotation of the single point u_0 about the central axis. δL is the distance between u_0 and the following u, $u_{0+1,}$ which represents the thickness of the segment of the APCE.

$$\iint_{w_0(u_0,v_0,)}^{w(u,v)} \boldsymbol{t}_i dS \approx \delta L \int_{w_0(v_0,)}^{w(v)} \boldsymbol{t}_i ds \tag{5.11}$$

When integrating over a revolved curve of radius R_{rev}, the line integral becomes:

$$\delta L R_{rev} \int_{v_o}^{v} \boldsymbol{t}_i dv \tag{5.12}$$

Equation 5.12 is the approximate force contribution in direction i of a discrete point in profile of an APCE $F_i(u_0)$. Note that for a single point, R_{rev} is simply the $x(u_0)$. These force contributions in the first and second principal direction are given in Equations 5.13 – 5.14:

$$F_1(u_o) = \delta Lx(u_0) \int_{-\pi/2}^{\pi/2} 2\mathrm{H}(\mathrm{u}_0) n_1(\mathrm{u}_0) \cos(v)\, dv = 4\,\delta Lx(u_0)\mathrm{H}(\mathrm{u}_0) n_1(\mathrm{u}_0) \quad 5.13$$

$$F_2(u_o) = \delta Lx(u_0) \int_0^{2\pi} 2\mathrm{H}(\mathrm{u}_0) n_2(\mathrm{u}_0) dv = 4\,\pi\delta Lx(u_0)\mathrm{H}(\mathrm{u}_0) n_2(\mathrm{u}_0) \quad 5.14$$

In the context of the defined variables in coordinate system depicted in Figure 5.3, the first and second principal directions, A_1 and A_2 are

$$A_1(u_o) = 2x(u_0)\delta L \quad 5.15$$

$$A_2(u_o) = \pi(\mathrm{x}(u_0))^2 \quad 5.16$$

For the first principal direction, the $F_1(u_0)$ constitutes all of the force exerted by the APCE in direction ***1*** on A_1, given that segments in the APCE are oriented with their normals parallel to the central axis. This, however, is not the case for principal direction ***2***. In principal direction ***2***, the APCE surface connected to A_2 consists of all the discrete points of the meridian profile from u = 0 to u_0. To find the total force exerted on A_2, all of the individual $F_2(u_0)$ are summed to determine the net force on A_2. Consequently, σ_1 and σ_2 are

$$\sigma_1(\mathrm{w}) = 2\mathrm{H}(\mathrm{u}_0) n_1(\mathrm{u}_0) \quad 5.17$$

$$\sigma_2(\mathrm{w}) = \frac{\sum_{u=0}^{u=u_0} 4\pi\delta Lx(u_0)\mathrm{H}(\mathrm{u}_0) n_2(\mathrm{u}_0)}{\pi(\mathrm{x}(u_0))^2} \quad 5.18$$

Recall that from Section 4.2.2 that the effective von Mises stress, Φ_{vM}, is a function of the three principal stresses computes σ_1, σ_2, and σ_3 [40]:

$$\Phi_{vM}([\sigma_i(w)]) = \sqrt{\frac{(\sigma_1-\sigma_2)^2+(\sigma_1-\sigma_3)^2+(\sigma_3-\sigma_2)^2}{2}} \quad 5.19$$

where again *w* denotes a position along the central axis. For perfectly axisymmetric particles whose central axis is aligned with the second principal direction (see Figure 4.5), $\sigma_3 = \sigma_1$. This is again because all points on the surface that are at the same axial

position have the same surface curvature. Since $\sigma_3 = \sigma_1$, the effective von Mises stress for an axisymmetric particle reduces to:

$$\Phi_{vM}([\sigma_i(w)]) = \sqrt{(\sigma_1 - \sigma_2)^2} \qquad 5.20$$

Using this reduced equation, the program computes the effective von Mises stress based on one half of the meridian profile.

In Section 3 of the MATLAB program, the effective von Mises stress of the other half of the meridian profile is computed using the same method used in Section 2.

Section 4 of the MATLAB program plots the two effective von Mises stresses computed from the two halves of the meridian profile. By inspecting this plot, the user can determine the degree to which the two effective von Mises stresses are the same (i.e. the degree of particle symmetry) and the location of the maximum effective von Mises stress.

5.3 Preliminary Results

Using the first MATLAB program, we were able to compute surface tractions like the one given in Figure 5.4. By visual inspection, the surface traction program appears to give reasonable values for the magnitude and direction of the surface traction. Lower tractions are observed in the spherical portion of the APCE given in Figure 5.4 as expected; all the vectors appear to be normal. However, without comparing these tractions to a known surface, we have no basis to assert that this method is accurate.

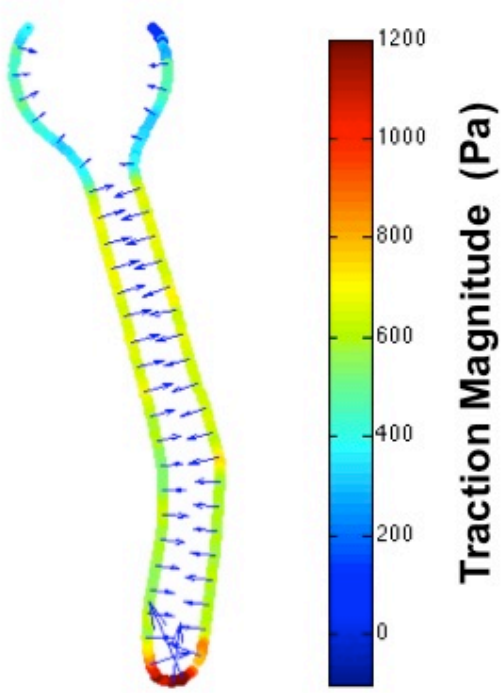


Figure 5.4: Surface tractions in distorted ball and stick type APCE. Surface tension assumed to be 10 dyne/cm.

Plotting the internal stress distributions (Figure 5.5) for nearly perfectly axisymmetric APCE distribution hints at inaccuracy in the model. The internal stress distribution is not a continuous flat profile within the cylindrical portion of the particle, as is expected for a perfectly axisymmetric spherocylinder particle (Figure 4.7). Instead, the experimental internal stress distribution contains scattered peaks across all axial positions. This unexpected result could be an artifact of the approximations made in the surface traction function, the segment stress function, or both functions.

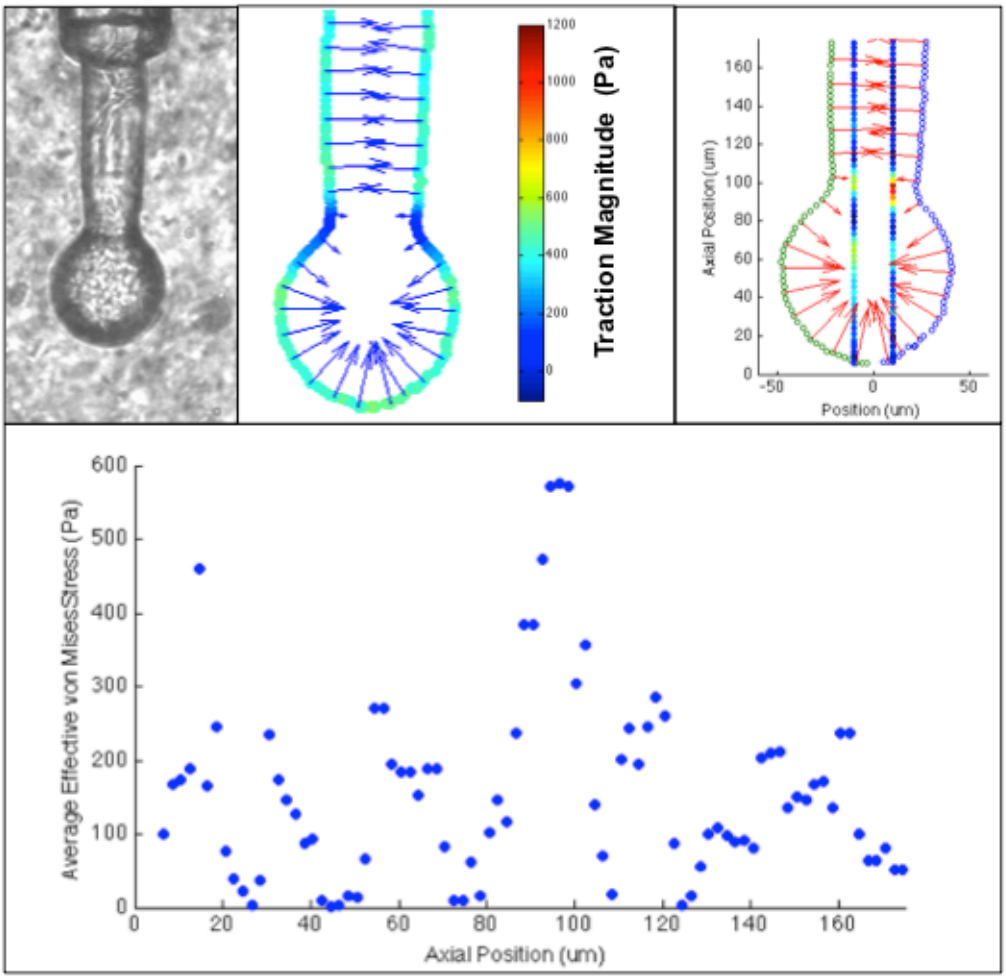


Figure 5.5: Internal stress distribution of experimental APCE. A) Original image of axisymmetric APCE ball and stick. B) Traction forces acting on surface obtained from first MATLAB program. C) Effective von Mises stress (vertical color-graded lines) for two halves of the APCE meridian profile. D) Local average effective von Mises stress for the two meridian profile halves.

5.3.1 Theoretical Example

To investigate the potentially inaccurate internal stress distributions produced by the discrete modeling programs, we began by comparing the known surface tractions of an analytically defined surface with those computed using the first discrete

modeling program. Specifically, we modeled the meridian profile of a perfectly axisymmetric particle (Figure 5.6) as a Cassini oval. The Cassini oval was chosen specifically for this test because the Cassini oval has a highly variable curvature distribution, unlike simpler shapes like the spherocylinder.

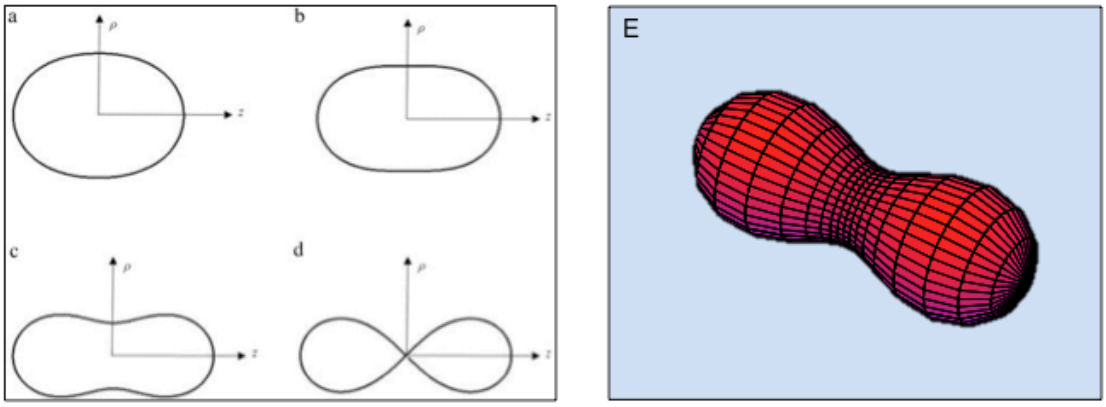


Figure 5.6: Idealized axisymmetric APCE modeled by a rotated Cassini oval. Cassini ovals. Various shapes of Cassini oval holding d constant with $(c/d)^2$ of A) 0.3; B) 0.5; C) 0.8; and D) 1.0. E) Surface generated by rotating c = 0.95 and d = 1 Cassini oval about the central axis. Images A – D from Yeh et al [42].

The Cassini oval can be defined analytically using as a parametric two-dimensional vector function, $\boldsymbol{f}_{cas}$, where the components of $\boldsymbol{f}_{cas}$ give the (x, y) coordinates of a point on the oval

$$\boldsymbol{f}(u) = \mathrm{x}(u) \cdot \boldsymbol{i} + y(u) \cdot \boldsymbol{j} \qquad 5.21a$$

$$\mathrm{x}(u) = \sin(\mathrm{u}) \left(c^2 \cos(2t) + \sqrt{-c^4 + d^4 + c^4(\cos{(2t)})^2}\right)^{1/2} \qquad 5.21b$$

$$y(u) = \cos(u)\left(c^2\cos(2t) + \sqrt{-c^4 + d^4 + c^4(\cos(2t))^2}\right)^{1/2} \qquad 5.21c$$

In Equations 5.21b and 5.21c c and d are constants that determine the aspect ratio and shape of the curve (See Figure 5.6 A – D), much like the lengths of the major- and semi-major axis in an ellipse. A surface formed by rotating half of the Cassini oval about its axis of symmetry, S_{cas}, is given by a function parametric in *u* and *v*:

$$S_{Cas}: \quad \mathbf{r}(u,v) = x(u)\cdot\cos(v)\,\boldsymbol{i} + y(u)\cdot\boldsymbol{j} + x(u)\cdot\sin(v)\boldsymbol{k} \qquad 5.22$$

The complete S_{cas} spans $0 \le u \le \pi$ and $0 \le v \le 2\pi$. The exact surface traction of $S_{cassinian}$ can be computed directly by applying Equations 4.2 – 4.9 to $S_{cassinian}$. This computation was performed using Maple 15 Software (Maplesoft, see code in Appendix D).

To evaluate the accuracy of the discrete program, we then took an image of the Cassini oval, reduced the image using the same methodology described in Section 5.2.2, and computed the surface tractions using the first MATLAB program. The Cassinian oval image (Figure 5.7) was generated using Maple 15. L_{cas} is length of the major axis of the oval.

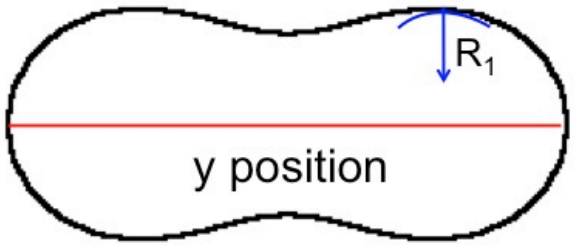


Figure 5.7: Meridian profile of the Cassini oval (c = 0.9, d = 1).

Plotting the two normalized principal radii of curvature , $\frac{R_1}{L_{cass}}$ and $\frac{R_2}{L_{cass}}$, computed by the surface traction program along with the analytical principal curvatures (Figure 5.8) illustrates which regions of the APCE image are prone to error in the curvature calculation. Examining Figure 5.7 shows that the first MATLAB program calculates the magnitude of the first and second principal radii of curvature to within 9.5 % from $0.2 \leq w_{norm} \leq 1.8$ where w_{norm} is the normalized axial position, equivalent to

$$w = \frac{y}{L_{cass}} \tag{5.23}$$

However, from $0 \leq w_{norm} < 0.2$ and $1.8 < w_{norm} \leq 2.0$ the error in the first and second principal radii of curvature is greater than 38.4%.

The larger error at the ends of the APCE is likely associated with the pixel resolution of the image. Fitting polynomial to flat sections (see the ends of image in Figure 5.7) that represent a curved interface results in error, as the surface cannot be modeled as a continuous profile.

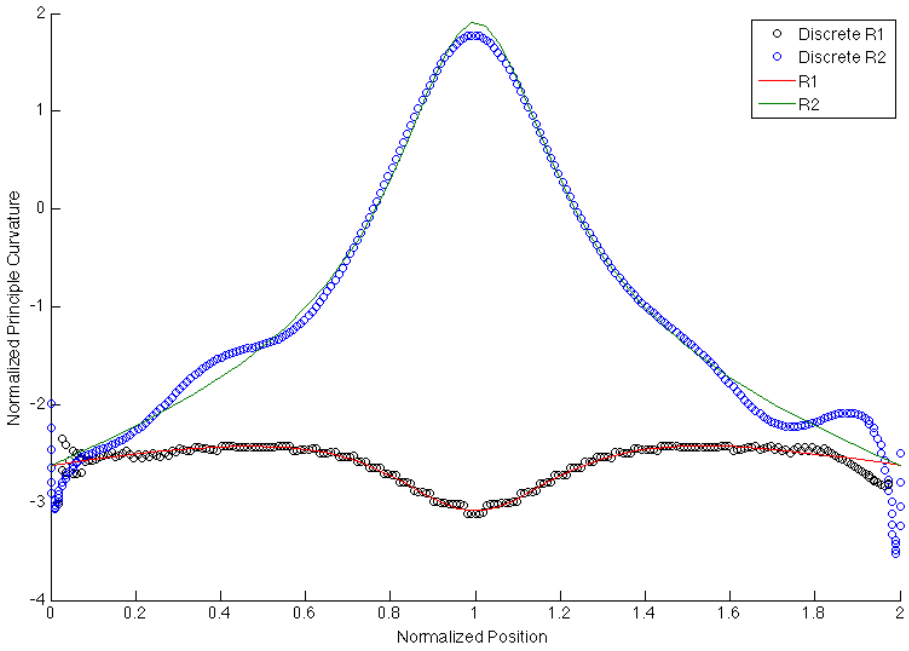


Figure 5.8. Magnitude of the two normalized principal radii of curvature for a Cassini oval (c = 0.9, d = 1) determined from an analytical parameterization (solid lines) and the discretized model (individual points). Magnitudes that are negative denote radii that point inward toward the central axis of the particle.

5.4 Development Recommendations

To improve upon the accuracy of the discrete modeling program, we could use meshes (three dimensional surface set by nodes) to model the droplet surface. In the current program, the surface is modeled by rotating the meridian profile about the droplet's central axis. Since the experimental APCE are not perfectly axisymmetric, rotating the two halves of the meridian profile independently about the central axis results in a surface that is not continuous. Consequently, we can only obtain an average local effective von Mises stress for a given segment. Additionally, if the pixelated microscopy image contains aligned pixels that make it so that the local

parameterization of the profile is flat or jagged (see Figure 5.7), the polynomial parameterization of the meridian profile results in a surface that is not continuous.

In pendant drop surface tension measurements, meridian profile images of a droplet hanging from a capillary are compared with a theoretical profile until the best fit between the meridian and the theoretical profile is obtained. The surface tension corresponding to this theoretical profile becomes the measured surface tension of the fluid [43 – 45]. The theoretical profiles are determined by simultaneously solving the differential equations given by the force balance between interfacial forces (Laplace pressure) and gravity. While iterating through several theoretical profiles works in the pendant drop case, it would not work for the APCE. This is because there is no known theoretical solution to the profile. The profile of an APCE is defined by the shape of the internal crystalline network, rather than a result opposing forces (gravity and Laplace pressure), which can be described explicitly by mathematical equations.

Though the pendant drop algorithms are not directly applicable to the APCE system, we can draw from their methods to improve the accuracy of the discrete modeling program. We could develop a program that takes an APCE image into the program, extracts the two halves of the meridian profile, and then iterate through a set of points out of the meridian profile plane plane that, when combined, result in a nodal surface that contains the two meridian profiles. Scienti et al. have previously used nodal surfaces to model droplets in emulsions, and specifically emulsion droplets' deformations in sheer flow [46]. By using the similar nodal surface parameterization methods, we could more accurately model the APCE surface. Since the surface traction and stresses are computed directly from the shape of the surface, having a more accurate surface model should improve surface traction and stress calculations.

Chapter 6

CONCLUSIONS AND FUTURE WORK

The chapter summarizes the major conclusions made throughout this work and suggestions for continued study. Since the inception of this project, the primary objective of this work has been to:

> Goal 1: Develop a method to produce soft particles with non-spherical, anisotropic shapes to deliver actives in emulsion-based products.

Development of a new technique, capillary assisted molding, successfully accomplished this goal. Chapter 2 describes the original methods used to produce anisotropic partially crystalline emulsion particles (APCE). The first half of Chapter 3 describes and depicts the three types of APCE morphologies—the spherocylinder, the ball and stick, and the dual ball and stick— produced to date using capillary assisted molding. Additionally, Chapter 3 describes a fourth morphology type, the torus, whose production is inspired by capillary assisted molding. By virtue of their anisotropy, namely their high length-to-width and high surface area-to-volume ratios, the four APCE show promise in enhanced delivery of actives in emulsion-based products.

Further investigation of APCE shows that the particles possess an unanticipated and potentially useful characteristic: the ability to change their morphology in response to external stimuli. When an APCE's local environment is altered in way that affects the crystalline network or the continuous-dispersed phase

interface, the particle deforms. Morphology stability, dynamics and evolution and their relation to the local emulsion are described in the second half of Chapter 3. Three types of external stimuli—thermal, surfactant dilution and magnetic—have been applied to APCE to date. Increasing the bulk emulsion temperature and diluting the local surfactant causes rapid deformation of APCE to spherical droplets. The utility of morphology control via eternal stimuli was demonstrated through a proof of concept deposition test. While this proof of concept test illustrated the potential benefits of using APCE in emulsion products, quantitative understanding of morphology dynamics needed to be developed to successfully incorporate APCE into products. As such, the second, ongoing goal of this work is to:

Goal 2: Develop a model to quantitatively describe the morphology of anisotropic partially crystalline emulsion droplets.

In the development of models for morphology stability, we build upon the hypothesis that APCE deformation is linked to opposing forces acting at the continuous-dispersed phase interface, namely the compressive forces exerted by the interface acting to reduce the total interfacial area of the system, and resistance of the crystalline network to deformation. Chapter 4 describes three models derived from fundamental laws of interfacial phenomena and differential geometry to describe morphological stability in APCE. The spherocylinder and axisymmetric models appear to capture qualitative trends in static APCE stability well, while the torus model shows more promise in capturing morphology dynamics.

Chapter 5 describes the preliminary work performed to quantitatively evaluate the generalized axisymmetric model's utility in predicting experimental axisymmetric particle stability. A discrete modeling program is developed to take an image of axisymmetric APCE, apply the axisymmetric force model, and determine the

magnitude and location of the segment under the maximum stress. To date, the results of the validation attempt are inconclusive. As is discussed in Chapter 5, numerical artifacts of the discrete modeling program limit our confidence in the internal stress distributions produced from experimental images in this preliminary version of the program.

There is significant opportunity to improve upon the accuracy of the discrete modeling program and potentially validate the axisymmetric model. Specifically, we suggest using more rigorous, mesh-based parameterization methods to construct the three-dimensional surface of an APCE from a two-dimensional microscopy image. Complete development of the envisioned discrete modeling program would be useful in two applications. First, the program could be used to validate or disprove the axisymmetric model as previously discussed. Second, if the axisymmetric model shows utility in predicting APCE static stability, the program could be used to take any sketch of an APCE design and determine the locations where the APCE is most likely to undergo deformation. Product engineers could use the program to rapidly evaluate APCE morphologies, specifically those that are not easily parameterized and consequently not easily defined analytically.

There is also significant opportunity to explore the potential benefits of APCE experimentally. Incorporating unique materials, such as a ferrofluid that is completely miscible in the dispersed phase or a UV initiated crystalline network, into APCE could reveal a new types of APCE response to an entirely different set of eternal stimuli. Deposition tests onto substrates of various geometries and chemistries would further determine the increased effectiveness of APCE in the delivery of active ingredients. By continuing study of anisotropic partially crystalline emulsion particles through the

three approaches—experimental, analytical, and computational—begun here, we can advance the understanding of this promising emulsion system and promote its successful incorporation into real products.

REFERENCES

1. Clauzel, M.; Johnson, E. S.; Nylander, T.; Panandiker, R. K.; Sivik, M. R.; Piculell, L. Surface deposition and phase behavior of oppositely charged polyion-surfactant ion complexes. Delivery of silicone oil emulsions to hydrophobic and hydrophilic surfaces. *ACS applied materials & interfaces* **2011**, *3*, 2451–62.

2. Champion, J.A; Katare, Y. K.; Mitragotri, S. Particle shape: a new design parameter for micro- and nanoscale drug delivery carriers. *Journal of controlled release : official journal of the Controlled Release Society* **2007**, *121*, 3–9.

3. Decuzzi, P.; Ferrari, M. The adhesive strength of non-spherical particles mediated by specific interactions. *Biomaterials* **2006**, *27*, 5307–14.

4. Larson, R. G. *The Structure and Rheology of Complex Fluids*. Oxford University Press. **1999**.

5. Philipse, A. P.; Wierenga, A. M. On the Density and Structure Formation in Gels and Clusters of Colloidal Rods and Fibers. **1998**, *7463*, 49–54.

6. Wankhade, R. P.; Pundir, A. R.; Bhalerao, S. S. Microemulsion as Novel Cosmeceutical Drug Delivery System. *IJPI ' s Journal of Pharmaceutics and Cosmetology*. **2012**, 9, 79 – 92.

7. Spicer, P. T.; Lynch, M.L; Visscher, M.; Hoath, S. Bicontinuous cubic liquid crystalline phase and cubosoume personal care delivery systems. Personal Care Delivery Systems and Formulations. Noyes Publishing. **2003**.

8. Ljungberg, L. Y. Materials selection and design for development of sustainable products. *Materials & Design* **2007**, *28*, 466–479.

9. Katifori, E.; Alben, S.; Cerda, E.; Nelson, D. R.; Dumais, J. Foldable structures and the natural design of pollen grains. *Proceedings of the National Academy of Sciences of the United States of America* **2010**, *107*, 7635–9.

10. Evans, E. A.; Skalak, R. *Mechanics and Thermodynamics of Biomembranes*. CRC Press. **1980**.

11. Kwok, R.; Evans, E. Thermoelasticity of large lecithin bilayer vesicles. *Biophysical journal* **1981**, *35*, 637–52.

12. Lee, S.; Kim, D. H.; Needham, D..Static and Dynamic Surface Tension Measurement of Microscopic Interfaces Using a Micropipet Technique: 1. A New Method for Determination of Surface Tension. *Langmuir* **2001**, 17, 5537-5543.

13. Furst, E. M. Directing colloidal assembly at fluid interfaces. *Proceedings of the National Academy of Sciences of the United States of America* **2011**, *108*, 20853–4.

14. Glotzer, S. C.; Solomon, M. J. Anisotropy of building blocks and their assembly into complex structures. *Nature materials* **2007**, *6*, 557–62.

15. Velev, O. D. A Class of Microstructured Particles Through Colloidal Crystallization. *Science* **2000**, *287*, 2240–2243.

16. Dendukuri, D.; Gu, S. S.; Pregibon, D. C.; Hatton, T. A.; Doyle, P. S. Stop-flow lithography in a microfluidic device. *Lab on a chip* **2007**, *7*, 818–28.

17. Merkel, T.; Herlihy, K.; Nunes, J.; Orgel, R. scalable, shape-specific, top-down fabrication methods for synthesis of engineered colloidal particles. *Langmuir* **2009**, *26*, 13086–13096.

18. Mitragotri, S.; Lahann, J. physical approaches to biomaterial design. *Nature materials* **2009**, *8*, 15–23.

19. Nguyen, T. D.; Jankowski, E.; Glotzer, S. C. Self-assembly and reconfigurability of shape-shifting particles. *ACS nano* **2011**, *5*, 8892–903.

20. Randall, C. L.; Leong, T. G.; Bassik, N.; Gracias, D. H. 3D lithographically fabricated nanoliter containers for drug delivery. *Advanced drug delivery reviews* **2007**, *59*, 1547–61.

21. Langer, R.; Peppas, N. *Journal of Macromolecular Science-Reviews in Macromolecular Chemistry and Physics* **1983**, 23, 61-126.

22. Khomane, R. B. Microemulsion-mediated sol-gel synthesis of mesoporous rutile TiO2 nanoneedles and its performance as anode material for Li-ion batteries. *Journal of colloid and interface science* **2011**, *356*, 369–72.

23. Manoharan, V.N.; Elsesser, M.T.; Pine, D.J.; Dense packing and symmetry in small clusters of microspheres. *Science* **2003**, 301, 483 – 487.

24. Dufresne, E. R.; Noh, H.; Saranathan, V.; Mochrie, S. G. J.; Cao, H.; Prum, R. O. Self-assembly of amorphous biophotonic nanostructures by phase separation. *Soft Matter* **2009**, *5*, 1792.

25. Blaaderen, A. Van Colloids get complex. *Nature* **2006**, *439*, 1–2.

26. Suh, S.; Yuet, K.; Hwang, D.; Bong, K.; Doyla, P.; Hatton, T.; Synthesis of Nonspherical Superparamagnetic Particles. *J Chem Soc* **2012***,* 134, 7337-7343.

27. Ramirez, L.; Smith, A.; Unal, D.; Colby, R.; Velegol, D.; Self-Assembly of Doublets from Flattened Polymer Colloids. *Langmuir* **2012** 28, 4086-4094.

28. Oh ,J.; Drumright, R.; Siegwart, D,; Matyjaszewski, K.; The development of microgels/nanogels for drug delivery applications. *Prog Polym Sci* **2008,** 33,448-477.

29. McClements, D.E.; Crystals and crystallization in oil-in-water emulsions: Implications for emulsion based delivers systems. *Adv. Colloid Interface Sci.* **2012,** 174, 1-30.

30. Rousseau, D; Fat crystals and emulsion stability: a review. *Food Research International*. **2000**, *33*, 3-14.

31. Pawar, A. B.; Caggioni, M.; Hartel, R. W.; Spicer, P. T. Arrested coalescence of viscoelastic droplets with internal microstructure. *Faraday Discussions* **2012**, *158*, 341.

32. Binks, B.P.; *Modern Aspects of Emulsion Science,* The Royal Society of Chemistry, **1998**.

33. Boode, K.; Walstra, P.; de Groot-Mostert, a. E. a. Partial coalescence in oil-in-water emulsions 2. Influence of the properties of the fat. *Colloids and Surfaces A: Physicochemical and Engineering Aspects* **1993**, *81*, 139–151.

34. Tang, D.; Marangoni, A. G. Quantitative study on the microstructure of colloidal fat crystal networks and fractal dimensions. *Advances in colloid and interface science* **2006**, *128-130*, 257–65.

35. Mukerjee, P.; Mysels, K. Critical Micelle Concentrations of Aqueous Surfactant Systems. NSRDS-NBS 36, US. Government Printing Office, Washington, D.C. **1971**.

36. Gray, A. Modern Differential Geometry of Curves and Surfaces with Mathematica. 2nd Edition. CRC Press 1998.

37. Truesdall, C.; Toupin, R. A.; The Classical Field Theories, in Handbuch der Physik (ed. S. Flu¨gge) Vol. III/1, Springer-Verlag, Berlin. **1960**.

38. Ugural; Fenster; *Advanced Mechanics of Materials and Applied Elasticity*, 5th Ed. **1998**.

39. Roark, R.J. Formulas of Stress and Strain. McGraw Hill. **1943**.

40. de With G. *Structure, Deformation and Integrity of Materials*. Wiley-VCH: New York, NY 2006.

41. Har'el Z .; Curvature of curves and surfaces – a parabolic approach. Department of Mathematics, Technion – Israel Institute of Technology. **1995**.

42. Yeh, H. Y.; Keh, H. J. Axisymmetric creeping motion of a prolate particle in a cylindrical pore. *European Journal of Mechanics - B/Fluids* **2013**, *39*, 52–58.

43. Yakhshi-Tafti, E.; Kumar, R.; Cho, H. J. Measurement of Surface Interfacial Tension as a Function of Temperature Using Pendant Drop Images. *International Journal of Optomechatronics* **2011**, *5*, 393–403.

44. Cheng, P; Li, D.; Boruka, L.; Rotengerg, Y.; Neumann, A.W. Automation of axis symmetric drop shape analysis for measurements of interfacial tensions and contact angles. *Colloids and Surfaces*. **1990**, *43*, 151-167.

45. Rotenberg, Y.; Boruvka, L.; Newman, A.W. Determination of surface tension and contact angle from the shapes of axisymmetric fluid interfaces. *Journal of Colloid and Interface Science*. **1983**, 93, 169 – 183.

46. Scienti, C.; Cunha, F. R.; Almeida, M. H. P.; Loewenberg, M. Direct Numerical simulations of Emulsion flows. *Journal of the Brazilian Society of Mechanical Sciences and Engineering.* **2003**.

Appendix A

TECHNICAL DRAWINGS FOR CUSTOM FABRICATED MICROSCOPE STAGE

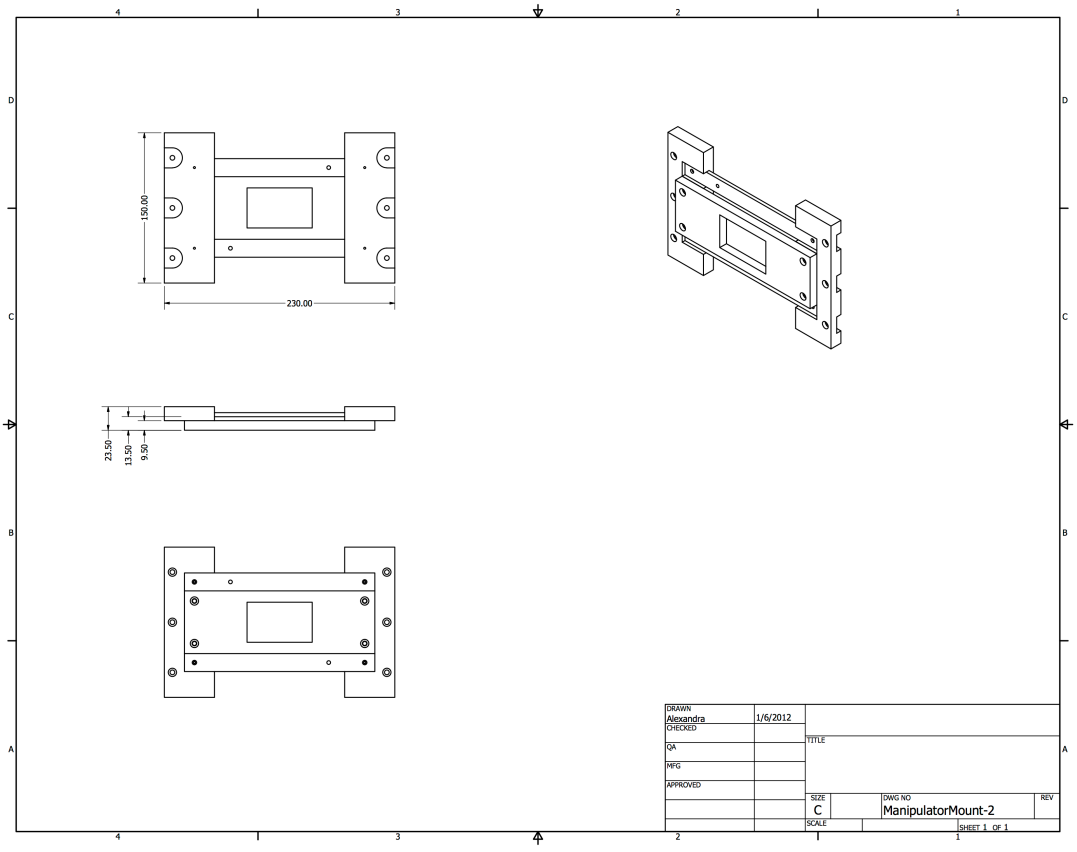


Figure A.1: Assembled 5 piece microscope stage. Note that all dimensions are in centimeters.

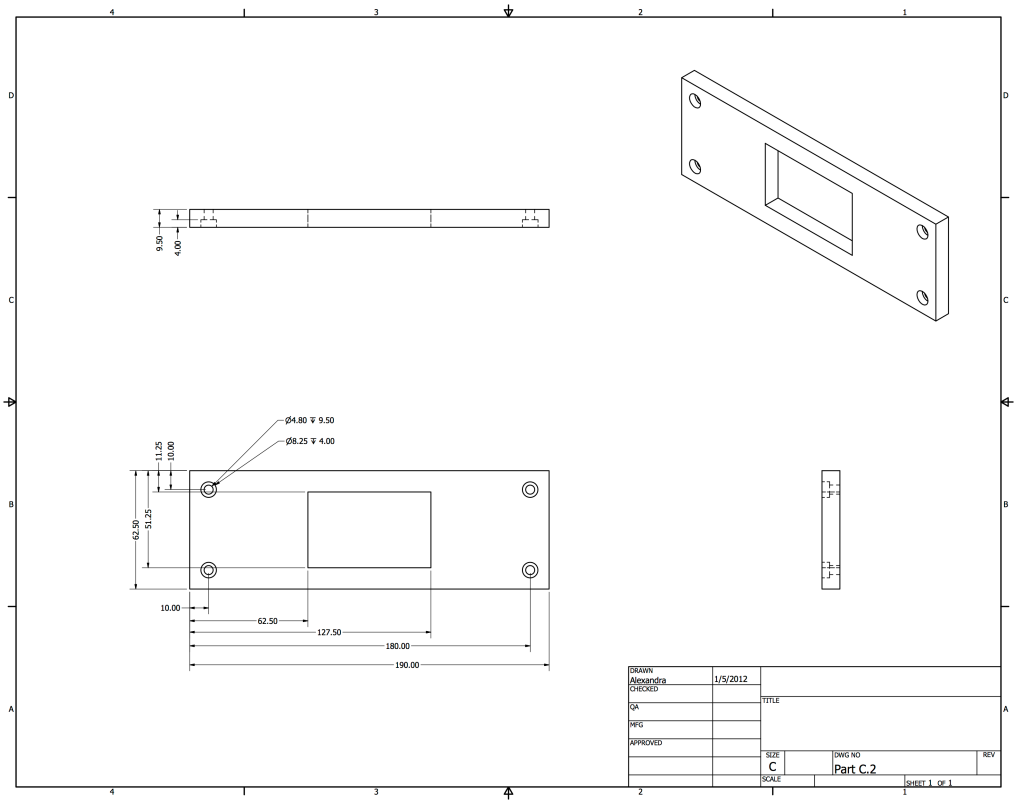


Figure A.2: Top piece (chamber support) of microscope stage. Note that all dimensions are in centimeters.

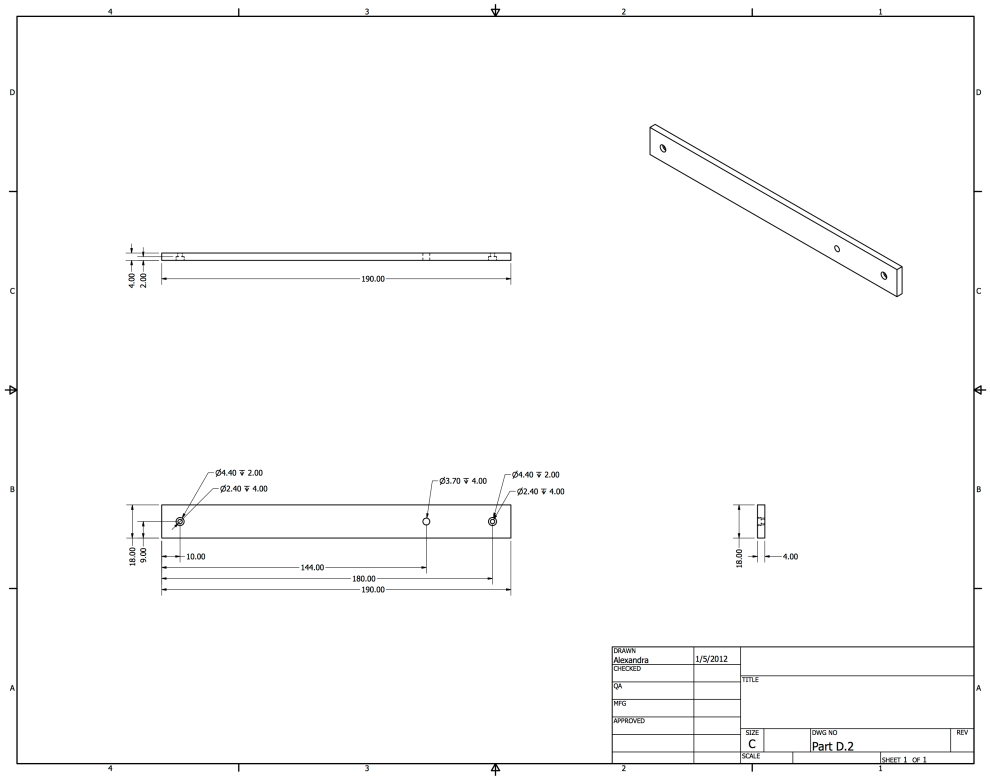


Figure A.3: Micromanipulator supports (2 pieces per stage). Note that all dimensions are in centimeters.

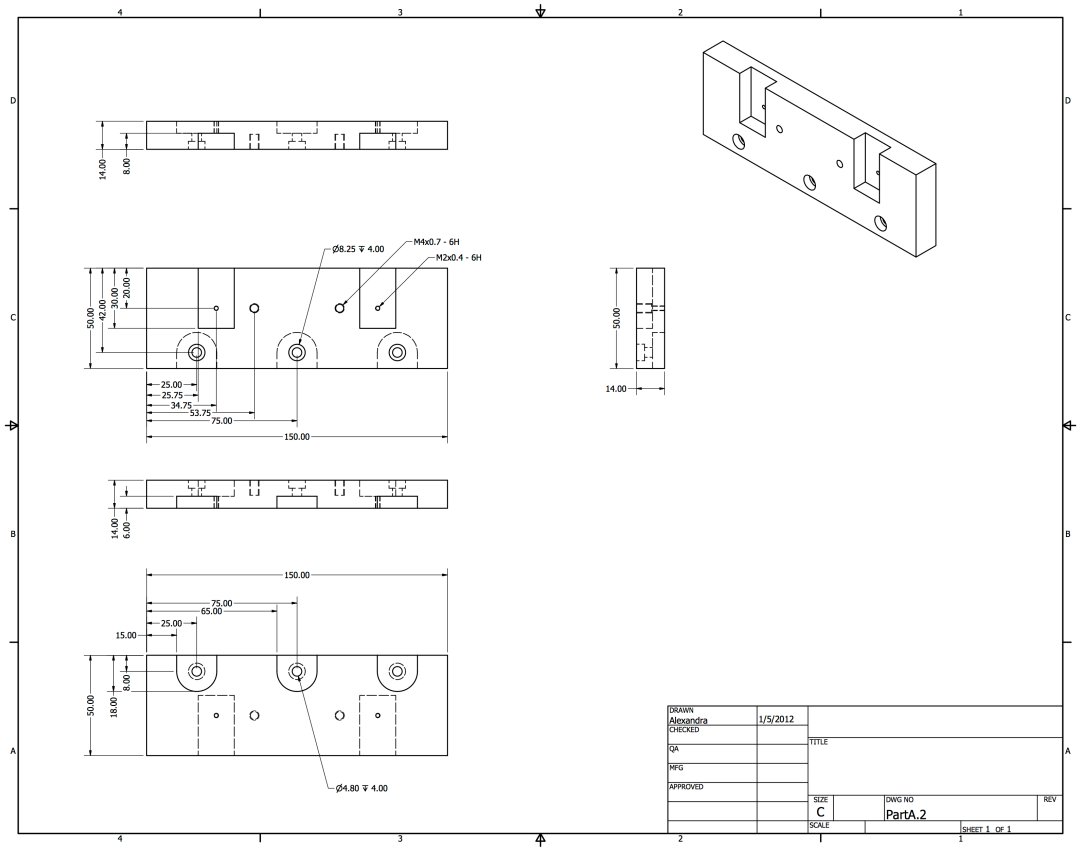


Figure A.4: Mount to microscope (2 pieces per stage). Note that all dimensions are in centimeters.

Appendix B

MATLAB FUNCTION USED TO CALCULATE SURFACE TRACTION

```
function output = surfaceTraction(textfilename, inc, polyindex, grab)
%This function takes as an input a text file of a droplet, and
outputs the
%Laplace pressure difference acting at each point in the profile.
%D is the diameter
%inc is the spacing increments for lower resolution
%polyindex is the number of points along the meridian curve included
in the
%instantaneous curvature calculation
%grab is increment for selecting points along the axis of symmetry

% % % % Section 1 % % % % % % % % % % % % % % % % % % % % % % % % % % %
% %
%Loading Text File Data and parameterizing in terms of u%
greyscalemat = load(textfilename);
[rows, columns]= size(greyscalemat);
k=1;

%Makes vectors of xy-coordinates in the image that outline the
profile
for i=1:rows
    for j = 1:columns
        if greyscalemat(i,j) == 255
            xvector(k) = j;
            yvector(k) = -i + rows;
            k=k+1;
        end
    end
end

figure (1)
plot(xvector, yvector, '.');
axis equal
division = ginput;
Aall= size(division);
a = Aall(1);
for i = 1:(a-1);
    coefdivision = polyfit(division(i:(i+1),1), division(i:(i+1),2),
1);
    divisionmat(i, :) = [coefdivision(1), coefdivision(2),
division(i,2), division((i+1),2)];
end
hold on
title('Divide')
plot(division(:, 1), division(:,2));
allpoints = [xvector', yvector'];
```

```
axis equal

% Reording vectors
k = 1;
jleft=1;
jright=1;
for i = 1: length(xvector')
    if allpoints(i, 2) < divisionmat(k,4)
        k = k+1;
    elseif allpoints(i, 2) > divisionmat(k,4)
        xtest = (allpoints(i,2) - divisionmat(k,
2))/(divisionmat(k,1));
        if allpoints(i,1) < xtest
            leftx(jleft)= allpoints(i,1);
            lefty(jleft)= allpoints(i,2);
            jleft= jleft+1;
        else
            rightx(jright)= allpoints(i,1);
            righty(jright)= allpoints(i,2);
            jright = jright + 1;
        end
    end
end

figure (2)
plot(leftx', lefty', '.b');
hold on
plot(rightx', righty', '.r');
axis equal
% flipping vectors
rightxf = fliplr(rightx);
rightyf = fliplr(righty);
allpointsordered = [leftx', lefty'; rightxf', rightyf'];

%Makes the vectors low resolution for speed
xvectorlowres = allpointsordered(1:inc:length(allpointsordered),1);
yvectorlowres = allpointsordered(1:inc:length(allpointsordered),2);


%Turns off ill-conditioned polynomial warning
warning off all

%Finding distance between points before and after turn
leftxshort = leftx(1:inc:length(leftx));
turnindex = length(leftxshort);
dist = ((xvectorlowres(turnindex+1)-
xvectorlowres(turnindex))^2+(yvectorlowres(turnindex+1)-
yvectorlowres(turnindex))^2)^(1/2);


% % % % Section 2 % % % % % % % % % % % % % % % % % % % % % % % % % % % % %
% %
```

```
%Radius of Curvature for Plane Curves (Parametric) and Normal vector%


for i = 1: (length(xvectorlowres)-polyindex)
    %Define vectors around each set of points for the regression
    xpoly = xvectorlowres(i:(i+polyindex));
    ypoly = yvectorlowres(i:(i+polyindex));
    tvec = (i:(i+polyindex))';
    coefx=polyfit(tvec, xpoly, 3);
    coefy=polyfit(tvec,ypoly, 3);

    u = ceil((polyindex+1)/2);
    v = floor((polyindex+1)/2);
    %Middle index is i+v
    rp(i+v,:) =
[3*coefx(1)*(i+v)^2+2*coefx(2)*(i+v)+coefx(3),3*coefy(1)*(i+v)^2+2*co
efy(2)*(i+v)+coefy(3), 0];
    rpp(i+v,:) =
[6*coefx(1)*(i+v)+2*coefx(2),6*coefy(1)*(i+v)+2*coefy(2), 0];
    cprod(i+v, :) = cross(rp(i+v,:), rpp(i+v,:));
    magcprod = (dot(cprod(i+v,:), cprod(i+v,:)))^(.5);
    magrp = (dot(rp(i+v,:), rp(i+v,:)))^(.5);
    %K1(i+v) = magcprod / (magrp^3);
    if cprod(i+v,3)<0
        K1(i+v) = (-1)*magcprod / (magrp^3);
    else
        K1(i+v) = magcprod / (magrp^3);
    end
    geofactor(i+v) = K1(i+v);

    %Normal Vector
    np(i+v,:) = [(-
1)*((3)*coefy(1)*(i+v)^2+2*coefy(2)*(i+v)+coefy(3)) ,
3*coefx(1)*(i+v)^2+2*coefx(2)*(i+v)+coefx(3)];
    %Making a unit normal
    nunit(i+v,:)= np(i+v,:)./( (dot(np(i+v,:), np(i+v, :))^(.5)));
    n(i+v,:) = nunit(i+v,:).*K1(i+v);

    %Sets geofactor at the turn points to be automatically 0 if the
turn
    %point is far away
    if dist > 50
        if (i > (turnindex-polyindex)) && (i<(turnindex+1))
            geofactor (i+v) = 0;
        end
    end
end

geofactor;
%Fixing ends of of geofactor

%if the ends are not connected, we set the points that are affected
```

```
to 0
distend = ((xvectorlowres(1)-
xvectorlowres(length(xvectorlowres)))^2+(yvectorlowres(1)-
yvectorlowres(length(yvectorlowres)))^2)^(1/2);
if distend > 50
    difference = length(xvectorlowres) - length (geofactor);
    originallength = length(geofactor);
    for i=1:difference
        geofactor(originallength + i) = 0;
        %note that the points at the beginning should be 0 by default
    end
else %If the ends are connected
    %Define a new set of vectors
    xvectorend = [xvectorlowres((length(xvectorlowres)-
polyindex):length(xvectorlowres)); xvectorlowres(1:polyindex)];
    yvectorend = [yvectorlowres((length(yvectorlowres)-
polyindex):length(yvectorlowres)); yvectorlowres(1:polyindex)];
    %Radius of Curvature for Plane Curves (Parametric)
    for i = 1: (length(xvectorend)-polyindex)
        %Define vectors around each set of points for the regression
        xpoly = xvectorend(i:(i+polyindex));
        ypoly = yvectorend(i:(i+polyindex));
        tvec = (i:(i+polyindex))';
        coefx=polyfit(tvec, xpoly, 3);
        coefy=polyfit(tvec,ypoly, 3);

        %Middle index is i+v
        rpend(i+v,:) =
[3*coefx(1)*(i+v)^2+2*coefx(2)*(i+v)+coefx(3),3*coefy(1)*(i+v)^2+2*co
efy(2)*(i+v)+coefy(3), 0];
        rppend(i+v,:) =
[6*coefx(1)*(i+v)+2*coefx(2),6*coefy(1)*(i+v)+2*coefy(2), 0];
        cprodend(i+v, :) = cross(rpend(i+v,:), rppend(i+v,:));
        magcprodend = (dot(cprodend(i+v,:), cprodend(i+v,:)))^(.5);
        magrpend = (dot(rpend(i+v,:), rpend(i+v,:)))^(.5);
        K1end(i+v) = magcprodend / (magrpend^3);
        geofactorend(i+v) = K1end(i+v);
    end
    %Beginning of geofactor
    for i=1:v
        geofactor(i) = geofactorend(i+polyindex+1);
    end
    for i=(length(xvectorlowres) - v):length(xvectorlowres)
        j = i - length(xvectorlowres) + polyindex +1;
        geofactor(i)=geofactorend(j);
    end

end

%Making sure normal vector is the right size
sizenormal = size(n);
difference = length(xvectorlowres) - sizenormal(1) ;
    originallength = sizenormal(1);
```

```
    for i=1:difference
        n(originallength + i,:) = [0,0];
        nunit(originallength+i, :) = [0,0];
        cprod(originallength+i,:) =[0,0,0];
    end

figure (3)
title('K1 with vector')
scatter(xvectorlowres, yvectorlowres, 50, geofactor, 'filled');
axis equal
colorbar
caxis([0,.03])

hold on
title('K1 with vector')
quiver(xvectorlowres, yvectorlowres, 100.*n(:,1), 100.*n(:,2));

% % % % Section 3 % % % % % % % % % % % % % % % % % % % % % % % % % %
% %
%Calculating Second Principle Curvature%

%Axis of symmetry point selection
figure (4)
title('Axis of Symmetry')
scatter(xvectorlowres(1:grab:length(xvectorlowres)),
yvectorlowres(1:grab:length(yvectorlowres)), 50,
geofactor(1:grab:length(xvectorlowres)), 'filled');
axis equal
colorbar
caxis([0,.03])
hold on
title('Axis of Symmetry')
quiver(xvectorlowres(1:grab:length(xvectorlowres)),
yvectorlowres(1:grab:length(yvectorlowres)),
nunit((1:grab:length(xvectorlowres)),1),
nunit((1:grab:length(xvectorlowres)), 2), 1.5);
hold on
title('Axis of Symmetry')
scatter(xvectorlowres, yvectorlowres,10, 'filled')
xvectorgrab = xvectorlowres(1:grab:length(xvectorlowres));
yvectorgrab = yvectorlowres(1:grab:length(yvectorlowres));
nunitgrab = nunit(1:grab:length(xvectorlowres), :);
first =1;
for i = 1:length(xvectorgrab)
    hold on
    title('Axis of Symmetry')
    scatter(xvectorgrab(i), yvectorgrab(i), 100)
    center = input('Input 1 if you want to designate center \n');
    if center ~=1
        [xsym(i), ysym(i)] = ginput(1);
        hold on
        title('Axis of Symmetry')
        scatter(xsym(i), ysym(i))
```

```
        gdist(i) = ((xvectorgrab(i)-xsym(i))^2+(yvectorgrab(i)-
ysym(i))^2)^(1/2);
        %finding the angle
        c = nunitgrab(i,:);
        d = [1, (yvectorgrab(i)-ysym(i))/(xvectorgrab(i)-xsym(i))];
        magc = dot(c,c)^(.5);
        magd = dot(d,d)^(.5);
        theta(i) = radtodeg(acos(dot(c,d)/(magc*magd)));
        fprintf('%g \n', theta(i));
        if theta(i) > 90
            theta(i) = 180 - theta(i);
        end
        if theta(i)>5
            K2(1+grab*(i-1)) = cos(degtorad(theta(i)))/gdist(i);
        else
            K2(1+grab*(i-1)) = 1/(gdist(i));
        end
    elseif center ==1 && first ==1
        [xcent, ycent] = ginput(1);
        gdist(i) = ((xvectorgrab(i)-xcent)^2+(yvectorgrab(i)-
ycent)^2)^(1/2);
        K2(1+grab*(i-1)) = 1/(gdist(i));
        first = 2;
    elseif center ==1
        gdist(i) = ((xvectorgrab(i)-xcent)^2+(yvectorgrab(i)-
ycent)^2)^(1/2);
        K2(1+grab*(i-1)) = 1/(gdist(i));
    end
    fprintf('%g \n \n', K2(1+grab*(i-1)));
end


%Making the complete K2
lowi = 1;
highi = 1 + grab;
m = 1;
for i=1:(length(K2)-grab)
    if K2(i) == 0
        if K2(lowi) ==0
            fprintf('warning')
        end
        K2(i) = m*(K2(highi)-K2(lowi))/grab + K2(lowi);
        m = m+1;
    elseif i~=1 && K2(i)~=0
        m=1;
        lowi = lowi + grab;
        highi = highi +grab;
    end
end
length(K2);
length(xvectorlowres);

%Making sure K2 is the right size
```

```
%sizenormal = size(n);
difference = length(xvectorlowres) - length(K2) ;
    originallength = length(K2);
    for i=1:difference
        K2(originallength + i) = 0;
        K2(originallength+i) = 0;
        %note that the points at the beginning should be 0 by default
    end

% % % % Section 4 % % % % % % % % % % % % % % % % % % % % % % % % % %
% %
%Calculating Twice the Mean curvature%

geofactor = K2+geofactor;
geofactor;
figure (5)
title('K1+K2')
scatter(xvectorlowres, yvectorlowres, 50, geofactor, 'filled');
axis equal
colorbar
caxis([0,.1])

figure (6)
title('K2')
scatter(xvectorlowres, yvectorlowres, 50, K2, 'filled');
axis equal
colorbar
caxis([0,.1])

% % % % Section 5 % % % % % % % % % % % % % % % % % % % % % % % % % %
% %
%Calculating Surface Traction%

for i = 1: (length(xvectorlowres)-v)
    n(i+v,:) = nunit(i+v,:).*geofactor(i+v);
end

figure (7)
scatter(xvectorlowres, yvectorlowres, 50, (10^4)*geofactor,
'filled');
axis equal
colorbar
caxis([-100,1200])
hold on
axis off
quiver(xvectorlowres(1:grab:length(xvectorlowres)),
yvectorlowres(1:grab:length(yvectorlowres)),
3000.*n(1:grab:length(yvectorlowres),1),
3000.*n(1:grab:length(yvectorlowres),2));
hold on
title(textfilename);
```

```
figure (8)
scatter(xvectorlowres, yvectorlowres, 50, geofactor, 'filled');
axis equal
colorbar
caxis([-.01,.1])
axis off

size(xvectorlowres);
size(yvectorlowres);
size(n);
size(geofactor);

output = [xvectorlowres, yvectorlowres,geofactor', n,];
end
```

Appendix C

MATLAB FUNCTION USED TO CALCULATE THE EFFECTIVE VON MISES STRESS

```
function avgplanevonMis(mat, grab, a)
%This function takes as an input a matrix, mat,whose columns are:
%1) the ordered x coordinate of the meridian profile, 2) the ordered
y
%coordinate of the meridian profile, 3) the magnitude of the surface
%traction, 4) the i component of the unit traction and 5) the j
component
%of the unit traction.
%It also takes the variable, grab, as an input, which is the
increment
%of points selected along the the axis of symmetry

% % % % Section 1 % % % % % % % % % % % % % % % % % % % % % % % % % % %
% %
%Reorienting system%
matdim = size(mat);

%defining vectors for consistency with surface traction function

xvectorlowres = mat(:,1);
yvectorlowres = mat(:, 2);
geofactor = mat(:, 3);
n = mat(:, 4:5);

figure (1)
scatter(xvectorlowres, yvectorlowres, 10, geofactor, 'filled');
axis equal
colorbar
caxis([-.01,.05])
hold on
axis on
quiver(xvectorlowres(1:grab:length(xvectorlowres)),
yvectorlowres(1:grab:length(yvectorlowres)),
3000.*n(1:grab:length(yvectorlowres),1),
3000.*n(1:grab:length(yvectorlowres),2));

%Defining the Axis of Symmetry
symcheck = 0;
while symcheck==0
    axissym = ginput(2);
    if axissym(1,1)~=axissym(2,1)
```

```
        coefsym = polyfit(axissym(1:2,1), axissym(1:2,2),1);
        for i = round(axissym(1,2)):round(axissym(2,2))
            sympoint(i,:)= [(i-coefsym(2))/coefsym(1), i];
        end
    else
        for i = round(axissym(1,2)):round(axissym(2,2))
            sympoint(i, :) = [axissym(1,1) ,i];
        end
    end
    hold on
    scatter(sympoint(1:length(sympoint),
1),sympoint(1:length(sympoint),2));
    symcheck = input('If you are satisfied with your axis, input 1
\n');
end

if axissym(1,1)~=axissym(2,1)
    coeforthosym = [(-1/coefsym(1)),
(1/coefsym(1))*axissym(1,1)+axissym(1,2)];
end

for i=1:400
    %orthosym(i,:) = [i, (-1/coefsym(1))*(i-
axissym(1,2))+axissym(2,2)];
    if axissym(1,1)~=axissym(2,1)
       orthosym(i,:) = [i, coeforthosym(1)*i+coeforthosym(2)];
    else
       orthosym(i,:) = [i, axissym(1,2)];
    end
end

%Reploting all points with the correct axis of symmetry and y-axis
figure (2)
scatter(xvectorlowres, yvectorlowres, 10, geofactor, 'filled');
axis equal
colorbar
caxis([-.01,.05])
hold on
quiver(xvectorlowres(1:grab:length(xvectorlowres)),
yvectorlowres(1:grab:length(yvectorlowres)),
3000.*n(1:grab:length(yvectorlowres),1),
3000.*n(1:grab:length(yvectorlowres),2));
hold on
scatter(sympoint(1:length(sympoint),
1),sympoint(1:length(sympoint),2));
hold on
scatter (orthosym(1:length(orthosym),1),
orthosym(1:length(orthosym),2));



%defining variables for the distance
```

```
asym = -1;
bsym = 1/coefsym(1);
csym = (-1)*coefsym(2)/coefsym(1);
aosym = -1;
bosym = 1/coeforthosym(1);
cosym = (-1)*coeforthosym(2)/coeforthosym(1);
bx = [1, coeforthosym(1)];
by = [1, coefsym(1)];

symindex = 0;
k=1;
m=1;

for i = 1: length(yvectorlowres)
    xtest = round((yvectorlowres(i)-coefsym(2))/coefsym(1));
    if xtest < xvectorlowres(i)
        %leftx (i) = xvectorlowres(i);
        %lefty (i) = yvectorlowres(i);
        transx(i) = abs(asym*xvectorlowres(i)
+bsym*yvectorlowres(i)+csym)/sqrt(asym^2+bsym^2);
        transy (i) = abs(aosym*xvectorlowres(i)
+bosym*yvectorlowres(i)+cosym)/sqrt(aosym^2+bosym^2);
        %transforming the vector
        transxnorm(i) = dot(n(i, 1:2), bx)/(dot(bx,bx)^(1/2));
        transynorm (i) = (a)*dot(n(i, 1:2), by)/(dot(by,by)^(1/2));
        %defining the left only
        transxl(k) = transx(i);
        transyl(k) = transy(i);
        geol(k) = geofactor(i);
        transxnorml(k) = transxnorm(i);
        transynorml(k) = transynorm(i);
        k=k+1;
    elseif xtest == xvectorlowres(i)
        symindex = i;
        transx(i) = (-1)*abs(asym*xvectorlowres(i)
+bsym*yvectorlowres(i)+csym)/sqrt(asym^2+bsym^2);
        transy(i) = abs(aosym*xvectorlowres(i)
+bosym*yvectorlowres(i)+cosym)/sqrt(aosym^2+bosym^2);
        transxnorm (i) = dot(n(i, 1:2), bx)/(dot(bx,bx)^(1/2));
        transynorm (i) = (a)*dot(n(i, 1:2), by)/(dot(by,by)^(1/2));
    elseif xtest>xvectorlowres(i)
        if symindex == 0
            symindex = i-1;
        end
        %rightx(i-symindex) = xvectorlowres(i);
        %righty(i-symindex) = yvectorlowres(i);
        transx(i) = (-1)*abs(asym*xvectorlowres(i)
+bsym*yvectorlowres(i)+csym)/sqrt(asym^2+bsym^2);
        transy(i) = abs(aosym*xvectorlowres(i)
+bosym*yvectorlowres(i)+cosym)/sqrt(aosym^2+bosym^2);
        transxnorm (i) = dot(n(i, 1:2), bx)/(dot(bx,bx)^(1/2));
        transynorm (i) = (a)*dot(n(i, 1:2), by)/(dot(by,by)^(1/2));
        %defining the right only
```

```
        transxr(m) = transx(i);
        transyr(m) = transy(i);
        transxnormr(m) = transxnorm(i);
        transynormr(m) = transynorm(i);
        m=m+1;
    end
end


%Transformed plot
figure (3)
scatter(transx, transy, 10, geofactor, 'filled');
axis equal
colorbar
caxis([-.01,.05])
hold on
quiver(transx(1:grab:length(transx)), transy(1:grab:length(transy)),
3000.*transxnorm(1:grab:length(transx)),
3000.*transynorm(1:grab:length(transx)));

% % % % Section 2 % % % % % % % % % % % % % % % % % % % % % % % % % % % %
% %
%Effective von Mises Stress from one side of the meridian profile%

%Summing for the left side (actually the right side)
%Finding the bottom of the vector to ensure correct summing
transyl;
[miny, minyind] = min(transyl);
minyind
for i = minyind: length(transxl)-(minyind+1)
    segforcel(i, :) = abs(transyl(i)-transyl(i+1)).*[
transynorml(i)*pi*abs(transxl(i)), 2*transxnorml(i)*abs(transxl(i))];
end

for i = minyind:length(segforcel)
    stressyl(i) = (2/(pi*(transxl(i)^2)))*sum(segforcel(minyind:i,
1)); %this is the sum
    %stresszl(i) = (abs(transyl(i)-
transyl(i+1))/(pi*transxl(i))^2)*sum(i:length(segforcel), 2);
    %%%%this was old%%%%% stresszl(i) = segforcel(i,
2)/(abs(transyl(i)-transyl(i+1))*(pi*transxl(i))^2); %this is not the
sum, instantaneous
    stresszl(i) = segforcel(i, 2)/(abs(transyl(i)-
transyl(i+1))*(2*abs(transxl(i))));
    if stresszl(i)<0 %if the z stress at the interface is inward,
take the difference
        vonmisl(i,:) = [10, transyl(i), sqrt(abs((stressyl(i))-
abs(stresszl(i)))^2), transxl(i), stresszl(i),
stressyl(i),abs(transyl(i)-transyl(i+1))];
    else
        vonmisl(i,:) = [10, transyl(i),
sqrt(abs((stressyl(i))+abs(stresszl(i)))^2), transxl(i), stresszl(i),
stressyl(i),abs(transyl(i)-transyl(i+1))];
```

```
    end
end

vonmisl;

figure (4)
scatter(transxl, transyl, 10)
axis equal
hold on
scatter(vonmisl(:, 1), vonmisl(:,2), 10, vonmisl(:,3), 'filled')
colorbar
%caxis([0,300])
%hold on
%quiver(transx(1:grab:length(transx)), transy(1:grab:length(transy)),
3000.*transxnorm(1:grab:length(transx)),
3000.*transynorm(1:grab:length(transx)));

% % % % Section 3 % % % % % % % % % % % % % % % % % % % % % % % % % % % %
% %
%Effective von Mises Stress for the other side of the meridian
profile%

%Summing for the right side (actually the left side)
%Finding the bottom of the vector to ensure correct summing
transyr;
[miny, minyindr] = min(transyr);
minyindr
for i = 1:(minyindr-1)
    segforcer(i, :) = abs(transyr(i)-transyr(i+1)).*[
transynormr(i)*pi*abs(transxr(i)), 2*transxnormr(i)*abs(transxr(i))];
end

for i = 1:(minyindr-1)
    stressyr(i) = (2/(pi*(transxr(i)^2)))*sum(segforcer(i:(minyindr-
1), 1)); %this is the sum
    %stresszl(i) = (abs(transyl(i)-
transyl(i+1))/(pi*transxl(i))^2)*sum(i:length(segforcel), 2);
    %%%%stresszr(i) = segforcer(i, 2)/(abs(transyr(i)-
transyr(i+1))*(pi*transxr(i))^2);
    stresszr(i) = segforcer(i, 2)/(abs(transyr(i)-
transyr(i+1))*(2*abs(transxr(i))));%this is not the sum,
instantaneous
    if stresszr(i)>0 %if the z stress at the interface is inward,
take the difference
        vonmisr(i,:) = [-10, transyr(i), sqrt(abs((stressyr(i))-
abs(stresszr(i)))^2), transxr(i), stresszr(i),
stressyr(i),(abs(transyr(i)-transyr(i+1)))];
    else
        vonmisr(i,:) = [-10, transyr(i),
sqrt(abs((stressyr(i))+abs(stresszr(i)))^2), transxr(i), stresszr(i),
stressyr(i),(abs(transyr(i)-transyr(i+1)))];
    end
end
```

```
vonmisr;
hold on
scatter(transxr, transyr, 10)
axis equal
hold on
scatter(vonmisr(:, 1), vonmisr(:,2), 10, vonmisr(:,3), 'filled')
colorbar
%caxis([0,300])
hold on
quiver(transx(1:grab:length(transx)), transy(1:grab:length(transy)),
3000.*transxnorm(1:grab:length(transx)),
3000.*transynorm(1:grab:length(transx)));


% % % % Section 4 % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % %
% %
%Plotting the average effective von Mises Stress%

stresscheck = 0;
while stresscheck == 1
    dist = 100;
    stresslocation = ginput(1);
    if stresslocation(1) < 0
        for i=1:(minyindr-1)
            distcheck = abs(vonmisr(i, 2)-stresslocation(2));
            if distcheck<dist
                dist=distcheck;
                checkind = i;
            end
        end
        fprintf('The stress at %g %g is %g \n', vonmisr(checkind, 4),
vonmisr(checkind, 2), vonmisr(checkind, 3))
    else
        for i= minyind:length(segforcel)
            distcheck = abs(vonmisl(i, 2)-stresslocation(2));
            if distcheck<dist
                dist=distcheck;
                checkind = i;
            end
        end
        fprintf('The stress at %g %g is %g \n', vonmisl(checkind, 4),
vonmisl(checkind, 2), vonmisl(checkind, 3))
    end

    stresscheck = input('To choose again press 1 \n');
end

stresschecksum = 0;
while stresschecksum == 1
    dist = 100;
    stresslocation = ginput(2);
```

```
    if stresslocation(1,1) < 0
        for i=1:(minyindr-1)
            distcheck = abs(vonmisr(i, 2)-stresslocation(1,2));
            if distcheck<dist
                dist=distcheck;
                checkindr = i;
            end
        end
        fprintf('The left stress at %g %g is %g \n z is %g y is %g
\n', vonmisr(checkindr, 4), vonmisr(checkindr, 2), vonmisr(checkindr,
3), vonmisr(checkindr, 5), vonmisr(checkindr,6))
    end
    dist = 100;
    if stresslocation(2,1)>0
        for i= minyind:length(segforcel)
            distcheck = abs(vonmisl(i, 2)-stresslocation(2,2));
            if distcheck<dist
                dist=distcheck;
                checkindl = i;
            end
        end
        fprintf('The right stress at %g %g is %g \n z is %g y is %g
\n', vonmisl(checkindl, 4), vonmisl(checkindl, 2), vonmisl(checkindl,
3), vonmisl(checkindl, 5), vonmisl(checkindl, 6) )
    end

    %making sure profile stresses have the correct sign

    if vonmisl(checkindl, 4)*vonmisl(checkindl,5)<0
        fprintf('The right force in inward \n')
    end
    if vonmisr(checkindr, 4)*vonmisr(checkindr, 5)>0
        fprintf('The left force is inward \n');
    end
    %The summed z stress will be positive if inward
    stresszsum = (abs(vonmisl(checkindl,
4)*vonmisl(checkindl,5))*vonmisl(checkindl, 7)+ vonmisr(checkindr,
4)*vonmisr(checkindr, 5))*vonmisr(checkindr, 7) /
(abs(vonmisl(checkindl,4))*vonmisl(checkindl,
7)+abs(vonmisr(checkindr,4))*vonmisr(checkindr, 7))

    stressysum = (vonmisl(checkindl, 6)*vonmisl(checkindl,
4)^2+vonmisr(checkindr, 6)*vonmisr(checkindr,
4)^2)/(vonmisl(checkindl, 4)^2+vonmisr(checkindr, 4)^2)
    totalvonmis = sqrt( (abs(stressysum) - stresszsum)^2);
    fprintf('The total vonmis stress at y = %g %g is %g \n',
vonmisl(checkindl,2), vonmisr(checkindr, 2), totalvonmis);
    fprintf('Scaled, this is %g \n', totalvonmis*30*10^3);
    %totalvonmis*30*10^3
    stresschecksum = input('To choose again press 1 \n');
end
```

```
%plotting the discrete vonmises stress
transyl;
transyl(length(transyl));
transyl(1);
yplot = transyl(minyind):2:transyl(length(transyl));
for j = 1:length(yplot)
    stresschecksum = 1;
    stresslocation = [-10, yplot(j); 10, yplot(j)];
    while stresschecksum == 1
        dist = 100;
        %stresslocation = [-10, yplot(j); 10, yplot(j)];
        if stresslocation(1,1) < 0
            for i=1:(minyindr-1)
                distcheck = abs(vonmisr(i, 2)-stresslocation(1,2));
                if distcheck<dist
                    dist=distcheck;
                    checkindr = i;
                end
            end
            fprintf('The left stress at %g %g is %g \n z is %g y is
%g \n', vonmisr(checkindr, 4), vonmisr(checkindr, 2),
vonmisr(checkindr, 3), vonmisr(checkindr, 5), vonmisr(checkindr,6))
        end
        dist = 100;
        if stresslocation(2,1)>0
            for i= minyind:length(segforcel)
                distcheck = abs(vonmisl(i, 2)-stresslocation(2,2));
                if distcheck<dist
                    dist=distcheck;
                    checkindl = i;
                end
            end
            fprintf('The right stress at %g %g is %g \n z is %g y is
%g \n', vonmisl(checkindl, 4), vonmisl(checkindl, 2),
vonmisl(checkindl, 3), vonmisl(checkindl, 5), vonmisl(checkindl, 6) )
        end
        stresszsum = (abs(vonmisl(checkindl,
4)*vonmisl(checkindl,5))*vonmisl(checkindl, 7)+ vonmisr(checkindr,
4)*vonmisr(checkindr, 5))*vonmisr(checkindr, 7) /
(abs(vonmisl(checkindl,4))*vonmisl(checkindl,
7)+abs(vonmisr(checkindr,4))*vonmisr(checkindr, 7));

        stressysum = (vonmisl(checkindl, 6)*vonmisl(checkindl,
4)^2+vonmisr(checkindr, 6)*vonmisr(checkindr,
4)^2)/(vonmisl(checkindl, 4)^2+vonmisr(checkindr, 4)^2);
        totalvonmis(j) = sqrt( (abs(stressysum) - stresszsum)^2);
        totalvonmisscale(j)=sqrt(3)*totalvonmis(j)*10^4;
        stresschecksum = 0;
    end
end
```

```
[totalvonmisscale', yplot']

figure (5)
title('totalvonmis')
%scatter(vonmisr(:,2), vonmisr(:,3))
scatter(yplot, totalvonmisscale)
%axis([0 300 0 .05])
```

Appendix D

THEORETICAL COMPUTATIONS IN MAPLE

See supplementary PDF.

Alexandra Bayles
Supporting Maple Files: Honors Senior Thesis

Spherocylinder Figure Generation

```
restart;
with(LinearAlgebra) :
 with(VectorCalculus) :
 with(plots) :
g := t;
 R := 1;
 l := 7;
f := piecewise(t > 0 and t < R, sqrt(R^2 - (g - R)^2), t ≥ R and t ≤ R + l, R, t > R + l and t
     < 2·R + l, sqrt(R^2 - (g - (R + l))^2) );
```

$$t$$

$$1$$

$$7$$

$$\begin{cases} \sqrt{2\,t - t^2} & 0 < t \text{ and } t < 1 \\ 1 & 1 \le t \text{ and } t \le 8 \\ \sqrt{-63 + 16\,t - t^2} & 8 < t \text{ and } t < 9 \end{cases} \tag{1.1}$$

```
P3D := plot3d( [f·cos(theta), g, f·sin(theta)], t = 0 ..2· R + l , theta = 0 .. 2· Pi, axes = none,
     scaling = constrained, color = blue, grid = [60, 20]);
 P0 := plot( [f, -f], t = 0 ..2· R + l , axes = none, scaling = constrained, color = blue);
```

$$PLOT3D(\dots)$$

$$PLOT(\dots) \tag{1.2}$$

```
P1 := arrow( {[1, R], [2, R], [3, R], [4, R], [5, R], [6, R], [7, R], [8, R]} , {[0, -1/(2·R)]}, shape
     = arrow, color = green );
P2 := arrow( {[1, -1· R], [2, -1· R], [3, -1·R], [4, -1·R], [5, -1·R], [6, -1·R], [7, -1·R], [8, -1
     ·R], } , {[0, 1/(2·R)]}, shape = arrow, color = green );
P3 := arrow( {[0, 0]}, {[R, 0]}, shape = arrow, color = red); P4 := arrow( {[2·R + l, 0]}, {[
     -R, 0]}, shape = arrow, color = red);
P5 := arrow( {[R + sqrt(2)/2 ·R + l, sqrt(2)/2 ·R]}, {[ -sqrt(2)/2 · (1/R), -sqrt(2)/2 · (1/R) ]},
     shape = arrow, color = red ); P6 := arrow( {[R + sqrt(2)/2 ·R + l, -sqrt(2)/2 ·R]},
```

$$\left\{\left[\frac{-\mathrm{sqrt}(2)}{2}\cdot\left(\frac{1}{R}\right),\ \frac{\mathrm{sqrt}(2)}{2}\cdot\left(\frac{1}{R}\right)\right]\right\},\ shape = arrow,\ color = red\Bigg);$$

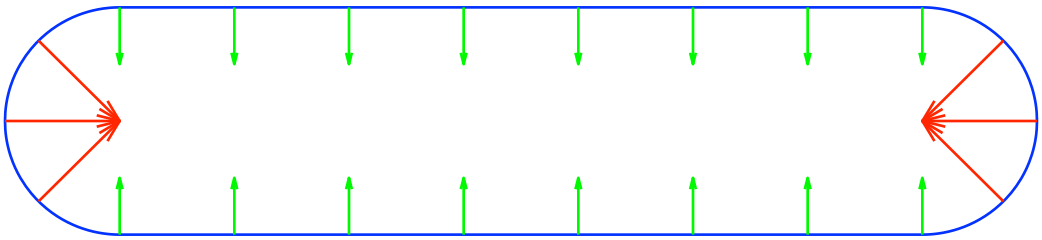
$$P7 := arrow\left(\left\{\left[\left(1-\frac{\mathrm{sqrt}(2)}{2}\cdot R\right),\ \frac{\mathrm{sqrt}(2)}{2}\cdot R\right]\right\},\ \left\{\left[\frac{\mathrm{sqrt}(2)}{2}\cdot\left(\frac{1}{R}\right),\ \frac{-\mathrm{sqrt}(2)}{2}\cdot\left(\frac{1}{R}\right)\right]\right\},\ shape = arrow,\ color = red\right);$$

$$P8 := arrow\left(\left\{\left[\left(1-\frac{\mathrm{sqrt}(2)}{2}\cdot R\right),\ \frac{-\mathrm{sqrt}(2)}{2}\cdot R\right]\right\},\ \left\{\left[\frac{\mathrm{sqrt}(2)}{2}\cdot\left(\frac{1}{R}\right),\ \frac{\mathrm{sqrt}(2)}{2}\cdot\left(\frac{1}{R}\right)\right]\right\},\ shape = arrow,\ color = red\right);$$

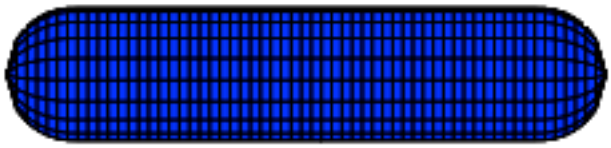
$$PLOT(\ldots)$$
$$PLOT(\ldots)$$
$$PLOT(\ldots)$$
$$PLOT(\ldots)$$
$$PLOT(\ldots)$$
$$PLOT(\ldots)$$
$$PLOT(\ldots)$$

(1.3)

$$display(P0, P1, P2, P3, P4, P5, P6, P7, P8);$$
$$display(P0, P3, P4);$$



display(*P3D*)



Cassini Oval Figure Generation 1

```
restart;
 with(LinearAlgebra) :
 with(VectorCalculus) :
 with(plots);
```

[*animate*, *animate3d*, *animatecurve*, *arrow*, *changecoords*, *complexplot*, *complexplot3d*, *conformal*, *conformal3d*, *contourplot*, *contourplot3d*, *coordplot*, *coordplot3d*, *densityplot*, *display*, *dualaxisplot*, *fieldplot*, *fieldplot3d*, *gradplot*, *gradplot3d*, *implicitplot*, *implicitplot3d*, *inequal*, *interactive*, *interactiveparams*, *intersectplot*, *listcontplot*, *listcontplot3d*, *listdensityplot*, *listplot*, *listplot3d*, *loglogplot*, *logplot*, *matrixplot*, *multiple*, *odeplot*, *pareto*, *plotcompare*, *pointplot*, *pointplot3d*, *polarplot*, *polygonplot*, *polygonplot3d*, *polyhedra_supported*, *polyhedraplot*, *rootlocus*, *semilogplot*, *setcolors*, *setoptions*, *setoptions3d*, *spacecurve*, *sparsematrixplot*, *surfdata*, *textplot*, *textplot3d*, *tubeplot*] **(2.1)**

```
c := .95;
```

$$d := 1;$$
$$M := 2 \cdot c^2 \cdot \cos(2 \cdot t) + 2 \cdot \mathrm{sqrt}\left(\left(-c^4 + d^4 \right) + c^4 \cdot (\cos(2 \cdot t))^2 \right);$$
$$g := \cos(t) \cdot \mathrm{sqrt}\left(\frac{M}{2} \right);$$
$$R := 1;$$
$$l := 5;$$
$$f := \sin(t) \cdot \mathrm{sqrt}\left(\frac{M}{2} \right);$$
$$evalf\left(subs\left(\left\{ t = \frac{\mathrm{Pi}}{4} \right\}, f \right) \right);$$

$$0.95$$

$$1$$

$$1.8050 \cos(2\,t) + 2 \sqrt{0.18549375 + 0.81450625 \cos(2\,t)^2}$$

$$\cos(t) \sqrt{0.9025000000 \cos(2\,t) + \sqrt{0.18549375 + 0.81450625 \cos(2\,t)^2}}$$

$$1$$

$$5$$

$$\sin(t) \sqrt{0.9025000000 \cos(2\,t) + \sqrt{0.18549375 + 0.81450625 \cos(2\,t)^2}}$$

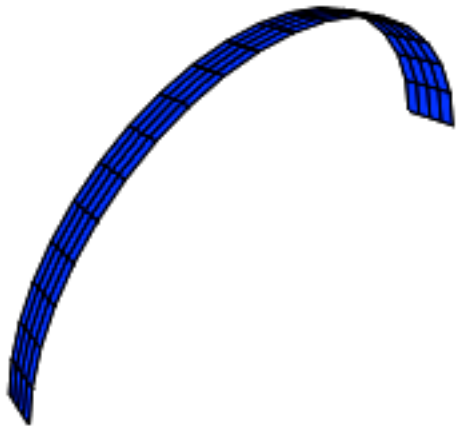
$$0.4640527198 \qquad \textbf{(2.2)}$$

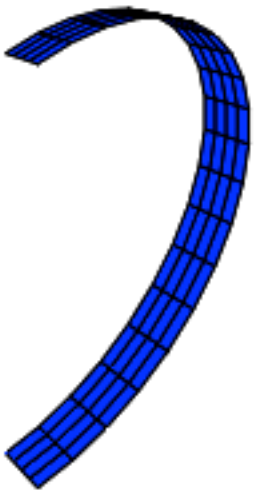
$$plot3d(\ [\, f \cdot \cos(\mathrm{theta}), g, f \cdot \sin(\mathrm{theta}) \,], t = 0 \,..\, 2 \cdot \mathrm{Pi}, \mathrm{theta} = 0 \,..\, \mathrm{Pi}, axes = none, scaling = constrained, grid = [60, 20], color = blue, \ );$$

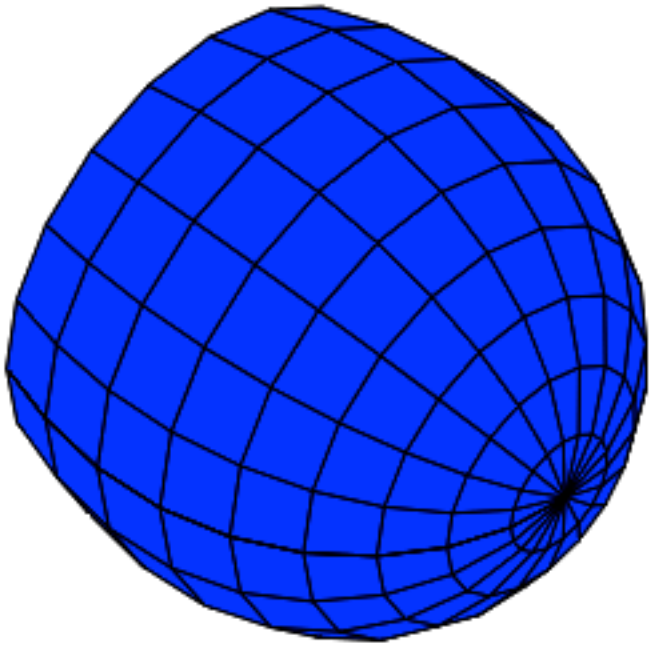
$$plot3d\left(\ [\, f \cdot \cos(\mathrm{theta}), g, f \cdot \sin(\mathrm{theta}) \,], t = \frac{\mathrm{Pi}}{3.75} \,..\, \frac{\mathrm{Pi}}{4}, \mathrm{theta} = 0 \,..\, \mathrm{Pi}, axes = none, scaling = constrained, color = blue, grid = [5, 20] \right);$$

$$plot3d\left(\ [\, f \cdot \cos(\mathrm{theta}), g, f \cdot \sin(\mathrm{theta}) \,], t = \frac{\mathrm{Pi}}{3.75} \,..\, \frac{\mathrm{Pi}}{4}, \mathrm{theta} = \frac{\mathrm{Pi}}{2} \,..\, \frac{3 \cdot \mathrm{Pi}}{2}, axes = none, scaling = constrained, color = blue, grid = [5, 20] \right);$$

$$plot3d\left(\ [\, f \cdot \cos(\mathrm{theta}), g, f \cdot \sin(\mathrm{theta}) \,], t = 0 \,..\, \frac{\mathrm{Pi}}{4}, \mathrm{theta} = 0 \,..\, 2 \cdot \mathrm{Pi}, axes = none, scaling = constrained, color = blue, grid = [10, 20] \right);$$







Discrete Modeling Accuracy: Analytical Curvature of Cassini Oval

```
restart;
 with(LinearAlgebra) :
 with(VectorCalculus) :
 with(plots);
```

(3.1)

```
[animate, animate3d, animatecurve, arrow, changecoords, complexplot, complexplot3d,
    conformal, conformal3d, contourplot, contourplot3d, coordplot, coordplot3d, densityplot,
    display, dualaxisplot, fieldplot, fieldplot3d, gradplot, gradplot3d, implicitplot,
    implicitplot3d, inequal, interactive, interactiveparams, intersectplot, listcontplot,
    listcontplot3d, listdensityplot, listplot, listplot3d, loglogplot, logplot, matrixplot, multiple,
    odeplot, pareto, plotcompare, pointplot, pointplot3d, polarplot, polygonplot,
    polygonplot3d, polyhedra_supported, polyhedraplot, rootlocus, semilogplot, setcolors,
    setoptions, setoptions3d, spacecurve, sparsematrixplot, surfdata, textplot, textplot3d,
    tubeplot]
c := .9; #defines the shape of the oval
```

$d := 1;$

$M := 2 \cdot c^2 \cdot \cos(2 \cdot t) + 2 \cdot \mathrm{sqrt}\left(\left(-c^4 + d^4 \right) + c^4 \cdot (\cos(2 \cdot t))^2 \right);$

$g := \cos(t) \cdot \mathrm{sqrt}\left(\frac{M}{2} \right);$

$R := 1;$

$l := 5;$

$f := \sin(t) \cdot \mathrm{sqrt}\left(\frac{M}{2} \right);$

$evalf\left(subs\left(\left\{ t = \frac{\mathrm{Pi}}{4} \right\}, f \right) \right);$

$$0.9$$

$$1$$

$$1.62 \cos(2\,t) + 2\sqrt{0.3439 + 0.6561 \cos(2\,t)^2}$$

$$\cos(t) \sqrt{0.8100000000 \cos(2\,t) + \sqrt{0.3439 + 0.6561 \cos(2\,t)^2}}$$

$$1$$

$$5$$

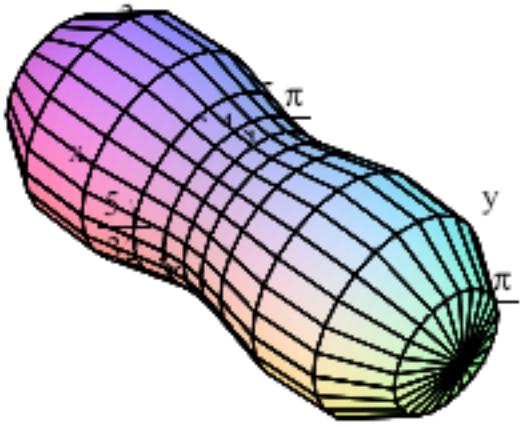
$$\sin(t) \sqrt{0.8100000000 \cos(2\,t) + \sqrt{0.3439 + 0.6561 \cos(2\,t)^2}}$$

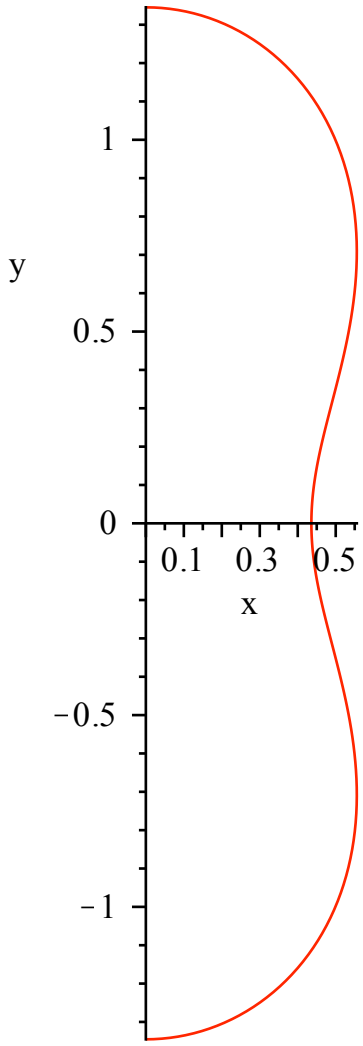
$$0.5414932485 \qquad \textbf{(3.2)}$$

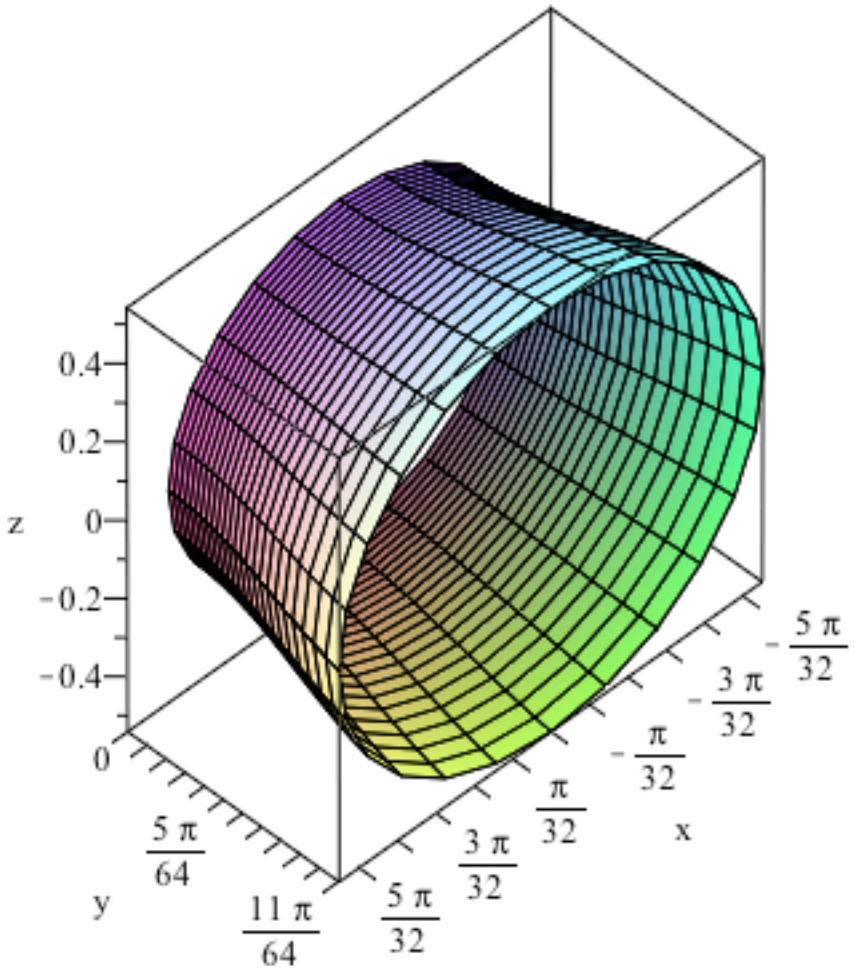
$plot3d([f \cdot \cos(\mathrm{theta}), g, f \cdot \sin(\mathrm{theta})], t = 0 .. 2 \cdot \mathrm{Pi}, \mathrm{theta} = 0 .. 2 \cdot \mathrm{Pi}, axes = normal, scaling = constrained, labels = ["x", "y", "z"]);$

$plot([f, g, t = 0 .. \mathrm{Pi}], axes = normal, scaling = constrained, labels = ["x", "y"]);$

$plot3d\left([f \cdot \cos(\mathrm{theta}), g, f \cdot \sin(\mathrm{theta})], t = \frac{\mathrm{Pi}}{4} .. \frac{\mathrm{Pi}}{2}, \mathrm{theta} = 0 .. 2 \cdot \mathrm{Pi}, axes = boxed, scaling = constrained, labels = ["x", "y", "z"] \right);$


π
y
π


y
1
0.5
0
0.1
0.3
0.5
x
−0.5
−1



$fp := diff(f, t)$:

$fpp := diff(fp, t)$:

$gp := diff(g, t)$:

$gpp := diff(gp, t)$:

$rp := \langle fp, gp, 0 \rangle$:

$rpp := \langle fpp, gpp, 0 \rangle$:

$cprod := CrossProduct(rp, rpp)$:

$magcprod := DotProduct(cprod, cprod)^{.5}$:

$magrp := DotProduct(rp, rp)^{.5}$:

$$k1 := \frac{\frac{cprod(3)}{\mathrm{abs}(cprod(3))} \cdot magcprod}{magrp^3};$$

$$\left(\left(\left(\cos(t)\sqrt{0.8100000000\cos(2t) + \sqrt{0.3439 + 0.6561\cos(2t)^2}}\right.\right.\right. \tag{3.3}$$

$$+\frac{1}{2}\frac{\sin(t)\left(-1.620000000\sin(2t)-\frac{1.312200000\cos(2t)\sin(2t)}{\sqrt{0.3439+0.6561\cos(2t)^2}}\right)}{\sqrt{0.8100000000\cos(2t)+\sqrt{0.3439+0.6561\cos(2t)^2}}}\Bigg)\Bigg($$

$$-\cos(t)\sqrt{0.8100000000\cos(2t)+\sqrt{0.3439+0.6561\cos(2t)^2}}$$

$$-\frac{\sin(t)\left(-1.620000000\sin(2t)-\frac{1.312200000\cos(2t)\sin(2t)}{\sqrt{0.3439+0.6561\cos(2t)^2}}\right)}{\sqrt{0.8100000000\cos(2t)+\sqrt{0.3439+0.6561\cos(2t)^2}}}$$

$$-\frac{1}{4}\frac{\cos(t)\left(-1.620000000\sin(2t)-\frac{1.312200000\cos(2t)\sin(2t)}{\sqrt{0.3439+0.6561\cos(2t)^2}}\right)^2}{\left(0.8100000000\cos(2t)+\sqrt{0.3439+0.6561\cos(2t)^2}\right)^{3/2}}$$

$$+\frac{1}{2}\Bigg(\cos(t)\Bigg(-3.240000000\cos(2t)-\frac{1.721868840\cos(2t)^2\sin(2t)^2}{\left(0.3439+0.6561\cos(2t)^2\right)^{3/2}}$$

$$+\frac{2.624400000\sin(2t)^2}{\sqrt{0.3439+0.6561\cos(2t)^2}}-\frac{2.624400000\cos(2t)^2}{\sqrt{0.3439+0.6561\cos(2t)^2}}\Bigg)\Bigg)\Bigg/$$

$$\sqrt{0.8100000000\cos(2t)+\sqrt{0.3439+0.6561\cos(2t)^2}}\Bigg)-\Bigg($$

$$-\sin(t)\sqrt{0.8100000000\cos(2t)+\sqrt{0.3439+0.6561\cos(2t)^2}}$$

$$+\frac{1}{2}\,\frac{\cos(t)\left(-1.620000000\sin(2\,t)-\dfrac{1.312200000\cos(2\,t)\sin(2\,t)}{\sqrt{0.3439+0.6561\cos(2\,t)^2}}\right)}{\sqrt{0.8100000000\cos(2\,t)+\sqrt{0.3439+0.6561\cos(2\,t)^2}}}\Bigg)\Bigg($$

$$-\sin(t)\sqrt{0.8100000000\cos(2\,t)+\sqrt{0.3439+0.6561\cos(2\,t)^2}}$$

$$+\frac{\cos(t)\left(-1.620000000\sin(2\,t)-\dfrac{1.312200000\cos(2\,t)\sin(2\,t)}{\sqrt{0.3439+0.6561\cos(2\,t)^2}}\right)}{\sqrt{0.8100000000\cos(2\,t)+\sqrt{0.3439+0.6561\cos(2\,t)^2}}}$$

$$-\frac{1}{4}\,\frac{\sin(t)\left(-1.620000000\sin(2\,t)-\dfrac{1.312200000\cos(2\,t)\sin(2\,t)}{\sqrt{0.3439+0.6561\cos(2\,t)^2}}\right)^2}{\left(0.8100000000\cos(2\,t)+\sqrt{0.3439+0.6561\cos(2\,t)^2}\right)^{3/2}}$$

$$+\frac{1}{2}\left(\sin(t)\left(-3.240000000\cos(2\,t)-\frac{1.721868840\cos(2\,t)^2\sin(2\,t)^2}{\left(0.3439+0.6561\cos(2\,t)^2\right)^{3/2}}\right.\right.$$

$$\left.\left.+\frac{2.624400000\sin(2\,t)^2}{\sqrt{0.3439+0.6561\cos(2\,t)^2}}-\frac{2.624400000\cos(2\,t)^2}{\sqrt{0.3439+0.6561\cos(2\,t)^2}}\right)\right)\Bigg/$$

$$\left.\sqrt{0.8100000000\cos(2\,t)+\sqrt{0.3439+0.6561\cos(2\,t)^2}}\right)\Bigg)$$

$$\Bigg(\Bigg(\Bigg(\cos(t)\sqrt{0.8100000000\cos(2\,t)+\sqrt{0.3439+0.6561\cos(2\,t)^2}}$$

$$+\frac{1}{2}\frac{\sin(t)\left(-1.620000000\sin(2t)-\frac{1.312200000\cos(2t)\sin(2t)}{\sqrt{0.3439+0.6561\cos(2t)^2}}\right)}{\sqrt{0.8100000000\cos(2t)+\sqrt{0.3439+0.6561\cos(2t)^2}}}\Bigg)\Bigg($$

$$-\cos(t)\sqrt{0.8100000000\cos(2t)+\sqrt{0.3439+0.6561\cos(2t)^2}}$$

$$-\frac{\sin(t)\left(-1.620000000\sin(2t)-\frac{1.312200000\cos(2t)\sin(2t)}{\sqrt{0.3439+0.6561\cos(2t)^2}}\right)}{\sqrt{0.8100000000\cos(2t)+\sqrt{0.3439+0.6561\cos(2t)^2}}}$$

$$-\frac{1}{4}\frac{\cos(t)\left(-1.620000000\sin(2t)-\frac{1.312200000\cos(2t)\sin(2t)}{\sqrt{0.3439+0.6561\cos(2t)^2}}\right)^2}{\left(0.8100000000\cos(2t)+\sqrt{0.3439+0.6561\cos(2t)^2}\right)^{3/2}}$$

$$+\frac{1}{2}\Bigg(\cos(t)\Bigg(-3.240000000\cos(2t)-\frac{1.721868840\cos(2t)^2\sin(2t)^2}{\left(0.3439+0.6561\cos(2t)^2\right)^{3/2}}$$

$$+\frac{2.624400000\sin(2t)^2}{\sqrt{0.3439+0.6561\cos(2t)^2}}-\frac{2.624400000\cos(2t)^2}{\sqrt{0.3439+0.6561\cos(2t)^2}}\Bigg)\Bigg)\Bigg/$$

$$\sqrt{0.8100000000\cos(2t)+\sqrt{0.3439+0.6561\cos(2t)^2}}\Bigg)-\Bigg($$

$$-\sin(t)\sqrt{0.8100000000\cos(2t)+\sqrt{0.3439+0.6561\cos(2t)^2}}$$

$$+\frac{1}{2}\,\frac{\cos(t)\left(-1.620000000\sin(2\,t)-\dfrac{1.312200000\cos(2\,t)\sin(2\,t)}{\sqrt{0.3439+0.6561\cos(2\,t)^2}}\right)}{\sqrt{0.8100000000\cos(2\,t)+\sqrt{0.3439+0.6561\cos(2\,t)^2}}}\Bigg)\Bigg($$

$$-\sin(t)\sqrt{0.8100000000\cos(2\,t)+\sqrt{0.3439+0.6561\cos(2\,t)^2}}$$

$$+\frac{\cos(t)\left(-1.620000000\sin(2\,t)-\dfrac{1.312200000\cos(2\,t)\sin(2\,t)}{\sqrt{0.3439+0.6561\cos(2\,t)^2}}\right)}{\sqrt{0.8100000000\cos(2\,t)+\sqrt{0.3439+0.6561\cos(2\,t)^2}}}$$

$$-\frac{1}{4}\,\frac{\sin(t)\left(-1.620000000\sin(2\,t)-\dfrac{1.312200000\cos(2\,t)\sin(2\,t)}{\sqrt{0.3439+0.6561\cos(2\,t)^2}}\right)^2}{\left(0.8100000000\cos(2\,t)+\sqrt{0.3439+0.6561\cos(2\,t)^2}\right)^{3/2}}$$

$$+\frac{1}{2}\left(\sin(t)\left(-3.240000000\cos(2\,t)-\frac{1.721868840\cos(2\,t)^2\sin(2\,t)^2}{\left(0.3439+0.6561\cos(2\,t)^2\right)^{3/2}}\right.\right.$$

$$\left.\left.+\frac{2.624400000\sin(2\,t)^2}{\sqrt{0.3439+0.6561\cos(2\,t)^2}}-\frac{2.624400000\cos(2\,t)^2}{\sqrt{0.3439+0.6561\cos(2\,t)^2}}\right)\right)\Bigg/$$

$$\left.\sqrt{0.8100000000\cos(2\,t)+\sqrt{0.3439+0.6561\cos(2\,t)^2}}\right)\Bigg)^2\Bigg)^{0.5}\Bigg)\Bigg/\Bigg(\Bigg|$$

$$-\left(\cos(t)\sqrt{0.8100000000\cos(2\,t)+\sqrt{0.3439+0.6561\cos(2\,t)^2}}\right.$$

$$+\frac{1}{2}\,\frac{\sin(t)\left(-1.620000000\sin(2\,t)-\dfrac{1.312200000\cos(2\,t)\sin(2\,t)}{\sqrt{0.3439+0.6561\cos(2\,t)^2}}\right)}{\sqrt{0.8100000000\cos(2\,t)+\sqrt{0.3439+0.6561\cos(2\,t)^2}}}\Bigg)\Bigg($$

$$-\cos(t)\sqrt{0.8100000000\cos(2t)+\sqrt{0.3439+0.6561\cos(2t)^2}}$$

$$-\frac{\sin(t)\left(-1.620000000\sin(2t)-\dfrac{1.312200000\cos(2t)\sin(2t)}{\sqrt{0.3439+0.6561\cos(2t)^2}}\right)}{\sqrt{0.8100000000\cos(2t)+\sqrt{0.3439+0.6561\cos(2t)^2}}}$$

$$-\frac{1}{4}\frac{\cos(t)\left(-1.620000000\sin(2t)-\dfrac{1.312200000\cos(2t)\sin(2t)}{\sqrt{0.3439+0.6561\cos(2t)^2}}\right)^2}{\left(0.8100000000\cos(2t)+\sqrt{0.3439+0.6561\cos(2t)^2}\right)^{3/2}}$$

$$+\frac{1}{2}\left(\cos(t)\left(-3.240000000\cos(2t)-\frac{1.721868840\cos(2t)^2\sin(2t)^2}{\left(0.3439+0.6561\cos(2t)^2\right)^{3/2}}\right.\right.$$

$$\left.\left.+\frac{2.624400000\sin(2t)^2}{\sqrt{0.3439+0.6561\cos(2t)^2}}-\frac{2.624400000\cos(2t)^2}{\sqrt{0.3439+0.6561\cos(2t)^2}}\right)\right)\Bigg/$$

$$\left.\sqrt{0.8100000000\cos(2t)+\sqrt{0.3439+0.6561\cos(2t)^2}}\right)+\Bigg($$

$$-\sin(t)\sqrt{0.8100000000\cos(2t)+\sqrt{0.3439+0.6561\cos(2t)^2}}$$

$$+\frac{1}{2}\frac{\cos(t)\left(-1.620000000\sin(2t)-\dfrac{1.312200000\cos(2t)\sin(2t)}{\sqrt{0.3439+0.6561\cos(2t)^2}}\right)}{\sqrt{0.8100000000\cos(2t)+\sqrt{0.3439+0.6561\cos(2t)^2}}}\Bigg)\Bigg($$

$$-\sin(t)\sqrt{0.8100000000\cos(2t)+\sqrt{0.3439+0.6561\cos(2t)^2}}$$

$$+\frac{\cos(t)\left(-1.620000000\sin(2t)-\frac{1.312200000\cos(2t)\sin(2t)}{\sqrt{0.3439+0.6561\cos(2t)^2}}\right)}{\sqrt{0.8100000000\cos(2t)+\sqrt{0.3439+0.6561\cos(2t)^2}}}$$

$$-\frac{1}{4}\frac{\sin(t)\left(-1.620000000\sin(2t)-\frac{1.312200000\cos(2t)\sin(2t)}{\sqrt{0.3439+0.6561\cos(2t)^2}}\right)^2}{\left(0.8100000000\cos(2t)+\sqrt{0.3439+0.6561\cos(2t)^2}\right)^{3/2}}$$

$$+\frac{1}{2}\left(\sin(t)\left(-3.240000000\cos(2t)-\frac{1.721868840\cos(2t)^2\sin(2t)^2}{\left(0.3439+0.6561\cos(2t)^2\right)^{3/2}}\right.\right.$$

$$\left.\left.+\frac{2.624400000\sin(2t)^2}{\sqrt{0.3439+0.6561\cos(2t)^2}}-\frac{2.624400000\cos(2t)^2}{\sqrt{0.3439+0.6561\cos(2t)^2}}\right)\right)\Bigg/$$

$$\left.\sqrt{0.8100000000\cos(2t)+\sqrt{0.3439+0.6561\cos(2t)^2}}\right)\Bigg|$$

$$\left(\left(\left(\cos(t)\sqrt{0.8100000000\cos(2t)+\sqrt{0.3439+0.6561\cos(2t)^2}}\right.\right.\right.$$

$$\left.+\frac{1}{2}\frac{\sin(t)\left(-1.620000000\sin(2t)-\frac{1.312200000\cos(2t)\sin(2t)}{\sqrt{0.3439+0.6561\cos(2t)^2}}\right)}{\sqrt{0.8100000000\cos(2t)+\sqrt{0.3439+0.6561\cos(2t)^2}}}\right)^2+\left(\right.$$

$$-\sin(t)\sqrt{0.8100000000\cos(2t)+\sqrt{0.3439+0.6561\cos(2t)^2}}$$

$$\left.\left.\left.+\frac{1}{2}\frac{\cos(t)\left(-1.620000000\sin(2t)-\frac{1.312200000\cos(2t)\sin(2t)}{\sqrt{0.3439+0.6561\cos(2t)^2}}\right)}{\sqrt{0.8100000000\cos(2t)+\sqrt{0.3439+0.6561\cos(2t)^2}}}\right)^2\right)^{1.5}\right)$$

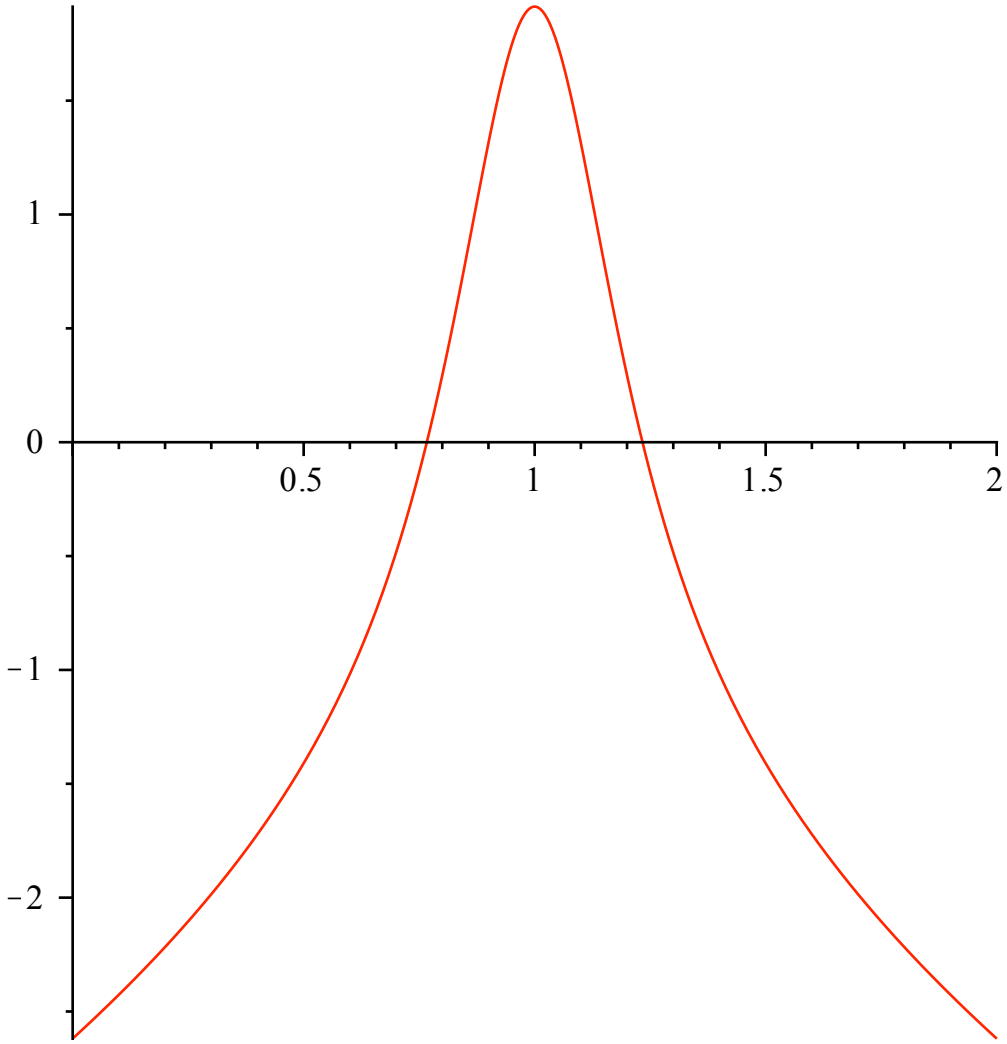
$k1norm := k1 \cdot 1.345$:

$gnorm := \frac{(g + 1.345)}{1.345}$:

$plot([gnorm, k1norm, t = 0..\mathrm{Pi}])$;

$\#plot\left(\left\{g, \frac{cprod(3)}{\mathrm{abs}(cprod(3))}, t = 0..\mathrm{Pi}\right\}\right)$;

$evalf(subs(t = .1, (\langle gnorm, k1norm\rangle)))$;



$$(1.987217288)e_x + (-2.594983228)e_y \qquad \textbf{(3.4)}$$

for i **from** 0 **to** 3.14 **by** .1 **do** $evalf(subs(t = i, k1norm))$; **end do**;

-2.619294240

-2.594983228

-2.521832347

-2.399134332

-2.225533124

-1.998788590

$$
\begin{aligned}
&-1.715631560\\
&-1.372299220\\
&-0.9670399872\\
&-0.5053737460\\
&-0.005185912437\\
&0.5032241806\\
&0.9820668915\\
&1.392300422\\
&1.699032803\\
&1.875655228\\
&1.906707031\\
&1.789461640\\
&1.534201325\\
&1.163173498\\
&0.7082539448\\
&0.2073428153\\
&-0.3005172946\\
&-0.7811591954\\
&-1.211181617\\
&-1.580331870\\
&-1.888087173\\
&-2.137843638\\
&-2.333217603\\
&-2.476877913\\
&-2.570518382\\
&-2.615091078
\end{aligned}
\qquad \textbf{(3.5)}
$$

$n := \langle (-1)\cdot diff(g,t), diff(f,t)\rangle :$

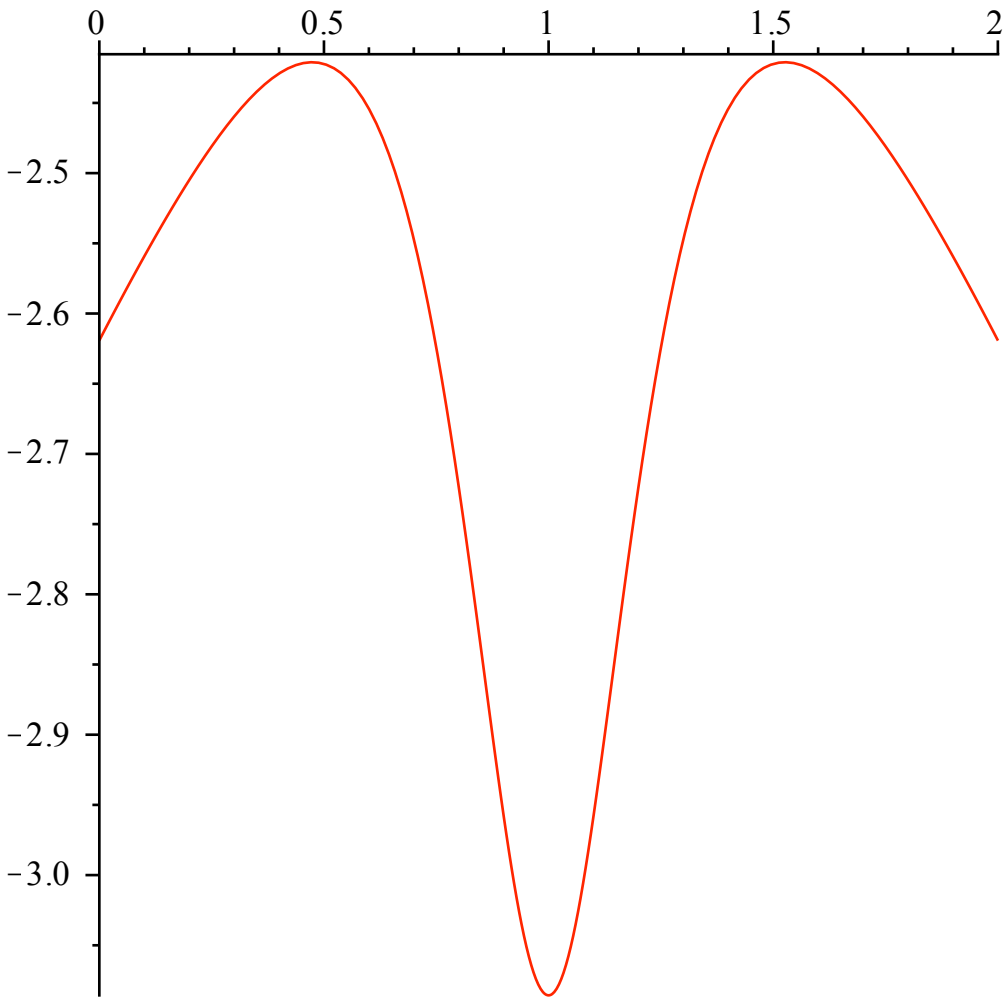
$nunit := \dfrac{n}{\left(DotProduct(n,n)^{.5}\right)} :$

$xvec := \langle (-1)\cdot f, 0\rangle :$

$K2 := \dfrac{DotProduct(xvec, nunit)}{DotProduct(xvec, xvec)} :$

$k2norm := K2\cdot 1.345 :$

$plot([gnorm, k2norm, t = 0..\mathrm{Pi}]);$



for *i* **from** 0 **to** 3.14 **by** .1 **do** *evalf* (*subs* (*t* = *i*, *k2norm*)); **end do**;

Float (*undefined*)

−2.611256655

−2.587888458

−2.551594136

−2.506991958

−2.461917843

−2.428623928

−2.423469174

−2.461052872

−2.542832914

−2.653732995

−2.772879034

-2.883520279

-2.974943240

-3.040884216

-3.077887712

-3.084320402

-3.059917844

-3.005713058

-2.924318972

-2.820669108

-2.703392726

-2.586637535

-2.490263924

-2.433409862

-2.422069107

-2.445767650

-2.487710516

-2.533689639

-2.574157091

-2.603303258

-2.617895154 **(3.6)**

for *i* **from** 0 **to** 3.14 **by** .1 **do** *evalf* (*subs* (*t* = *i*, *gnorm*)); **end do**;

2.000269446

1.987217288

1.948678923

1.886523777

1.803935724

1.705574889

1.597871438

1.489191834

1.388758811

1.303270497

1.234003541

1.178149374

1.131918393

1.092126668

1.056412871

1.023018043

0.9905304278

0.9576723152

0.9231114435

0.8852619409

0.8420457381

0.7906352495

0.7274040352

0.6488158712

0.5540712308

0.4477487176

0.3385246433

0.2353839323

0.1455758840

0.0744559465

0.0257945212

0.0019958610 **(3.7)**

Effective von Mises Stress for an Idealized Rotated Surface

```
restart;
with(VectorCalculus) :
rt := <fp·cos(theta), gp, fp·sin(theta)>;
rtheta := <f·(-sin(theta)), 0, f·cos(theta)>;
nunit := simplify( CrossProduct(rt, rtheta) / sqrt(DotProduct(CrossProduct(rt, rtheta), CrossProduct(rt, rtheta))) );
```

$$(fp\cos(\theta))e_x + (gp)e_y + (fp\sin(\theta))e_z$$

$$-f\sin(\theta)e_x + (f\cos(\theta))e_z$$

$$\left(\frac{gp\,f\cos(\theta)}{\sqrt{f^2\,(gp^2+fp^2)}}\right)e_x - \frac{f\,fp}{\sqrt{f^2\,(gp^2+fp^2)}}e_y + \left(\frac{gp\,f\sin(\theta)}{\sqrt{f^2\,(gp^2+fp^2)}}\right)e_z \quad \textbf{(4.1)}$$

```
rtt := <fpp·cos(theta), gpp, fpp·sin(theta)>;
rthetatheta := <f·(-cos(theta)), 0, f·(-sin(theta))>;
rttheta := <fp·(-sin(θ)), 0, fp·cos(theta)>;
```

$$(fpp\cos(\theta))e_x + (gpp)e_y + (fpp\sin(\theta))e_z$$

$$-f\cos(\theta)e_x - f\sin(\theta)e_z$$

$$-fp\sin(\theta)e_x + (fp\cos(\theta))e_z \tag{4.2}$$

$E := simplify(DotProduct(rt, rt));$
$F := DotProduct(rt, rtheta);$
$G := simplify(DotProduct(rtheta, rtheta));$
$L := simplify(DotProduct(rtt, nunit));$
$LL := DotProduct(rtt, nunit);$
$M := simplify(DotProduct(rttheta, nunit));$
$N := simplify(DotProduct(rthetatheta, nunit));$

$$gp^2 + fp^2$$

$$0$$

$$f^2$$

$$-\frac{f(-fpp\,gp + gpp\,fp)}{\sqrt{f^2(gp^2 + fp^2)}}$$

$$\frac{fpp\cos(\theta)^2 gp\,f}{\sqrt{f^2(gp^2 + fp^2)}} - \frac{gpp\,f\,fp}{\sqrt{f^2(gp^2 + fp^2)}} + \frac{fpp\sin(\theta)^2 gp\,f}{\sqrt{f^2(gp^2 + fp^2)}}$$

$$0$$

$$-\frac{f^2 gp}{\sqrt{f^2(gp^2 + fp^2)}} \tag{4.3}$$

$$H := \left(\frac{1}{2}\right)\cdot\left(\frac{(E\cdot N - 2\cdot F\cdot M + G\cdot L)}{(E\cdot G - F^2)}\right);$$

$$\frac{1}{2}\,\frac{-\frac{(gp^2 + fp^2)f^2 gp}{\sqrt{f^2(gp^2 + fp^2)}} - \frac{f^3(-fpp\,gp + gpp\,fp)}{\sqrt{f^2(gp^2 + fp^2)}}}{f^2(gp^2 + fp^2)} \tag{4.4}$$

$simplify(H)$

$$-\frac{1}{2}\,\frac{gp^3 + gp\,fp^2 - f\,fpp\,gp + gpp\,f\,fp}{\sqrt{f^2(gp^2 + fp^2)}\,(gp^2 + fp^2)} \tag{4.5}$$

$dA := simplify(\mathrm{sqrt}(DotProduct(CrossProduct(rt, rtheta), CrossProduct(rt, rtheta))));$

$$\sqrt{f^2(fp^2 + gp^2)} \tag{4.6}$$

$\#\sigma_1 := 2\cdot\mathrm{gamma}\cdot H\cdot nunit(2);$

$$\sigma_1 := simplify\left(\frac{4\cdot b\cdot int\left(H2\cdot gp\cdot f, t = \left(a - \frac{R}{1000}\right)..\left(a + \frac{R}{1000}\right)\right)}{2\cdot int\left(f, t = \left(a - \frac{R}{1000}\right)..\left(a + \frac{R}{1000}\right)\right)}\right);$$

$$\frac{\gamma\left(-\frac{(fp^2+gp^2)f^2\,gp}{\sqrt{f^2(fp^2+gp^2)}}-\frac{f^3(gpp\,fp-fpp\,gp)}{\sqrt{f^2(fp^2+gp^2)}}\right)\left(-fp\sin(\theta)^2 f-fp\cos(\theta)^2 f\right)}{f^2(fp^2+gp^2)\sqrt{gp^2f^2\cos(\theta)^2+\left(-fp\sin(\theta)^2 f-fp\cos(\theta)^2 f\right)^2+gp^2f^2\sin(\theta)^2}} \tag{4.7}$$

$$\#\sigma_2 := \frac{4\cdot \text{gamma}\cdot int(H, t=a..b)}{f^2};$$

$$\sigma_2 := \frac{-4\cdot b\cdot int(H2\cdot(-fp)\cdot f, t=0..a)}{f^2};$$

$$\frac{2\gamma\left(-\frac{(fp^2+gp^2)f^2\,gp}{\sqrt{f^2(fp^2+gp^2)}}-\frac{f^3(gpp\,fp-fpp\,gp)}{\sqrt{f^2(fp^2+gp^2)}}\right)(b-a)}{f^4(fp^2+gp^2)} \tag{4.8}$$

$$\sigma_{vm} := \text{abs}\left(\sigma_1-\sigma_2\right);$$

$$\left|\frac{\gamma\left(-\frac{(fp^2+gp^2)f^2\,gp}{\sqrt{f^2(fp^2+gp^2)}}-\frac{f^3(gpp\,fp-fpp\,gp)}{\sqrt{f^2(fp^2+gp^2)}}\right)\left(-fp\sin(\theta)^2 f-fp\cos(\theta)^2 f\right)}{f^2(fp^2+gp^2)\sqrt{gp^2f^2\cos(\theta)^2+\left(-fp\sin(\theta)^2 f-fp\cos(\theta)^2 f\right)^2+gp^2f^2\sin(\theta)^2}}-\frac{2\gamma\left(-\frac{(fp^2+gp^2)f^2\,gp}{\sqrt{f^2(fp^2+gp^2)}}-\frac{f^3(gpp\,fp-fpp\,gp)}{\sqrt{f^2(fp^2+gp^2)}}\right)(b-a)}{f^4(fp^2+gp^2)}\right| \tag{4.9}$$

Effective von Mises Stress for a Sphere

```
restart;
with(VectorCalculus):
 with(plots):
 R := 1;
f := sqrt(R^2 - (R - t)^2);
fp := diff(f, t);
fpp := diff(fp, t);
g := R-t;
gp := diff(g, t);
gpp := diff(gp, t);
p0 := plot3d( [f·cos(theta), g, f·sin(theta)], t = 0 .. 2·R, theta = 0 .. 2· Pi, axes = boxed, scaling
      = constrained, labels = ["x", "y", "z"], color = blue);
p1 := plot3d( [f·cos(theta), g, f·sin(theta)], t = R/5 .. R/4, theta = 0 .. 2· Pi, axes = boxed, scaling
      = constrained, labels = ["x", "y", "z"], style = surface, color = red);
display(p0, p1);
```

$$1$$

$$\sqrt{2\,t-t^2}$$

$$\frac{1}{2}\ \frac{2-2\,t}{\sqrt{2\,t-t^2}}$$

$$-\frac{1}{4}\ \frac{(2-2\,t)^2}{\left(2\,t-t^2\right)^{3/2}}-\frac{1}{\sqrt{2\,t-t^2}}$$

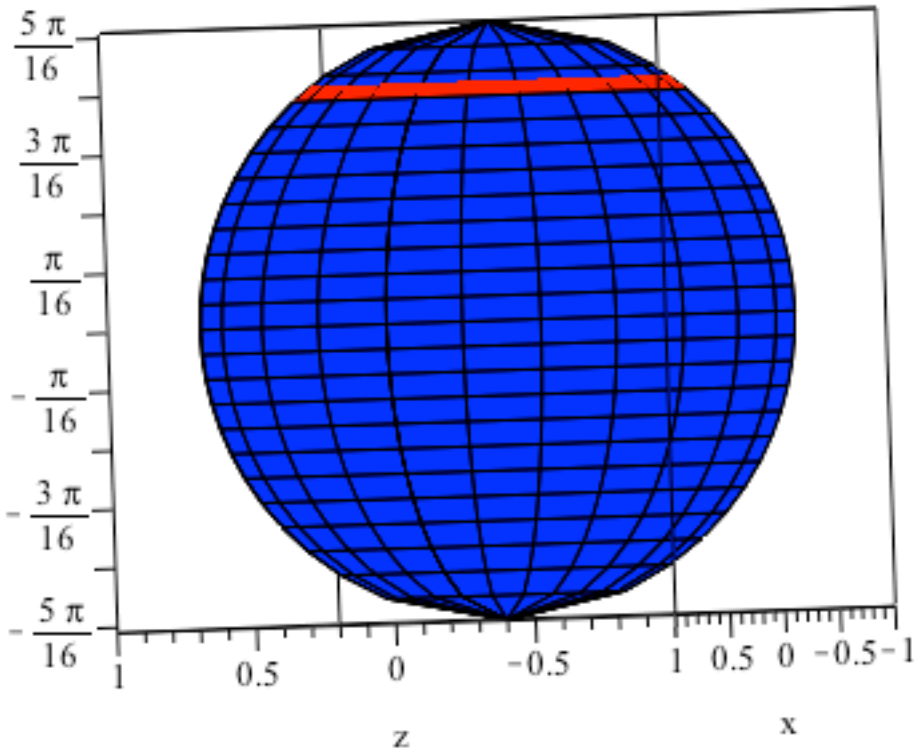
$$1-t$$

$$-1$$

$$0$$

$$PLOT3D(\ldots)$$

$$PLOT3D(\ldots)$$




```
rt := ⟨fp·cos(theta), gp, fp·sin(theta)⟩;
rtheta := ⟨f·(-sin(theta)), 0, f·cos(theta)⟩;
```

$$nunit := \frac{CrossProduct(rt, rtheta)}{\mathrm{sqrt}(DotProduct(CrossProduct(rt, rtheta), CrossProduct(rt, rtheta)))};$$

$$simplify(nunit);$$

$$\frac{1}{2}\,\frac{(2R-2t)\cos(\theta)}{\sqrt{2Rt-t^2}}e_x - e_y + \frac{1}{2}\,\frac{(2R-2t)\sin(\theta)}{\sqrt{2Rt-t^2}}e_z$$

$$-\sqrt{2Rt-t^2}\,\sin(\theta)e_x + \left(\sqrt{2Rt-t^2}\,\cos(\theta)\right)e_z$$

$$-\left(\sqrt{2Rt-t^2}\,\cos(\theta)\right) \Big/ \left(\left(2Rt-t^2\right)\cos(\theta)^2 + \left(-\frac{1}{2}\,(2R-2t)\sin(\theta)^2 - \frac{1}{2}\,(2R-2t)\cos(\theta)^2\right)^2 + \left(2Rt-t^2\right)\sin(\theta)^2\right)^{1/2} e_x + \left(\left(-\frac{1}{2}\,(2R-2t)\sin(\theta)^2 - \frac{1}{2}\,(2R-2t)\cos(\theta)^2\right) \Big/ \left(\left(2Rt-t^2\right)\cos(\theta)^2 + \left(-\frac{1}{2}\,(2R-2t)\sin(\theta)^2 - \frac{1}{2}\,(2R-2t)\cos(\theta)^2\right)^2 + \left(2Rt-t^2\right)\sin(\theta)^2\right)^{1/2}\right)e_y - \left(\sqrt{2Rt-t^2}\,\sin(\theta)\right) \Big/ \left(\left(2Rt-t^2\right)\cos(\theta)^2 + \left(-\frac{1}{2}\,(2R-2t)\sin(\theta)^2 - \frac{1}{2}\,(2R-2t)\cos(\theta)^2\right)^2 + \left(2Rt-t^2\right)\sin(\theta)^2\right)^{1/2} e_z$$

$$-\frac{\mathrm{csgn}(R)\sqrt{t(-t+2R)}\,\cos(\theta)}{R}e_x - \frac{\mathrm{csgn}(R)\,(R-t)}{R}e_y - \frac{\mathrm{csgn}(R)\sqrt{t(-t+2R)}\,\sin(\theta)}{R}e_z \qquad \textbf{(5.1)}$$

$$rtt := \langle fpp\cdot\cos(\mathrm{theta}), gpp, fpp\cdot\sin(\mathrm{theta})\rangle;$$

$$rthetatheta := \langle f\cdot(-\cos(\mathrm{theta})), 0, f\cdot(-\sin(\mathrm{theta}))\rangle;$$

$$rttheta := \langle fp\cdot(-\sin(\theta)), 0, fp\cdot\cos(\mathrm{theta})\rangle;$$

$$\left(\left(-\frac{1}{4}\,\frac{(2R-2t)^2}{(2Rt-t^2)^{3/2}} - \frac{1}{\sqrt{2Rt-t^2}}\right)\cos(\theta)\right)e_x + \left(\left(-\frac{1}{4}\,\frac{(2R-2t)^2}{(2Rt-t^2)^{3/2}} - \frac{1}{\sqrt{2Rt-t^2}}\right)\sin(\theta)\right)e_z$$

$$-\sqrt{2\,R\,t-t^2}\,\cos(\theta)\,e_x-\sqrt{2\,R\,t-t^2}\,\sin(\theta)\,e_z$$

$$-\frac{1}{2}\,\frac{(2\,R-2\,t)\,\sin(\theta)}{\sqrt{2\,R\,t-t^2}}\,e_x+\frac{1}{2}\,\frac{(2\,R-2\,t)\,\cos(\theta)}{\sqrt{2\,R\,t-t^2}}\,e_z \tag{5.2}$$

$E := DotProduct(rt, rt);$
$F := DotProduct(rt, rtheta);$
$G := DotProduct(rtheta, rtheta);$
$L := DotProduct(rtt, nunit):$
$M := DotProduct(rttheta, nunit);$
$N := DotProduct(rthetatheta, nunit):$

$$1+\frac{1}{4}\,\frac{(2\,R-2\,t)^2\,\cos(\theta)^2}{2\,R\,t-t^2}+\frac{1}{4}\,\frac{(2\,R-2\,t)^2\,\sin(\theta)^2}{2\,R\,t-t^2}$$

$$0$$

$$\left(2\,R\,t-t^2\right)\sin(\theta)^2+\left(2\,R\,t-t^2\right)\cos(\theta)^2$$

$$0 \tag{5.3}$$

$H := \left(\frac{1}{2}\right)\cdot\left(\frac{(E\cdot N-2\cdot F\cdot M+G\cdot L)}{\left(E\cdot G-F^2\right)}\right):$
$simplify(H)$

$$\frac{csgn(R)}{R} \tag{5.4}$$

$H := \frac{1}{R};$
$\sigma_1 := 2\cdot gamma\cdot H\cdot nunit(1);$
$simplify\left(\sigma_1\right);$
$\#\sigma_1 := \frac{2\cdot gamma\cdot t}{R^2};$
$\#\sigma_1 := \frac{2\cdot gamma\cdot sqrt(t\cdot(2\cdot R-t))}{R^2\cdot f};$
$\#\sigma_1 := -1\cdot gamma\cdot H\cdot int(nunit(3), theta=0..Pi);$
$\sigma_1 := \frac{-4\cdot gamma\cdot int\left(H\cdot gp\cdot f, t=\left(a-\frac{R}{100}\right)..\left(a+\frac{R}{100}\right)\right)}{2\cdot int\left(f, t=\left(a-\frac{R}{100}\right)..\left(a+\frac{R}{100}\right)\right)};$

$$\frac{1}{R}$$

$$-\left(4\,\gamma^2\sqrt{2\,R\,t-t^2}\,\cos(\theta)\right)\Big/$$

$$\left(R^2\left(\left(2Rt-t^2\right)\cos(\theta)^2+\left(-\frac{1}{2}(2R-2t)\sin(\theta)^2-\frac{1}{2}(2R-2t)\cos(\theta)^2\right)^2+\left(2Rt-t^2\right)\sin(\theta)^2\right)^{1/2}\right)$$

$$-\frac{2\gamma\left(\int_{a-\frac{1}{100}R}^{a+\frac{1}{100}R}\left(-\frac{\sqrt{2Rt-t^2}}{R}\right)dt\right)}{\int_{a-\frac{1}{100}R}^{a+\frac{1}{100}R}\sqrt{2Rt-t^2}\,dt} \tag{5.5}$$

$$\sigma_2 := \frac{-4\cdot \text{gamma}\cdot int(H\cdot(-fp)\cdot f, t=0..a)}{f^2}$$

$$-\frac{4\gamma\left(-a+\frac{1}{2}\frac{a^2}{R}\right)}{2Rt-t^2} \tag{5.6}$$

$$\sigma_{vm} := \text{abs}\left(\sigma_1-\sigma_2\right);$$

$$\left|-\frac{2\gamma\left(\int_{a-\frac{1}{100}R}^{a+\frac{1}{100}R}\left(-\frac{\sqrt{2Rt-t^2}}{R}\right)dt\right)}{\int_{a-\frac{1}{100}R}^{a+\frac{1}{100}R}\sqrt{2Rt-t^2}\,dt}+\frac{4\gamma\left(-a+\frac{1}{2}\frac{a^2}{R}\right)}{2Rt-t^2}\right| \tag{5.7}$$

$$\sigma_{2\,t} := simplify\left(subs\left(\left\{a=\frac{R}{4}, t=\frac{R}{4}\right\}, \sigma_2\right)\right);$$

$$\sigma_{1\,t} := simplify\left(subs\left(\left\{a=\frac{R}{4}\right\}, \sigma_1\right)\right);$$

$$\sigma_{vm} := \text{abs}\left(\sigma_{1\,t}-\sigma_{2\,t}\right);$$

$$\frac{2\gamma}{R}$$

$$\frac{2\gamma}{R}$$

$$0 \tag{5.8}$$

Effective von Mises Stress for a Sphereocylinder

```
restart;
with(VectorCalculus):
f := R;
fp := diff(f, t);
fpp := diff(fp, t);
g := t;
gp := diff(g, t);
gpp := diff(gp, t);
```

$$\begin{bmatrix} R \\ 0 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} t \\ 1 \\ 0 \end{bmatrix} \tag{6.1}$$

```
rt := <fp*cos(theta), gp, fp*sin(theta)>;
rtheta := <f*(-sin(theta)), 0, f*cos(theta)>;
nunit := CrossProduct(rt, rtheta) / sqrt(DotProduct(CrossProduct(rt, rtheta), CrossProduct(rt, rtheta)));
```

$$e_y$$

$$-R\sin(\theta)e_x + (R\cos(\theta))e_z$$

$$\left(\frac{R\cos(\theta)}{\sqrt{R^2\cos(\theta)^2 + R^2\sin(\theta)^2}}\right)e_x + \left(\frac{R\sin(\theta)}{\sqrt{R^2\cos(\theta)^2 + R^2\sin(\theta)^2}}\right)e_z \tag{6.2}$$

```
rtt := <fpp*cos(theta), gpp, fpp*sin(theta)>;
rthetatheta := <f*(-cos(theta)), 0, f*(-sin(theta))>;
rttheta := <fp*(-sin(θ)), 0, fp*cos(theta)>;
```

$$0e_x$$

$$-R\cos(\theta)e_x - R\sin(\theta)e_z$$

$$0e_x \tag{6.3}$$

```
E := DotProduct(rt, rt);
F := DotProduct(rt, rtheta);
G := DotProduct(rtheta, rtheta);
L := DotProduct(rtt, nunit);
M := DotProduct(rttheta, nunit);
N := DotProduct(rthetatheta, nunit);
```

$$\begin{bmatrix} 1 \\ 0 \\ R^2\cos(\theta)^2 + R^2\sin(\theta)^2 \\ 0 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} -\dfrac{R^2\cos(\theta)^2}{\sqrt{R^2\cos(\theta)^2 + R^2\sin(\theta)^2}} \\ -\dfrac{R^2\sin(\theta)^2}{\sqrt{R^2\cos(\theta)^2 + R^2\sin(\theta)^2}} \end{bmatrix} \quad (6.4)$$

$H := \left(\frac{1}{2}\right) \cdot \left(\frac{(E \cdot N - 2 \cdot F \cdot M + G \cdot L)}{(E \cdot G - F^2)}\right);$

$$\begin{bmatrix} \dfrac{1}{2} \\ \dfrac{-\dfrac{R^2\cos(\theta)^2}{\sqrt{R^2\cos(\theta)^2 + R^2\sin(\theta)^2}} - \dfrac{R^2\sin(\theta)^2}{\sqrt{R^2\cos(\theta)^2 + R^2\sin(\theta)^2}}}{R^2\cos(\theta)^2 + R^2\sin(\theta)^2} \end{bmatrix} \quad (6.5)$$

$simplify(H);$

$H := \frac{1}{2 \cdot R};$

$a := \frac{R}{4};$

$$\begin{bmatrix} \dfrac{1}{2R} \\ \dfrac{1}{2R} \\ \dfrac{1}{4}R \end{bmatrix} \quad (6.6)$$

$$\#\sigma_1 := \frac{4 \cdot \text{gamma} \cdot int\left(H \cdot gp \cdot f, t = \left(a - \frac{R}{100}\right) .. \left(a + \frac{R}{100}\right)\right)}{2 \cdot int\left(f, t = \left(a - \frac{R}{100}\right) .. \left(a + \frac{R}{100}\right)\right)};$$

$$\frac{\gamma}{R} \quad (6.7)$$

Effective von Mises Stress for an Ellipsoid

```
restart;
 with(LinearAlgebra) :
 with(VectorCalculus) :
 with(plots);
```

[*animate*, *animate3d*, *animatecurve*, *arrow*, *changecoords*, *complexplot*, *complexplot3d*, (7.1)

conformal, *conformal3d*, *contourplot*, *contourplot3d*, *coordplot*, *coordplot3d*, *densityplot*, *display*, *dualaxisplot*, *fieldplot*, *fieldplot3d*, *gradplot*, *gradplot3d*, *implicitplot*, *implicitplot3d*, *inequal*, *interactive*, *interactiveparams*, *intersectplot*, *listcontplot*, *listcontplot3d*, *listdensityplot*, *listplot*, *listplot3d*, *loglogplot*, *logplot*, *matrixplot*, *multiple*, *odeplot*, *pareto*, *plotcompare*, *pointplot*, *pointplot3d*, *polarplot*, *polygonplot*, *polygonplot3d*, *polyhedra_supported*, *polyhedraplot*, *rootlocus*, *semilogplot*, *setcolors*, *setoptions*, *setoptions3d*, *spacecurve*, *sparsematrixplot*, *surfdata*, *textplot*, *textplot3d*, *tubeplot*]

#*c* := .95;

#*d* := 1;

#*M* := $2\cdot c^2\cdot\cos(2\cdot t)+2\cdot\mathrm{sqrt}\left(\left(-c^4+d^4\right)+c^4\cdot(\cos(2\cdot t))^2\right)$;

$g := t$;

$R := 1$;

$l := 5$;

$f := \mathrm{sqrt}\left(R^2-\left(R-\frac{g}{4}\right)^2\right)$;

$$t$$

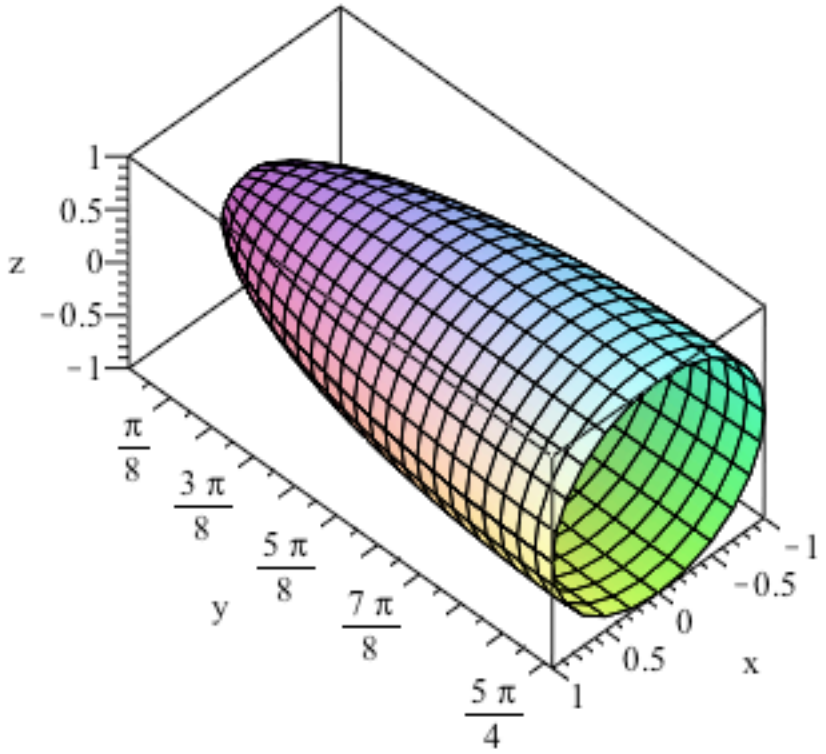
$$1$$

$$5$$

$$\frac{1}{4}\sqrt{8\,t-t^2} \qquad \textbf{(7.2)}$$

#*plot3d*([$f\cdot\cos(\mathrm{theta})$, g, $f\cdot\sin(\mathrm{theta})$], $t = 0..2\cdot\mathrm{Pi}$, theta = 0 ..$2\cdot\mathrm{Pi}$, *axes* = *normal*, *scaling* = *constrained*, *labels* = ["x", "y", "z"]);

plot3d([$f\cdot\cos(\mathrm{theta})$, g, $f\cdot\sin(\mathrm{theta})$], $t = 0 .. 4\cdot R$, theta = 0 .. $2\cdot\mathrm{Pi}$, *axes* = *boxed*, *scaling* = *constrained*, *labels* = ["x", "y", "z"]);



$fp := diff(f, t)$:
$fpp := diff(fp, t)$:
$gp := diff(g, t)$:
$gpp := diff(gp, t)$:
$rt := \langle fp \cdot \cos(\text{theta}), gp, fp \cdot \sin(\text{theta}) \rangle$:
$rtheta := \langle f \cdot (-\sin(\text{theta})), 0, f \cdot \cos(\text{theta}) \rangle$:
$nunit := \dfrac{CrossProduct(rt, rtheta)}{\text{sqrt}(DotProduct(CrossProduct(rt, rtheta), CrossProduct(rt, rtheta)))}$:
$rtt := \langle fpp \cdot \cos(\text{theta}), gpp, fpp \cdot \sin(\text{theta}) \rangle$:
$rthetatheta := \langle f \cdot (-\cos(\text{theta})), 0, f \cdot (-\sin(\text{theta})) \rangle$:
$rttheta := \langle fp \cdot (-\sin(\theta)), 0, fp \cdot \cos(\text{theta}) \rangle$:

$E := DotProduct(rt, rt)$:
$F := DotProduct(rt, rtheta)$:
$G := DotProduct(rtheta, rtheta)$:
$L := DotProduct(rtt, nunit)$:
$M := DotProduct(rttheta, nunit)$:
$N := DotProduct(rthetatheta, nunit)$:

$$H := \left(\frac{1}{2}\right)\cdot\left(\frac{(E\cdot N - 2\cdot F\cdot M + G\cdot L)}{(E\cdot G - F^2)}\right):$$

$$H3 := -1\cdot simplify(H):$$

$$H2 := evalf\left(subs\left(\left\{\text{theta} = \frac{\pi}{2}\right\}, H3\right)\right);$$

$$-\frac{8.\,(15.\,t^2 - 120.\,t - 32.)}{(16. + 120.\,t - 15.\,t^2)^{3/2}} \tag{7.3}$$

$$a := .5\cdot R:$$

$$b := 1:$$

$$\sigma_1 := \left(\frac{4\cdot b\cdot evalf\left(int\left(H2\cdot gp\cdot f, t = \left(a - \frac{R}{1000}\right)..\left(a + \frac{R}{1000}\right)\right)\right)}{2\cdot evalf\left(int\left(f, t = \left(a - \frac{R}{1000}\right)..\left(a + \frac{R}{1000}\right)\right)\right)}\right):$$

$$\#\sigma_1 := \left(\frac{4\cdot b\cdot evalf\left(int\left(H2\cdot gp\cdot f, t = \left(a - \frac{R}{1000}\right)..\left(a + \frac{R}{1000}\right)\right)\right)}{2\,\text{Pi}\cdot evalf\left(int\left(f\cdot\text{sqrt}\left(fp^2 + gp^2\right), t = \left(a - \frac{R}{1000}\right)..\left(a + \frac{R}{1000}\right)\right)\right)}\right);$$

$$\#\sigma_1 := \left(\frac{4\cdot b\cdot evalf\left(int\left(H2\cdot gp\cdot f, t = \left(a - \frac{R}{1000}\right)..\left(a + \frac{R}{1000}\right)\right)\right)}{2\text{Pi}\cdot subs(\{t = a\}, f)^2}\right);$$

$$evalf(\sigma_1);$$

$$\#surfaceA := 2\,Pi\cdot evalf\left(int\left(f\cdot\text{sqrt}\left(fp^2 + gp^2\right), t = (0)..\left(\frac{R}{2}\right)\right)\right);$$

$$test := evalf(int(H2\cdot(-fp)\cdot f, t = 0..a)):$$

$$2.299206635 \tag{7.4}$$

$$\sigma_2 := \frac{-4\cdot b\cdot evalf(int(H2\cdot(-fp)\cdot f, t = 0..a))}{(subs(\{t = a\}, f))^2};$$

$$3.764705884 \tag{7.5}$$

$$\#\sigma_{2\,t} := simplify(subs(\{t = a\}, \sigma_2));$$

$$\sigma_{vm} := \text{abs}(\sigma_1 - \sigma_2);$$

$$evalf(\sigma_{vm});$$

$$1.465499249$$

$$1.465499249 \tag{7.6}$$

Effective von Mises Stress for Junction of Ball and Stick

```
restart;
with(LinearAlgebra):
with(VectorCalculus):
with(plots);
```

(8.1)

```
[animate, animate3d, animatecurve, arrow, changecoords, complexplot, complexplot3d,
    conformal, conformal3d, contourplot, contourplot3d, coordplot, coordplot3d, densityplot,
    display, dualaxisplot, fieldplot, fieldplot3d, gradplot, gradplot3d, implicitplot,
    implicitplot3d, inequal, interactive, interactiveparams, intersectplot, listcontplot,
    listcontplot3d, listdensityplot, listplot, listplot3d, loglogplot, logplot, matrixplot, multiple,
    odeplot, pareto, plotcompare, pointplot, pointplot3d, polarplot, polygonplot,
    polygonplot3d, polyhedra_supported, polyhedraplot, rootlocus, semilogplot, setcolors,
    setoptions, setoptions3d, spacecurve, sparsematrixplot, surfdata, textplot, textplot3d,
    tubeplot]
```

```
c := .95;
d := 1;
M := 2·c^2·cos(2·t) + 2·sqrt((-c^4 + d^4) + c^4·(cos(2·t))^2);
g := cos(t)·sqrt(M/2);
R := 1;
l := 5;
f := sin(t)·sqrt(M/2);
evalf(subs({t = Pi/4}, f));
```

$$0.95$$

$$1$$

$$1.8050\cos(2t) + 2\sqrt{0.18549375 + 0.81450625\cos(2t)^2}$$

$$\cos(t)\sqrt{0.9025000000\cos(2t) + \sqrt{0.18549375 + 0.81450625\cos(2t)^2}}$$

$$1$$

$$5$$

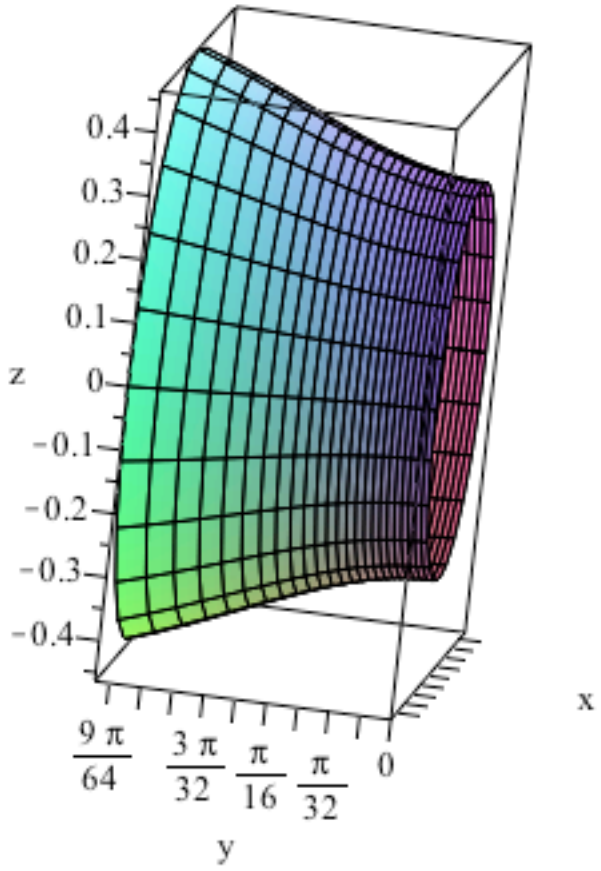
$$\sin(t)\sqrt{0.9025000000\cos(2t) + \sqrt{0.18549375 + 0.81450625\cos(2t)^2}}$$

$$0.4640527198$$

(8.2)

```
#plot3d([f·cos(theta), g, f·sin(theta)], t = 0..2·Pi, theta = 0..2·Pi, axes = normal, scaling
    = constrained, labels = ["x", "y", "z"]);
```

$$plot3d\left([\, f\cdot\cos(\text{theta}), g, f\cdot\sin(\text{theta})\,], t = \frac{\text{Pi}}{4} .. \frac{\text{Pi}}{2}, \text{theta} = 0 .. 2\cdot \text{Pi}, axes = boxed, scaling = constrained, labels = [\text{"x"}, \text{"y"}, \text{"z"}] \right);$$

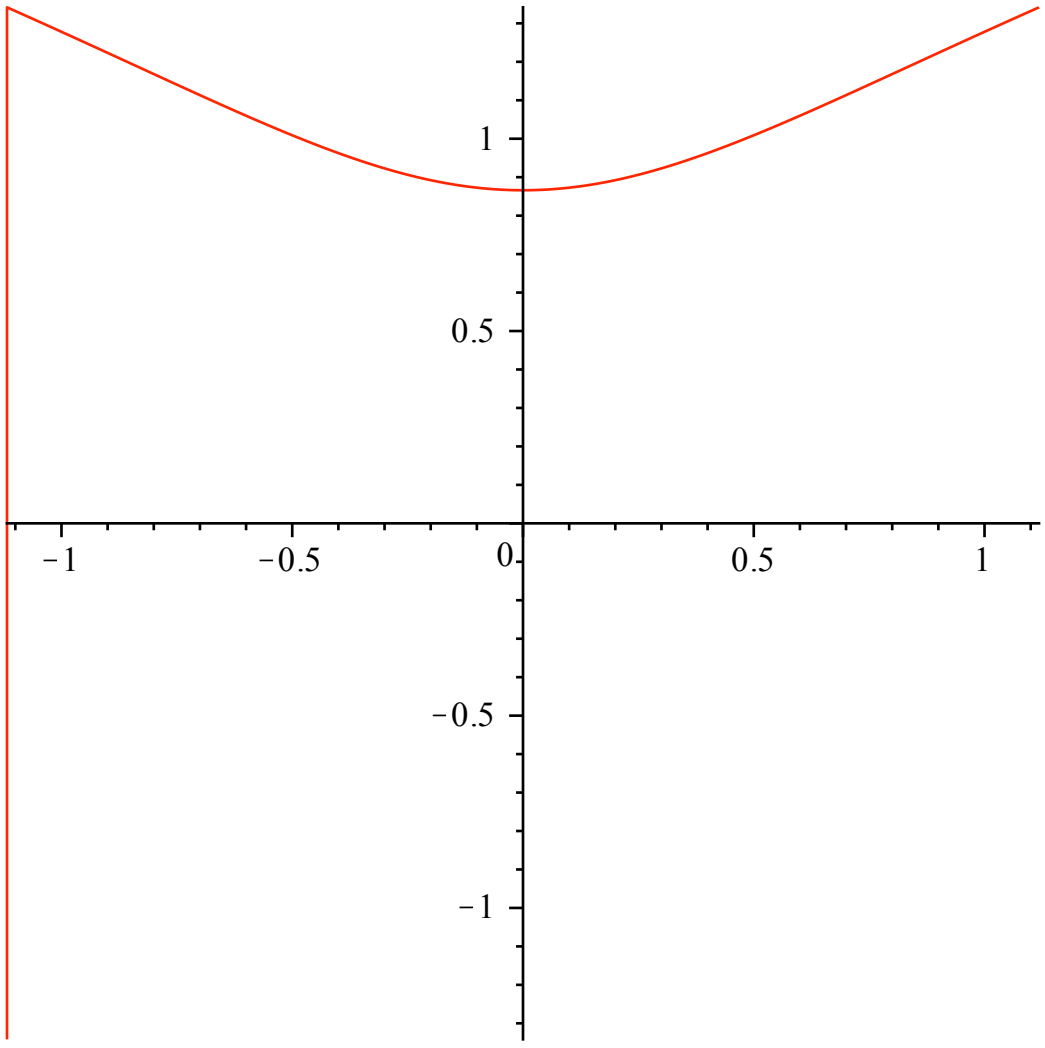


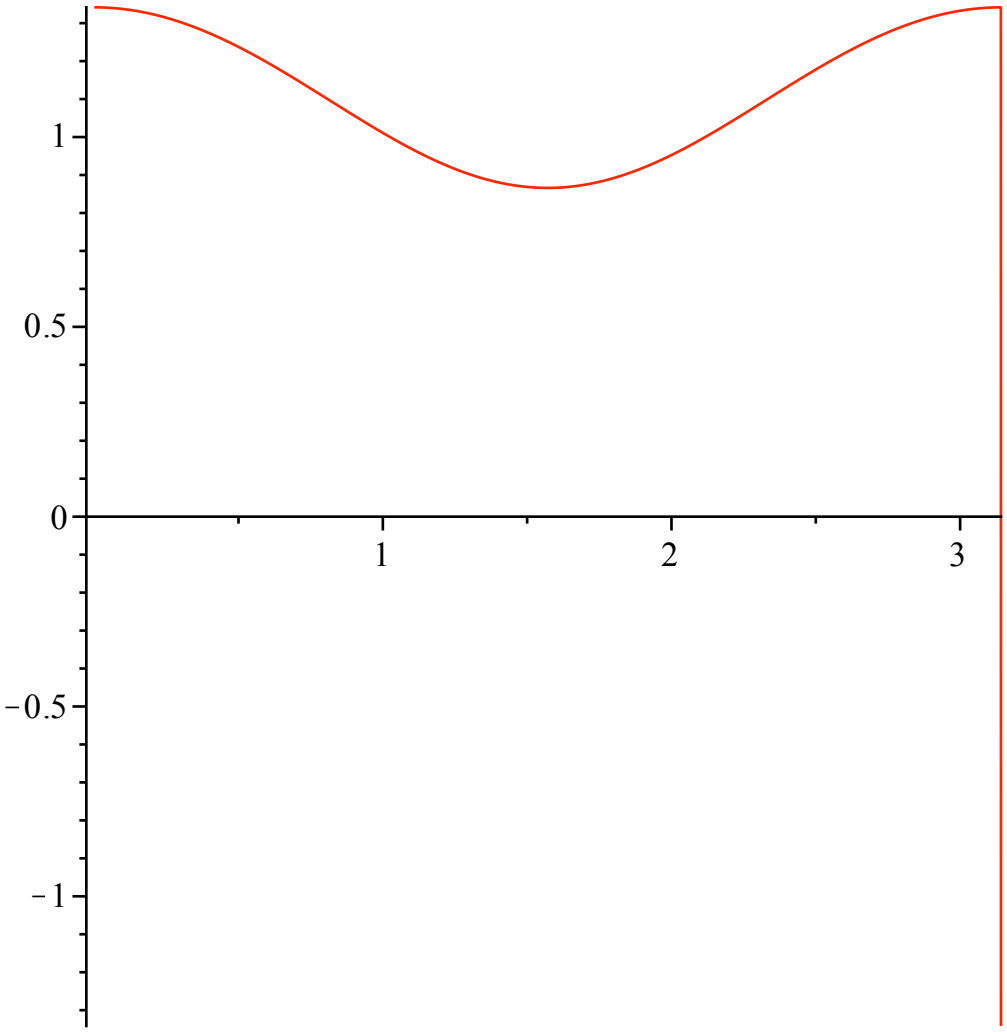
$$fp := diff(f, t):$$
$$fpp := diff(fp, t):$$
$$gp := diff(g, t):$$
$$gpp := diff(gp, t):$$
$$rt := \langle fp\cdot\cos(\text{theta}), gp, fp\cdot\sin(\text{theta})\rangle:$$
$$rtheta := \langle f\cdot(-\sin(\text{theta})), 0, f\cdot\cos(\text{theta})\rangle:$$
$$nunit := \frac{CrossProduct(rt, rtheta)}{\text{sqrt}(DotProduct(CrossProduct(rt, rtheta), CrossProduct(rt, rtheta)))}:$$
$$rtt := \langle fpp\cdot\cos(\text{theta}), gpp, fpp\cdot\sin(\text{theta})\rangle:$$
$$rthetatheta := \langle f\cdot(-\cos(\text{theta})), 0, f\cdot(-\sin(\text{theta}))\rangle:$$
$$rttheta := \langle fp\cdot(-\sin(\theta)), 0, fp\cdot\cos(\text{theta})\rangle:$$

$$E := DotProduct(rt, rt):$$
$$F := DotProduct(rt, rtheta):$$

$$G := DotProduct(rtheta, rtheta) :$$
$$L := DotProduct(rtt, nunit) :$$
$$M := DotProduct(rttheta, nunit) :$$
$$N := DotProduct(rthetatheta, nunit) :$$
$$H := \left(\frac{1}{2}\right) \cdot \left(\frac{(E \cdot N - 2 \cdot F \cdot M + G \cdot L)}{(E \cdot G - F^2)}\right) :$$
$$H3 := -1 \cdot simplify(H) :$$
$$H2 := evalf\left(subs\left(\left\{\text{theta} = \frac{\pi}{2}\right\}, H3\right)\right) :$$
$$Htest := evalf(subs(\{t = 0.0001\}, H2));$$
$$plot([g, -1 \cdot H2, t = 0..Pi]);$$
$$plot([t, -1 \cdot H2, t = 0..Pi]);$$

$$-1.341640782$$





$$a := \frac{\mathrm{Pi}}{5.75} :$$

$$b := 1 :$$

$$\sigma_1 := \left(\frac{4 \cdot b \cdot evalf\left(int\left(H2 \cdot gp \cdot f, t = \left(a - \frac{R}{1000} \right) .. \left(a + \frac{R}{1000} \right) \right) \right)}{2 \cdot evalf\left(int\left(f, t = \left(a - \frac{R}{1000} \right) .. \left(a + \frac{R}{1000} \right) \right) \right)} \right);$$

$$\#evalf\left(\sigma_1 \right) :$$

$$test := evalf\left(int\left(H2 \cdot \left(-fp \right) \cdot f, t = 0 .. a \right) \right);$$

$$evalf\left(subs\left(\left\{ t = \frac{\mathrm{Pi}}{2.05} \right\}, f \right) \right);$$

1.817476609

0.1888582033

0.8657073885 **(8.3)**

$$\#\sigma_2 := \frac{-4 \cdot b \cdot evalf\left(int\left(H2 \cdot \left(-fp \right) \cdot f, t = 0 .. a \right) \right)}{\left(subs\left(\left\{ t = a \right\}, f \right) \right)^2};$$

$$\sigma_2 := \frac{4 \cdot b \cdot test}{evalf\left((subs(\{t = a\}, f))^2 \right)};$$

$$2.567025713 \qquad \textbf{(8.4)}$$

$$\#\sigma_{2\,t} := simplify\left(subs\left(\{t = a\}, \sigma_2\right)\right);$$

$$\sigma_{vm} := \sigma_1 - \sigma_2;$$

$$evalf\left(\sigma_{vm}\right);$$

$$-0.749549104$$

$$-0.749549104 \qquad \textbf{(8.5)}$$