Doctoral Thesis

Wood-fibre collapse upon drying

Amir Akbari

Department of Chemical Engineering
McGill University

April 2015

A thesis submitted in partial fulfilment of the requirements for the degree of Doctor of Philosophy

Preface

This is a manuscript-based thesis written in accordance with the McGill Guidelines for Thesis Preparation. It includes four submitted manuscripts (chapters 2, 3, 5, 6) and a manuscript in preparation (chapter 4). The first four have three authors whose contributions are detailed below.

- **A. Akbari (main author):** conducted literature review, developed theory, developed branch continuation (Appendix G) and stability analysis (Appendix H) codes, and wrote manuscripts.
- **R.J. Hill (coauthor):** supervised research and revised manuscripts.
- **T.G.M. van de Ven (coauthor):** supervised research and revised manuscripts.

The last manuscript has three authors whose contributions are outlined below.

- **A. Akbari (main author):** conducted literature review, developed theory, prepared experimental setup and procedures for experiments, developed core code for image processing (Appendix I), and wrote manuscript.
- **J.J. Zhang (coauthor):** conducted experiments according to procedures provided by A. Akbari, and wrote image processing code (Appendix J) based on core code provided by A. Akbari.
- **R.J. Hill (coauthor):** supervised research and revised manuscript.

MCGILL UNIVERSITY

Abstract

Faculty of Engineering

Department of Chemical Engineering

Doctor of Philosophy

Wood-fibre collapse upon drying

by Amir Akbari

Wood fibres are diverse in application and abundant in species. They are isolated from parent wood by pulping, where their morphological features appreciably change. Upon drying, adsorbed water is removed from amorphous parts of the cell wall, whereas absorbed water evaporates from wood cavities (*e.g.*, cell lumen and micropores). The former results in shrinkage, affecting the submicroscopic structure, and the latter causes collapse, affecting the cell shape. A comprehensive understanding of wood-fibre interaction with water is crucial to product development and is lacking in the literature. Water evaporates from large pores where it is held in wood tissues by capillary forces at moisture contents above the fibre saturation point (FSP). However, below the FSP, water is trapped in nanometer-sized pores where water molecules are directly bound to the hydroxyl groups. Shrinkage is caused by non-uniform contractions of the cell-wall layers, altering the effective mechanical properties, wall thickness, and lumen conformation as moisture content decreases. Collapse occurs through the coupling of the wall-elastic and water-surface-tension energies, bringing the walls into close contact as the air-water interface recedes into cavities. The overall volumetric shrinkage can be obtained by superimposing collapse and shrinkage deformations.

In this thesis, an elastocapillary model is developed to study drying deformations in wood fibres above the FSP in the collapse regime. The model comprises a circular elastic membrane with a hole at the center, interacting with simply connected (bubble) and doubly connected (bridge) menisci at the air-water interface. The dry-state conformation is determined from the stability of equilibrium branches at fixed liquid volume from the fully-saturated to collapse states. First, the stability of liquid bridges with a free contact line is determined with respect to arbitrary perturbations. Constant-volume and constant-pressure stability regions are constructed in the cylindrical volume versus slenderness diagram. Compared with liquid bridges pinned at two equal discs, equatorial and reflective symmetries are broken by the free contact line, altering the characteristics of bifurcations along the lower boundary. Here, pitchfork bifurcations unfold into turning points, and critical perturbations have no symmetry. Theoretical predictions of the stability limits are experimentally verified under neutral buoyancy. Moreover, the destabilizing effect of free contact lines is theoretically and experimentally demonstrated. Next, using spectral and variational methods, the stability of the elastocapillary model is rigorously related to the shape of equilibrium branches, supporting the principle of stability exchange in the catastrophe theory. An elastocapillary number is introduced, measuring the membrane axial rigidity relative to the water surface tension. Upper bounds on the critical elastocapillary number are determined as functions of the scaled hole radius and contact angle, providing ranges of geometrical and mechanical parameters in which the dry-state conformation is open. Estimating the model characteristic length scales from the structural features of Norway spruce fibres, it is shown that capillary-induced collapse over the length scales of pit-holes and the cell lumen is unlikely to contribute to the overall volumetric shrinkage.

UNIVERSITÉ MCGILL

Abrégé

Faculté de génie

Département de génie chimique

Docteur en Philosophie

Le rétrécissement et l'effondrement de la fibre de bois lors de l'assèchement

par Amir Akbari

Il y a divers sortes de fibres de bois et de nombreuses applications peuvent être proposées pour ce matériel. Pendant la procédé de mise en pâte, la morphologie des fibres de bois est altérée. Après la mise en pâte et l'assèchement, les gouttelettes d'eau peuvent ou bien adsorbées sur les parties amorphes des parois cellulaires ou bien absorbées dans les cavités. Dans le premier cas où les gouttelettes d'eau sont adsorbées, l'assèchement des parois cause le rétrécissement du lumen ce qui affecte la structure submicroscopique. Dans le deuxième cas où l'eau est absorbée dans les cavités, un effondrement se produit et altère la forme cellulaire. Une meilleure connaissance du phénomène est essentielle quant à la commercialisation et le développement puisque ce domaine de la science est encore mal compris. À un taux d'humidité plus élevé que le point de saturation d'un fibre (PSF), l'eau s'évapore des larges pores au lieu que de rester sur la surface céllulosique. À un taux d'humidité inférieur au PSF, les molécules d'eau sont immobilisées dans les pores nanoscopiques et ceux-ci interagissent directement avec les groupes hydroxyles. Le rétrécissement est causé par la contraction non-uniforme des couches dans la paroi cellulaire et provoque un changement dans les propriétés mécaniques, l'épaisseur de la paroi et la nature du lumen pendant que l'eau s'évapore. L'effondrement se produit par l'énergie élastique de la paroi et la tension superficielle. Ces facteurs causent le rapprochement des parois lorsque la

diminution du volume d'eau initie le rétrécissement des fibres. Ce changement de volume peut être calculé en superposant la déformation causée par l'effondrement et celle causée par le rétrécissement.

Dans cette thèse, le modèle de l'élasto-capillarité est développé afin d'étudier la déformation causée par l'assèchement lorsque le taux d'humidité est supérieur au PSF et lorsque les conditions favorisent l'effondrement. Le modèle en question est une membrane élastique et circulaire en forme de tube. Celui-ci interagit avec des ménisques simples (bulles) ou doubles (ponts) à une interface entre l'air et l'eau. La conformation à l'état sec est déterminée par la stabilité des branches d'équilibre à un volume de liquide fixe à partir d'un état complètement saturé jusqu'à l'état effondré. La stabilité des ponts liquides avec la ligne de contact est déterminée avec une perturbation aléatoire. Les régions de volume et de pression constantes sont construites dans le diagramme indiquant la relation entre le volume cylindrique et l'élancement. Si on compare ces régions avec les ponts liquides pris entre deux disques, les symétries équatoriales et réflectives sont brisées par la ligne de contact. Ce phénomène change les caractéristiques de la bifurcation à la limite inférieure. Les bifurcations en forme de fourche deviennent des embranchements et les perturbations graves n'ont aucune symétrie. Les prédictions théoriques à propos de la limite de stabilité sont expérimentalement vérifiées sous flottabilité nulle. De plus, l'effet déstabilisant des lignes de contact a été théoriquement et expérimentalement démontré. En utilisant des méthodes spectrales et variantes, la stabilité du modèle élasto-capillaire est associée à la forme des branches d'équilibres. Cette stabilité démontre le principe de l'échange de stabilité dans la théorie des catastrophes. Un chiffre élasto-capillaire est établi; ceci quantifie la rigidité de la membrane axiale par rapport à la tension superficielle. Les limites supérieures du chiffre élasto-capillaire sont liées au rayon et à l'angle de contact. Ces limites donnent une fourchette de valeurs pour les paramètres géométriques et mécaniques où la forme à l'état sec est ouverte. Les caractéristiques dimensionnelles des épinettes norvégiennes ont été utilisées pour cette œuvre. L'effondrement capillaire sur une échelle de longueur de diamètre d'un lumen de fibre ne contribue pas intégralement au rétrécissement volumétrique

Acknowledgements

I would like to thank my supervisor Prof. Reghan J. Hill and cosupervisor Prof. Theo G. M. van de Ven for their guidance, patience, and help. Accomplishing this thesis would not have been possible without their encouragement and assistance. I am also thankful to my family for their unconditional support. Contributions of the summer student, Jessie Zhang, to the experimental study of liquid-bridge breakup and image processing code is appreciated. Financial supports from the NSERC Innovative Green Wood Fibre Products Network and the McGill University Faculty of Engineering is gratefully acknowledged.

Contents

List of Figures

List of Tables

Abbreviations

BBT	**B**ubble to **B**ridge **T**ransition
CMC	**C**ritical **M**icelle **C**oncentration
FSP	**F**ibre **S**aturation **P**oint
MEMS	**M**icro **E**lectro **M**echanical **S**ystems
MFA	**M**icro **F**ibril **A**ngle
ML	**M**iddle **L**amella
MSR	**M**aximal **S**tability **R**egion
SDS	**S**odium **D**odecyl **S**ulfate

Physical Constants

Gravitational acceleration	g	$=$	$9.80665\ \mathrm{ms}^{-2}$

In memory of my mother

Chapter 1

Introduction

1.1 Wood-fibre chemistry and microstructure

Wood is a complex fibrous material. It has been used as fuel, construction material, and a host of other reconstituted products. It is also the main source of fibres for the papermaking industry. Commercial timbers are usually categorized into two types, namely, softwoods and hardwoods. The cellular structure of softwoods and hardwoods is quite different. Softwoods have a relatively simple structure and are more uniform than hardwoods. They are made up of fewer cell types with long fibrous cells called tracheids. Tracheids provide structural support and conducting pathways for water transport. Hardwoods, on the other hand, are composed of several different cell types with specialized cells for water transport, termed a vessel. Structural support in hardwoods is provided by another specialized cell, termed a fibre (Walker, 2006).

Tracheids in softwoods are long, slim cells with a length approximately one hundred times greater than the diameter and are almost rectangular in cross section. The hollow center of tracheids is the cell lumen. It has closed ends, which are rounded in the radial or pointed in tangential orientation. In mature wood, the length and width of tracheids range from 2 to 5 mm and about 15 to 60 μm, respectively (Walker, 2006). Libriform and fibre tracheids are two types of hardwood fibres, and together with vessels form the basic structure of hardwoods. Similar to the tracheid in softwood, libriform fibres are slender cells with pointed ends, in the range 0.7–2.0 mm long and 10–60 μm wide (Monica *et al.*, 2009).

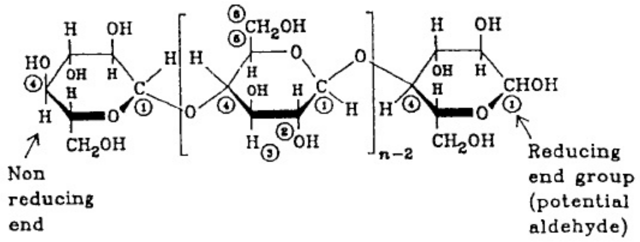


FIGURE 1.1: Cellulose structure (Bledzki & Gassan, 1999).

Cellulose, hemicelluloses, and lignin are the three basic components of all woods. Cellulose and hemicelluloses are polysaccharides, whereas lignin is an oxygenated polymer of phenylpropane units. Furthermore, there are other components, such as some extraneous chemicals known collectively as extractives and inorganic elements (*e.g.*, calcium, magnesium and potassium) (Walker, 2006). Cellulose is the essential component of all plant-fibres. It is widely accepted that cellulose is a linear condensation polymer consisting of D-anhydroglucopyranose units joined together by β-1,4-glycosidic bonds. Pyranose rings have a 4C1 conformation, which means that the -CH_2OH and -OH groups, and the glycosidic bonds are equatorial with respect to the mean planes of the rings (Bledzki & Gassan, 1999). Cellulose structure is shown in Fig. 1.1.

Microfibrils are long, thin crystalline filaments of cellulose in the fibre wall. The orientation of microfibrils with respect to the fibre axis varies in different parts of the wall, and is the principal cause of the anisotropic behaviour of wood-based materials. They are roughly oriented along the fibre axis, and this is why wood and fibres are stronger in the fibre direction and weaker in the transverse directions. This anisotropic behaviour is also observed in wood shrinkage. It is considered that the fibre geometry contributes to this anisotropic behaviour, but this is not as significant as the effect of the cellulose microfibrils. Moreover, cellulose is resistant to chemical attack due to its crystalline nature, so that during chemical pulping the majority of hemicelluloses and lignin can be removed having the remaining fibres rich in cellulose (Walker, 2006).

Hemicellulose is not a form of cellulose, as one might expect from the name. Hemicellulose is made of polysaccharides that remains associated with the cellulose after lignin is removed. There are three important differences between hemicellulose and cellulose. Firstly, they contain several different sugar units, whereas cellulose contains only 1,4-β-D-glucopyranose units. Secondly, they are branched, whereas

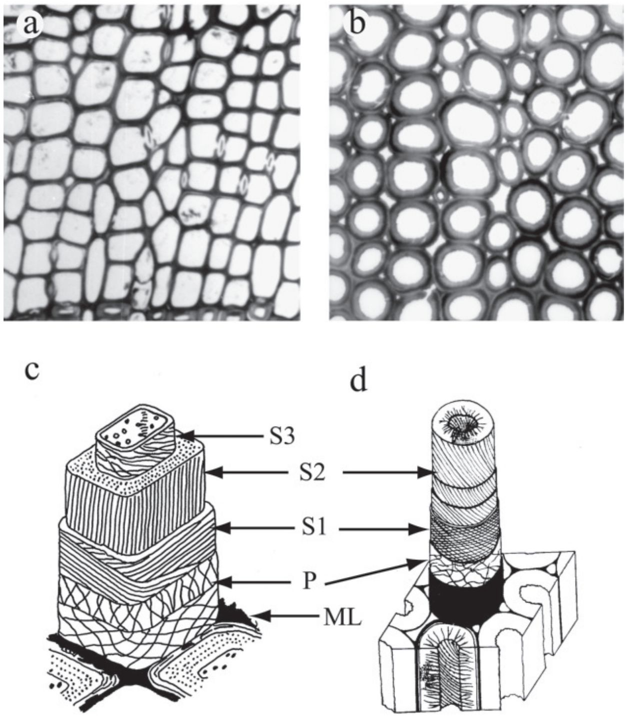


FIGURE 1.2: Cell wall structure; (a) normal wood, (b) compression wood, (c) normal-wood idealized schematic, (d) compression-wood idealized schematic (Kwon *et al.*, 2001).

cellulose is a linear polymer. Thirdly, the degree of polymerization of native cellulose is much higher than that of hemicellulose (Bledzki & Gassan, 1999, Walker, 2006). The intrinsic tensile strength of the cellulose molecule is very high (around 134 GPa); to some extent, the compressive strength is provided by lignin . It prevents slender microfibrils from buckling and hinders biodegradation. The function of the hemicelluloses is uncertain. The simplest role assigned to the hemicelluloses is to provide links between cellulose and lignin, allowing effective transfer of shear stresses. Hemicellulose can form hydrogen bonds with cellulose and sometimes an ester bond with lignin (Walker, 2006).

Lignins have a completely different structure than cellulose and hemicellulose. They are complex aliphatic and aromatic hydrocarbon polymers. There is no general consensus in the literature on the structural details of lignins. Mechanical properties are much lower than those of cellulose (Bledzki & Gassan, 1999).

The cell wall comprises three distinct layers: the middle lamella (ML), primary wall (P), and secondary wall (S). Figure 1.2 shows the three layers of the cell wall and the microfibril orientation in the various layers. The middle lamella is the intercellular region. It mainly comprises lignin and has no cellulose microfibrils.

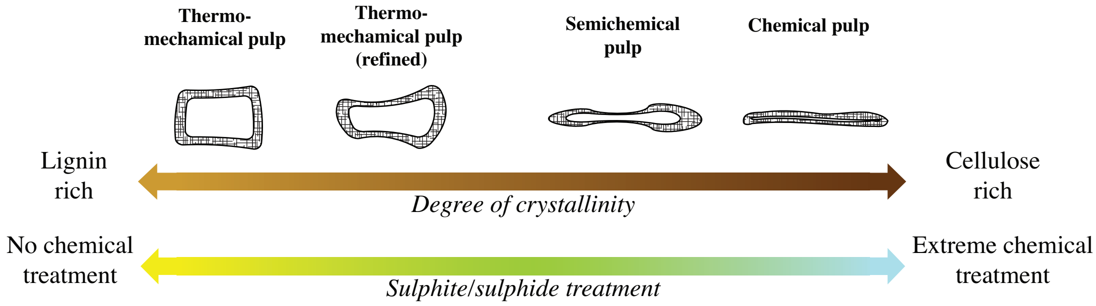


FIGURE 1.3: Typical fibre conformations from various pulping processes.

The primary wall is very thin (about 0.1 μm) and indistinguishable from the middle lamella. The middle lamella and primary wall were regarded together as one single compound middle lamella layer in many studies (Walker, 2006). The microfibrillar network of the primary wall is unstructured where microfibrils have random orientation, except near the corners of the cell where they are oriented along the fibre axis. Microfibrils in the primary wall are embedded in a matrix of hemicelluloses and pectic compounds. The secondary wall lies after the primary wall in which three distinct layers can be recognized: S_1, S_2, and S_3. Unlike the primary wall, the cellulose microfibrils are highly structured and oriented parallel to one another within these three layers. However, the microfibril orientation is different in the three layers of the secondary wall (Walker, 2006).

1.2 Pulping processes

Fibres are isolated from parent wood by a variety of pulping processes, including mechanical, semi-chemical, and chemical. The dry-state conformation of the resulting fibres depends on the extremities of chemical and thermal treatment (Fig. 1.3). Mechanical pulps are produced by disintegrating wood using rotating pulp stones or woodchipper machines at high temperature without chemical treatment. They are not suitable for bleaching to high brightness levels due to their high lignin content, and their mechanical strength is lower then that of chemical pulps. However, they exhibit excellent printing properties and high opacity. Chemical pulps, on the other hand, are produced by digesting wood chips in sulphate or alkaline solutions at high temperature where most lignin is removed. Due to their low lignin content, they can be highly brightened by bleaching. They also exhibit high mechanical strength, making them suitable for printing-grade papers (Walker, 2006).

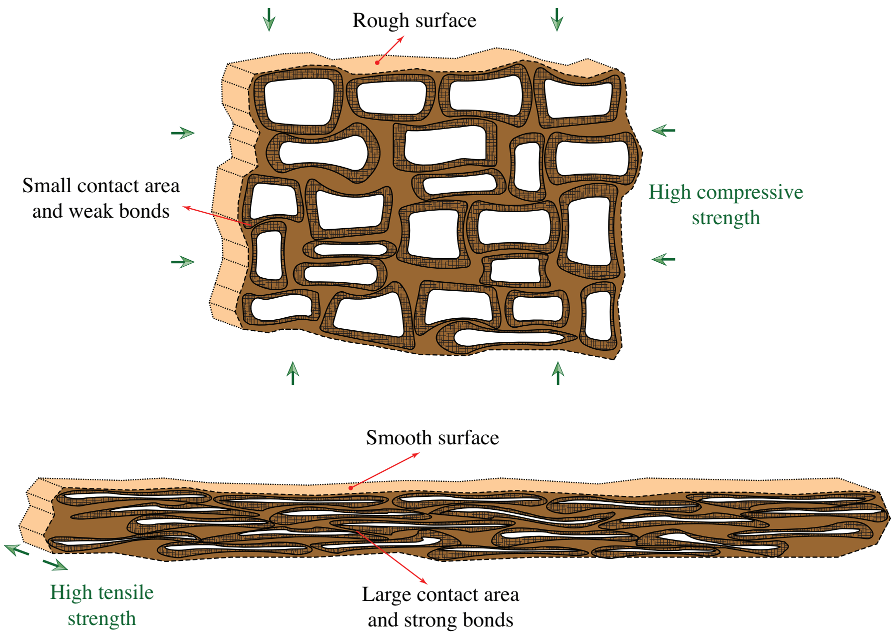


FIGURE 1.4: Contrasting properties of composites from fibres in open (top) and collapsed (bottom) states.

Thermomechanical pulping is a major variant of mechanical processes in which wood chips are presteamed at a temperature ranging from 120 to 140°C and then defibred in a disc refiner at 150 to 500 kPa. The resulting fibres are rich in lignin, thick-walled, bulky with an open conformation, and have high compressive strength (Fig. 1.4); consequently, they are suitable for high-speed printing techniques (Walker, 2006). Moreover, due to the small contact area, fibre-fibre interactions are weak in suspensions and slurries. Thus, they are vastly used for concrete reinforcement because of their low agglomeration tendency in cement slurries (Campbell & Coutts, 1980).

Semi-chemical and chemical pulping have similar processes. However, the chemical treatment is less extreme, and the cooking time is shorter in the former. Kraft pulping is a predominant chemical process based on sodium sulphate and sodium hydroxide solutions. Wood is initially defibred mechanically using much less electrical energy than in mechanical pulping. The resulting chips are fed into a digester operating in a temperature range 140–180°C at 0.6–1.0 MPa for delignification. Fibres produced by chemical pulping are lignin free, thin-walled, collapsed, and have high tensile strength (Fig. 1.4); thus, they are suitable for high quality printing applications (Walker, 2006). Furthermore, due to their high contact area in suspensions, they are prone to agglomeration in cement slurries and less desirable for concrete reinforcement (Campbell & Coutts, 1980).

1.3 Applications

Papermaking is an age-old craft, dating back to 200 BCE. Nowadays, applications of fibre-based materials have gone far beyond the printing paper. Using plant fibres as alternative packaging materials has been of increasing interest in the literature and industry (Dury-Brun *et al.*, 2006, Gällstedt *et al.*, 2005, Kjellgren *et al.*, 2006, Walker, 2006). Mechanical strength and low water vapour and oxygen transmission rate are the major requirements of packaging materials. However, cellulose fibres are porous and hygroscopic due to their hydroxyl groups, making them less attractive for food packaging (Dury-Brun *et al.*, 2006). As a result, chemical treatment and coating techniques have been commonly used to produce papers with competitive properties to their synthetic alternatives. In this regard, chitosan and protein coatings have proven effective for enhancing the mechanical and barrier properties of papers (Dury-Brun *et al.*, 2006, Gällstedt *et al.*, 2005, Kjellgren *et al.*, 2006).

In the last few decades, there has been enormous interest in using natural fibres as reinforcement for polymeric materials (Bledzki & Gassan, 1999, Mwaikambo, 2006). This is mainly due to the environmental manufacturing cost and energy required for synthetic fibres, such as glass, carbon, and Kevlar. Plant fibres also have other advantages over fossil-based fibres, such as low density, high specific stiffness, low cost, and renewability. They are also carbon neutral and environmentally friendly. The lower specific density of cellulose-based fibres leads to weight saving in composite products, making them ideal materials for the transportation industry (Mwaikambo, 2006). Nevertheless, the poor mechanical strength of these materials requires improvement. Moreover, on a weight-for-weight basis, many have claimed that the performance of the best plant-fibre reinforced and conventional glass epoxy composites are comparable (Mwaikambo, 2006). Thermosetting plant fibre composites and a polymer matrix can be used for electrical insulators and semi-structural applications. Another potential application for plant-fibre reinforced composites is in the building industry. For example, kenaf-polypropylene composites are reported as a competitive alternative for glass-polypropylene and mica-polypropylene composites (Mwaikambo, 2006).

Using natural fibres as reinforcement in composite materials is very challenging because of the problems arising from fibre-polymer interface due to imperfect bonding. As a result, fibre modification by chemical treatment has been extensively studied in the literature, attempting to enhance fibre-polymer compatibility. However, modifications of wood fibres also affect the mechanical properties. They can increase or decrease the fibre strength, so it is important to understand what occurs structurally during surface modification (Eichhorn *et al.*, 2001). Natural fibres require surface modification for other purposes. During pulping, for example, where fibres are separated through a chemical process, the intercellular lignin is removed together with some of the matrix materials in the fibre walls. The resulting fibre is more flexible and often collapses when dried, leading to a large fibre-to-fibre contact area for bonding. However, if fibres are hydrophobically modified, the lumen may remain open (Eichhorn *et al.*, 2001), imparting toughness to reinforced composites (Michell *et al.*, 1976).

Recently, new biomaterials have been developed from natural fibres with applications to tissue engineering (Cheung *et al.*, 2009). Plant fibres are non-corrosive, biodegradable, biocompatible, and have high fracture toughness. Conventional materials commonly used to fabricate biocomposites, implants, and medical devices are epoxy, polyester resin, polyurethanes, stainless steel, and titanium. Although biocompatible, these materials are not degradable, requiring several subsequent surgical operations to be removed at the end of their life time. These materials also interfere with the natural growth of the surrounding tissues (Cheung *et al.*, 2009). Mechanical tests on new composites from biodegradable polymers such as polyglycolide and polylactic acid reinforced with plant fibres have been promising (Cheung *et al.*, 2009). Scaffolds from cellulose fibres have also been established for cartilage and bone tissue engineering (Müller *et al.*, 2006).

1.4 Wood-water interactions during drying

Wood tissues are naturally formed in a saturated environment. Water affects the non-crystalline constituents of the cell wall, *i.e.*, the hemicelluloses and lignin, to some degree. As a result, wood experiences dimensional changes from losing and gaining water. Structural and mechanical properties are also affected by hydration and dehydration. It is, therefore, important to determine how much water wood tissues hold in their native state, at various stages in pulping processes, and in the end products. This is measured by the moisture content, which is defined as the weight ratio of water to solid content (Walker, 2006).

The extent to which wood tissues are affected by drying depends on the strength of the drying stresses arising from water removal. These stresses are correlated with the energy required to evaporate a unit volume of water held within wood tissues relative to that required to evaporate a unit volume of bulk water. This relative energy is the excess enthalpy of vaporization, which reflects how strongly water molecules are associated with wood tissues. Accordingly, water in wood tissues is classified into three types (Nakamura *et al.*, 1981, Weise *et al.*, 1996): (i) free water, having the same transition temperature as bulk water, (ii) freezing bound water, having lower transition temperature than bulk water, and (iii) non-freezing bound water, which cannot be detected from the first-order transition (see Fig. 1.5). Molecules of non-freezing bound water are directly bound to cellulose hydroxyl groups, whereas those of freezing bound water and free water are trapped in the lumen and cell-wall micropores by capillary condensation. Different types of water in this classification can be quantified using Differential Scanning Calorimetry (Weise *et al.*, 1996).

Classifying water into adsorbed and absorbed is another common way of identifying different types of water (Walker, 2006). Defining absorption as taking up of water by capillary condensation (Walker, 2006), the two foregoing classifications can be related. Accordingly, non-freezing bound water can be regarded as adsorbed, whereas freezing bound and free water can be regarded as absorbed. Based on the latter classification, the fibre saturation point (FSP) is defined as the moisture content at which all the absorbed water has evaporated, and the cell wall is saturated only with adsorbed water. Based on this definition, the FSP typically corresponds to $\sim 30\%$ moisture content.

The FSP is an important concept for understanding dimensional changes at various drying stages. Based on the definition of the FSP, two drying-deformation regimes are identified (Tiemann, 1941): (i) collapse above the FSP where fibre macro- and microscopic structures are affected, and (ii) shrinkage below the FSP where the cell-wall submicroscopic structure (ultrastructure) is affected (see Fig. 1.5). Drying stresses are characterized by the enthalpy of vaporization of water with curved interfaces in the former and the heat of desorption in the latter. Since the heat of desorption for breaking water-hydroxyl group bonds for non-freezing bound water is much larger than the excess enthalpy of vaporization of water held in wood tissues by capillary condensation for freezing bound and free water

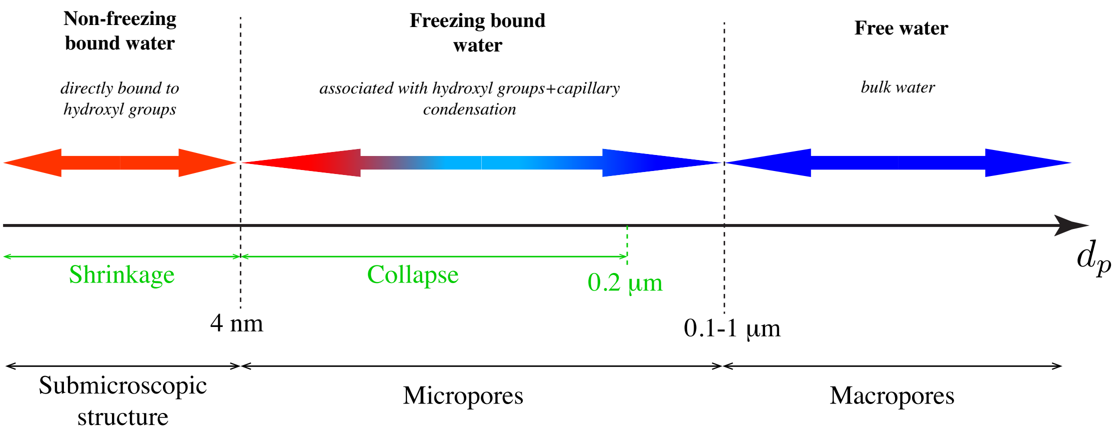


FIGURE 1.5: Classification of water in wood tissues with respect to pore size.

(Walker, 2006), drying stresses are expected to be much larger in the shrinkage regime than in the collapse regime.

Successive hydration and dehydration of wood results in a decrease in the water retention value, referred to as hornification in the literature (Weise *et al.*, 1996). How much wood tissues are affected by drying is also related to the drying intensity (Walker, 2006, Weise *et al.*, 1996). Depending on the drying intensity, a fraction of pores may irreversibly close and become inaccessible to water, reducing the water retention value in subsequent cycles. Experiments on low-yield Kraft pulp revealed that there is a critical moisture content, typically between 30% and 50%, below which the water retention value significantly drops (Robertson, 1964, Weise *et al.*, 1996). Laivins & Scallan (1993) identified this critical moisture content as the FSP. The trend of variations in the water retention value with moisture content was also shown to significantly depend on the duration of drying and temperature (Weise *et al.*, 1996). These observations point to a disparity in drying stresses above and below the FSP, manifesting in the shrinkage and collapse regimes. This implies that, well above the FSP, capillary forces are not strong enough to induce collapse and permanent closure, so that the water retention value remains unaffected. However, below the fibre saturation point, drying stresses are strong enough to permanently close cell-wall micropores and intermolecular voids, significantly reducing the water retention value.

As shown in Fig. 1.5, different types of water can be correlated with the pore size d_p. Accordingly, the trapped water in macropores, micrometer-sized pores, and nanometer-sized pores are respectively identified as free, freezing bound and non-freezing bound (Topgaard & Söderman, 2002). Moreover, $d_p \approx 4$ nm is the

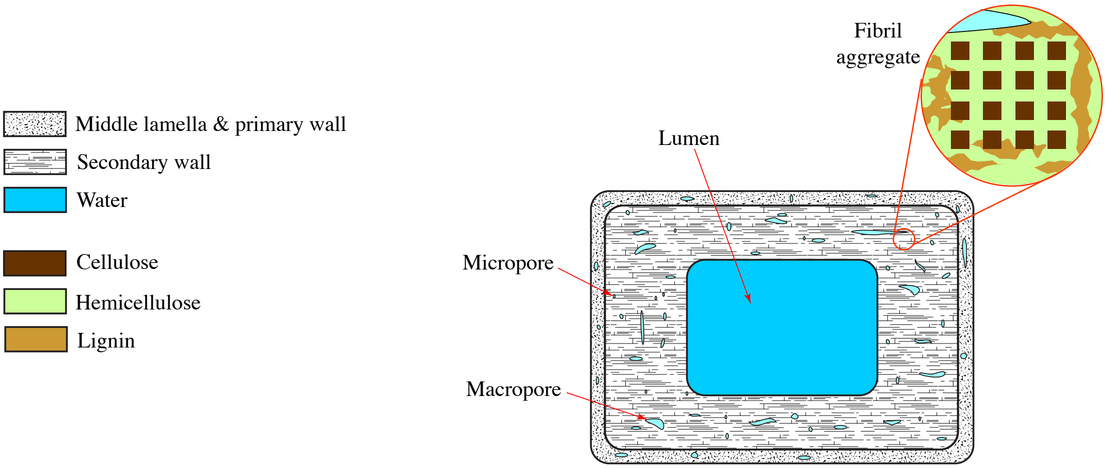


FIGURE 1.6: Schematic of cell-wall ultrastructural features.

pore size separating the shrinkage and collapse regimes. This has been reported in the literature as the largest pore in which only non-freezing bound water can be found (Weise *et al.*, 1996). Below this pore size, wood-water interactions can be described by multilayer adsorption models (Walker, 2006).

Because wood-water association energy for adsorbed water is much larger then for absorbed, the former is believed to evaporate after the latter is completely removed (Walker, 2006). Using Differential Scanning Calorimetry measurements, Weise *et al.* (1996) studied the chronology of water removal by examining parallel and consecutive evaporation scenarios. In the former, all three types evaporate simultaneously at any moisture content during drying, whereas, in the latter, the type with larger association energy starts to evaporate after other types with smaller association energy are completely removed. Experimental measurements on chemical and mechanical pulps revealed that at large moisture contents, different types of water evaporate consecutively. Thus, when the moisture content is well above the FSP, only free water evaporates. However, in an intermediate range of moisture contents around the FSP, there is a transition from consecutive to parallel evaporation where two or all three types can evaporate simultaneously. In the final phase of drying, where the moisture content is very small, only non-freezing water is left, and so the consecutive scenario applies (Weise *et al.*, 1996).

1.5 Cell-wall ultrastructure

Cell-wall structure at the nanometer level is termed ultrastructure in the literature; it significantly influences the mechanical properties of wood fibres (Cristian Neagu *et al.*, 2006). Pore-size distribution, pore geometry, and composition of wood polymers within the cell wall are among the ultrastructural features (see Fig. 1.6). Pore-size distribution is particularly important for estimating drying stresses at various drying stages. Experimental measurements indicate that the BET surface area of the cell wall is roughly three orders of magnitude larger than the lumen surface area, demonstrating that there is a massive internal surface area within the cell wall for water adsorption (Walker, 2006). From the BET theory, multilayer adsorbed water is ∼4–5 nm thick, which corresponds to the pore size separating adsorbed and absorbed water in the cell wall.

1.5.1 Pore-size distribution

Several techniques have been proposed in the literature for measuring the pore-size distribution of porous materials. Mercury intrusion porosimetry is a classical technique, which has been widely used to determine the pore-size distribution of various porous materials, including cement paste, concrete, solid catalysts, and rocks (Burdine *et al.*, 1950, Diamond, 2000, Kong *et al.*, 2002). Stone & Scallan (1968) introduced the solute exclusion technique to determine the pore-size distribution of the cell wall and distinguish between the adsorbed and absorbed water. In this method, samples are saturated in a solution prepared with a series of water soluble polymer probes in the range 0.8–5.6 nm, which are small enough to penetrate the cell-wall micropores. Only non-freezing bound water exists in the pores that are accessible to these probe (see Fig. 1.5). Adsorption of the polymer molecules on the cell-wall internal surface area dilutes the polymer concentration in the bulk solution, so the difference from the initial concentration furnishes the accessible pore volume and pore diameter. A third of all the adsorbed water was found in pores smaller than 0.8 nm, and the maximum pore size carrying 100% adsorbed water was measured ∼3.6 nm.

Alince & van de Ven (1997) studied the pore-size distribution of solvent-exchange Kraft pulps and found that the distribution is bimodal with the characteristic pore sizes 4 and 75 nm separating micropores respectively from submicroscopic structure and macropores (see Fig. 1.5). Drying from the fully-saturated state, the transition from the free to freezing bound water and from freezing bound to non-freezing bound water respectively occur in the moisture content ranges ∼70–80% and ∼20–30% (Topgaard & Söderman, 2002). Moreover, beating and chemical treatment increase the mean pore size and hornification (Hafren *et al.*, 1999, Topgaard & Söderman, 2002, Weise *et al.*, 1996). Pore-size distribution and porosity also change considerably during drying (Park *et al.*, 2006, Topgaard & Söderman, 2002), and, therefore, are not intrinsic structural properties of wood species. The geometry of cell-wall pores are not well-defined and are characterized by several length scales (Hafren *et al.*, 1999, Topgaard & Söderman, 2002, Walker, 2006). Transmission electron microscopy of the primary and secondary walls revealed that cell-wall micropores are slit-like and scattered between microfibrils that are held together by numerous crosslinks passing through the micropores (Hafren *et al.*, 1999).

1.5.2 Composition of wood polymers

As previously discussed, cellulose, hemicellulose, and lignin are the main polymers in wood. The concentration of these polymers varies within each cell-wall layer. Table 1.1 summarizes typical compositions of wood polymers in the cell-wall layers for Norway spruce. How microfibrils are oriented is also an ultrastructural feature of wood fibres (Cristian Neagu *et al.*, 2006). The extent to which the cell wall responds to drying stresses depends on its mechanical properties, which, in turn, depend on the foregoing ultrastructural properties of wood fibres. Table 1.2 summarizes typical values of the longitudinal Young modulus, Poisson ratio, and microfibril angle (MFA) in each layer reported in the literature for the main wood polymers.

TABLE 1.1: Structural and ultrastructural properties of the cell-wall layers in Norway spruce early wood (Cristian Neagu *et al.*, 2006). Compositions are reported as volume fractions.

Layer	Thickness (μm)	Cellulose	Hemicellulose	Lignin	MFA (°)
P	0.11	0.12	0.26	0.62	—
S_1	0.25	0.35	0.30	0.35	70–90
S_2	1.54	0.50	0.27	0.23	30
S_3	0.06	0.45	0.35	0.20	40–50

Table 1.2: Mechanical properties of wood polymers (Cristian Neagu *et al.*, 2006).

Polymer	Young modulus (GPa)	Poisson ratio
Cellulose	120–140	0.38
Hemicellulose	2–8	—
Lignin	3	—

The cell wall is a multilayered composite of wood polymers, exhibiting macroscopic mechanical properties that reflect its microstructure and the intrinsic properties of its constituent polymers (Akbari *et al.*, 2013, Torquato, 2002). These properties are also moisture dependent. Moreover, the cell-wall microstructure changes during drying. Consequently, quantifying wood-fibre deformations upon drying is challenging. Several studies experimentally measured the mechanical properties of individual fibres without controlling the moisture content (Groom *et al.*, 2002, Yan & Li, 2008) since control and measurement of the moisture content in these mechanical tests are difficult. In contrasts, theoretical models based on homogenization techniques can reliably account for the microstructuralal changes and partial saturation of composite materials (Hofstetter & Gamstedt, 2009, Torquato, 2002), which have been commonly used in the literature for modelling drying deformations in wood fibres (Thuvander *et al.*, 2002).

1.6 Literature review

Early studies in the 1930s and 1940s on drying-induced wood deformation indicated that changes in the dimension and mechanical properties of wood specimens linearly vary when the moisture content is below the FSP, and they are negligible above the FSP (Stamm, 1935, Walker, 2006). However, improvement in imaging techniques revealed two distinct drying deformation regimes at the submicroscopic and microscopic structure levels, termed shrinkage and collapse (Tiemann, 1941). Moreover, recent observations indicate that drying deformations can start at moisture contents well above the FSP (Hernández & Pontin, 2006). Numerous theoretical studies in the literature have examined shrinkage below the FSP, which is reviewed in section 1.6.1. Although the collapse of the lumen and micropores of the cell wall has been experimentally verified (Hafren *et al.*, 1999, Topgaard & Söderman, 2002, Walker, 2006), theoretical studies in this area are scarce (Innes, 1995). Deformations and collapse of cavities in wood resulting from drying stresses are characterized by interactions between elastic and capillary forces. Nevertheless, the physics of this problem are similar to those of elastocapillary systems, which have recently attracted attention in the literature. Section 1.6.2 reviews the relevant elastocapillarity literature.

1.6.1 Wood shrinkage

Barber & Meylan (1964) developed one of the first major theoretical models to quantify shrinkage below the FSP. This model, known as the laminate theory, only accounts for the S_2 layer and predicts anisotropic shrinkage as a function of the MFA. They considered a flat element of the cell wall comprising a matrix of amorphous materials in which crystalline microfibrils are embedded. The amorphous part linearly and isotropically shrinks with decreasing the moisture content, whereas the crystalline part remains intact. As a result, the overall shrinkage is anisotropic because the expansion or contraction of the amorphous part is constrained by microfibrils. They also assumed that the MFA in the S_2 layer is constant. This model relates anisotropic shrinkage to the MFA and explained why longitudinal shrinkage increases with the MFA. This model was later modified to account for the compound ML layer and curved cell wall (Barber, 1968). Here, the compound ML is assumed to be an elastic sheath with high rigidity, imposing an additional constraint on the cell-wall shrinkage. This modification can account for negative radial shrinkage, agreeing with the few experimental observations reported in the literature.

Cousins (1976) studied the moisture-content effect on the Young and shear moduli of lignin. He examined Klaxon, periodate, and dioxan lignin and found that both moduli increase linearly with decreasing the moisture content. However, one should cautiously apply these data to estimate the cell-wall rigidity. Note that native lignin is intimately mixed with cellulose and hemicellulose in the cell-wall microstructure, so it cannot be tested as an independent entity. Therefore, lignin must be isolated through a separation process, whereupon its chemical and physical structure are altered. Cousins (1978) studied the moisture-content dependency of the Young modulus for hemicellulose. Here, the modulus also increases with decreasing the moisture content. However, the relationship is not linear, and the modulus is larger than that of lignin.

Yamamoto (1999) included the S_1 layer in his analysis. To accurately predict the volumetric shrinkage from the FSP to the dry state, this model accounts for the moisture-content dependency of the mechanical properties of the amorphous matrix in each layer. The volumetric shrinkage was compared with experimental data for Jeffrey pine fibres in the range MFA=20°–50° (Yamamoto *et al.*, 2001). The cell-wall Young modulus was calculated from that of wood polymers using a simple mixing rule based on a cross-sectional-area weighted average.

Thuvander *et al.* (2002) considered S_1, S_2, and S_3 layers in the laminate theory to estimate drying stresses in wood fibres. They assumed that lignin is isotropic and treated cellulose and hemicellulose as anisotropic materials. Experimental data of Cousins were used to account for the moisture-content dependency of elastic moduli. Volume-fraction-weighted averages were used to relate the composite mechanical properties to those of the constituent wood polymers. From the very high drying stresses that the model predicted, they concluded that drying wood below the FSP would lead to irreversible changes in the cell wall at the ultrastructure level.

Shrinkage measurements are typically performed on centimetre-sized specimens of wood, comprising numerous fibres that are bundled by the ML matrix. Thus, the overall specimen shrinkage simultaneously reflects the cell-wall shrinkage and changes in the shape of individual fibres. To account for the latter effect, Pang (2002) introduced semi-imperial relations, connecting the sample macroscopic strains to those of the cell wall predicted by the model of Barber & Meylan (1964). The modification improves on the model of Barber & Meylan (1964) by predicting a macroscopic shrinkage larger in the tangential direction than in the radial direction at small MFAs, agreeing with experimental observations. Pang (2002) also applied this model to multilayer composites with nonuniform mechanical and structural properties across the thickness, predicting an asymmetric stress distribution in the sample, which explains experimentally observed bows, crooks, and twists.

The mechanical properties of the cell wall as a multi-scale composite of amorphous and crystalline materials can be more accurately estimated from its microstructural features and constituent material properties by micromechanical models (Hofstetter & Gamstedt, 2009). Halpin & Kardos (1976) reviewed several classical models based on the self-consistent scheme, providing closed-form expressions for the effective moduli of fibre-reinforced composites. Salmén & de Ruvo (1985) showed how discontinuous reinforcing elements can be treated in Haplin-Tsai equations

(Halpin & Kardos, 1976). Hofstetter *et al.* (2005) examined the hierarchical structure of wood, upscaling cell-wall structural features at three homogenization levels. This model rigorously treats the interaction of water with hemicellulose and lignin at the ultrastructure level, reliably estimating the cell-wall effective moduli.

1.6.2 Elastocapillarity

Studying the fabrication processes of microelectromechanical systems (MEMS), Mastrangelo & Hsu (1993*a*) examined the capillary-driven collapse of microstructures during wet etching. They considered the elastic deformation of beams and plates and developed an elastocapillary model, comprising an elastic structure deformed by the capillary pressure of curved interfaces, to predict collapse in MEMS from geometrical and mechanical properties. In this model, neglecting the exact geometry of menisci by omitting the corresponding energy term, equilibria were determined from the stationary points of the total energy. Open conformations in the dry state were attributed to stability loss at critical states. Without presenting a formal proof, a general concept was adopted from catastrophe theory (Arnold, 1992, Seydel, 2009), stating that turning points[1] with respect to the problem constraint are the points of stability exchange[2] along equilibrium branches. Applying this concept, a collapse criterion was derived in terms of an elastocapillary number, measuring the plate rigidity relative to the liquid surface tension. This model is based on von Kármán's plate theory (Timoshenko *et al.*, 1959) and accounts for both the stretching and bending contributions to the total energy. However, Mastrangelo & Hsu (1993*a*) greatly simplified the problem by assuming that the axial force tensor is isotropic and constant throughout the plate.

The collapse criteria derived by Mastrangelo & Hsu (1993*a*) only ensures that the microstructure walls make contact. However, whether the structure remains collapsed when completely dried depends on the strength of adhesion energy. Mastrangelo & Hsu (1993*b*) derived the corresponding criteria in terms of an elastocapillary number, measuring the ratio of the wall elasticity and the interfacial

[1]Turning point is a critical state of a nonlinear system where the derivatives of all the dependent variables with respect to the branching parameter are zero.

[2]Stability exchange refers to a situation where two equilibrium branches intersect at a bifurcation point, splitting each branch into a stable and an unstable segment in an open neighbourhood of the intersection. While tracing equilibrium states, stability is exchanged between the two branches at the bifurcation point (Seydel, 2009). These bifurcations are non-generic and break into two folds when system parameters are perturbed. The resulting folds, also referred to as points of stability exchange (Maddocks, 1987), connect a stable and an unstable segment.

energy between the solid walls. The model predictions were then validated by experimental data for polycrystalline silicon microstructures.

Bico *et al.* (2004) studied the capillary rise of liquids between flexible lamellae arising in the self-assembly and aggregation of paint-brush bristles. When an arrangement of two parallel lamellae of length L clamped and spaced apart by a distance d on a rigid support is withdrawn from a bath, the liquid wets a length L_w that is greater than Jurin's height $L_J = 2\ell_C^2/d$. Here, $\ell_C = \sqrt{\gamma/\rho g}$ is the capillary length where γ and ρ are the liquid surface tension and density. This problem is governed by a balance between capillary and elastic forces. Bico *et al.* (2004) found that L_w increases linearly with L, while the dry length $L_d = L - L_w$ remains constant. They showed that this problem is characterized by the elastocapillary length

$$\ell_{EC} = \sqrt{D/\gamma}, \tag{1.1}$$

furnishing $L_d^4 = 9d^2\ell_{EC}^2/2$, where D is the bending rigidity (Timoshenko *et al.*, 1959). They also extended this model to a cluster of lamellae and found that the lamellae aggregate into bundles with various sizes, in a self-similar arrangement. This model furnishes the maximum number of lamellae per bundle. Boudaoud *et al.* (2007) extended this work by accounting for the size distribution of lamellae.

Kim & Mahadevan (2006) quantified the scaling estimates of Bico *et al.* (2004) by applying the linear theory of plates to determine the lamella profile. The stretching energy is neglected in the linear theory. They considered the Laplace pressure in the equations of equilibrium and the contact-line force in the boundary conditions. Moreover, they identified two regimes corresponding to lamellae with (i) separate ends and (ii) contacting ends for capillary rise. The theoretical predictions of capillary rise in these regimes were also validated experimentally. Kwon *et al.* (2008) considered an analogous problem, comprising a liquid drop that bridges elastic and rigid plates. They approximated the meniscus meridian curve as a truncated circle. Here, unlike the work of Kim & Mahadevan (2006), the liquid volume is the control parameter, leading to a different scaling estimate for the wet length L_w.

Taroni & Vella (2012) studied the drying-induced stiction of two elastic plates clamped on a rigid support, trapping a prescribed volume of liquid. They accounted for the Laplace pressure and contact line force in the equations of equilibrium, where the meniscus meridian curve was assumed to be circular. Unlike the work of Kwon *et al.* (2008), the liquid completely fills the gap between the plates, furnishing a more relevant model for structural failures in the fabrication of MEMS. The stiction regimes identified here were similar to those of Kim & Mahadevan (2006). For a given liquid volume, this problem may have several equilibrium solutions. Therefore, Taroni & Vella (2012) applied a linear stability analysis to determine the stable equilibria and found that multiple stable equilibria can coexist. Then, using lubrication theory to solve the flow field between the plates, they performed a dynamic analysis to determine which stable equilibrium solution is realized in practice. They showed that the initial dynamics determine which equilibrium solution is reached as the liquid is removed, and that the system does not necessarily evolve towards a global minimum of the potential energy.

Duprat *et al.* (2012) examined the behaviour of flexible fibre arrays in contact with water droplets. They studied the basic structure, comprising two parallel cylindrical fibres that are clamped to a rigid support, deforming in response to a drop deposited near the support. Here, for a given drop volume v, fibre length L, and fibre spacing to fibre diameter ratio d/r, they identified three spreading regimes: (i) no spreading, (ii) partial spreading, and (iii) total spreading. For a fixed d/r, they derived the scaling estimate $v \sim L$ corresponding to the boundary between total and partial spreading. Moreover, $v \sim L^9$ furnishes an estimate of the boundary between spreading and no spreading.

Elastocapillary imbibition is another class of problems, arising in biomimetic adhesives, closure of pulmonary airways, and failure of micro-machined structures (Aristoff *et al.*, 2011, Duprat *et al.*, 2011, van Honschoten *et al.*, 2007, Kim & Kim, 2012). When an open-end capillary tube with flexible walls is connected to a reservoir at atmospheric pressure, a meniscus forms at the air-liquid interface, which generates a pressure gradient between the liquid in the reservoir and behind the interface, driving the liquid into the tube. The liquid pressure inside the tube is negative with respect to atmospheric pressure, causing the tube wall to radially contract. If the tube is long enough, the tube wall may completely collapse, blocking the liquid flow. Aristoff *et al.* (2011) studied the dynamics of elastocapillary imbibition in the gap between two elastic plates and determined the conditions under which the plates coalesce. They found the characteristic length scale $\ell_c = [Dd^2/(4\gamma \cos\theta_c)]^{1/4}$ and time scale $t_c = [9D\eta^2/(\gamma^3 \cos^3\theta_c)]^{1/2}$ corresponding to the onset of coalescence, where η and θ_c are the liquid viscosity and contact angle.

Capillary origami and folding is an emerging field of research, which also hinges on interplays between elastic and capillary forces (Péraud & Lauga, 2014, Py *et al.*, 2007, 2009). Creating complex three-dimensional objects from two dimensional structures is a promising space-saving strategy, which is desirable in many industrial and engineering applications. Capillary forces are particularly relevant to microfabrication and surface patterning because the surface tension dominates bulk forces at small scales. For example, the self-organization of micropillars by capillary forces has been demonstrated to produce hierarchical structures at the micrometer scale (Chandra & Yang, 2009, Pokroy *et al.*, 2009). These are well-defined helical patterns with controllable roughness and handedness, with applications to dry adhesive and particle-trapping systems. Py *et al.* (2007) studied the capillary-induced folding of flexible sheets. If a drop of liquid is placed onto an elastic sheet, it may completely encapsulate the drop as the liquid evaporates, depending on the rigidity, size, and shape of the sheet. Numerically solving the equations of equilibrium, Py *et al.* (2007) calculated the critical encapsulation length for square and triangular sheets $\ell_c \sim 10\ell_{EC}$ below which capillary forces are not strong enough to completely wrap the sheet around the drop. To elucidate this, they constructed equilibrium branches and showed that there is no continuous solution branch from the initial state to the closed state when $L < \ell_c$.

1.7 Objectives

Drying wood from the fully-saturated state results in deformations and structural changes across multiple length scales. Large pores are initially affected by drying stresses as free and freezing bound water evaporate above the FSP. At the final stage of drying well below the FSP, the remaining water is only non-freezing bound, and nanometer-sized pores and interfibrillar spaces are affected. Wood shrinkage below the FSP has been extensively studied in the literature. Theoretical models quantitatively agree with the experimental measurements of the anisotropic shrinkage. In the shrinkage regime, drying stresses are strong enough to completely close all the void space, and the overall volumetric shrinkage is expected to vary linearly with the moisture content. This is supported by experimental observations and is the basis of the foregoing theoretical models. However, the overall volumetric shrinkage has been shown to extend far beyond the FSP with a nonlinear trend for many species (*e.g.*, Hernández & Pontin (2006)), where dimensional changes can be attributed to capillary-induced deformations in the collapse regime. Interactions between elastic and capillary forces determine to what extent these deformations may contribute to the overall volumetric shrinkage.

Studies on the collapse of cell-wall micropores and the lumen above the FSP are scarce. This is mainly due to the complex and hierarchical structure of the cell wall and diversity among the species, which are not fully understood and established (Cristian Neagu *et al.*, 2006). Another challenge is measuring the cell-wall mechanical properties, particularly, those of wood polymers. The cell-wall ultrastructure, and, consequently, its mechanical properties change during drying, requiring extensive mechanical tests under controlled moisture content. Wood polymers isolated from the cell wall through separation processes have different physical and chemical properties than in their native state (Cousins, 1976, 1978). Therefore, applying micromechanical models to account for changes in structural and mechanical properties during drying is also not straightforward.

As discussed in section 1.6, there is no comprehensive theoretical model in the literature to elucidate elastic-capillary force interactions and provide collapse criteria based on the geometrical and mechanical features of the lumen and cell-wall micropores. However, the analogous problem in elastocapillary systems, namely, drying-induced structural collapse in MEMS has recently attracted a lot of attention. Here, dry-state conformations are determined by stability analyses. These investigations mostly apply dynamic simulations or linear stability analyses, which are computationally expensive and suitable for case studies (*e.g.*, Taroni & Vella (2012)). Moreover, the studied geometries are not necessarily representative models for the lumen and cell-wall micropores. Therefore, the following general objectives are pursued in this thesis:

- Understand drying deformations in the collapse regime, and whether they can contribute to the overall volumetric shrinkage.
- Develop an elastocapillary model that captures the basic elements of the underlying physics to elucidate elastic-capillary force interactions above the FSP during drying and derive collapse criteria. The model is not required to resemble the exact geometry of the lumen or cell-wall micropores. Rather, a simple multidimensional geometry that captures the important length scales

is sought. Emphasis is placed on tractable models that can readily be integrated into stochastic homogenization schemes, furnishing a macroscopic description of the cell-wall volumetric shrinkage from size distributions of micropores.

- Derive stability criteria based on catastrophe theory, where stability is related to the shape of equilibrium branches. This provides an algorithmically convenient setting to examine collapse criteria in the parameter space, and to construct phase diagrams.

- Apply the above stability criteria in a computational framework to predict the dry-state conformation from mechanical properties and geometrical parameters, demonstrating whether conformational changes in the collapse regime are practically controllable in pulping processes.

1.8 Numerical methods

This thesis mainly concerns the quasi-static evolution of elastic membranes and capillary surfaces. These are well-known nonlinear mechanical systems with complex equilibrium structures. They evolve through a sequence of equilibrium states in the quasi-static regime, following equilibrium trajectories. Studying nonlinear systems is generally challenging because multiple equilibrium solutions may exist, and equilibrium branches may bifurcate. In this thesis, Keller's arclength continuation method, as described by Seydel (2009), is applied to construct solution branches and identify branch points. Nonlinear systems of equations are solved using predictor-corrector iterative methods. A first-order continuation is used, where the solution is approximated from the slope of the solution branch in the prediction step. The approximated solution is refined using the Newton-Raphson method in the correction step. Initial value and boundary value problems are solved using the fourth-order classical Runge-Kutta and finite-difference methods, respectively. The trapezoidal rule is applied for numerical integrations.

Several open source packages have been written for the dynamic and bifurcation analysis of finite-dimensional systems, such as AUTO and BIFPACK (Seydel, 2009). Finite-dimensional systems (*e.g.*, particulate systems) are dynamically described by n equations of motion. Consequently, the state of the system at equilibrium is determined by solving n algebraic equations, and equilibrium branches must be traced in $\mathbb{R}^n$. In contrast, elastic membranes and capillary surfaces are continua, the dynamics of which are described by transient partial differential equations. They are infinite-dimensional mechanical systems where a boundary-value problem determines their equilibrium states. Accordingly, equilibrium branches must be traced in an appropriate function space. Since branch continuation in infinite-dimensional spaces are practically impossible, equilibrium branches are numerically approximated by discretizing the equations of equilibrium and traced in $\mathbb{R}^n$ using standard packages for finite-dimensional systems (Seydel, 2009). These computations are substantially expensive, and, thus, discretization is usually done on coarse grids.

Transforming an infinite-dimensional system into a finite-dimensional system is a computationally efficient alternative to the forgoing technique, provided the equations of equilibrium can be analytically solved. In this thesis, axisymmetric simply connected (bubble) and doubly connected (bridge) capillary surfaces are studied under zero-gravity by solving the Young-Laplace equation. Solutions belong to the families of axisymmetric surfaces, including sphere, cylinder, catenoid, nodoid, and unduloid, all can be analytically represented. Among these, the sphere is specified by one, while others are specified by three geometric parameters (Myshkis *et al.*, 1987). Here, finding a set of parameters, referred to as canonical, that can be uniquely associated with an equilibrium state is crucial. For example, as discussed in chapters 2 and 3, axisymmetric liquid bridges are uniquely specified by the meridian-curve arclengths at the boundaries and a shape parameter. These parameters furnish a three-dimensional parameter space to trace equilibrium branches. They are implicitly related to experimentally controllable parameters, referred to as favourable (*e.g.*, bridge volume, slenderness, and contact angles), by the boundary conditions and constraints. The relationships between the canonical and favourable parameters furnish a system of nonlinear algebraic equations, which can be examined using the foregoing packages for bifurcation analysis. However, these relationships merely provide a mapping between the finite-dimensional space of parameters and function space of equilibrium solutions, and, consequently, do not describe near-equilibrium dynamics. Therefore, stability cannot be determined by the eigenvalues of the foregoing system of equations. In this thesis, equilibrium branches are first constructed in the space of canonical parameters, and then stability is separately determined from the second variation of the total potential energy using spectral and variational methods. These methods are detailed in chapters 3 and 5.

A generic C++ code has been developed to construct the equilibrium branches of infinite-dimensional systems (see Appendix G). This code identifies branch points, traces equilibrium branches in a finite-dimensional parameter space, and records the results by writing the corresponding sequences of canonical parameters in binary files. Another C++ code has been written (see Appendix H) for liquid bridges to read the forgoing binary files and determine the stability of equilibrium states along solution branches with respect to axisymmetric and non-axisymmetric perturbations using the variational method outlined in chapter 3.

Chapter 2

Catenoid stability with a free contact line

2.1 Preface

To better understand elastic-capillary force interactions in wood fibres, it is expedient to study the stability of capillary surfaces and elastic structures in isolation and coupled together. In this thesis, axisymmetric liquid bridges under zero gravity are examined, independently of the elastic responses of the supports, as idealized geometries for air-water interfaces that arise during drying. Catenoids as important special cases of axisymmetric liquid bridges are considered in this chapter. Examining the stability criteria for liquid bridges with small but finite mean curvatures is numerically problematic due to large condition numbers. Thus, the stability of catenoids should be separately determined to avoid large rounding errors. To resolve this issue for liquid bridges in the vicinity of catenoids, perturbation-based conditioning techniques can be applied by expanding the stability criteria into power series in the mean curvature with respect to those of the catenoid as the base state.

2.2 Abstract

Contact-drop dispensing is central to many small-scale applications, such as direct-scanning probe lithography and micromachined fountain-pen techniques. Accurate and controllable dispensing required for nanometer-resolved surface patterning hinges on the stability and breakup of liquid bridges. Here, we analytically study the stability of catenoids pinned at one contact line with the other free to move on a substrate subject to axisymmetric and non-axisymmetric perturbations. We apply a variational formulation to derive the corresponding stability criteria. The maximal stability region and stability region are represented in the favourable and canonical phase diagrams, providing a complete description of catenoid equilibrium and stability. All catenoids are stable with respect to non-axisymmetric perturbations. For a fixed contact angle, there exists a critical volume below which catenoids are unstable to axisymmetric perturbations. Equilibrium solution multiplicity is discussed in detail, and we elucidate how geometrical symmetry is reflected in the maximal stability and stability regions.

2.3 Introduction

Systematic studies of liquid bridge stability and breakup began with the celebrated treatise of Plateau (1873). Early investigations were motivated by applications such as liquid-jet breakup (Rayleigh, 1879, Tomotika, 1935), crystal growth in microgravity pit-hole(García Velarde, 1988, Myshkis *et al.*, 1987), oil recovery and floatation (Smith & van de Ven, 1985), and paper wet strength (Tejado & van de Ven, 2010). Recently, interests have grown into areas such as contact-drop dispensing (Qian *et al.*, 2009) with applications to scanning-probe lithography (Liu *et al.*, 2000) and micromachined fountain-pen techniques (Deladi *et al.*, 2004). Molecular-resolution surface patterning provides new opportunities for advanced tissue engineering (Gadegaard *et al.*, 2006), DNA self-assembled nanoconstructs (Shen *et al.*, 2009), and highly sensitive protein chips (Choi *et al.*, 2009).

Static stability analysis of liquid bridges can be traced to the nineteenth century (Howe, 1887, Plateau, 1873). Howe's variational formulation extended Plateau's primitive theory for cylindrical interfaces to unbounded axisymmetric capillary surfaces subject to a constant-volume. His criteria (sufficient conditions for the weak extrema of a functional) guarantee a surface to have the minimum energy among all the neighbouring surfaces of revolution. Gillette & Dyson (1971) applied Howe's method to predict the stability limit of bounded axisymmetric liquid bridges with respect to axisymmetric perturbations. These criteria were later generalized for arbitrary interfaces with arbitrary perturbations (Myshkis *et al.*, 1987).

Catenoids are doubly connected surfaces of revolution with zero mean curvature. They are special cases of constant-mean-curvature axisymmetric surfaces, and are important to stability studies on weightless liquid bridges for two reasons: (1) Stability criteria can be obtained analytically for catenoids, which helps guide numerical algorithms for general liquid brides in the small-pressure limit, and (2) the curve corresponding to catenoidal interfaces in the volume-slenderness phase diagram defines a boundary between regions of positive and negative capillary pressure (Myshkis *et al.*, 1987). This is important for mechanical systems with dynamics that are driven by capillary pressure (*e.g.*, elasto-capillary systems (Mastrangelo & Hsu, 1993*a*)). Previous studies have considered catenoids bridging two circular discs of the same radius (Erle *et al.*, 1970) and catenoids between two parallel plates with both contact lines free to move (Strube, 1992, Zhou, 1997). However, these results are not applicable to contact-drop dispensing applications where the liquid forms a bridge with a free contact line at one end.

Recent studies on contact-drop dispensing have shown that the deposited drop size is influenced by the needle retraction speed, needle-tip size, surface characteristics, dispensing control parameters, and the interplay between the surface tension γ_{gl}, dynamic viscosity η_l, and density ρ_l of the dispensed liquid (Qian & Breuer, 2011, Qian *et al.*, 2009). Interestingly, the deposited drop volume in pressure-controlled and volume-controlled dispensing behave differently with the needle retraction speed. Faster retraction reduces the drop size to a minimum and monotonically increases the drop size in pressure-controlled and volume-controlled dispensing, respectively. Qian *et al.* (2009) applied the one-dimensional approximation of axisymmetric free-surface flows (Eggers & Dupont, 1994) to study the breakup dynamics of stretching liquid bridges with a free contact line subject to a constant-pressure. Three regimes were experimentally identified with respect to the retraction speed U_n. In the first two, U_n is much smaller than the capillary-wave speed $u_w = \sqrt{\gamma_{gl}/\rho_l R_0}$, the contact line is advancing in the first and stationary in the second regime, and the drop size scales as $U_n^{-1/2}$. A scaling argument based on the static stability limit of nearly cylindrical bridges (Slobozhanin *et al.*, 1997) was presented to ascertain this scale. However, the control-parameter effect on the stability region was neglected. In fact, Slobozhanin *et al.* (1997) presented the static stability limits of liquid bridges between equal discs with respect to constant-volume perturbations, which are significantly different than those with respect to constant-pressure perturbations (Akbari *et al.*, 2015*c*, Lowry & Steen, 1995). The third regime corresponds to fast retraction speeds ($U_n/u_w \sim O(10^{-2})$) where the dynamics dramatically change, and the drop size does not scale with U_n as a simple power law. Here, the drop size is almost two orders of magnitude smaller than the first two regimes, which Qian *et al.* (2009) attributed to a fast receding contact line with a speed approaching u_w. Qian & Breuer (2011) considered the breakup dynamics and static stability of constant-volume liquid bridges with a free contact line for a few contact angles and bridge volumes. In the static case, using the variational method of Myshkis *et al.* (1987), they incorrectly applied the constraint corresponding to the free contact line in the Sturm-Liouville problem. For constant-volume bridges, the drop size weakly depends on the retraction speed as compared to the constant-pressure case.

Dynamical effects on the breakup of free-surface flows are negligible if the time scales of disturbances induced by surface tension are much smaller than other time scales, and stability is purely determined from geometrical considerations (static stability) (Eggers, 1997). Vanishingly small retraction speeds produce dynamics where the velocity inside the bridge u scales with U_n, while the Weber number $\mathrm{We} = \rho_l R_0 U_n^2/\gamma_{gl}$ and capillary number $\mathrm{Ca} = \eta_l U_n/\gamma_{gl}$ approach zero. Thus, surface tension dominates viscous stresses and inertia, and the interfacial dynamics are quasi-static with a timescale t_d set by the retraction speed ($t_d = R_0/U_n$). The onset of instability coincides with an abrupt change in the time scale: the dynamics become much faster with u scaling with u_w, and the time scale set by the internal fluid properties ($t_d = \eta_l^3/\gamma_{gl}^2\rho_l$) as the singularity (breakup) is approached. Motivated by this observation, one may extend this scaling to small but finite retraction speeds when $\mathrm{We} \ll 1$ and $\mathrm{Ca} \ll 1$. For example, Qian & Breuer (2011) experimentally and numerically examined stretching liquid bridges for $\mathrm{Ca} \sim O(10^{-3})$ and $\mathrm{We} \sim O(10^{-6})$. They divided the breakup dynamics into quasi-static and pinch-off phases that are characterized by U_n and u_w, respectively. The point at which pinch-off begins was identified as the static stability limit of the bridge for a given volume. However, their experimental and numerical results for the contact-line speed reveals a smooth transition in the dynamics from the quasi-static to pinch-off phase. As will be shown in an accompanying paper (Akbari *et al.*, 2015*b*), static liquid bridges undergo a hard stability loss at turing points when stretched in both volume-controlled and pressure-controlled cases (Akbari *et al.*, 2015*c*) with u diverging indefinitely as the stability limit is approached. Although indefinitely large velocities are not observed in practice, because of viscous dissipation and inertia, a sufficiently large u develops before the static stability limit for the quasi-static assumption to break down, thereby invalidating the description of Qian & Breuer (2011). Moreover, Qian & Breuer (2011) did not consider the dynamics near the singularity. In a notable paper, Eggers (1993) established a universal self-similar solution for the neck shape and velocity inside the neck, close to pinch-off, independent of the boundary and initial conditions. Recently, Eggers (2012) proved that this solution is the only dynamically stable solution among an infinite sequence of similarity solutions for the pinching of free-surface flows (Brenner *et al.*, 1996) with respect to infinitesimal disturbances. However, Eggers' solution also loses stability to finite-amplitude perturbations (Brenner *et al.*, 1994), leading to instability cascades where several smaller necks grow out of the original neck (Shi *et al.*, 1994), resulting in the formation of satellite drops. Satellites are highly undesirable in lithography and printing, since they uncontrollably land on surfaces due to their non-uniform and small size (Cheng & Kricka, 2001, Eggers, 1997).

To address the smooth transition in the breakup dynamics and the near-singularity self-similar solution, we consider the pinching of liquid bridges in three phases: quasi-static, intermediate, and pinch-off. The quasi-static phase ends before the static stability limit is reached, and the bridge evolution is a sequence of equilibrium states. The pinch-off phase describes near-singularity dynamics and is characterized by Eggers' self-similar solution. However, the similarity solution is expected to have a larger stability margin than for decaying jets because the bridge boundedness in contact-drop dispensing damps the growth rate of disturbances. Solving the fully transient equations of motion is required in the intermediate phase since neither the quasi-static evolution of equilibrium states nor self-similar solutions can adequately characterize the dynamics in this phase. A static stability analysis of liquid bridges with a free contact line furnishes the critical bridge hight and the respective critical perturbations for a given bridge volume, which are necessary to understand the transition from the quasi-static to the intermediate phase. This static stability limit reasonably approximates the onset of the quasi-static to intermediate phase transition, while the critical perturbations estimate how the bridge dynamically evolves. We examine the static stability of liquid

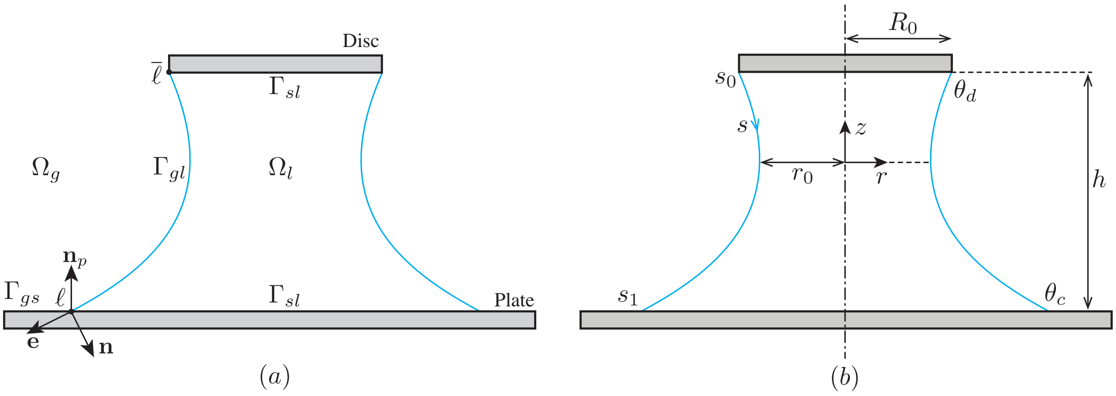


FIGURE 2.1: Catenoidal liquid bridge: (a) schematic and (b) coordinate system with meridian curve parametrization.

bridges with a free contact line in two parts. In this paper, we only focus on the catenoid as an important special case since equilibrium solution multiplicity and stability criteria can be determined analytically. The stability of general liquid bridges will be presented in an accompanying paper (Akbari *et al.*, 2015*b*).

In this paper, we analytically study the static stability of catenoids pinned at one contact line with the other free to move on a flat substrate. This furnishes a two-dimensional phase diagram in which the stability region is represented with respect to the catenoid volume and slenderness. The effect of the catenoid geometrical symmetry on the stability region boundaries is discussed. We also present a phase diagram with respect to canonical variables, which facilitates the representation of symmetry in the stability region, maximal stability region, and multiple equilibrium solutions subject to various constraints. Myshkis *et al.* (1987) described how free contact lines are generally treated in their variational method. However, the stability criteria were not presented for liquid bridges with free contact lines. Therefore, we first present an exposition of Myshkis's variational formulation (Myshkis *et al.*, 1987), and then derive the stability criteria in section 2.4 for axisymmetric liquid bridges with a free contact line. Equilibrium solution multiplicity is discussed in section 2.5.1, and the maximal stability and stability regions are determined for cylinders and catenoids in sections 2.5.2.1 and 2.5.2.2, respectively. The results are summarized in sections 2.6.

2.4 Theory

We consider a liquid of volume v bridging a circular disk with radius R_0 and a large plate. The disc and plate are separated by a distance h, as shown in Fig. 2.1. The region occupied by the liquid bridge is denoted Ω_l, and that occupied by the surrounding fluid (of a different phase) is denoted Ω_g. The bridge is pinned to the disc and free to slide horizontally on the plate. We restrict our analysis to catenoidal liquid bridges, which implies that the regions g and l have the same density and pressure, and the interface Γ_{gl} is a surface of revolution. The formulation is presented as the limit of axisymmetric weightless liquid bridges with mean curvature approaching zero. The origin of the coordinate system is placed on the plane passing through the catenoid neck such that the z-axis is the symmetry axis. The meridian curve is parametrized with respect to its arclength s such that $s = 0$ at $z = 0$. An equilibrium surface is specified by

$$\begin{cases} r = r(s) \\ z = z(s) \end{cases} \qquad s \in [s_0, s_1], \tag{2.1}$$

extremizing the potential energy

$$U = \gamma_{sl}\Gamma_{sl} + \gamma_{gl}\Gamma_{gl} + \gamma_{sg}\Gamma_{sg}, \tag{2.2}$$

where γ_{ij} is the surface tension between the phases i and j, and Γ_{ij} is the interfacial surface area. Following Myshkis *et al.* (1987), this leads to the Young-Laplace equation

$$\begin{cases} r'' = -z'(q - z'/r) \\ z'' = r'(q - z'/r) \end{cases} \qquad (' \equiv \mathrm{d}/\mathrm{d}s) \tag{2.3}$$

for axisymmetric equilibrium surfaces and

$$\gamma_{gl}\cos\theta_c = \gamma_{sg} - \gamma_{sl}, \qquad \text{and} \qquad \cos\theta_c = \mathbf{n}\cdot\mathbf{n}_p, \tag{2.4}$$

where $q = -2k_m$ and θ_c is the contact angle. Here, k_m is the mean curvature, which is zero for catenoids. Equation (2.4) shows that the contact angle is a thermodynamic property of the three-phase (g, l, and s) contact line, which is a constant for a specific substrate (plate) and the fluids occupying Ω_g and Ω_l. Note that the dihedral angle θ_d (Myshkis *et al.*, 1987) can vary independently with the bridge volume to extremize the potential energy. Introducing the following lengths, which are scaled with the neck radius,

$$\hat{r} = r/r_0, \quad \hat{z} = z/r_0, \quad \hat{s} = s/r_0, \tag{2.5}$$

the cylindrical volume $V = v/(\pi R_0^2 h)$ and slenderness $\Lambda = h/R_0$ are two dimensionless parameters with which to present the phase diagram.

Following the method of Myshkis *et al.* (1987), the interface stability is determined by the sign of the second variation. Using the Ritz method (Gelfand & Fomin, 2000), the second variation is associated with the eigenvalues of the corresponding Strum-Liouville problem. Stability studies are generally concerned with determining stability regions in the phase diagram. Stability-region boundaries, identified by $\delta^2 U = 0$, correspond to critical states, separating stable equilibrium surfaces from unstable ones. Hence, we seek the conditions where λ_0 or $\lambda_1 = 0$, resulting in

$$\begin{cases} \mathcal{L}\varphi_0 + \mu = 0 \\ \varphi_0(\hat{s}_0) = 0, \qquad \varphi_0'(\hat{s}_1) + \hat{\chi}\varphi_0(\hat{s}_1) = 0 \\ \int_{\hat{s}_0}^{\hat{s}_1} \hat{r}\varphi_0 \mathrm{d}\hat{s} = 0 \end{cases} \tag{2.6}$$

for axisymmetric perturbations and

$$\begin{cases} (\mathcal{L} - 1/\hat{r}^2)\varphi_1 = 0 \\ \varphi_1(\hat{s}_0) = 0, \qquad \varphi_1'(\hat{s}_1) + \hat{\chi}\varphi_1(\hat{s}_1) = 0 \end{cases} \tag{2.7}$$

for non-axisymmetric perturbations, where

$$\chi = \frac{k_{1\ell}\cos\theta_c - k_{p\ell}}{\sin\theta_c} \quad \text{at } \ell, \tag{2.8}$$

$$\mathcal{L} \equiv \frac{\mathrm{d}^2}{\mathrm{d}\hat{s}^2} + \frac{\hat{r}'}{\hat{r}}\frac{\mathrm{d}}{\mathrm{d}\hat{s}} + \left[\left(\hat{q} - \frac{\hat{z}'}{\hat{r}}\right)^2 + \left(\frac{\hat{z}'}{\hat{r}}\right)^2\right] \tag{2.9}$$

with $\hat{q} = qr_0$, $\hat{\chi} = r_0\chi$, and $k_{1\ell}$, $k_{p\ell}$ the first principal curvatures of the interface and plate at the contact line ℓ. The solutions of Eqs. (2.6) and (2.7) can be written

$$\varphi_0(\hat{s}) = C_1 w_1(\hat{s}) + C_2 w_2(\hat{s}) + \mu w_3(\hat{s}), \tag{2.10}$$

$$\varphi_1(\hat{s}) = C_4 w_4(\hat{s}) + C_5 w_5(\hat{s}) \tag{2.11}$$

for axisymmetric and non-axisymmetric perturbations, respectively (Myshkis *et al.*, 1987). These satisfy the following differential equations and their initial conditions

$$\mathcal{L}w_1 = 0, \qquad w_1(0) = 0, \quad w_1'(0) = 1, \tag{2.12}$$

$$\mathcal{L}w_2 = 0, \qquad w_2(0) = 1, \quad w_2'(0) = 0, \tag{2.13}$$

$$\mathcal{L}w_3 + 1 = 0, \qquad w_3(0) = -1/4, \quad w_3'(0) = 0, \tag{2.14}$$

$$(\mathcal{L} - 1/\hat{r}^2)w_4 = 0, \qquad w_4(0) = 0, \quad w_4'(0) = 1, \tag{2.15}$$

$$(\mathcal{L} - 1/\hat{r}^2)w_5 = 0, \qquad w_5(0) = 1, \quad w_5'(0) = 0, \tag{2.16}$$

where w_1, w_4 are odd and w_2, w_3, w_5 are even functions. Note that the initial conditions in Eq. (2.14) can be arbitrarily chosen because they do not affect the conditions describing the critical states of equilibrium surfaces (Eqs. (2.17) and (2.20)). The homogeneous solution of Eq. (2.14) is obtained from a linear combination of w_1 and w_2. From Eq. (2.10), the homogeneous part of w_3 makes no independent contribution to the general solution of φ_0. Therefore, the initial conditions for w_3 are chosen such that the general solution for w_3 comprises only the particular part.

The critical state of an equilibrium surface is identified by the existence of a nontrivial solution for φ_0 or φ_1. These existence conditions can be obtained from Eqs. (2.10) and (2.11) as

$$\hat{\chi}^0 = -\frac{\begin{vmatrix} w_1(\hat{s}_0) & w_2(\hat{s}_0) & w_3(\hat{s}_0) \\ w_1'(\hat{s}_1) & w_2'(\hat{s}_1) & w_3'(\hat{s}_1) \\ \int_{\hat{s}_0}^{\hat{s}_1} \hat{r}w_1 \mathrm{d}\hat{s} & \int_{\hat{s}_0}^{\hat{s}_1} \hat{r}w_2 \mathrm{d}\hat{s} & \int_{\hat{s}_0}^{\hat{s}_1} \hat{r}w_3 \mathrm{d}\hat{s} \end{vmatrix}}{\begin{vmatrix} w_1(\hat{s}_0) & w_2(\hat{s}_0) & w_3(\hat{s}_0) \\ w_1(\hat{s}_1) & w_2(\hat{s}_1) & w_3(\hat{s}_1) \\ \int_{\hat{s}_0}^{\hat{s}_1} \hat{r}w_1 \mathrm{d}\hat{s} & \int_{\hat{s}_0}^{\hat{s}_1} \hat{r}w_2 \mathrm{d}\hat{s} & \int_{\hat{s}_0}^{\hat{s}_1} \hat{r}w_3 \mathrm{d}\hat{s} \end{vmatrix}}, \tag{2.17}$$

$$\hat{\chi}^1 = -\frac{\begin{vmatrix} w_4(\hat{s}_0) & w_5(\hat{s}_0) \\ w_4'(\hat{s}_1) & w_5'(\hat{s}_1) \end{vmatrix}}{\begin{vmatrix} w_4(\hat{s}_0) & w_5(\hat{s}_0) \\ w_4(\hat{s}_1) & w_5(\hat{s}_1) \end{vmatrix}}. \tag{2.18}$$

Here, $\hat{\chi}^0$ and $\hat{\chi}^1$ are the critical $\hat{\chi}$ corresponding to axisymmetric and non-axisymmetric perturbations, respectively. Note that $\hat{\chi}$ =max$\{\hat{\chi}^0, \hat{\chi}^1\}$ identifies a critical state. For a fixed Γ_{gl}, the minimum eigenvalue of the Sturm-Liouville problem is monotonically increasing with χ. Hence, $\lambda_i > 0$ for $\chi > \chi^i$ $(i = 0, 1)$. It follows that an equilibrium surface is unstable with respect to axisymmetric (non-axisymmetric) perturbations if $\hat{\chi}^1 < \hat{\chi}^0$ ($\hat{\chi}^1 > \hat{\chi}^0$) when $\hat{\chi} <$max$\{\hat{\chi}^0, \hat{\chi}^1\}$. Moreover, fixed contact lines can be represented as the limiting case of free contact lines when $\chi \to \infty$. Therefore, for a fixed Γ_{gl}, $\lambda \to -\infty$ as $\chi \to -\infty$ and $\lambda \to \nu$ as $\chi \to \infty$; here, λ is the smallest eigenvalue of the Sturm-Liouville problem, and ν is the smallest eigenvalue of a similar problem with $\varphi_i = 0$ at ℓ. Hence,

$$\lambda \leq \nu, \tag{2.19}$$

implying that the stability region of capillary surfaces with free contact lines is a subset of the corresponding stability region for the same capillary surfaces with fixed contact lines[1]. The latter is termed the maximal stability region (MSR) (Myshkis *et al.*, 1987). The critical states are determined by

$$D^0 = \begin{vmatrix} w_1(\hat{s}_0) & w_2(\hat{s}_0) & w_3(\hat{s}_0) \\ w_1(\hat{s}_1) & w_2(\hat{s}_1) & w_3(\hat{s}_1) \\ \int_{\hat{s}_0}^{\hat{s}_1} \hat{r} w_1 \mathrm{d}\hat{s} & \int_{\hat{s}_0}^{\hat{s}_1} \hat{r} w_2 \mathrm{d}\hat{s} & \int_{\hat{s}_0}^{\hat{s}_1} \hat{r} w_3 \mathrm{d}\hat{s} \end{vmatrix}, \tag{2.20}$$

$$D^1 = \begin{vmatrix} w_4(\hat{s}_0) & w_5(\hat{s}_0) \\ w_4(\hat{s}_1) & w_5(\hat{s}_1) \end{vmatrix}. \tag{2.21}$$

For a fixed $\hat{s}_0$, the first $\hat{s}_1$ along the meridian curve at which $D^0 = 0$ ($D^1 = 0$) corresponds to a critical state of the MSR with respect to axisymmetric (non-axisymmetric) perturbations. Note that the MSR coincides with the stability region for capillary surfaces with only pinned contact lines. Moreover, determining the MSR for capillary surfaces with free contact lines prior to testing the stability criteria given by Eqs. (2.17) and (2.18) is necessary, since $\hat{\chi} = \hat{\chi}^0$ and $\hat{\chi} = \hat{\chi}^1$ generally have more than one solution. Therefore, $\hat{\chi} >$max$\{\hat{\chi}^0, \hat{\chi}^1\}$ indicates stability only for surfaces belonging to the MSR. All equilibrium surfaces outside the MSR are unstable.

[1]What 'the same' means here depends on how a capillary surface is specified. For example, as will be discussed in section 2.5.1, a catenoid, such as the one shown in Fig. 2.1, is uniquely specified by $\hat{s}_0$ and $\hat{s}_1$. Hence, catenoids with free contact lines are being compared to those with the same $\hat{s}_0$ and $\hat{s}_1$, but fixed at $\hat{s}_1$.

2.5 Results and discussion

2.5.1 Equilibrium solution

Solving Eq. (2.3) for $q = 0$ furnishes the equilibrium meridian curve

$$\begin{cases} \hat{r}(\hat{s}) = \sqrt{\hat{s}^2 + 1} \\ \hat{z}(\hat{s}) = -\ln(\hat{s} + \sqrt{\hat{s}^2 + 1}) \end{cases} \tag{2.22}$$

with

$$\Lambda = \frac{1}{\sqrt{\hat{s}_0^2 + 1}} \ln\left(\frac{\hat{s}_1 + \sqrt{\hat{s}_1^2 + 1}}{\hat{s}_0 + \sqrt{\hat{s}_0^2 + 1}}\right), \tag{2.23}$$

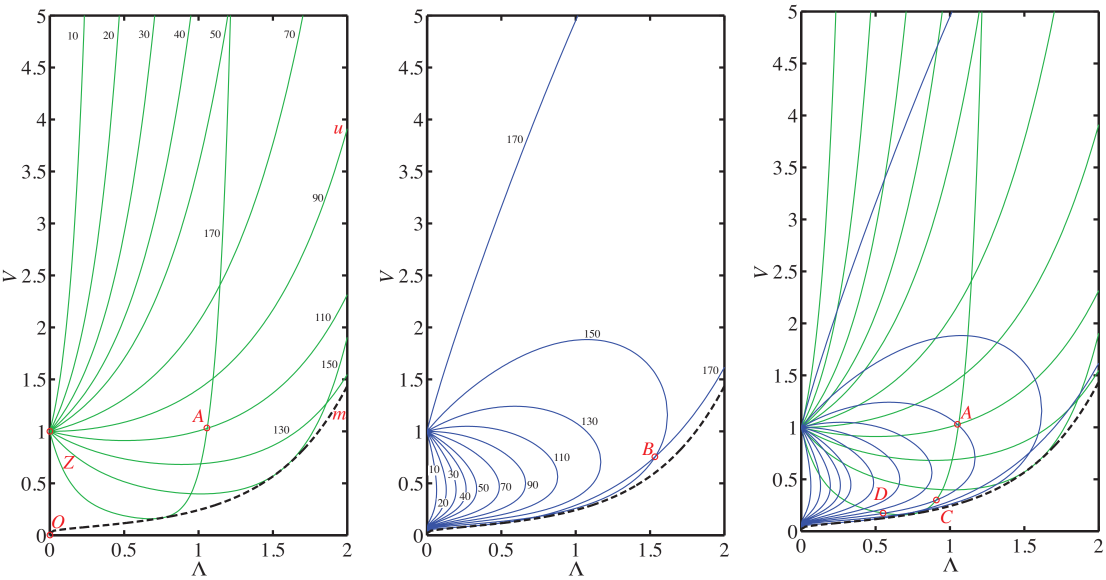
$$V = \frac{\hat{s}_1\sqrt{\hat{s}_1^2 + 1} - \hat{s}_0\sqrt{\hat{s}_0^2 + 1} + \Lambda\sqrt{\hat{s}_0^2 + 1}}{2\Lambda(\hat{s}_0^2 + 1)^{3/2}}, \tag{2.24}$$

$$\theta_d = \tan^{-1}(1/\hat{s}_0), \tag{2.25}$$

$$\theta_c = \tan^{-1}(-1/\hat{s}_1). \tag{2.26}$$

Equations (2.23)-(2.26) furnish four constraints on Λ, V, θ_c, θ_d, $\hat{s}_0$, and $\hat{s}_1$, leaving two degrees of freedom. Fixing two variables, the other four and the catenoid geometry, as shown in Fig. 2.1, are completely specified. Therefore, among these six, one can select two variables to represent stability and equilibrium data as two-dimensional phase diagrams. Furthermore, Eqs. (2.25) and (2.26) provide a one-to-one correspondence between $(\hat{s}_0, \hat{s}_1)$ and (θ_c, θ_d). Thus, they can be interchanged without affecting phase diagram characteristics, albeit (θ_c, θ_d) are preferred to $(\hat{s}_0, \hat{s}_1)$ as they vary in a finite range. In this paper, the catenoid equilibrium solution and stability region are presented with respect to Λ, V, θ_c, and θ_d subject to various constraints.

The cylindrical volume and slenderness are the two favourable quantities with which stability regions have been represented in the literature (Bayramli & van de Ven, 1987, Lowry & Steen, 1995, Martínez & Perales, 1986, Myshkis *et al.*, 1987, Slobozhanin & Perales, 1993, 1996) because they can be readily measured experimentally. We refer to (Λ, V) as 'favourable parameters' and the respective



(a) (b) (c)

FIGURE 2.2: Equilibrium solution and existence region (above dashed line) with respect to the favourable parameters: (a) constant-dihedral angle isocontours (numbers indicate θ_d in degrees); (b) constant-contact angle isocontours (numbers indicate θ_c in degrees), and (c) constant-dihedral and contact angle isocontours (the same as (a) and (b)).

phase diagram as the 'favourable phase diagram'. Moreover, Eqs. (2.23)-(2.26) are single-valued functions of $\hat{s}_0$ and $\hat{s}_1$. Consequently, the left-hand-side parameters characterizing a catenoid described in Fig. 2.1 are uniquely specified with respect to $\hat{s}_0$ and $\hat{s}_1$. Thus, $(\hat{s}_0, \hat{s}_1)$ are more convenient for representing the MSR and stability region. We refer to $(\hat{s}_0, \hat{s}_1)$ or (θ_c, θ_d), as 'canonical parameters' and the respective phase diagram as the 'canonical phase diagram'.

To properly represent equilibrium solutions with respect to the favourable and canonical parameters, one needs to identify the existence-region boundary in the corresponding space. This is straightforward for the canonical parameters because $\theta_d \in (\pi - \theta_c, \pi]$ for $\theta_c \in [0, \pi]$. However, determining the existence-region boundary with respect to the favourable parameters is non-trivial. One may naturally suppose that $V(\hat{s}_0, \hat{s}_1, \Lambda)$ has a minimum for a given Λ. Therefore, the cylindrical volume given by Eq. (2.24) is to be minimized subject to a constant slenderness Λ, leading to

$$\left(\frac{\partial V}{\partial \hat{s}_0}\right)_{\hat{s}_1,\Lambda} + \vartheta \left(\frac{\partial \Lambda}{\partial \hat{s}_0}\right) = 0, \tag{2.27}$$

$$\left(\frac{\partial V}{\partial \hat{s}_1}\right)_{\hat{s}_0,\Lambda} + \vartheta \left(\frac{\partial \Lambda}{\partial \hat{s}_1}\right) = 0, \tag{2.28}$$

$$\Lambda(\hat{s}_0, \hat{s}_1) = \text{const.}, \tag{2.29}$$

where ϑ is a Lagrange multiplier. Moreover, Eqs. (2.27)-(2.29) define the existence-region boundary in the Λ-V space, also furnishing a lower bound on V.

Figure 2.2 shows equilibrium isocontours and existence region with respect to the favourable parameters. The Om curve (dashed line) is the existence-region boundary that corresponds to (Λ, V) satisfying Eqs. (2.27)-(2.29). No catenoid can be found with volume and slenderness below this curve. The existence-region boundary in the favourable diagram is of practical importance. The contact angle is constant only on ideal surfaces. It varies between the advancing and receding contact angles on real surfaces due to contact-angle hysteresis. It also depends on the contact-line position for inhomogeneous surfaces. Thus, for a given cylindrical volume (slenderness), Om provides the maximum (minimum) slenderness (volume) for which catenoidal bridges can exist, irrespective of the contact angle. This is particularly important for capillary driven mechanical systems because catenoids separate liquid bridges with positive mean curvature (capillary pressure) from those with negative mean curvature in stability diagrams (Akbari *et al.*, 2015*c*).

Iso-θ_d curves are plotted in Fig. 2.2(a) facilitating solution multiplicity representation with respect to Λ, V, and θ_d. These can be viewed as the level curves of a multivalued function $\theta_d = \theta_d(\Lambda, V)$. For $\theta_d > \pi/2$, isocontours cross others corresponding to smaller θ_d. For example, the level curves with $\theta_d = 110°$ and $\theta_d = 170°$ intersect at A, implying that two equilibrium solutions exist for the corresponding (Λ, V). For a fixed Λ, there is only one solution corresponding to a given θ_d. For a fixed V, one (two) solution(s) can be found corresponding to a given $\theta_d \le \pi/2$ ($\theta_d > \pi/2$). Furthermore, all the isocontours are tangent to Om at one point when $\theta_d > \pi/2$. The isocontour with $\theta_d = \pi/2$ is a special case that asymptotes to Om. It can be proved that

$$\begin{aligned}\lim_{\Lambda\to\infty} V(\hat{s}_0, \hat{s}_1)|_{(\hat{s}_0,\hat{s}_1)\in Zu} &= \lim_{\Lambda\to\infty} V(\hat{s}_0, \hat{s}_1)|_{(\hat{s}_0,\hat{s}_1)\in Om} \\ &= \lim_{\Lambda\to\infty} \frac{\sinh^2\Lambda}{2\Lambda} = \infty,\end{aligned} \tag{2.30}$$

where Zu is the isocontour with $\theta_d = \pi/2$. Isocontours for $\theta_d < \pi/2$ have no contact point with Om. Figure 2.2(b) can be similarly interpreted. Here, the solution multiplicity is represented with respect to Λ, V, and θ_c, and isocontours are the level curves of a multivalued function $\theta_c = \theta_c(\Lambda, V)$. All the isocontours are half-loops starting at Z and ending at O. Iso-θ_c curves for large contact angles cross others corresponding to smaller θ_c. For example, isocontours with $\theta_c = 150°$ and $\theta_c = 170°$ intersect at B, implying that two equilibrium solutions exist for the corresponding (Λ, V). For a given θ_c, there is a slenderness above which there are no catenoids. Two equilibrium solutions exist for smaller slendernesses. For a fixed V, one (two) solution(s) can be found corresponding to a given $\theta_c \leq \pi/2$ ($\theta_c > \pi/2$). Moreover, all the isocontours are tangent to the existence-region boundary at O. When $\theta_c \leq \pi/2$, isocontours have no other contact point with Om, whereas those with $\theta_c > \pi/2$ are tangent to the existence-region boundary at another point along Om. Iso-θ_d and iso-θ_c curves are plotted together in Fig. 2.2(c). Here, the solution multiplicity is determined with respect to θ_d and θ_c. As previously mentioned, Λ and V are uniquely specified for a given θ_d and θ_c. Therefore, one should expect iso-θ_d and iso-θ_c curves to intersect at only one point (besides Z). However, it is clear from Fig. 2.2(c) that the isocontours do not meet this requirement. For example, the isocontours with $\theta_d = 170°$ and $\theta_c = 130°$ intersect at A, C, and D corresponding to three different (Λ, V). This can be explained by differentiating between 'crossing' and 'intersection' points. This requires a proper representation of equilibrium solutions in relation to Eqs. (2.23)-(2.26). A direct approach involves representing all equilibrium solutions as points on a two-dimensional surface embedded in a four-dimensional manifold. In a simpler approach, one may regard an equilibrium solution as two corresponding points on the multivalued surfaces $\theta_d = \theta_d(\Lambda, V)$ and $\theta_c = \theta_c(\Lambda, V)$ in the respective space. Thus, there corresponds a curve on $\theta_c = \theta_c(\Lambda, V)$ to any curve defined on $\theta_d = \theta_d(\Lambda, V)$. In particular, a level curve of the first surface corresponds to a curve (not a level curve) on the second. Within this framework, a crossing is defined as a point at which the level curves of two multivalued surfaces, when projected onto the plane of respective independent variables, cross each other. Hence, a crossing may or may not represent the equilibrium solution corresponding to θ_d and θ_c of the respective level curves since the intersection of projections does not imply a correspondence between θ_d and θ_c. An intersection is defined as a crossing point that corresponds to θ_d and θ_c of the same equilibrium solution. Returning to the previous example, D is the only intersection point of the isocontours with $\theta_d = 170°$ and $\theta_c = 130°$, whereas A and C are crossing points. Furthermore, at crossing points such as A, the intersection of two iso-θ_d and two iso-θ_c coincide. These correspond to two equilibrium solutions for (Λ, V) at A. Consequently, an iso-θ_d curve crosses several iso-θ_c curves, but intersects only one. This is also the

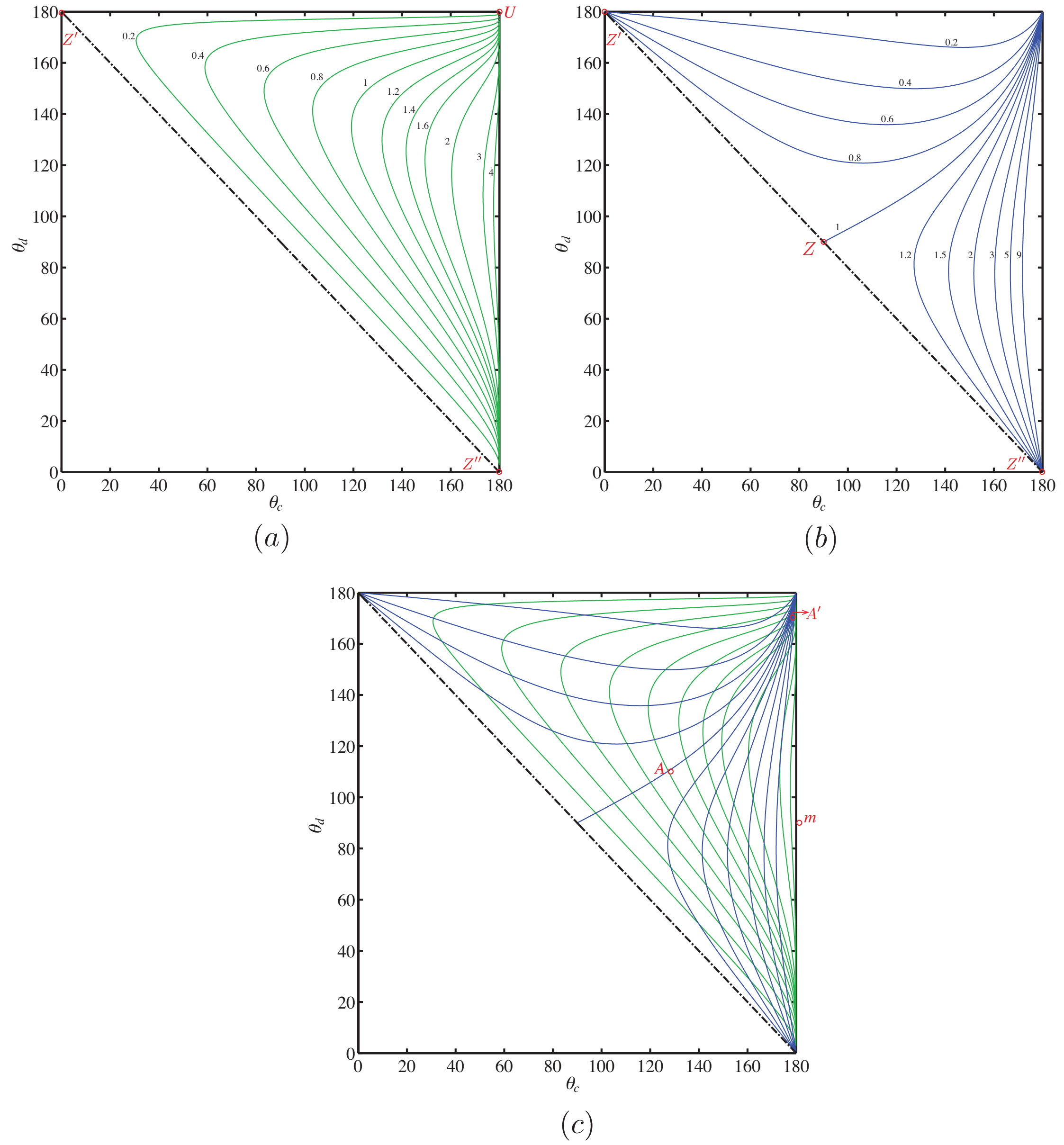


FIGURE 2.3: Equilibrium solution and existence region (above dash-dotted line) with respect to the canonical parameters:(a) constant-slenderness isocontours (numbers indicate Λ); (b) constant-cylindrical volume isocontours (numbers indicate V), and (c) constant-slenderness and cylindrical volume isocontours (the same as (a) and (b)).

intersection of another iso-θ_d and an iso-θ_c curve.

Figure 2.3 shows equilibrium isocontours and the existence region with respect to the canonical parameters. As previously discussed, equilibrium solutions for catenoids cannot be conveniently represented with respect to the favourable parameters because isocontour intersections are not always associated with equilibrium solutions. Moreover, two solutions may correspond to the same point (Λ, V)

in the existence region. In contrast, the canonical parameters furnish a one-to-one correspondence between points in the existence region and equilibrium solutions. Here, the existence region is the upper triangle indicated by $Z'Z''U$. Note that the existence-region boundary $Z'Z''$ does not correspond to Om in Fig. 2.2. Iso-Λ curves are plotted in Fig. 2.3(a). Isocontours are the level curves of the single-valued function $\Lambda = \Lambda(\theta_c, \theta_d)$ given by Eq. (2.23). All the isocontours are half-loops starting at Z'' and ending at U. The entire $Z'U$ boundary corresponds to the point O in Fig. 2.2. The points Z', Z'', and U correspond to OZ, Zgm, and $OZgm$, respectively. Note that gm represents the range in Λ confined between the V-axis and the existence-region boundary Om as $V \to \infty$. Iso-V curves are plotted in Fig. 2.3(b). These are the level curves of the single-valued function $V = V(\theta_c, \theta_d)$ given by Eq. (2.24). They are, except $V = 1$, half-loops starting at U and ending at Z' and Z'' for $V < 1$ and $V > 1$, respectively. The isocontour with $V = 1$ starts at U and ends at Z. Note that Z, Z', and Z'' cannot be distinguished in Fig. 2.2, and they all correspond to the point Z. Iso-Λ and iso-V curves are plotted together in Fig. 2.3(c). All the conclusions drawn from Fig. 2.2 regarding the solution multiplicity hold for Fig. 2.3. Here, unlike the favourable parameters, every isocontour intersection uniquely represents an equilibrium solution. Interesting to note are the two equilibrium solutions corresponding to the point A in Fig. 2.2, which are denoted by A and A', and are distinctly represented with respect to the canonical parameters. Furthermore, m represents the asymptotic approach of the existence-region boundary to the isocontour with $\theta_d = \pi/2$ (Fig. 2.2(a)) as $\Lambda, V \to \infty$.

Orr *et al.* (1975) studied liquid bridges between a sphere and a plate with two free contact lines and presented the equilibrium solutions of catenoids in a similar diagram to the canonical diagram (Fig. 2.3) by plotting constant-filling angle isocontours. Here, the filling angle is implicitly related to the experimentally more convenient parameters Λ, and V. Moreover, they did not present the equilibrium solutions and respective existence region in the slenderness versus volume diagram.

2.5.2 Stability

Several factors affect the equilibrium state and stability of capillary surfaces, including fluid inertia, external fields (*e.g.*, gravitational and centrifugal forces), and

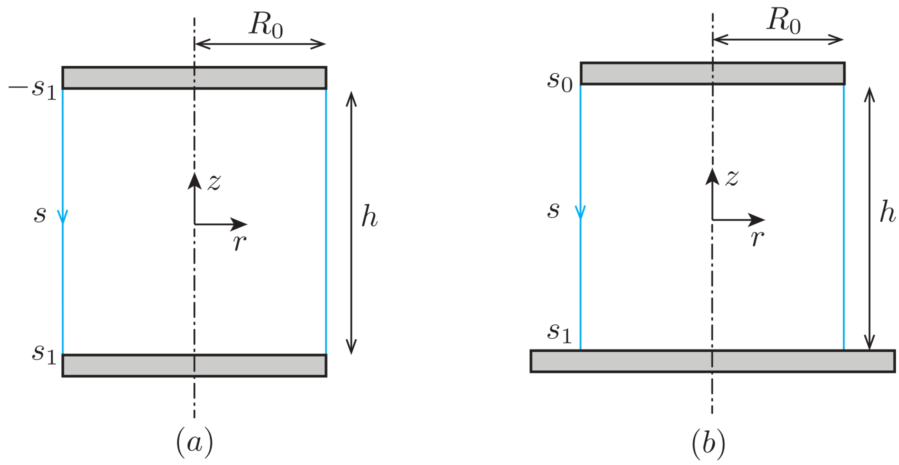


FIGURE 2.4: Cylindrical liquid bridge: (a) equilibrium surface is pinned to both discs; (b) equilibrium surface is pinned at the upper disc and free to move on the lower plate.

boundary conditions at contact lines. The latter differentiates contact-drop dispensing applications from classical liquid bridge problems where the equilibrium surface is pinned to two coaxial discs. The contact-line condition can be easily accounted for in the equilibrium solution by integration constants of the integral curve obtained from Eq. (2.3). The influence on stability, however, is not straightforward. It affects the eigenvalues of the Sturm-Liouville problem through the boundary condition at ℓ. This plays a far more significant role in the stability of capillary surfaces. For example, the notion of wavenumber introduced for classifying equilibrium solution branches and characterizing the bifurcation of liquid bridges (Lowry & Steen, 1995, Slobozhanin *et al.*, 2002) is directly related to the conditions at the contact lines. Before proceeding to catenoids, we elucidate the contact-line condition effect on the stability limit of cylindrical liquid bridges.

2.5.2.1 Cylinder

Consider the cylindrical liquid bridge between two plates shown in Fig. 2.4. Plateau (1873) theoretically obtained the stability region $\Lambda < 2\pi$ for cylindrical liquid bridges pinned at two equal coaxial discs (Fig. 2.4(a)). In this section, the corresponding stability limit is obtained for cylindrical liquid bridges that are pinned to a disc and free to move on a plate (Fig. 2.4(b)). All the lengths are scaled with q as

$$\rho = |q|r, \quad \xi = qz, \quad \tau = |q|s \tag{2.31}$$

with

$$L \equiv \frac{\mathcal{L}}{\hat{q}^2} = \frac{\mathrm{d}^2}{\mathrm{d}\tau^2} + \frac{\rho'}{\rho}\frac{\mathrm{d}}{\mathrm{d}\tau} + \left[\left(1 - \frac{\xi'}{\rho}\right)^2 + \left(\frac{\xi'}{\rho}\right)^2\right] \tag{2.32}$$

used in Eqs. (2.12)-(2.16) instead of $\mathcal{L}$. The solutions of Eqs. (2.12)-(2.16) are

$$w_1(\tau) = \sin\tau, \quad w_2(\tau) = \cos\tau, \quad w_3(\tau) = -1 \tag{2.33}$$

for axisymmetric and

$$w_4(\tau) = \tau, \quad w_5(\tau) = 1 \tag{2.34}$$

for non-axisymmetric perturbations. These furnish

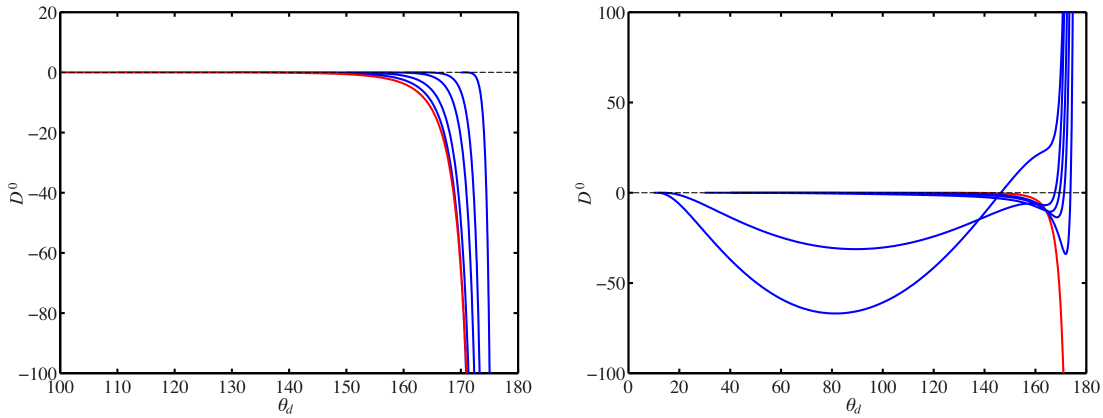
$$D^0(\Delta\tau) = -\Delta\tau\sin\Delta\tau + 2(1-\cos\Delta\tau), \tag{2.35}$$

$$D^1(\Delta\tau) = \Delta\tau, \tag{2.36}$$

$$\tilde{\chi}^0(\Delta\tau) = \frac{\Delta\tau\cos\Delta\tau - \sin\Delta\tau}{-\Delta\tau\sin\Delta\tau + 2(1-\cos\Delta\tau)}, \tag{2.37}$$

$$\tilde{\chi}^1(\Delta\tau) = -\frac{1}{\Delta\tau}, \tag{2.38}$$

where $\Delta\tau = \tau_1 - \tau_0$ and $\tilde{\chi} = \chi/|q|$. The MSR boundary is determined by $D^0(\Delta\tau) = 0$ and $D^1(\Delta\tau) = 0$ with respect to axisymmetric and non-axisymmetric perturbations, respectively. The first root of Eq. (2.35) occurs where $\Delta\tau^0_{MSR} = 2\pi$, whereas Eq. (2.36) has no non-trivial root. This implies that all cylindrical bridges with $\Delta\tau > 2\pi$ are unstable to axisymmetric perturbations, irrespective of the contact-line condition at s_1. Note that the MSR of Fig. 2.4(b) is equivalent to the stability region of Fig. 2.4(a), and, therefore, the foregoing condition coincides with Plateau's stability criterion. The first non-trivial root of $\tilde{\chi}^0 = \tilde{\chi}$ and $\tilde{\chi}^1 = \tilde{\chi}$ inside the respective MSR identifies the stability region with respect to axisymmetric and non-axisymmetric perturbations, respectively. The former gives $\Delta\tau^0_{cr} \simeq 4.4934$, whereas the latter has no non-trivial root. For cylindrical liquid bridges, Λ and $\Delta\tau$ are equal; thus, the MSR and stability region can be summarized as $\Lambda < 2\pi$ and $\Lambda < 4.4934$ where axisymmetric perturbations are the most dangerous. Note that the stability region of cylindrical liquid bridges with two free contact lines is $\Lambda < \pi$ (Langbein, 2002), which clearly indicates the destabilizing effect of free contact lines.



(*a*) (*b*)

FIGURE 2.5: The effect of contact angle θ_c on D^0 for (a) $\theta_c = 10, 20, 30, 50, 70°$ (blue, right to left) and $\theta_c = 90°$ (red) and (b) $\theta_c = 130, 140, 150, 168, 170°$ (blue, right to left) and $\theta_c = 90°$ (red).

2.5.2.2 Catenoid

Here, we apply the same procedure as for cylindrical liquid bridges. Solving Eqs. (2.12)-(2.16) using the integral curve of Eq. (2.22) gives

$$w_1(\hat{s}) = \frac{\hat{s}}{\sqrt{\hat{s}^2+1}}, \tag{2.39}$$

$$w_2(\hat{s}) = 1 - \frac{\hat{s}}{\sqrt{\hat{s}^2+1}} \ln(\hat{s} + \sqrt{\hat{s}^2+1}), \tag{2.40}$$

$$w_3(\hat{s}) = -\frac{\hat{s}^2+1}{4}, \tag{2.41}$$

$$w_4(\hat{s}) = \frac{\hat{s}}{2} + \frac{\hat{s}}{2\sqrt{\hat{s}^2+1}} \ln(\hat{s} + \sqrt{\hat{s}^2+1}), \tag{2.42}$$

$$w_5(\hat{s}) = \frac{1}{\sqrt{\hat{s}^2+1}}. \tag{2.43}$$

Substituting Eqs. (2.39)-(2.43) into Eqs. (2.17), (2.18), (2.20), and (2.21) furnishes D^0, D^1, $\hat{\chi}^0$, and $\hat{\chi}^1$ as functions of $\hat{s}_0$ and $\hat{s}_1$. These, unlike for cylinders, cannot generally be represented as functions of only $\Delta\hat{s}$, implying that the stability of catenoids demands two independent parameters to be completely specified. This is consistent with the equilibrium solution discussed in section 2.5.1.

Erle *et al.* (1970) showed that catenoids pinned to two equal coaxial discs are unstable to axisymmetric perturbations when $\Delta\hat{s}/2 > 4.6395$. We will determine the stability region for catenoids with a free contact line as shown in Fig. 2.1 and

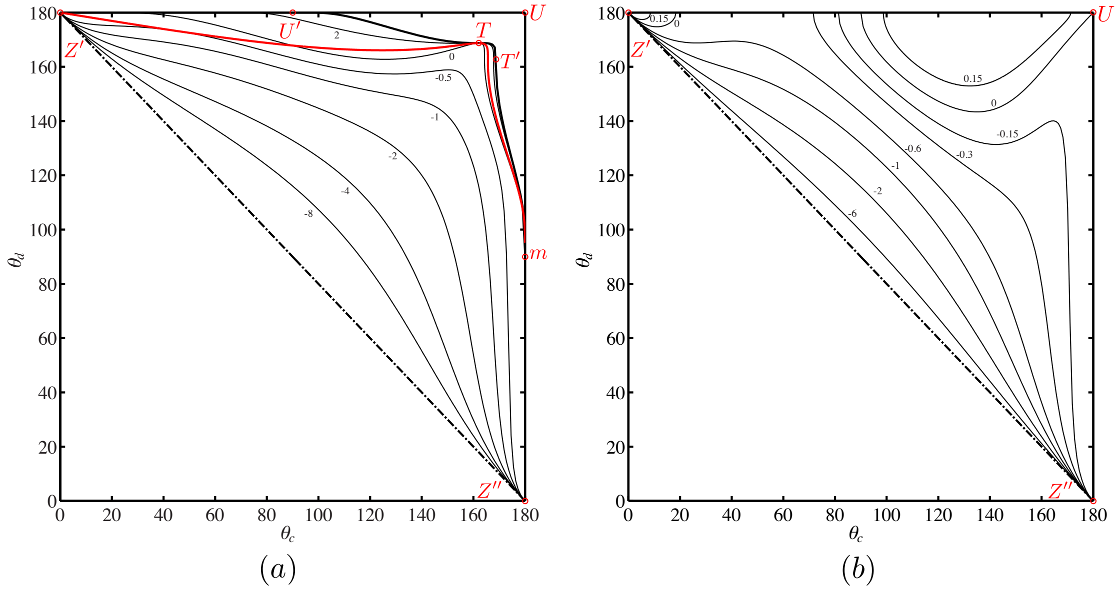


FIGURE 2.6: Canonical phase diagram: (a) axisymmetric perturbations: constant-$\hat{\chi}^0$ isocontours (thin solid black lines, numbers indicate $\hat{\chi}^0$), vanishingly small catenoids as $\hat{\chi}^0 \to -\infty$ (dash-dotted line), the MSR boundary as $\hat{\chi}^0 \to \infty$ (thick solid black line), and the stability region boundary (thick solid red line); (b) non-axisymmetric perturbations: constant-$\hat{\chi}^1$ isocontours (thin solid black lines, numbers indicate $\hat{\chi}^1$), vanishingly small catenoids as $\hat{\chi}^1 \to -\infty$ (dash-dotted line).

demonstrate that they lose stability to axisymmetric perturbations. This can be accomplished by showing that $D^1 = 0$ and $\hat{\chi}^1 = \hat{\chi}$ have no non-trivial root (proved in Appendix A). Accordingly, the MSR and stability region with respect to non-axisymmetric perturbations coincide with the existence region. Moreover, one can prove that $D^0 = 0$ has a non-trivial root only when $\theta_c > \pi/2$ (Appendix A). This is clearly illustrated in Fig. 2.5. Here, $D^0(\theta_c, \theta_d)$ is plotted in Fig. 2.5(a) for $\theta_c \leq \pi/2$ and $\theta_d \in (\pi - \theta_c, \pi]$. For a given θ_c, $D^0 \to 0^-$ as $\theta_d \to \pi - \theta_c$ and $D^0 \to -\infty$ as $\theta_d \to \pi$. Thus, no $\hat{s}_0$ can be found along the integral curve where D^0 vanishes, and the MSR spans the entire existence region. In contrast, for a given $\theta_c > \pi/2$, there exists a θ_d (or $\hat{s}_0$) at which D^0 vanishes, as indicated in Fig. 2.5(b). Here, $D^0 \to 0^-$ as $\theta_d \to \pi - \theta_c$ and $D^0 \to \infty$ as $\theta_d \to \pi$. Note that $\theta_c \simeq 168.75°$ is a special case because D^0 and $\partial D^0/\partial\theta_d$ vanish simultaneously at $\theta_d \simeq 162.07°$.

Figure 2.6(a) shows the canonical phase diagram representing the MSR and stability region for axisymmetric perturbations. The regions confined by $Z'Z''mT'TU'Z'$ and $Z'Z''mTZ'$ represent the MSR and stability region, respectively. The MSR boundary $mT'TU'$ and stability-region boundary mTZ' are determined, respectively, by $D^0(\hat{s}_0, \hat{s}_1) = 0$ and $\hat{\chi}^0(\hat{s}_0, \hat{s}_1) = \hat{\chi}(\hat{s}_1)$. The meridian curve for catenoids corresponding to points on the MSR boundary satisfies $\nu = 0$, and that corresponding to points on the stability region boundary satisfies $\lambda = 0$. All the catenoids corresponding to points outside the MSR $mUU'TT'm$ are, regardless of the contact-line condition at $\hat{s}_1$, unstable to axisymmetric perturbations. D^0 and $\partial D^0/\partial\theta_c$ vanish simultaneously at T where $(\theta_c, \theta_d) \simeq (162.07°, 168.75°)$. Similarly, D^0 and $\partial D^0/\partial\theta_d$ vanish simultaneously at T' where $(\theta_c, \theta_d) \simeq (168.75°, 162.07°)$. Note that the MSR here is equivalent to the stability region of catenoids pinned to two unequal coaxial discs. Hence, Fig. 2.6(a) also allows a comparison between two stability problems: (1) Catenoids pinned to a disc and free to move on a plate (Fig. 2.1), and (2) catenoids pinned to two unequal coaxial discs with exactly the same $\hat{s}_0$ and $\hat{s}_1$. The region confined by $mT'TU'Z'Tm$ represents the catenoids that are unstable to axisymmetric perturbations in the first problem, but stable in the second. Iso-$\hat{\chi}^0$ curves are thin solid black lines approaching $Z'Z''$ ($mT'TU'$) as $\hat{\chi}^0 \to -\infty$ ($\hat{\chi}^0 \to \infty$). Therefore, catenoids corresponding to points in the close vicinity of $Z'Z''$ ($mT'TU'$) are highly stable (unstable) since $\hat{\chi}_0 \ll \hat{\chi}$ ($\hat{\chi}_0 \gg \hat{\chi}$). Figure 2.6(b) shows the canonical phase diagram representing the MSR and stability region for non-axisymmetric perturbations. Here, $D^1(\hat{s}_0, \hat{s}_1) = 0$ and $\hat{\chi}^1(\hat{s}_0, \hat{s}_1) = \hat{\chi}(\hat{s}_1)$ have no non-trivial solution. Thus, the MSR and stability region coincide with the existence region, implying that catenoids are always stable with respect to non-axisymmetric perturbations. Iso-$\hat{\chi}^1$ curves are plotted as thin solid black lines approaching $Z'Z''$ as $\hat{\chi}^1 \to -\infty$. Note that $\hat{\chi}^1$ does not approach infinity for isocontours near the existence-region boundary $Z'UZ''$.

Figure 2.6 also illustrates how the catenoid geometrical symmetry is reflected in its phase diagram. Catenoids that are pinned to two equal coaxial discs (Erle *et al.*, 1970) have equatorial symmetry, resulting in a one-dimensional phase digram in $\Delta\hat{s}$. Even though catenoids bridging two unequal coaxial discs generally have no equatorial symmetry, and they require a two-dimensional phase diagram, a symmetric stability region can be constructed by choosing a proper set of parameters. For instance, one may choose the ratio of the lower and upper disc diameters K to represent the phase diagram (the second parameter can arbitrarily be selected). These catenoids are reflectively symmetric with respect to K. Clearly, inverting this ratio has no effect on the stability limit. Hence, the stability-region boundary must be invariant with respect to the transformation $K = 1/\bar{K}$. Alternatively, one can choose the dihedral angle that the catenoid forms with the upper disc θ_d

and the lower one θ_c. The foregoing transformation can equivalently be written

$$\begin{cases} \theta_c = \bar{\theta}_d \\ \theta_d = \bar{\theta}_c \end{cases}, \tag{2.44}$$

which is why the MSR boundary in Fig. 2.6(a) is symmetric with respect to the phase diagram minor diagonal described by $\theta_d = \theta_c$. This is formally proved in Appendix B. Note that the stability-region boundary has no such symmetry since the contact line condition at ℓ (see Fig. 2.1) completely breaks the equatorial and reflective symmetries.

Figure 2.7(a) shows the favourable phase diagram representing the MSR and stability region for axisymmetric perturbations. The regions above OTm and $Z'YTm'$ are the existence and stability regions, respectively. Unlike the canonical phase diagram, the stability-region boundary $Z'YTm'$ does not separate points in the existence region corresponding to stable and unstable catenoids. As discussed in section 2.5.1, two equilibrium solutions correspond to each point in the existence region (except its boundary OTm) with respect to the favourable parameters. All the points on the existence-region boundary correspond to only one equilibrium solution. A point in the stability region may correspond to either two stable catenoids or one stable and one unstable catenoid. This also holds for the region confined between Tm and Tm'. All the points in the region confined between OT and $Z'YT$ correspond to two unstable catenoids. The stability-region boundary is a decreasing curve from $(\Lambda, V) \simeq (0, 0.2617)$ at Z' to $(\Lambda, V) \simeq (0.7901, 0.1787)$ at Y and is an increasing curve from Y to m', where it asymptotes to the existence-region boundary OTm; it is also tangent to OTm at T where $(\Lambda, V) \simeq (0.8124, 0.1827)$. These can be represented more conveniently in the canonical phase diagram, as illustrated in Fig. 2.7(c). Here, the existence-region boundary in the favourable phase diagram (Om) and stability-region boundary are represented, respectively, by UTm and $Z'Tm$. The curve UTm is the locus of points at which an iso-θ_d curve is tangent to an iso-θ_c curve (see Fig. 2.7(d)). The curve corresponding to the existence-region boundary with respect to the favourable parameters (UTm) intersects the stability region boundary at T where $(\theta_c, \theta_d) \simeq (162.07°, 168.75°)$. This is the point at which the slope of the MSR boundary is zero, as discussed for Fig. 2.6(a). Figure 2.7(c) clearly demonstrates that all the points on the segments UT and Tm correspond to unstable and stable catenoids, respectively. Figures 2.7(b) and 2.7(d) show isocontours

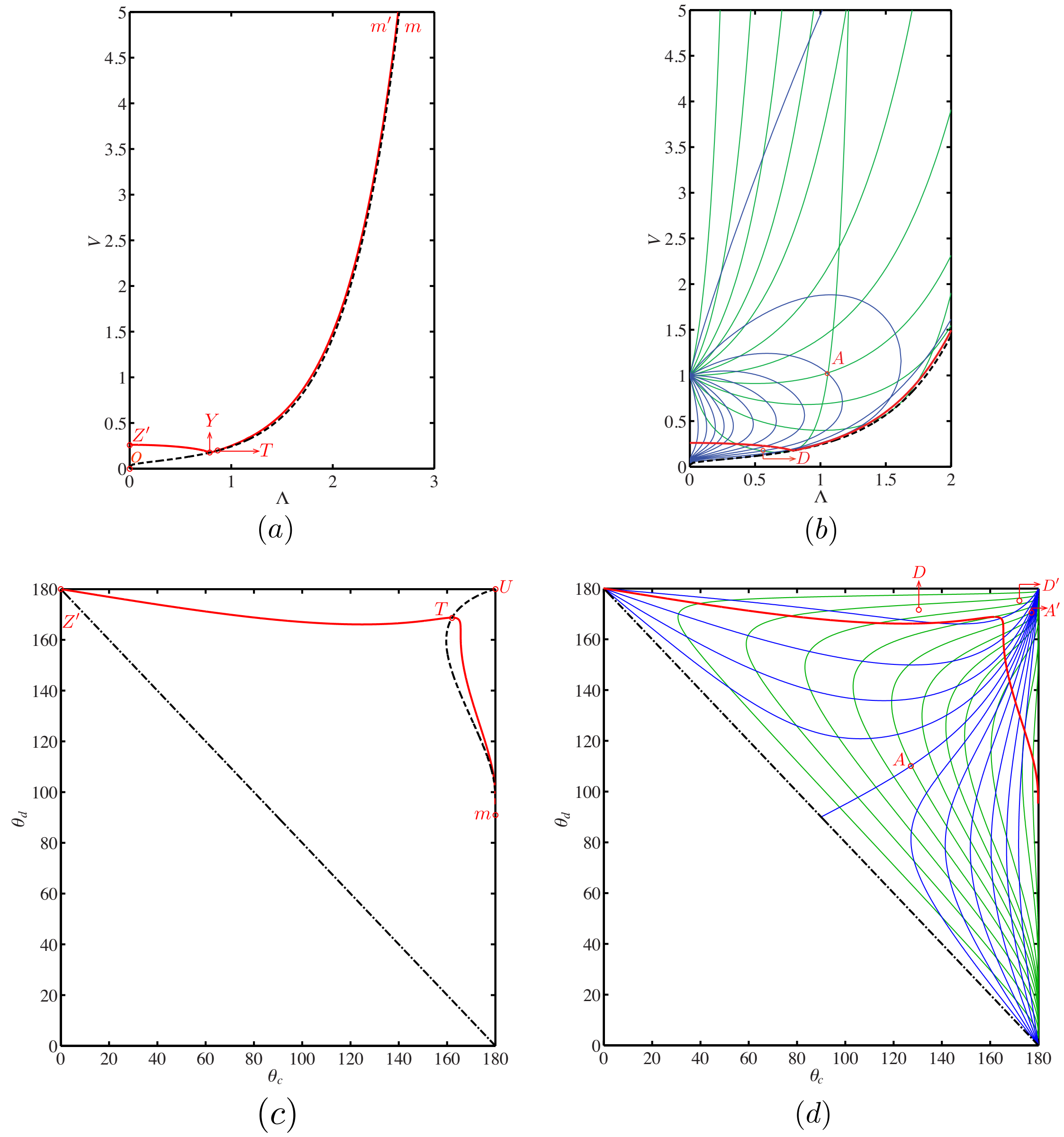


Figure 2.7: Existence region in the phase diagram, existence-region boundary (dashed black for favourable and dash-dotted black for canonical), stability-region boundary (solid red): (a) existence region in the favourable phase diagram; (b) isocontours (the same as Fig. 2.2) in the favourable phase diagram; (c) existence region in the canonical phase diagram; (d) isocontours (the same as Fig. 2.3) in the canonical phase diagram.

in the favourable and canonical phase diagrams, respectively. Selecting two variables among Λ, V, θ_c, and θ_d, these figures completely describe the equilibrium solution and stability of the corresponding catenoids. Consider the point A at $(\Lambda, V) \simeq (1.0518, 1.0264)$ in Fig. 2.7(b), for example. It lies at the intersection of $\theta_c = 130°$ and $\theta_d = 110°$. This can be located in the canonical phase diagram, as shown in Fig. 2.7(d). Here, A is inside the stability region, indicating that the corresponding catenoid is stable. Furthermore, the second equilibrium solution can be determined by identifying the other intersection point of the same iso-Λ and iso-V. This occurs at A', where $(\theta_c, \theta_d) \simeq (177.08°, 170.1°)$. The second solution lies outside the stability region, which corresponds to an unstable catenoid. The point D at $(\Lambda, V) \simeq (0.5555, 0.1742)$ in Fig. 2.7(b) can be described in the same manner. The two equilibrium solutions are represented in Fig. 2.7(d) by D at $(\theta_c, \theta_d) = (130°, 170°)$ and D' at $(\theta_c, \theta_d) \simeq (175.32°, 174.96°)$. Both equilibrium solutions lie outside the stability region and correspond to unstable catenoids.

2.6 Concluding remarks

We have examined the equilibrium and stability of catenoids bridging a circular disc and plate where the equilibrium surface is pinned at one contact line to the disc edge with the other free to move on the plate. Drawing on the second variation of potential energy, the existence, maximal stability, and stability regions were analytically determined. These were represented in the favourable and canonical phase diagrams. The equilibrium solution multiplicity subject to various constraints was discussed in detail. The results showed that all catenoids are stable with respect to non-axisymmetric perturbations; for a fixed contact angle, there exists a critical volume below which catenoids are unstable to axisymmetric perturbations. The canonical phase diagram furnishes a one-to-one correspondence between points in the existence region and equilibrium solutions where the stability-region boundary separates the points corresponding to stable catenoids from those corresponding to unstable ones. No such correspondence can be established in the favourable phase diagram. Furthermore, the canonical phase diagram conveniently demonstrates how the catenoid geometrical symmetry affects the stability regions. For example, the maximal stability region symmetry with respect to the phase diagram minor diagonal indicates the reflective symmetry (with respect to the ratio of lower and upper disc diameters) of catenoids with two pinned contact lines. Moreover, the asymmetric shape of the stability region shows how a catenoid free contact line with a substrate breaks the equatorial and reflective symmetries. The stability limit presented here is a limiting case for the minimum volume stability limit of liquid bridges when the mean curvature approaches zero (Akbari *et al.*, 2015*c*). The static stability limits are useful for predicting the transition of the time scale from the quasi-static to the intermediate phases of contact-drop dispensing.

Chapter 3

Liquid bridge stability with a free contact line

3.1 Preface

This chapter generalizes the results of chapter 2 to liquid bridges with non-zero mean curvature. Catenoids are specified by two, whereas general liquid bridges are specified by three geometrical parameters. Note that liquid bridges with a free contact line have no equatorial symmetry, which is emphasized in this chapter by comparing the stability region and bifurcations at critical states to those of liquid bridges pinned at two equal discs.

3.2 Abstract

The static stability of weightless liquid bridges, having a free contact line with respect to axisymmetric and non-axisymmetric perturbations is studied. Constant-volume and constant-pressure stability regions are constructed in slenderness versus cylindrical volume diagrams for fixed contact angles. Bifurcations along the stability-region boundaries are characterized from the structure of axisymmetric bridge branches and families of equilibria. A wavenumber definition is presented based on the pieces-of-sphere states at branch terminal points to classify equilibrium branches and identify branch connections. Compared with liquid bridges pinned at two equal discs, the free contact line breaks the equatorial and reflective

symmetries, affecting the lower boundary of the constant-volume stability region where axisymmetric perturbations are critical. Stability is lost at transcritical bifurcations and turning points along this boundary. The nature of bifurcations along the stability-region boundaries where non-axisymmetric perturbations are critical is not influenced by the free contact line, and stability is lost at pitchfork bifurcations. Our results furnish the maximum slenderness stability limit for drop deposition on real surfaces when the contact angle approaches the receding contact angle.

3.3 Introduction

Recent advances in nano-printing and nano-lithography (Huo *et al.*, 2008, Salaita *et al.*, 2007, Shim *et al.*, 2011) have provided new directions for studying liquid bridges. These techniques are the basis for nanoarray fabrication, which is central to data storage, pharmaceutical screening and detection, proteomics, and genotyping (Choi *et al.*, 2009, Drmanac *et al.*, 2010, Lee *et al.*, 2002, Salaita *et al.*, 2007). The ever-shrinking trend in electronic and diagnostic devices requires novel molecular-resolution and cost-effective patterning techniques. Direct-write constructive lithographic tools have been developed over the last decade to address this demand (Hwang *et al.*, 2010, Salaita *et al.*, 2007).

Contact-drop dispensing is the basis of several direct-write lithographic techniques, such as dip-pen nano-lithography (Piner *et al.*, 1999) and polymer pen lithography (Huo *et al.*, 2008), where hard and soft materials are directly deposited onto a surface from the scanning probe tip. This contrasts with non-contact-drop dispensing where the drop separates from the nozzle tip before contact with the surface. Satellite drop formation (Tjahjadi *et al.*, 1992) is a major disadvantage of non-contact-drop dispensing compared with contact-drop dispensing (Cheng & Kricka, 2001). Satellites are undesirable in surface patterning because they usually do not merge with or follow the trajectory of primary drops, landing unpredictably on the surface (Cheng & Kricka, 2001, Eggers, 1997). This degrades the pattern resolution of nanoarrays, causing background signals (Lee *et al.*, 2002) and complicating the interpretation of readings (Cheng & Kricka, 2001).

Liquid bridges in contact-drop dispensing feature a moving contact line. Recent studies have shown that the stability and breakup dynamics of these liquid bridges are greatly influenced by surface characteristics that manifest in wettability and contact-angle hysteresis (Akbari *et al.*, 2015*a*, Dodds *et al.*, 2011, Qian *et al.*, 2009). However, most previous studies have addressed the stability of liquid bridges pinned at both contact lines. Gillette & Dyson (1971) initiated the first attempt to construct the stability region[1] of weightless liquid bridges spanning two equal circular discs. They considered volume-preserving, axisymmetric perturbations and obtained the upper and lower boundary of the stability region with respect to the cylindrical volume V and slenderness Λ. These correspond to the minimum (maximum) volume (slenderness) and maximum (minimum) volume (slenderness) stability limits for a given slenderness (volume) (Gillette & Dyson, 1971). For short liquid bridges ($\Lambda < 0.81$), the dihedral angle of the interface with the disc edge θ_d is limited by a geometric constraint since $\theta_d > \pi$ cannot be experimentally realized for liquid bridges between flat plates (see Fig. 3.1). Gillette & Dyson (1971) attributed the lower boundary for short liquid bridges to this constraint rather than critical states of the interface. They also associated the upper boundary with nodoids of complete period with $\theta_d = -\pi/2$. However, later studies provided a more comprehensive picture of the stability-region structure (Myshkis *et al.*, 1987). According to Myshkis *et al.* (1987), Slobozhanin constructed the stability region with respect to arbitrary perturbations and showed that, for a fixed slenderness, liquid bridges lose stability to non-axisymmetric perturbations at a smaller volume than predicted by Gillette & Dyson (1971) when the volume is increased. This corresponds to nodoids with incomplete period and $\theta_d = 0$. Slobozhanin also demonstrated that the lower boundary for short liquid bridges corresponds to critical states of the interface with respect to non-axisymmetric perturbations with $\theta_d = \pi$. Therefore, short liquid bridges with $\theta_d > \pi$ cannot be experimentally realized, regardless of geometric constraints.

Equal circular supports impart geometrical symmetry to bridge equilibrium shapes. Combined with symmetric equations of equilibrium (Young-Laplace equation under zero-gravity), this influences the dynamics, structural stability, and bifurcation characteristics along the stability region boundaries (Seydel, 2009). For example, Gillette & Dyson (1971) proved that all stable axisymmetric liquid bridges between two equal circular discs are also equatorially symmetric. The effect of several symmetry-breaking parameters has been studied in the literature. Unequal support discs break the equatorial and reflective symmetries. Martínez &

[1]Unless stated otherwise, stability region refers to the stability region with respect to constant-volume perturbations throughout the paper.

Perales (1986) studied the stability of liquid bridges between unequal discs with respect to axisymmetric perturbations and only constructed part of the lower boundary. Here, the ratio of lower to upper disc diameters K is the symmetry-breaking parameter. Slobozhanin *et al.* (1995) constructed the entire stability region for a broad range of K, accounting for axisymmetric and non-axisymmetric perturbations. The nature of instabilities for the lower boundary is the same as the equal-disc case ($K = 1$). However, the behaviour of the upper boundary is more complicated. Slobozhanin *et al.* (1995) showed that, for $0.307 \leq K \leq 1$, non-axisymmetric perturbations are the most dangerous along the entire upper boundary, whereas, for $0 < K < 0.307$, non-axisymmetric (axisymmetric) perturbations are critical for a part of the upper boundary corresponding to equilibrium surfaces with a large (small) dihedral angle at the lower disc. Slobozhanin & Perales (1993) considered the effect of gravity on the stability region where the Bond number Bo is a non-geometrical symmetry-breaking parameter. Unlike weightless liquid bridges, the stability region boundary is a closed curve, shrinking in the (Λ, V) diagram with increasing Bo. This clearly indicates the destabilizing effect of gravity. The nature of instabilities for the lower and upper boundaries is similar to the weightless case ($\mathrm{Bo} = 0$). Slobozhanin & Perales (1996) constructed the stability region of isorotating liquid bridges under zero-gravity for a wide range of Weber numbers We. Although not a symmetry-breaking parameter, We significantly influences the stability-region structure. Similar to the latter case, the stability region boundary is a closed curve for $\mathrm{We} \lesssim 2.05$. Here, instabilities are of similar nature to the static case ($\mathrm{We} = 0$) along the stability region boundaries, except the lower boundary for short liquid bridges. Here, critical states of the interface do not always occur at $\theta_d = \pi$, corresponding to the limiting surfaces resulting from the foregoing geometric constraint. The stability region breaks into two disconnected parts at $\mathrm{We} \approx 2.05$, the smaller of which disappears when $\mathrm{We} \gtrsim 2.45$. Similar to Bo, We has a large destabilizing effect. Slobozhanin & Alexander (1998) studied the combined effect of disc inequality and gravity with respect to arbitrary perturbations, providing a deeper insight into the complex structure of the stability region.

When losing stability, the nature of instabilities at critical equilibrium states has significant implications for the dynamics and evolution of capillary surfaces. The stability-region boundaries correspond to critical states at which continuous branch continuation is not uniquely possible (Myshkis *et al.*, 1987, Seydel, 2009), and the equilibrium branches bifurcate. Depending on the structure of the potential energy, this may result in a hard, soft-dangerous, or soft-safe stability loss (Myshkis *et al.*, 1987). Slobozhanin *et al.* (1997) examined the bifurcation of weightless liquid bridges between equal discs along the entire stability-region boundaries. For a fixed Λ, axisymmetric bridges lose stability to non-axisymmetric perturbations with increasing V at a supercritical (subcritical) bifurcation along the stability-region upper boundary when $\Lambda > 0.4946$ ($\Lambda < 0.4946$). Axisymmetric bridges experience a soft-safe stability loss at supercritical pitchforks, leading to a continuous deformation to non-axisymmetric shapes with incremental increase in volume, whereas they undergo a hard stability loss at subcritical pitchforks, resulting in a sharp, discontinuous deformation to non-axisymmetric shapes. Here, axisymmetric bridges seek the closest stable and dynamically accessible non-axisymmetric configurations. This was proved to be in quantitative agreement with experimental observations (Russo & Steen, 1986, Slobozhanin *et al.*, 1997). Similarly, axisymmetric bridges lose stability at supercritical and subcritical pitchfork bifurcations to non-axisymmetric perturbations along the lower boundary for small slendernesses. However, this results in contact-line detachment due to the geometric constraints (Meseguer *et al.*, 1995, Slobozhanin & Perales, 1993). At larger slendernesses, axisymmetric bridges lose stability to axisymmetric perturbations at turning points (subcritical pitchforks) for $\Lambda < 2.13$ ($\Lambda > 2.13$). This is a hard stability loss, causing the bridge to break into two primary drops. The latter stability-region boundary is of particular interest to contact-drop dispensing where the drop volume is to be controlled. Here, critical perturbations are reflectively symmetric (antisymmetric) at turning points (pitchforks) resulting in the formation of two equal (unequal) primary drops upon breakup (Meseguer *et al.*, 1995).

Having a free contact line, liquid bridges in contact-drop dispensing are expected to exhibit a different behaviour than those considered in the foregoing studies. We have recently demonstrated the symmetry-breaking effect of a free contact line for catenoids as a special case (Akbari *et al.*, 2015*a*). Equilibrium solutions and their stability are affected through the integration constants of the Young-Laplace equation and the boundary conditions of the corresponding Sturm-Liouville problem, respectively. The latter has not been fully appreciated in the literature. Dodds *et al.* (2009) examined the dynamics of stretching liquid bridges with two free contact lines between plates and cavities. The breakup length is then compared to Plateau's stability limit for a cylinder spanning the same support plates with the same radius as the bridge neck, providing an upper bound on the capillary number for which the quasi-static assumption is valid. However, the contact-line influence on the cylinder static stability limit is neglected, noting that the critical slenderness for cylinders with a free contact line and two free contact lines is less than Plateau's stability limit by $\sim 30\%$ (Akbari *et al.*, 2015*a*) and 50% (Langbein, 2002), respectively. In another paper, Qian & Breuer (2011) studied the breakup dynamics of stretching liquid bridges having a pinned/free contact line with a substrate. The static stability limit was determined for selected bridge volumes and contact angles as a benchmark for dynamic results, without constructing the stability region. They experimentally identified the static stability limit when the contact line was free.

In this paper, we examine the static stability of weightless liquid bridges, having a free contact line with respect to axisymmetric and non-axisymmetric perturbations. We construct the entire stability region for contact angles ranging from hydrophilic to hydrophobic. The stability region is presented in slenderness versus cylindrical volume diagrams with respect to constant-volume and constant-pressure perturbations. Bifurcations along the stability-region boundaries are characterized from the structure of axisymmetric bridge branches and families of equilibria, similarly to Lowry & Steen (1995). A detailed analysis, such as that of Myshkis *et al.* (1987) and Slobozhanin *et al.* (1997), provides a thorough understanding of the bifurcation structure and dynamics of equilibrium surfaces in the vicinity of critical states. We also modify Lowry and Steen's wavenumber classification and pieces-of-sphere configurations (Lowry & Steen, 1995) to account for the symmetry-breaking effect of the free contact line. Pieces-of-sphere configurations are the states at the terminal points of equilibrium solution branches (except the rotund limit of the primary branch where the terminal point corresponds to a bulged nodoid), which can serve as the starting point in branch continuation techniques. Moreover, approximate expressions for the upper and lower boundary of the stability region in the small slenderness limit are presented and compared with available formulas in the literature for liquid bridges between equal discs.

3.4 Theory

We consider a liquid of volume v bridging a circular disk with radius R_0 and a large plate. The disc and plate are separated by a distance h, as shown in Fig. 3.1. The region occupied by the liquid bridge is denoted Ω_l, and that occupied

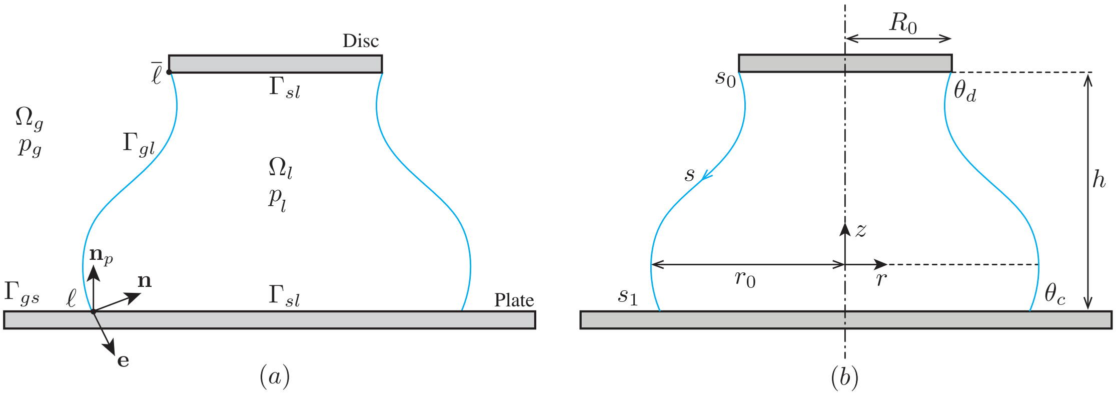


FIGURE 3.1: Weightless liquid bridge; (a) schematic and (b) coordinate system with meridian curve parametrization.

by the surrounding fluid Ω_g. The bridge is pinned to the disc and is free to slide horizontally on the plate. The gravity force is neglected in this analysis, which is a reasonable approximation when the fluids are in microgravity ($Bo \ll 1$), the bridge dimensions are much smaller than the capillary length ($R_0, h \ll \sqrt{\gamma_{gl}/g|\rho_l - \rho_g|}$), or their densities are perfectly matched. Consequently, there is a constant pressure differential between the non-hydrostatic pressure of the bridge p_l and the surrounding fluid p_g. The origin of the coordinate system is placed on the bridge equatorial plane such that the z-axis is the symmetry axis. The meridian curve is parametrized with respect to its arclength s. Axisymmetric equilibrium surfaces are specified by

$$\begin{cases} r = r(s) \\ z = z(s) \end{cases} \qquad s \in [s_0, s_1], \tag{3.1}$$

which are the stationary points of the energy functional

$$U[r(s), z(s)] = \gamma_{sl}\Gamma_{sl} + \gamma_{gl}\Gamma_{gl} + \gamma_{sg}\Gamma_{sg}, \tag{3.2}$$

where γ_{ij} is the surface tension between the phases i and j, and Γ_{ij} is the interfacial surface area. This is an isoperimetric variational problem for volume-controlled bridges, and the extremization is subject to $v[r(s), z(s)] = \text{const}$. In contrast, pressure-controlled bridges are unconstrained. However, the pressure-work contribution to the energy functional due to volume changes must be accounted for. Therefore, the energy functional to be extremized in both pressure-controlled (unconstrained) and volume-controlled (constrained) cases is

$$F[r(s), z(s)] = U[r(s), z(s)] - (p_l - p_g)v[r(s), z(s)]. \tag{3.3}$$

The corresponding Euler-Lagrange equation is the well-known Young-Laplace equation

$$\begin{cases} r'' = -z'(q - z'/r) \\ z'' = r'(q - z'/r) \end{cases} \qquad ('\equiv \mathrm{d}/\mathrm{d}s) \tag{3.4}$$

with

$$\gamma_{gl}\cos\theta_c = \gamma_{sg} - \gamma_{sl}, \qquad \text{and} \qquad \cos\theta_c = \mathbf{n}\cdot\mathbf{n}_p, \tag{3.5}$$

where $q = (p_g - p_l)/\gamma_{gl}$ measures the non-hydrostatic pressure differential (or mean curvature) (Myshkis *et al.*, 1987). Here, θ_c and θ_d are the contact and dihedral angles that the interface Γ_{gl} forms with the plate and disc, respectively. Equilibrium solutions belong to the families of doubly connected, constant-mean curvature axisymmetric surfaces, including cylinders, spheres, catenoids, nodoids, and unduloids. The catenoid is a limiting case of the nodoid and unduloid family as $q \to 0$, which has been previously studied (Akbari *et al.*, 2015*a*). The scaled lengths

$$\rho = |q|r, \quad \xi = qz, \quad \tau = |q|s \tag{3.6}$$

are adopted when $q \neq 0$ to nondimensionalize the Young-Laplace equation, furnishing

$$\begin{cases} \rho'' = -\xi'(1 - \xi'/\rho) \\ \xi'' = \rho'(1 - \xi'/\rho) \end{cases} \qquad ('\equiv \mathrm{d}/\mathrm{d}\tau) \tag{3.7}$$

with

$$\rho(0) = \rho_0, \quad \rho'(0) = 0, \quad \xi(0) = 0, \quad \xi'(0) = 1. \tag{3.8}$$

The cylindrical volume $V = v/(\pi R_0^2 h)$, scaled volume $v^* = v/(4\pi R_0^3/3)$, scaled pressure (mean curvature) $Q = qR_0$, and slenderness $\Lambda = h/R_0$ are the dimensionless parameters adopted in this paper to present the stability region and branching diagrams.

The stability region for constant-volume perturbations has a more complicated structure than at constant-pressure. The entire upper boundary and part of the lower boundary correspond to pitchfork bifurcations where non-axisymmetric perturbations are critical. Axisymmetric perturbations are critical along the lower boundary for longer liquid bridges, and the stability loss occurs at turning points and transcritical bifurcations. Hence, to capture these complexities, we apply the variational method of Myshkis *et al.* (1987) to determine the stability of equilibrium surfaces with respect to arbitrary volume-preserving perturbations. This method associates the second variation of the potential energy with the eigenvalues of the corresponding spectral (Sturm-Liouville) problem where critical states satisfy

$$\begin{cases} \mathcal{L}\varphi_0 + \mu = 0 \\ \varphi_0(\tau_0) = 0, \qquad \varphi_0'(\tau_1) + \tilde{\chi}\varphi_0(\tau_1) = 0 \\ \int_{\tau_0}^{\tau_1} \rho\varphi_0 \mathrm{d}\tau = 0 \end{cases} \tag{3.9}$$

for axisymmetric perturbations and

$$\begin{cases} (\mathcal{L} - 1/\rho^2)\varphi_1 = 0 \\ \varphi_1(\tau_0) = 0, \qquad \varphi_1'(\tau_1) + \tilde{\chi}\varphi_1(\tau_1) = 0 \end{cases} \tag{3.10}$$

for non-axisymmetric perturbations. Here, $\tau_0 = |q|s_0$, $\tau_1 = |q|s_1$, and

$$\chi = \frac{k_{1\ell}\cos\theta_c - k_{p\ell}}{\sin\theta_c} \quad \text{at } \ell, \tag{3.11}$$

and

$$\mathcal{L} \equiv \frac{\mathrm{d}^2}{\mathrm{d}\tau^2} + \frac{\rho'}{\rho}\frac{\mathrm{d}}{\mathrm{d}\tau} + \left[\left(1 - \frac{\xi'}{\rho}\right)^2 + \left(\frac{\xi'}{\rho}\right)^2\right] \tag{3.12}$$

with $\tilde{\chi} = \chi/|q|$; the first principal curvatures of the interface and plate at the contact line ℓ are denoted $k_{1\ell}$ and $k_{p\ell}$, respectively. Note that $\varphi_0(\tau)$ and $\varphi_1(\tau)$ represent the axisymmetric and non-axisymmetric perturbations corresponding to the first harmonic mode in θ. The solutions of Eqs. (3.9) and (3.10) can be written

$$\varphi_0(\tau) = C_1 w_1(\tau) + C_2 w_2(\tau) + \mu w_3(\tau), \tag{3.13}$$

$$\varphi_1(\tau) = C_4 w_4(\tau) + C_5 w_5(\tau). \tag{3.14}$$

These satisfy the following differential equations and their initial conditions

$$\mathcal{L}w_1 = 0, \qquad w_1(0) = 0, \quad w_1'(0) = 1, \tag{3.15}$$

$$\mathcal{L}w_2 = 0, \qquad w_2(0) = 1, \quad w_2'(0) = 0, \tag{3.16}$$

$$\mathcal{L}w_3 + 1 = 0, \qquad w_3(0) = 1, \quad w_3'(0) = 0, \tag{3.17}$$

$$(\mathcal{L} - 1/\rho^2)w_4 = 0, \qquad w_4(0) = 0, \quad w_4'(0) = 1, \tag{3.18}$$

$$(\mathcal{L} - 1/\rho^2)w_5 = 0, \qquad w_5(0) = 1, \quad w_5'(0) = 0. \tag{3.19}$$

An equilibrium-surface state is critical if φ_0 or φ_1 has a non-trivial solution. It can be shown (Akbari *et al.*, 2015*a*) that a non-trivial solution for φ_0 (φ_1) exists provided $\tilde{\chi} = \tilde{\chi}^0$ ($\tilde{\chi} = \tilde{\chi}^1$), where

$$\tilde{\chi}^0 = -\frac{\begin{vmatrix} w_1(\tau_0) & w_2(\tau_0) & w_3(\tau_0) \\ w_1'(\tau_1) & w_2'(\tau_1) & w_3'(\tau_1) \\ \int_{\tau_0}^{\tau_1} \rho w_1 \mathrm{d}\tau & \int_{\tau_0}^{\tau_1} \rho w_2 \mathrm{d}\tau & \int_{\tau_0}^{\tau_1} \rho w_3 \mathrm{d}\tau \end{vmatrix}}{\begin{vmatrix} w_1(\tau_0) & w_2(\tau_0) & w_3(\tau_0) \\ w_1(\tau_1) & w_2(\tau_1) & w_3(\tau_1) \\ \int_{\tau_0}^{\tau_1} \rho w_1 \mathrm{d}\tau & \int_{\tau_0}^{\tau_1} \rho w_2 \mathrm{d}\tau & \int_{\tau_0}^{\tau_1} \rho w_3 \mathrm{d}\tau \end{vmatrix}}, \tag{3.20}$$

$$\tilde{\chi}^1 = -\frac{\begin{vmatrix} w_4(\tau_0) & w_5(\tau_0) \\ w_4'(\tau_1) & w_5'(\tau_1) \end{vmatrix}}{\begin{vmatrix} w_4(\tau_0) & w_5(\tau_0) \\ w_4(\tau_1) & w_5(\tau_1) \end{vmatrix}} \tag{3.21}$$

with $\tilde{\chi}^0$ and $\tilde{\chi}^1$ the critical $\tilde{\chi}$ corresponding to axisymmetric and non-axisymmetric perturbations, respectively. One can deduce from the properties of the spectral problem that an equilibrium surface is stable (unstable) if $\tilde{\chi}$ >max$\{\tilde{\chi}^0, \tilde{\chi}^1\}$ ($\tilde{\chi}$ <max$\{\tilde{\chi}^0, \tilde{\chi}^1\}$), and it is in a critical state when $\tilde{\chi}$ =max$\{\tilde{\chi}^0, \tilde{\chi}^1\}$. Details of this method are given by Myshkis *et al.* (1987).

The critical-state criterion $\tilde{\chi}$ =max$\{\tilde{\chi}^0, \tilde{\chi}^1\}$ defines a boundary between the stability region and its complement in the space of physical parameters that the system depends on. Because this nonlinear equation has multiple solutions, it is necessary to restrict the search for critical states to a region that can be systematically constructed. The maximal stability region (MSR), a concept introduced by Slobozhanin & Tyuptsov (1974), addresses this need. The critical states associated with the MSR are determined by

$$D^0 = \begin{vmatrix} w_1(\tau_0) & w_2(\tau_0) & w_3(\tau_0) \\ w_1(\tau_1) & w_2(\tau_1) & w_3(\tau_1) \\ \int_{\tau_0}^{\tau_1} \rho w_1 \mathrm{d}\tau & \int_{\tau_0}^{\tau_1} \rho w_2 \mathrm{d}\tau & \int_{\tau_0}^{\tau_1} \rho w_3 \mathrm{d}\tau \end{vmatrix}, \tag{3.22}$$

$$D^1 = \begin{vmatrix} w_4(\tau_0) & w_5(\tau_0) \\ w_4(\tau_1) & w_5(\tau_1) \end{vmatrix}. \tag{3.23}$$

For a fixed τ_0, the first τ_1 along the meridian curve at which $D^0 = 0$ ($D^1 = 0$)

corresponds to a critical state of the MSR with respect to axisymmetric (non-axisymmetric) perturbations. Thus, one needs to seek non-trivial solutions of $\tilde{\chi}$ =max$\{\tilde{\chi}^0, \tilde{\chi}^1\}$ only for surfaces belonging to the MSR. All equilibrium surfaces outside the MSR are unstable.

The entire stability-region boundary for constant-pressure perturbations correspond to simple turning points in pressure. Hence, Maddocks's theorem can be applied to deduce stability from the structure of equilibrium branches in volume-pressure diagrams without additional analysis (Maddocks, 1987). This theorem can be rephrased as follows: Stability exchange only occurs at simple turning points for equilibrium branches without bifurcation points. Results are independent of the meridian curve boundary conditions, and, thus, are applicable to liquid bridges with free and pinned contact lines. The stability of the segment confined between the two pressure turning points is inferred from the truncated sphere stability. Moreover, it immediately follows from Maddocks's theorem that the two branch segments beyond the turning points correspond to bridges that are unstable to constant-pressure perturbations.

3.5 Results and discussion

3.5.1 Equilibrium branch construction

Solving Eq. (3.7) with the initial conditions of Eq. (3.8) furnishes the equilibrium meridian curve (Myshkis *et al.*, 1987)

$$\begin{cases} \rho(\tau) = \sqrt{1 + a^2 + 2a\cos\tau} \\ \xi(\tau) = \int_0^\tau \frac{1+a\cos t}{\rho(t)}\mathrm{d}t \end{cases}, \tag{3.24}$$

giving

$$|Q| = \sqrt{1 + a^2 + 2a\cos\tau_0}, \tag{3.25}$$

$$\Lambda = -\frac{1}{Q}\int_{\tau_0}^{\tau_1} \frac{1 + a\cos t}{\rho(t)}\mathrm{d}t, \tag{3.26}$$

$$\tan\theta_d = \mathrm{sign}(Q)\frac{1 + a\cos\tau_0}{a\sin\tau_0}, \tag{3.27}$$

$$\tan\theta_c = -\mathrm{sign}(Q)\frac{1 + a\cos\tau_1}{a\sin\tau_1}, \tag{3.28}$$

$$V = -\frac{1}{Q^3\Lambda}\int_{\tau_0}^{\tau_1} \rho(t)(1 + a\cos t)\mathrm{d}t, \tag{3.29}$$

where $a = \rho(0) - 1$. Equations (3.25)-(3.29) furnish five constraints on τ_0, τ_1, a, Q, Λ, V, θ_c, and θ_d, leaving three degrees of freedom. The last five variables are single-valued functions of the first three. Consequently, fixing (τ_0, τ_1, a), an equilibrium state characterized by $(Q, \Lambda, V, \theta_c, \theta_d)$ is uniquely specified[2]. One can choose any set of three variables to specify equilibrium states; however, this uniqueness is not necessarily preserved. Hereafter, any chosen set is denoted $\mathbf{p}$ and will be referred to as 'independent parameters'. The remaining variables are thereby termed 'dependent parameters' or 'norms' and are used as the ordinate in branching diagrams. The scaled volume v^* and cylindrical volume V can be interchanged without affecting the representation of equilibrium solution multiplicity. In this paper, we present equilibrium branches in (v^*, Q) diagrams for fixed Λ and θ_c and the stability region in (Λ, V) diagrams for fixed θ_c. Note that (v^*, Q) are the preferred coordinates in which stability limits can be associated with turning points for constant-volume and constant-pressure axisymmetric perturbations (Maddocks, 1987).

Equations (3.25)-(3.29) define a three-dimensional surface embedded in an eight-dimensional manifold. All equilibrium solutions can be associated with points belonging to the region $\mathcal{R}$ (existence region) represented by this three-dimensional manifold. The critical state conditions alluded to in section 3.4 split $\mathcal{R}$ into two mutually exclusive regions of stability $\mathcal{R}_{st}$ and instability $\mathcal{R}_{us}$. Therefore, stability diagrams are projections of $\mathcal{R}_{st}$ onto the subspace of independent parameters. This furnishes a convenient setting for constructing equilibrium branches since standard branch continuation techniques can be applied in the space of independent parameters. Note that this simplification is possible because the meridian curve and bridge geometry can be described analytically by Eqs. (3.24)-(3.29). However, the Young-Laplace equation has no analytical solution for several other problems, such as liquid bridges under gravitational or centrifugal forces (Myshkis *et al.*, 1987). Here, the meridian curve is obtained numerically, and equilibrium

[2]This does not imply that there is a one-to-one correspondence between (τ_0, τ_1, a) and equilibrium states. In fact, equilibrium meridian curves are invariant with respect to the transformation

$$\begin{cases} \bar{\tau}_0 = \tau_0 + (2n-1)\pi \\ \bar{\tau}_1 = \tau_1 + (2n-1)\pi \\ \bar{a} = -a \end{cases}, \qquad n \in \mathbb{Z}.$$

All (τ_0, τ_1, a) satisfying this transformation specify identical equilibrium states, and distinguishing them is insignificant.

branches are traced in a function space. A numerical procedure for such problems is given by Slobozhanin & Perales (1993).

Solving Eqs. (3.25), (3.26) and (3.28), we choose $\mathbf{p} = (\tau_0, \tau_1, a)$ and Q as the branch parameter to construct equilibrium branches for fixed Λ and θ_c. This significantly reduces the computational cost as compared to problems in which the meridian curve is computed numerically. For example, Martínez & Perales (1986) applied the same idea to construct equilibrium branches for liquid bridges pinned at two unequal discs and documented the minimum volume stability limit in terms of three physical parameters. The stability region and equilibrium branches are presented for $\mathbf{p} = (\Lambda, V, \theta_c)$. We use Keller's arclength continuation method, as outlined by Seydel (2009). Branch continuation begins at a pieces-of-sphere state and terminates at another pieces-of-sphere or a bulged nodoid with $\theta_d = -\pi$. Note that the liquid bridges in Fig. 3.1 are restricted by the geometric constraint $\theta_d \leq \pi$. This constraint is, nevertheless, relaxed to compute the entire equilibrium branch. However, we exclude self-intersecting meridian curves from equilibrium branches as they are non-physical (Slobozhanin *et al.*, 2002). Of course, this constraint is automatically satisfied by limiting equilibrium branches between two pieces-of-sphere states.

3.5.2 Pieces-of-sphere configurations

Branch classification based on wavenumber plays a significant role in the bifurcation, dynamics, and breakup of liquid bridges. Several definitions have been proposed in the literature. The number of negative eigenvalues is associated with instability modes, and, thus, is useful for applications in which suppression of instabilities is sought. This provides insight into possible ways of stabilizing liquid bridges (Marr-Lyon *et al.*, 2000). Vogel (1989) considered bridges with free contact lines between two parallel plates and defined the wavenumber n_w as the number of inflexion points in the meridian curve. Here, the wavenumber is invariant along branches with no bifurcation point when the contact angles are equal. Lowry & Steen (1995) provided a definition for bridges pinned at two equal discs where the wavenumber is invariant for branches that do not intersect the $\cos\theta_d$ axis in V versus $\cos\theta_d$ diagrams. This intersection occurs at a pitchfork bifurcation where axisymmetric perturbations are critical. Here, an even-wavenumber branch intersects an odd-wavenumber one. The number of extrema in the meridian curve is

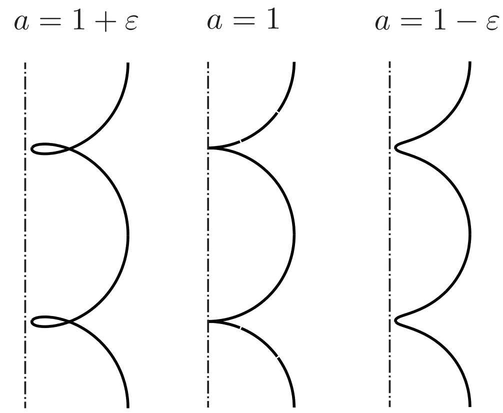


FIGURE 3.2: Transition from a self-intersecting profile to non-self-intersecting profile by varying the shape parameter a.

defined as the wavenumber in this case. They also showed that the latter two definitions are compatible with the constraints at the contact lines.

The free contact line of liquid bridges considered in this study breaks the equatorial symmetry. These bridges are subject to different constraints at the upper and lower contact lines, and their equilibrium branches exhibit no particular invariance property. This is also reflected in the bifurcations along the stability-region lower boundary. Here, branch intersections merely occur at transcritical bifurcations. Moreover, one can show by counterexample that neither the number of inflexion points nor the number of extrema is invariant along equilibrium branches. In this work, the wavenumber definition is based on the pieces-of-sphere configurations at the terminal points, which is used to label equilibrium branches. The wavenumber is invariant along each disconnected branch. For a given θ_c, there is a slenderness at which n_w and $n_w - 1$ branches intersect at a transcritical bifurcation (n_w being even). Beyond this slenderness, the transcritical bifurcation breaks into two folds where even- and odd-wavenumber half-branches meet. Here, the wavenumber is invariant along half-branches, from the pieces-of sphere state at the terminal point to the corresponding fold. This is a suitable definition because the two folds arising from the unfolding of transcritical bifurcations are indicated by a wavenumber transition.

We first demonstrate why equilibrium branches are limited by pieces-of-sphere configurations before formally defining the wavenumber for liquid bridges with a free contact line. Figure 3.2 shows how the meridian curves given by Eq. (3.24) vary with the shape parameter a. Pieces-of-sphere states are the limiting case of nodoidal (self-intersecting) and unduloidal (non-self-intersecting) liquid bridges

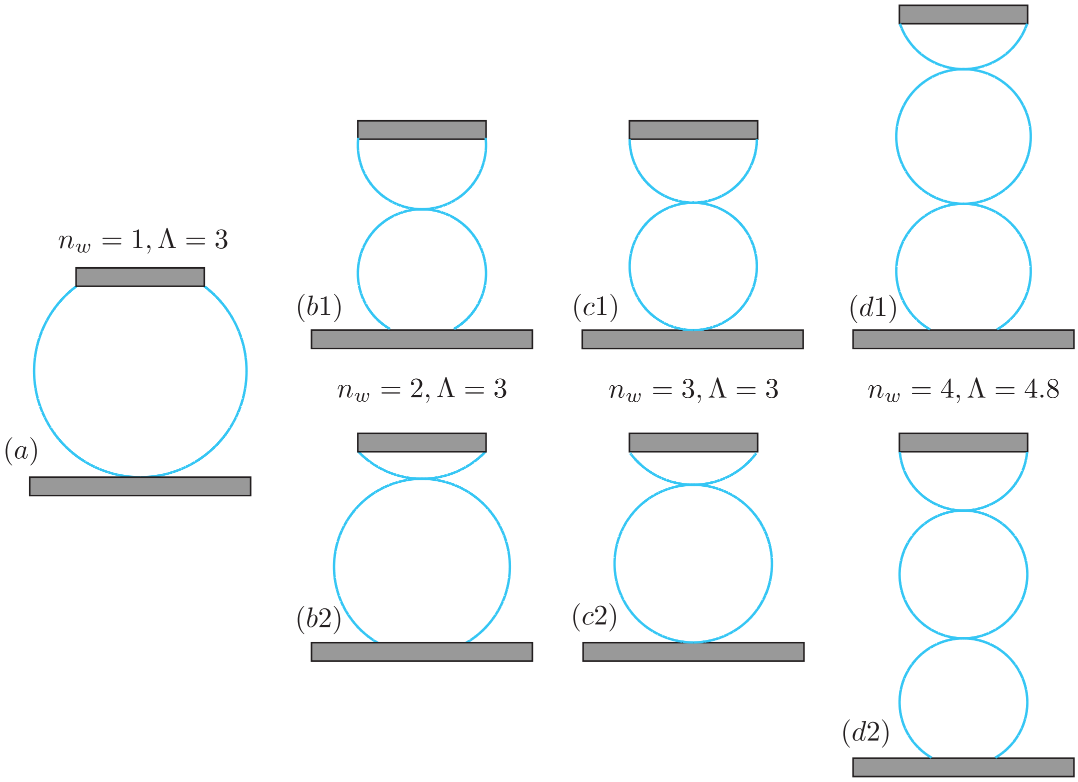


FIGURE 3.3: Wavenumber definition based on pieces-of-sphere states for the contact angle $\theta_c = 30°$.

as $a \to 1$. Increasing (decreasing) a by a small value ε, a pieces-of-sphere state is transformed into a(n) nodoid (unduloid). Note how a self-intersecting profile approaches ($a > 1$) and touches ($a = 1$) the symmetry axis, unfolds, and detaches ($a < 1$) from it as a decreases in the region $|a - 1| < \varepsilon$. For a fixed Λ and θ_c, the meridian curve changes continuously with a along the equilibrium branch. Therefore, excluding self-intersecting profiles leaves branches that are terminated at pieces-of-sphere states.

The wavenumber for pieces-of-sphere states is defined as a positive integer such that it is even (odd) when all (one of) the points at which the meridian curve touches the symmetry axis lie(s) between the plate and disc (on the plate). Each complete (truncated) sphere in the chain of spheres spanning the disc and plate adds two (one) to the wavenumber. This definition is illustrated in Fig. 3.3, where $n_w = 1$ and $n_w = 2$ are the basic states for odd-wavenumber and even-wavenumber pieces-of-sphere, respectively. These configurations can be specified analytically, furnishing a convenient starting point for branch continuation methods. We first present the solution of the basic states. Solutions for higher odd- and even-wavenumber immediately follow from the respective basic states. Denoting the slenderness and scaled arclength at the disc for the basic states by $\tilde{\Lambda}$ and $\tilde{\tau}_0$, the state $n_w = 1$ is specified by

$$\Lambda = \tilde{\Lambda}, \tag{3.30}$$

$$Q = -\frac{4\tilde{\Lambda}}{\tilde{\Lambda}^2 + 1}, \tag{3.31}$$

$$\tilde{\tau}_0 = 2\text{arctan2}\left(-\frac{2\tilde{\Lambda}}{\tilde{\Lambda}^2 + 1}, -\frac{\tilde{\Lambda}^2 - 1}{\tilde{\Lambda}^2 + 1} \right) + \pi, \tag{3.32}$$

$$\tau_1 = \pi, \tag{3.33}$$

$$a = 1, \tag{3.34}$$

$$\theta_d = (\tilde{\tau}_0 + \pi)/2, \tag{3.35}$$

where arctan2 is the four-quadrant inverse tangent. This state (Fig. 3.3(a)) exists for all Λ and is a terminal point of the primary branch. Note that Eqs. (3.31)-(3.35) are independent of θ_c, so $a \to 1$ as $\tau_1 \to \pi$ such that Eq. (3.28) is satisfied. Similarly, the state $n_w = 2$ is specified by

$$\Lambda = \tilde{\Lambda}, \tag{3.36}$$

$$Q = -2M, \tag{3.37}$$

$$\tilde{\tau}_0 = 2\text{arctan2}(-M, -M\tilde{\Lambda} + \cos\theta_c + 2) + \pi, \tag{3.38}$$

$$\tau_1 = 3\pi - 2\theta_c, \tag{3.39}$$

$$a = 1, \tag{3.40}$$

$$\theta_d = (\tilde{\tau}_0 + \pi)/2 \tag{3.41}$$

with

$$M = \frac{\tilde{\Lambda}(\cos\theta_c + 2) \pm \sqrt{1 - (2 + \cos\theta_c)^2 + \tilde{\Lambda}^2}}{\tilde{\Lambda}^2 + 1}. \tag{3.42}$$

This state has two solutions corresponding to the two terminal points of the branch $n_w = 2$ (Fig. 3.3(b1) and (b2)) and exists for all $\Lambda > \Lambda^{(2)}$, where

$$\Lambda^{(2)} = \sqrt{(2 + \cos\theta_c)^2 - 1}. \tag{3.43}$$

Here, $\Lambda^{(2)}$ is the slenderness at which the branch $n_w = 2$ originates. Unlike liquid bridges between equal discs, the two states corresponding to the terminal points of the branch $n_w = 2$ are not axial mirror images (no reflective symmetry). This reveals an intimate connection between pitchfork bifurcations and geometrical symmetry, as pointed out by Seydel (2009). The geometric idealization of liquid bridges pinned at perfectly equal discs is unlikely to be realized in practice. These axisymmetric liquid bridges are also equatorially or reflectively symmetric, the combination of which underlies the branch intersections at pitchfork bifurcations (non-generic). Small changes in geometry and constraints destroy non-generic bifurcations (Seydel, 2009), which is why they are rarely, if at all, encountered in practical problems such as that considered in this paper. The solution of higher odd-wavenumber states is

$$\Lambda = \tilde{\Lambda} + \frac{(n_w - 1)(\tilde{\Lambda}^2 + 1)}{2\tilde{\Lambda}}, \tag{3.44}$$

$$\tau_0 = \tilde{\tau}_0 - n_w \pi, \tag{3.45}$$

where Q, τ_1, a, θ_d are the same as for the basic state. These states are independent of θ_c and exist for all $\Lambda > \Lambda^{(n_w)}$, where

$$\Lambda^{(n_w)} = \sqrt{n_w^2 - 1}, \qquad n_w \in \{3, 5, 7, \cdots\}. \tag{3.46}$$

Equation (3.44) has the solutions

$$\tilde{\Lambda} = \frac{\Lambda \pm \sqrt{\Lambda^2 - n_w^2 + 1}}{n_w + 1} \tag{3.47}$$

for a given Λ corresponding to the two terminal points of the respective branch (Fig. 3.3(c1) and (c2) for $n_w = 3$). Similarly,

$$\Lambda = \tilde{\Lambda} + \frac{(n_w - 2)}{M}, \tag{3.48}$$

$$\tau_0 = \tilde{\tau}_0 - (n_w - 1)\pi \tag{3.49}$$

for higher even-wavenumber states. These states depend on θ_c and exist for all $\Lambda > \Lambda^{(n_w)}$, where

$$\Lambda^{(n_w)} = \sqrt{n_w^2 + 2n_w \cos\theta_c + (\cos 2\theta_c - 1)/2}, \quad n_w \in \{4, 6, 8, \cdots\}. \tag{3.50}$$

Equation (3.48) has the solutions

$$\tilde{\Lambda} = \frac{\begin{array}{c} 2n_w\Lambda(2+\cos\theta_c) + \Lambda(\cos 2\theta_c + 4\cos\theta_c - 1) \\ \pm\sqrt{2}(n_w - 2)\sqrt{2\Lambda^2 + 1 - 2n_w^2 - 4n_w\cos\theta_c - \cos 2\theta_c} \end{array}}{2n_w^2 + 4n_w\cos\theta_c + \cos 2\theta_c - 1} \tag{3.51}$$

for a given Λ, corresponding to the two terminal points of the respective branch (Fig. 3.3(d1) and (d2) for $n_w = 4$).

3.5.3 Stability of equilibrium branches

In this section, the typical behaviour of equilibrium branches is illustrated based on the $n_w = 1$ and $n_w = 2$ branches. Except the primary branch ($n_w = 1$), all higher wavenumber branches correspond to liquid bridges that are unstable to constant-volume and constant-pressure axisymmetric perturbations. The secondary branch ($n_w = 2$), however, plays an important role in constructing the stability region. In a certain range of Λ and θ_c, the secondary branch intersects the stable part of primary branches, splitting it into two disconnected stable segments. This appears as a kink in the stability-region lower boundary, which is similar to the one in the lower boundary for short bridges that correspond to critical states with respect to non-axisymmetric perturbations.

Figure 3.4 shows how equilibrium shapes vary along a typical equilibrium branch for a fixed Λ and θ_c. At this slenderness, only the primary branch exists. The branch starts at a pieces-of-sphere state A and ends at a bulged nodoid[3] with $\theta_d = -\pi$ (not shown). The segment EF corresponds to bridges with $\theta_d > \pi$, which cannot be realized in practice between a disc and plate (see Fig. 3.1) due to the geometric constraint mentioned in the introduction. Similar to bridges pinned at two equal discs, these are unstable to constant-volume non-axisymmetric perturbations. The remaining stable segment DE loses stability at the volume turning point D to axisymmetric perturbations. In the rotund limit, stability is lost at K, a pitchfork bifurcation, to non-axisymmetric perturbations where the bridge is a nodoid with $\theta_d = 0$. Constant-pressure stability is determined by pressure turning points. There are two turning points in pressure (G and J) in Fig. 3.4 at which stability exchange occurs according to Maddocks's theorem. The truncated sphere state I belongs to the segment GJ, implying that it is a stable

[3] What bulged and constricted mean here is not as evident as for bridges between equal discs. In this paper, bulged (constricted) bridges refer to surfaces with $r(s_1)/R_0 > 1$ ($r(s_1)/R_0 < 1$).

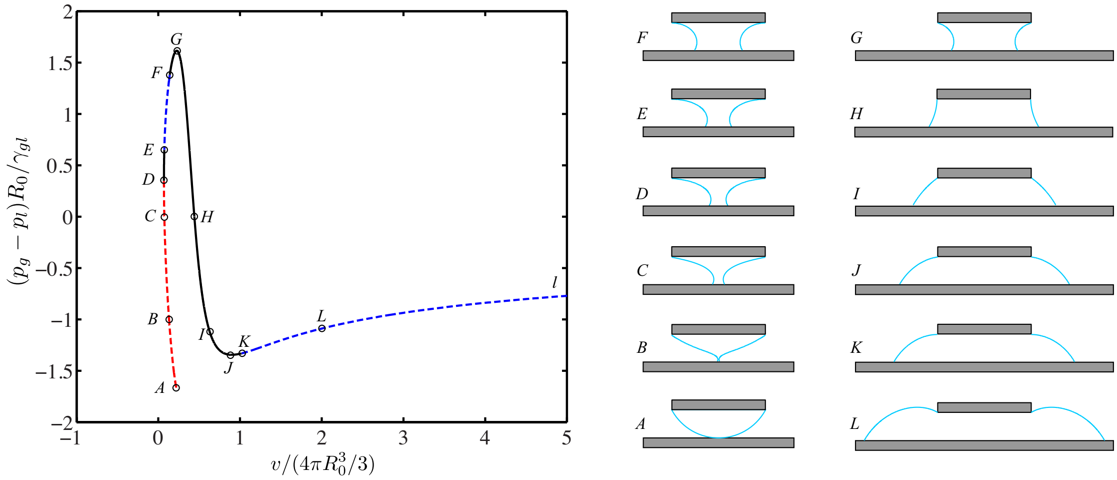


FIGURE 3.4: Equilibrium shapes along the equilibrium branch with the slenderness $\Lambda = 0.53$ and contact angle $\theta_c = 120°$. Stable states (solid) and unstable states (dashed) to axisymmetric (red) and non-axisymmetric (blue) perturbations are represented for constant-volume perturbations.

segment. Stability is lost at G and J to axisymmetric perturbations; thus, AG and Jl are unstable segments. Furthermore, two catenoids exist for the given Λ and θ_c (Akbari *et al.*, 2015*a*), corresponding to the points C and H where $Q = 0$. The catenoid C (H) is unstable (stable) to both constant-volume and constant-pressure perturbations.

The segment AD belongs to the MSR with respect to axisymmetric perturbations, and the minimum eigenvalue of the spectral problem considered by Qian & Breuer (2011) is positive alone this branch. Therefore, neglecting the role of the free-contact line in the Sturm-Liouville problem results in the misidentification of the segment AD as a stable branch.

The remainder of this section focuses on how equilibrium branches change with Λ for $\theta_c = 90°$ as an example. This is the only contact angle that is compatible with cylindrical bridges; and, thus, of particular interest. Figure 3.5 shows how equilibrium branches behave for short bridges. Here, non-axisymmetric perturbations are critical at the maximum and minimum volume stability limits. The secondary branch does not exist at these small slendernesses. For $\Lambda < 0.364$ (Fig. 3.5(a)), moving along the branch from D to A, the dihedral angle first exceeds π at C, but does not drop below π before the turning point B. Hence, there is one connected stable segment. In the rotund limit (maximum volume), constant-volume stability is lost to non-axisymmetric perturbations at D where the bridge is a bulged nodoid with $\theta_d = 0$. In the slender limit (minimum volume), constant-volume

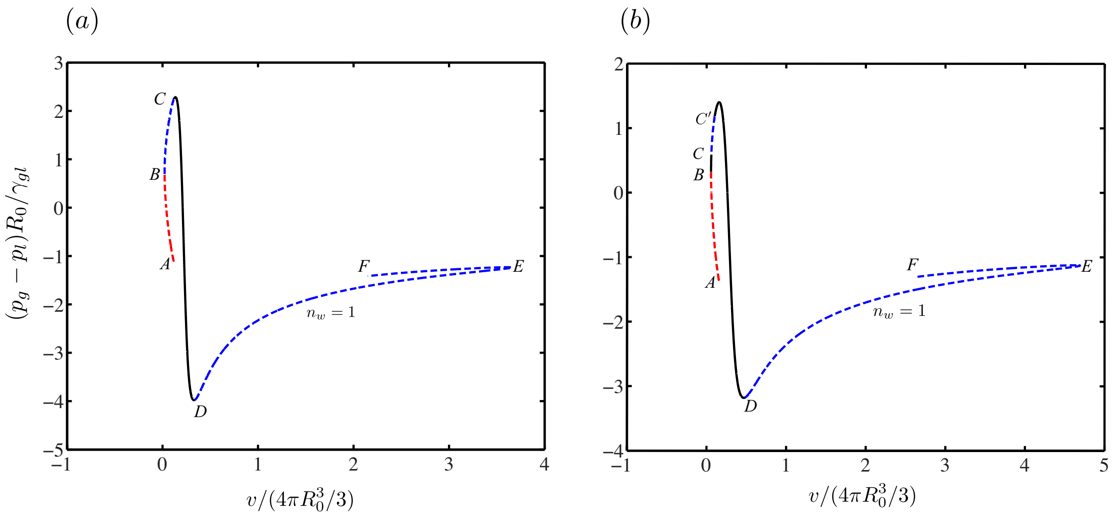


FIGURE 3.5: Equilibrium branch for short liquid bridges at fixed slenderness Λ and contact angle θ_c; (a) $\Lambda = 0.3$, $\theta_c = 90°$ and (b) $\Lambda = 0.39$, $\theta_c = 90°$. Stable states (solid) and unstable states (dashed) to axisymmetric (red) and non-axisymmetric (blue) perturbations are represented for constant-volume perturbations.

stability is lost to non-axisymmetric perturbations at C where the bridge is a constricted nodoid with $\theta_d = \pi$. At the volume turning point B, the nature of critical perturbations changes. Beyond this turning point, axisymmetric perturbations are the most dangerous along AB. Moreover, the two terminal points A and F correspond to a pieces-of-sphere and nodoid with $\theta_d = -\pi$, respectively. Note how this branch is limited by two volume and two pressure turning points. Constant-volume stability, nevertheless, cannot be determined from Maddocks's theorm since the branch has two bifurcation points between the turning points. However, Maddocks's theorm can be applied to constant-pressure stability. The segment between the pressure turning points has no bifurcation points and is stable to constant-pressure perturbations. For $0.364 < \Lambda < 0.404$ (Fig. 3.5(b)), equilibrium branches behave similarly to the previous case, except in the slender limit. Here, the dihedral angle exceeds π along CC' where bridges are unstable to non-axisymmetric perturbations. This results in two disconnected stable segments (BC and $C'D$).

Further increase in the slenderness significantly influences the behaviour of equilibrium branches. No sign change occurs in the pressure differential along the branch when $\Lambda > 0.663$. Here, there are no points on the branch corresponding to catenary profiles, and Q is always negative. Figure 3.6(a) shows the equilibrium branch for $\Lambda = 1.5$. The rotund limit D corresponds to a bulged nodiod

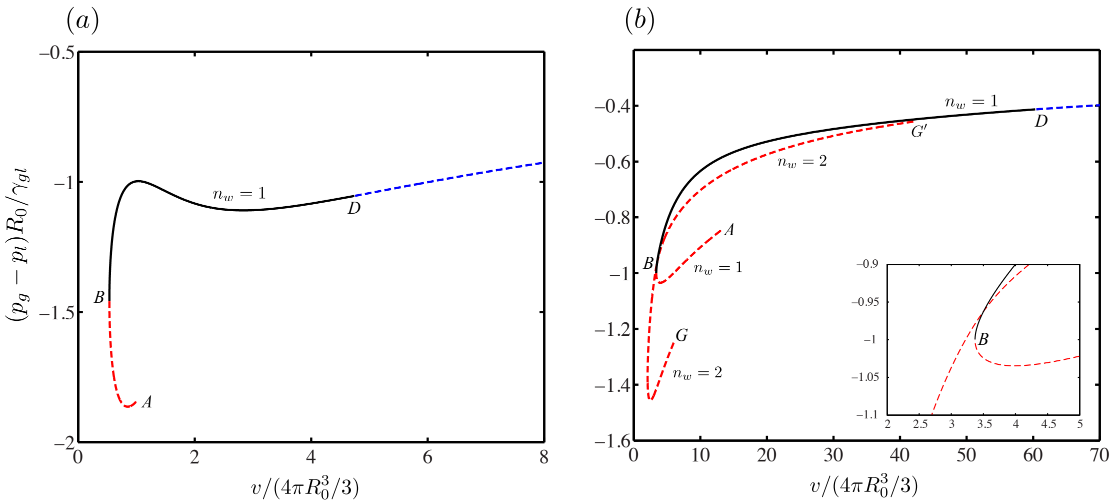


FIGURE 3.6: Equilibrium branch for medium-length liquid bridges at fixed slenderness Λ and contact angle θ_c; (a) $\Lambda = 1.5$, $\theta_c = 90°$ and (b) $\Lambda = 4.4934$, $\theta_c = 90°$. Stable states (solid) and unstable states (dashed) to axisymmetric (red) and non-axisymmetric (blue) perturbations are represented for constant-volume perturbations.

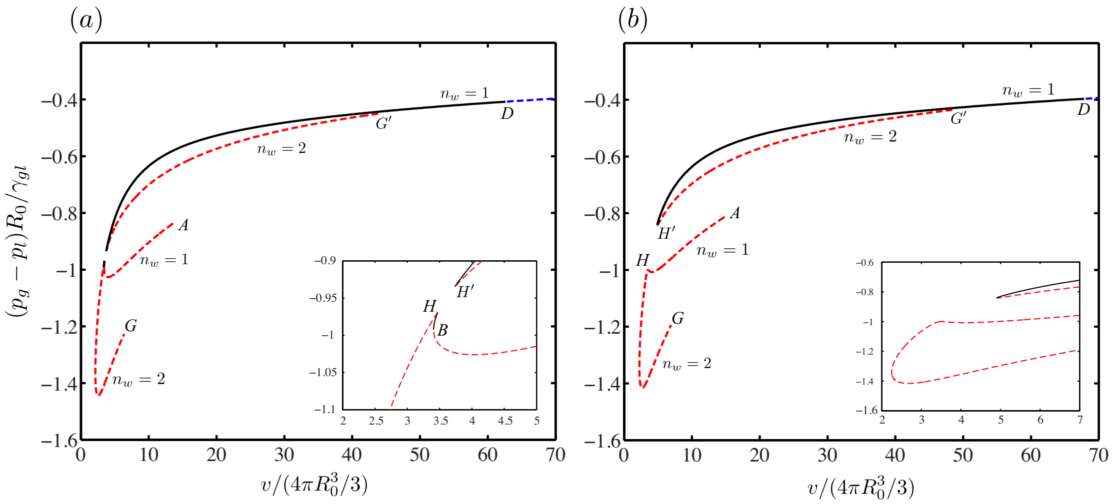


FIGURE 3.7: Equilibrium branch for long liquid bridges at fixed slenderness Λ and contact angle θ_c; (a) $\Lambda = 4.555$, $\theta_c = 90°$ and (b) $\Lambda = 4.57$, $\theta_c = 90°$. Stable states (solid) and unstable states (dashed) to axisymmetric (red) and non-axisymmetric (blue) perturbations are represented for constant-volume perturbations.

with $\theta_d = 0$ where constant-volume non-axisymmetric perturbations are critical. However, in the slender limit, stability is lost to constant-volume axisymmetric perturbations at the volume turning point B where the bridge is a constricted unduloid. The pressure turning points are the constant-pressure stability limits where axisymmetric perturbations are critical. Note that the difference between Q at the maximum and minimum pressure stability limits decreases with increasing Λ. Furthermore, the secondary branch does not exist at this slenderness. At

$\Lambda \simeq 1.862$, the two pressure turning points coalesce. The entire branch is unstable to constant-pressure perturbations beyond this slenderness. The secondary branch originates at $\Lambda = \sqrt{3}$ and grows in length with Λ. Equilibrium branches for $\Lambda \simeq 4.4934$ are illustrated in Fig. 3.6(b). This is the critical slenderness at which cylindrical bridges with a free contact line lose stability to axisymmetric perturbations (Akbari *et al.*, 2015*a*). The primary branch behaves similarly to the previous case in the rotund and slender limits. Interesting to note is the slender limit where the bridge is a cylinder at the turning point B. Increasing the slenderness beyond $\Lambda \simeq 4.4934$, bulged unduloids become the critical equilibria in the slender limit, while the cylindrical state moves past the turning point to the unstable segment AB. The secondary branch is limited by two pieces-of-sphere states at G and G'. It also crosses the primary branch above the volume turning point. However, this crossing does not correspond to a branch intersection because the equilibrium states corresponding to this (v^*, Q) on the primary and secondary branches are different.

Figure 3.7 shows typical equilibrium branches for long bridges. When $\Lambda \simeq 4.549$, the primary and secondary branches intersect at a transcritical bifurcation, and the $n_w = 1$ and $n_w = 2$ families become connected. The equilibrium state belongs to both $n_w = 1$ and $n_w = 2$ families of equilibria at the intersection. Here, the secondary branch intersects the stable part of the primary branch. Increasing Λ by a small value, the $n_w = 1$ and $n_w = 2$ branches split into two half-branches, breaking the bifurcation point into two folds (H and H'), as shown in Fig. 3.7(a). These are turning points in volume. As a result, two disconnected stable branches emerge, which lose stability to constant-volume axisymmetric perturbations at H and H'. These are the points at which $n_w = 1$ and $n_w = 2$ half-branches meet. Increasing the slenderness beyond $\Lambda \simeq 4.567$, the stable segment BH disappears, leaving two separate branches with one connected stable segment (see Fig. 3.7(b)). Further increase in the slenderness does not affect the behaviour of equilibrium branches significantly, and constant-volume stability is lost to axisymmetric perturbations at the turning point H' where the bridge is a bulged unduloid. Moreover, the cylindrical state moves from the $n_w = 1$ to $n_w = 2$ branch at larger slendernesses.

Transcritical bifurcations do not affect the stability region for all contact angles. The secondary branch intersects the stable part of the primary branch when $\theta_c \lesssim 125°$. At larger contact angles, the secondary branch intersects the unstable part below the volume turning point, as illustrated in Fig. 3.8. When $\theta_c = 120°$,

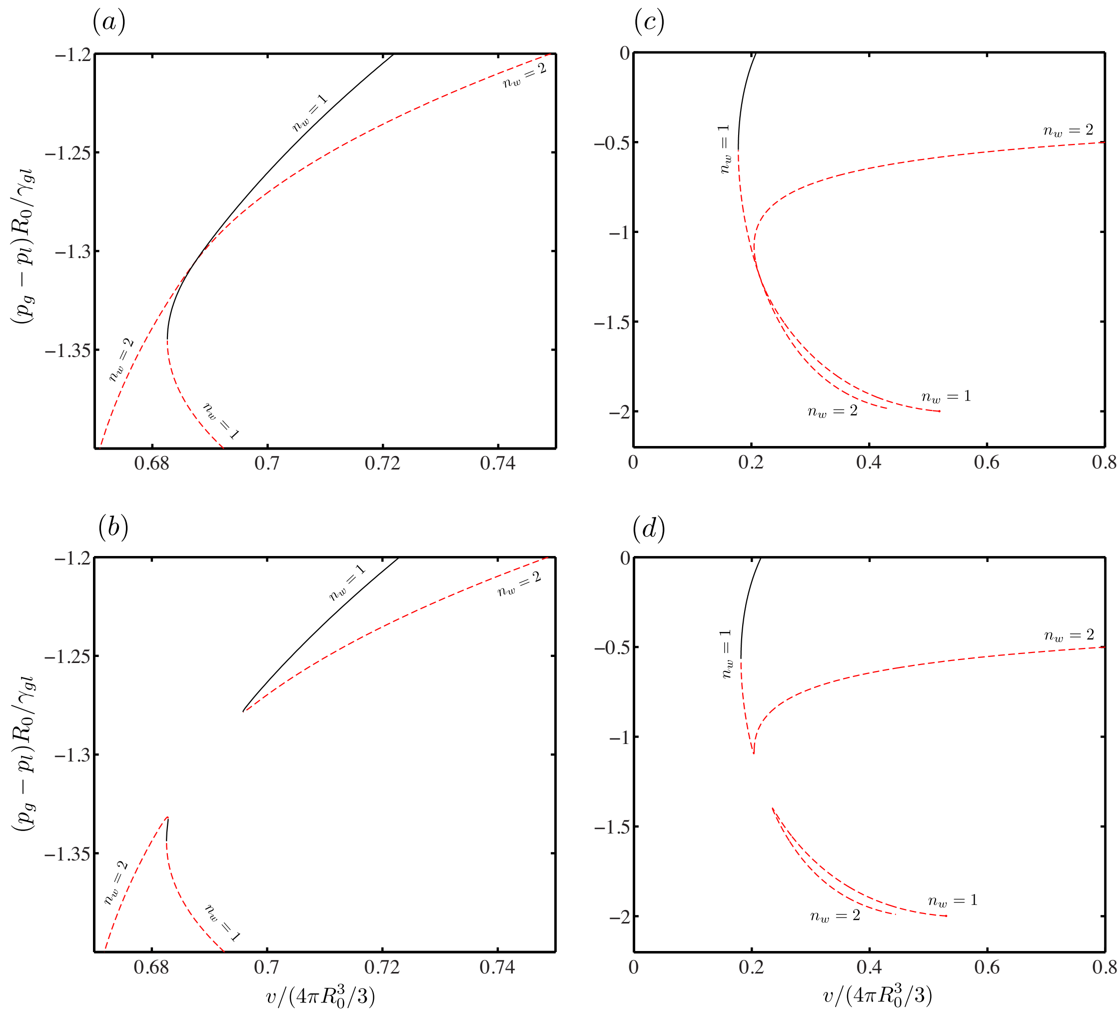


FIGURE 3.8: Equilibrium branch in the vicinity of transcritical bifurcations when the slenderness is (a) below ($\Lambda = 2.323$) and (b) above ($\Lambda = 2.325$) the transcritical bifurcation at $\Lambda \simeq 2.3233$ for the contact angle $\theta_c = 120°$, while it is (c) below ($\Lambda = 1.026$) and (d) above ($\Lambda = 1.04$) the transcritical bifurcation at $\Lambda \simeq 1.0285$ for $\theta_c = 150°$. Stable states (solid) and unstable states (dashed) to axisymmetric perturbations are represented for constant-volume perturbations.

the transcritical bifurcation lies slightly above the turning point on the stable part (Fig. 3.8(a)). A small increase in Λ splits the stable part into two disconnected stable segments (Fig. 3.8(b)). The smaller stable segment only exists in a narrow range of Λ beyond the transcritical bifurcation and eventually disappears with increasing Λ. Note that the foregoing range becomes larger with decreasing θ_c. However, when $\theta_c = 150°$, the transcritical bifurcation lies below the volume turning point on the unstable part (Fig. 3.8(c)). Increasing Λ by a small value splits the unstable part into two disconnected segments, leaving the stable part unaffected (Fig. 3.8(d)).

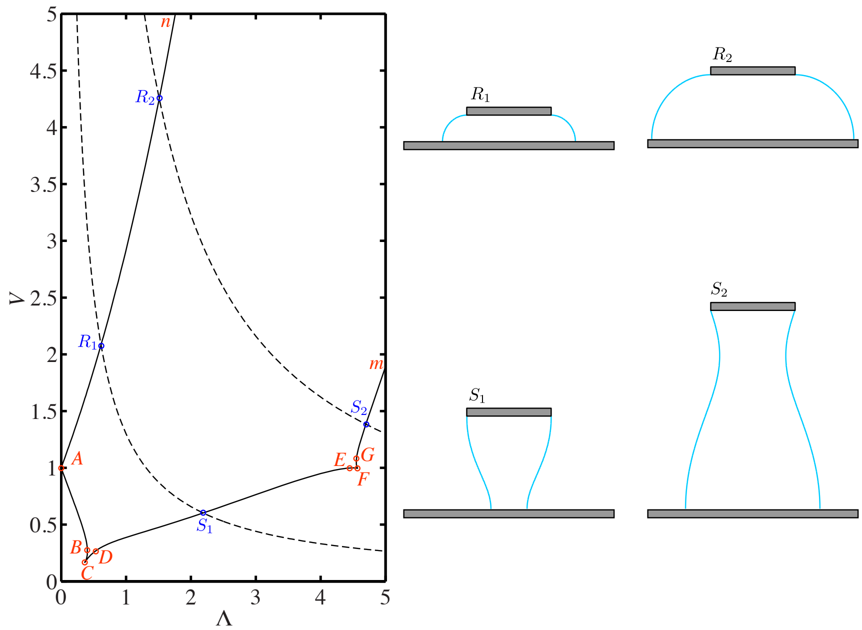


FIGURE 3.9: Stability region with respect to constant-volume perturbations with $\theta_c = 90°$. Dashed lines correspond to constant-volume drop dispensing with $v^* = 1, 4.9$ (upward).

3.5.4 Stability region

Here, we present the stability region in (Λ, V) diagrams for fixed θ_c. Note that equilibrium states are not uniquely specified by the independent parameters $\mathbf{p} = (\Lambda, V, \theta_c)$. Therefore, points inside the stability region in this space may simultaneously correspond to stable and unstable bridges. These parameters, nevertheless, can be readily measured experimentally, furnishing a convenient representation of the stability limits. We first demonstrate bifurcation characteristics and the critical perturbations along the stability region boundaries for $\theta_c = 90°$ as an example.

Figure 3.9 illustrates the stability region with respect to constant-volume perturbations for $\theta_c = 90°$. The upper boundary An and the lower boundary for short bridges ABC correspond to nodoids with $\theta_d = 0$ and $\theta_d = \pi$, respectively, where stability is lost at pitchfork bifurcations to non-axisymmetric perturbations. This behaviour is the same as for bridges pinned at two equal discs. However, a detailed bifurcation analysis is required to differentiate between supercritical and subcritical pitchforks along $nABC$ (Myshkis *et al.*, 1987, Slobozhanin *et al.*, 1997). The nodoids belonging to the boundary segment ABC are also the limiting surfaces resulting from the geometric constraint imposed by the disc ($\theta_d \le \pi$). Nodoids

Table 3.1: Bifurcation characteristics along the stability-region boundaries in Fig. 3.9 for the contact angle $\theta_c = 90°$.

Open segment	Critical surface	θ_d	Critical perturbations	Bifurcation type
nA	Bulged nodoid	$0°$	Non-axi.[1]	Pitchfork
ABC	Const.[3] nodoid	$180°$	Non-axi.	Pitchfork
CD	Const. nodoid	$(167.8°, 180°)$	Axi.[2]	Turning point
Point D	Catenoid	$167.8°$	Axi.	Turning point
DE	Const. unduloid	$(81.9°, 167.8°)$	Axi.	Turning point
Point E	Cylinder	$90°$	Axi.	Turning point
EFG	Bulged unduloid	$(90°, 102.3°)$	Axi.	Turning point
Point G	Bulged unduloid	$102.3°$	Axi.	Transcritical
Gm	Bulged unduloid	$(102.3°, 180°)$	Axi.	Turning point

[1] Non-axisymmetric perturbations.
[2] Axisymmetric perturbations.
[3] Constricted.

are the critical surfaces along the segment CD where stability is lost at turning points to axisymmetric perturbations. The critical surface at D is a catenoid with $\theta_d \simeq 167.8°$. Axisymmetric perturbations are the most dangerous along DE where the dihedral angle first decreases from $\theta_d \simeq 167.8°$ to $\theta_d \simeq 81.9°$ and then increases to $\theta_d = 90°$ at E. The critical surface at E is a cylinder with $\Lambda \simeq 4.4934$ corresponding to the stability limit of cylindrical bridges with a free contact line (B in Fig. 3.6(b)). This is the point where the line $V = 1$ is tangent to the boundary segment $CDEF$. This line intersects the segment FGm at $\Lambda \simeq 4.5667$. The segment AE of the $V = 1$ line is the locus of stable cylindrical bridges. However, the remaining segment inside the stability region is the locus of stable unduliods with $V = 1$; the cylindrical bridges corresponding to this segment are unstable. This contrasts with the stability region of liquid bridges pinned at two equal discs where the slope of the lower boundary at $(\Lambda, V) = (2\pi, 1)$ is $1/\pi$. Axisymmetric perturbations are critical along $EFGm$. Here, except the point G, which corresponds to a transcritical bifurcation, stability is lost at turning points. Bifurcation characteristics along the stability-region boundaries are summarized in table 3.1. As discussed in section 3.5.3, the kink EFG is a result of the secondary branch intersecting and splitting the stable part of the primary branch when $4.549 < \Lambda < 4.567$. Note that transcritical bifurcation only occurs at one point along the lower boundary, indicating why transcritical bifurcations are non-generic and unlikely to be realized in practice.

Dashed lines in Fig. 3.9 are the paths corresponding to constant-volume drop

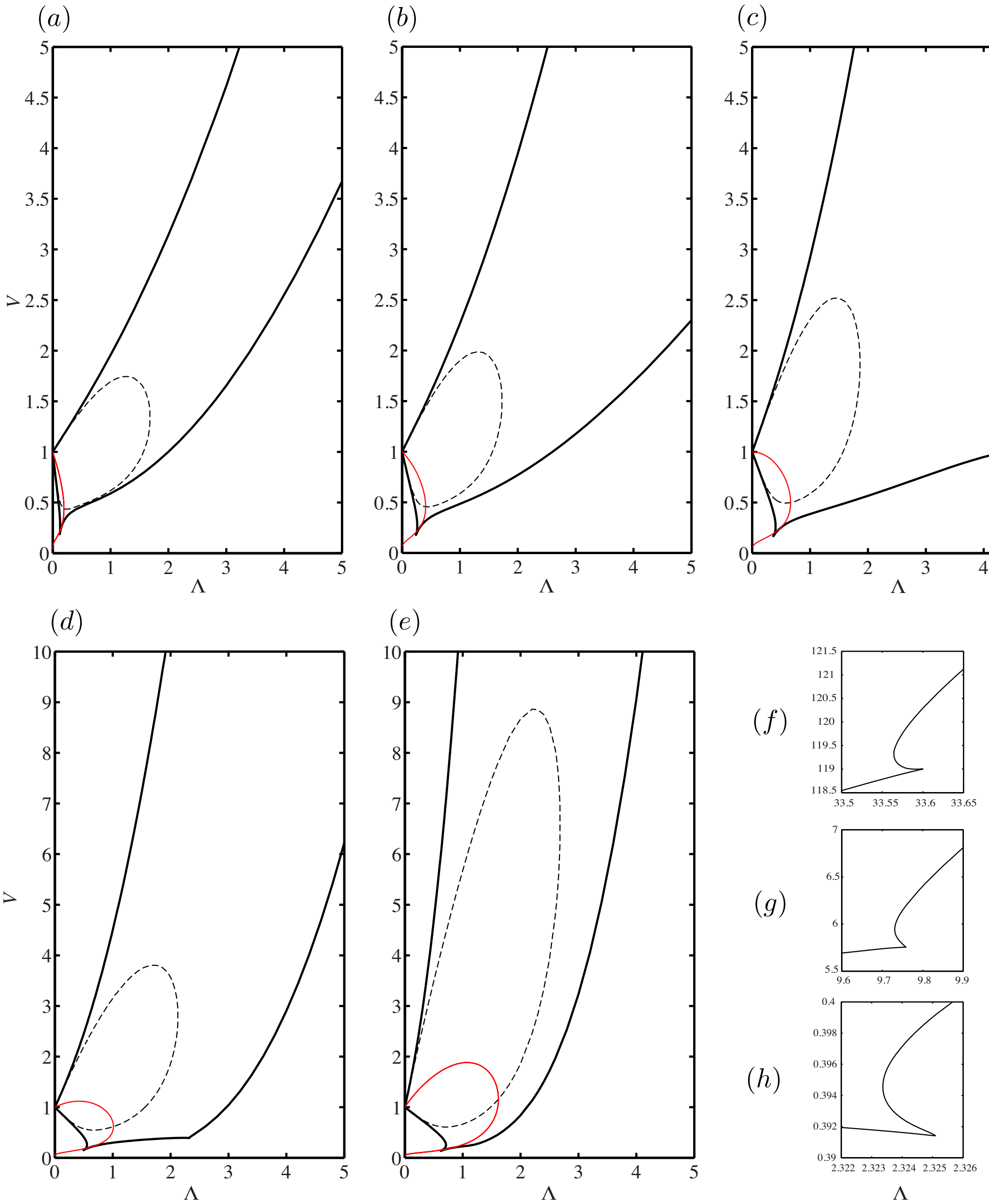


FIGURE 3.10: Stability region with respect to constant-volume (thick solid) and constant-pressure (dashed) perturbations with (a) $\theta_c = 30°$, (b) $\theta_c = 60°$, (c) $\theta_c = 90°$, (d) $\theta_c = 120°$, and (e) $\theta_c = 150°$. Red lines indicate the locus of catenoids at the respective contact angle. The lower boundary in the transcritical bifurcation neighbourhood is magnified for (f) $\theta_c = 30°$, (g) $\theta_c = 60°$, and (h) $\theta_c = 120°$.

dispensing. The bridge interface is deformed into a non-axisymmetric surface in the rotund limit if the bridge is squeezed, whereas the bridge breaks into two primary drops in the slender limit if it is stretched. The points R_1 (R_2) and S_1 (S_2) correspond to the rotund and slender limits for $v^* = 1$ ($v^* = 4.9$), respectively. The critical surface is a bulged nodiod with $\theta_d = 0$ in the rotund limit for both paths. However, the slender limit behaves differently for small and large bridge volumes. At small volumes, the drop dispensing path intersects the segment DE where the critical surface is a constricted unduloid, and a neck forms close to the plate, whereas, at large volumes, the drop dispensing path intersects the segment $EFGm$ where the critical surface is a bulged unduloid, and a neck forms close to the disc. Consequently, the ratio of the dispensed drop volume to the bridge volume in the former case is smaller than in the latter case. A full dynamic analysis is, nevertheless, necessary to precisely quantify this ratio (Akbari *et al.*, 2015*b*).

The stability region with respect to constant-volume and constant-pressure perturbations is plotted as (Λ, V) diagrams for fixed θ_c in Fig. 3.10. The constant-volume stability region is an open area, which completely encompasses the constant-pressure stability region at all θ_c. This implies that all the liquid bridges that are stable to constant-pressure perturbations are also stable to constant-volume perturbations, but the converse does not hold. The maximum volume stability limit increases with Λ more rapidly as θ_c increases. The same behaviour is observed for the minimum volume stability limit at large slendernesses. Red lines indicate the locus of catenoids ($Q = 0$) at the respective contact angle. Along these curves, Λ reaches a maximum when intersecting the boundary of the constant-pressure stability region for a fixed θ_c. This point splits the catenoid curve into two segments. The upper segment corresponds to stable catenoids, while the lower one corresponds to catenoids that are unstable with respect to constant-pressure axisymmetric perturbations. For a given θ_c, liquid bridges with $Q > 0$ can only correspond to the points inside the region confined between the blue curves and the V-axis. Furthermore, the kink in the lower boundary that precedes the transcritical bifurcation only exists when $\theta_c \lesssim 125°$ and occurs at larger Λ and V for smaller θ_c.

We conclude this section by comparing the limiting behaviour of the maximum and minimum volume stability limits when $\Lambda \ll 1$ for liquid bridges considered in this study (case-I) and those pinned at two equal discs (case-II). As previously mentioned, the upper and lower boundaries of the constant-volume stability region, respectively, correspond to bulged nodoids with $\theta_d = 0$ and constricted nodoids with $\theta_d = \pi$ for case-I where non-axisymmetric perturbations are critical. Expanding the cylindrical volume as a power series in slenderness for $\Lambda \ll 1$, we

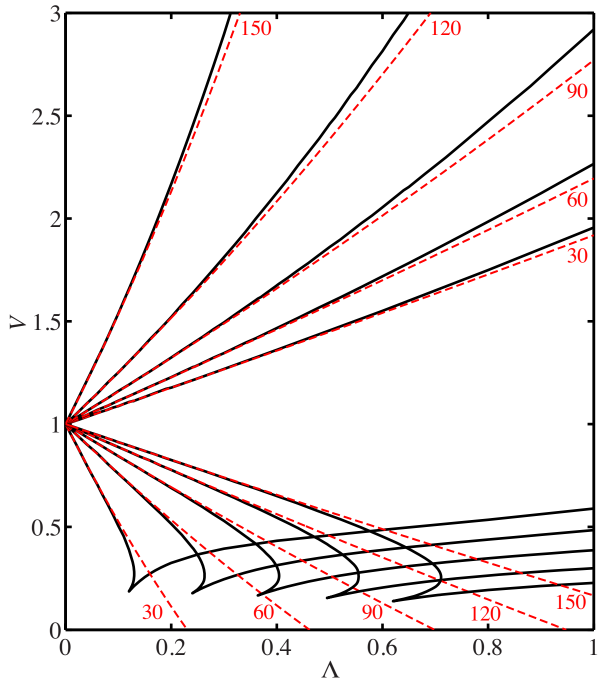


FIGURE 3.11: Comparison of approximate formulas Eqs. (3.52) and (3.53) (dashed) and numerical computations (solid) for the upper and lower boundaries of the constant-volume stability region in the small-slenderness limit. Labels denote the contact angle θ_c in degrees.

find the approximate expressions

$$\begin{aligned} V = {} & 1 + \frac{1}{4}\sec^4(\theta_c/2)(\pi - \theta_c + \cos\theta_c \sin\theta_c)\Lambda \\ & - \frac{1}{384}\sec^8(\theta_c/2)[-97 + 24(\pi-\theta_c)^2 - 136\cos\theta_c - 32\cos(2\theta_c) \\ & + 8\cos(3\theta_c) + \cos(4\theta_c) + 24(\pi-\theta_c)\sin(2\theta_c)]\Lambda^2 + O(\Lambda^3), \end{aligned} \tag{3.52}$$

$$\begin{aligned} V = {} & 1 + \frac{1}{4}\csc^4(\theta_c/2)(-\theta_c + \cos\theta_c \sin\theta_c)\Lambda \\ & - \frac{1}{384}\csc^8(\theta_c/2)[-97 + 24\theta_c^2 + 136\cos\theta_c - 32\cos(2\theta_c) \\ & - 8\cos(3\theta_c) + \cos(4\theta_c) - 24\theta_c\sin(2\theta_c)]\Lambda^2 + O(\Lambda^3) \end{aligned} \tag{3.53}$$

for the upper and lower boundaries, respectively. These are accurate to second order in Λ and are in good agreement with numerical computations for short bridges (Fig. 3.11). Letting $\theta_c = \pi/2$, Eqs. (3.52) and (3.53), respectively, simplify to

$$V = 1 + \frac{\pi}{2}\Lambda + \left(\frac{8}{3} - \frac{\pi^2}{4}\right)\Lambda^2 + O(\Lambda^3), \tag{3.54}$$

$$V = 1 - \frac{\pi}{2}\Lambda + \left(\frac{8}{3} - \frac{\pi^2}{4}\right)\Lambda^2 + O(\Lambda^3), \tag{3.55}$$

which are identical to the expressions in the literature for case-II (Slobozhanin *et al.*, 1997) with a slenderness twice that of case-I. Recall, bulged nodoids with $\theta_d = 0$ and constricted nodoids with $\theta_d = \pi$ are also the critical surfaces along the upper and lower boundaries of the constant-volume stability region for case-II. Therefore, when $\theta_c = \pi/2$, a critical surface for case-I is half as slender as the corresponding critical surface for case-II, while their cylindrical volumes are equal. Consequently, the maximum and minimum volume stability limits of case-I for short bridges in a (Λ^I, V) diagram must approach those of case-II in a $(\Lambda^{II}/2, V)$ diagram as $\theta_c \to \pi/2$, where Λ^I and Λ^{II} denote the slenderness in case-I and -II, respectively.

3.6 Concluding remarks

We have examined the equilibrium and stability of weightless liquid bridges that are pinned at one contact line to a disc with the other free to move on a parallel plate. Constant-volume and constant-pressure stability regions were constructed in slenderness versus cylindrical volume diagrams for fixed contact angles. Bifurcations along the stability-region boundaries were determined from the structure of equilibrium branches and families of equilibria. A branch classification was proposed based on the wavenumber of pieces-of-sphere states at branch terminal points, accounting for the symmetry-breaking role of the free contact line. In comparison with liquid bridges pinned at two equal discs, the free contact line completely breaks the equatorial and reflective symmetries, destroying the pitchfork bifurcations along the lower boundary of the constant-volume stability region where axisymmetric perturbations are critical. Furthermore, the free contact line gives rise to stability loss at a transcritical bifurcation to axisymmetric perturbations, distorting the lower boundary for large liquid bridges in the contact-angle range $\theta_c \lesssim 125°$. However, the nature of bifurcations is not influenced by the free contact line along the entire upper boundary and lower boundary for short liquid bridges, where stability is lost to non-axisymmetric perturbations at pitchfork bifurcations. This is because the axial symmetry is not broken for critical surfaces with respect to non-axisymmetric perturbations.

Our results can be directly applied to drop deposition on ideal surfaces. However, real surfaces exhibit contact-angle hysteresis. The contact line remains fixed until the advancing (receding) contact angle is reached from below (above) if the bridge is squeezed (stretched). When dispensing drops on real surfaces, the interface does not admit perturbations that displace the contact line for contact angles below (above) the advancing (receding) contact angle. When the advancing or receding contact angle is reached, the contact line can freely move on the plate. Therefore, there is a transition in the stability limits between two ideal regimes for contact-drop dispensing on real surfaces: (i) liquid bridges between unequal discs with perfectly pinned contact lines, as studied by Slobozhanin *et al.* (1995), and (ii) liquid bridges with a perfectly free contact line, as studied in this paper. For a given advancing (receding) contact angle and drop volume, the stability diagrams presented in section 3.5.4 furnish the minimum (maximum) slenderness stability limit. Note that the receding contact angle is constant to a good approximation in the quasi-static phase of drop deposition due to small contact line speeds (Akbari *et al.*, 2015*a*). Thus, the maximum slenderness stability limit in our static analysis provides an upper bound for the onset of the quasi-static to intermediate regime transition in the pinching of liquid bridges.

Chapter 4

Experimental investigation of liquid bridge breakup in contact drop dispensing

4.1 Preface

In the last two chapters, the stability regions of catenoids and general liquid bridges with a free contact line were theoretically determined, and the destabilizing effect of free contact lines was demonstrated. At the maximum-volume stability limit, liquid bridges assume non-axisymmetric shapes, while at the minimum-volume stability limit, they break into two primary drops for large slendernesses. These observations are experimentally tested in this chapter, and the theoretical predictions of the stability limits are validated. Since this problem also arises in applications, such as scanning-probe lithography, the experiments are set out in the context of contact-drop dispensing. The role of contact lines in the stability of liquid bridges is also isolated, and the destabilizing effect of free contact lines is illustrated. The results shown in Figs. 4.3, 4.5, and 4.6 were obtained using the code presented in Appendix J, and those shown in Fig. 4.8 were obtained using the code presented in Appendix I.

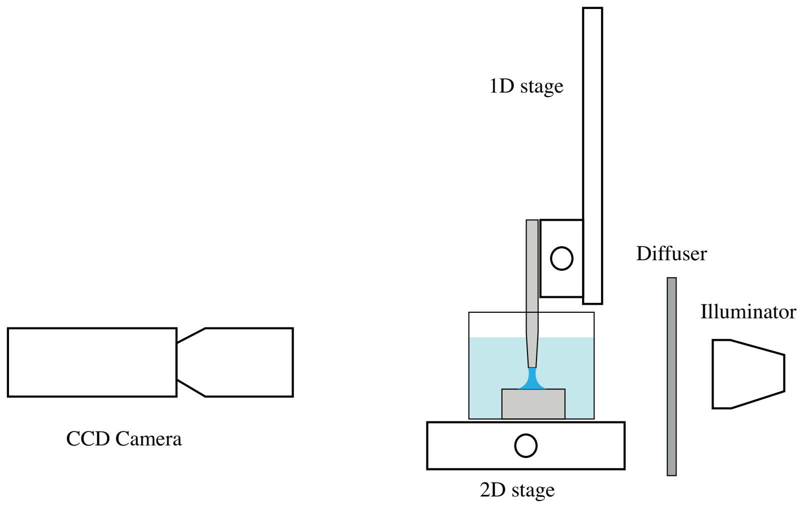


FIGURE 4.1: Schematic of the experimental setup.

4.2 Materials and methods

Experiments were performed in a cubic Plateau tank under neutrally buoyant conditions (Fig. 4.1). A silicon oil (5 cSt, Sigma Aldrich) with specific gravity 0.92 as the bridge liquid and a water-methanol mixture (volumetric mixing ratio 42:58) as the surrounding bath liquid were used. The bath temperature during experiments was constant at $\approx 68°$F. The composition of the water-methanol solution was adjusted so that its density matched that of silicon oil at the experiment temperature. Using a microsyringe, a drop with a prescribed volume in the range 5–50 μl was deposited onto a plastic coverslip (Fischer Scientific), which had been soaked in a 0.1 M hydrochloric acid solution, rinsed with DI water, and placed in the tank. A bridge was produced by gently pressing a needle with a tip diameter 1.5 mm into the drop. The needle was mounted on a one-dimensional vertical translation stage to control the bridge height, and the tank was placed on a two-dimensional positioning stage to align the drop and needle centers before contact, ensuring that the bridge is axisymmetric. The maximum (minimum) slenderness stability limit was ascertained by stretching (squeezing) the bridge until the bridge ruptured (bulged asymmetrically). A CCD camera (Prosilica GX1050, Allied Vision) with a 5X lens (Nikon GMicro-NIKKOR) was used to record the dynamics of the bridge. Images were analyzed using MATLAB. The bridge contact angle with the coverslip was adjusted by changing the interfacial tensions in the system using the anionic surfactant sodium dodecyl sulfate (SDS) (Sigma Aldrich) at concentrations in the range 0–10 gl^{-1}.

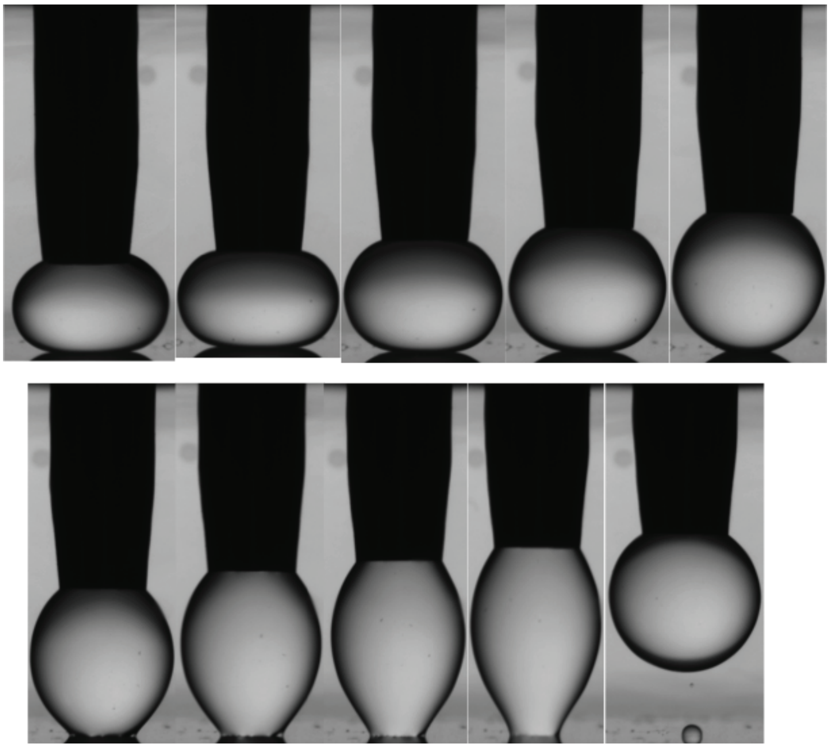


FIGURE 4.2: A typical sequence in stretching (bottom) and squeezing (top) experiments (20 μl drop).

4.3 General behaviour of liquid bridges

Axisymmetric liquid bridges were produced by aligning the drop and needle centers. Since the needle is hollow with a sharp edge, the bridge remained pinned to the needle throughout the experiments. After a drop was deposited onto a coverslip, it was squeezed in 0.01 inch steps to reach the minimum-slenderness stability limit, at which it bulges asymmetrically; the bridge was imaged at each step. The maximum-slenderness stability limit was similarly measured by stretching the bridge until it ruptured. Figure 4.2 shows a typical sequence during the squeezing and stretching of a 20 μl drop. In the rotund limit, the the non-axisymmetric deformation of the drop displaces the contact line, leading to different results upon repeating the experiment. In the slender limit, the bridge breaks into two primary drops, leaving several satellite drops suspended in the bath. Without SDS, the contact line moved only at breakup for small drop volumes (less than 10μl), and was otherwise pinned at larger volumes. Since the emphasis in this work is on the effect of moving contact lines, SDS was added to the bath solution to reduce the receding contact angle (see Fig. 3.1) of the bridge on the coverslip.

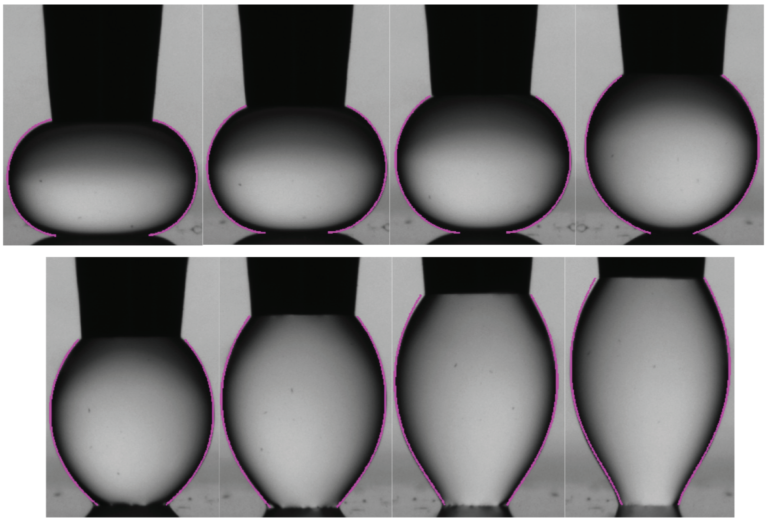


FIGURE 4.3: Typical image processing: fitting theoretical meridian curve to bridge extracted boundaries in stretching (bottom) and squeezing (top) experiments.

4.4 Feature extraction

A MATLAB code was developed to process drop and bridge images. The bridge interface with the bath solution was extracted using a gradient-based edge detection method with a Gaussian optimal smoothing filter (Marr & Hildreth, 1980). In this method, pixels on an interface are identified by finding maxima in the first directional derivative of intensity; or, equivalently, seeking zero-crossings in the second directional derivative. Derivatives were taken along normals to interfaces using high-order (8-10 points) central schemes. Then, the analytical solution of the bridge meridian curve, given by Eq. (3.24), was fitted to the extracted interfaces. Here, the unknown parameters (Q_c, τ_0, τ_1, a) were determined by minimizing the root-mean-squared normal distances between the extracted interface pixels and the theoretical meridian curve. Figure 4.3 shows typical results of the image-processing code in stretching and squeezing experiments.

4.5 Results and discussion

4.5.1 Surfactant effect

As previously stated, for large drops, the contact angle θ_c remains smaller than the receding contact angle θ_r during stretching. Thus, the contact line is pinned

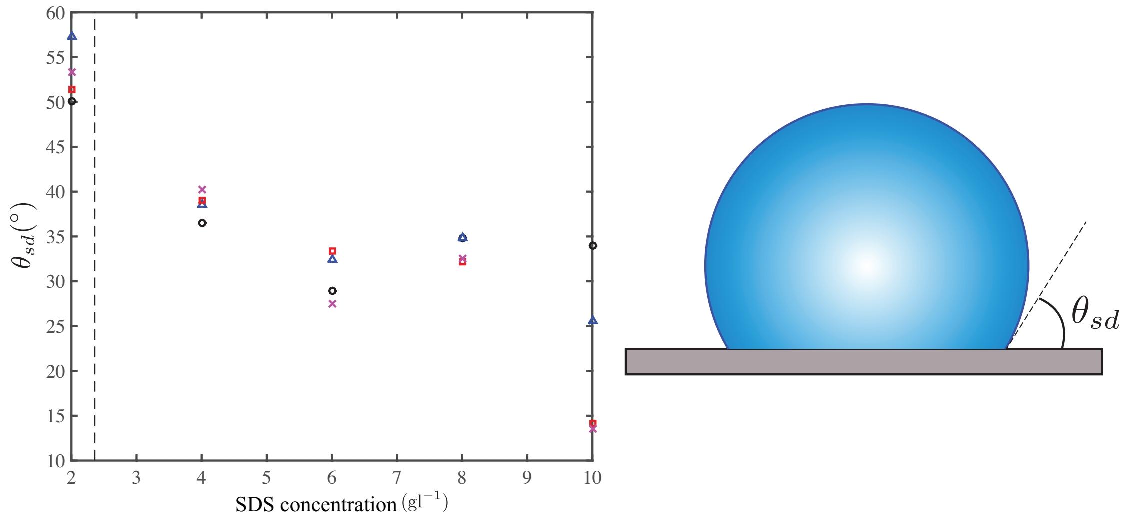


FIGURE 4.4: The surfactant-concentration effect on the sessile-drop contact angle θ_{sd} at drop volumes 5 μl (○), 10 μl (△), 15 μl (□), 20 μl (×). Dashed line indicates the critical micelle concentration of the surfactant in pure water at 25°C (Mukerjee & Mysels, 1971).

to the coverslip at breakup. To assess the stability limits of liquid bridges with a free contact line in a wider range of drop volumes, the contact angle is reduced by adding SDS to the bath solution. The critical micelle concentration (CMC) of SDS in pure water at 25°C is ≈ 2.36 gl^{-1} (Mukerjee & Mysels, 1971). Previous measurements in the literature have shown that the air-water surface tension exhibits no minimum near the CMC (Lucassen-Reynders *et al.*, 1981). According to Eq. (3.5), this implies that the contact angle also does not exhibit a minimum if the air-solid and water-solid surface tensions remain constant. Here, we examine how the contact angle varies with the SDS concentration at various drop volumes, in the silicon oil-water-methanol system, to determine the surfactant concentration at which the contact angle is minimum for all volumes. The contact angle was measured using the sessile-drop method (Bachmann *et al.*, 2000).

Figure 4.4 shows the surfactant effect on the contact angle. The contact angle decreases almost linearly around the CMC and below ~ 6 gl^{-1} for all drop volumes. At higher concentrations, the relationship is nonlinear. Nevertheless, the smallest value of the contact angle in the range 2–10 gl^{-1}occurs at 10 gl^{-1} for all volumes, except 5 μl. Moreover, the contact angle for larger drops is affected more by the surfactant at this concentration. Since larger drops tend to have a pinned contact line at breakup more than smaller drops, only the stability-limit results for experiments where the SDS concentration is 10 gl^{-1} are reported for all drop

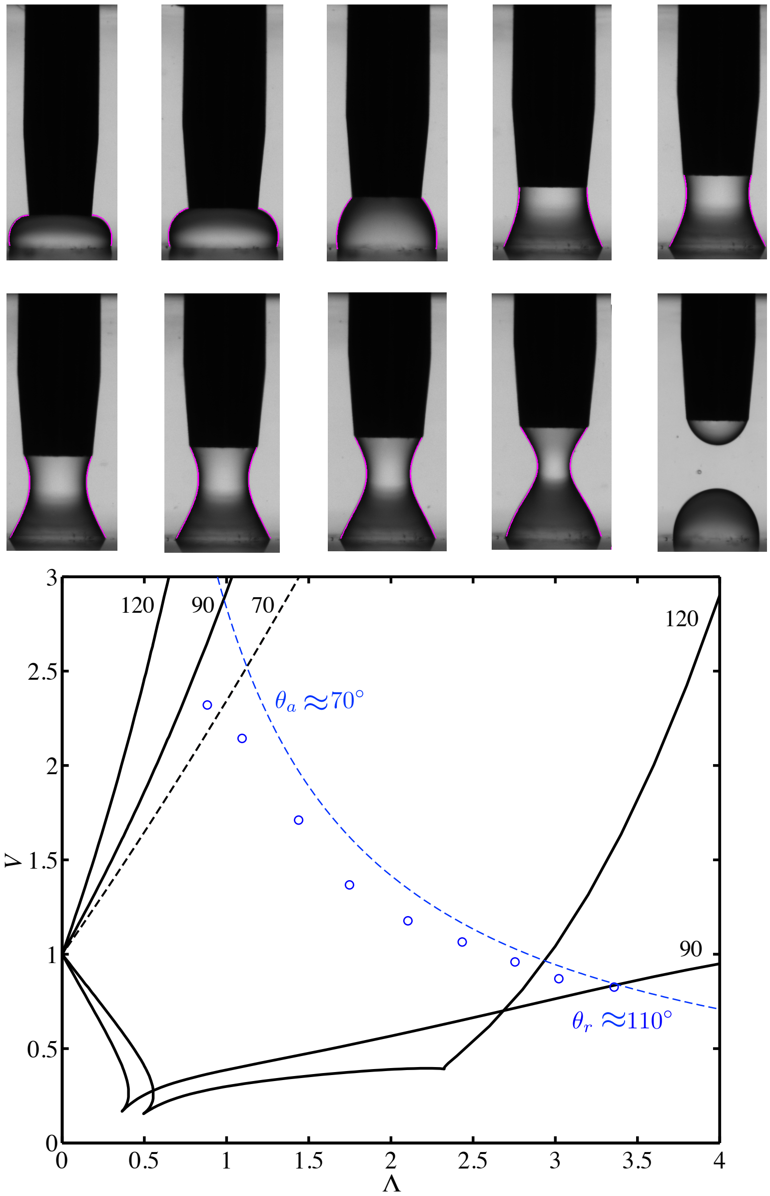


FIGURE 4.5: Comparison of the theoretical prediction and experimental measurement of the stability limits with drop volume $v = 5$ μl, receding contact angle $\theta_r \approx 110°$, advancing contact angle $\theta_a \approx 70°$, and without adding surfactant (bottom). An image sequence of the bridge evolution corresponding to the data points (top). Dashed blue and black lines respectively indicate the constant-v isocontour at the dispensed drop volume and the maximum-volume stability limit estimated by Eq. (3.52) at $\theta_c = 70°$. Labels denote the contact angle in degrees.

volumes. At this concentration, the bridge contact line with the coverslip was moving for all drop volumes.

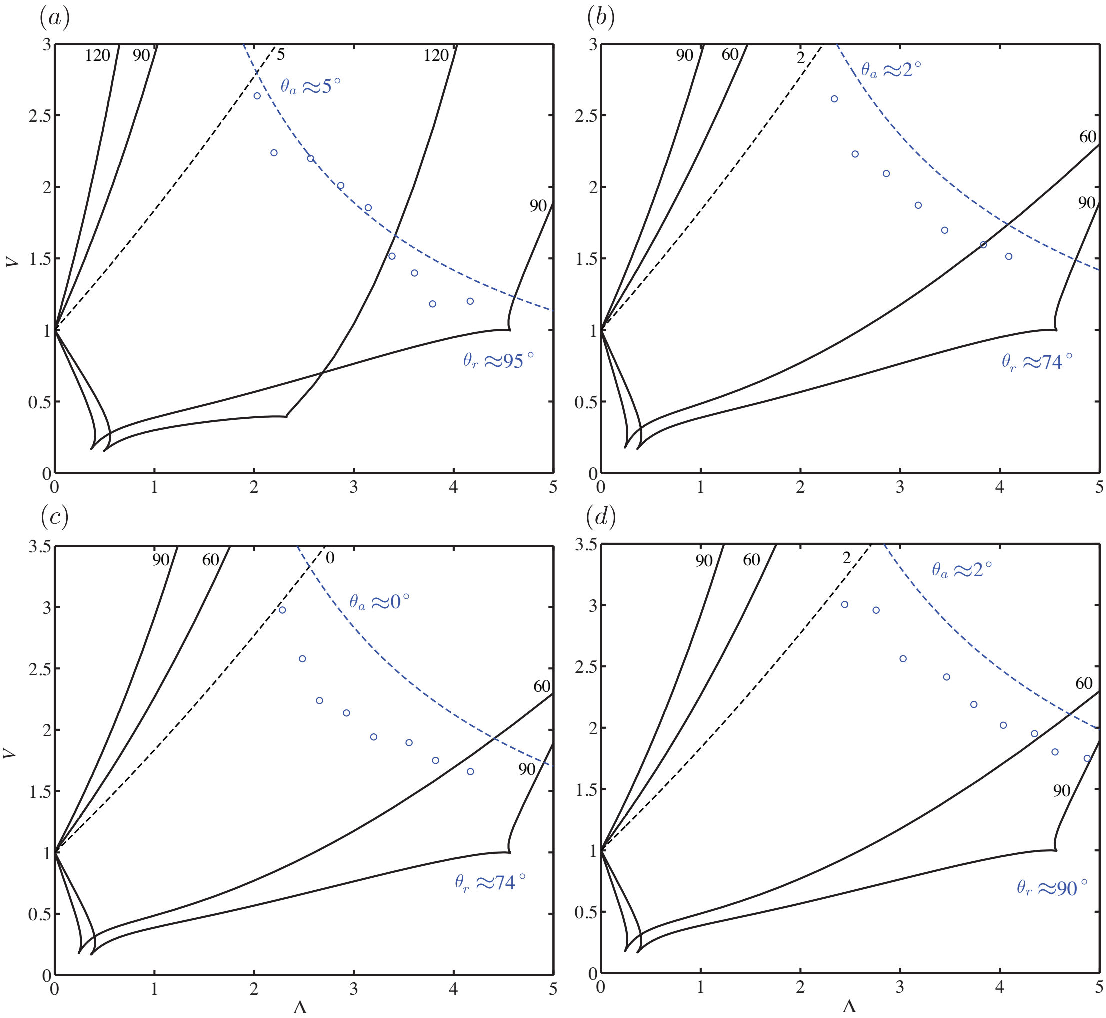


FIGURE 4.6: Comparison of the theoretical prediction and experimental measurement of the stability limits where the SDS concentration is 10 gl^{-1} at drop volumes (a) v = 10 μl, (b) v = 12.5 μl, (c) v = 15 μl, and (d) v = 17.5 μl. Dashed blue and black lines respectively indicate constant-v isocontours at the corresponding dispensed drop volume and the maximum-volume stability limit estimated by Eq. (3.52) at the corresponding advancing contact angle. Labels denote the contact angle in degrees.

4.5.2 Stability limits

The maximum-slenderness stability limit is determined by stepwise quasi-static stretching of a liquid bridge with fixed volume. Similarly, the minimum-slenderness stability limit is determined by stepwise quasi-static squeezing of a bridge. In all experiments, the contact line was either pinned or receding when stretching, and always advancing when squeezing. Therefore, when the contact line is moving, the receding contact angle θ_r is the relevant contact angle in stretching, and the advancing contact angle θ_a is the relevant contact angle in squeezing. Figure 4.5

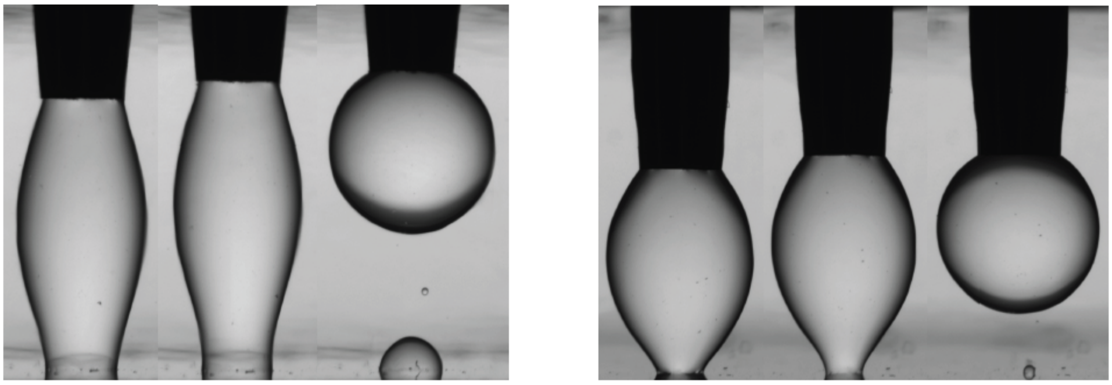


FIGURE 4.7: Contact-line effect on the breakup height of liquid bridges. Stretching experiment on a 20 μl drop with 10 gl^{-1} SDS added to the bath solution, having a pinned contact line with $\theta_c \approx 84°$ (left) and a free contact line with $\theta_c \approx 81°$ (right).

shows a sequence of images recorded during the stretching and squeezing of a 5 μl drop. Here, no surfactant was added to the bath solution, and the contact line moved in the stretching and squeezing. The receding contact angle at breakup was measured $\theta_r \approx 110°$; thus, the corresponding data point (far right) in the stability diagram is expected to fall between the stability-region lower boundary calculated in chapter 3 for $\theta_c = 90°$ and $120°$, as demonstrated in Fig. 4.5. The advancing contact angle was measured $\theta_a \approx 70°$; here, the minimum-slenderness stability limit is estimated by Eq. (3.52) and then compared with the measured value. As shown in Fig. 4.5, experimental measurements of the stability limits are in good agreement with the theoretical predictions presented in chapter 3. Note that, because the needle is hollow, part of the initial drop volume is driven into the needle in squeezing experiments. Therefore, the bridge volume does not reflect the initial drop volume; moreover, data points deviate more from the constant-v isocontour corresponding to the initial dispensed volume (dashed blue line) near the upper boundary of the stability-region.

Stretching and squeezing experiments were conducted in the range v = 5–20 μl with SDS added to the bath solution. At 10 gl^{-1} SDS, the advancing and receding contact angles drop to $\theta_a \approx$ 0–5° and $\theta_r \approx$ 75–95°. Reasonable agreement is observed between the experimental measurements of the stability limits and theoretical predictions of chapter 3 (see Fig. 4.6).

Figure 4.7 shows the effect of a free contact line on the maximum-slenderness stability limit. Here, a stretching experiment was conducted using a 20 μl drop with 10 gl^{-1} SDS added to the bath solution on two coverslips. These coverslips

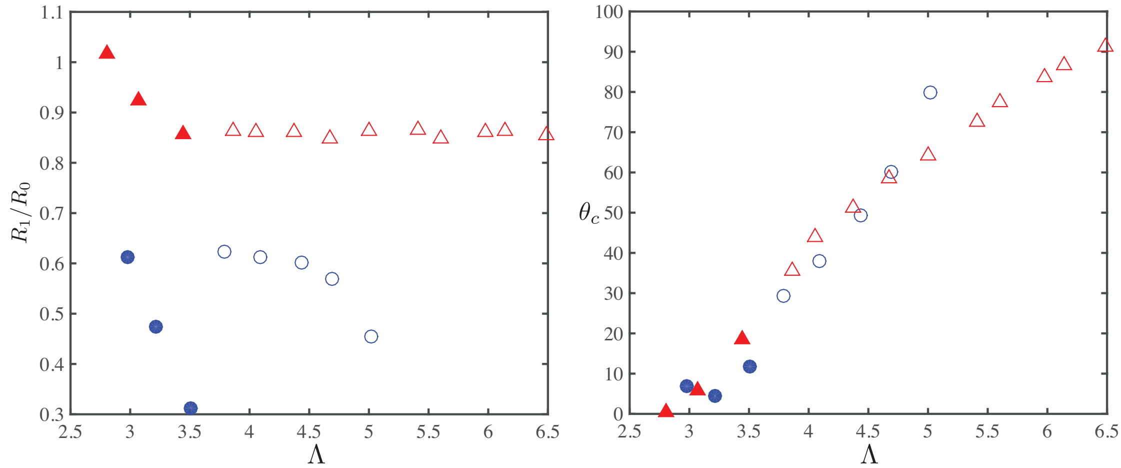


FIGURE 4.8: Squeezing (filled markers) and stretching (hollow markers) of a 20 μl drop with 10 gl^{-1} SDS added to the bath solution, corresponding to the experiments shown in Fig. 4.7. The radius of the meniscus contact line R_1 (left) and contact angle θ_c (right) versus slenderness are plotted when the bridge contact-line on the coverslip at breakup is free (○, right panel in Fig. 4.7) and pinned (△, left panel in Fig. 4.7).

exhibited slightly different contact angles at breakup, presumably due to different surface characteristics, such that θ_c was below θ_r at breakup on one (Fig. 4.7, left panel) and θ_c reached θ_r before breakup on the other (Fig. 4.7, right panel); consequently, the contact line was pinned on the former and free on the latter, and the slendernesses at breakup were measured $\Lambda_b \approx 5.99$ and $\Lambda_b \approx 4.94$, respectively. This $\sim 20\%$ decrease in the breakup height reflects the destabilizing effect of free contact lines, as theoretically shown for catenoidal and cylindrical liquid bridges in chapter 2.

Figure 4.8 shows the radius of the meniscus contact line R_1 and contact angle θ_c in the experiments depicted in Fig. 4.7. For the liquid bridge shown in the left panel of Fig. 4.7, the contact line is pinned, and the contact angle varies during stretching. Here, the contact angle remains below the receding contact angle in the entire experiment. In contrast, for the liquid bridge shown in the right panel, the contact angle reaches the receding contact angle before breakup, and the contact line recedes during stretching. Here, the contact-line motion with stretching the bridge is significantly larger than that in the left panel, indicating that the contact line on the coverslip shown in the left panel of Fig. 4.7 is more constrained than that in the right panel.

Chapter 5

Stability and folds in an elastocapillary system

5.1 Preface

An elastocapillary model is developed in this chapter to study the coupling of elastic and capillary forces in systems where conformations are controlled by the liquid content. Stability criteria are rigorously derived using spectral and variational methods, accounting for interactions between meniscus and elastic-structure perturbations. Here, menisci are axisymmetric simply connected and doubly connected capillary surfaces. This chapter elucidates whether the stability criteria applied in chapters 2 and 3 are sufficient to determine the stability of the elastocapillary system as a whole. The model can also be applied to structural failures that arise in the fabrication of microelectromechanical systems during the wet etching and drying steps.

5.2 Abstract

We examine the equilibrium and stability of an elastocapillary system to model drying-induced structural failures. The model comprises a circular elastic membrane with a hole at the center that is deformed by the capillary pressure of simply connected and doubly connected menisci. Using variational and spectral methods, stability is related to the slope of equilibrium branches in the liquid content versus

pressure diagram for the constrained and unconstrained problems. The second-variation spectra are separately determined for the membrane and meniscus, showing that the membrane out-of-plane spectrum and the in-plane spectrum at large elatocapillary numbers are both positive, so that only meniscus perturbations can cause instability. At small elastocapillary numbers, the in-plane spectrum has a negative eigenvalue, inducing wrinkling instabilities in thin membranes. In contrast, the smallest eigenvalue of the meniscus spectrum always changes sign at a pressure turning point where stability exchange occurs in the unconstrained problem. We also examine configurations in which the meniscus and membrane are individually stable, while the elastocapillary system as a whole is not; this emphasizes the connection between stability and the coupling of elastic and capillary forces.

5.3 Introduction

Elastic deformations induced by capillary forces have been identified as leading causes of pattern collapse in miniature electronic devices and sensors (Chandra & Yang, 2009, Farshid-Chini & Amirfazli, 2010). These microstructures are more prone to collapse when miniaturized because adhesion and capillary forces become comparable to elastic forces when there is a high surface area to volume ratio (Chandra & Yang, 2010). Capillary driven collapse poses major problems for micro-fabrication techniques that are based on wet etching. In particular, microelectronic systems are commonly fabricated through wet lithography where structures often experience permanent deformation and stiction upon drying, significantly limiting the design and operating conditions (Roman & Bico, 2010).

Elastocapillary systems have been extensively studied over the past two decades (Bico *et al.*, 2004, Duprat *et al.*, 2012, Giomi & Mahadevan, 2012, Mastrangelo & Hsu, 1993*a*, Pokroy *et al.*, 2009). Aggregation, coalescence, and self-assembly of filaments and flexible fibres (Boudaoud *et al.*, 2007, Cohen & Mahadevan, 2003, Kim & Mahadevan, 2006, Pokroy *et al.*, 2009), failure of microelectronic devices (Mastrangelo & Hsu, 1993*b*, Raccurt *et al.*, 2004), and capillary wrinkling of elastic membranes (Huang *et al.*, 2007, Vella *et al.*, 2010) are applications where system configurations and structures are determined by elastic-capillary force interactions. The foregoing studies are mostly concerned with systems where equilibrium configurations are always stable (or assumed to be). However, mechanical stability is central to applications in which preventing structural failure upon drying is crucial.

Recently, a few studies have focused on the mechanical stability of elastocapillary systems. Giomi & Mahadevan (2012) examined the equilibrium and stability of minimal surfaces spanning deformable frames. Subjecting a circular frame to spatial perturbations, they approximated instability modes and the critical elastocapillary numbers corresponding to the primary and secondary buckling of the frame into elliptical and twisted structures. Taroni & Vella (2012) identified multiple equilibria in an elastocapillary system related to the aggregation of paint-brush bristles where the stable solutions for a given liquid content were determined through a temporal stability and dynamic analysis.

Mastrangelo & Hsu (1993*a*) took a different approach to determine stability in elastocapillary systems. Their approach hinges on a pervasive theory, known as catastrophe theory (Arnold, 1992) in nonlinear dynamics, stating that stability exchanges only occur at folds and branch points on equilibrium branches (Seydel, 2009). While this has not been generally proved for all mechanical systems, the idea has been extensively examined for purely capillary (Akbari *et al.*, 2015*a*,*c*, Myshkis *et al.*, 1987, Slobozhanin *et al.*, 1997, Vogel, 1989) and purely elastic (Thompson & Hunt, 1984) problems. In this context, Maddocks (1987) established a theory for systems where equilibria are described by a continuous functional of a single function with prescribed boundary conditions (from a Hilbert space). This theory relates the stability of constrained and unconstrained variational problems to the shape of equilibrium branches with no branch point where stability exchanges occur only at simple folds.

Determining the stability of an elastic structure deformed by the Laplace pressure or contact line force of a meniscus is more challenging than determining its equilibria. Elastic and capillary parts for equilibrium states can be decoupled and determined separately by imposing the proper boundary conditions where the meniscus and structure meet. However, the stability of elastic and capillary parts alone is not sufficient to deduce the stability of elastocapillary systems for which the control-parameter role is particularly important.

In the present work, we study an elastocapillary problem where conformations are controlled by the liquid content, similarly to Kwon *et al.* (2008). This is analogous to the problem of disconnected free surfaces considered by Slobozhanin (1983).

Complexities in elastocapillary problems arise because meniscus perturbations are neither pressure controlled nor volume controlled, as in purely capillary problems (Akbari *et al.*, 2015*c*, Lowry & Steen, 1995). Moreover, elastic structures and menisci in many practical elastocapillary systems have free boundaries (Bico *et al.*, 2004, Duprat *et al.*, 2012, Roman & Bico, 2010, Taroni & Vella, 2012), which considerably complicate the stability analysis. Vogel (2000) highlights two major difficulties for analyzing the quadratic forms arising from the second variation of systems with free boundaries: (i) The function space of perturbations is not necessarily a symmetric Hilbert space $\mathscr{H}^0$. Instead, the quadratic forms are naturally expressed in $\mathscr{H}^1$, and an additional analysis is required to link the arising operators to the corresponding operators in a symmetric $\mathscr{H}^0$ space[1]. (ii) Perturbed surfaces resulting from normal variations of an equilibrium surface are not generally guaranteed to satisfy the boundary conditions at the free boundaries.

In this paper, we examine the elastocapillary system shown in Fig. 5.1 and relate stability to the slope of equilibrium branches in pressure versus volume diagrams, similarly to Maddocks (1987). This system is a model for drying-induced structural failures arising in practical applications, such as the stiction of micro-machined sensors and collapse of wood fibres upon drying. It comprises a circular elastic membrane, with a hole at the center, anchored above a rigid plate, trapping a prescribed volume of liquid. We examine membrane deformations caused by a meniscus at the hole as the liquid is slowly removed. Our approach is variational, so that linear stability is determined by the sign of the second variation. We demonstrate that there are configurations in which the meniscus and membrane are individually stable, while the elastocapillary system as a whole is not. This emphasizes the significance of instabilities arising from the coupling of elastic and capillary forces. This result can be interpreted as the equivalent of the Weierstrass–Erdmann condition (Gelfand & Fomin, 2000) for the second variation, and it is relevant to applications where extrema are represented by non-smooth functions, such as for elastocapillary systems, threshold phenomena (Clarke, 1990), and data visualization (Mumford & Shah, 1989).

[1]Note that $\mathscr{H}^k = W^{k,2}$ denotes Sobolev spaces equipped with the Euclidean norm (Adams & Fournier, 2003). Moreover, throughout the paper, 'symmetric space' refers to a space in which all bilinear forms are symmetric.

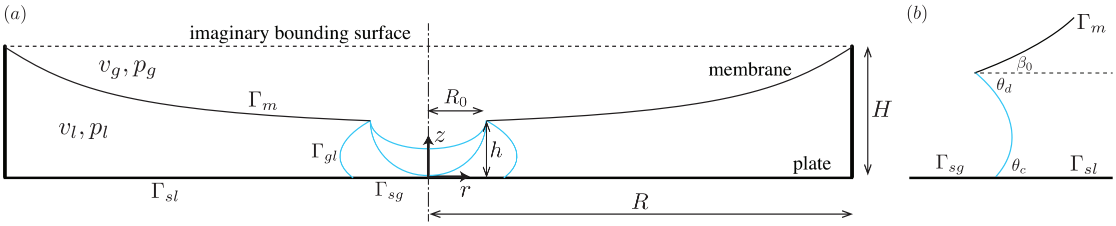


FIGURE 5.1: Elastocapillary model; (a) schematic showing simply connected meniscus (top), doubly connected meniscus (bottom), transition from simply to doubly connected meniscus (middle), and (b) contact angles.

5.4 Formulation

We consider an elastocapillary model comprising a circular elastic membrane with a hole at the center supported on the sidewall of a cylindrical cavity with rigid walls, trapping a liquid volume v_l below the membrane and air volume v_g between the bounding surface (dashed line in Fig. 5.1(a)) and membrane, as shown in Fig. 5.1. The cavity is open to the atmosphere from the top. A meniscus forms at the hole as the liquid is removed, resulting in a difference between the liquid pressure p_l and atmospheric pressure p_g, which causes the membrane to deform. Here, the membrane radius R, hole radius R_0, and cylinder hight H are the model length scales that control the interplay between elastic and capillary forces. To determine the equilibria at a given v_l, we consider an imaginary bounding surface that covers the cavity from the top. The system is isolated from the surrounding by the bounding surface and cylinder walls, also preventing energy and mass transfer. The meniscus is initially a bubble, which can bridge the gap upon contact with the plate at the bottom of the cylinder, forming a free contact line with the plate. Assuming that all the dimensions are small compared to the capillary length, the gravity force is neglected. The membrane and meniscus are assumed axisymmetric in equilibrium and perturbed configurations.

5.4.1 Variational principle

Following the development of Neumann *et al.* (2012), we apply a variational principle to determine stable equilibria for the system depicted in Fig. 5.1. Noting that the liquid volume is the control parameter in drying, the total internal energy is to be minimized subject to $v_l = \text{const.}$, maintaining fixed total entropy and mass. Here, the grand canonical potential is a suitable free-energy representation because it restricts the minimization to states that are already in thermal (constant uniform temperature) and chemical (constant uniform chemical potential) equilibrium. The grand canonical potentials for bulk phases and interfaces, respectively, are (Neumann *et al.*, 2012)

$$\omega^{(v)} = u^{(v)} - Ts^{(v)} - \mu\rho^{(v)} = -p_v, \qquad v = l, g, \tag{5.1}$$

$$\omega^{(a)} = u^{(a)} - Ts^{(a)} - \mu\rho^{(a)} = \gamma_a, \qquad a = gl, sl, sg \tag{5.2}$$

with γ, ω, u, s, and ρ the surface tension, specific grand canonical potential, specific internal energy, specific entropy, and density of the respective phase. The superscripts (v) and (a) denote volume density and area density for bulk phases and interfaces. Note that the temperature T and chemical potential μ can be regarded as the Lagrange multipliers associated with the entropy and mass in the foregoing constrained minimization of the total internal energy. Hence, stable equilibria minimize

$$E_t = \omega^{(g)} v_g + \omega^{(l)} v_l + \omega^{(gl)} \Gamma_{gl} + \omega^{(sl)} \Gamma_{sl} + \omega^{(sg)} \Gamma_{sg} + \Omega^{(m)}, \tag{5.3}$$

where Γ_{ij} are interfacial surface areas. Here, the membrane strain energy $\Omega^{(m)}$ is separately incorporated into the total energy E_t to account for variable and anisotropic stresses. Neglecting the bending energy, we only consider the stretching part of the elastic strain energy in von Kármán's theory for moderately large deflections

$$\Omega^{(m)} = \frac{1}{2} \int_{\Gamma_{m0}} (N_{rr}\varepsilon_{rr} + N_{tt}\varepsilon_{tt}) \mathrm{d}A, \tag{5.4}$$

where Γ_{m0}, N_{ii}, and ε_{ii} are the membrane in the referential configuration (undeflected state), axial forces, and nonlinear strains (Timoshenko *et al.*, 1959). Substituting Eqs. (5.1), (5.2), and (5.4) into Eq. (5.3) and omitting additive and multiplicative constants that do not affect the minimization,

$$E_t[r, u, w, P] = U[r, u, w] - PJ[r, u, w], \quad E_t : L^2 \times L^2 \times L^2 \times \mathbb{R} \to \mathbb{R} \tag{5.5}$$

is the functional to be minimized subject to $v_l = \text{const.}$, where

$$U[r, u, w] = \int_{z_0}^{h} F(z, r, r')\mathrm{d}z + \int_{R_{00}}^{R} G(r_p, u, w, u', w')\mathrm{d}r_p + \frac{R_1^2}{2}(\gamma_{sg} - \gamma_{sl}), \tag{5.6}$$

$$J[r, u, w] = \int_{z_0}^{h} K(z, r, r')\mathrm{d}z + \int_{R_{00}}^{R} M(r_p, u, w, u', w')\mathrm{d}r_p + \frac{R_0^2 h}{2} \tag{5.7}$$

with integrands

$$F(z, r, r') = \gamma_{gl} r\sqrt{1 + r'^2}, \tag{5.8}$$

$$K(z, r, r') = -\frac{r^2}{2}, \tag{5.9}$$

$$G(r_p, u, w, u', w') = \frac{C}{2}\left(u'^2 + u'w'^2 + \frac{2\nu uu'}{r_p} + \frac{\nu uw'^2}{r_p} + \frac{u^2}{r_p^2} + \frac{w'^4}{4}\right) r_p, \tag{5.10}$$

$$M(r_p, u, w, u', w') = uH + (r_p + u)H + (r_p + u)(H + w)u'. \tag{5.11}$$

Note that C, ν, u, w, R_1, and R_{00} are the membrane axial rigidity, Poisson ratio, membrane in-plane displacement[2], membrane deflection, radius of the meniscus contact line with the plate, and hole radius in the referential configuration. Here, primes denote derivatives with respect to the function argument, $P = p_l - p_g$ can be regarded as the Lagrange multiplier associated with the constant v_l constraint, and the menisci are represented by $r(z)$. The membrane deformations are represented by $u(r_p)$ and $w(r_p)$, where r_p is the radial coordinate in the referential configuration. When the meniscus is a bubble (simply connected), the last term in Eq. (5.6) is zero and $z_0 = \ell$, where the meniscus intersects the symmetry axis. When the meniscus is a bridge (doubly connected), $z_0 = 0$, where the free contact line rests on the plate.

5.4.2 Equilibrium from first variation

To construct the increment of E_t in Eq. (5.5) with respect to axisymmetric perturbations, perturbed states are represented by

$$z(\hat{z}) = \hat{z} + \eta_1(\hat{z})\varepsilon + \eta_2(\hat{z})\varepsilon^2, \qquad \eta_1, \eta_2 : [\hat{z}_0, \hat{h}] \to \mathbb{R}, \tag{5.12}$$

$$r(\hat{z}) = \hat{r}(\hat{z}) + \xi_1(\hat{z})\varepsilon + \xi_2(\hat{z})\varepsilon^2, \qquad \hat{r}, \xi_1, \xi_2 : [\hat{z}_0, \hat{h}] \to \mathbb{R}, \tag{5.13}$$

$$u(r_p) = \hat{u}(r_p) + \phi_1(r_p)\varepsilon + \phi_2(r_p)\varepsilon^2, \qquad \hat{u}, \phi_1, \phi_2 : [R_{00}, R] \to \mathbb{R}, \tag{5.14}$$

$$w(r_p) = \hat{w}(r_p) + \psi_1(r_p)\varepsilon + \psi_2(r_p)\varepsilon^2, \qquad \hat{w}, \psi_1, \psi_2 : [R_{00}, R] \to \mathbb{R}, \tag{5.15}$$

accounting for the linear and nonlinear parts of the increment when the meniscus is a bridge. Note that the form of the functionals in Eqs. (5.6) and (5.7) demands $\hat{r}', \hat{u}', \hat{w}'$ to be continuous, so $\hat{r}, \hat{u}, \hat{w} \in L^2 \cap C^1$. Here, equilibrium and perturbed states are denoted by hatted and unhatted variables, respectively. Equation (5.12)

[2] Not to be confused with the specific internal energy in Eqs. (5.1) and (5.2).

is particularly important, because it admits perturbations that can displace the position of the bubble apex and hole edge along the z-axis. We impose a simply supported boundary condition for the membrane at $r_p = R$ where $u, w = 0$ (Timoshenko *et al.*, 1959), resulting in

$$\phi_1(R) = \phi_2(R) = \psi_1(R) = \psi_2(R) = 0. \tag{5.16}$$

The meniscus is assumed to be pinned to the hole edge at $r_p = R_{00}$, where $r(\hat{h})|_{\Gamma_{gl}} = r(R_{00})|_{\Gamma_m}$ and $z(\hat{h})|_{\Gamma_{gl}} = z(R_{00})|_{\Gamma_m}$, furnishing

$$\hat{R}_0 = R_{00} + \hat{u}(R_{00}), \quad \xi_1(\hat{h}) = \phi_1(R_{00}), \quad \xi_2(\hat{h}) = \phi_2(R_{00}), \tag{5.17}$$

$$\hat{h} = H + \hat{w}(R_{00}), \quad \eta_1(\hat{h}) = \psi_1(R_{00}), \quad \eta_2(\hat{h}) = \psi_2(R_{00}). \tag{5.18}$$

Moreover, the meridian curve intersects the symmetry axis at $\hat{z} = \hat{\ell}$, where it can only move vertically when the meniscus is a bubble, so

$$\xi_1(\hat{\ell}) = \xi_2(\hat{\ell}) = 0, \quad \eta_1(\hat{\ell}), \eta_2(\hat{\ell}) = \text{finite}, \tag{5.19}$$

whereas the contact line with the plate can only move horizontally at $\hat{z} = 0$ when the meniscus is a bridge, so

$$\xi_1(0), \xi_2(0) = \text{finite}, \quad \eta_1(0) = \eta_2(0) = 0. \tag{5.20}$$

Since the domain of $r(z)$ is variable in Eqs. (5.6) and (5.7), the functional variations are properly represented with respect to the barred component of ξ (see Gelfand & Fomin (2000) for details)

$$\xi_1 = \bar{\xi}_1 + \eta_1 \hat{r}', \tag{5.21}$$

$$\xi_2 = \bar{\xi}_2 + \eta_2 \hat{r}' + \eta_1 \xi_1' + \frac{1}{2}\eta_1^2 \hat{r}''. \tag{5.22}$$

Substituting Eqs. (5.12)-(5.15) into Eq. (5.5), the first variation of E_t with respect to an equilibrium state is

$$\begin{aligned} \frac{\delta E_t}{\varepsilon} = \langle U'_{(\hat{r})} - PJ'_{(\hat{r})}, \bar{\xi}_1 \rangle + \langle U'_{(\hat{u})} - PJ'_{(\hat{u})}, \phi_1 \rangle + \langle U'_{(\hat{w})} - PJ'_{(\hat{w})}, \psi_1 \rangle \\ + \left[F_{r'}|_{\hat{h}} - P\hat{R}_0 \hat{h} - G_{u'}|_{R_{00}} + PM_{u'}|_{R_{00}} \right] \xi_1(\hat{h}) \\ + \left[F|_{\hat{h}} - \hat{r}' F_{r'}|_{\hat{h}} - G_{w'}|_{R_{00}} \right] \eta_1(\hat{h}) + \left[\hat{R}_1(\gamma_{sg} - \gamma_{sl}) - F_{r'}|_0 \right] \xi_1(0), \end{aligned} \tag{5.23}$$

where all the functional integrands are evaluated at the equilibrium. Here, primes operating on functionals denote the first Fréchet derivative (Bobylov *et al.*, 1999) with respect to the function in the subscript, and $\langle \cdot, \cdot \rangle$ is the inner product over the domain of the respective function. The last term in Eq. (5.23) must be replaced with $[\hat{r}' F_{r'} - F]_{\hat{\ell}}\, \eta_1(\hat{\ell})$ when the meniscus is a bubble. Equilibria are the stationary points of the total energy where $\delta E_t = 0$ for arbitrary $\bar{\xi}_1$, ϕ_1, and ψ_1, requiring

$$U'_{(\hat{r})} - PJ'_{(\hat{r})} = F_r - PK_r - \frac{\mathrm{d}}{\mathrm{d}\hat{z}}\left(F_{r'} - PK_{r'}\right) = 0, \tag{5.24}$$

$$U'_{(\hat{u})} - PJ'_{(\hat{u})} = G_u - PM_u - \frac{\mathrm{d}}{\mathrm{d}r_p}\left(G_{u'} - PM_{u'}\right) = 0, \tag{5.25}$$

$$U'_{(\hat{w})} - PJ'_{(\hat{w})} = G_w - PM_w - \frac{\mathrm{d}}{\mathrm{d}r_p}\left(G_{w'} - PM_{w'}\right) = 0, \tag{5.26}$$

with each boundary term in square brackets equal to zero. Substituting the integrands from Eqs. (5.8)-(5.11) furnishes

$$\frac{\hat{r}''}{(1+\hat{r}'^2)^{3/2}} - \frac{1}{\hat{r}(1+\hat{r}'^2)^{1/2}} = \frac{P}{\gamma_{gl}}, \tag{5.27}$$

$$r_p \hat{N}'_{rr} + \hat{N}_{rr} - \hat{N}_{tt} - P(r_p + \hat{u})\hat{w}' = 0, \tag{5.28}$$

$$(\hat{N}_{rr}\hat{w}' r_p)' + P(r_p + \hat{u})(1 + \hat{u}') = 0 \tag{5.29}$$

with boundary conditions

$$\gamma_{gl}\hat{R}_0 \cos\theta_d = -\hat{N}_{rr}(R_{00})R_{00} \quad \text{at} \quad \hat{z} = \hat{h}, \tag{5.30}$$

$$\gamma_{gl}\hat{R}_0 \sin\theta_d = \hat{N}_{rr}(R_{00})\hat{w}'(R_{00})R_{00} \quad \text{at} \quad \hat{z} = \hat{h}, \tag{5.31}$$

$$\gamma_{sg} - \gamma_{sl} = \gamma_{gl}\cos\theta_c \quad \text{at} \quad \hat{z} = 0. \tag{5.32}$$

Here, θ_c and θ_d are the contact and dihedral angles that the interface Γ_{gl} forms with the plate and membrane, respectively. The last boundary condition only holds when the meniscus is a bridge, while the boundary term associated with $\eta_1(\hat{\ell})$ when the meniscus is a bubble is always zero with $\hat{r}'(\hat{\ell}) \to \infty$. Equation (5.27) is the Young-Laplace equation, and Eqs. (5.28) and (5.29) are the in- and out-of-plane equations of equilibrium for membranes in von Kármán's theory with the capillary pressure acting normal to the neutral plane. Note that Eqs. (5.30) and (5.31) demand $\hat{w}'(R_{00})\hat{r}'(\hat{h}) = 1$, implying $\hat{w}' \to \infty$ as $\theta_d \to \pi/2$, contradicting the assumptions of von Kármán's theory (Timoshenko *et al.*, 1959). To resolve this

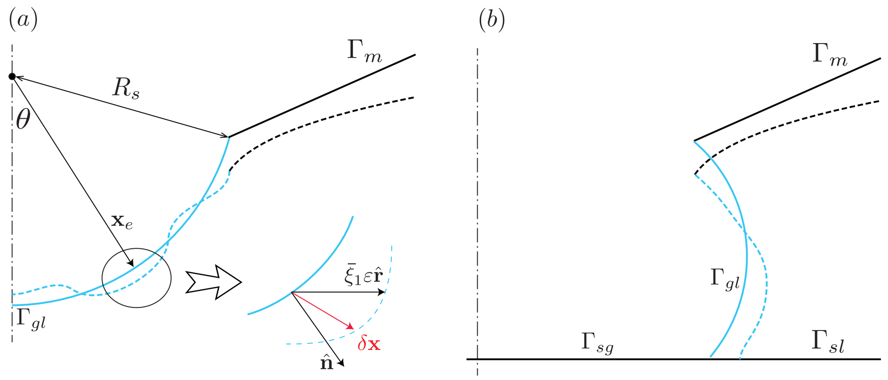


FIGURE 5.2: Schematic of perturbations to (a) simply connected (bubble), and (b) doubly connected (bridge) menisci.

issue, we undertake a scaling analysis in section 5.5 to simplify Eqs. (5.30) and (5.31).

5.4.3 Stability from second variation

The Jacobian matrices of the functional integrants in Eqs. (5.6) and (5.7) are not symmetric. Since analyzing the second variation for functionals of multiple functions with non-symmetric Jacobians is intractable (Gelfand & Fomin, 2000), we simplify the problem by neglecting in-plane variations at the hole edge, prescribing $\phi_i(R_{00}) = 0$. Accordingly, the second variation is

$$\frac{2\delta^2 E_t}{\varepsilon^2} = \int_{\hat{z}_0}^{\hat{h}} \left(\mathcal{P}^{(\hat{r})}\bar{\xi}_1'^2 + \mathcal{Q}^{(\hat{r})}\bar{\xi}_1^2\right) \mathrm{d}\hat{z} + \int_{R_{00}}^{R} \left(\mathcal{P}^{(\hat{u})}\phi_1'^2 + \mathcal{Q}^{(\hat{u})}\phi_1^2\right) \mathrm{d}r_p + \int_{R_{00}}^{R} \left(\mathcal{P}^{(\hat{w})}\psi_1'^2 + \mathcal{Q}^{(\hat{w})}\psi_1^2\right) \mathrm{d}r_p + [\mathcal{A}\bar{\xi}_1^2]_{\hat{h}} - [\mathcal{A}\bar{\xi}_1^2]_{\hat{z}_0}, \quad (5.33)$$

where $\mathcal{F} = F - PK$, $\mathcal{G} = G - PM$, and

$$\mathcal{P}^{(\hat{r})} = \mathcal{F}_{r'r'}, \quad \mathcal{Q}^{(\hat{r})} = \mathcal{F}_{rr} - \frac{\mathrm{d}}{\mathrm{d}\hat{z}}\mathcal{F}_{rr'}, \quad (5.34)$$

$$\mathcal{P}^{(\hat{u})} = \mathcal{G}_{u'u'}, \quad \mathcal{Q}^{(\hat{u})} = \mathcal{G}_{uu} - \frac{\mathrm{d}}{\mathrm{d}r_p}\mathcal{G}_{uu'}, \quad (5.35)$$

$$\mathcal{P}^{(\hat{w})} = \mathcal{G}_{w'w'}, \quad \mathcal{Q}^{(\hat{w})} = \mathcal{G}_{ww} - \frac{\mathrm{d}}{\mathrm{d}r_p}\mathcal{G}_{ww'}, \quad (5.36)$$

$$\mathcal{A} = \mathcal{F}_{rr'} - \frac{\mathcal{F}_r}{\hat{r}'}. \quad (5.37)$$

Note that the boundary term at $\hat{z}_0$ in Eq. (5.33) arises only when the meniscus is a bubble, so $\delta^2 E_t$ has only one boundary term at $\hat{h}$ when the meniscus is a bridge. Furthermore, when the meniscus is a bubble, $\bar{\xi}_1$ is not bounded due to the axial symmetry condition at $\hat{z}_0 = \hat{\ell}$ (see Appendix C), and it is unsuitable for representing quadratic forms. To resolve this issue, using the mapping $(\hat{z}, \bar{\xi}_1) \to (y, N)$, the first integral and boundary terms in Eq. (5.33) are represented with respect to the normal variation $N(y)$, where $y = \cos\theta$ (see Fig. 5.2(a) and Appendix C). Equation (5.33) then reduces to

$$\begin{aligned} Q = \gamma_{gl} \int_{y_0}^{1} \left[(1-y^2)N'^2 + 2yNN' - N^2\right] \mathrm{d}y + \gamma_{gl} \frac{N^2(y_0)}{y_0} - \gamma_{gl} N^2(1) \\ + \int_{R_{00}}^{R} \left[r_p C \phi_1'^2 + \left(\frac{C}{r_p} + P\hat{w}'\right)\phi_1^2\right] \mathrm{d}r_p + \int_{R_{00}}^{R} (\hat{N}_{rr} + C\hat{w}'^2) r_p \psi_1'^2 \mathrm{d}r_p, \end{aligned} \tag{5.38}$$

$$\begin{aligned} Q = \int_{0}^{\hat{h}} \left[\mathcal{P}^{(\hat{r})} \bar{\xi}_1'^2 + \mathcal{Q}^{(\hat{r})} \bar{\xi}_1^2\right] \mathrm{d}\hat{z} + \left[\mathcal{A}\bar{\xi}_1^2\right]_{\hat{h}} \\ + \int_{R_{00}}^{R} \left[r_p C \phi_1'^2 + \left(\frac{C}{r_p} + P\hat{w}'\right)\phi_1^2\right] \mathrm{d}r_p + \int_{R_{00}}^{R} (\hat{N}_{rr} + C\hat{w}'^2) r_p \psi_1'^2 \mathrm{d}r_p, \end{aligned} \tag{5.39}$$

for simply connected and doubly connected menisci, respectively. Here, $Q = 2\delta^2 E_t/\varepsilon^2$, $y_0 = \cos\theta_0$ corresponding to the polar angle at the hole edge (Fig. 5.2(a)), and

$$\mathcal{A} = -\frac{\gamma_{gl}\hat{r}\hat{r}''}{\hat{r}'(1+\hat{r}'^2)^{3/2}}, \quad \mathcal{P}^{(\hat{r})} = \frac{\gamma_{gl}\hat{r}}{(1+\hat{r}'^2)^{3/2}}, \quad \mathcal{Q}^{(\hat{r})} = -\frac{\gamma_{gl}}{\hat{r}(1+\hat{r}'^2)^{1/2}}. \tag{5.40}$$

A necessary (sufficient) condition for E_t to have a minimum is that Q be nonnegative (strongly positive) for an equilibrium solution (Gelfand & Fomin, 2000). Here, strong positivity, referred to as nonlinear stability in the literature (Vogel, 1999), must be distinguished from positive-definiteness. According to Vogel (1996), $Q > 0$ does not imply a strict local minimum for constrained infinite-dimensional problems. Nevertheless, defining stable equilibria as those for which $Q > 0$ holds, Maddocks (1987) derived sufficient conditions for the positive-definiteness of the second variation. Interestingly, Vogel (1996) showed that these conditions are sufficient for Madoccks' functional to have a strict minimum. Therefore, following Maddocks (1987), we adopt Madoccks' definition of stability to relate stability to the slope of equilibrium branches.

Our analysis differs from Madoccks' theory in two respects: (i) The elastocapillary energy E_t is a functional of multiple functions, the stationary points of which are represented by non-smooth functions. Moreover, E_t has three separate spectra, which complicates the relationship between stability exchanges and equilibrium-branch turning points. (ii) The boundary condition at the hole edge, where the membrane and meniscus meet, is not prescribed. Here, perturbations are arbitrarily finite, posing the same difficulties as discussed by Myshkis *et al.* (1987) and Vogel (2000) for analyzing the stability of capillary surfaces with free contact lines.

The quadratic forms in Eqs. (5.38) and (5.39) demand $N, \bar{\xi}_1, \phi_1, \psi_1 \in \mathscr{H}^1$, where perturbations satisfy the boundary conditions Eqs. (5.16)-(5.20). Since each bilinear term in Eqs. (5.38) and (5.39) is bounded, Q can be represented as (Akhiezer & Glazman, 1993, Vogel, 2000)

$$Q = \gamma_{gl} \left\langle \bar{\mathcal{L}}_{(\hat{r})} N, N \right\rangle_1 + \left\langle \bar{\mathscr{L}}_{(\hat{u})} \phi_1, \phi_1 \right\rangle_1 + \left\langle \bar{\mathscr{L}}_{(\hat{w})} \psi_1, \psi_1 \right\rangle_1, \tag{5.41}$$

$$Q = \left\langle \bar{\mathscr{L}}_{(\hat{r})} \bar{\xi}_1, \bar{\xi}_1 \right\rangle_1 + \left\langle \bar{\mathscr{L}}_{(\hat{u})} \phi_1, \phi_1 \right\rangle_1 + \left\langle \bar{\mathscr{L}}_{(\hat{w})} \psi_1, \psi_1 \right\rangle_1 \tag{5.42}$$

for simply connected and doubly connected menisci, where $\langle \cdot, \cdot \rangle_1$ is the $\mathscr{H}^1$ inner product (Adams & Fournier, 2003) and $\bar{\mathcal{L}}_{(\hat{r})}, \bar{\mathscr{L}}_{(\hat{r})}, \bar{\mathscr{L}}_{(\hat{u})}, \bar{\mathscr{L}}_{(\hat{w})}$ are uniquely determined by the respective bilinear terms in Eqs. (5.38) and (5.39). Because perturbations are arbitrary at the hole edge, the $\mathscr{H}^1$ spaces from which perturbations are selected are not symmetric, and, thus, $\bar{\mathcal{L}}_{(\hat{r})}, \bar{\mathscr{L}}_{(\hat{r})}, \bar{\mathscr{L}}_{(\hat{u})}, \bar{\mathscr{L}}_{(\hat{w})}$ are not generally self-adjoint. On the other hand, integrating Eqs. (5.38) and (5.39) by parts leads to

$$Q = \gamma_{gl} \left\langle \mathcal{L}_{(\hat{r})} N, N \right\rangle + \left\langle \mathscr{L}_{(\hat{u})} \phi_1, \phi_1 \right\rangle + \left\langle \mathscr{L}_{(\hat{w})} \psi_1, \psi_1 \right\rangle, \tag{5.43}$$

$$Q = \left\langle \mathscr{L}_{(\hat{r})} \bar{\xi}_1, \bar{\xi}_1 \right\rangle + \left\langle \mathscr{L}_{(\hat{u})} \phi_1, \phi_1 \right\rangle + \left\langle \mathscr{L}_{(\hat{w})} \psi_1, \psi_1 \right\rangle \tag{5.44}$$

for simply connected and doubly connected menisci, where $\mathscr{L}_{(\hat{y})} \equiv U''_{(\hat{y})} - PJ''_{(\hat{y})}$ with double primes denoting the second Fréchet derivative and perturbations satisfying the boundary conditions

$$N'(y_0) - N(y_0)/y_0 = 0, \quad N(1) = \text{finite}, \tag{5.45}$$

$$\bar{\xi}'_1(0) = 0, \quad \bar{\xi}'_1(\hat{h}) + [\mathcal{A}/\mathcal{P}^{(\hat{r})}]\bar{\xi}_1(\hat{h}) = 0, \tag{5.46}$$

$$\phi_1(R_{00}) = 0, \quad \phi_1(R) = 0, \tag{5.47}$$

$$\psi_1'(R_{00}) = 0, \quad \psi_1(R) = 0, \tag{5.48}$$

furnishing a symmetric $\mathscr{H}^0$ for evaluating Q. Here,

$$\mathcal{L}_{(\hat{r})}N = -\frac{\mathrm{d}}{\mathrm{d}y}\left[(1-y^2)\frac{\mathrm{d}N}{\mathrm{d}y}\right] - 2N, \tag{5.49}$$

$$\mathscr{L}_{(\hat{r})}\bar{\xi}_1 = -\frac{\mathrm{d}}{\mathrm{d}\hat{z}}\left[\mathcal{P}^{(\hat{r})}\frac{\mathrm{d}\bar{\xi}_1}{\mathrm{d}\hat{z}}\right] + \mathcal{Q}^{(\hat{r})}\bar{\xi}_1, \tag{5.50}$$

$$\mathscr{L}_{(\hat{u})}\phi_1 = -\frac{\mathrm{d}}{\mathrm{d}r_p}\left[r_p C\frac{\mathrm{d}\phi_1}{\mathrm{d}r_p}\right] + \left(\frac{C}{r_p} + P\hat{w}'\right)\phi_1, \tag{5.51}$$

$$\mathscr{L}_{(\hat{w})}\psi_1 = -\frac{\mathrm{d}}{\mathrm{d}r_p}\left[r_p(\hat{N}_{rr} + C\hat{w}'^2)\frac{\mathrm{d}\psi_1}{\mathrm{d}r_p}\right], \tag{5.52}$$

subject to Eqs. (5.45)-(5.48) are regular Sturm-Liouville operators; thus, they are self-adjoint and Fredholm (Walter, 1998). Establishing a relationship between the spectrum of the barred operators on $\mathscr{H}^1$ and those of unbarred operators on $\mathscr{H}^0$ significantly simplifies the analysis, providing a setting to apply the well-studied Sturm-Liouville theory. Lemma 2.5 of Vogel (1999) furnishes this relationship by stating that the barred operators have the same number of negative and non-positive eigenvalues as the corresponding unbarred operators. Moreover, the constant-volume constraint ($v_l = \text{const.}$) can be written

$$\left\langle J'_{(\hat{r})}, \bar{\xi}_1\right\rangle + \left\langle J'_{(\hat{u})}, \phi_1\right\rangle + \left\langle J'_{(\hat{w})}, \psi_1\right\rangle = 0 \tag{5.53}$$

since $\delta v_l = \delta J$.

5.5 Scaling analysis

Starting from Eq. (5.28) and taking $R_{00} \ll R$, we begin by scaling all lengths with R. Given $\hat{N}_{rr}/C = \tilde{u}' + \tilde{w}'^2/2 + \nu\tilde{u}/\tilde{r}_p$ and $\hat{N}_{tt}/C = \nu\tilde{u}' + \nu\tilde{w}'^2/2 + \tilde{u}/\tilde{r}_p$, where $\tilde{u} = \hat{u}/R$, $\tilde{w} = \hat{w}/R$, and $\tilde{r}_p = r_p/R$, we find $\tilde{u} \sim \tilde{w}^2$ to balance all the terms in the in-plane equilibrium. Considering the bending and stretching parts of the strain energy in von Kármán's theory for axisymmetric plates (Timoshenko *et al.*, 1959)

$$\Omega_B = \frac{D}{2}\int_{R_{00}}^{R}\left(w''^2 + \frac{w'^2}{r_p^2} + \frac{2\nu w' w''}{r_p}\right) r_p \mathrm{d}r_p,$$

$$\Omega_S = \frac{C}{2}\int_{R_{00}}^{R}\left(u'^2 + u'w'^2 + \frac{2\nu u u'}{r_p} + \frac{\nu u w'^2}{r_p} + \frac{u^2}{r_p^2} + \frac{w'^4}{4}\right) r_p \mathrm{d}r_p,$$

and noting that $r_p \sim R$, $w \sim H$, $C \sim Eb$, and $D \sim Eb^3$, where D and b are the bending rigidity and plate thickness, one infers $\Omega_B \ll \Omega_S$ when $b/H \ll 1$ by comparing the energy scales $\Omega_B \sim DH^2/R^2$ and $\Omega_S \sim CH^4/R^2$. For thin membranes, this justifies neglecting the bending energy compared to the stretching energy.

Given $N_{rr} \sim C(w/R)^2$, $|\tilde{w}| \sim (|Q_c|/\kappa N_C)^{1/3}$ follows from the out-of-plane equilibrium, where $\kappa = R_{00}/R$, $Q_c = PR_{00}/\gamma_{gl}$, $N_C = C/\gamma_{gl}$ are the scaled hole radius, scaled capillary pressure and elastocapillary number. Noting that $|\tilde{w}| + \kappa\Lambda = \Pi$, the elastocapillary number corresponding to a specific state of the system in Fig. 5.1 can be estimated as $N_C \sim |Q_c|/\kappa(\Pi - \kappa\Lambda)^3$, where $\Lambda = h/R_{00}$ and $\Pi = H/R$ are the meniscus slenderness and aspect ratio. For example, when the meniscus is a bubble contacting the plate, before bridging the gap, as will be discussed elsewhere, $Q_c = -2\sin\theta_d$ and $\Lambda = \cot(\theta_d/2)$; thus, at $\theta_d = \theta_c$, which is the critical dihedral angle below which the bubble cannot be stably pinned to the hole edge (Myshkis *et al.*, 1987), we have

$$N_C \sim \frac{2\sin\theta_c}{\kappa(\Pi - \kappa\cot\theta_c/2)^3}. \tag{5.54}$$

From Eq. (5.30), $\hat{N}_{rr}(R_{00}) \sim \gamma_{gl}$, while $\hat{N}_{rr} \sim C\tilde{w}^2$ in the main body of the membrane. Thus, $\hat{N}_{rr}(R_{00}) \ll \hat{N}_{rr}$ when $N_C \gg 1/\Pi^2$, implying that Eqs. (5.30) and (5.31) can be approximated by a free-edge boundary condition for slender cavities when N_C is large.

5.6 Membrane profile

The stretching part of the strain energy leads to nonlinear equations of equilibrium, which can be solved numerically in most practical problems. Although the case considered in this paper, namely, axisymmetric plate with a hole at the center, has a series solution (Timoshenko *et al.*, 1959), expressions for the unknown coefficients are cumbersome. Therefore, we apply a variational approximation, as commonly used in the literature (Banerjee & Datta, 1981, Mastrangelo & Hsu, 1993*a*), to construct a general solution for the membrane deflection.

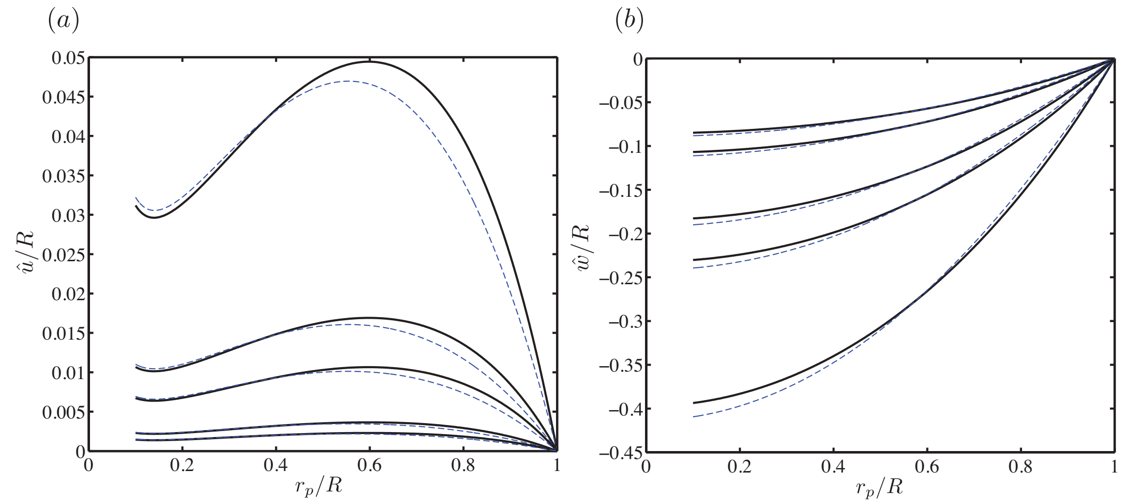


FIGURE 5.3: Comparison of the numerically exact solution (solid) and variational approximation (dashed) of (a) in-plane and (b) out-of-plane displacements: $Q_c = -2$, $\kappa = 0.1$, $\nu = 0.3$, and $N_C = 100, 500, 1000, 5000, 10000$ (downward in (a) and upward in (b)).

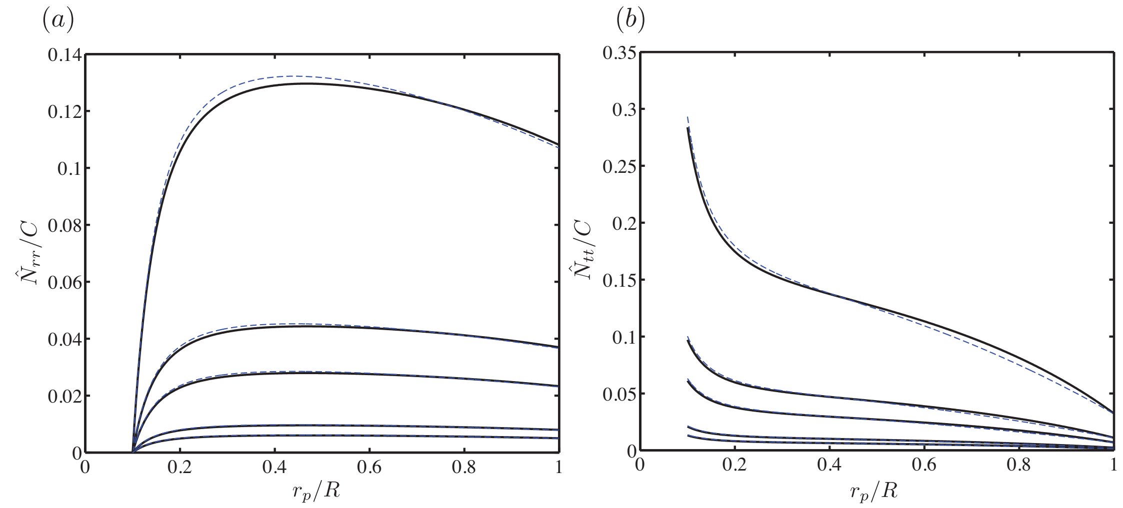


FIGURE 5.4: Comparison of the numerically exact solution (solid) and variational approximation (dashed) of axial forces in the (a) radial and (b) tangential directions: $Q_c = -2$, $\kappa = 0.1$, $\nu = 0.3$, and $N_C = 100, 500, 1000, 5000, 10000$ (downward).

Here, we approximate the boundary condition at $r_p = R_{00}$ as a free edge. Noting that the test function

$$\tilde{w} = \tilde{w}_0(1 - \tilde{r}_p^2) \tag{5.55}$$

is consistent with the boundary conditions and the plate stress distribution under tension at zero deflection (Timoshenko *et al.*, 1959), $\tilde{w}_0 = (Q_c/N_C)^{1/3} K_w$ is obtained by minimizing the stretching energy, where

$$K_w = \left[\frac{3[1-\nu+\kappa^2(1+\nu)]}{\kappa(1-\kappa^2)(1-\nu^2)[7-\nu+\kappa^2(1+\nu)]}\right]^{1/3}, \tag{5.56}$$

which is in agreement with the scaling relation for $\tilde{w}$ derived in section 5.5. Equation (5.55) contrasts with the test function $\tilde{w} = \tilde{w}_0(1-\tilde{r}_p^2)^2$ that Mastrangelo & Hsu (1993*a*) used to describe the bending and stretching contribution to the overall deflection of beams. The approximation accuracy hinges on choosing appropriate test functions for bending- and stretching-dominated regimes. Figures 5.3 and 5.4 demonstrate a reasonable agreement between the variational approximation and numerically exact solutions of the membrane equilibrium, for a wide range of deflections.

5.7 Second variation spectra

We adopt the foregoing variational approximation in this section to determine the second variation spectra corresponding to the in-plane and out-of-plane equilibria.

5.7.1 In-plane spectrum

The spectrum of the membrane in-plane equilibrium is determined by

$$\begin{cases} \mathscr{L}_{(\hat{u})} Z = r_p \vartheta Z \\ Z(R_{00}) = 0, \quad Z(R) = 0, \end{cases} \tag{5.57}$$

where ϑ, Z, and r_p are the eigenvalue, eigenfunction, and weight function of $\mathscr{L}_{(\hat{u})}$. Non-dimensionalizing Eq. (5.57) furnishes

$$\begin{cases} \tilde{r}_p \dfrac{\mathrm{d}}{\mathrm{d}\tilde{r}_p}\left(\tilde{r}_p \dfrac{\mathrm{d}z}{\mathrm{d}\tilde{r}_p}\right) + [(\vartheta^* + B)\tilde{r}_p^2 - 1] = 0 \\ Z(\kappa) = 0, \quad Z(1) = 0, \end{cases} \tag{5.58}$$

with $\vartheta^* = R^2\vartheta/C$ and $B = 2(Q_c/N_C)^{4/3}K_w/\kappa$. The general solution of Eq. (5.57) is $Z(\tilde{r}_p) = C_1\mathrm{J}_1(m\tilde{r}_p) + C_2\mathrm{Y}_1(m\tilde{r}_p)$ with $m = \sqrt{B+\vartheta^*}$, where J_1 and Y_1 are the Bessel functions of first and second kind. A similar eigenvalue problem was derived by Timoshenko & Gere (2009) for the buckling of circular plates under in-plane

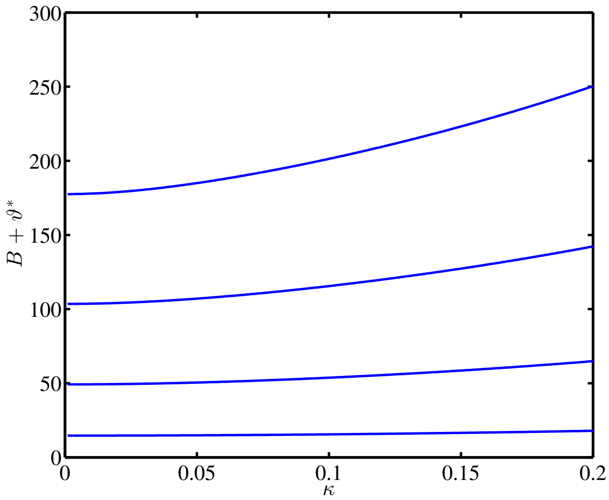


FIGURE 5.5: In-plane spectrum ϑ_i^* with $i = 0, 1, 2, 3$ (upward).

compressive loads. Imposing the boundary conditions furnishes

$$Z_i(\tilde{r}_p) = C_{1,i}\left[\mathrm{J}_1(m_i\tilde{r}_p) - \frac{\mathrm{J}_1(m_i\kappa)}{\mathrm{Y}_1(m_i\kappa)}\mathrm{Y}_1(m_i\tilde{r}_p)\right], \tag{5.59}$$

$$\vartheta_i^* = m_i^2 - B, \quad i = 0, 1, 2, \cdots \tag{5.60}$$

where m_i is the ith root of

$$\mathrm{J}_1(m)\mathrm{Y}_1(m\kappa) - \mathrm{J}_1(m\kappa)\mathrm{Y}_1(m) = 0. \tag{5.61}$$

From Eq. (5.60), unless $B = m_0^2$ at a given κ, Z_0 corresponding to $\vartheta_0^* = 0$ has only a trivial solution where $\ker(\mathscr{L}_{(\hat{u})}) = \{\mathbf{0}\}$ and $\dot{\hat{u}} = \mathbf{0}$ at $\dot{P} = 0$. Here, $\mathbf{0}$ is the identically zero function, and the overdot denotes differentiation along equilibrium branches. Therefore, stability loss due to in-plane perturbations is not generally related to pressure turning points. Studying these instabilities, which are responsible for wrinkling in thin elastic membranes (Coman & Bassom, 2007, Davidovitch *et al.*, 2011, Piñeirua *et al.*, 2013), is beyond the scope of the present work and will not be further elaborated upon.

Figure 5.5 shows the first four eigenvalues of the in-plane spectrum. Note that $m_0^2 \approx 16$ is accurate in the range $0 < \kappa < 0.2$ to within 12% of computed values. A positive spectrum is ensured by $B < m_0^2$. As water is removed from the elastocapillary system of Fig. 5.1, B reaches its maximum value when the membrane touches the plate where $B_{max} = 2\Pi^4/[K_w^3\kappa(1-\kappa^2)^4]$. To ensure that $\vartheta_i^* > 0$ always holds during drying, we require $B_{max} < m_0^2$, leading to

$$\Pi \lesssim 2^{3/4}(1-\kappa^2), \tag{5.62}$$

based on the foregoing approximation of m_0 and the scaling relation $K_w^3 \sim 1/\kappa$. Consequently, $N_C \gg 1/[2^{3/2}(1-\kappa^2)^2]$ guarantees that the in-plane spectrum is positive, and that the membrane profile can be accurately approximated by Eq. (5.55).

5.7.2 Out-of-plane spectrum

The spectrum of the membrane out-of-plane equilibrium is given by

$$\begin{cases} \mathscr{L}_{(\hat{w})}Y = r_p\lambda Y \\ Y'(R_{00}) = 0, \quad Z(R) = 0, \end{cases} \tag{5.63}$$

where λ, Y, and r_p are the eigenvalue, eigenfunction, and weight function of $\mathscr{L}_{(\hat{w})}$. Without attempting to solve Eq. (5.63), we demonstrate that $\mathscr{L}_{(\hat{w})}$ has a positive spectrum. From the quadratic form

$$\left\langle \mathscr{L}_{(\hat{w})}Y_i, Y_i \right\rangle = \lambda_i \left\langle r_p Y_i, Y_i \right\rangle = \int_{R_{00}}^{R} r_p(\hat{N}_{rr} + C\hat{w}'^2)Y_i'^2 \mathrm{d}r_p,$$

it follows that $\lambda_i > 0$ because $\hat{N}_{rr} > 0$, based on the variational approximation discussed in section 5.6. Similarly to the in-plane spectrum, Y_0 corresponding to $\lambda_0 = 0$ has only a trivial solution where $\ker(\mathscr{L}_{(\hat{w})}) = \{\mathbf{0}\}$ and $\dot{\hat{w}} = \mathbf{0}$ at $\dot{P} = 0$.

5.7.3 Meniscus spectrum

When the meniscus is a bridge, the spectrum cannot be determined analytically. Therefore, in this section, we only study the meniscus spectrum for simply connected menisci determined by

$$\begin{cases} \mathcal{L}_{(\hat{r})}X = \mu X \\ y_0 X'(y_0) - X(y_0) = 0, \quad X(1) = \text{finite}, \end{cases} \tag{5.64}$$

where μ and X are the eigenvalue[3] and eigenfunction of $\mathcal{L}_{(\hat{r})}$. These also denote the eigenvalue and eigenfunction of $\mathscr{L}_{(\hat{r})}$ for doubly-connect menisci. Solving Eq. (5.64)

[3]Not to be confused with the chemical potential in Eqs. (5.1) and (5.2).

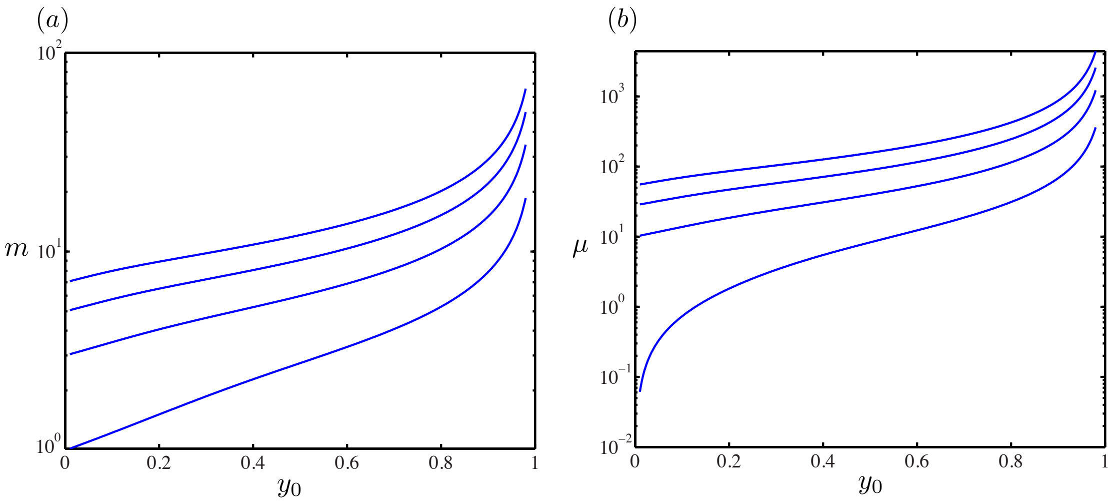


FIGURE 5.6: Meniscus spectrum; (a) order of eigenfunctions m_i and (b) corresponding eigenvalues μ_i with $i = 0, 1, 2, 3$ (upward).

furnishes

$$X_i(y) = C_{1,i} \mathrm{P}_{m_i}(y), \tag{5.65}$$

$$\mu_i = [(2m_i + 1)^2 - 9]/4, \quad i = 0, 1, 2, \cdots \tag{5.66}$$

where P_m is the real-valued order Legendre function of first kind, and m_i is the ith root of

$$y_0 \mathrm{P}'_{m_i}(y_0) - \mathrm{P}_{m_i}(y_0) = 0. \tag{5.67}$$

The boundary condition of Eq. (5.64) at $y = y_0$, resulting from perturbations that can displace the hole edge, is a key feature of this study. It implies that the meniscus contact line at the hole edge exhibits a mixed characteristics of free and pinned contact lines, depending on the bubble size. In the limit $y_0 \to 0$, where the bubble is a hemisphere, the contact line behaves similarly to a pinned contact line. As shown in Fig. 5.6, this limit, which corresponds to the pressure turning point ($\dot{P} = 0$) of the elastocapillary model in Fig. 5.1 with simply connected menisci, occurs at the point of stability exchange where $\mu_0 = 0$. Therefore, as expected, this limit coincides with the stability limit of pressure-controlled spherical menisci with a pinned contact line (Michael, 1981). Here, the order of the Legendre function takes integer values where $m_i = 2i + 1$, and $\ker(\mathcal{L}_{(\hat{r})}) \neq \{\mathbf{0}\}$. An important implication is that, unlike the in-plane and out-of-plane spectrum, $\dot{\hat{r}} \neq \mathbf{0}$ at $\dot{P} = 0$.

5.8 Stability along equilibrium branches

In this section, a relation between stability and the slope of equilibrium branches in v_l versus P diagrams for constrained and unconstrained problems is established. We assume that the foregoing variational approximation for the membrane equilibrium and, particularly, Eq. (5.62) always hold, so $\vartheta_i > 0$ and $\lambda_i > 0$ for all i. As discussed in section 5.4.3, stability can be determined by the sign of Q in Eqs. (5.43) and (5.44) with $N, \bar{\xi}_1, \phi_1, \psi_1 \in \mathscr{H}^0$ satisfying Eqs. (5.45)-(5.48). Following the Ritz method (Gelfand & Fomin, 2000), Q is examined in a countable dense subspace of $\mathscr{H}^0$, the existence of which is guaranteed by the separability of $\mathscr{H}^0$ (Akhiezer & Glazman, 1993), because there always exits a function in the dense subspace that is arbitrarily close to any $f \in \mathscr{H}^0$. Furthermore, the eigenfunctions of the Sturm-Liouville operators in Eqs. (5.49)-(5.52) form a complete orthogonal basis in $\mathscr{H}^0$ (Walter, 1998), and, thus, span the respective dense subspace. Hence, $\phi_1 \in \operatorname{span}\{Z_i\}$, $\psi_1 \in \operatorname{span}\{Y_i\}$, $N \in \operatorname{span}\{X_i\}$ when the meniscus is a bubble, and $\bar{\xi}_1 \in \operatorname{span}\{X_i\}$ when the meniscus is a bridge.

We unify the representation of quadratic forms in this section by expressing Q in terms of $\bar{\xi}_1$ for simply connected and doubly connected menisci. When the meniscus is a bubble, because $\left\langle \mathscr{L}_{(\hat{r})}\bar{\xi}_1, \bar{\xi}_1 \right\rangle = \gamma_{gl} \left\langle \mathcal{L}_{(\hat{r})} N, N \right\rangle$, the sign of Q is determined by the eigenvalues of $\mathcal{L}_{(\hat{r})}$, no matter how Q is expressed.

Lemma 1. *The necessary condition for an equilibrium branch to be stable in the unconstrained problem is $\mu_i \geq 0, \vartheta_i \geq 0, \lambda_i \geq 0$ for all i.*

We assume the contrary is true. For example, let $\mu_0 < 0$ and $\mu_i > 0$ for $i \geq 1$. Therefore, $\bar{\mu}_0 < 0$. Choosing $\bar{\xi}_1 = a_0 \bar{X}_0$ with

$$a_0 > \left[\frac{\left\langle \bar{\mathscr{L}}_{(\hat{u})}\phi_1, \phi_1 \right\rangle_1 + \left\langle \bar{\mathscr{L}}_{(\hat{w})}\psi_1, \psi_1 \right\rangle_1}{|\bar{\mu}_0| \left\langle \bar{X}_0, \bar{X}_0 \right\rangle_1} \right]^{1/2}$$

results in $Q < 0$, which is a contradiction. Here, $\bar{\mu}$ and $\bar{X}$ denote eigenvalues and eigenfunctions of $\bar{\mathscr{L}}_{(\hat{r})}$. $\square$

Note that in lemma 1, Eqs. (5.41)-(5.42) are used to illustrate how the relationship between the spectrum of barred and unbarred operators can be applied to prove stability. Hereafter, Eqs. (5.43)-(5.44) are directly used to determine the sign of Q. The following lemma connects stability to the slope of equilibrium branches in the unconstrained problem.

Lemma 2. *The slope of a stable equilibrium branch at any point in the J versus P diagram is non-negative.*

Differentiating Eqs. (5.24)-(5.26) along an equilibrium branch results in

$$[U''_{(\hat{y})} - PJ''_{(\hat{y})}]\dot{\hat{y}} = \dot{P}J'_{(\hat{y})}, \quad \hat{y} = \hat{r}, \hat{u}, \hat{w}, \tag{5.68}$$

furnishing

$$\left\langle [U''_{(\hat{r})} - PJ''_{(\hat{r})}]\dot{\hat{r}}, \dot{\hat{r}} \right\rangle + \left\langle [U''_{(\hat{u})} - PJ''_{(\hat{u})}]\dot{\hat{u}}, \dot{\hat{u}} \right\rangle + \left\langle [U''_{(\hat{w})} - PJ''_{(\hat{w})}]\dot{\hat{w}}, \dot{\hat{w}} \right\rangle$$
$$= \dot{P}\left[\left\langle J'_{(\hat{r})}, \dot{\hat{r}} \right\rangle + \left\langle J'_{(\hat{u})}, \dot{\hat{u}} \right\rangle + \left\langle J'_{(\hat{w})}, \dot{\hat{w}} \right\rangle \right].$$

Using Eq. (D.2),

$$\dot{J} = \left\langle J'_{(\hat{r})}, \dot{\hat{r}} \right\rangle + \left[\dot{\hat{r}}K_{r'} + \dot{\hat{z}}K \right]_{\hat{z}_0}^{\hat{h}} + \left\langle J'_{(\hat{u})}, \dot{\hat{u}} \right\rangle + \left[\dot{\hat{u}}M_{u'} \right]_{R_{00}}^{R}$$
$$+ \left\langle J'_{(\hat{w})}, \dot{\hat{w}} \right\rangle + \left[\dot{\hat{w}}M_{w'} \right]_{R_{00}}^{R} + \hat{R}_0\dot{\hat{R}}_0\hat{h} + \frac{\hat{R}_0^2\dot{\hat{h}}}{2}.$$

Substituting for K and M from Eqs. (5.9) and (5.11), all the boundary terms cancel each other, leading to

$$\dot{J} = \left\langle J'_{(\hat{r})}, \dot{\hat{r}} \right\rangle + \left\langle J'_{(\hat{u})}, \dot{\hat{u}} \right\rangle + \left\langle J'_{(\hat{w})}, \dot{\hat{w}} \right\rangle$$

and, consequently,

$$\left\langle [U''_{(\hat{r})} - PJ''_{(\hat{r})}]\dot{\hat{r}}, \dot{\hat{r}} \right\rangle + \left\langle [U''_{(\hat{u})} - PJ''_{(\hat{u})}]\dot{\hat{u}}, \dot{\hat{u}} \right\rangle + \left\langle [U''_{(\hat{w})} - PJ''_{(\hat{w})}]\dot{\hat{w}}, \dot{\hat{w}} \right\rangle$$
$$= \dot{P}\dot{J} = \dot{P}^2\frac{\mathrm{d}J}{\mathrm{d}P}.$$

Since the branch is stable, $\mu_i \geq 0, \vartheta_i \geq 0, \lambda_i \geq 0$ according to Lemma 1. Thus, the left-hand side is non-negative. □

Similarly, the following lemma connects stability to the slope of equilibrium branches in the constrained problem.

Lemma 3. *Suppose that, on a segment of an equilibrium branch, $\mathscr{L}_{(\hat{r})}$ is non-singular and has precisely one negative eigenvalue. Then, the segment is stable if and only if the slope at any point in the J versus P diagram is negative.*

We prove the lemma for doubly connected menisci. Consider the following perturbation decompositions

$$\bar{\xi}_1 = v_r + \alpha\eta_r, \quad \phi_1 = v_u + \alpha\eta_u, \quad \psi_1 = v_w + \alpha\eta_w,$$

where

$$\mathscr{L}_{(\hat{y})}\eta_y = J'_{(\hat{y})}, \quad y = r, u, w. \tag{5.69}$$

Because $\mathscr{L}_{(\hat{y})}$ are all non-singular, $\ker(\mathscr{L}_{(\hat{y})}) = \{\mathbf{0}\}$ and $J'_{(\hat{y})} \in \ker(\mathscr{L}_{(\hat{y})})^{\perp}$. Therefore, η_y always have a solution because $\mathscr{L}_{(\hat{y})}$ are Fredholm operators. From the volume constraint in Eq. (5.53),

$$\begin{aligned}\left\langle \mathscr{L}_{(\hat{r})}\eta_r, v_r\right\rangle + \left\langle \mathscr{L}_{(\hat{u})}\eta_u, v_u\right\rangle + \left\langle \mathscr{L}_{(\hat{w})}\eta_w, v_w\right\rangle& \\ = -\alpha\left[\left\langle \mathscr{L}_{(\hat{r})}\eta_r, \eta_r\right\rangle + \left\langle \mathscr{L}_{(\hat{u})}\eta_u, \eta_u\right\rangle + \left\langle \mathscr{L}_{(\hat{w})}\eta_w, \eta_w\right\rangle\right]&,\end{aligned}$$

furnishing

$$\begin{aligned}Q = \left\langle \mathscr{L}_{(\hat{r})}v_r, v_r\right\rangle + \left\langle \mathscr{L}_{(\hat{u})}v_u, v_u\right\rangle + \left\langle \mathscr{L}_{(\hat{w})}v_w, v_w\right\rangle& \\ - \alpha^2\left[\left\langle \mathscr{L}_{(\hat{r})}\eta_r, \eta_r\right\rangle + \left\langle \mathscr{L}_{(\hat{u})}\eta_u, \eta_u\right\rangle + \left\langle \mathscr{L}_{(\hat{w})}\eta_w, \eta_w\right\rangle\right]&.\end{aligned}$$

Note that $\left\langle \mathscr{L}_{(\hat{u})}\eta_u, \eta_u\right\rangle, \left\langle \mathscr{L}_{(\hat{w})}\eta_w, \eta_w\right\rangle, \left\langle \mathscr{L}_{(\hat{u})}v_u, v_u\right\rangle, \left\langle \mathscr{L}_{(\hat{w})}v_w, v_w\right\rangle > 0$ because the in-plane and out-of-plane spectrum are positive. We first show that the necessary condition for $Q > 0$ is

$$\left\langle \mathscr{L}_{(\hat{r})}\eta_r, \eta_r\right\rangle + \left\langle \mathscr{L}_{(\hat{u})}\eta_u, \eta_u\right\rangle + \left\langle \mathscr{L}_{(\hat{w})}\eta_w, \eta_w\right\rangle < 0. \tag{5.70}$$

We assume the contrary holds. Choosing $v_r = a_0 X_0$ with

$$a_0 > \left[\frac{\left\langle \mathscr{L}_{(\hat{u})}v_u, v_u\right\rangle + \left\langle \mathscr{L}_{(\hat{w})}v_w, v_w\right\rangle}{|\mu_0|\left\langle X_0, X_0\right\rangle}\right]^{1/2}$$

results in $Q < 0$, which is a contradiction. Therefore, it is always possible to construct perturbations that lead to instability if Eq. (5.70) does not hold. Here, μ_0 and X_0 are the negative eigenvalue and corresponding eigenfunction of $\mathscr{L}_{(\hat{r})}$. Next, we show that Eq. (5.70) is sufficient for $Q > 0$. Since perturbations are selected from a countable dense space, η_r can be written $\eta_r = b_0 X_0 + \sum_{i=1}^{\infty} b_i X_i$

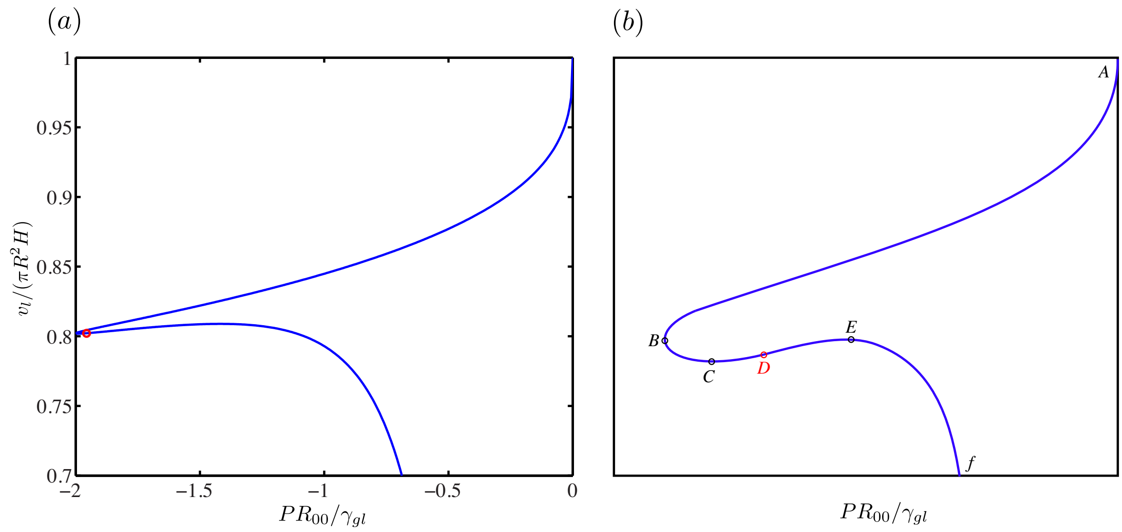


FIGURE 5.7: Equilibrium branch of the elastocapillary model in Fig. 5.1 for simply connected menisci with $\kappa = 0.1, \nu = 0.3, N_C = 15000, \Pi = 0.2$; (a) numerical computation and (b) schematic representation of pressure and volume turning points.

such that

$$b_0 > \left[\frac{\sum_{i=1}^{\infty} b_i^2 \mu_i \langle X_i, X_i \rangle + \left\langle \mathscr{L}_{(\hat{u})} \eta_u, \eta_u \right\rangle + \left\langle \mathscr{L}_{(\hat{w})} \eta_w, \eta_w \right\rangle}{|\mu_0| \langle X_0, X_0 \rangle} \right]^{1/2}$$

for Eq. (5.70) to hold. Moreover, any arbitrary meniscus perturbation can be written $\bar{\xi}_1 = a_0 X_0 + \sum_{i=1}^{\infty} a_i X_i$. Choosing $\alpha = a_0/b_0$ leads to $v_r = \sum_{i=1}^{\infty} c_i X_i$, implying that $\left\langle \mathscr{L}_{(\hat{r})} v_r, v_r \right\rangle > 0$, and, consequently, $Q > 0$. Hence, all meniscus perturbations, including those with $\left\langle \bar{\xi}_1, X_0 \right\rangle \neq 0$, can be decomposed into η_r and v_r, leading to a strictly positive second variation, provided Eq. (5.70) holds.

Substituting Eq. (5.69) into Eq. (5.68) furnishes

$$\eta_y = \dot{y}/\dot{P}, \quad y = r, u, w,$$

giving

$$\begin{aligned} &\left\langle \mathscr{L}_{(\hat{r})} \eta_r, \eta_r \right\rangle + \left\langle \mathscr{L}_{(\hat{u})} \eta_u, \eta_u \right\rangle + \left\langle \mathscr{L}_{(\hat{w})} \eta_w, \eta_w \right\rangle \\ &\qquad = \frac{1}{\dot{P}} \left[\left\langle J'_{(\hat{r})}, \dot{\hat{r}} \right\rangle + \left\langle J'_{(\hat{u})}, \dot{\hat{u}} \right\rangle + \left\langle J'_{(\hat{w})}, \dot{\hat{w}} \right\rangle \right] = \frac{\dot{J}}{\dot{P}} = \frac{\mathrm{d}J}{\mathrm{d}P}, \end{aligned}$$

which completes the proof. □

The proof of lemma 3 for simply connected menisci similarly proceeds using the

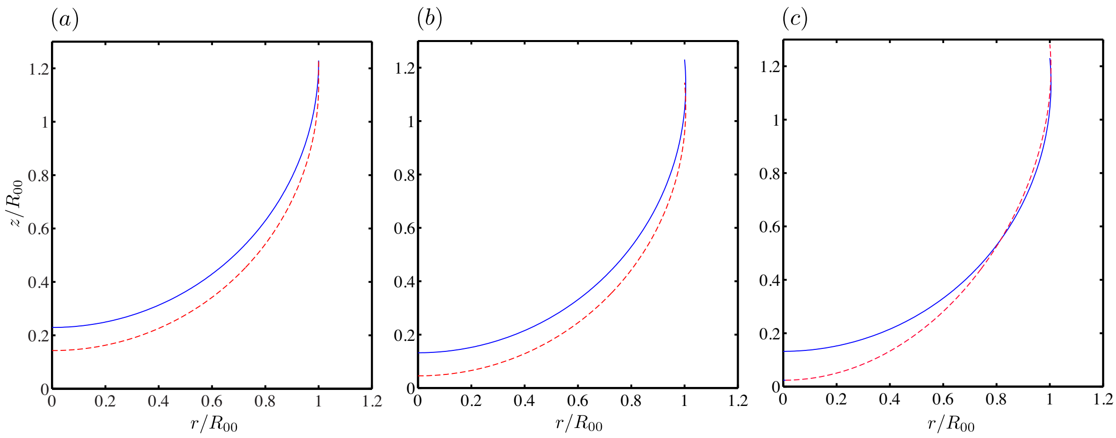


FIGURE 5.8: Meniscus equilibrium (solid) and perturbed (dashed) states at volume and pressure turning points in Fig. 5.7; The most dangerous perturbation normalized by $\langle N, N\rangle = R_{00}^2$ at (a) the pressure turning point B where $N(y) = \sqrt{3}R_{00}\mathrm{P}_1(y)$ and (b) the volume turning point C where $N(y) = \sqrt{1-y_0^2}/\langle \mathrm{P}_{m_0}, \mathrm{P}_{m_0}\rangle R_{00}\mathrm{P}_{m_0}(y)$. (c) A safe perturbation $N(y) = C_0X_0(y)+C_1X_1(y)$ at C normalized by $\langle N, N\rangle+\langle \psi_1, \psi_1\rangle = R_{00}^2$ where $\phi_1(\tilde{r}_p) = 0$ and $\psi_1(\tilde{r}_p) = a_0+a_1\tilde{r}_p+a_2\tilde{r}_p^2$ such that Eqs. (5.16)-(5.20) and (5.53) are satisfied, leading to $Q > 0$.

decomposition $N = v_{rn} + \alpha\eta_{rn}$ and accounting for the relation between N and $\bar{\xi}_1$ given by Eq. (C.4). Figure 5.7 shows how lemmas 1-3 can be applied to determine stability from the shape of an equilibrium branch in the unconstrained and constrained problems. In the unconstrained problem, where P is the control parameter, only the segments AB and CE can be stable according to lemma 2. The stability of the segment AB, excluding B, is deduced from the system configuration at A, corresponding to the fully saturated state. Here, the meniscus and membrane are planar, and the system is evidently stable, implying that $\mu_i, \vartheta_i, \lambda_i > 0$ for all i. Assuming that μ_0 varies continuously along the equilibrium branch, stability is lost at B, and the entire segment $BCDEf$ is unstable. The segment AB is also stable in the constrained problem, where v_l is the control parameter. Moreover, the stability of BC, along which $\mathcal{L}_{(\hat{r})}$ has one negative eigenvalue, is deduced from lemma 3. Beyond the volume turning point C, the entire segment $CDEf$ is unstable with respect to constant-volume perturbations.

Determining the stability of equilibrium branches is essential for predicting the dry-state conformation of the present model. This is illustrated by an example in Fig. 5.7. Here, the point D corresponds to the state where the bubble is tangent to the plate. Further decrease in v_l forces the bubble to bridge the membrane and plate, which is a necessary step for the elastocapillary system of Fig. 5.1 to collapse. However, when drying from the fully saturated state at A, collapse does not occur for the given parameters in Fig. 5.7, because the system loses stability at C before the bubble can bridge the membrane and plate.

Note that spherical menisci with a pinned contact line are always stable to constant-volume perturbations (Myshkis *et al.*, 1987). Furthermore, as discussed in section 5.7, the membrane is stable for all deflections, provided Eq. (5.62) is satisfied. Therefore, the meniscus and membrane are individually stable along the entire branch $ABCDEf$. However, the elastocapillary system as a whole subject to v_l = const. is unstable along $CDEf$, revealing an intimate connection between stability and the coupling of elastic and capillary forces. This manifests in the boundary shared by the elastic and capillary part where the meniscus and membrane interact through boundary displacing perturbations. Moreover, the nature of instabilities are influenced by the control parameter and how the meniscus and membrane interact with each other, as demonstrated in Fig. 5.8.

5.9 Concluding remarks

We have developed an elastocapillary model to study drying-induced structural failures, such as those arising from stiction in microelectromechanical systems. The model comprises an elastic membrane and a meniscus, deformed by the same pressure differential, interacting through a shared boundary. The existence of a stable equilibrium branch from the fully saturated to collapsed state is an essential precursor for structural failures. We examined the model stability and equilibrium using variational and spectral methods. Stability was related to the slope of equilibrium branches in the liquid content versus pressure diagram for the constrained and unconstrained problems. A variational approximation, complemented by scaling analysis, was derived, furnishing closed-form expressions for membrane equilibria. This approximation leads to a positive out-of-plane spectrum. For a given geometry, there is a critical elstocapillary number above (below) which the in-plane spectrum is positive (has a negative eigenvalue). These in-plane instabilities are a common cause of wrinkling in thin membranes. Thus, except for thin membranes, only meniscus perturbations can be dangerous for the elstocapillary system. This paper extends the work of Maddocks (1987) to elastocapillary systems that are subjected to boundary displacing perturbations, revealing a close connection between stability and the coupling of elastic and capillary forces. We demonstrated that the stability of the meniscus and membrane alone does not imply that the elastocapillary system as a whole is stable; the destabilizing effect of the membrane and meniscus interacting through their shared boundary must also be accounted for. Moreover, our results support a general concept in catastrophe theory that stability exchanges occur at tuning points with respect to the control parameter, thereby, reducing costly stability computations to search for folds on equilibrium branches.

Chapter 6

An elastocapillary model of wood-fibre collapse

6.1 Preface

This chapter applies the elastocapillary model and stability criteria developed in chapter 5 to study the drying-induced collapse of the cell lumen and cell-wall micropores. This model captures three characteristic length scales of wood cavities and predicts the dry-state conformation from the mechanical properties of cavity walls. These length scales can be correlated with the geometrical features of the lumen and micropores. Fixing the geometrical length scales, the dry-state conformation is determined by examining whether capillary forces are strong enough to overcome the wall stretching resistence and cause collapse.

6.2 Abstract

An elastocapilary model of wood-fibre collapse upon drying above the fibre saturation point is proposed. The model considers a circular elastic membrane with a hole at the center that is deformed by the capillary pressure of simply and doubly connected menisci. The membrane overlays a cylindrical cavity with rigid walls, trapping a prescribed volume of water. The dry-state is determined using the dihedral-angle and volume-turning-point stability criteria. Open and collapsed conformations are predicted from the scaled hole radius, cavity aspect ratio,

meniscus contact angle with the membrane and cavity walls, and an elastocapillary number measuring the membrane stretching rigidity relative to the water surface tension. For a given scaled hole radius and cavity aspect ratio, there is a critical elastocapillary number above which the system does not collapse upon drying. The critical elastocapillary number is weakly influenced by the contact angle over a wide range of the scaled hole radius, thus indicating a limitation of surface hydrophobization for controlling the dry-state conformation in pulping processes.

6.3 Introduction

Cellulose fibres are abundant natural resources with desirable characteristics, such as high-strength to weight ratio, corrosion-resistance, biodegradability, and biocompatibility (Cheung *et al.*, 2009, Eichhorn *et al.*, 2001, Walker, 2006). Besides their traditional application in papermaking, they have been used as reinforcements for plastic (Bledzki & Gassan, 1999) and cement (Campbell & Coutts, 1980) composites, scaffolds for tissue engineering (Müller *et al.*, 2006), and food-packaging materials (Sirviö *et al.*, 2014). The properties of fibre-based materials depend on wood-polymer characteristics and fibre conformation in the material microstructure.

Isolating fibres from parent wood by pulping is accompanied by dramatic conformational changes, the extent of which depends on the pulping process and the mechanical, physical, and structural properties of fibres. The extremities of heating and chemical treatment in pulping also play an important role. Fibres from chemical pulping are delignified and usually collapsed, whereas fibres from thermomechanical pulping are lignin-coated and mostly uncollapsed, imparting contrasting properties to fibre-based composites (Campbell & Coutts, 1980). For example, papers from chemical pulps have high tensile strength and smooth surfaces, suiting high-quality printing, while those from mechanical pulps have high compressive strength and rough surfaces, suiting fast, economical printing (Walker, 2006).

Wood-fibre deformation upon drying results from complex interactions between water and wood tissues in pulping (Walker, 2006). The degree to which water is associated with wood can be quantified by differential scanning calorimetry

(Nakamura *et al.*, 1981, Weise *et al.*, 1996). Here, three types of water are identified (Nakamura *et al.*, 1981): (i) free water, having the same transition temperature as bulk water, (ii) freezing bound water, having lower transition temperature than bulk water, and (iii) non-freezing bound water, which cannot be detected from the first-order transition. Molecules of non-freezing bound water are directly bound to cellulose hydroxyl groups, whereas those of freezing bound water and free water are trapped in the lumen and cell-wall micropores by capillary condensation. Noting that the heat of desorption required to break water-hydroxyl group bonds for non-freezing bound water is much larger than the excess enthalpy of vaporization due to meniscus formation for freezing bound and free water (Walker, 2006), drying stresses are expected to increase significantly at low moisture contents where the remaining water is mostly non-freezing bound.

Defining the fibre saturation point (FSP) as the moisture content where all the remaining water is adsorbed (non-freezing bound) (Walker, 2006) identifies two drying-deformation regimes (Tiemann, 1941): (i) collapse above the FSP where fibre macro- and microscopic structure is affected, and (ii) shrinkage below the FSP where cell-wall submicroscopic structure is affected. Inconsistent reports in the literature as to whether deformations upon drying begin above or below the FSP (Bariska, 1992, Hernández & Pontin, 2006, Pang, 2002) motivate further studies on wood-fibre collapse and shrinkage at various drying stages.

Barber & Meylan (1964) developed one of the first theoretical models to quantify shrinkage below the FSP. They considered a flat element of the cell wall, comprising only the S_2 layer with cellulose microfibrils embedded in an amorphous matrix. This matrix shrinks isotropically with a linear relationship between normal strain and moisture content, while the cell-wall overall shrinkage is anisotropic because the matrix is constrained by crystalline microfibrils. This model was later extended to account for the effect of a cylindrical cell wall (Barber, 1968), a multilayer cell wall (Yamamoto, 1999), changing lumen shape during drying (Pang, 2002), and moisture dependent mechanical properties (Thuvander *et al.*, 2002). Key observations from these studies include: (i) drying deformations and changes in the physical and mechanical properties of wood occur below the FSP, (ii) an anisotropic three-dimensional shrinkage is predicted with the radial component lower than the tangential component at small microfibril angles, agreeing with experiments (Cown & McConchie, 1980, Quirk, 1984), and (iii) volumetric shrinkage is proportional to the amount of water removed during drying.

In contrast with the foregoing studies, Hernández & Pontin (2006) reviewed experimental evidence in the literature to show that drying deformations can begin at moisture contents well above the FSP. These are attributed to collapsed lumen and micropores resulting from an interplay between water surface tension and cell-wall elastic resistance (Tejado & van de Ven, 2010, Tiemann, 1941). In fact, non-collapsed fibres can be produced by drying solvent-exchanged pulps where water is substituted with a low-surface tension liquid before drying (Walker, 2006), supporting the latter proposition. Depending on the cell-wall microstructure, mechanical properties, and pore-size, a spectrum of collapsed, partially-collapsed, and uncollapsed pores are observed (Tiemann, 1941, Walker, 2006), implying that the overall volumetric shrinkage in this regime is not linearly related to the volume of removed water. Here, we consider an elastocapillary model of the lumen and cell-wall micropores, rigorously examining the interactions between elastic and capillary forces during drying to provide a better understanding of drying deformation at moisture contents above the FSP.

Elastocapillary phenomena have been extensively studied over the past two decades in several areas, such as capillary-induced wrinkling (Huang *et al.*, 2007), microelectromechanical systems (Farshid-Chini & Amirfazli, 2010, Mastrangelo & Hsu, 1993*a*,*b*), capillary wrapping and origami (Py *et al.*, 2007), and self-assembly, coalescence, and bundling of lamellae (Bico *et al.*, 2004, Boudaoud *et al.*, 2007, Chandra & Yang, 2009, Duprat *et al.*, 2012). Mastrangelo & Hsu (1993*a*) studied the capillary driven deformation of beams and plates in micro-machined structures and derived collapse criteria based on the bending stiffness to surface tension ratio. Neglecting the meniscus contribution to the total energy, continuous equilibrium trajectories without stability exchange that connect the initial and collapsed states were identified as leading to collapsed conformations. A two-dimensional model of capillary rise between flexible sheets is given by Kim & Mahadevan (2006) based on the linear bending theory of beams. Here, the final conformation is determined by a balance involving gravity, elastic, and contact-line forces. This model was further examined by Kwon *et al.* (2008) for systems in which conformations are controlled by the liquid content. Taroni & Vella (2012) studied the equilibrium and stability of the same problem. Accounting for the meniscus shape, the Laplace pressure was also incorporated into the equilibrium equation of the beam. Moreover, multiple stable equilibria were found, and final conformations were determined through a dynamic analysis.

In this paper, we study an elastocapillary model of wood-fibre collapse for cavities that are characterized by three length scales. The model comprises a circular elastic membrane with a hole at the center. The membrane is anchored above a rigid plate, trapping a prescribed volume of water. We examine membrane deformations caused by a meniscus that forms at the hole as liquid is removed. This furnishes an idealized structure where stability and equilibrium are determined by the elastic-capillary force interactions, providing collapse criteria for lumen and cell-wall micropores at moisture contents above the FSP. Neglecting the contact line force, we adopt a variational formulation to examine stability and equilibria. Here, the meniscus geometry is exactly treated for the interfacial energies to account for the significant role of the Laplace pressure when collapsed conformations are approached. Using the stability criteria we recently derived based on a spectral analysis of the potential energy (Akbari *et al.*, 2015*d*), the dry-state conformation is determined from the stability of equilibrium branches.

Previous elastocapillary models in the literature are mostly based on the linear theory of plates, where the bending contribution to the total potential energy is only considered (Kim & Mahadevan, 2006, Kwon *et al.*, 2008, Taroni & Vella, 2012). In contrast, the present model only considers the stretching energy, accounting for variable and anisotropic axial forces. The latter is more challenging because the equilibrium equations are nonlinear and the in-plane displacement is not neglected. Moreover, the model does not admit buckling, which typically occurs in geometries with closed structures, such as those of wood fibres. Buckling and the bending rigidity are additional resistances to deformation. Hence, the collapse criteria derived in this paper can be regarded as upper bounds, which are suitable for predicting open conformations from geometrical parameters and mechanical properties.

6.4 Theory

We consider an elastocapillary model shown in Fig. 6.1 comprising a circular elastic membrane with a hole at the center supported on the sidewall of a cylindrical cavity with rigid walls, trapping a volume v_l of liquid below the membrane and volume v_g air between the bounding surface and membrane. The cavity is open to the atmosphere from the top. A meniscus forms at the hole as the liquid (water) is removed, resulting in a difference between the liquid pressure p_l and atmospheric

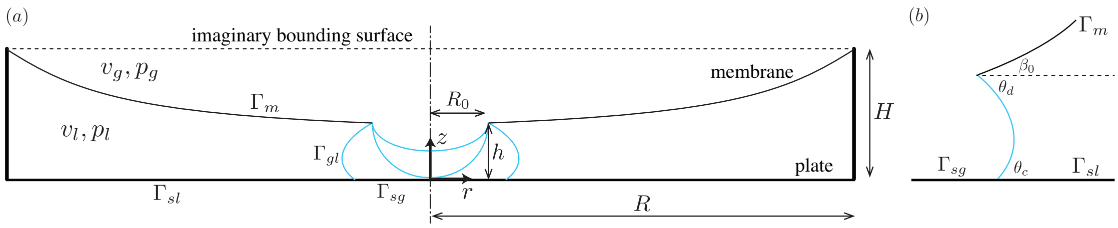


FIGURE 6.1: Elastocapillary model; (a) schematic showing simply connected meniscus (top), doubly connected meniscus (bottom), transition from simply to doubly connected meniscus (middle), and (b) contact angles.

pressure p_g that causes the membrane to deform. Here, the membrane radius R, hole radius R_0, and cylinder height H are the model length scales that control the interplay between elastic and capillary forces. To determine the equilibria at a given v_l, we consider an imaginary bounding surface (dashed line in Fig. 6.1(a)) that covers the cavity from the top. The system is completely isolated from the surrounding by the bounding surface and cylinder walls. The meniscus is initially a bubble, which can bridge the gap upon contact with the plate at the bottom of the cylinder, forming a free contact line with the plate. Here, θ_c and θ_d are the thermodynamic contact angle and dihedral angle that the meniscus forms with the plate and membrane, respectively. Assuming that all the dimensions are small compared to the capillary length, the gravity force is neglected. The membrane and meniscus are assumed axisymmetric in equilibrium and perturbed configurations. Drying dynamics are assumed to be slow (quasi-static), so that the system evolves through a sequence of equilibrium states. We also assume that the inner surfaces of the plate and membrane that are in contact with water produce the same thermodynamic contact angle θ_c.

We derived a variational principle for the stability and equilibrium of the elasto-capillary model shown in Fig. 6.1 (Akbari *et al.*, 2015*d*). Neglecting the bending contribution to the elastic strain energy, the membrane in-plane and out-of-plane displacement profiles are (see Appendix E)

$$\tilde{w}(\tilde{r}_p) = \tilde{w}_0(1 - \tilde{r}_p^2), \tag{6.1}$$

$$\tilde{u}(\tilde{r}_p) = \frac{\tilde{w}_0^2(1 - \tilde{r}_p^2)}{4}\left[\tilde{r}_p(3-\nu) + \frac{\kappa^2[(3-\nu)(1+\nu) - \kappa^2(1-\nu^2)]}{\tilde{r}_p[1-\nu+\kappa^2(1+\nu)]}\right], \tag{6.2}$$

where $\tilde{r}_p$, $\tilde{u}$, and $\tilde{w}$ are the radial position in the referential coordinate r_p, in-plane displacement u, and out-of-plane displacement w, all scaled with R; moreover,

$\tilde{w}_0 = (Q_c/N_C)^{1/3} K_w$, and

$$K_w = \left[\frac{3[1-\nu+\kappa^2(1+\nu)]}{\kappa(1-\kappa^2)(1-\nu^2)[7-\nu+\kappa^2(1+\nu)]} \right]^{1/3}. \tag{6.3}$$

Here, $\kappa = R_{00}/R$, $Q_c = PR_{00}/\gamma_{gl}$, $N_C = C/\gamma_{gl}$ are the scaled hole radius, scaled capillary pressure and elastocapillary number where $P = p_l - p_g$ and C, ν, γ_{gl}, R_{00} are the membrane axial rigidity, Poisson ratio, air-water surface tension, and hole radius in the referential configuration. Equations (6.1) and (6.2) are derived using von Kármán's plate theory, which requires $\mathrm{d}w/\mathrm{d}r_p \sim H/R \ll 1$ (Akbari *et al.*, 2015*d*). Therefore, we restrict our analysis to cases for which $H/R \ll 1$, so that the membrane displacements can be accurately approximated by Eqs. (6.1) and (6.2).

To compute the liquid volume v_l, meniscus meridian curve, and minimum gap between the membrane and plate h, we neglect the in-plane displacement, which is a reasonable approximation when $H/R \ll 1$ (Akbari *et al.*, 2015*d*). Neglecting the in-plane displacement implies that the hole expansion resulting from the membrane deflection is negligible ($R_0 \approx R_{00}$). Furthermore, one must ensure that a given membrane profile corresponding to the capillary pressure of a meniscus does not violate any geometrical constraint. Specifically, the sum of the membrane deflection and the gap between the membrane and plate at the hole edge must equal H, and volume bounded by the meniscus and membrane must equal the liquid volume. These constraints are respectively $H = h - w(R_{00})$ and $v_l + v_g = \pi R^2 h$, which upon substituting Eq. (6.1) furnish

$$\kappa\Lambda - \Pi - \left(\frac{Q_c}{N_C}\right)^{1/3} K_w(1-\kappa^2) = 0, \tag{6.4}$$

$$\left[\frac{3}{4} + \frac{3(1-\kappa^2)}{8\kappa^2}\right]\Lambda + \frac{3(1-\kappa^2)}{8\kappa^3}\Pi - \hat{v}_a - \hat{v}_l = 0, \tag{6.5}$$

where $\Lambda = h/R_{00}$, $\Pi = H/R$, and v_a are the slenderness, aspect ratio, and the volume of the air between the meniscus and the plane $z = h$, and $\hat{v}_i = v_i/(4\pi R_{00}^3/3)$.

The existence of a continuous and stable solution branch from the fully-saturated state ($h = H$) to the collapsed state ($h = 0$) is a necessary condition for the elastocapillary system in Fig. 6.1 to collapse. Akbari *et al.* (2015*d*) demonstrated that this system loses stability during drying at volume turning points[1], provided the meniscus is pinned to the hole edge and $\Pi \lesssim 2^{3/4}(1-\kappa^2)$. However, assuming that menisci are pinned to the hole edge implies that the contact line is restrained by an external force. This assumption can be relaxed by permitting the contact line to move if it is energetically favorable to do so, which leads to the necessary condition $\theta_d + \beta_0 > \theta_c$ for menisci to be stably pinned to the hole edge (Myshkis *et al.*, 1987), provided the meniscus contact angles with the membrane and plate are the same. Here, β_0 measures the membrane slope at the hole edge (see Fig. 6.1b), which is negligible because $\tan\beta_0 \approx \tilde{w}'(\kappa) \sim \Pi\kappa \ll 1$. This simplifies the foregoing stability criterion to

$$\theta_d > \theta_c. \tag{6.6}$$

Moreover, when the meniscus is a bridge, equilibrium solutions of the meniscus meridian curve can intersect the boundaries or be self-intersecting (Akbari *et al.*, 2015*c*). These menisci are clearly non-physical and must be excluded before examining the stability of equilibrium branches.

The foregoing stability criteria only ensure that the membrane can contact the plate, which corresponds to the 'contact bound' defined by Mastrangelo & Hsu (1993*a*). However, these are not sufficient for predicting the dry-state conformation. Accounting for adhesion energy between the membrane and plate is also required to determine whether the membrane and plate remain attached when dry (Mastrangelo & Hsu, 1993*b*). In the remainder of the paper, assuming that the adhesion energy is always strong enough to maintain the membrane and plate attached upon contact, the dry state is identified as collapsed if there exists a continuous solution branch without volume turning points and non-physical menisci from $h = H$ to 0 along which Eq. (6.6) is satisfied.

We solve nonlinear systems of equations using Newton-Raphson based predictor-corrector techniques. Solution branches are constructed using Keller's arclength continuation method (Seydel, 2009). Branch continuation begins from a state that either has an analytical solution or can be readily constructed (*i.e.*, cases with low sensitivity to initial guess and problem parameters) and terminates when the solution does not satisfy the geometrical constraints (*e.g.*, negative $\hat{v}_a$ or Λ).

[1] Unless stated otherwise, volume turning point refers to a turning point in v_l throughout this paper.

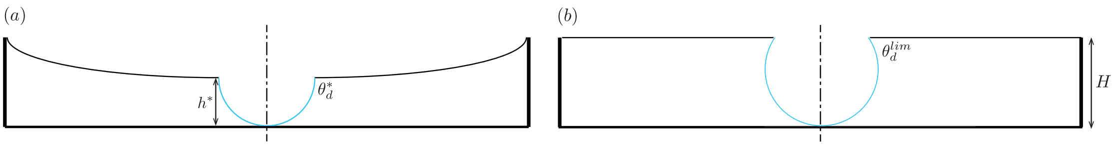


FIGURE 6.2: Special states for simply connected menisci; (a) tangent bubble and (b) rigid membrane ($N_C \to \infty$).

6.5 Results and discussion

We determine the dry-state conformation in the parameter space (κ, N_C, θ_c) at fixed ν and Π. Drying from the fully-saturated state, the elastocapillary system of Fig. 6.1 must undergo three deformation phases to collapse: (i) The initial phase where the meniscus is a bubble (simply connected), which can bridge the membrane and plate when the bubble apex reaches the plate, (ii) the transition phase where the bubble transforms into a bridge at fixed liquid volume v_l^*, and (iii) the final phase where the meniscus is a bridge (doubly connected) with the slenderness approaching zero as v_l decreases further below v_l^*.

6.5.1 Initial phase

Here, the meniscus is a truncated sphere because gravity is neglected. Accordingly, the scaled bubble volume and capillary pressure are

$$\hat{v}_a = \frac{2 + 3\cos\theta_d - \cos^3\theta_d}{4\sin^3\theta_d}, \tag{6.7}$$

$$Q_c = -2\sin\theta_d. \tag{6.8}$$

The system state at any $\hat{v}_l$ does not depend on θ_c. Solving Eqs. (6.4), (6.5), (6.7), and (6.8), the state is specified for a given κ and N_C, where the dependent variables are $(\Lambda, \theta_d, Q_c, \hat{v}_a)$. The tangent bubble and rigid membrane, shown in Fig. 6.2, are special states in this phase that are determined by $\Lambda^* = \cot(\theta_d^*/2)$ and $\Lambda^{lim} = \cot(\theta_d^{lim}/2) = \Pi/\kappa$, respectively. Here, the superscripts '$*$' and 'lim' denote variables in the tangent-bubble and rigid-membrane states. The latter corresponds to $N_C \to \infty$, where the membrane is rigid and does not deform.

Note that the dependent variables in the tangent-bubble state are $(\Lambda^*, \theta_d^*, Q_c^*, \hat{v}_a^*, \hat{v}_l^*)$, and the system state is specified by fixing κ and N_C. Choosing θ_d^* as the solution norm, we first determine equilibria at the tangent-bubble state in the

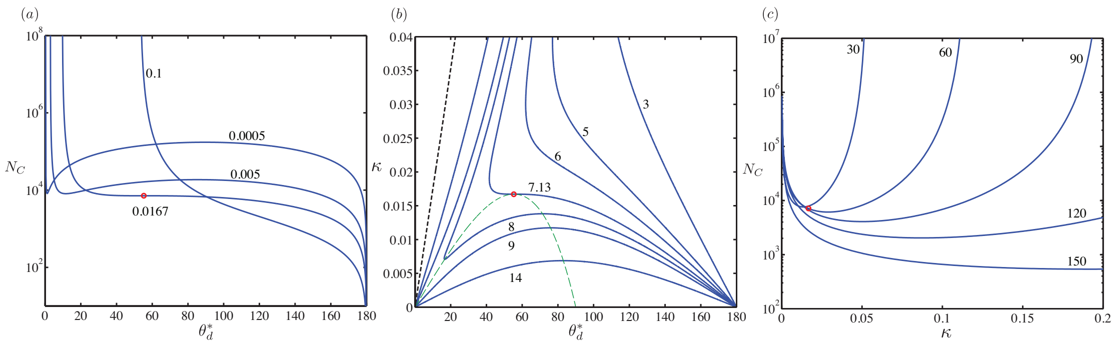


FIGURE 6.3: Equilibria at the tangent-bubble state (Fig. 6.2a) with $\nu = 0.3$, $\Pi = 0.2$. Circles indicate the critical point where $(\partial N_C/\partial\theta_d^*)_\kappa$ and $(\partial^2 N_C/\partial\theta_d^{*2})_\kappa = 0$; (a) constant-$\kappa$ isocontours (labels denote κ), (b) constant-N_C isocontours (labels denote $N_C/10^3$) with dashed black and green lines indicating the rigid-membrane limit and locus of isocontour extrema, and (c) constant-θ_d^* isocontours (labels denote θ_d^* in degrees).

space $(\kappa, N_C, \theta_d^*)$, as shown in Fig. 6.3. At fixed κ, there is a critical κ below which the tangent-sphere state has three equilibrium solutions for a given N_C. Similarly, at fixed N_C, there is a critical N_C above which the tangent-sphere state has three equilibrium solutions for a given κ. Here, the stability of the equilibrium branch from the fully-saturated to tangent-bubble state must be examined to determine which equilibrium solution is realized for given (κ, N_C). Hereafter, we refer to equilibrium branches parametrized with v_l as 'drying trajectories', which can be conveniently represented by x_l, Q_c, θ_d, and Λ. Note that

$$x_l = v_l/(\pi R^2 H) \tag{6.9}$$

is a volume fraction, measuring the liquid content relative to the fully-saturated state.

Figure 6.4 illustrates a drying trajectory corresponding to (κ, N_C) for which the tangent-bubble state has one equilibrium solution. The equilibrium branch has no turning point in v_l, so no stability exchange occurs from the fully-saturated state to the tangent-bubble state. At the tangent-bubble state, $\theta_d^* \simeq 110°$. Note that θ_d monotonically decreases with decreasing v_l. Therefore, when $\theta_c > 110°$, the contact line is detached from the hole edge before the bubble can bridge the membrane and plate according to Eq. (6.6), implying that the dry-state conformation is open. Figure 6.5 shows a drying trajectory where the tangent-bubble state has three equilibrium solutions. The equilibria at B and C are past the turning point in v_l on an unstable branch (Akbari *et al.*, 2015*d*), and, thus, can

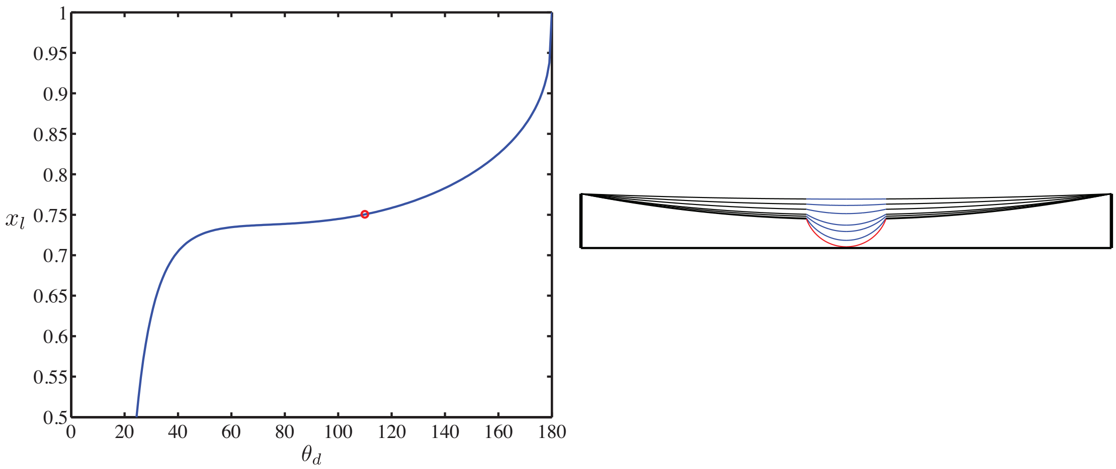


FIGURE 6.4: Drying trajectory with $\nu = 0.3$, $\Pi = 0.2$, $\kappa = 0.15$, $N_C = 5000$. Circle indicates the tangent-bubble state at $x_l \simeq 0.7505$ (left). Conformations along the drying trajectory at $x_l = 0.95, 0.9, 0.85, 0.8, 0.78, 0.76, 0.7505$ (right, downward).

not be practically realized. The equilibrium at A, where $\theta_d^* \simeq 121°$, is on a stable branch; thus, drying from the fully-saturated state, the tangent-bubble state at A can be practically realized. Similarly to Fig. 6.4, when $\theta_c > 121°$, the contact line is detached from the hole edge before the bubble can bridge the membrane and plate.

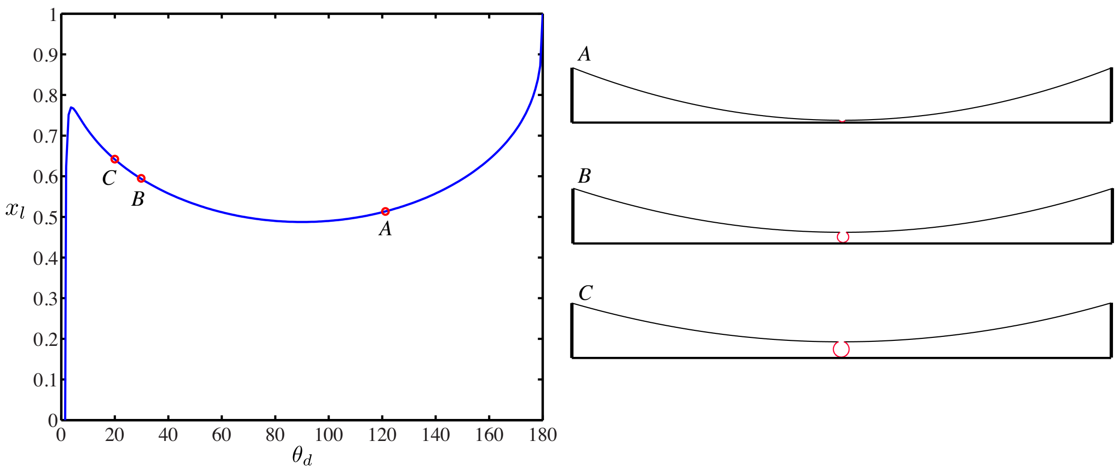


FIGURE 6.5: Drying trajectory with $\nu = 0.3$, $\Pi = 0.2$, $\kappa = 0.01$, $N_C = 8000$. Circles indicate multiple equilibria at the tangent-bubble state (left). Conformations at the tangent-bubble state corresponding to $x_l \simeq 0.5140, 0.5932, 0.6410$ (right, downward).

To determine the region in (κ, N_C) in which the tangent-bubble state is reached before stability is lost at a volume turning point, we determine tangent-bubble states that coincide with volume turning points (dashed black lines in Fig. 6.6).

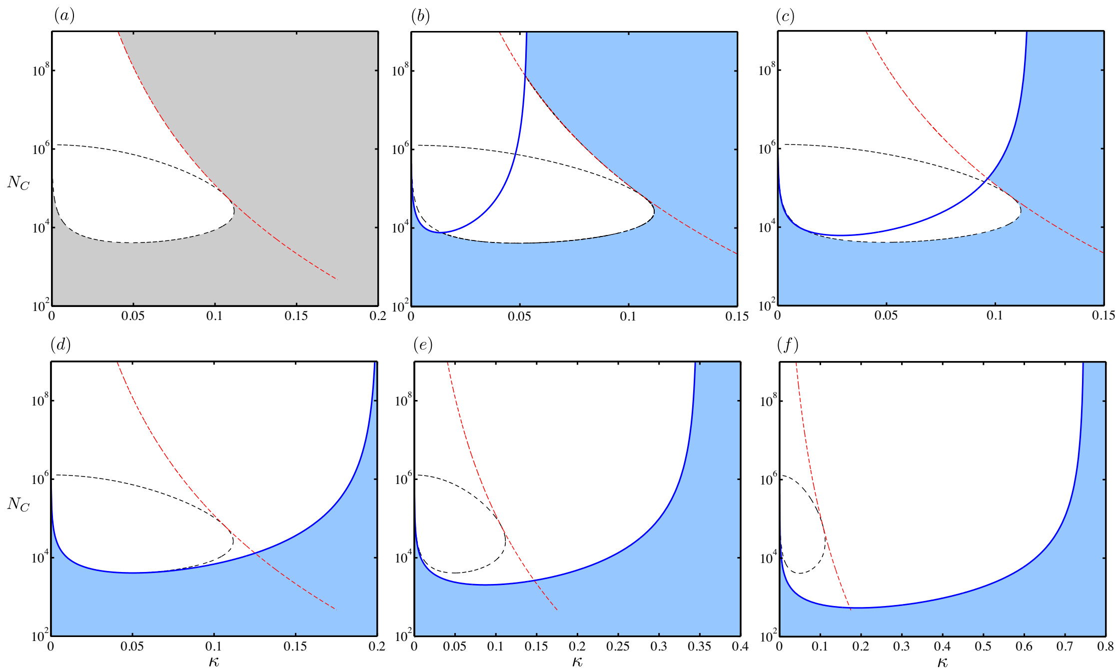


FIGURE 6.6: Phase diagrams providing upper bounds on the region of collapsed conformation in the parameter space from the locus of volume turning points coinciding with the tangent-bubble state (dashed black), volume inflection points (dashed red), and the dihedral-angle stability limit ($\theta_d^* = \theta_c$) according to Eq. (6.6) at the tangent-bubble state (solid) with $\nu = 0.3$, $\Pi = 0.2$; (a) region of the parameter space where the tangent-bubble state is reached before stability is lost at a volume turning point (gray shade) and its subregions (blue shade) in which Eq. (6.6) is also satisfied when (b) $\theta_c = 30°$, (c) $\theta_c = 60°$, (d) $\theta_c = 90°$, (e) $\theta_c = 120°$, and (f) $\theta_c = 150°$.

These states are determined by $(\partial \hat{v}_l/\partial \theta_d)_{\kappa,N_C} = 0$, where $(\Lambda^*, \theta_d^*, Q_c^*, \hat{v}_a^*, \hat{v}_l^*, N_C)$ are specified for a given κ. Moreover, at fixed κ, there is a critical N_C, determined by $(\partial \hat{v}_l/\partial \theta_d)_{\kappa,N_C} = 0$ and $(\partial^2 \hat{v}_l/\partial \theta_d^2)_{\kappa,N_C} = 0$, at which the two volume turning points on drying trajectories coalesce into an inflection point; above this elastocapillary number, drying trajectories have no volume turning points, and no stability exchange occurs along equilibrium branches. These criteria define another state (dashed red lines in Fig. 6.6) that is specified for a given κ, similarly to the tangent-bubble state at volume turning point. The shaded region in Fig. 6.6a, constructed based on the foregoing states, furnishes a range of (κ, N_C) for which the tangent-bubble state is reached before the volume turning point. However, one must also ensure that Eq. (6.6) is satisfied along the drying trajectory from the fully-saturated to tangent-bubble state. The shaded regions in Fig. 6.6b-f are subregions of the shaded region in Fig. 6.6a in which the dihedral-angle and turning-point stability criteria are simultaneously satisfied, therefore, providing a range of (κ, N_C) in which the tangent-bubble state can be realized.

Tangent-bubble states coincide with volume turning points only for $\theta_d^* < \pi/2$. To prove this, we choose θ_d as an independent variable, so that $\hat{v}_l = \hat{v}_l(\kappa, N_C, \theta_d)$ and $\Lambda = \Lambda(\kappa, N_C, \theta_d)$ for the equilibria from the fully-saturated to tangent-bubble state. Differentiating Eq. (6.5) with respect to θ_d furnishes

$$\left(\frac{\partial \hat{v}_l}{\partial \theta_d}\right)_{\kappa, N_C} = \left[\frac{3}{4} + \frac{3(1-\kappa^2)}{8\kappa^2}\right] \left(\frac{\partial \Lambda}{\partial \theta_d}\right)_{\kappa, N_C} - \frac{\mathrm{d}\hat{v}_a}{\mathrm{d}\theta_d}, \tag{6.10}$$

leading to

$$\left(\frac{\partial \hat{v}_a}{\partial \Lambda}\right)_{\kappa, N_C} = \left[\frac{3}{4} + \frac{3(1-\kappa^2)}{8\kappa^2}\right] > 0 \tag{6.11}$$

at volume turning points. Because Λ decreases (increases) with increasing $\hat{v}_l$ when the bubble is smaller (larger) than a hemisphere, it follows that $(\partial \hat{v}_a/\partial \Lambda)_{\kappa, N_C} < 0$ ($(\partial \hat{v}_a/\partial \Lambda)_{\kappa, N_C} > 0$) when $\theta_d > \pi/2$ ($\theta_d < \pi/2$). Noting that Eq. (6.11) holds for the tangent-bubble states coinciding with volume turning points, it follows that $\theta_d^* < \pi/2$ along the dashed black lines in Fig. 6.6.

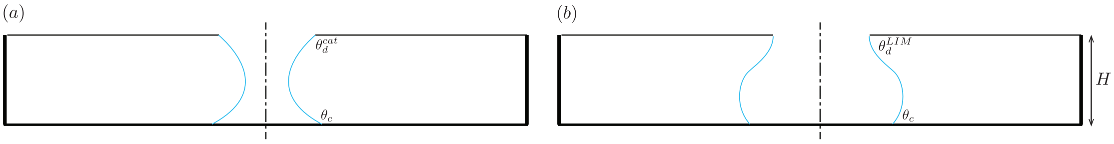


FIGURE 6.7: Special states for doubly connected menisci; (a) catenoidal bridge ($Q_c \to 0$) and (b) rigid membrane ($N_C \to \infty$).

6.5.2 Transition phase

When the bubble reaches the plate in the tangent-bubble state, it transforms into a doubly connected meniscus by bridging the membrane and plate. The dynamics of such transformations are comparable to those of stability loss in liquid-bridge breakup (Eggers, 1997, Rivas & Meseguer, 1984), so the time scale of the bubble-to-bridge transition (BBT) is very small compared to that of drying. Therefore, this transition is volume preserving ($v_l =$ const.), and the state after the transition is determined by (Akbari *et al.*, 2015*c*)

$$\sqrt{1 + a^2 + 2a\cos\tau_0} - |Q_c| = 0, \tag{6.12}$$

$$\int_{\tau_0}^{\tau_1} \frac{1 + a\cos t}{\sqrt{1 + a^2 + 2a\cos t}} \mathrm{d}t + \Lambda Q_c = 0, \tag{6.13}$$

$$\mathrm{sign}(Q_c)\frac{1 + a\cos\tau_0}{a\sin\tau_0} - \tan\theta_d = 0 \tag{6.14}$$

$$\mathrm{sign}(Q_c)\frac{1+a\cos\tau_1}{a\sin\tau_1}+\tan\theta_c=0, \tag{6.15}$$

$$\int_{\tau_0}^{\tau_1}\sqrt{1+a^2+2a\cos t}(1+a\cos t)\mathrm{d}t+\frac{4}{3}Q_c^3\hat{v}_a=0, \tag{6.16}$$

Eq. (6.5) with $\hat{v}_l = \hat{v}_l^*(\kappa, N_C)$, and Eq. (6.4), furnishing the dependent variables $(Q_c^{**}, \tau_0^{**}, \tau_1^{**}, a^{**}, \theta_d^{**}, \Lambda^{**}, \hat{v}_a^{**})$. Here, the superscript '**' denotes variables after the BBT, and the system state is specified for a given κ, N_C and θ_c. Note that τ_0 and τ_1 are, respectively, the mean-curvature-scaled arclengths at the hole edge and the bridge contact line with the plate, and a is the bridge shape parameter (Akbari *et al.*, 2015*c*).

The catenoidal bridge and rigid membrane are special limiting states in this phase, as shown in Fig. 6.7. After the transition, the meniscus mean curvature is zero ($Q_c \to 0$), and the membrane is undeflected in the catenoid limit. The dependent variables in this state $(\theta_d^{cat}, \hat{v}_a^{cat}, N_C)$ are determined by

$$\frac{1}{\sqrt{\hat{s}_0^2+1}}\ln\left(\frac{\hat{s}_1+\sqrt{\hat{s}_1^2+1}}{\hat{s}_0+\sqrt{\hat{s}_0^2+1}}\right)-\frac{\Pi}{\kappa}=0, \tag{6.17}$$

$$\frac{\hat{s}_1\sqrt{\hat{s}_1^2+1}-\hat{s}_0\sqrt{\hat{s}_0^2+1}+(\Pi/\kappa)\sqrt{\hat{s}_0^2+1}}{(\hat{s}_0^2+1)^{3/2}}-\frac{8\hat{v}_a}{3}=0, \tag{6.18}$$

and Eq. (6.5) with $\hat{v}_l = \hat{v}_l^*(\kappa, N_C)$. Moreover, $\hat{s}_0 = \cot\theta_d^{cat}$, $\hat{s}_1 = -\cot\theta_c$ (Akbari *et al.*, 2015*a*), $Q_c^{cat} = 0$, and $\Lambda^{cat} = \Pi/\kappa$ with the superscript '*cat*' denoting variables in the catenoid limit. This state is specified by fixing κ and θ_c. Similarly to the initial phase, the rigid-membrane limit ($N_C \to \infty$) is determined by $\Lambda^{LIM} = \cot(\theta_d^{LIM}/2) = \Pi/\kappa$, where the membrane is not deformed by the capillary pressure.

Figure 6.8a shows the system state after the BBT at fixed θ_c in the catenoid limit. For a given θ_c, the catenoidal-bridge state separates (κ, N_C) on the left for which $Q_c < 0$ from those on the right for which $Q_c > 0$ after the transition. This contrasts with simply connected menisci, where Q_c is always negative. The membrane is deflected downward (upward) when Q_c is negative (positive), so (κ, N_C) on the right of the catenoid curves are desirable if collapse is to be prevented. Moreover, no equilibrium solution exists for the catenoid state in Fig. 6.8a when $\theta_c \lesssim 30°$. This is because at any given θ_c there is a slenderness Λ_{max} above which no catenoid exists (Akbari *et al.*, 2015*a*). Since $\Lambda_{max} \approx 0.2$ at $\theta_c = 30°$, and noting that

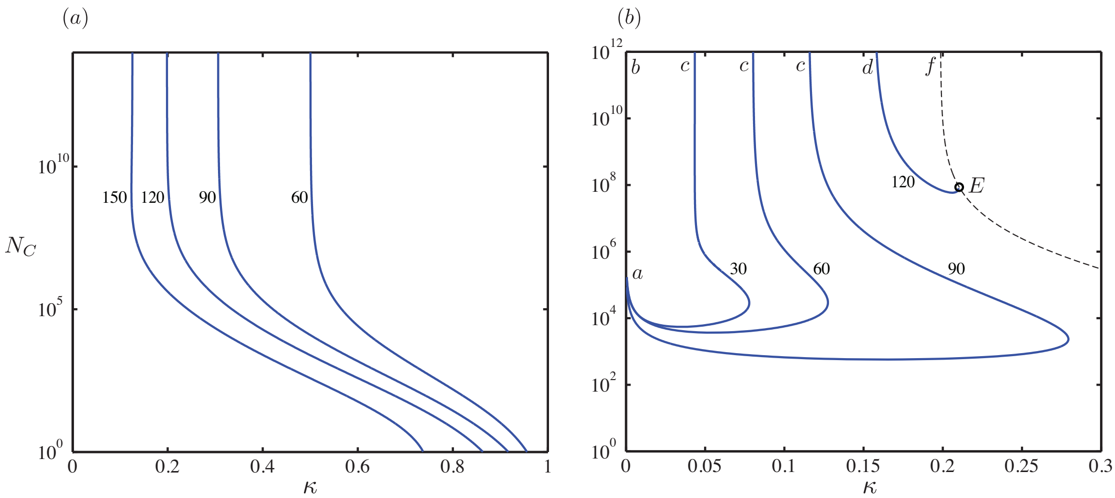


FIGURE 6.8: System state after the bubble to bridge transition with $\nu = 0.3$, $\Pi = 0.2$; the bridge after the transition is (a) a catenoid and (b) at the dihedral-angle stability limit ($\theta_d^{**} = \theta_c$) according to Eq. (6.6). Dashed line in the right figure indicates states where the bridge after the transition is a catenoid with $\theta_c = 120°$. Labels denote contact angle in degrees.

Λ_{max} monotonically decreases with decreaseing θ_c (Akbari *et al.*, 2015*a*), $\kappa_{min} = \Pi/\Lambda_{max} \gtrsim 1$ at $\Pi = 0.2$ when $\theta_c \lesssim 30°$, which is not physically possible. Therefore, hydrophilically modifying the membrane and plate enlarges the undesirable region of (κ, N_C), which can span the entire parameter space if θ_c becomes smaller than a critical value corresponding to a given Π.

Figure 6.8b shows the dihedral-angle stability limit at fixed θ_c with $\theta_d^{**} = \theta_c$. Here, N_C is a dependent variable, and the state is specified for a given κ and θ_c. For a fixed θ_c, there is a κ at which $N_C \to \infty$ at the stability limit, corresponding to the rigid-membrane limit (Fig. 6.7b). When $\theta_c < 90°$, (κ, N_C) in the complement of the region *acba* correspond to states in which the bridge after the transition satisfies Eq. (6.6). However, this behaviour changes when $\theta_c > 90°$. For example, at $\theta_c = 120°$, the curve corresponding to the dihedral-angle stability limit intersects the catenoid curve for the same contact angle at E. Here, (κ, N_C) in the region $dEfd$ correspond to states in which the bridge after the transition satisfies Eq. (6.6) and $Q_c < 0$. Interesting to note is the opposite behaviour of the dihedral-angle stability region for hydrophobic and hydrophilic surfaces. For hydrophobic surfaces, the stability region spans an area of the parameter space in which κ and N_C are large. As will be discussed in the next section, when the meniscus is a bridge, the elastocapillary system tends to lose stability at a volume turning point before collapse at large N_C. In contrast, for hydrophilic surfaces, decreasing θ_c enlarges the stability region, which spans a wide range of large and small κ and

N_C. Therefore, hydrophilically modifying the membrane and plate broadens the range of κ and N_C in which the dihedral-angle and volume-turning-point stability criteria are simultaneously satisfied during drying, thereby favoring collapsed conformations.

6.5.3 Final phase

Similarly to the initial phase, drying trajectories in this phase, where the meniscus is a bridge, describe how the system evolves with decreasing v_l. One must determine whether stability is lost from the state following the foregoing BBT to the collapsed state along drying trajectories. Thus, equilibrium branches, parametrized with v_l, must be constructed to identify volume turning points on drying trajectories. This is accomplished by solving Eqs. (6.4), (6.5), and (6.12)-(6.16), providing the dependent variables $(Q_c, \tau_0, \tau_1, a, \theta_d, \Lambda, \hat{v}_a)$ for a given $\hat{v}_l$, κ, N_C, and θ_c. We first consider the collapsed state, which is identified by $\Lambda = 0$, furnishing

$$Q_c^{col} = -\frac{\Pi N_C}{[K_w(1-\kappa^2)]^3}, \tag{6.19}$$

$$\cos\tau_0^{col} = \frac{-1+\tan\theta_c\sqrt{(a^{col})^2(1+\tan^2\theta_c)-1}}{1+\tan^2\theta_c}, \tag{6.20}$$

$$\hat{v}_l^{col} = \frac{3(1-\kappa^2)\Pi}{8\kappa^3}, \quad \text{and} \quad x_l^{col} = \frac{1}{2}(1-\kappa^2), \tag{6.21}$$

where a^{col} is a solution of

$$1+(a^{col})^2 + \frac{-2+2\tan\theta_c\sqrt{(a^{col})^2(1+\tan^2\theta_c)-1}}{1+\tan^2\theta_c} - (Q_c^{col})^2 = 0. \tag{6.22}$$

Note that, in the limit $\Lambda \to 0$, the meniscus is vanishingly small and the meniscus meridian curve is linear. Consequently, $\tau_1^{col} = \tau_0^{col}$, $\theta_d^{col} = \pi - \theta_c$, and $\hat{v}_a^{col} = 0$, where the superscript '*col*' denotes variables in the collapsed state. Furthermore, $\theta_d^{col} \leq \theta_c$ when $\theta_c \geq \pi/2$, implying that, according to Eq. (6.6), the collapsed state is always unstable for hydrophobic surfaces. Therefore, in the remainder of this section, we focus on hydrophilic surfaces with $\theta_c < \pi/2$.

At a fixed κ, there is a critical N_C below which Eq. (6.6) is satisfied along the entire drying trajectory, from the state after the BBT to the collapsed state. This critical N_C corresponds to a state, which we refer to as the dihedral-angle

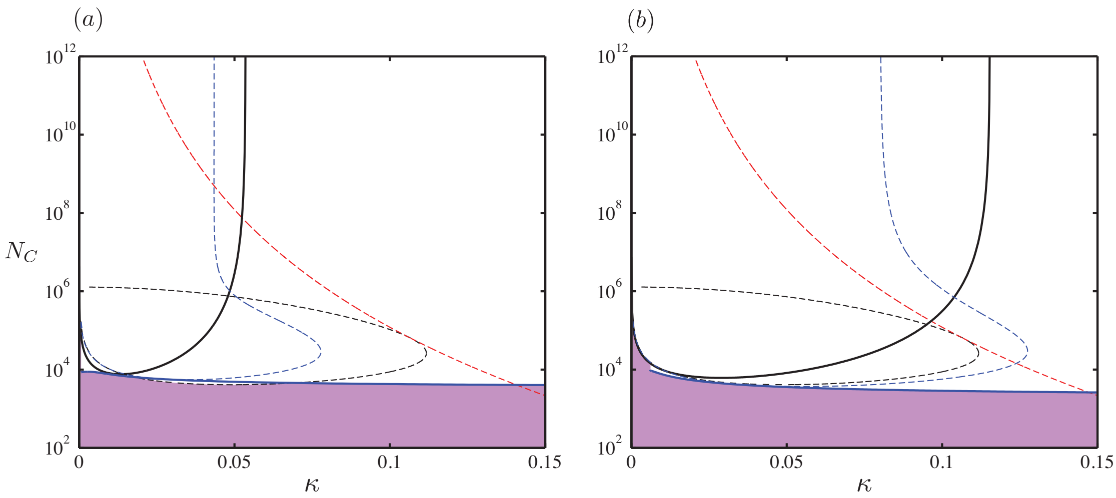


FIGURE 6.9: Phase diagrams providing upper bounds on the region of collapsed conformation in the parameter space from the locus of volume turning points coinciding with the tangent-bubble state (dashed black), volume inflection points when the meniscus is a bubble (dashed red), the dihedral-angle stability limit ($\theta_d = \theta_c$) according to Eq. (6.6) at the tangent-bubble state (solid black), after the bubble to bridge transition (dashed blue), and at dihedral-angle turning points when the meniscus is a bridge (solid blue) with $\nu = 0.3$, $\Pi = 0.2$ for (a) $\theta_c = 30°$ and (b) $\theta_c = 60°$. Shaded area indicates a region of the parameter space where the stability criteria are satisfied from the fully-saturated to collapsed state, except the volume-turning-point stability criterion for doubly connected menisci.

turning point limit, where $\theta_d = \theta_c$ at a dihedral-angle turning point where the dependent parameters $(Q_c^{dtp}, \tau_0^{dtp}, \tau_1^{dtp}, a^{dtp}, \Lambda^{dtp}, \hat{v}_a^{dtp}, \hat{v}_l^{dtp}, N_C)$ are determined by $(\partial\theta_d/\partial\hat{v}_l)_{\kappa,N_C,\theta_c} = 0$, Eqs. (6.4), (6.5), and (6.12)-(6.16). Here, the superscript '*dtp*' denotes variables in this state, which is specified for a given κ and θ_c. When the meniscus is a bridge, this state (solid blue lines in Fig. 6.9) provides a boundary in the parameter space, separating (κ, N_C) for which Eq. (6.6) is satisfied along the drying trajectory from those for which Eq. (6.6) is not satisfied on a segment of the drying trajectory. Overlaying this stability region with those constructed in Figs. 6.6 and 6.8b provides a region in the parameter space in which the volume-turning-point and dihedral-angle stability criteria are satisfied along drying trajectories from the fully-saturated to collapsed state, except the volume-turning-point stability criterion for doubly connected menisci. Moreover, one must ensure that all menisci are physically relizable along drying trajectories. The resulting region is, therefore, an upper bound on the region of collapsed conformations in the parameter space.

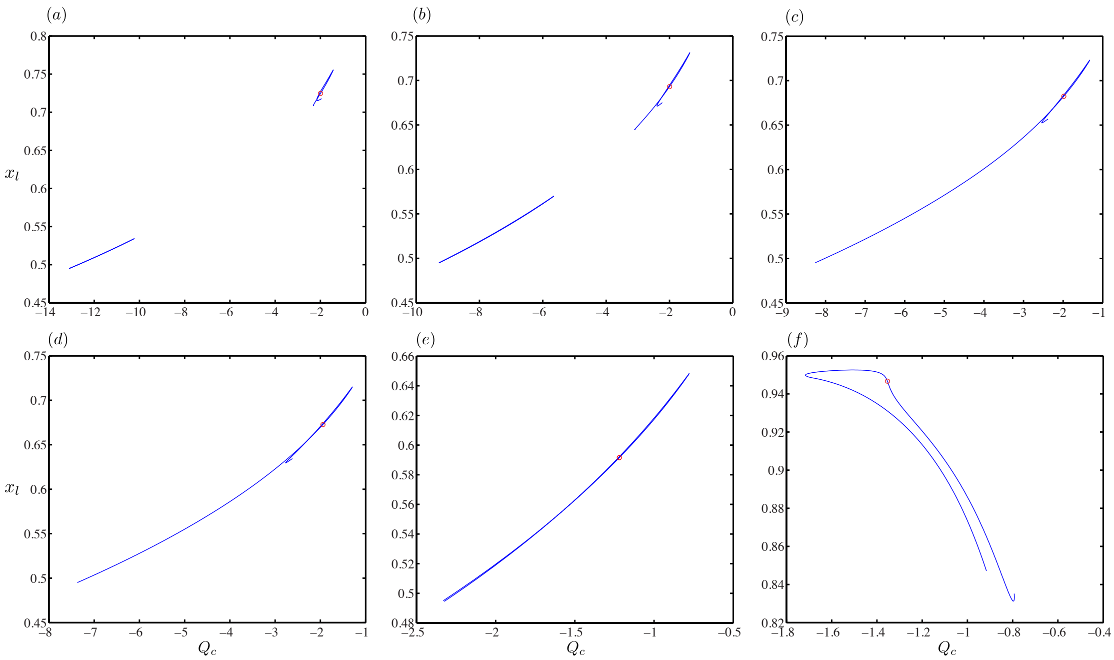


FIGURE 6.10: Drying trajectories in volume versus pressure diagrams for doubly connected menisci with $\nu = 0.3$, $\Pi = 0.2$, $\kappa = 0.1$, $\theta_c = 60°$, and (a) $\log N_C = 3.75$, (b) $\log N_C = 3.6$, (c) $\log N_C = 3.55$, (d) $\log N_C = 3.5$, (e) $\log N_C = 3$, (f) $\log N_C = 6.1$. Circles indicate the state after the bubble to bridge transition.

Figure 6.9 illustrates the forgoing region for $\theta_c = 30°$ and $60°$. The curve corresponding to the dihedral-angle turning point limit itself has a turning point in κ at $\kappa \simeq 0.0006$ and 0.0062 for $\theta_c = 30°$ and $60°$, respectively. When the meniscus is a bridge and below these values, Eq. (6.6) is satisfied along the entire drying trajectory for all N_C, and the dry-state conformation is determined only by the volume-turning-point stability criterion. This appears as a sudden jump in the boundary of the shaded areas in Fig. 6.9, from the dihedral-angle stability limit at the tangent-bubble state (solid black) to the dihedral-angle turning point limit (solid blue). Moreover, it is clear from Fig. 6.9 that, except for $\kappa \ll 1$, the stability region is not significantly influenced by the contact angle, and $N_C(\nu = 0.3, \Pi = 0.2) \sim 10^4$ furnishes an upper bound on the critical elastocapillary number, separating open and collapsed conformations the elastocapillary system of Fig. 6.1 assumes upon drying.

We illustrate how to examine the volume-turning-point stability criterion from the state after the BBT to the collapsed state by an example. Drying trajectories are shown in volume versus pressure and dihedral angle versus slenderness diagrams in Fig. 6.10 and 6.11 when $\nu = 0.3$, $\Pi = 0.2$, $\kappa = 0.1$, and $\theta_c = 60°$. The former is suitable for examining the volume-turning-point stability criterion since stability

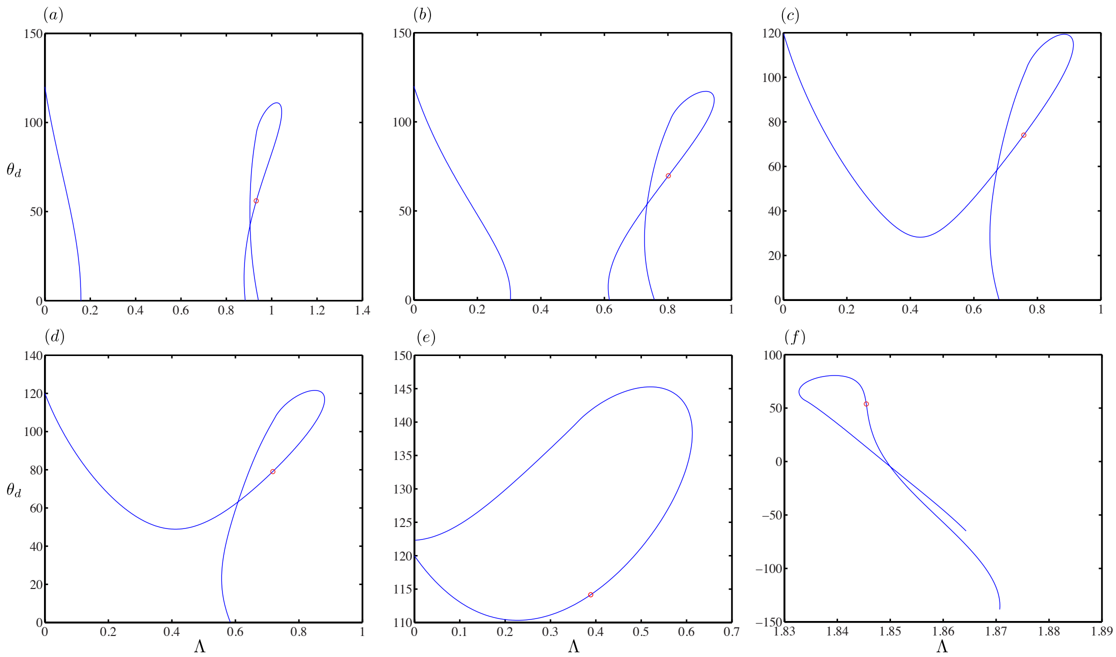


FIGURE 6.11: Drying trajectories in dihedral angle versus slenderness diagrams for doubly connected menisci with $\nu = 0.3$, $\Pi = 0.2$, $\kappa = 0.1$, $\theta_c = 60°$, and (a) $\log N_C = 3.75$, (b) $\log N_C = 3.6$, (c) $\log N_C = 3.55$, (d) $\log N_C = 3.5$, (e) $\log N_C = 3$, (f) $\log N_C = 6.1$. Circles indicate the state after the bubble to bridge transition.

can be related to the slope of drying trajectories (Akbari *et al.*, 2015*d*), whereas the latter is useful for ascertaining the dihedral-angle stability criterion along the drying trajectories. When $\kappa = 0.1$, the dihedral-angle stability limit after the BBT has the solutions $\log N_C \simeq 3.7538$ and 6.0523 (Fig. 6.8b), and the dihedral-angle turning point occurs at $\log N_C \simeq 3.4576$ (blue line in Fig. 6.9b). As shown in Fig. 6.10, the state after the BBT and collapsed state lie on disconnected branches at $\log N_C = 3.75$ and 3.6, and the system loses stability at a volume turning point before collapse. Decreasing the elastocapillary number to $\log N_C = 3.55$, the two disconnected segments merge, forming a continuous solution branch from the state after the BBT to the collapsed state. However, Eq. (6.6) is not satisfied on the middle part of the branch (Fig. 6.11c), and the contact line is detached from the hole edge before collapse. When the elastocapillary number drops below $\log N_C \simeq 3.4576$, for example at $\log N_C = 3$ (Fig. 6.10e and 6.11e), both stability criteria are satisfied simultaneously, and the system collapses upon drying. Therefore, in this example, the stability, and, thus, dry-state conformation of the system is determined by the dihedral-angle turning point limit.

Figures 6.10f and 6.11f show the drying trajectory when $\log N_C = 6.1$. Although the corresponding point in the parameter space lies in the dihedral-angle stability

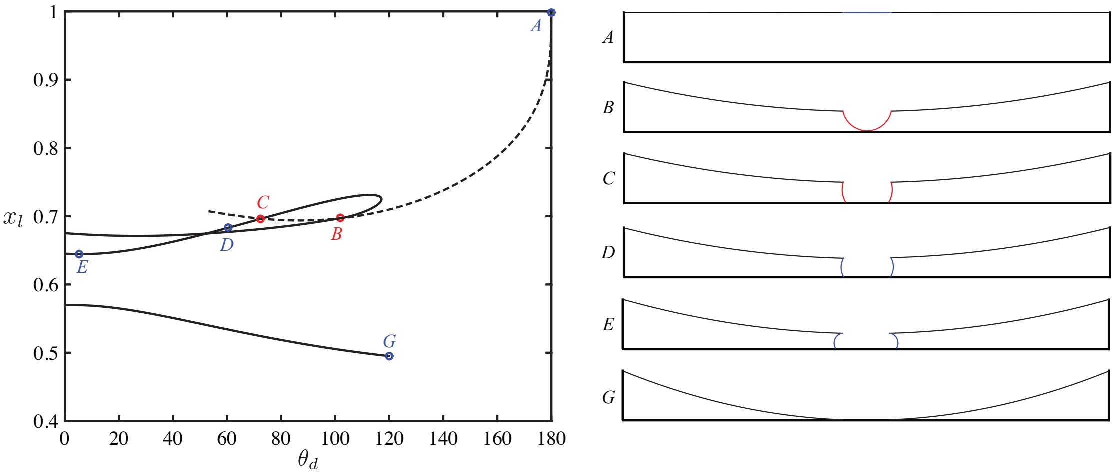


FIGURE 6.12: System conformations along a drying trajectory when the meniscus is a bubble (dashed) and bridge (solid) with $\nu = 0.3$, $\Pi = 0.2$, $\kappa = 0.1$, $\theta_c = 60°$, and $\log N_C = 3.6$.

region after the BBT (Fig. 6.8b), the state after the BBT is unstable because the corresponding point lies on an unstable branch, according to lemmas 2 and 3 of Akbari *et al.* (2015*d*). The dihedral-angle stability criterion is not satisfied along the entire branch, and the system does not evolve towards the collapsed state. In this case, stability is lost during the BBT.

We conclude this section by demonstrating how the elasticapillary system of Fig. 6.1 evolves along drying trajectories that lead to open and collapsed conformations upon drying. Figures 6.12 illustrate the drying trajectory at $\kappa = 0.1$, $\log N_C = 3.6$, and $\theta_c = 60°$. These parameters correspond to a point inside the shaded region in Fig. 6.6c and outside the shaded region in Fig. 6.9b. Starting from the fully-saturated state at A and removing water (decreasing x_l), the system follows the dashed line until the bubble reaches the plate at B, corresponding to the tangent-bubble state, where $\Lambda^* \simeq 0.8097$, $\theta_d^* \simeq 102°$, and $x_l^* \simeq 0.6970$. As predicted in Fig. 6.6c, the tangent-bubble state is on a stable branch, between the volume turning point and fully-saturated state, where Eq. (6.6) is satisfied. Then, the system undergoes a volume-preserving BBT ($x_l =$ const.) to the state C where $\Lambda^{**} \simeq 0.8165$ and $\theta_d^{**} \simeq 73°$, which also satisfies Eq. (6.6) (see Fig. 6.8b). According to lemmas 2 and 3 of Akbari *et al.* (2015*d*), this state is on a stable equilibrium branch (see Fig. 6.10b). Decreasing x_l below x_l^*, the system follows the solid line and reaches the dihedral-angle stability limit at D where $\Lambda \simeq 0.7601$, $\theta_d = 60°$, and $x_l \simeq 0.6825$. Beyond this point, the contact line does not remain pinned to the hole edge with decreasing x_l, as predicted in Fig. 6.9b. Note that the drying trajectory from C to the collapsed state at G is not continuous, and the system

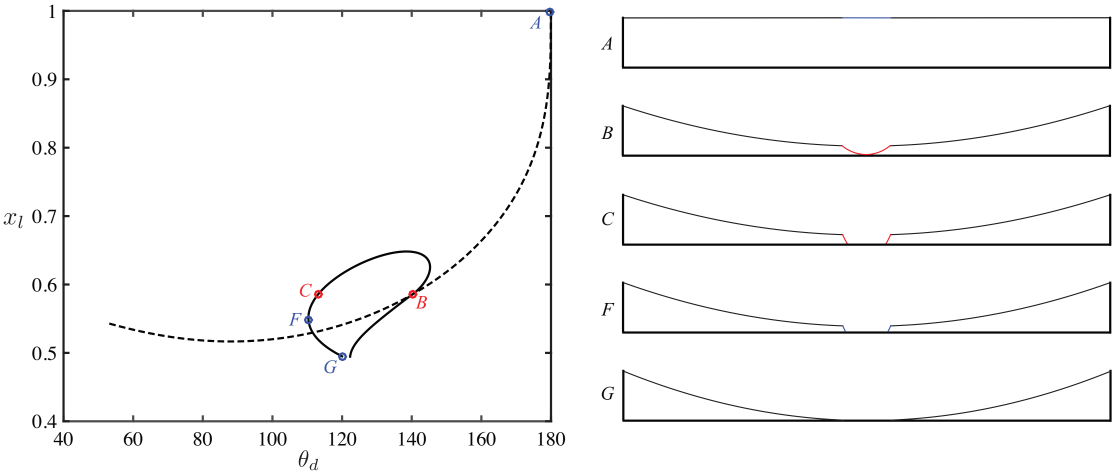


FIGURE 6.13: System conformations along a drying trajectory when the meniscus is a bubble (dashed) and bridge (solid) with $\nu = 0.3$, $\Pi = 0.2$, $\kappa = 0.1$, $\theta_c = 60°$, and $\log N_C = 3$.

loses stability at the volume turning point E before collapse, even if the contact line is maintained pinned to the hole edge by an external force. The dry-state conformation is consequently open in this case.

Figure 6.13 shows the drying trajectory at $\kappa = 0.1$, $\log N_C = 3$, and $\theta_c = 60°$. These parameters correspond to a point inside the shaded regions in Figs. 6.6c and 6.9b. Here, the system behaves similarly to the previous case (Fig. 6.12) until the point C. According to lemmas 2 and 3 of Akbari *et al.* (2015*d*), the corresponding state is stable because C lies on a stable equilibrium branch (see Fig. 6.10e). However, the drying trajectory has a turning point in θ_d at F and no turning point in v_l from C to the collapsed state at G. As predicted in Fig. 6.9b, $\theta_d > \theta_c$ along the entire trajectory from the fully-saturated to collapsed state, and stability is not lost with respect to the dihedral-angle stability criterion. Moreover, the drying trajectory continuously connects C to G, implying that both stability criteria are satisfied along the entire trajectory from A to G, and the dry-state conformation is collapsed. Note also that in Figs. 6.12 and 6.13, all menisci are physically realizable from the state after the BBT at C to the collapsed state at G.

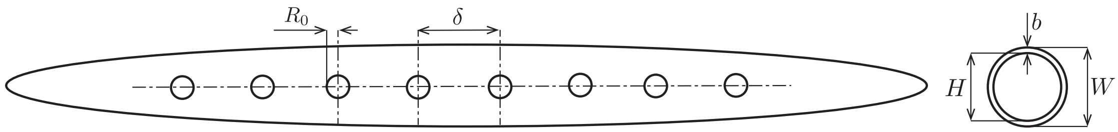


FIGURE 6.14: Schematic representation of wood fibres, showing characteristic length scales.

6.6 Wood-fibre lumen collapse

Here, we apply the elstocapillary model to lumen collapse of wood fibres, providing a quantitative estimate of the critical N_C, determining the dry-state conformation. The model has the length scales R_0, R, and H, which we adopt to characterize the meniscus curvature, deformation length, and maximum deflection. Wood-fibre geometry, mechanical properties, and ultrastructure have been extensively studied for various species in the literature in the past few decades (Walker, 2006). Here, we choose Norway spruce as an example with which to estimate the foregoing length scales (see Fig. 6.14) and approximate the critical N_C.

Cristian Neagu *et al.* (2006) reported the average lumen width $W = H + 2b \approx 20\ \mu$m for early and late wood, and Yan & Li (2008) measured the elasticity $E \approx 1.4$ GPa and wall thickness $b \approx 1.4\ \mu$m for Kraft spruce. Sirviö & Kärenlampi (1998) reported the normalized size of bordered pits $R_0/H \approx 0.075$ with an average longitudinal spacing $\delta \approx 10\ \mu$m. Taking $R \approx \delta/2$, we find $\kappa \approx 0.22$, $\Pi \approx 3.4$, and $N_C \approx 30500$. However, from Fig. 6.9, $N_C \approx 3700, 2300$ are the upper bounds on elastocapillary numbers for which the dry-state conformation is collapsed when $\Pi = 0.2$ and $\theta_c = 30°, 60°$. Noting that the critical N_C decreases with increasing Π, it follows that the capillary pressure of menisci, having length scales of the same order as the pit-hole diameter and lumen width, is small by at least an order of magnitude to overcome the cell-wall stretching resistance and induce collapse.

6.7 Concluding remarks

We have developed an elastocapillary model of wood-fibre collapse upon drying to predict the dry-state conformation from the cell-wall geometrical and mechanical properties. This model simulates drying-induced structural deformations arising in cavities with three characteristic length scales, providing a low-dimensional model to describe the collapse of lumen or cell-wall micropores at moisture contents above the fibre saturation point. The dry-state conformation was determined based on the dihedral-angle and volume-turning-point stability criteria. Fixing the cavity aspect ratio, an upper bound on the critical elastocapillary number, corresponding to a scaled hole radius and contact angle, was provided, above which the system does not collapse upon drying. The critical elastocapillary number weakly depends on the contact angle in a wide range of the scaled hole radius, indicating a limitation of surface hydrophobization for controlling the dry-state conformation in pulping processes.

Applying the model to lumen collapse based on the structural and mechanical properties of Norway spruce fibres revealed that the capillary pressure of menisci, spanning pit-hole openings and/or the lumen width, is not strong enough to overcome the cell-wall stretching resistance, so is unlikely to cause collapse. However, the capillary-induced collapse of cell-wall micropores is yet to be understood. Granted, this is more challenging because cell-wall microstructural properties, such as pore geometry and pore-size distribution have not been established in the literature. Nevertheless, the nonlinear trend of the overall volumetric shrinkage versus moisture content observed above the fibre saturation point (Hernández & Pontin, 2006) as compared to the linear trend below the fibre saturation point can serve as a suggestive piece of evidence. This nonlinear trend signifies deformations in the collapse regime and may be attributed to the capillary-induced collapse of micropores in a transition phase of drying when the moisture content approaches the fibre saturation point from above. Understanding the transition between the collapse and shrinkage regimes during drying demands reproducible and reliable characterizations of cell-wall micro-structural properties to provide an accurate prediction of the dry-state conformation.

We considered an idealized model of drying-induced collapse in wood fibres, capturing the basic elements of the underlying physics. The bending contribution to the total potential energy and membrane profile were neglected, limiting the accuracy for thick-walled and slender cavities. Accounting for the bending energy requires a more complicated test function for the membrane deflection in the bending- and stretching-dominated regimes. However, implementing complicated test functions is computationally expensive and algorithmically challenging. Moreover, our model does not account for geometrical complexities of cavities in wood fibres. Specifically, the lumen and cell-wall micropores have a closed structure, exhibiting a buckling resistance to compressive loads. However, the membrane in the present model can continuously deform from the fully-saturated to collapsed state without buckling. The critical elastocapillary number is consequently smaller than predicted in this work. Nevertheless, our results furnish an upper bound on the critical elastocapillary number and can reasonably estimate the range of parameters for which the dry-state conformation is open. More realistic representative geometries improve quantitative predictions at the expense of higher computational cost. We hope that the present study motivates further investigations into the underlying mechanisms of wood-water interactions, providing a better understanding of drying deformations in the shrinkage and collapse regimes in the entire moisture-content range.

Chapter 7

Conclusions

7.1 Summary

Wood fibres are cellulose composites that exhibit complex behaviour with water. They deform to a degree and assume a broad range of conformations upon drying, depending on the cell-wall geometry, microstructure, and constituent composition. These, in turn, depend on raw-fibre characteristics and the pulping process. Shrinkage and collapse are distinct deformation regimes that are dominant below and above the fibre saturation point, respectively. Wood-fibre collapse upon drying was theoretically studied in this thesis. An elastocapillary model was developed to quantify drying deformations in the collapse regime, where the moisture contend is above the fibre-saturation point. In this regime, water is held in the cell lumen and cell-wall micropores, and conformational changes are controlled by interactions between capillary and elastic forces. Capillary forces arise due to the formation of simply connected (bubble) and doubly connected (bridge) menisci spanning apertures and walls of wood-fibre cavities. The dry-state conformation is determined from the stability of equilibrium states along solution branches, and the model predicts open conformation if stability is lost during drying.

A variety of capillary surfaces arises during drying, depending on the geometry of cavities and apertures. Surface characteristics and the contact line position determine whether these menisci are pinned or free at their boundaries with cavity walls. Contact lines tend to be free on hydrophobic and smooth surfaces, whereas, on hydrophilic surfaces and sharp edges, they are usually pinned. In this thesis, axisymmetric bubbles and bridges under zero gravity were studied as idealized geometries for simply connected and doubly connected menisci. First, the stability of bridges was examined in isolation from the elastic responses of cavity walls by omitting the elastic strain energy in the total energy and fixing the bridge slenderness. Next, interactions between meniscus and wall deformations at contact lines and through the capillary pressure were accounted for in the total energy to determine stability at fixed liquid content.

The stability of catenoids with a free contact line was studied in chapter 2. Catenoids are the limiting cases of liquid bridges when the mean curvature approaches zero. The results showed that all catenoids are stable with respect to non-axisymmetric perturbations; for a fixed contact angle, there is a critical volume below which catenoids lose stability to axisymmetric perturbations. The existence, maximal stability, and stability regions were constructed in the dihedral angle versus contact angle (canonical) and volume versus slenderness (favourable) diagrams. The former proved more convenient for representing the stability region because they uniquely specify equilibria, and the critical states split the parameter space into mutually exclusive regions of stability and instability. Particularly, representing the stability region in the canonical diagram revealed the symmetry breaking effect of the free contact line. Moreover, the stability region was shown to be contained in the maximal stability region, demonstrating the destabilizing effect of free contact lines.

Constant-volume and constant-pressure stability regions for liquid bridges with a free contact line were constructed in slenderness versus cylindrical-volume diagrams at fixed contact angles in chapter 3. Equilibrium branches were classified using the wavenumber of pieces-of-sphere states, emphasizing the symmetry-breaking role of the free contact line. Critical perturbations and bifurcations along the stability-region boundaries were determined. Compared to liquid bridges pinned at two equal discs, the free contact line completely breaks the equatorial symmetry and changes the nature of bifurcations on the stability-region boundaries where axisymmetric perturbations are the most dangerous. The lower boundary of the stability region corresponds to critical states where stability loss leads to breakup, and it is, therefore, relevant to contact-drop dispensing. The maximum- and minimum-slenderness stability limits were experimentally ascertained under neutral buoyancy in chapter 4. The contact-line role in the stability of liquid bridges was isolated, demonstrating the destabilizing effect of free contact lines.

An elastocapillary model was developed in chapter 5 to study drying-induced structural deformations. The model comprises a circular elastic membrane with a hole at the canter that is deformed by the capillary pressure of simply and doubly connected menisci. The membrane covers a cylindrical cavity with rigid walls from the top, trapping a prescribed volume of liquid. Using spectral and variational methods, constant-pressure and constant-volume stability were related to the slope of equilibrium branches. These stability criteria provide computationally efficient alternatives for standard dynamic and linear stability techniques. It was demonstrated that the stability of the membrane and meniscus alone is not sufficient to determine the stability of the elastocapillary system as whole; interactions between the membrane and meniscus through the capillary pressure and their shared boundaries also play an important role. These criteria were applied to study the collapse of cell-wall micropores and the lumen above the FSP in chapter 6. For lumen collapse, the model captures three characteristic length scales, namely, the pit-hole radius, pit-hole spacing, and fibre width. An elastocapillary number was introduced to measure the axial rigidity relative to the liquid surface tension. Upper bounds on the critical elastocapillary number were determined as functions of the contact angle, scaled hole radius, and cavity aspect ratio, providing ranges of geometrical and mechanical parameters in which the dry-state conformation is open. It was shown that the contact angle weakly influences the critical elastocapillary number in a wide range of the scaled hole radius, indicating a limitation of surface hydrophobization for controlling the dry-state conformation in pulping processes. Applying the elastocapillary model to Norway spruce revealed that the capillary driven collapse of wood cavities over the length scales of the lumen and pit-holes is unlikely to contribute to the overall volumetric shrinkage.

7.2 Contribution to knowledge

The main achievements of this dissertation are summarized below:

- The stability of axisymmetric liquid bridges with a free contact line with respect to arbitrary perturbations was theoretically studied. Free contact lines are subject to a larger class of perturbations than pinned contact lines. Consequently, liquid bridges with free contact lines are more likely to lose stability than those with pinned contact lines for the same control parameters, as demonstrated in this thesis. Examining the stability in a wide range of contact angles, the effect of surface hydrophobization, a practical technique for reducing the receding contact angle and achieving free contact lines, was elucidated.

- Constant-volume and constant-pressure stability regions were determined for liquid bridges with a free contact line in a wide range of contact angles. Bifurcations along the stability-region boundaries were characterized. It was shown that, along the lower boundary of the constant-volume stability region where axisymmetric perturbation are the most dangerous, stability is lost at turning points. This contrasts with liquid bridges pinned at two equal discs where stability is lost at pitchfork bifurcations along the lower boundary, revealing the symmetry-breaking effect of the free contact line.

- The maximum- and minimum-volume stability limits of liquid bridge with a free contact line were experimentally ascertained, and the destabilizing effect of free contact lines was demonstrated.

- An elastocapillary model was developed to quantify drying deformations above the FSP and predict the dry-state conformation of cell-wall micropores and the cell lumen. It was shown that capillary surfaces, spanning pit-hole openings and/or the lumen width, are unlikely to be responsible for lumen collapse.

- The constant-volume and constant-pressure stability of the elastocapillary model was rigorously related to the slope of equilibrium branches in the volume versus pressure diagram. On a microscopic level, it was shown that the critical states of the elastocapillary system as a whole does not generally coincide with those of the meniscus or membrane due to interactions between the membrane and meniscus perturbations through the boundaries and capillary pressure. However, on a macroscopic level, it was proved that constant-volume stability is lost at turning points in the liquid volume, suggesting that the principle of stability exchange in catastrophe theory has a broad scope in mechanics: In pure elastic and capillary problems, stability is lost at turning points in the control parameter such that critical states and critical perturbations are consistent with the respective constraints and boundary conditions. Extending this rule to elastocapillary systems does not seem immediate at first sight because constraints on equilibrium states and perturbations, boundary conditions, and the control parameter can arbitrarily change, depending on interplays between elastic and capillary forces. However, the principle of stability exchange states that the equilibrium branches of the elastocapillary system change accordingly so that stability is still lost at turning points in the control parameter, where critical states and critical perturbations satisfy the constraints and boundary conditions of the elastocapillary problem.

Appendix A

Stability of catenoids with respect to non-axisymmetric perturbations

We shall prove that all catenoids are stable with respect to non-axisymmetric perturbations. First, we show that $D^1(\hat{s}_0, \hat{s}_1) = 0$ has no non-trivial root. Substituting Eqs. (2.39)-(2.43) into Eq. (2.21) yields

$$\hat{s}_1\sqrt{\hat{s}_1^2+1} + \ln(\hat{s}_1 + \sqrt{\hat{s}_1^2+1}) = \hat{s}_0\sqrt{\hat{s}_0^2+1} + \ln(\hat{s}_0 + \sqrt{\hat{s}_0^2+1}), \tag{A.1}$$

where we seek an $\hat{s}_0 \in (-\infty, \hat{s}_1)$ for a fixed $\hat{s}_1$. Equation (A.1) is obviously satisfied for the trivial solution $\hat{s}_0 = \hat{s}_1$. Denoting the left-hand side by $f(\hat{s}_1)$, Eq. (A.1) can be rewritten as

$$g(\hat{s}_0) = \hat{s}_0\sqrt{\hat{s}_0^2+1} + \ln(\hat{s}_0 + \sqrt{\hat{s}_0^2+1}) - f(\hat{s}_1) = 0 \tag{A.2}$$

with

$$\frac{dg}{d\hat{s}_0} = \sqrt{\hat{s}_0^2+1} + \frac{\hat{s}_0^2}{\sqrt{\hat{s}_0^2+1}} + \frac{1}{\sqrt{\hat{s}_0^2+1}}. \tag{A.3}$$

All the terms on the right-hand side of Eq. (A.3) are positive, indicating that $dg/d\hat{s}_0 > 0$ on $(-\infty, \hat{s}_1)$. The continuity of $g(\hat{s}_0)$ implies that g is a monotonically increasing function such that $g \to -\infty$ as $\hat{s}_0 \to -\infty$ and $g \to 0^-$ as $\hat{s}_0 \to \hat{s}_1$. Therefore, $\hat{s}_0 = \hat{s}_1$ is the only solution of Eq. (A.2).

Next, we prove that $\hat{\chi}^1(\hat{s}_0, \hat{s}_1) - \hat{\chi}(\hat{s}_1) = 0$ has no non-trivial root. Substituting Eqs. (2.39)-(2.43) into Eq. (2.18) results in

$$\frac{\hat{s}_1}{\hat{s}_1^2+1}\left[\frac{\frac{\sqrt{\hat{s}_1^2+1}(\hat{s}_1^2+2)}{\hat{s}_1} - \sinh^{-1}\hat{s}_1 + \hat{s}_0\sqrt{\hat{s}_0^2+1} + \sinh^{-1}\hat{s}_0}{\hat{s}_1\sqrt{\hat{s}_1^2+1} + \sinh^{-1}\hat{s}_1 - \hat{s}_0\sqrt{\hat{s}_0^2+1} - \sinh^{-1}\hat{s}_0} + 1\right] = 0. \tag{A.4}$$

Note that the denominator of the fraction in the square bracket is non-zero for $\hat{s}_0 \in (-\infty, \hat{s}_1)$. Two casesmust be considered separately: (1) $\hat{s}_1 \neq 0$ and (2) $\hat{s}_1 \to 0$. The first leads to

$$\sqrt{\hat{s}_1^2+1} + (\hat{s}_1^2+1)^{3/2} + \hat{s}_1^2\sqrt{\hat{s}_1^2+1} = 0. \tag{A.5}$$

A pair $(\hat{s}_0, \hat{s}_1)$ that satisfies Eq. (A.4) must also satisfy Eq. (A.5). Equation (A.5) clearly shows that Eq. (A.4) has no non-trivial root since it is independent of $\hat{s}_0$. In addition, no $\hat{s}_1$ can be found that satisfies Eq. (A.5) because the left-hand side is always greater than zero. From the second case,

$$\frac{2}{\hat{s}_0\sqrt{\hat{s}_0^2+1} - \sinh^{-1}\hat{s}_0} = 0, \tag{A.6}$$

which holds only when $\hat{s}_0 \to -\infty$. This indicates that, for any given contact angle, only infinitely large catenoids may lose stability to non-axisymmetric perturbations, and the stability-region boundary coincides with the existence-region boundary in the canonical phase diagram.

One can apply the same procedure as the two previous cases to prove that $D^0(\hat{s}_0, \hat{s}_1) = 0$ has no non-trivial root for $\theta_c \leq \pi/2$. However, the expressions are cumbersome and the analysis is tedious. We only demonstrate the limiting behaviour discussed in section 2.5.2.2. The Taylor-series expansion of D^0 is used for the small-interface limit:

$$D^0 = -\frac{\epsilon^4}{12} + O(\epsilon^5), \qquad \epsilon \ll 1, \tag{A.7}$$

where $\epsilon = \hat{s}_1 - \hat{s}_0$. It follows that

$$\lim_{\epsilon\to 0^+} D^0 = 0^-. \tag{A.8}$$

In the other limit, where catenoids are infinitely large, one can show that

$$\lim_{\hat{s}_0\to -\infty} D^0 = \mathrm{sign}(\hat{s}_1) \times \infty, \tag{A.9}$$

implying that there is at least one $\hat{s}_0 \in (-\infty, \hat{s}_1)$ at which $D^0(\hat{s}_0, \hat{s}_1) = 0$ for $\theta_c > \pi/2$. These limits are clearly illustrated in Fig. 2.5.

Appendix B

Symmetry of D-functions for catenoids

We prove that D^0 and D^1 are symmetric with respect to the canonical phase diagram minor diagonal. Consider the following transformation

$$\begin{cases} \hat{s}_0 = -\bar{\hat{s}}_1 \\ \hat{s}_1 = -\bar{\hat{s}}_0 \end{cases}, \tag{B.1}$$

which is equivalent to Eq. (2.44). Given that w_1, w_4 are odd and r, w_2, w_3, w_5 are even functions, we have

$$D^0(\bar{\hat{s}}_0, \bar{\hat{s}}_1) = \begin{vmatrix} -w_1(\hat{s}_1) & w_2(\hat{s}_1) & w_3(\hat{s}_1) \\ -w_1(\hat{s}_0) & w_2(\hat{s}_0) & w_3(\hat{s}_0) \\ -\int_{\hat{s}_0}^{\hat{s}_1} \hat{r} w_1 d\hat{s} & \int_{\hat{s}_0}^{\hat{s}_1} \hat{r} w_2 d\hat{s} & \int_{\hat{s}_0}^{\hat{s}_1} \hat{r} w_3 d\hat{s} \end{vmatrix}, \tag{B.2}$$

$$D^1(\bar{\hat{s}}_0, \bar{\hat{s}}_1) = \begin{vmatrix} -w_4(\hat{s}_1) & w_5(\hat{s}_1) \\ -w_4(\hat{s}_0) & w_5(\hat{s}_0) \end{vmatrix}. \tag{B.3}$$

Taking the determinant row exchange rules into consideration, it follows that

$$D^0(\bar{\hat{s}}_0, \bar{\hat{s}}_1) = D^0(\hat{s}_0, \hat{s}_1), \tag{B.4}$$

$$D^1(\bar{\hat{s}}_0, \bar{\hat{s}}_1) = D^1(\hat{s}_0, \hat{s}_1). \tag{B.5}$$

This completes the proof.

Appendix C

Normal variations for simply connected menisci

Expressing perturbations in $\bar{\xi}_1$ is problematic for axisymmetric simply connected menisci because $\bar{\xi}_1 \notin L^2$. Assuming that perturbed states are also axisymmetric (*i.e.*, $\mathrm{d}z/\mathrm{d}r = 0$ at $\hat{z} = \hat{\ell}$), we have

$$\frac{\mathrm{d}z}{\mathrm{d}r} = \frac{1 + \eta_1'\varepsilon + \cdots}{\hat{r}' + \xi_1'\varepsilon + \cdots} = \frac{1}{\hat{r}'} - \frac{\bar{\xi}_1' + \eta_1\hat{r}''}{\hat{r}'^2}\varepsilon + \cdots = 0 \quad \text{at} \quad \hat{z} = \hat{\ell} \tag{C.1}$$

in view of Eqs. (5.12) and (5.13). Since $\hat{r}'(\hat{\ell}) \to \infty$ and $\xi_1(\hat{\ell}), \eta_1(\hat{\ell}), \hat{r}''(\hat{\ell}) = $ finite, it follows that $|\bar{\xi}_1'(\hat{\ell})| \sim O(\hat{r}') \to \infty$ satisfies the axial symmetry condition. Moreover, From Eq. (5.21), $|\bar{\xi}_1(\hat{\ell})| \to \infty$. Therefore, $\bar{\xi}$ is unbounded, and, thus, unsuitable for representing perturbations when the meniscus is a bubble. Here, representing functional variations with respect to the normal variations of menisci resolves the issue.

When the meniscus is a bubble (sphere), it is convenient to express variables as functions of the meridian-curve arclength $\hat{s}$ or the polar angle $\theta = \hat{s}/R_s$ (see Fig. 5.2). The displacement vector from the meniscus equilibrium states to its perturbed states is written

$$\frac{\delta \mathbf{x}}{\varepsilon} = \xi_1 \hat{\mathbf{r}} + \eta_1 \hat{\mathbf{z}}. \tag{C.2}$$

Given $\hat{\mathbf{n}} = \sin\theta \hat{\mathbf{r}} - \cos\theta \hat{\mathbf{z}}$, the normal variation

$$N = \hat{\mathbf{n}} \cdot \frac{\delta \mathbf{x}}{\varepsilon} = \xi_1 \sin\theta - \eta_1 \cos\theta \tag{C.3}$$

is obtained, furnishing

$$\bar{\xi}_1 = \frac{N}{\sin\theta}, \tag{C.4}$$

resulting in $\mathrm{d}N/\mathrm{d}\hat{s} = -\mathrm{d}\eta_1/\mathrm{d}\hat{s}$ at $\hat{z} = \hat{\ell}$. Moreover, $\eta_1' \to \xi_1'/\hat{r}'$ and $\mathrm{d}\eta_1/\mathrm{d}\hat{s} \to \xi_1'/\hat{r}'\sqrt{1+\hat{r}'^2}$ as $\hat{z} \to \hat{\ell}$, giving $\mathrm{d}\eta_1/\mathrm{d}\hat{s}(\hat{\ell}) \to 0$ even if $|\xi_1'| \sim O(\hat{r}')$. Therefore, $\mathrm{d}N/\mathrm{d}\hat{s} = 0$ and $N =$ finite at $\hat{z} = \hat{\ell}$, implying $N \in L^2$.

The following formulas are useful for representing the meridian curve when the meniscus is a bubble:

$$\hat{r} = \sqrt{R_s^2 - (z_c - \hat{z})^2} = R_s \sin\theta, \tag{C.5}$$

$$\hat{r}' = \frac{z_c - \hat{z}}{\hat{r}} = \cot\theta, \tag{C.6}$$

$$\hat{r}'' = -\frac{1+\hat{r}'^2}{\hat{r}} = -\frac{1}{R_s \sin^3\theta}, \tag{C.7}$$

furnishing

$$\mathcal{P}^{(\hat{r})} = R_s \gamma_{gl} \sin^4\theta, \quad \mathcal{Q}^{(\hat{r})} = -\frac{\gamma_{gl}}{R_s}, \quad \mathcal{A} = \frac{\gamma_{gl}\sin^2\theta}{\cos\theta}, \tag{C.8}$$

where z_c and R_s are the z-coordinate at the center and radius of the sphere.

Appendix D

Functional differentiation along equilibrium branches

Consider the functional

$$J[y] = \int_{x_a}^{x_b} F(x, y, y')\mathrm{d}x, \quad J : L^2 \to \mathbb{R}, \quad y : [x_a, x_b] \to \mathbb{R}, \tag{D.1}$$

of continuously differentiable functions y defined on a variable domain where the branches of stationary points are parametrized with t, and the stationary points are represented by $\hat{y} = \hat{y}(\hat{x}, t)$. Then, differentiating the functional and its first Fréchet derivative along a branch furnishes

$$\dot{J} = \left\langle J'_{(\hat{y})}, \dot{\hat{y}} \right\rangle + \left[\dot{\hat{y}} F_{y'} + \dot{\hat{x}} F \right]_{\hat{x}_a}^{\hat{x}_b}, \tag{D.2}$$

$$\dot{J}'_{(\hat{y})} = J''_{(\hat{y})}[\dot{\hat{y}}], \tag{D.3}$$

where

$$J''_{(\hat{y})}[\varphi] = -\frac{\mathrm{d}}{\mathrm{d}\hat{x}} \left(F_{y'y'} \frac{\mathrm{d}\varphi}{\mathrm{d}\hat{x}} \right) + \left(F_{yy} - \frac{\mathrm{d}}{\mathrm{d}\hat{x}} F_{yy'} \right) \varphi, \quad J''_{(\hat{y})} : L^2 \to L^2. \tag{D.4}$$

Appendix E

Variational approximation for circular membranes in the stretching regime

The in-plane and out-of-plane equations of equilibrium for the membrane in the elastocapillary model of Fig. 6.1 are (Akbari *et al.*, 2015*d*)

$$r_p N'_{rr} + N_{rr} - N_{tt} - P(r_p + u)w' = 0, \tag{E.1}$$

$$(N_{rr} w' r_p)' + P(r_p + u)(1 + u') = 0 \tag{E.2}$$

with boundary conditions

$$N_{rr} = 0 \quad \text{at} \quad r_p = R_{00}, \tag{E.3}$$

$$u = 0, \quad w = 0 \quad \text{at} \quad r_p = R, \tag{E.4}$$

where

$$N_{rr} = C(\tilde{u}' + \tilde{w}'^2/2 + \nu\tilde{u}/\tilde{r}_p), \tag{E.5}$$

$$N_{tt} = C(\nu\tilde{u}' + \nu\tilde{w}'^2/2 + \tilde{u}/\tilde{r}_p). \tag{E.6}$$

Since we focus on cases where $\Pi \ll 1$, we have $w', u \ll 1$, and the capillary pressure can be regarded as acting in the z-direction, furnishing

$$r_p N'_{rr} + N_{rr} - N_{tt} = 0, \tag{E.7}$$

$$(N_{rr}w'r_p)' + Pr_p = 0. \tag{E.8}$$

Equations (E.7) and (E.8) are nonlinear and do not have tractable closed-form solutions. Note that the nonlinearity can be removed if N_{rr} is known in Eq. (E.8). Thus, as a first approximation, w is estimated from the out-of-plane equilibrium by substituting the radial distribution of the axial force from a simpler problem, namely, undeflected circular plates under boundary tension, such that the boundary conditions Eqs. (E.3) and (E.4) are satisfied. The radial and tangential components of the axial force in this problem are (Timoshenko & Goodier, 1951)

$$N_{rr} = \frac{A}{r_p} + B, \tag{E.9}$$

$$N_{tt} = -\frac{A}{r_p} + B \tag{E.10}$$

with $A = -TR^2R_{00}^2/(R^2 - R_{00}^2)$ and $B = TR^2/(R^2 - R_{00}^2)$, where T is the radial tension at $r_p = R$. Solving Eq. (E.8) using the radial component of the axial force in Eq. (E.9) furnishes

$$\tilde{w}(\tilde{r}_p) = \tilde{w}_0(1 - \tilde{r}_p^2). \tag{E.11}$$

The radial distribution of the in-plane displacement $\tilde{u}(\tilde{r}_p)$, given by Eq. (6.2), is obtained by solving Eq. (E.7) using Eq. (E.11).

We use Eq. (E.11) as a test function to derive a variational approximation for the membrane equilibrium. The total potential energy Ω_T comprises the membrane stretching energy

$$\Omega_S = \frac{2\pi C}{2}\int_{R_{00}}^{R}\left(u'^2 + u'w'^2 + \frac{2\nu uu'}{r_p} + \frac{\nu uw'^2}{r_p} + \frac{u^2}{r_p^2} + \frac{w'^4}{4}\right)r_p\mathrm{d}r_p, \tag{E.12}$$

and the work of the capillary pressure

$$\Omega_P = -2\pi\int_{R_{00}}^{R} Pwr_p\mathrm{d}r_p. \tag{E.13}$$

Introducing the dimensionless energies $\Omega_i^* = \Omega_i/(2\pi CR^2)$ and the scaling forms

$$\tilde{w}(\tilde{r}_p) = \tilde{w}_0\bar{w}(\tilde{r}_p), \tag{E.14}$$

$$\tilde{u}(\tilde{r}_p) = \tilde{w}_0^2\bar{u}(\tilde{r}_p), \tag{E.15}$$

$$\Omega_S^* = \tilde{w}_0^4 \bar{\Omega}_S, \tag{E.16}$$

$$\Omega_P^* = \tilde{w}_0 \bar{\Omega}_P, \tag{E.17}$$

the total potential energy can be written

$$\Omega_T^* = \tilde{w}_0^4 \bar{\Omega}_S + \tilde{w}_0 \bar{\Omega}_P, \tag{E.18}$$

where

$$\bar{\Omega}_S = \frac{1}{2} \int_\kappa^1 \left(\bar{u}'^2 + \bar{u}' \bar{w}'^2 + \frac{2\nu \bar{u} \bar{u}'}{\tilde{r}_p} + \frac{\nu \bar{u} \bar{w}'^2}{\tilde{r}_p} + \frac{\bar{u}^2}{\tilde{r}_p^2} + \frac{\bar{w}'^4}{4} \right) \tilde{r}_p \mathrm{d}\tilde{r}_p, \tag{E.19}$$

$$\bar{\Omega}_P = -\frac{Q_c}{\kappa N_C} \int_\kappa^1 \bar{w} \tilde{r}_p \mathrm{d}\tilde{r}_p. \tag{E.20}$$

Substituting $\bar{u}$ and $\bar{w}$ from Eqs. (6.1) and (6.2) in Eqs. (E.19) and (E.20) leads to

$$\bar{\Omega}_S = \frac{(1-\kappa^2)^3 (1-\nu^2)[7-\nu+\kappa^2(1+\nu)]}{48[1-\nu+\kappa^2(1+\nu)]} \tag{E.21}$$

$$\bar{\Omega}_P = -\frac{Q_c (1-\kappa^2)^2}{4 N_C \kappa}. \tag{E.22}$$

The membrane equilibrium is identified by $\delta\Omega_T^* = (\mathrm{d}\Omega_T^*/\mathrm{d}\tilde{w}_0)\delta\tilde{w}_0 = 0$, furnishing

$$\tilde{w}_0 = \left(\frac{-\bar{\Omega}_P}{4\bar{\Omega}_S} \right)^{1/3} = \left(\frac{Q_c}{N_C} \right)^{1/3} K_w. \tag{E.23}$$

Since the test function in Eq. (E.11) is derived based on the axial-force distribution of undeflected circular plates, the variational approximation must approach the exact solution at vanishingly small deflections. For the same reason, this approximation is expected to be less accurate at large deflections.

Appendix F

Geometrical constraints of the elastocapillary model

Here, the derivation of Eqs. (6.4) and (6.5) in chapter 6 is presented. Starting from $H = h - w(R_{00})$ and scaling all lengths with R, we have

$$\Pi = \kappa\Lambda - \tilde{w}(\kappa). \tag{F.1}$$

Substituting $\tilde{w}(\kappa)$ from Eq. (6.1) with $\tilde{w}_0 = (Q_c/N_C)^{1/3}K_w$ furnishes

$$\kappa\Lambda - \Pi - \left(\frac{Q_c}{N_C}\right)^{1/3} K_w(1-\kappa^2) = 0. \tag{F.2}$$

As stated in chapter 6, the in-plane displacement is neglected in the membrane profile for the derivation of the volume constraint. Defining $v_t = v_l + v_a$, we have

$$v_t = \pi R_{00}^2 h + 2\pi \int_{R_{00}}^{R} z r_p \mathrm{d}r_p, \tag{F.3}$$

where $z = H + w$. Substituting w from Eq. (6.1) and scaling all volumes with $4\pi R_{00}^3/3$ furnishes

$$\frac{3}{4}\Lambda + \frac{3(1-\kappa^2)}{4\kappa^3}\Pi + \frac{3\tilde{w}_0(1-\kappa^2)^2}{8\kappa^3} - \hat{v}_a - \hat{v}_l = 0. \tag{F.4}$$

Substituting $\tilde{w}_0 = -(\Pi - \kappa\Lambda)/(1-\kappa^2)$ from Eq. (F.1) in Eq. (F.4) leads to

$$\left[\frac{3}{4} + \frac{3(1-\kappa^2)}{8\kappa^2}\right]\Lambda + \frac{3(1-\kappa^2)}{8\kappa^3}\Pi - \hat{v}_a - \hat{v}_l = 0. \tag{F.5}$$

Appendix G

Branch continuation code

This is a C++ project with the header files `funcs.h`, `class.h`, `solver.h` and source files `class.cpp`, `solver.cpp`. System of equations are declared and defined in `funcs.h`. The core code is written in `solver.h` and `solver.cpp`, and classes are declared and defined in `class.h` and `class.cpp`, respectively.

funcs.h

```
#ifndef FUNCS_H
#define FUNCS_H

#include "solver.h"

//Declarations:
void f_norm(const int &arg_n,const double *arg_y,const double *arg_norm);
double f(const int &arg_i,const double *arg_y);

//Definitions:
inline void f_norm(const int &arg_i,const double *arg_y,double *arg_norm)
{
        int n_inv,jj;
        double t,dt;

        switch(arg_i)
        {
        case 0:
                n_inv=pub_sp.INT_n0+(int) (pub_sp.INT_ni*fabs(arg_y[2]-arg_y[1]));
                dt=(arg_y[2]-arg_y[1])/n_inv;
                arg_norm[0]=0;
                t=arg_y[1];
                if(arg_y[0]<0)
                {
                        for (jj=0;jj<n_inv;jj++)
                        {
                                arg_norm[0]+=(1+arg_y[3]*cos(t))*sqrt(1+pow(arg_y[3],2)+2*arg_y
    [3]*cos(t));
                                t+=dt;
                                arg_norm[0]+=(1+arg_y[3]*cos(t))*sqrt(1+pow(arg_y[3],2)+2*arg_y
    [3]*cos(t));
                        }
                }
                else
                {
                        for (jj=0;jj<n_inv;jj++)
```

```
                        {
                                arg_norm[0]+=(1-(arg_y[3]+2)*cos(t))*sqrt(1+pow((arg_y[3]+2),2)
    -2*(arg_y[3]+2)*cos(t));
                                t+=dt;
                                arg_norm[0]+=(1-(arg_y[3]+2)*cos(t))*sqrt(1+pow((arg_y[3]+2),2)
    -2*(arg_y[3]+2)*cos(t));
                        }
                }
                arg_norm[0]=-arg_norm[0]*(0.75*dt/pow(arg_y[0],3)/2);
        }
}

inline double f(const int &arg_i,const double *arg_y)
{
        double result,sum,dt,t;
        int i,n_inv;

        // Integrator adjusments ------------------
        n_inv=pub_sp.INT_n0+(int) (pub_sp.INT_ni*fabs(arg_y[2]-arg_y[1]));
        if (n_inv>10000)
                n_inv=10000;
        dt=(arg_y[2]-arg_y[1])/n_inv;
        //----------------------------------------
        if (arg_y[0]<0)
        {
                switch (arg_i)
                {
                case 0:
                        result=-(1+arg_y[3]*cos(arg_y[1]))/sin(arg_y[1])/arg_y[3]-tan(arg_y[pub_sp
    .n_eq]);
                        break;
                case 1:
                        result=sqrt(1+pow(arg_y[3],2)+2*arg_y[3]*cos(arg_y[1]))+arg_y[0];
                        break;
                case 2:
                        sum=0;
                        t=arg_y[1];
                        for (i=0;i<n_inv;i++)
                        {
                                sum+=(1+arg_y[3]*cos(t))/sqrt(1+pow(arg_y[3],2)+2*arg_y[3]*cos(t))
    ;
                                t+=dt;
                                sum+=(1+arg_y[3]*cos(t))/sqrt(1+pow(arg_y[3],2)+2*arg_y[3]*cos(t))
    ;
                        }
                        sum=sum*dt/2;
                        result=sum+pub_sp.p[0]*arg_y[0];
                        break;
                case 3:
                        #ifdef _RIGHTCONTACTANGLE
                                result=sin(arg_y[2]);
                        #else
                                result=(1+arg_y[3]*cos(arg_y[2]))/sin(arg_y[2])/arg_y[3]-tan(
    pub_sp.p[1]);
                        #endif
                }
        }
        if (arg_y[0]>0)
        {
                switch (arg_i)
                {
                case 0:
                        result=(1-(arg_y[3]+2)*cos(arg_y[1]))/sin(arg_y[1])/(arg_y[3]+2)+tan(arg_y
    [pub_sp.n_eq]);
                        break;
                case 1:
                        result=sqrt(1+pow(arg_y[3]+2,2)-2*(arg_y[3]+2)*cos(arg_y[1]))-arg_y[0];
                        break;
                case 2:
                        sum=0;
                        t=arg_y[1];
```

```
                        for (i=0;i<n_inv;i++)
                        {
                                sum+=(1-(arg_y[3]+2)*cos(t))/sqrt(1+pow(arg_y[3]+2,2)-2*(arg_y
    [3]+2)*cos(t));
                                t+=dt;
                                sum+=(1-(arg_y[3]+2)*cos(t))/sqrt(1+pow(arg_y[3]+2,2)-2*(arg_y
    [3]+2)*cos(t));
                        }
                        sum=sum*dt/2;
                        result=sum+pub_sp.p[0]*arg_y[0];
                        break;
                case 3:
                        #ifdef _RIGHTCONTACTANGLE
                                result=sin(arg_y[2]);
                        #else
                                result=-(1-(arg_y[3]+2)*cos(arg_y[2]))/sin(arg_y[2])/(arg_y[3]+2)+
    tan(pub_sp.p[1]);
                        #endif
                }
        }

        return result;
}
#endif
```

class.h

```
#ifndef CLASSES_H
#define CLASSES_H

#include "solver.h"

class FixedPoint
{
public:
        //Friendships:
        friend class BranchPoint;

        //Methods:
        FixedPoint();
        FixedPoint(const int &arg_n,const int &arg_nn);
        ~FixedPoint();
        int GetNumberEq() const throw();
        double GetNorm(const int &arg_i) const throw();
        void GetNorm(double *arg_norm) const throw();
        double GetTestFunc() const throw();
        bool IsStable() const throw();
        void GetPoint(double *arg_y) const throw();
        void GetTangent(double *arg_dyds) const throw();
        void GetJacobian(double **arg_fy) const throw();
        void GetParaDiff(double *arg_fl) const throw();
        void GetEigenR(double *arg_er) const throw();
        void GetEigenI(double *arg_ei) const throw();
        void SetSize(const int &arg_n,const int &arg_nn) throw();
        void SetFixedPoint(const double &arg_s,const double *arg_y) throw();
        void CalcDiffx() throw();
        void CalcTangent() throw();
        void CalcEigenvalues() throw(ErrorFP);
        void CalcNorm() throw();
        void CalcTestFunc() throw();
        virtual void GiveDirection(const int arg_dir,FixedPoint *arg_fp) throw();
        void Continuation(const FixedPoint *arg_fp) throw(ErrorFP);

        //Operators:
        FixedPoint& operator = (FixedPoint &arg_fp) throw();
protected:
        int n;
        int k;
        int n_norm;
        double s;
```

```
        double tf;
        bool is_stable;
        bool is_m_alloc;
        double *norm;
        double *y;
        double *dyds;
        double **fy;
        double *fl;
        double *eigen_r;
        double *eigen_i;
private:
        void DeleteMemory() throw();
};
//****************************************************************************************************

class BranchPoint: public FixedPoint
{
public:
        //Friendships:
        friend class FixedPoint;

        //Methods:
        BranchPoint();
        BranchPoint(const int &arg_n,const int &arg_nn);
        ~BranchPoint();
        int GetType() const throw();
        void GetTangent(const int &arg_i,double *arg_dyds) const throw();
        void SetSize(const int &arg_n,const int &arg_nn) throw();
        void SetType(const int &arg_type) throw();
        void SetPrincipalTangent(const double *arg_dy) throw();
        void CalcDiffxx() throw();
        bool CalcBranchPoint() throw();
        void CalcEigenvectors() throw();
        void CalcTangent() throw(ErrorFP);
        void GiveDirection(const int arg_dir,FixedPoint *arg_fp) throw();

        //Operators:
        BranchPoint& operator = (BranchPoint &arg_bp) throw();
        BranchPoint& operator = (FixedPoint &arg_fp) throw();
        bool operator == (FixedPoint &arg_fp) throw();
private:
        int l;
        int type;  // 0:Turning point   1:Transcritical bifurcation    2:Pitchfork bifurcation
     -1:No tangent calculated (negative delta)
        bool is_mm_alloc;
        double *dyds1;
        double *h;
        double *sai;
        double *v;
        double *fll;
        double **fyl;
        double ***fyy;

        void DeleteMemory() throw();
};
#endif
```

class.cpp

```
#include "classes.h"
#include "solver.h"
#include "funcs.h"
#include "ap.cpp"
#include "linalg.cpp"
#include "alglibinternal.cpp"
#include "alglibmisc.cpp"

// Class FixedPoint ************************************************************
FixedPoint::FixedPoint():is_m_alloc(false)
{
```

```
}

FixedPoint::FixedPoint(const int &arg_n,const int &arg_nn):n(arg_n),k(arg_n-1),n_norm(arg_nn),tf
     (0),is_m_alloc(true)
{
        int i;

        norm=new double [n_norm];
        y=new double [n+1];
        dyds=new double [n+1];
        fl=new double [n];
        eigen_r=new double [n];
        eigen_i=new double [n];

        fy=new double *[n];
        for(i=0;i<n;i++)
                *(fy+i)=new double [n];
}

FixedPoint::~FixedPoint()
{
        FixedPoint::DeleteMemory();
}

int FixedPoint::GetNumberEq() const throw()
{
        return n;
}

double FixedPoint::GetNorm(const int &arg_i) const throw()
{
        return norm[arg_i];
}

void FixedPoint::GetNorm(double *arg_norm) const throw()
{
        int i;

        for (i=0;i<n_norm;i++)
                arg_norm[i]=norm[i];
}

double FixedPoint::GetTestFunc() const throw()
{
        return tf;
}

bool FixedPoint::IsStable() const throw()
{
        return is_stable;
}

void FixedPoint::GetPoint(double *arg_y) const throw()
{
        int i;

        for (i=0;i<n+1;i++)
                arg_y[i]=y[i];
}

void FixedPoint::GetTangent(double *arg_dyds) const throw()
{
        int i;

        for (i=0;i<n+1;i++)
                arg_dyds[i]=dyds[i];
}

void FixedPoint::GetJacobian(double **arg_fy) const throw()
{
        int i,j;
```

```
        for (i=0;i<n;i++)
                for (j=0;j<n;j++)
                        arg_fy[i][j]=fy[i][j];
}

void FixedPoint::GetParaDiff(double *arg_fl) const throw()
{
        int i;

        for (i=0;i<n;i++)
                arg_fl[i]=fl[i];
}

void FixedPoint::GetEigenR(double *arg_er) const throw()
{
        int i;

        for (i=0;i<n;i++)
                arg_er[i]=eigen_r[i];
}

void FixedPoint::GetEigenI(double *arg_ei) const throw()
{
        int i;

        for (i=0;i<n;i++)
                arg_ei[i]=eigen_i[i];
}

void FixedPoint::SetSize(const int &arg_n,const int &arg_nn) throw()
{
        int i;

        FixedPoint::DeleteMemory();
        is_m_alloc=true;
        n=arg_n;
        k=n-1;
        n_norm=arg_nn;
        tf=0;
        norm=new double [n_norm];
        y=new double [n+1];
        dyds=new double [n+1];
        fl=new double [n];
        eigen_r=new double [n];
        eigen_i=new double [n];

        fy=new double *[n];
        for(i=0;i<n;i++)
                *(fy+i)=new double [n];
}

void FixedPoint::SetFixedPoint(const double &arg_s,const double *arg_y) throw()
{
        int i;

        s=arg_s;
        for (i=0;i<n+1;i++)
                y[i]=arg_y[i];
}

void FixedPoint::CalcDiffx() throw()
{
        int i,j;
        double *ff,temp,temp1;

        ff=new double [n];
        // fy ------------------------
        for (i=0;i<n;i++)
                ff[i]=f(i,y);
```

```
        for(j=0;j<n;j++)
        {
                temp1=y[j];
                y[j]+=pub_sp.DIFF_eps0;
                for(i=0;i<n;i++)
                {
                        temp=f(i,y);
                        fy[i][j]=(temp-ff[i])/pub_sp.DIFF_eps0;
                }
                y[j]=temp1;
        }
        // fl -------------------------
        temp1=y[n];
        y[n]+=pub_sp.DIFF_eps0;
        for (i=0;i<n;i++)
        {
                temp=f(i,y);
                fl[i]=(temp-ff[i])/pub_sp.DIFF_eps0;
        }
        y[n]=temp1;

        delete [] ff;
}

void FixedPoint::CalcTangent() throw()
{
        int i,j;
        double sum;
        Matrix mA,mB,mX;

        mA.SetSize(n+1,n+1);
        mB.SetSize(n+1,1);

        while (true)
        {
                for (i=0;i<n;i++)
                        for (j=0;j<n;j++)
                                mA(i,j)=fy[i][j];

                for (i=0;i<n;i++)
                        mA(i,n)=fl[i];

                for (j=0;j<n+1;j++)
                {
                        mA(n,j)=0;
                        mB(j,0)=0;
                }
                mA(n,k)=1;
                mB(n,0)=1;
                sum=fabs(mA.Det());

                if(sum>1e-20)
                {
                        mX=!mA*mB;
                        break;
                }
                k=(k+1)%n;
        }

        sum=0;
        for (j=0;j<n+1;j++)
        {
                dyds[j]=mX(j,0);
                sum+=pow(dyds[j],2);
        }
        sum=sqrt(sum);
        for (j=0;j<n+1;j++)
                dyds[j]/=sum;
}

void FixedPoint::CalcEigenvalues() throw(ErrorFP)
```

```
{
        int i,j;
        bool is_converged;
        ErrorFP err_fp;
        real_2d_array a;
        real_1d_array wr;
        real_1d_array wi;
        real_2d_array vl;
        real_2d_array vr;

        a.setlength(n,n);
        for (i=0;i<n;i++)
                for (j=0;j<n;j++)
                        a(i,j)=fy[i][j];

        is_converged=rmatrixevd(a,n,0,wr,wi,vl,vr);
        if(!is_converged)
        {
                err_fp.ErrID=0;
                err_fp.msg="Eigenvalue algorithm has not converged.\n";
                throw err_fp;
                return;
        }

        is_stable=true;
        for (i=0;i<n;i++)
        {
                eigen_r[i]=wr(i);
                eigen_i[i]=wi(i);
                if (eigen_r[i]>=0)
                        is_stable=false;
        }
}

void FixedPoint::CalcNorm() throw()
{
        int i;

        for (i=0;i<n_norm;i++)
                f_norm(i,y,norm);
}

void FixedPoint::CalcTestFunc() throw()
{
        int i;
        double temp,eigen_m;

        eigen_m=fabs(eigen_r[0]);
        for (i=1;i<n;i++)
        {
                temp=fabs(eigen_r[i]);
                if (temp<eigen_m)
                        eigen_m=temp;
        }

        if (eigen_m<pub_sp.CONT_tol_img)
        {
                eigen_m=fabs(eigen_i[0]);
                for (i=1;i<n;i++)
                {
                        temp=fabs(eigen_i[i]);
                        if (temp<eigen_m)
                                eigen_m=temp;
                }
        }
        tf=eigen_m;
}

void FixedPoint::GiveDirection(const int arg_dir,FixedPoint *arg_fp) throw()
{
        /*SetSize has to be called for arg_fp from outside the class to adjust its size.*/
```

```
        int i;

        //predictor
        for (i=0;i<n+1;i++)
                arg_fp->y[i]=y[i]+arg_dir*pub_sp.CONT_sl*dyds[i];
        arg_fp->s=s+arg_dir*pub_sp.CONT_ds;
}

void FixedPoint::Continuation(const FixedPoint *arg_fp) throw(ErrorFP)
{
        int i,j,counter;
        double err,temp;
        Matrix jac,ff,dY;
        ErrorFP err_fp;

        jac.SetSize(n+1,n+1);
        ff.SetSize(n+1,1);

        //corrector
        err=1.;
        counter=0;
        while (err>pub_sp.NR_tol)
        {
                counter+=1;
                FixedPoint::CalcDiffx();
                for (i=0;i<n;i++)
                        ff(i,0)=f(i,y);
                ff(n,0)=0;
                for (i=0;i<n+1;i++)
                        ff(n,0)+=pow(y[i]-arg_fp->y[i],2);
                ff(n,0)-=pow(pub_sp.CONT_ds,2);

                for (i=0;i<n;i++)
                        for (j=0;j<n;j++)
                                jac(i,j)=fy[i][j];
                for (i=0;i<n;i++)
                        jac(i,n)=fl[i];
                for (j=0;j<n+1;j++)
                        jac(n,j)=2*(y[j]-arg_fp->y[j]);

                dY=!jac*ff;
                err=0;
                for (i=0;i<n+1;i++)
                {
                        temp=fabs(dY(i,0));
                        if (err<temp)
                                err=temp;
                        y[i]-=dY(i,0);
                }

                if (counter>pub_sp.NR_iter_max)
                {
                        err_fp.ErrID=1;
                        err_fp.msg="Continuation algorithm has not converged.\n";
                        throw err_fp;
                        break;
                }
        }
}

FixedPoint& FixedPoint::operator = (FixedPoint &arg_fp) throw()
{
        int i,j;
        FixedPoint *fp=dynamic_cast <FixedPoint*>(&arg_fp);
        BranchPoint *bp=dynamic_cast <BranchPoint*>(&arg_fp);

        FixedPoint::SetSize(arg_fp.n,arg_fp.n_norm);
        k=arg_fp.k;
        s=arg_fp.s;
        tf=arg_fp.tf;
        is_stable=arg_fp.is_stable;
```

```
        for (i=0;i<n_norm;i++)
                norm[i]=arg_fp.norm[i];
        for (i=0;i<n;i++)
        {

                y[i]=arg_fp.y[i];
                fl[i]=arg_fp.fl[i];
                eigen_r[i]=arg_fp.eigen_r[i];
                eigen_i[i]=arg_fp.eigen_i[i];
        }
        y[n]=arg_fp.y[n];

        if (bp)
                for (i=0;i<n+1;i++)
                        dyds[i]=bp->dyds1[i];
        else
                for (i=0;i<n+1;i++)
                        dyds[i]=fp->dyds[i];

        for (i=0;i<n;i++)
                for (j=0;j<n;j++)
                        fy[i][j]=arg_fp.fy[i][j];

        return *this;
}

void FixedPoint::DeleteMemory() throw()
{
        int i;

        if(is_m_alloc)
        {
                delete [] norm;
                delete [] y;
                delete [] dyds;
                delete [] fl;
                delete [] eigen_r;
                delete [] eigen_i;

                for(i=n-1;i>=0;i--)
                        delete [] *(fy+i);
                delete [] fy;

                norm=NULL;
                y=NULL;
                dyds=NULL;
                fl=NULL;
                eigen_r=NULL;
                eigen_i=NULL;
                fy=NULL;
        }
}
// Class BranchPoint
     ******************************************************************************
BranchPoint::BranchPoint():is_mm_alloc(false)
{
}

BranchPoint::BranchPoint(const int &arg_n,const int &arg_nn):l(n-1),is_mm_alloc(true),FixedPoint(
     arg_n,arg_nn)
{
        int i,j;

        dyds1=new double [n];
        h=new double [n];
        sai=new double [n];
        v=new double [n];
        fll=new double [n];

        fyl=new double *[n];
        for(i=0;i<n;i++)
```

```
                *(fyl+i)=new double [n];

        fyy=new double **[n];
        for(i=0;i<n;i++)
        {
                *(fyy+i)=new double *[n];
                for(j=0;j<n;j++)
                        *(*(fyy+i)+j)=new double [n];
        }
}

BranchPoint::~BranchPoint()
{
        BranchPoint::DeleteMemory();
}

int BranchPoint::GetType() const throw()
{
        return type;
}

void BranchPoint::GetTangent(const int &arg_i,double *arg_dyds) const throw()
{
        int i;

        if (arg_i==1)
        {
                for (i=0;i<n+1;i++)
                        arg_dyds[i]=dyds[i];
        }
        if (arg_i==2)
        {
                for (i=0;i<n+1;i++)
                        arg_dyds[i]=dyds1[i];
        }
}

void BranchPoint::SetSize(const int &arg_n,const int &arg_nn) throw()
{
        int i,j;

        BranchPoint::DeleteMemory();
        FixedPoint::SetSize(arg_n,arg_nn);
        is_mm_alloc=true;
        l=arg_n-1;
        dyds1=new double [n+1];
        h=new double [n];
        sai=new double [n];
        v=new double [n];
        fll=new double [n];

        fyl=new double *[n];
        for(i=0;i<n;i++)
                *(fyl+i)=new double [n];

        fyy=new double **[n];
        for(i=0;i<n;i++)
        {
                *(fyy+i)=new double *[n];
                for(j=0;j<n;j++)
                        *(*(fyy+i)+j)=new double [n];
        }
}

void BranchPoint::SetType(const int &arg_type) throw()
{
        type=arg_type;
}

void BranchPoint::SetPrincipalTangent(const double *arg_dy) throw()
{
```

```
        int i;

        for (i=0;i<n+1;i++)
                dyds[i]=arg_dy[i];
}

void BranchPoint::CalcDiffxx() throw()
{
        int i,j,k;
        double *ff,temp1,temp2;
        double ff_j,ff_k,ff_jk;

        ff=new double [n];
        // fyy ------------------------
        for (i=0;i<n;i++)
                ff[i]=f(i,y);

        for(i=0;i<n;i++)
        {
                for(j=0;j<n;j++)
                {
                        for(k=0;k<n;k++)
                        {
                                temp1=y[j];
                                temp2=y[k];
                                y[j]+=pub_sp.DIFF_eps0;
                                y[k]+=pub_sp.DIFF_eps0;
                                ff_jk=f(i,y);
                                y[k]-=pub_sp.DIFF_eps0;
                                ff_j=f(i,y);
                                y[j]=temp1;
                                y[k]=temp2+pub_sp.DIFF_eps0;
                                ff_k=f(i,y);
                                fyy[i][j][k]=(ff_jk-ff_j-ff_k+ff[i])/pub_sp.DIFF_eps0/pub_sp.
    DIFF_eps0;
                                y[k]=temp2;
                        }
                }
        }
        // fyl -------------------------
        for (i=0;i<n;i++)
        {
                for (j=0;j<n;j++)
                {
                        temp1=y[j];
                        temp2=y[n];
                        y[j]+=pub_sp.DIFF_eps0;
                        y[n]+=pub_sp.DIFF_eps0;
                        ff_jk=f(i,y);
                        y[n]-=pub_sp.DIFF_eps0;
                        ff_j=f(i,y);
                        y[j]=temp1;
                        y[n]=temp2+pub_sp.DIFF_eps0;
                        ff_k=f(i,y);
                        fyl[i][j]=(ff_jk-ff_j-ff_k+ff[i])/pub_sp.DIFF_eps0/pub_sp.DIFF_eps0;
                        y[n]=temp2;
                }
        }
        // fll --------------------------
        for (i=0;i<n;i++)
        {
                temp1=y[n];
                y[n]+=pub_sp.DIFF_eps0;
                ff_k=f(i,y);
                y[n]=temp1-pub_sp.DIFF_eps0;
                ff_j=f(i,y);
                fll[i]=(ff_j+ff_k-2*ff[i])/pow(pub_sp.DIFF_eps0,2);
                y[n]=temp1;
        }

        delete [] ff;
```

```
}

bool BranchPoint::CalcBranchPoint() throw()
{
        bool result;
        int i,j,ii,km,lm,counter;
        double err,temp,det_max,*y0,*h0,**fyy_h,*fyl_h,*fy_h;
        Matrix jac,ff,dY,mA,mB,mX;

        jac.SetSize(2*n+1,2*n+1);
        ff.SetSize(2*n+1,1);
        mA.SetSize(n,n);
        mB.SetSize(n,1);
        y0=new double [n+1];
        fyl_h=new double [n];
        fy_h=new double [n];
        fyy_h=new double *[n];
        for (i=0;i<n;i++)
                *(fyy_h+i)=new double [n];

        for (i=0;i<n+1;i++)
                y0[i]=y[i];
        BranchPoint::CalcDiffx();

        //Approximate h
        det_max=0;
        for(k=0;k<n;k++)
        {
                for(l=0;l<n;l++)
                {
                        for (i=0;i<n;i++)
                                for (j=0;j<n;j++)
                                        mA(i,j)=fy[i][j];
                        for (i=0;i<n;i++)
                        {
                                mA(l,i)=0;
                                mB(i,0)=0;
                        }
                        mA(l,k)=1;
                        mB(l,0)=1;
                        temp=fabs(mA.Det());
                        if (temp>det_max)
                        {
                                lm=l;
                                km=k;
                                det_max=temp;
                        }
                }
        }
        k=km;
        l=lm;
        for (i=0;i<n;i++)
                for (j=0;j<n;j++)
                        mA(i,j)=fy[i][j];
        for (i=0;i<n;i++)
        {
                mA(l,i)=0;
                mB(i,0)=0;
        }
        mA(l,k)=1;
        mB(l,0)=1;
        mX=!mA*mB;

        for (i=0;i<n;i++)
                h[i]=mX(i,0);

        if (det_max<pub_sp.BRAN_tol_det)
        {
                h0=new double [n];
                for (i=0;i<n;i++)
                        h0[i]=h[i];
```

```
}
//Branch point calc main loop
for (i=0;i<2*n+1;i++)
        for (j=0;j<2*n+1;j++)
                jac(i,j)=0;
jac(2*n,n+1+k)=1;

err=1.;
counter=0;
while (true)
{
        counter+=1;
        BranchPoint::CalcDiffxx();
        for (i=0;i<n;i++)
        {
                fyl_h[i]=0;
                fy_h[i]=0;
                for (j=0;j<n;j++)
                {
                        fyl_h[i]+=fyl[i][j]*h[j];
                        fy_h[i]+=fy[i][j]*h[j];
                }
        }
        for (i=0;i<n;i++)
        {
                for (j=0;j<n;j++)
                {
                        fyy_h[i][j]=0;
                        for (ii=0;ii<n;ii++)
                                fyy_h[i][j]+=fyy[i][j][ii]*h[ii];
                }
        }

        for (i=0;i<n;i++)
        {
                for (j=0;j<n;j++)
                {
                        jac(i,j)=fy[i][j];
                        jac(i+n,j)=fyy_h[i][j];
                        jac(i+n,j+n+1)=fy[i][j];
                }
        }
        for (i=0;i<n;i++)
        {
                jac(i,n)=fl[i];
                jac(i+n,n)=fyl_h[i];

                ff(i,0)=f(i,y);
                ff(i+n,0)=fy_h[i];
        }
        ff(2*n,0)=h[k]-1;
        dY=!jac*ff;

        for (i=0;i<n;i++)
        {
                y[i]-=dY(i,0);
                h[i]-=dY(i+n+1,0);
        }
        y[n]-=dY(n,0);

        err=0;
        for (i=0;i<2*n+1;i++)
        {
                temp=fabs(dY(i,0));
                if (err<temp)
                        err=temp;
        }

        if (err<pub_sp.NR_tol)
        {
                temp=0;
```

```
                        for (i=0;i<n+1;i++)
                                temp+=pow(y0[i]-y[i],2);
                        temp=sqrt(temp);
                        if (temp<=pub_sp.CONT_ds)
                                result=true;
                        else
                                result=false;
                        break;
                }
                if (counter>pub_sp.NR_iter_max)
                {
                        result=false;
                        break;
                }
                BranchPoint::CalcDiffx();
        }

        if (det_max<pub_sp.BRAN_tol_det)
        {
                for (i=0;i<n;i++)
                        h[i]=h0[i];
                delete [] h0;
        }

        delete [] y0;
        delete [] fyl_h;
        delete [] fy_h;
        for (i=n-1;i>=0;i--)
                delete [] *(fyy_h+i);
        delete [] fyy_h;

        return result;
}

void BranchPoint::CalcEigenvectors() throw()
{
        int i,j;
        Matrix mA,mB,mX;

        mA.SetSize(n,n);
        mB.SetSize(n,1);

        for (i=0;i<n;i++)
                for (j=0;j<n;j++)
                        mA(i,j)=fy[j][i]; //Jacobian transpose

        for (i=0;i<n;i++)
        {
                mA(l,i)=h[i];
                mB(i,0)=0;
        }
        mB(l,0)=1;
        mX=!mA*mB;

        for (i=0;i<n;i++)
                sai[i]=mX(i,0);

        //Vector v
        for (i=0;i<n;i++)
                for (j=0;j<n;j++)
                        mA(i,j)=fy[i][j];

        for (i=0;i<n;i++)
        {
                mA(l,i)=sai[i];
                mB(i,0)=-fl[i];
        }
        mB(l,0)=0;

        mX=!mA*mB;
```

```
        for (i=0;i<n;i++)
                v[i]=mX(i,0);
}

void BranchPoint::CalcTangent() throw(ErrorFP)
{
        int i,j,kk;
        double a,b,c,beta[2],delta,**dy;
        double term1,term2,term3;
        ErrorFP err_fp;

        dy=new double *[2];
        for (i=0;i<2;i++)
                *(dy+i)=new double [n+1];

        a=0;
        for (i=0;i<n;i++)
                for (j=0;j<n;j++)
                        for (kk=0;kk<n;kk++)
                                a+=sai[i]*fyy[i][j][kk]*h[j]*h[kk];

        term1=0;
        for (i=0;i<n;i++)
                for (j=0;j<n;j++)
                        for (kk=0;kk<n;kk++)
                                term1+=sai[i]*fyy[i][j][kk]*v[kk]*h[j];
        term2=0;
        for (j=0;j<n;j++)
                for (kk=0;kk<n;kk++)
                        term2+=sai[j]*fyl[j][kk]*h[kk];
        b=term1+term2;

        term1=0;
        for (i=0;i<n;i++)
                for (j=0;j<n;j++)
                        for (kk=0;kk<n;kk++)
                                term1+=sai[i]*fyy[i][j][kk]*v[j]*v[kk];
        term2=0;
        for (j=0;j<n;j++)
                for (kk=0;kk<n;kk++)
                        term2+=sai[j]*fyl[j][kk]*v[kk];
        term2*=2;
        term3=0;
        for (i=0;i<n;i++)
                term3+=sai[i]*fll[i];
        c=term1+term2+term3;

        term1=0;
        for (i=0;i<n;i++)
                term1+=sai[i]*fl[i];
        if (fabs(term1)>pub_sp.BRAN_tol_tp)
        {
                type=0;
                return;
        }

        delta=pow(b,2)-a*c;
        if (delta<=0)
        {
                type=-1;
                err_fp.ErrID=0;
                err_fp.msg="Non-positive delta encountered when calculating tangents for
    bifurcation point.\n";
                throw err_fp;
                return;
        }

        if (a<pub_sp.BRAN_tol_a)
        {
                type=2;
                beta[0]=-c/b/2;
```

```
        for (j=0;j<n;j++)
        {
            dy[0][j]=h[j];
            dy[1][j]=v[j]+beta[0]*h[j];
        }
        dy[0][n]=0;
        dy[1][n]=1;

        term1=0;
        term2=0;
        for (j=0;j<n;j++)
        {
            term1+=pow(dy[1][j],2);
            term2+=pow(dy[0][j],2);
        }
        term1+=1;
        term1=sqrt(term1);
        term2=sqrt(term2);
        for (j=0;j<n+1;j++)
        {
            dy[1][j]/=term1;
            dy[0][j]/=term2;
        }
    }
    else
    {
        type=1;
        beta[0]=(-b+sqrt(delta))/a;
        beta[1]=(-b-sqrt(delta))/a;
        for (i=0;i<2;i++)
        {
            for (j=0;j<n;j++)
                dy[i][j]=v[j]+beta[i]*h[j];
            dy[i][n]=1;
            term3=0;
            for (j=0;j<n+1;j++)
                term3+=pow(dy[i][j],2);
            term3=sqrt(term3);
            for (j=0;j<n+1;j++)
                dy[i][j]/=term3;
        }
    }

    for (i=0;i<2;i++)
    {
        beta[i]=0;
        for (j=0;j<n+1;j++)
            beta[i]+=dyds[j]*dy[i][j];
        beta[i]=fabs(beta[i]);
    }
    if (beta[0]<beta[1])
    {
        for (j=0;j<n+1;j++)
        {
            dyds1[j]=dy[0][j];
            dyds[j]=dy[1][j];
        }
    }
    else
    {
        for (j=0;j<n+1;j++)
        {
            dyds1[j]=dy[1][j];
            dyds[j]=dy[0][j];
        }
    }

    for (i=1;i>=0;i--)
        delete [] *(dy+i);
    delete [] dy;
}
```

```
void BranchPoint::GiveDirection(const int arg_dir,FixedPoint *arg_fp) throw()
{
        /*SetSize has to be called for arg_fp from outside the class to adjust its size.*/
        int i;

        //predictor
        for (i=0;i<n+1;i++)
                arg_fp->y[i]=y[i]+arg_dir*pub_sp.CONT_sl*dyds1[i];
        arg_fp->s=s+arg_dir*pub_sp.CONT_ds;
}

BranchPoint& BranchPoint::operator = (BranchPoint &arg_bp) throw()
{
        int i,j,kk;

        BranchPoint::SetSize(arg_bp.n,arg_bp.n_norm);
        k=arg_bp.k;
        l=arg_bp.l;
        s=arg_bp.s;
        tf=arg_bp.tf;
        is_stable=arg_bp.is_stable;
        for (i=0;i<n_norm;i++)
                norm[i]=arg_bp.norm[i];
        for (i=0;i<n;i++)
        {
                y[i]=arg_bp.y[i];
                dyds[i]=arg_bp.dyds[i];
                fl[i]=arg_bp.fl[i];
                eigen_r[i]=arg_bp.eigen_r[i];
                eigen_i[i]=arg_bp.eigen_i[i];
                dyds1[i]=arg_bp.dyds1[i];
                h[i]=arg_bp.h[i];
                sai[i]=arg_bp.sai[i];
                v[i]=arg_bp.v[i];
                fll[i]=arg_bp.fll[i];
        }
        y[n]=arg_bp.y[n];
        dyds[n]=arg_bp.dyds[n];
        dyds1[n]=arg_bp.dyds1[n];

        for (i=0;i<n;i++)
        {
                for (j=0;j<n;j++)
                {
                        fy[i][j]=arg_bp.fy[i][j];
                        fyl[i][j]=arg_bp.fyl[i][j];
                }
        }

        for (i=0;i<n;i++)
                for (j=0;j<n;j++)
                        for (kk=0;kk<n;kk++)
                                fyy[i][j][kk]=arg_bp.fyy[i][j][kk];

        return *this;
}

BranchPoint& BranchPoint::operator = (FixedPoint &arg_fp) throw()
{
        int i,j;

        BranchPoint::SetSize(arg_fp.n,arg_fp.n_norm);
        k=arg_fp.k;
        s=arg_fp.s;
        tf=arg_fp.tf;
        is_stable=arg_fp.is_stable;
        for (i=0;i<n_norm;i++)
                norm[i]=arg_fp.norm[i];
        for (i=0;i<n;i++)
        {
```

```
                y[i]=arg_fp.y[i];
                dyds[i]=arg_fp.dyds[i];
                fl[i]=arg_fp.fl[i];
                eigen_r[i]=arg_fp.eigen_r[i];
                eigen_i[i]=arg_fp.eigen_i[i];
        }
        y[n]=arg_fp.y[n];
        dyds[n]=arg_fp.dyds[n];

        for (i=0;i<n;i++)
                for (j=0;j<n;j++)
                        fy[i][j]=arg_fp.fy[i][j];

        return *this;
}

bool BranchPoint::operator == (FixedPoint &arg_fp) throw()
{
        int i;
        double dist;

        dist=0;
        for(i=0;i<n+1;i++)
                dist+=pow(y[i]-arg_fp.y[i],2);
        dist=sqrt(dist);
        if (dist>=pub_sp.BRAN_tol_bp)
                return false;
        else
                return true;
}

void BranchPoint::DeleteMemory() throw()
{
        int i,j;

        if (is_mm_alloc)
        {
                delete [] dyds1;
                delete [] h;
                delete [] sai;
                delete [] v;
                delete [] fll;

                for(i=n-1;i>=0;i--)
                        delete [] *(fyl+i);
                delete [] fyl;

                for(i=n-1;i>=0;i--)
                {
                        for(j=n-1;j>=0;j--)
                                delete [] *(*(fyy+i)+j);
                        delete [] *(fyy+i);
                }
                delete [] fyy;

                dyds1=NULL;
                h=NULL;
                sai=NULL;
                v=NULL;
                fll=NULL;
                fyl=NULL;
                fyy=NULL;
        }
}
```

solver.h

```
#ifndef SOLVER_H
#define SOLVER_H
```

```
#include <sstream>
#include <fstream>
#include <string>
#include <iomanip>
#include <direct.h>
#include <matrix_tlc.h>
#include <ap.h>
#include <linalg.h>
#include <alglibinternal.h>
#include <alglibmisc.h>
#include <engine.h>
#include <Windows.h>

#define _STANDALONE
//#define _EXTERNALPROCESS
//#define _RIGHTCONTACTANGLE
using namespace std;
using namespace math;
using namespace alglib;
typedef matrix<double> Matrix;


struct SolutionP
{
        int n_eq;        //Number of equations
        int n_norm;  //Number of norms
        int n_fp;        //Maximum number of fixed points per half-branch
        int *n_fpa;      //Actual number of fixed points of a half-branch
        int n_bp;        //Maximum number of branch points
        int n_br;        //Maximum number of half-branches
        int counter_br; //Branch counter
        int counter_bp; //Branch point counter
        int INT_n0;
        int INT_ni;
        int branch_index;
        double DIFF_eps0;
        double CONT_sl;
        double CONT_ds;
        double CONT_tol_img;   //Criterion on test function to check if eigenvalues are purely
     imaginary
        double BRAN_tol_det;   //Criterion on the required det of Jacobian in calculating branch
     points
        double BRAN_tol_bp;    //Criterion on the minimum distance between computed branch points
     to avoid continuation overlapping
        double BRAN_tol_a;     //Criterion on 'a' for pitchfork bifurcation detection
        double BRAN_tol_tp;        //Criterion to distinguish turning point from bifurcation point
        double BRAN_tol_tf;    //Criterion on test function for branch point detection
        double NR_tol;
        int NR_iter_max;
        double y0[5];
        double p[2];
        string strDataPath;
        string strSharedDataPath;
};

struct ErrorFP
{
        int ErrID;
        char *msg;
};

//Global variable --------------------------------
extern FILE *fp;
extern SolutionP pub_sp;
//------------------------------------------------
#endif
```

solver.cpp

```
#include "solver.h"
```

```
#include "classes.h"
#include "funcs.h"

FILE *fp;
SolutionP pub_sp;

//************************************************************************
void ErrorHandler(ErrorFP arg_err)
{
        cout << arg_err.msg;
        system ("PAUSE");
        exit(1);
}

void MAllocation(FixedPoint ***arg_br,BranchPoint **arg_bp)
{
        int i;

        pub_sp.n_fpa=new int [pub_sp.n_br];
        *arg_bp=new BranchPoint [pub_sp.n_bp];
        *arg_br=new FixedPoint *[pub_sp.n_br];
        for (i=0;i<pub_sp.n_br;i++)
                *(*arg_br+i)=new FixedPoint [pub_sp.n_fp];
}

void MDelete(FixedPoint ***arg_br,BranchPoint **arg_bp)
{
        int i;

        delete [] pub_sp.n_fpa;
        delete [] *arg_bp;
        for (i=pub_sp.n_br-1;i>=0;i--)
                delete [] *(*arg_br+i);
        delete *arg_br;
}

void GetParameters(char **argv)
{
        char *chrAppPath=NULL;
        string strAppPath;

        pub_sp.n_eq=4;
        pub_sp.n_norm=1;
        pub_sp.n_fp=20000;
        pub_sp.n_bp=10;
        pub_sp.n_br=30;
        pub_sp.counter_br=0;
        pub_sp.counter_bp=0;
        pub_sp.INT_n0=500;
        pub_sp.INT_ni=500;
        pub_sp.NR_tol=1.e-8;
        pub_sp.NR_iter_max=15;
        pub_sp.DIFF_eps0=1e-7;
        pub_sp.CONT_tol_img=1e-6;
        pub_sp.BRAN_tol_det=0.001;
        pub_sp.BRAN_tol_bp=0.001;
        pub_sp.BRAN_tol_tf=0.1;
        pub_sp.BRAN_tol_tp=1e-5;
        pub_sp.BRAN_tol_a=1e-6;

    /*****************************************************************************************************

        * strAppPath is required when the application is run in standalone mode to store results
    in ASCII format.
        * strDataPath reletive to strAppPath is clear. strAppPath is not required when the
    application is called from
        * an external program as a process. strDataPath reletive to strAppPath is unclear since
    the process could be
        * placed in any arbitrary folder. The following assumes it is in the project working
    directory.
```

```
****************************************************************************************************/

    #ifdef _STANDALONE
            chrAppPath=_getcwd(chrAppPath,100);
        //standalone
            strAppPath=chrAppPath;
                //standalone
            pub_sp.strDataPath=strAppPath+"\\data";                                          //
standalone
            pub_sp.strSharedDataPath=pub_sp.strDataPath;
            pub_sp.p[0]=0.1;
                        //standalone
            pub_sp.p[1]=30*pi()/180;
                //standalone
            pub_sp.CONT_sl=0.005;
                //standalone
            pub_sp.CONT_ds=0.005;
                //standalone
            pub_sp.branch_index=1;
                //standalone
    #endif
    #ifdef _EXTERNALPROCESS
            strAppPath=argv[0];
                        //external process
            strAppPath=strAppPath.substr(0,strAppPath.length()-16);        //external process
            pub_sp.strDataPath=strAppPath+"\\data";                                          //
external process
            pub_sp.strSharedDataPath="C:\\Amir\\PhD - McGill\\PhD project\\Axi-meniscus\\
shared data\\branch data";
            pub_sp.y0[0]=atof(argv[1]);
                //external process
            pub_sp.y0[1]=atof(argv[2]);
                //external process
            pub_sp.y0[2]=atof(argv[3]);
                //external process
            pub_sp.y0[3]=atof(argv[4]);
                //external process
            pub_sp.y0[4]=atof(argv[5]);
                //external process

            pub_sp.p[0]=atof(argv[6]);
                //external process
            pub_sp.p[1]=atof(argv[7]);
                //external process
            pub_sp.CONT_sl=atof(argv[8]);
        //external process
            pub_sp.CONT_ds=atof(argv[8]);
        //external process
            pub_sp.branch_index=atoi(argv[9]);
        //external process
    #endif
}

void InitialPoint(FixedPoint *arg_fp)
{
    double *y0;

    y0=new double [pub_sp.n_eq+1];

    #ifdef _STANDALONE
            y0[0]=-19.304032371117295;                                 //standalone
            y0[1]=-1.4476967231016771;                                 //standalone
            y0[2]=1.0924227310774302;                                  //standalone
            y0[3]=19.155715744900000;                                  //standalone
            y0[pub_sp.n_eq]=0.17453292519940003;                   //standalone
    #endif
    #ifdef _EXTERNALPROCESS
            y0[0]=pub_sp.y0[0];                                                   //external
 process
```

```
                y0[1]=pub_sp.y0[1];                                              //external
      process
                y0[2]=pub_sp.y0[2];                                              //external
      process
                y0[3]=pub_sp.y0[3];                                              //external
      process
                y0[pub_sp.n_eq]=pub_sp.y0[4];                            //external process
        #endif
        arg_fp->SetSize(pub_sp.n_eq,pub_sp.n_norm);
        arg_fp->SetFixedPoint(0,y0);
        arg_fp->CalcDiffx();
        arg_fp->CalcTangent();
        arg_fp->CalcEigenvalues();
        arg_fp->CalcTestFunc();
        arg_fp->CalcNorm();

        delete [] y0;
}

inline void ConstraintCheck(const double *arg_y0,const double *arg_y1) throw(ErrorFP)
{
        ErrorFP errfp;
        double taw_z,taw0,taw1;

        if (arg_y1[0]<0 && arg_y1[pub_sp.n_eq]>0 && arg_y1[pub_sp.n_eq]<pi())
        {
                taw_z=acos(-1/arg_y1[3]);
                taw0=arg_y1[1];
                taw1=arg_y1[2];
                if (arg_y1[3]<-1 && taw0>0)
                {
                        taw0-=2*pi();
                        taw1-=2*pi();
                }
                if (arg_y1[3]>1 && (taw1>taw_z||taw0<-taw_z))
                {
                        errfp.msg="Self-intersecting profile encountered (positive 'a').\n";
                        errfp.ErrID=1;
                        throw errfp;
                }
                if (arg_y1[3]<-1 && (taw1>-taw_z||taw0<taw_z-2*pi()))
                {
                        errfp.msg="Self-intersecting profile encountered (negative 'a').\n";
                        errfp.ErrID=1;
                        throw errfp;
                }
        }
        if (arg_y1[0]*arg_y0[0]<0)
        {
                errfp.msg="Wrong sign for 'q' encountered.\n";
                errfp.ErrID=1;
                throw errfp;
        }
        if (arg_y1[pub_sp.n_eq]>2*pi()||arg_y1[pub_sp.n_eq]<-pi())
        {
                errfp.msg="Non-physical branching parameter.\n";
                errfp.ErrID=1;
                throw errfp;
        }
        if ((arg_y1[0]<0 && 1+arg_y1[3]*cos(arg_y1[2])<0)||(arg_y1[0]>0 && 1-(arg_y1[3]+2)*cos(
     arg_y1[2])>0))
        {
                errfp.msg="Complement contact angle encountered.\n";
                errfp.ErrID=1;
                throw errfp;
        }
}

void BranchContinuation(FixedPoint **arg_br,BranchPoint *arg_bp,FixedPoint *arg_fp0) throw(ErrorFP
     )
{
```

```
    int i,j,ib,dir,dir_corr,bp_type;
    double tf1,tf2,tf3,d_tf1,d_tf2,tf_min,s_min,m1,m2,m3,dir_chk,*y0,*y1,*y2,*y3;
    bool chk_bp,is_bp;

    y0=new double [pub_sp.n_eq+1];
    y1=new double [pub_sp.n_eq+1];
    y2=new double [pub_sp.n_eq+1];
    y3=new double [pub_sp.n_eq+1];

    for (dir=-1;dir<2;dir+=2)
    {
            dir_corr=1;
            pub_sp.counter_br+=1;
            ib=pub_sp.counter_br-1;
            pub_sp.n_fpa[ib]=pub_sp.n_fp;
            arg_br[ib][0]=*arg_fp0;
            arg_br[ib][1].SetSize(pub_sp.n_eq,pub_sp.n_norm);
            arg_fp0->GiveDirection(dir,*(arg_br+ib)+1);
            try
            {
                    arg_br[ib][1].Continuation(*(arg_br+ib));
                    arg_fp0->GetPoint(y0);
                    arg_br[ib][1].GetPoint(y1);
                    ConstraintCheck(y0,y1);
            }
            catch (ErrorFP err_fp)
            {
                    if (err_fp.ErrID==1)
                    {
                            pub_sp.n_fpa[ib]=1;
                            cout << err_fp.msg << "On branch="<< ib+1 <<", at point=1\n";
                            continue;
                    }
                    else
                    {
                            throw err_fp;
                    }
            }
            arg_br[ib][1].CalcTangent();
            arg_br[ib][1].CalcEigenvalues();
            arg_br[ib][1].CalcTestFunc();
            arg_br[ib][1].CalcNorm();
            //Correct continuation direction
            arg_br[ib][0].GetTangent(y1);
            arg_br[ib][1].GetTangent(y2);
            dir_chk=0;
            for (j=0;j<pub_sp.n_eq+1;j++)
                    dir_chk+=y1[j]*y2[j];
            if (dir_chk<0)
                    dir_corr*=-1;
            //------------------------------
            for (i=2;i<pub_sp.n_fp;i++)
            {
                    arg_br[ib][i].SetSize(pub_sp.n_eq,pub_sp.n_norm);
                    arg_br[ib][i-1].GiveDirection(dir_corr*dir,*(arg_br+ib)+i);
                    //Filter out invalid parts of the branch
-------------------------------------
                    try
                    {
                            arg_br[ib][i].Continuation(*(arg_br+ib)+i-1);
                            arg_br[ib][i-1].GetPoint(y0);
                            arg_br[ib][i].GetPoint(y1);
                            ConstraintCheck(y0,y1);
                    }
                    catch (ErrorFP err_fp)
                    {
                            if (err_fp.ErrID==1)
                            {
                                    pub_sp.n_fpa[ib]=i;
                                    cout << err_fp.msg << "On branch="<< ib+1 <<", at point="
<<i<<"\n";
```

```
                                break;
                        }
                        else
                        {
                                throw err_fp;
                        }
                }

//-------------------------------------------------------------------------------------------------

                arg_br[ib][i].CalcTangent();
                arg_br[ib][i].CalcEigenvalues();
                arg_br[ib][i].CalcTestFunc();
                arg_br[ib][i].CalcNorm();
                tf1=arg_br[ib][i-2].GetTestFunc();
                tf2=arg_br[ib][i-1].GetTestFunc();
                tf3=arg_br[ib][i].GetTestFunc();
                //Correct continuation direction
                arg_br[ib][i-1].GetTangent(y1);
                arg_br[ib][i].GetTangent(y2);
                dir_chk=0;
                for (j=0;j<pub_sp.n_eq+1;j++)
                        dir_chk+=y1[j]*y2[j];
                if (dir_chk<0)
                        dir_corr*=-1;
                //------------------------------
                chk_bp=false;
                d_tf1=tf2-tf1;
                d_tf2=tf3-tf2;
                tf_min=tf2-pow(tf3-tf1,2)/(tf1-2*tf2+tf3)/8;
                if (d_tf1*d_tf2<0 && tf_min<pub_sp.BRAN_tol_tf)
                {
                        chk_bp=true;
                        s_min=(tf1-tf3)/(tf1-2*tf2+tf3)/2;
                        m2=1-fabs(s_min);
                        if (s_min<0)
                        {
                                m1=1-m2;
                                m3=0;
                        }
                        else
                        {
                                m3=1-m2;
                                m1=0;
                        }
                }

                if (chk_bp)
                {
                        arg_br[ib][i-2].GetPoint(y1);
                        arg_br[ib][i-1].GetPoint(y2);
                        arg_br[ib][i].GetPoint(y3);
                        for (j=0;j<pub_sp.n_eq+1;j++)
                                y0[j]=(m1*y1[j]+m2*y2[j]+m3*y3[j]);
                        arg_bp[pub_sp.counter_bp].SetSize(pub_sp.n_eq,pub_sp.n_norm);
                        arg_bp[pub_sp.counter_bp].SetFixedPoint(0,y0);
                        is_bp=arg_bp[pub_sp.counter_bp].CalcBranchPoint();
                        if(is_bp)
                        {
                                for (j=0;j<pub_sp.counter_bp;j++)
                                        if (arg_bp[j]==arg_bp[pub_sp.counter_bp])
                                                is_bp=false;
                        }
                        if(is_bp)
                        {
                                arg_br[ib][i-2].GetTangent(y1);
                                arg_br[ib][i-1].GetTangent(y2);
                                arg_br[ib][i].GetTangent(y3);
                                for (j=0;j<pub_sp.n_eq+1;j++)
                                        y0[j]=(m1*y1[j]+m2*y2[j]+m3*y3[j]);
                                arg_bp[pub_sp.counter_bp].SetPrincipalTangent(y0);
```

```
                                        arg_bp[pub_sp.counter_bp].CalcEigenvalues();
                                        arg_bp[pub_sp.counter_bp].CalcNorm();
                                        arg_bp[pub_sp.counter_bp].CalcEigenvectors();
                                        arg_bp[pub_sp.counter_bp].CalcTangent();
                                        pub_sp.counter_bp+=1;
                                        bp_type=arg_bp[pub_sp.counter_bp-1].GetType();
                                        if (bp_type)
                                                BranchContinuation(arg_br,arg_bp,arg_bp+pub_sp.
    counter_bp-1);
                                }
                        }
                }
        }

        delete [] y0;
        delete [] y1;
        delete [] y2;
        delete [] y3;
}

void PrintResult(FixedPoint **arg_br,BranchPoint *arg_bp)
{
        int i,j,jj,k,type;
        double *y0,vol;
        bool is_stable,is_stable1;
        ostringstream ostrIndex,ostrP0,ostrP1;
        string strIndex,strP0,strP1,strFileName,strA,strCommand,strSpec;
        Engine *eng;
        mxArray **mxA=NULL;
        bool *is_stable_curve;
        int n_curve,*l_curve,i_curve,i_curve_t;
        double *dblA;

        y0=new double [pub_sp.n_eq+1];
        ostrP0.str("");
        ostrP1.str("");
        ostrP0<< setiosflags(ios::fixed) << setprecision(2) << pub_sp.p[0];
        ostrP1<< setiosflags(ios::fixed) << setprecision(0) << pub_sp.p[1]*180/pi();
        strP0=ostrP0.str();
        strP1=ostrP1.str();
        for (i=0;i<pub_sp.counter_br;i++)
        {
                ostrIndex.str("");
                ostrIndex<<(i+1);
                strIndex=ostrIndex.str();
                strFileName=pub_sp.strDataPath+"\\branch"+strIndex+"-SN"+strP0+"-CA"+strP1+".txt";
                fp=fopen(strFileName.c_str(),"w");
                        for (j=0;j<pub_sp.n_fpa[i];j++)
                        {
                                arg_br[i][j].GetPoint(y0);
                                vol=arg_br[i][j].GetNorm(0);
                                fprintf(fp,"%d\t%f\t%f\t%f\t%f\t%f\t%f\n",j,y0[pub_sp.n_eq],y0[0],
    y0[1],y0[2],y0[3],vol);
                        }
                fclose(fp);

                /*strFileName=pub_sp.strDataPath+"\\eigen"+strIndex+"-SN"+strP0+"-CA"+strP1+".txt
    ";
                fp=fopen(strFileName.c_str(),"w");
                        for (j=0;j<pub_sp.n_fpa[i];j++)
                        {
                                arg_br[i][j].GetEigenR(y0);
                                vol=arg_br[i][j].GetNorm(0);
                                fprintf(fp,"%d\t%f\t%f\t%f\t%f\t%f\n",j,vol,y0[0],y0[1],y0[2],y0
    [3]);
                        }
                fclose(fp);*/
        }
        strFileName=pub_sp.strDataPath+"\\branchpoint"+"-SN"+strP0+"-CA"+strP1+".txt";
        fp=fopen(strFileName.c_str(),"w");
                for (i=0;i<pub_sp.counter_bp;i++)
```

```
        {
                type=arg_bp[i].GetType();
                arg_bp[i].GetPoint(y0);
                fprintf(fp,"type=\t%d\n",type);
                fprintf(fp,"y[1]=\t%10.6f\n",y0[0]);
                fprintf(fp,"y[2]=\t%10.6f\n",y0[1]);
                fprintf(fp,"y[3]=\t%10.6f\n",y0[2]);
                fprintf(fp,"y[4]=\t%10.6f\n",y0[3]);
                fprintf(fp,"lambda=\t%10.6f\n\n",y0[pub_sp.n_eq]);
        }
fclose(fp);

//Running Matlab engine to plot
------------------------------------------------------------------------
n_curve=10; //number of curves per half-branch
l_curve=new int [n_curve];
mxA=new mxArray *[n_curve*pub_sp.counter_br];
is_stable_curve=new bool [n_curve*pub_sp.counter_br];

if (!(eng = engOpen("\0")))
{
        fprintf(stderr, "\nCan't start MATLAB engine\n");
        return;
}

//Data transfer
i_curve_t=0;
for (i=0;i<pub_sp.counter_br;i++)
{
        for (j=0;j<n_curve;j++)
                l_curve[j]=0;
        i_curve=0;
        for (j=1;j<pub_sp.n_fpa[i]-1;j++)
        {
                l_curve[i_curve]+=1;
                is_stable=arg_br[i][j].IsStable();
                is_stable1=arg_br[i][j+1].IsStable();
                if (is_stable!=is_stable1)
                        i_curve+=1;
        }
        l_curve[0]++;
        l_curve[i_curve]++;
        if (pub_sp.n_fpa[i]==1)
                l_curve[0]=1;
        i_curve++;
        jj=0;
        for (j=0;j<i_curve;j++)
        {
                ostrIndex.str("");
                ostrIndex<<i_curve_t;
                strIndex=ostrIndex.str();
                strA="A"+strIndex;
                mxA[i_curve_t]=mxCreateDoubleMatrix(l_curve[j],3,mxREAL);
                dblA=mxGetPr(mxA[i_curve_t]);
                for (k=0;k<l_curve[j];k++)
                {
                        arg_br[i][jj].GetPoint(y0);
                        vol=arg_br[i][jj].GetNorm(0);
                        dblA[k]=vol/*y0[pub_sp.n_eq]*180/pi()*/;
                        dblA[k+l_curve[j]]=y0[0];
                        dblA[k+2*l_curve[j]]=y0[1];
                        jj++;
                }
                engPutVariable(eng,strA.c_str(),mxA[i_curve_t]);
                is_stable_curve[i_curve_t]=arg_br[i][jj-1].IsStable();
                i_curve_t++;
        }
}

//plot formatting
for (i=0;i<i_curve_t;i++)
```

```
        {
                ostrIndex.str("");
                ostrIndex<<i;
                strIndex=ostrIndex.str();
                strA="A"+strIndex;
                if (is_stable_curve[i])
                        strSpec=",'-r','LineWidth',2";
                else
                        strSpec=",'-b','LineWidth',2";
                //strCommand="plot3("+strA+"(:,1),"+strA+"(:,2),"+strA+"(:,3)"+strSpec+");";
                strCommand="plot("+strA+"(:,1),"+strA+"(:,2)"+strSpec+");";
                engEvalString(eng,strCommand.c_str());
                engEvalString(eng,"hold on");
        }
        engEvalString(eng, "xlabel('\\lambda','FontName','Times New Roman');");
        engEvalString(eng, "ylabel('y_1','FontName','Times New Roman');");
        //engEvalString(eng, "zlabel('y_2','FontName','Times New Roman');");

        system ("PAUSE");
        for (i=i_curve_t-1;i>=0;i--)
                mxDestroyArray(mxA[i]);
        engEvalString(eng, "close;");
        engClose(eng);

        delete [] y0;
        delete [] is_stable_curve;
        delete [] l_curve;
        delete [] mxA;
}

void SaveResult(FixedPoint **arg_br)
{
        int i,j,k;
        double *y0,*y1;
        ofstream ofile;
        ostringstream ostrIndex,ostrBIndex;
        string strFileName,strIndex;

        y0=new double [pub_sp.n_eq+1];
        y1=new double [pub_sp.n_norm];
        ostrBIndex.str("");
        ostrBIndex<<pub_sp.branch_index;
        strFileName=pub_sp.strSharedDataPath+"\\specs-"+ostrBIndex.str()+".dat";
        ofile.open(strFileName.c_str(),ios::out|ios::binary);
                ofile.seekp(0,ios::beg);
                ofile.write((char*) &pub_sp.counter_br,sizeof(int));
                for (i=0;i<pub_sp.counter_br;i++)
                        ofile.write((char*) &pub_sp.n_fpa[i],sizeof(int));
                ofile.write((char*) &pub_sp.p[0],sizeof(double));
                ofile.write((char*) &pub_sp.p[1],sizeof(double));
                ofile.write((char*) &pub_sp.n_norm,sizeof(int));
        ofile.close();

        for (i=0;i<pub_sp.counter_br;i++)
        {
                ostrIndex.str("");
                ostrIndex<<(i+1);
                strIndex=ostrIndex.str();
                strFileName=pub_sp.strSharedDataPath+"\\branch"+strIndex+"-"+ostrBIndex.str()+".
    dat";
                ofile.open(strFileName.c_str(),ios::out|ios::binary);
                        ofile.seekp(0,ios::beg);
                        for (j=0;j<pub_sp.n_fpa[i];j++)
                        {
                                arg_br[i][j].GetPoint(y0);
                                arg_br[i][j].GetNorm(y1);
                                for (k=0;k<=pub_sp.n_eq;k++)
                                        ofile.write((char*) (y0+k),sizeof(double));
                                for (k=0;k<pub_sp.n_norm;k++)
                                        ofile.write((char*) (y1+k),sizeof(double));
                        }
```

```
                ofile.close();
        }

        delete [] y0;
        delete [] y1;
}

int main(int argc, char **argv)
{
        FixedPoint **br;
        FixedPoint *fp0=new FixedPoint;
        BranchPoint *bp;

        GetParameters(argv);
        MAllocation(&br,&bp);
        InitialPoint(fp0);
        try
        {
                BranchContinuation(br,bp,fp0);
        }
        catch (ErrorFP err_fp)
        {
                ErrorHandler(err_fp);
        }
        #ifdef _STANDALONE
                PrintResult(br,bp);
        #endif
        SaveResult(br);
        MDelete(&br,&bp);
        #ifdef _EXTERNALPROCESS
                system ("PAUSE");
        #endif
        return 0;
}
```

Appendix H

Stability-analysis code for liquid bridges

This is a C++ project with the header file `stability.h` and source file `stability.cpp`.

stability.h

```
#include <fstream>
#include <string>
#include <sstream>
#include <direct.h>
#include <engine.h>
#include <iomanip>

//#define _STANDALONE                                   //setting 1, option 1
#define _EXTERNALPROCESS                        //setting 1, option 2
//#define _READDATA_V0                          //setting 2, option 1
#define _READDATA_V1                            //setring 2, option 2
//#define _SPHEREBRANCH                         //setting 3
//#define _AXISADJUSTMENT                       //setting 4
//#define _RIGHTCONTACTANGLE                    //setting 5

using namespace std;
const double pi=3.14159265358979323846;
FILE *fp;

struct SolutionP //Solution parameters
{
        int n_norm;
        int n_parts;
        int n_br;                       //Total half-branches
        int *n_brp;                     //Number of branches per part
        int *n_fpa;                     //Actual number of points in a branch from all of half-
    branches
        int n_int0;
        int n_inti;
        int n_int_max;          //Maximum allowable number of intervals for solving ODEs using
    Runge-Kutta routine
        int *n_cr;
        double tol_taw1;        //Criterion on whether taw1 should be regarded as zero or not.
        double tol_msr;     //Maximum tolerable variation in 'D' functions when calculating
    maximal stability region
        double Lambda;
        double th_tm;
        double v_t;
```

```
        double th_cp;
        double taw_cr_inf;      //A large value used to initialized maximum stable arc length
        double vup_pmax;        //Range of volume in which search for locus of maximum pressure
    points (turning points w.r.t. pressure) is done
        double vlow_pmax;       //Range of volume in which search for locus of maximum pressure
    points (turning points w.r.t. pressure) is done
        string strDataPath;
};

struct SolutionB //Solution Branches
{
        double q;
        double taw0;
        double taw1;
        double a;
        double th_m;
        double v;
};

struct StabilityP //Stability parameters
{
        double chi;
        double chi0;
        double chi1;
        double taw10_cr;
        double taw11_cr;
        int st_id;  // 0- Unstable to axisymmetric perturbations, 1- Unstable to antisymmetric
    perturbations, 2- Stable
        int msr;   // 0- Maximal stability region is not reached, 1- Maximal stability region is
    reached
};

struct Grid
{
        int i_0;
        int i_1;
        int i_m;
        int i_e;
        double dt_0;
        double dt_1;
        double dt_m;
        double dt_e;
};
```

stability.cpp

```
#include "stability.h"
#pragma comment(lib,"libmx.lib")
#pragma comment(lib,"libmat.lib")
#pragma comment(lib,"libeng.lib")

// Functions *************************************************
/*Each function has a main and secondary form. The main and secondary form
  are used for when dksi_drho=1 and dksi_drho=-1, respectively.*/

double Coef_A(SolutionB *arg_sb,const double &arg_taw)
{
        return -arg_sb->a*sin(arg_taw)/(1+pow(arg_sb->a,2)+2*arg_sb->a*cos(arg_taw));
}

double Coef_A_(SolutionB *arg_sb,const double &arg_taw)
{
        return (arg_sb->a+2)*sin(arg_taw)/(1+pow(arg_sb->a+2,2)-2*(arg_sb->a+2)*cos(arg_taw));
}

double Coef_B23(SolutionB *arg_sb,const double &arg_taw)
{
        return (pow(arg_sb->a*(arg_sb->a+cos(arg_taw)),2)+pow(1+arg_sb->a*cos(arg_taw),2))/pow(1+
    pow(arg_sb->a,2)+2*arg_sb->a*cos(arg_taw),2);
}
```

```
double Coef_B23_(SolutionB *arg_sb,const double &arg_taw)
{
        return (pow((arg_sb->a+2)*((arg_sb->a+2)-cos(arg_taw)),2)+pow(1-(arg_sb->a+2)*cos(arg_taw)
     ,2))/pow(1+pow(arg_sb->a+2,2)-2*(arg_sb->a+2)*cos(arg_taw),2);
}

double Coef_B4(SolutionB *arg_sb,const double &arg_taw)
{
        return (pow(arg_sb->a*(arg_sb->a+cos(arg_taw)),2)+pow(1+arg_sb->a*cos(arg_taw),2))/pow(1+
     pow(arg_sb->a,2)+2*arg_sb->a*cos(arg_taw),2)-1/(1+pow(arg_sb->a,2)+2*arg_sb->a*cos(arg_taw));
}

double Coef_B4_(SolutionB *arg_sb,const double &arg_taw)
{
        return (pow((arg_sb->a+2)*((arg_sb->a+2)-cos(arg_taw)),2)+pow(1-(arg_sb->a+2)*cos(arg_taw)
     ,2))/pow(1+pow(arg_sb->a+2,2)-2*(arg_sb->a+2)*cos(arg_taw),2)-1/(1+pow(arg_sb->a+2,2)-2*(
     arg_sb->a+2)*cos(arg_taw));
}

inline double Fun_w1(SolutionB *arg_sb,const double &arg_taw)
{
        /*#ifdef _RIGHTCONTACTANGLE
                return sin(arg_taw);    //cylinder
        #else
                return -arg_sb->a*sin(arg_taw)/sqrt(1+pow(arg_sb->a,2)+2*arg_sb->a*cos(arg_taw));
        #endif*/
        return -arg_sb->a*sin(arg_taw)/sqrt(1+pow(arg_sb->a,2)+2*arg_sb->a*cos(arg_taw));
}

inline double Fun_w1_(SolutionB *arg_sb,const double &arg_taw)
{
        return (arg_sb->a+2)*sin(arg_taw)/sqrt(1+pow(arg_sb->a+2,2)-2*(arg_sb->a+2)*cos(arg_taw));
}

inline double Fun_w5(SolutionB *arg_sb,const double &arg_taw)
{
        return (1+arg_sb->a*cos(arg_taw))/sqrt(1+pow(arg_sb->a,2)+2*arg_sb->a*cos(arg_taw));
}

inline double Fun_w5_(SolutionB *arg_sb,const double &arg_taw)
{
        return (1-(arg_sb->a+2)*cos(arg_taw))/sqrt(1+pow(arg_sb->a+2,2)-2*(arg_sb->a+2)*cos(
     arg_taw));
}

inline double Fun_rho(SolutionB *arg_sb,const double &arg_taw)
{
        return sqrt(1+pow(arg_sb->a,2)+2*arg_sb->a*cos(arg_taw));
}

inline double Fun_rho_(SolutionB *arg_sb,const double &arg_taw)
{
        return sqrt(1+pow(arg_sb->a+2,2)-2*(arg_sb->a+2)*cos(arg_taw));
}

inline int Sign(const double &arg_x)
{
        if (arg_x>0)
                return 1;
        else
                return -1;
}
//Type generic form of Determinant in case they are used with MPFR data types
template <class T_real>
inline T_real Det2(T_real **arg_x)
{
        return arg_x[1][1]*arg_x[0][0]-arg_x[1][0]*arg_x[0][1];
}

template <class T_real>
```

```
inline T_real Det3(T_real **arg_x)
{
        return arg_x[0][0]*arg_x[1][1]*arg_x[2][2]-arg_x[0][0]*arg_x[1][2]*arg_x[2][1]+
                arg_x[0][1]*arg_x[1][2]*arg_x[2][0]-arg_x[0][1]*arg_x[1][0]*arg_x[2][2]+
                arg_x[0][2]*arg_x[1][0]*arg_x[2][1]-arg_x[0][2]*arg_x[1][1]*arg_x[2][0];
}
//Subroutines ******************************************************
void CalcTheta_tm(SolutionP &arg_sp)
{
        double tol,dth;

        tol=1e-8;
        arg_sp.th_tm=pi/2;
        while(1)
        {
                dth=(cos(arg_sp.th_tm)+1-arg_sp.Lambda*sin(arg_sp.th_tm))/(sin(arg_sp.th_tm)+
    arg_sp.Lambda*cos(arg_sp.th_tm));
                arg_sp.th_tm+=dth;
                if (fabs(dth)<tol)
                        break;
        }
        arg_sp.v_t=(0.5-0.25*pow(cos(arg_sp.th_tm),3)+0.75*cos(arg_sp.th_tm))/pow(sin(arg_sp.th_tm
    ),3);

}

void CalcVolume(SolutionP &arg_sp,SolutionB **arg_sb)
{
        int i,j,k,n_inv;
        double t,dt;

        for (i=0;i<arg_sp.n_br;i++)
        {
                for (j=0;j<arg_sp.n_fpa[i];j++)
                {
                        n_inv=arg_sp.n_int0+(int) (arg_sp.n_inti*fabs(arg_sb[i][j].taw1-arg_sb[i][
    j].taw0));
                        dt=(arg_sb[i][j].taw1-arg_sb[i][j].taw0)/n_inv;
                        arg_sb[i][j].v=0;
                        t=arg_sb[i][j].taw0;
                        if (arg_sb[i][0].q<0)
                        {
                                for (k=0;k<n_inv;k++)
                                {
                                        arg_sb[i][j].v+=(1+arg_sb[i][j].a*cos(t))*sqrt(1+pow(
    arg_sb[i][j].a,2)+2*arg_sb[i][j].a*cos(t));
                                        t+=dt;
                                        arg_sb[i][j].v+=(1+arg_sb[i][j].a*cos(t))*sqrt(1+pow(
    arg_sb[i][j].a,2)+2*arg_sb[i][j].a*cos(t));
                                }
                        }
                        else
                        {
                                for (k=0;k<n_inv;k++)
                                {
                                        arg_sb[i][j].v+=(1-(arg_sb[i][j].a+2)*cos(t))*sqrt(1+pow(
    arg_sb[i][j].a+2,2)-2*(arg_sb[i][j].a+2)*cos(t));
                                        t+=dt;
                                        arg_sb[i][j].v+=(1-(arg_sb[i][j].a+2)*cos(t))*sqrt(1+pow(
    arg_sb[i][j].a+2,2)-2*(arg_sb[i][j].a+2)*cos(t));
                                }
                        }
                        arg_sb[i][j].v=-arg_sb[i][j].v*(0.75*dt/pow(arg_sb[i][j].q,3)/2);
                }
        }
}

void ReadSolutionParameter(SolutionP &arg_sp)
{
        int i,j,k;
        ifstream ifile[10];
```

```
        ostringstream ostrIndex;
        string strFileName,strIndex;

        arg_sp.n_brp=new int [arg_sp.n_parts];
        arg_sp.n_br=0;
        for (i=0;i<arg_sp.n_parts;i++)
        {
                ostrIndex.str("");
                ostrIndex<<(i+1);
                strIndex=ostrIndex.str();
                strFileName=arg_sp.strDataPath+"\\specs-"+strIndex+".dat";
                ifile[i].open(strFileName.c_str(),ios::in|ios::binary);
                        ifile[i].seekg(0,ios::beg);
                        ifile[i].read((char*) (arg_sp.n_brp+i),sizeof(int));
                        arg_sp.n_br+=arg_sp.n_brp[i];
        }
        arg_sp.n_fpa=new int [arg_sp.n_br];

        k=0;
        for (i=0;i<arg_sp.n_parts;i++)
        {
                for (j=0;j<arg_sp.n_brp[i];j++)
                {
                        ifile[i].read((char*) &arg_sp.n_fpa[k],sizeof(int));
                        k++;
                }
        }
        // Assuming Lambda, th_cp, n_norm are identical for all the parts
        ifile[0].read((char*) &arg_sp.Lambda,sizeof(double));
        ifile[0].read((char*) &arg_sp.th_cp,sizeof(double));
        #ifdef _READDATA_V1
                ifile[0].read((char*) &arg_sp.n_norm,sizeof(int));
        #endif
        for (i=0;i<arg_sp.n_parts;i++)
                ifile[i].close();
        #ifdef _SPHEREBRANCH
                CalcTheta_tm(arg_sp);
        #endif
}

void ReadSolutionBranch(SolutionP &arg_sp,SolutionB **arg_sb)
{
        int i,j,k,kk;
        ifstream ifile;
        ostringstream ostrIndex;
        string strIndexi,strIndexk,strFileName;

        kk=0;
        for (k=0;k<arg_sp.n_parts;k++)
        {
                ostrIndex.str("");
                ostrIndex<<(k+1);
                strIndexk=ostrIndex.str();
                for (i=0;i<arg_sp.n_brp[k];i++)
                {
                        ostrIndex.str("");
                        ostrIndex<<(i+1);
                        strIndexi=ostrIndex.str();
                        strFileName=arg_sp.strDataPath+"\\branch"+strIndexi+"-"+strIndexk+".dat";
                        ifile.open(strFileName.c_str(),ios::in|ios::binary);
                                ifile.seekg(0,ios::beg);
                                for (j=0;j<arg_sp.n_fpa[kk];j++)
                                {
                                        ifile.read((char*) &arg_sb[kk][j].q,sizeof(double));
                                        ifile.read((char*) &arg_sb[kk][j].taw0,sizeof(double));
                                        ifile.read((char*) &arg_sb[kk][j].taw1,sizeof(double));
                                        ifile.read((char*) &arg_sb[kk][j].a,sizeof(double));
                                        ifile.read((char*) &arg_sb[kk][j].th_m,sizeof(double));
                                        #ifdef _READDATA_V1
                                                ifile.read((char*) &arg_sb[kk][j].v,sizeof(double)
    );
```

```
                                #endif
                        }
                ifile.close();
                kk++;
            }
    }
    #ifdef _READDATA_V0
            CalcVolume(arg_sp,arg_sb);
    #endif
}

void MAllocation(SolutionP &arg_sp,SolutionB ***arg_sb,StabilityP ***arg_stp)
{
        int i;

        *arg_sb=new SolutionB *[arg_sp.n_br];
        *arg_stp=new StabilityP *[arg_sp.n_br];
        for (i=0;i<arg_sp.n_br;i++)
        {
                *(*arg_sb+i)=new SolutionB [arg_sp.n_fpa[i]];
                *(*arg_stp+i)=new StabilityP [arg_sp.n_fpa[i]];
        }
}

void MDelete(SolutionP &arg_sp,SolutionB ***arg_sb,SolutionB ***arg_sbcr,StabilityP ***arg_stp)
{
        int i;

        delete [] arg_sp.n_fpa;
        delete [] arg_sp.n_brp;
        for (i=arg_sp.n_br-1;i>=0;--i)
        {
                delete [] *(*arg_sb+i);
                delete [] *(*arg_sbcr+i);
                delete [] *(*arg_stp+i);
        }
        delete [] *arg_sb;
        delete [] *arg_sbcr;
        delete [] *arg_stp;
        delete [] arg_sp.n_cr;
}

void GetParameters(SolutionP &arg_sp,char **argv)
{
        char *chrAppPath=NULL;
        string strAppPath;

        arg_sp.n_int0=300;
        arg_sp.n_inti=200;
        arg_sp.tol_taw1=1e-8;
        arg_sp.tol_msr=1e-20;
        arg_sp.taw_cr_inf=100;
        arg_sp.n_int_max=3000;
        #ifdef _STANDALONE
                arg_sp.n_parts=1;
                arg_sp.vlow_pmax=0;
                arg_sp.vup_pmax=10;
                chrAppPath=_getcwd(chrAppPath,100);
                strAppPath=chrAppPath;
                arg_sp.strDataPath=strAppPath+"\\data";
        #endif
        #ifdef _EXTERNALPROCESS
                arg_sp.n_parts=atoi(argv[1]);
                arg_sp.vlow_pmax=atof(argv[2]);
                arg_sp.vup_pmax=atof(argv[3]);
                arg_sp.strDataPath="C:\\Amir\\PhD - McGill\\PhD project\\Axi-meniscus\\shared data
     \\branch data";
        #endif
}
```

```
inline void RungeKutta(Grid &arg_grd,const double &arg_bcd,const double &arg_bcn,SolutionB *arg_sb
     ,double (*ptCoefA)(SolutionB *,const double &),
                                              double (*ptCoefB)(SolutionB *,const double &),
     const double arg_coefC,double *arg_w)
{
        int i;
        double *y1,*y2,t;
        double k[2][4];

        y1=new double [arg_grd.i_e+1];
        y2=new double [arg_grd.i_e+1];

        y1[0]=arg_bcd;
        y2[0]=arg_bcn;
        t=0;
        for(i=0;i<arg_grd.i_m;i++)
        {
                k[0][0]=arg_grd.dt_m*y2[i];
                k[1][0]=-arg_grd.dt_m*(arg_coefC+ptCoefB(arg_sb,t)*y1[i]+ptCoefA(arg_sb,t)*y2[i]);

                t+=0.5*arg_grd.dt_m;
                k[0][1]=arg_grd.dt_m*(y2[i]+0.5*k[1][0]);
                k[1][1]=-arg_grd.dt_m*(arg_coefC+ptCoefB(arg_sb,t)*(y1[i]+0.5*k[0][0])+ptCoefA(
     arg_sb,t)*(y2[i]+0.5*k[1][0]));

                k[0][2]=arg_grd.dt_m*(y2[i]+0.5*k[1][1]);
                k[1][2]=-arg_grd.dt_m*(arg_coefC+ptCoefB(arg_sb,t)*(y1[i]+0.5*k[0][1])+ptCoefA(
     arg_sb,t)*(y2[i]+0.5*k[1][1]));

                t+=0.5*arg_grd.dt_m;
                k[0][3]=arg_grd.dt_m*(y2[i]+k[1][2]);
                k[1][3]=-arg_grd.dt_m*(arg_coefC+ptCoefB(arg_sb,t)*(y1[i]+k[0][2])+ptCoefA(arg_sb,
     t)*(y2[i]+k[1][2]));

                y1[i+1]=y1[i]+(k[0][0]+2*k[0][1]+2*k[0][2]+k[0][3])/6;
                y2[i+1]=y2[i]+(k[1][0]+2*k[1][1]+2*k[1][2]+k[1][3])/6;
        }
        for(i=arg_grd.i_m;i<arg_grd.i_e;i++)
        {
                k[0][0]=arg_grd.dt_e*y2[i];
                k[1][0]=-arg_grd.dt_e*(arg_coefC+ptCoefB(arg_sb,t)*y1[i]+ptCoefA(arg_sb,t)*y2[i]);

                t+=0.5*arg_grd.dt_e;
                k[0][1]=arg_grd.dt_e*(y2[i]+0.5*k[1][0]);
                k[1][1]=-arg_grd.dt_e*(arg_coefC+ptCoefB(arg_sb,t)*(y1[i]+0.5*k[0][0])+ptCoefA(
     arg_sb,t)*(y2[i]+0.5*k[1][0]));

                k[0][2]=arg_grd.dt_e*(y2[i]+0.5*k[1][1]);
                k[1][2]=-arg_grd.dt_e*(arg_coefC+ptCoefB(arg_sb,t)*(y1[i]+0.5*k[0][1])+ptCoefA(
     arg_sb,t)*(y2[i]+0.5*k[1][1]));

                t+=0.5*arg_grd.dt_e;
                k[0][3]=arg_grd.dt_e*(y2[i]+k[1][2]);
                k[1][3]=-arg_grd.dt_e*(arg_coefC+ptCoefB(arg_sb,t)*(y1[i]+k[0][2])+ptCoefA(arg_sb,
     t)*(y2[i]+k[1][2]));

                y1[i+1]=y1[i]+(k[0][0]+2*k[0][1]+2*k[0][2]+k[0][3])/6;
                y2[i+1]=y2[i]+(k[1][0]+2*k[1][1]+2*k[1][2]+k[1][3])/6;
        }

        for(i=0;i<=arg_grd.i_e;i++)
                arg_w[i]=y1[i];

        delete [] y1;
        delete [] y2;
}

void GridGen(Grid &arg_grd,SolutionP &arg_sp,SolutionB *arg_sb,double arg_taw)
{
        int n_m,n_e;
        double taw0_a,taw1_a,dtaw_a;
```

```
taw0_a=fabs(arg_sb->taw0);
taw1_a=fabs(arg_taw);
if (taw0_a<taw1_a)
{
        dtaw_a=taw1_a-taw0_a;
        if (taw0_a<dtaw_a)
        {
                #ifdef _RIGHTCONTACTANGLE
                        n_m=0;
                        n_e=arg_sp.n_int0;
                #else
                        n_m=arg_sp.n_int0;
                        n_e=(int) (n_m*dtaw_a/taw0_a);
                #endif
                if (n_e>arg_sp.n_int_max)
                        n_e=arg_sp.n_int_max;
        }
        else
        {
                n_e=arg_sp.n_int0;
                n_m=(int) (n_e*taw0_a/dtaw_a);
                if (n_m>arg_sp.n_int_max)
                        n_m=arg_sp.n_int_max;
        }
        arg_grd.i_0=n_m;
        arg_grd.i_1=n_m+n_e;
        arg_grd.i_m=arg_grd.i_0;
        arg_grd.i_e=arg_grd.i_1;
        #ifdef _RIGHTCONTACTANGLE
                arg_grd.dt_m=taw0_a/n_e;
        #else
                arg_grd.dt_m=taw0_a/n_m;
        #endif
        arg_grd.dt_e=dtaw_a/n_e;
        arg_grd.dt_0=arg_grd.dt_m;
        arg_grd.dt_1=arg_grd.dt_e;
}
else
{
        if (taw1_a>arg_sp.tol_taw1)
        {
                dtaw_a=taw0_a-taw1_a;
                if (taw1_a<dtaw_a)
                {
                        n_m=arg_sp.n_int0;
                        n_e=(int) (n_m*dtaw_a/taw1_a);
                        if (n_e>arg_sp.n_int_max)
                                n_e=arg_sp.n_int_max;
                }
                else
                {
                        n_e=arg_sp.n_int0;
                        n_m=(int) (n_e*taw1_a/dtaw_a);
                        if (n_m>arg_sp.n_int_max)
                                n_m=arg_sp.n_int_max;
                }
                arg_grd.i_0=n_m+n_e;
                arg_grd.i_1=n_m;
                arg_grd.i_m=arg_grd.i_1;
                arg_grd.i_e=arg_grd.i_0;
                arg_grd.dt_m=taw1_a/n_m;
                arg_grd.dt_e=dtaw_a/n_e;
                arg_grd.dt_0=arg_grd.dt_e;
                arg_grd.dt_1=arg_grd.dt_m;
        }
        else
        {
                dtaw_a=taw0_a-taw1_a;
                n_m=0;
                n_e=arg_sp.n_int0;
```

```
                        arg_grd.i_0=n_m+n_e;
                        arg_grd.i_1=n_m;
                        arg_grd.i_m=arg_grd.i_1;
                        arg_grd.i_e=arg_grd.i_0;
                        arg_grd.dt_m=dtaw_a/n_e;
                        arg_grd.dt_e=dtaw_a/n_e;
                        arg_grd.dt_0=arg_grd.dt_e;
                        arg_grd.dt_1=arg_grd.dt_m;
                }
        }
}

void CriticalChi(SolutionP &arg_sp,SolutionB *arg_sb,StabilityP *arg_stp)
{
        int i;
        int sgn0,sgn1;
        double t,*rho,**w,**ddw;
        double *d0,*d1,**iw_t,iw[3],temp0,temp1;
        double **num0,**num1,**den0,**den1;
        Grid grd;

        //Memory allocation --------------------------------
        num1=new double *[2];
        for (i=0;i<2;i++)
                *(num1+i)=new double [2];
        num0=new double *[3];
        for (i=0;i<3;i++)
                *(num0+i)=new double [3];
        den1=new double *[2];
        for (i=0;i<2;i++)
                *(den1+i)=new double [2];
        den0=new double *[3];
        for (i=0;i<3;i++)
                *(den0+i)=new double [3];
        ddw=new double *[5];
        for (i=0;i<5;i++)
                *(ddw+i)=new double [2];
        // Grid ---------------------------------------------
        sgn0=Sign(arg_sb->taw0);
        sgn1=Sign(arg_sb->taw1);
        GridGen(grd,arg_sp,arg_sb,arg_sb->taw1);

        //Solution ----------------------------------------------------
        w=new double *[5];
        for (i=0;i<5;i++)
                *(w+i)=new double [grd.i_e+1];
        rho=new double [grd.i_e+1];
        if (arg_sb->q<0)
        {
                RungeKutta(grd,1,0,arg_sb,&Coef_A,&Coef_B23,0,*(w+1));
                RungeKutta(grd,1,0,arg_sb,&Coef_A,&Coef_B23,1,*(w+2));
                RungeKutta(grd,0,1,arg_sb,&Coef_A,&Coef_B4,0,*(w+3));

                t=0;
                for(i=0;i<grd.i_m;i++)
                {
                        rho[i]=Fun_rho(arg_sb,t);
                        w[0][i]=Fun_w1(arg_sb,t);
                        w[4][i]=Fun_w5(arg_sb,t);
                        t+=grd.dt_m;
                }
                for(i=grd.i_m;i<=grd.i_e;i++)
                {
                        rho[i]=Fun_rho(arg_sb,t);
                        w[0][i]=Fun_w1(arg_sb,t);
                        w[4][i]=Fun_w5(arg_sb,t);
                        t+=grd.dt_e;
                }
        }
        else
        {
```

```
            RungeKutta(grd,1,0,arg_sb,&Coef_A_,&Coef_B23_,0,*(w+1));
            RungeKutta(grd,1,0,arg_sb,&Coef_A_,&Coef_B23_,1,*(w+2));
            RungeKutta(grd,0,1,arg_sb,&Coef_A_,&Coef_B4_,0,*(w+3));

            t=0;
            for(i=0;i<grd.i_m;i++)
            {
                    rho[i]=Fun_rho_(arg_sb,t);
                    w[0][i]=Fun_w1_(arg_sb,t);
                    w[4][i]=Fun_w5_(arg_sb,t);
                    t+=grd.dt_m;
            }
            for(i=grd.i_m;i<=grd.i_e;i++)
            {
                    rho[i]=Fun_rho_(arg_sb,t);
                    w[0][i]=Fun_w1_(arg_sb,t);
                    w[4][i]=Fun_w5_(arg_sb,t);
                    t+=grd.dt_e;
            }
    }

    //Derivatives at boundaries --------------------------------------------------
    if (fabs(arg_sb->taw1)>arg_sp.tol_taw1)
    {
            //ddw[0][0]=(3*w[0][grd.i_0]-4*w[0][grd.i_0-1]+w[0][grd.i_0-2])/grd.dt_0/2;
            ddw[0][1]=(3*w[0][grd.i_1]-4*w[0][grd.i_1-1]+w[0][grd.i_1-2])/grd.dt_1/2;
            //ddw[1][0]=sgn0*(3*w[1][grd.i_0]-4*w[1][grd.i_0-1]+w[1][grd.i_0-2])/grd.dt_0/2;
            ddw[1][1]=sgn1*(3*w[1][grd.i_1]-4*w[1][grd.i_1-1]+w[1][grd.i_1-2])/grd.dt_1/2;
            //ddw[2][0]=sgn0*(3*w[2][grd.i_0]-4*w[2][grd.i_0-1]+w[2][grd.i_0-2])/grd.dt_0/2;
            ddw[2][1]=sgn1*(3*w[2][grd.i_1]-4*w[2][grd.i_1-1]+w[2][grd.i_1-2])/grd.dt_1/2;
            //ddw[3][0]=(3*w[3][grd.i_0]-4*w[3][grd.i_0-1]+w[3][grd.i_0-2])/grd.dt_0/2;
            ddw[3][1]=(3*w[3][grd.i_1]-4*w[3][grd.i_1-1]+w[3][grd.i_1-2])/grd.dt_1/2;
            //ddw[4][0]=sgn0*(3*w[4][grd.i_0]-4*w[4][grd.i_0-1]+w[4][grd.i_0-2])/grd.dt_0/2;
            ddw[4][1]=sgn1*(3*w[4][grd.i_1]-4*w[4][grd.i_1-1]+w[4][grd.i_1-2])/grd.dt_1/2;
    }
    else
    {
            ddw[0][1]=1;
            ddw[1][1]=0;
            ddw[2][1]=0;
            ddw[3][1]=1;
            ddw[4][1]=0;
    }

    //Integrals & maximal stability length
-----------------------------------------------------------
    arg_stp->taw10_cr=arg_sp.taw_cr_inf;
    arg_stp->taw11_cr=arg_sp.taw_cr_inf;

    if (arg_sb->taw0*arg_sb->taw1>0)
    {
            iw_t=new double *[3];
            for (i=0;i<3;i++)
                    *(iw_t+i)=new double [grd.i_e-grd.i_m+1];
            d0=new double [grd.i_e-grd.i_m+1];
            d1=new double [grd.i_e-grd.i_m+1];
            iw_t[0][0]=0;
            iw_t[1][0]=0;
            iw_t[2][0]=0;
            for (i=grd.i_m;i<grd.i_e;i++)
            {
                    iw_t[0][i-grd.i_m+1]=iw_t[0][i-grd.i_m]+(rho[i]*w[0][i]+rho[i+1]*w[0][i
+1])*sgn0*grd.dt_e/2;
                    iw_t[1][i-grd.i_m+1]=iw_t[1][i-grd.i_m]+(rho[i]*w[1][i]+rho[i+1]*w[1][i
+1])*grd.dt_e/2;
                    iw_t[2][i-grd.i_m+1]=iw_t[2][i-grd.i_m]+(rho[i]*w[2][i]+rho[i+1]*w[2][i
+1])*grd.dt_e/2;
            }
            iw[0]=iw_t[0][grd.i_e-grd.i_m];
            iw[1]=iw_t[1][grd.i_e-grd.i_m];
            iw[2]=iw_t[2][grd.i_e-grd.i_m];
```

```
            d0[0]=d1[0]=0;
            if (arg_sb->taw0>0)
            {
                    den0[0][0]=w[0][grd.i_0];       //positive sign
                    den0[0][1]=w[1][grd.i_0];
                    den0[0][2]=w[2][grd.i_0];

                    den1[0][0]=w[3][grd.i_0];       //positive sign
                    den1[0][1]=w[4][grd.i_0];
                    for (i=grd.i_m;i<grd.i_e;i++)
                    {
                            den0[1][0]=w[0][i+1];   //positive sign
                            den0[1][1]=w[1][i+1];
                            den0[1][2]=w[2][i+1];
                            den0[2][0]=iw_t[0][i-grd.i_m+1];
                            den0[2][1]=iw_t[1][i-grd.i_m+1];
                            den0[2][2]=iw_t[2][i-grd.i_m+1];

                            den1[1][0]=w[3][i+1];   //positive sign
                            den1[1][1]=w[4][i+1];
                            d0[i-grd.i_m+1]=Det3(den0);
                            d1[i-grd.i_m+1]=Det2(den1);

                    }
            }
            else
            {
                    den0[0][0]=-w[0][grd.i_0];      //negative sign
                    den0[0][1]=w[1][grd.i_0];
                    den0[0][2]=w[2][grd.i_0];

                    den1[0][0]=-w[3][grd.i_0];      //negative sign
                    den1[0][1]=w[4][grd.i_0];
                    for (i=grd.i_e;i>grd.i_m;i--)
                    {
                            den0[1][0]=-w[0][i-1];  //negative sign
                            den0[1][1]=w[1][i-1];
                            den0[1][2]=w[2][i-1];
                            den0[2][0]=iw[0]-iw_t[0][i-grd.i_m-1];
                            den0[2][1]=iw[1]-iw_t[1][i-grd.i_m-1];
                            den0[2][2]=iw[2]-iw_t[2][i-grd.i_m-1];

                            den1[1][0]=-w[3][i-1];  //negative sign
                            den1[1][1]=w[4][i-1];
                            d0[grd.i_e-i+1]=Det3(den0);
                            d1[grd.i_e-i+1]=Det2(den1);
                    }
            }
            t=arg_sb->taw0+grd.dt_e;
            for (i=2;i<grd.i_e-grd.i_m;i++)
            {
                    t+=grd.dt_e;
                    if (d0[i]*d0[i+1]<0 && fabs(d0[i])>arg_sp.tol_msr && fabs(d0[i+1])>arg_sp.
tol_msr)
                    {
                            arg_stp->taw10_cr=t+0.5*grd.dt_e;
                            break;
                    }
            }
            t=arg_sb->taw0+grd.dt_e;
            for (i=2;i<grd.i_e-grd.i_m;i++)
            {
                    t+=grd.dt_e;
                    if (d1[i]*d1[i+1]<0 && fabs(d1[i])>arg_sp.tol_msr && fabs(d1[i+1])>arg_sp.
tol_msr)
                    {
                            arg_stp->taw11_cr=t+0.5*grd.dt_e;
                            break;
                    }
            }
   }
```

```
    else
    {
            iw_t=new double *[3];
            for (i=0;i<3;i++)
                    *(iw_t+i)=new double [grd.i_e+grd.i_m+1];
            d0=new double [grd.i_e+grd.i_m+1];
            d1=new double [grd.i_e+grd.i_m+1];

            iw_t[0][0]=0;
            iw_t[1][0]=0;
            iw_t[2][0]=0;
            if (fabs(arg_sb->taw0)<fabs(arg_sb->taw1))
            {
                    for (i=grd.i_m;i>0;i--)
                    {
                            iw_t[0][grd.i_m-i+1]=iw_t[0][grd.i_m-i]+sgn0*(rho[i]*w[0][i]+rho[i
-1]*w[0][i-1])*grd.dt_m/2;
                            iw_t[1][grd.i_m-i+1]=iw_t[1][grd.i_m-i]+(rho[i]*w[1][i]+rho[i-1]*w
[1][i-1])*grd.dt_m/2;
                            iw_t[2][grd.i_m-i+1]=iw_t[2][grd.i_m-i]+(rho[i]*w[2][i]+rho[i-1]*w
[2][i-1])*grd.dt_m/2;
                    }
                    for (i=0;i<grd.i_m;i++)
                    {
                            iw_t[0][grd.i_m+i+1]=iw_t[0][grd.i_m+i]+(rho[i]*w[0][i]+rho[i+1]*w
[0][i+1])*grd.dt_m/2;
                            iw_t[1][grd.i_m+i+1]=iw_t[1][grd.i_m+i]+(rho[i]*w[1][i]+rho[i+1]*w
[1][i+1])*grd.dt_m/2;
                            iw_t[2][grd.i_m+i+1]=iw_t[2][grd.i_m+i]+(rho[i]*w[2][i]+rho[i+1]*w
[2][i+1])*grd.dt_m/2;
                    }
                    for (i=grd.i_m;i<grd.i_e;i++)
                    {
                            iw_t[0][grd.i_m+i+1]=iw_t[0][grd.i_m+i]+(rho[i]*w[0][i]+rho[i+1]*w
[0][i+1])*grd.dt_e/2;
                            iw_t[1][grd.i_m+i+1]=iw_t[1][grd.i_m+i]+(rho[i]*w[1][i]+rho[i+1]*w
[1][i+1])*grd.dt_e/2;
                            iw_t[2][grd.i_m+i+1]=iw_t[2][grd.i_m+i]+(rho[i]*w[2][i]+rho[i+1]*w
[2][i+1])*grd.dt_e/2;
                    }
            }
            else
            {
                    for (i=grd.i_e;i>grd.i_m;i--)
                    {
                            iw_t[0][grd.i_e-i+1]=iw_t[0][grd.i_e-i]+sgn0*(rho[i]*w[0][i]+rho[i
-1]*w[0][i-1])*grd.dt_e/2;
                            iw_t[1][grd.i_e-i+1]=iw_t[1][grd.i_e-i]+(rho[i]*w[1][i]+rho[i-1]*w
[1][i-1])*grd.dt_e/2;
                            iw_t[2][grd.i_e-i+1]=iw_t[2][grd.i_e-i]+(rho[i]*w[2][i]+rho[i-1]*w
[2][i-1])*grd.dt_e/2;
                    }
                    for (i=grd.i_m;i>0;i--)
                    {
                            iw_t[0][grd.i_e-i+1]=iw_t[0][grd.i_e-i]+sgn0*(rho[i]*w[0][i]+rho[i
-1]*w[0][i-1])*grd.dt_m/2;
                            iw_t[1][grd.i_e-i+1]=iw_t[1][grd.i_e-i]+(rho[i]*w[1][i]+rho[i-1]*w
[1][i-1])*grd.dt_m/2;
                            iw_t[2][grd.i_e-i+1]=iw_t[2][grd.i_e-i]+(rho[i]*w[2][i]+rho[i-1]*w
[2][i-1])*grd.dt_m/2;
                    }
                    for (i=0;i<grd.i_m;i++)
                    {
                            iw_t[0][grd.i_e+i+1]=iw_t[0][grd.i_e+i]+(rho[i]*w[0][i]+rho[i+1]*w
[0][i+1])*grd.dt_m/2;
                            iw_t[1][grd.i_e+i+1]=iw_t[1][grd.i_e+i]+(rho[i]*w[1][i]+rho[i+1]*w
[1][i+1])*grd.dt_m/2;
                            iw_t[2][grd.i_e+i+1]=iw_t[2][grd.i_e+i]+(rho[i]*w[2][i]+rho[i+1]*w
[2][i+1])*grd.dt_m/2;
                    }
            }
```

```
iw[0]=iw_t[0][grd.i_e+grd.i_m];
iw[1]=iw_t[1][grd.i_e+grd.i_m];
iw[2]=iw_t[2][grd.i_e+grd.i_m];

d0[0]=d1[0]=0;
den0[0][0]=-w[0][grd.i_0];      //negative sign
den0[0][1]=w[1][grd.i_0];
den0[0][2]=w[2][grd.i_0];

den1[0][0]=-w[3][grd.i_0];      //negative sign
den1[0][1]=w[4][grd.i_0];
if (fabs(arg_sb->taw0)<fabs(arg_sb->taw1))
{
        for (i=grd.i_m;i>0;i--)
        {
                den0[1][0]=-w[0][i-1];  //negative sign
                den0[1][1]=w[1][i-1];
                den0[1][2]=w[2][i-1];
                den0[2][0]=iw_t[0][grd.i_m-i+1];
                den0[2][1]=iw_t[1][grd.i_m-i+1];
                den0[2][2]=iw_t[2][grd.i_m-i+1];

                den1[1][0]=-w[3][i-1];  //negative sign
                den1[1][1]=w[4][i-1];
                d0[grd.i_m-i+1]=Det3(den0);
                d1[grd.i_m-i+1]=Det2(den1);
        }
        for (i=0;i<grd.i_m;i++)
        {
                den0[1][0]=w[0][i+1];   //positive sign
                den0[1][1]=w[1][i+1];
                den0[1][2]=w[2][i+1];
                den0[2][0]=iw_t[0][grd.i_m+i+1];
                den0[2][1]=iw_t[1][grd.i_m+i+1];
                den0[2][2]=iw_t[2][grd.i_m+i+1];

                den1[1][0]=w[3][i+1];   //positive sign
                den1[1][1]=w[4][i+1];

                d0[grd.i_m+i+1]=Det3(den0);
                d1[grd.i_m+i+1]=Det2(den1);
        }
        for (i=grd.i_m;i<grd.i_e;i++)
        {
                den0[1][0]=w[0][i+1];   //positive sign
                den0[1][1]=w[1][i+1];
                den0[1][2]=w[2][i+1];
                den0[2][0]=iw_t[0][grd.i_m+i+1];
                den0[2][1]=iw_t[1][grd.i_m+i+1];
                den0[2][2]=iw_t[2][grd.i_m+i+1];

                den1[1][0]=w[3][i+1];   //positive sign
                den1[1][1]=w[4][i+1];

                d0[grd.i_m+i+1]=Det3(den0);
                d1[grd.i_m+i+1]=Det2(den1);
        }
}
else
{
        for (i=grd.i_e;i>grd.i_m;i--)
        {
                den0[1][0]=-w[0][i-1];  //negative sign
                den0[1][1]=w[1][i-1];
                den0[1][2]=w[2][i-1];
                den0[2][0]=iw_t[0][grd.i_e-i+1];
                den0[2][1]=iw_t[1][grd.i_e-i+1];
                den0[2][2]=iw_t[2][grd.i_e-i+1];

                den1[1][0]=-w[3][i-1];  //negative sign
                den1[1][1]=w[4][i-1];
```

```
                        d0[grd.i_e-i+1]=Det3(den0);
                        d1[grd.i_e-i+1]=Det2(den1);
                }
                for (i=grd.i_m;i>0;i--)
                {
                        den0[1][0]=-w[0][i-1];  //negative sign
                        den0[1][1]=w[1][i-1];
                        den0[1][2]=w[2][i-1];
                        den0[2][0]=iw_t[0][grd.i_e-i+1];
                        den0[2][1]=iw_t[1][grd.i_e-i+1];
                        den0[2][2]=iw_t[2][grd.i_e-i+1];

                        den1[1][0]=-w[3][i-1];  //negative sign
                        den1[1][1]=w[4][i-1];

                        d0[grd.i_e-i+1]=Det3(den0);
                        d1[grd.i_e-i+1]=Det2(den1);
                }
                for (i=0;i<grd.i_m;i++)
                {
                        den0[1][0]=w[0][i+1];   //positive sign
                        den0[1][1]=w[1][i+1];
                        den0[1][2]=w[2][i+1];
                        den0[2][0]=iw_t[0][grd.i_e+i+1];
                        den0[2][1]=iw_t[1][grd.i_e+i+1];
                        den0[2][2]=iw_t[2][grd.i_e+i+1];

                        den1[1][0]=w[3][i+1];   //positive sign
                        den1[1][1]=w[4][i+1];

                        d0[grd.i_e+i+1]=Det3(den0);
                        d1[grd.i_e+i+1]=Det2(den1);
                }
        }
        if (fabs(arg_sb->taw0)<fabs(arg_sb->taw1))
        {
                t=arg_sb->taw0+2*grd.dt_m;
                for (i=2;i<2*grd.i_m;i++)
                {
                        if (d0[i]*d0[i+1]<0 && fabs(d0[i])>arg_sp.tol_msr && fabs(d0[i+1])
>arg_sp.tol_msr)
                        {
                                arg_stp->taw10_cr=t+0.5*grd.dt_m;
                                break;
                        }
                        t+=grd.dt_m;
                }
                if (arg_stp->taw10_cr==arg_sp.taw_cr_inf)
                {
                        /*#ifdef _RIGHTCONTACTANGLE
                        for (i=2*grd.i_m+2;i<grd.i_m+grd.i_e;i++) //For cylinder-> 2*grd.
i_m+2
                        #else
                        for (i=2*grd.i_m;i<grd.i_m+grd.i_e;i++)
                        #endif*/
                        for (i=2*grd.i_m;i<grd.i_m+grd.i_e;i++)
                        {
                                if (d0[i]*d0[i+1]<0 && fabs(d0[i])>arg_sp.tol_msr && fabs(
d0[i+1])>arg_sp.tol_msr)
                                {
                                        arg_stp->taw10_cr=t+0.5*grd.dt_e;
                                        break;
                                }
                                t+=grd.dt_e;
                        }
                }
                t=arg_sb->taw0+2*grd.dt_m;
                for (i=2;i<=2*grd.i_m;i++)
                {
                        if (d1[i]*d1[i+1]<0 && fabs(d1[i])>arg_sp.tol_msr && fabs(d1[i+1])
>arg_sp.tol_msr)
```

```
                                {
                                        arg_stp->taw11_cr=t+0.5*grd.dt_m;
                                        break;
                                }
                                t+=grd.dt_m;
                        }
                        if (arg_stp->taw11_cr==arg_sp.taw_cr_inf)
                        {
                                for (i=2*grd.i_m;i<grd.i_m+grd.i_e;i++)
                                {
                                        if (d1[i]*d1[i+1]<0 && fabs(d1[i])>arg_sp.tol_msr && fabs(
    d1[i+1])>arg_sp.tol_msr)
                                        {
                                                arg_stp->taw11_cr=t+0.5*grd.dt_e;
                                                break;
                                        }
                                        t+=grd.dt_e;
                                }
                        }
                }
                else
                {
                        t=arg_sb->taw0+2*grd.dt_e;
                        for (i=2;i<grd.i_e-grd.i_m;i++)
                        {
                                if (d0[i]*d0[i+1]<0 && fabs(d0[i])>arg_sp.tol_msr && fabs(d0[i+1])
    >arg_sp.tol_msr)
                                {
                                        arg_stp->taw10_cr=t+0.5*grd.dt_e;
                                        break;
                                }
                                t+=grd.dt_e;
                        }
                        if (arg_stp->taw10_cr==arg_sp.taw_cr_inf)
                        {
                                for (i=grd.i_e-grd.i_m;i<grd.i_m+grd.i_e;i++)
                                {
                                        if (d0[i]*d0[i+1]<0 && fabs(d0[i])>arg_sp.tol_msr && fabs(
    d0[i+1])>arg_sp.tol_msr)
                                        {
                                                arg_stp->taw10_cr=t+0.5*grd.dt_m;
                                                break;
                                        }
                                        t+=grd.dt_m;
                                }
                        }
                        t=arg_sb->taw0+2*grd.dt_e;
                        for (i=2;i<grd.i_e-grd.i_m;i++)
                        {
                                if (d1[i]*d1[i+1]<0 && fabs(d1[i])>arg_sp.tol_msr && fabs(d1[i+1])
    >arg_sp.tol_msr)
                                {
                                        arg_stp->taw11_cr=t+0.5*grd.dt_e;
                                        break;
                                }
                                t+=grd.dt_e;
                        }
                        if (arg_stp->taw11_cr==arg_sp.taw_cr_inf)
                        {
                                for (i=grd.i_e-grd.i_m;i<grd.i_m+grd.i_e;i++)
                                {
                                        if (d1[i]*d1[i+1]<0 && fabs(d1[i])>arg_sp.tol_msr && fabs(
    d1[i+1])>arg_sp.tol_msr)
                                        {
                                                arg_stp->taw11_cr=t+0.5*grd.dt_m;
                                                break;
                                        }
                                        t+=grd.dt_m;
                                }
                        }
                }
```

```
    }

    //Stability ----------------------------------------------------------------------
    arg_stp->msr=0;
    if (arg_stp->taw10_cr!=arg_sp.taw_cr_inf || arg_stp->taw11_cr!=arg_sp.taw_cr_inf)
    {
            if (arg_stp->taw10_cr<arg_stp->taw11_cr)
                    arg_stp->st_id=0;
            else
                    arg_stp->st_id=1;
            arg_stp->msr=1;
    }
    num0[0][0]=sgn0*w[0][grd.i_0];
    num0[0][1]=w[1][grd.i_0];
    num0[0][2]=w[2][grd.i_0];
    num0[1][0]=ddw[0][1];
    num0[1][1]=ddw[1][1];
    num0[1][2]=ddw[2][1];
    num0[2][0]=iw[0];
    num0[2][1]=iw[1];
    num0[2][2]=iw[2];

    den0[0][0]=sgn0*w[0][grd.i_0];
    den0[0][1]=w[1][grd.i_0];
    den0[0][2]=w[2][grd.i_0];
    den0[1][0]=sgn1*w[0][grd.i_1];
    den0[1][1]=w[1][grd.i_1];
    den0[1][2]=w[2][grd.i_1];
    den0[2][0]=iw[0];
    den0[2][1]=iw[1];
    den0[2][2]=iw[2];

    num1[0][0]=sgn0*w[3][grd.i_0];
    num1[0][1]=w[4][grd.i_0];
    num1[1][0]=ddw[3][1];
    num1[1][1]=ddw[4][1];

    den1[0][0]=sgn0*w[3][grd.i_0];
    den1[0][1]=w[4][grd.i_0];
    den1[1][0]=sgn1*w[3][grd.i_1];
    den1[1][1]=w[4][grd.i_1];

    temp0=-Det3(num0)/Det3(den0);
    temp1=-Det2(num1)/Det2(den1);
    arg_stp->chi0=temp0;
    arg_stp->chi1=temp1;

    if (arg_sb->q<0)
            arg_stp->chi=-Sign(arg_sb->q)*(cos(arg_sp.th_cp)/sin(arg_sp.th_cp))*(pow(arg_sb->a
,2)+arg_sb->a*cos(arg_sb->taw1))
                                    /(1+pow(arg_sb->a,2)+2*arg_sb->a*cos(arg_sb->taw1));
    else
            arg_stp->chi=-Sign(arg_sb->q)*(cos(arg_sp.th_cp)/sin(arg_sp.th_cp))*(pow(arg_sb->a
+2,2)-(arg_sb->a+2)*cos(arg_sb->taw1))
                                    /(1+pow(arg_sb->a+2,2)-2*(arg_sb->a+2)*cos(arg_sb->taw1));

    if (arg_stp->msr==0)
    {
            if (arg_stp->chi>arg_stp->chi0 && arg_stp->chi>arg_stp->chi1)
            {
                    arg_stp->st_id=2;
            }
            else
            {
                    if (arg_stp->chi0>arg_stp->chi1)
                            arg_stp->st_id=0;
                    else
                            arg_stp->st_id=1;
            }
    }
    //-------------------------------------------
```

```
        for (i=2;i>=0;i--)
                delete [] *(den0+i);
        delete [] den0;
        for (i=2;i>=0;i--)
                delete [] *(num0+i);
        delete [] num0;
        for (i=1;i>=0;i--)
                delete [] *(den1+i);
        delete [] den1;
        for (i=1;i>=0;i--)
                delete [] *(num1+i);
        delete [] num1;
        //-------------------------------------------
        for (i=4;i>=0;--i)
                delete [] *(ddw+i);
        for (i=4;i>=0;--i)
                delete [] *(w+i);
        for (i=2;i>=0;--i)
                delete [] *(iw_t+i);
        delete [] ddw;
        delete [] w;
        delete [] iw_t;
        delete [] rho;
        delete [] d0;
        delete [] d1;
}

void AnalyzeStability(SolutionP &arg_sp,SolutionB **arg_sb,SolutionB ***arg_sbcr,StabilityP **
     arg_stp)
{
        int i,j,im0,im1,im2,im3,jm0,jm1,jm2,jm3,jcr,id0,id1;

        arg_sp.n_cr=new int [arg_sp.n_br];
        for (i=0;i<arg_sp.n_br;i++)
        {
                for (j=0;j<arg_sp.n_fpa[i];j++)
                        CriticalChi(arg_sp,*(arg_sb+i)+j,*(arg_stp+i)+j);

                arg_sp.n_cr[i]=0;
                id0=arg_stp[i][0].st_id;
                for (j=1;j<arg_sp.n_fpa[i];j++)
                {
                        id1=arg_stp[i][j].st_id;
                        if (id1!=id0 && id0+id1>1)
                        {
                                arg_sp.n_cr[i]+=1;
                                id0=id1;
                        }
                }
        }

        *arg_sbcr=new SolutionB *[arg_sp.n_br+1]; // the last one is used to store outer boundary
     of the branching diagram
        for (i=0;i<arg_sp.n_br;i++)
                *(*arg_sbcr+i)=new SolutionB [arg_sp.n_cr[i]];
        *(*arg_sbcr+arg_sp.n_br)=new SolutionB [4]; // component storing the followings: 0-Vmax 1-
     Vmin 2-qmax 3-qmin
        //branch points:
        for (i=0;i<arg_sp.n_br;i++)
        {
                jcr=0;
                id0=arg_stp[i][0].st_id;
                for (j=1;j<arg_sp.n_fpa[i];j++)
                {
                        id1=arg_stp[i][j].st_id;
                        if (id1!=id0 && id0+id1>1)
                        {
                                (*(*arg_sbcr+i)+jcr)->q=(arg_sb[i][j-1].q+arg_sb[i][j].q)/2;
                                (*(*arg_sbcr+i)+jcr)->taw0=(arg_sb[i][j-1].taw0+arg_sb[i][j].taw0)
     /2;
```

```
                    (*(*arg_sbcr+i)+jcr)->taw1=(arg_sb[i][j-1].taw1+arg_sb[i][j].taw1)
/2;
                    (*(*arg_sbcr+i)+jcr)->a=(arg_sb[i][j-1].a+arg_sb[i][j].a)/2;
                    (*(*arg_sbcr+i)+jcr)->th_m=(arg_sb[i][j-1].th_m+arg_sb[i][j].th_m)
/2;
                    (*(*arg_sbcr+i)+jcr)->v=(arg_sb[i][j-1].v+arg_sb[i][j].v)/2;
                    id0=id1;
                    jcr++;
                }
            }
    }
    //outer boundary points:
    SolutionB **sbcr=*arg_sbcr;
    sbcr[arg_sp.n_br][0].v=1e10;    //v_min
    sbcr[arg_sp.n_br][1].v=-1e10;   //v_max
    sbcr[arg_sp.n_br][2].q=-1e10;   //q_max
    sbcr[arg_sp.n_br][3].q=1e10;    //q_min
    im2=im3=jm2=jm3=0;
    for (i=0;i<arg_sp.n_br;i++)
    {
            for (j=1;j<arg_sp.n_fpa[i]-1;j++)
            {
                    if (arg_sb[i][j].v<sbcr[arg_sp.n_br][0].v)
                    {
                            sbcr[arg_sp.n_br][0].v=arg_sb[i][j].v;
                            im0=i;
                            jm0=j;
                    }
                    if (arg_sb[i][j].v>sbcr[arg_sp.n_br][1].v)
                    {
                            sbcr[arg_sp.n_br][1].v=arg_sb[i][j].v;
                            im1=i;
                            jm1=j;
                    }
                    /*if (arg_sb[i][j].q>sbcr[arg_sp.n_br][2].q && arg_sb[i][j].v<arg_sp.
vup_pmax && arg_sb[i][j].v>arg_sp.vlow_pmax && arg_stp[i][j].st_id==2)
                    {
                            sbcr[arg_sp.n_br][2].q=arg_sb[i][j].q;
                            im2=i;
                            jm2=j;
                    }
                    if (arg_sb[i][j].q<sbcr[arg_sp.n_br][3].q && arg_sb[i][j].v<arg_sp.
vup_pmax && arg_sb[i][j].v>arg_sp.vlow_pmax && arg_stp[i][j].st_id==2)
                    {
                            sbcr[arg_sp.n_br][3].q=arg_sb[i][j].q;
                            im3=i;
                            jm3=j;
                    }*/
                    if (arg_sb[i][j].q>arg_sb[i][j+1].q && arg_sb[i][j].q>arg_sb[i][j-1].q &&
arg_sb[i][j].v<arg_sp.vup_pmax && arg_sb[i][j].v>arg_sp.vlow_pmax && arg_stp[i][j].st_id==2)
                    {
                            sbcr[arg_sp.n_br][2].q=arg_sb[i][j].q;
                            im2=i;
                            jm2=j;
                    }
                    if (arg_sb[i][j].q<arg_sb[i][j+1].q && arg_sb[i][j].q<arg_sb[i][j-1].q &&
arg_sb[i][j].v<arg_sp.vup_pmax && arg_sb[i][j].v>arg_sp.vlow_pmax && arg_stp[i][j].st_id==2)
                    {
                            sbcr[arg_sp.n_br][3].q=arg_sb[i][j].q;
                            im3=i;
                            jm3=j;
                    }
            }
    }
    sbcr[arg_sp.n_br][0].q=arg_sb[im0][jm0].q;
    sbcr[arg_sp.n_br][0].taw0=arg_sb[im0][jm0].taw0;
    sbcr[arg_sp.n_br][0].taw1=arg_sb[im0][jm0].taw1;
    sbcr[arg_sp.n_br][0].a=arg_sb[im0][jm0].a;
    sbcr[arg_sp.n_br][0].th_m=arg_sb[im0][jm0].th_m;

    sbcr[arg_sp.n_br][1].q=arg_sb[im1][jm1].q;
```

```
        sbcr[arg_sp.n_br][1].taw0=arg_sb[im1][jm1].taw0;
        sbcr[arg_sp.n_br][1].taw1=arg_sb[im1][jm1].taw1;
        sbcr[arg_sp.n_br][1].a=arg_sb[im1][jm1].a;
        sbcr[arg_sp.n_br][1].th_m=arg_sb[im1][jm1].th_m;

        sbcr[arg_sp.n_br][2].v=arg_sb[im2][jm2].v;
        sbcr[arg_sp.n_br][2].taw0=arg_sb[im2][jm2].taw0;
        sbcr[arg_sp.n_br][2].taw1=arg_sb[im2][jm2].taw1;
        sbcr[arg_sp.n_br][2].a=arg_sb[im2][jm2].a;
        sbcr[arg_sp.n_br][2].th_m=arg_sb[im2][jm2].th_m;

        sbcr[arg_sp.n_br][3].v=arg_sb[im3][jm3].v;
        sbcr[arg_sp.n_br][3].taw0=arg_sb[im3][jm3].taw0;
        sbcr[arg_sp.n_br][3].taw1=arg_sb[im3][jm3].taw1;
        sbcr[arg_sp.n_br][3].a=arg_sb[im3][jm3].a;
        sbcr[arg_sp.n_br][3].th_m=arg_sb[im3][jm3].th_m;
}

void PrintResult(SolutionP &arg_sp,SolutionB **arg_sb,SolutionB **arg_sbcr,StabilityP **arg_stp)
{
        int i,j,jj,k;
        ostringstream ostrIndex,ostrP0,ostrP1;
        string strIndex,strFileName,strA,strCommand,strSpec;
        Engine *eng;
        mxArray **mxA=NULL;
        int is_stable,is_stable1;
        int *is_stable_curve;
        int n_curve,*l_curve,i_curve,i_curve_t;
        double *dblA;

        #ifdef _STANDALONE
                for (i=0;i<arg_sp.n_br;i++)
                {
                        ostrIndex.str("");
                        ostrIndex<<(i+1);
                        strIndex=ostrIndex.str();
                        strFileName=arg_sp.strDataPath+"\\stp-"+strIndex+".plt";
                        fp=fopen(strFileName.c_str(),"w");
                                fprintf(fp,"TITLE=Plots\n");
                                fprintf(fp,"VARIABLES=j,th_m,v,km,id,taw10_cr,taw11_cr,taw1\n");
                                for (j=0;j<arg_sp.n_fpa[i];j++)
                                        fprintf(fp,"%d\t%f\t%f\t%f\t%d\t%f\t%f\t%f\n",j,arg_sb[i][
    j].th_m,arg_sb[i][j].v,arg_sb[i][j].q,arg_stp[i][j].st_id,arg_stp[i][j].taw10_cr,arg_stp[i][j
    ].taw11_cr,arg_sb[i][j].taw1);
                        fclose(fp);
                }
        #endif
        jj=0;
        for (i=0;i<arg_sp.n_br;i++)
        {
                for (j=0;j<arg_sp.n_cr[i];j++)
                {
                        ostrIndex.str("");
                        ostrIndex<<jj+1;
                        strIndex=ostrIndex.str();
                        strFileName=arg_sp.strDataPath+"\\branchpoint-"+strIndex+".txt";
                        fp=fopen(strFileName.c_str(),"a+");
                                fprintf(fp,"%f\t",arg_sp.Lambda);
                                fprintf(fp,"%f\t",arg_sbcr[i][j].v);
                                fprintf(fp,"%f\t",arg_sbcr[i][j].q);
                                fprintf(fp,"%f\t",arg_sbcr[i][j].taw0);
                                fprintf(fp,"%f\t",arg_sbcr[i][j].taw1);
                                fprintf(fp,"%f\t",arg_sbcr[i][j].a);
                                fprintf(fp,"%f\n",arg_sbcr[i][j].th_m);
                        fclose(fp);
                        jj++;
                }
        }
        for (i=0;i<4;i++)
        {
                ostrIndex.str("");
```

```
        ostrIndex<<i+1;
        strIndex=ostrIndex.str();
        strFileName=arg_sp.strDataPath+"\\boundarypoint-"+strIndex+".txt";
        fp=fopen(strFileName.c_str(),"a+");
                fprintf(fp,"%f\t",arg_sp.Lambda);
                fprintf(fp,"%f\t",arg_sbcr[arg_sp.n_br][i].v);
                fprintf(fp,"%f\t",arg_sbcr[arg_sp.n_br][i].q);
                fprintf(fp,"%f\t",arg_sbcr[arg_sp.n_br][i].taw0);
                fprintf(fp,"%f\t",arg_sbcr[arg_sp.n_br][i].taw1);
                fprintf(fp,"%f\t",arg_sbcr[arg_sp.n_br][i].a);
                fprintf(fp,"%f\n",arg_sbcr[arg_sp.n_br][i].th_m);
        fclose(fp);
}
//Running Matlab engine to plot
-----------------------------------------------------------------------
n_curve=25; //number of curves per half-branch
l_curve=new int [n_curve];
mxA=new mxArray *[n_curve*arg_sp.n_br];
is_stable_curve=new int [n_curve*arg_sp.n_br];

if (!(eng = engOpen("\0")))
{
        fprintf(stderr, "\nCan't start MATLAB engine\n");
        return;
}

//Data transfer
i_curve_t=0;
for (i=0;i<arg_sp.n_br;i++)
{
        for (j=0;j<n_curve;j++)
                l_curve[j]=0;
        i_curve=0;
        for (j=1;j<arg_sp.n_fpa[i]-1;j++)
        {
                l_curve[i_curve]++;
                is_stable=arg_stp[i][j].st_id;
                is_stable1=arg_stp[i][j+1].st_id;
                if (is_stable!=is_stable1)
                        i_curve++;
        }
        l_curve[0]++;
        l_curve[i_curve]++;
        if (arg_sp.n_fpa[i]==1)
                l_curve[0]=1;
        i_curve++;
        jj=0;
        for (j=0;j<i_curve;j++)
        {
                ostrIndex.str("");
                ostrIndex<<i_curve_t;
                strIndex=ostrIndex.str();
                strA="A"+strIndex;
                mxA[i_curve_t]=mxCreateDoubleMatrix(l_curve[j],3,mxREAL);
                dblA=mxGetPr(mxA[i_curve_t]);
                for (k=0;k<l_curve[j];k++)
                {
                        dblA[k]=arg_sb[i][jj].v;        //For cylinder->[].v->[].taw1
                        dblA[k+l_curve[j]]=arg_sb[i][jj].q;
                        dblA[k+2*l_curve[j]]=arg_sb[i][jj].th_m*180/pi;
                        jj++;
                }
                engPutVariable(eng,strA.c_str(),mxA[i_curve_t]);
                is_stable_curve[i_curve_t]=arg_stp[i][jj-1].st_id;
                i_curve_t++;
        }
}
#ifdef _SPHEREBRANCH
        mxArray *mxB,*mxC;
        double *dblB,*dblC;
        double dth,th,th_i,q_max;
```

```
            int n_inv_BC;

            n_inv_BC=200; //number of interval for plotting curves of B, C matrices (sphere
    branch & vertical line at v_t)
            q_max=arg_sbcr[arg_sp.n_br][2].q;
            mxB=mxCreateDoubleMatrix(n_inv_BC+1,2,mxREAL);
            mxC=mxCreateDoubleMatrix(n_inv_BC+1,2,mxREAL);
            dblB=mxGetPr(mxB);
            dblC=mxGetPr(mxC);
            dth=(q_max+2)/n_inv_BC;
            th=-2;
            for (i=0;i<=n_inv_BC;i++)
            {
                    dblC[i]=arg_sp.v_t;
                    dblC[i+n_inv_BC+1]=th;
                    th+=dth;
            }
            th_i=0.4;
            dth=(pi-th_i)/n_inv_BC;
            th=th_i;
            for (i=0;i<=n_inv_BC;i++)
            {
                    dblB[i]=(0.5-0.25*pow(cos(th),3)+0.75*cos(th))/pow(sin(th),3);
                    dblB[i+n_inv_BC+1]=-2*sin(th);
                    th+=dth;
            }
            engPutVariable(eng,"B",mxB);
            engPutVariable(eng,"C",mxC);
    #endif _SPHEREBRANCH
    //plot formatting
    engEvalString(eng,"set(0,'defaultaxesfontsize',18);");
    engEvalString(eng,"set(0,'defaulttextfontsize',18);");
    engEvalString(eng,"set(0,'defaultaxeslinewidth',2);");
    engEvalString(eng,"set(0,'defaultlinelinewidth',2);");
    for (i=0;i<i_curve_t;i++)
    {
            ostrIndex.str("");
            ostrIndex<<i;
            strIndex=ostrIndex.str();
            strA="A"+strIndex;
            if (is_stable_curve[i]==2)
                    strSpec=",'-k'";
            if (is_stable_curve[i]==1)
                    strSpec=",'--b'";
            if (is_stable_curve[i]==0)
                    strSpec=",':r'";
            //strCommand="plot3("+strA+"(:,1),"+strA+"(:,2),"+strA+"(:,3)"+strSpec+");";
            strCommand="plot("+strA+"(:,1),"+strA+"(:,2)"+strSpec+");";
            engEvalString(eng,strCommand.c_str());
            engEvalString(eng,"hold on");
    }
    #ifdef _SPHEREBRANCH
            strSpec=",'-k'";
            strCommand="plot(B(:,1),B(:,2)"+strSpec+");";
            engEvalString(eng,strCommand.c_str());                                   //
    For cylinder->deactivate
            engEvalString(eng,"hold on");
            strSpec=",'-g','LineWidth',0.5";
            strCommand="plot(C(:,1),C(:,2)"+strSpec+");";                        //For
    cylinder->deactivate
            engEvalString(eng,strCommand.c_str());
    #endif
    engEvalString(eng,"xlabel('$V / (4 \\pi R_0^3 / 3)$','FontSize',18,'Interpreter','latex')
    ;");          //For cylinder->deactivate
    //engEvalString(eng,"xlabel('$\\Lambda$','FontSize',18,'Interpreter','latex');");
                                //For cylinder-> activate
    engEvalString(eng,"ylabel('$\\Delta p R_0 / \\gamma$','FontSize',18,'Interpreter','latex')
    ;");
    engEvalString(eng,"set(gca,'FontSize',18,'FontName','Times');");
    #ifdef _AXISADJUSTMENT
            ostringstream ostrXmax;
```

```
                string strXmax;

                ostrXmax.str("");
                ostrXmax<<min((int)v_max+1,10);
                strXmax=ostrXmax.str();
                strCommand="xlim([0 "+strXmax+"]);";            //For cylinder->deactivate
                //strCommand="xlim([0 10]);";               //For cylinder-> activate
                engEvalString(eng,strCommand.c_str());
        #endif
        //temp adjustment -----------------------
        strCommand="xlim([80 120]);";
        engEvalString(eng,strCommand.c_str());
        //-------------------------------------

        ostrP0.str("");
        ostrP1.str("");
        ostrP0<< setiosflags(ios::fixed) << setprecision(2) << arg_sp.Lambda;
        ostrP1<< setiosflags(ios::fixed) << setprecision(0) << arg_sp.th_cp*180/pi;
        strFileName=arg_sp.strDataPath+"\\BS_SN_"+ostrP0.str()+"_CA_"+ostrP1.str()+".eps";
        strCommand="print('-depsc','"+strFileName+"');";
        engEvalString(eng,strCommand.c_str());
        engEvalString(eng,"set(gcf, 'PaperUnits', 'centimeters');");
        engEvalString(eng,"set(gcf, 'PaperSize', [20 15]);");
        engEvalString(eng,"set(gcf, 'PaperPosition', [0 0 20 15]);");
        strFileName=arg_sp.strDataPath+"\\BS_SN_"+ostrP0.str()+"_CA_"+ostrP1.str()+".pdf";
        strCommand="print('-dpdf','"+strFileName+"');";
        engEvalString(eng,strCommand.c_str());
        //engEvalString(eng, "zlabel('y_2','FontName','Times New Roman');");

        system ("PAUSE");
        for (i=i_curve_t-1;i>=0;i--)
                mxDestroyArray(mxA[i]);
        engEvalString(eng, "close;");
        engClose(eng);

        delete [] is_stable_curve;
        delete [] l_curve;
        delete [] mxA;

}

int main(int argc, char **argv)
{
        SolutionP sp;
        SolutionB **sb,**sb_cr;
        StabilityP **stp;

        GetParameters(sp,argv);
        ReadSolutionParameter(sp);
        MAllocation(sp,&sb,&stp);
        ReadSolutionBranch(sp,sb);
        AnalyzeStability(sp,sb,&sb_cr,stp);
        PrintResult(sp,sb,sb_cr,stp);
        MDelete(sp,&sb,&sb_cr,&stp);

        return 0;
}
```

Appendix I

Image processing code I

```
function read_bridge()
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%The origin of the Initial Coordinate System measured in pixels is located at top-left
%corner. Its second coordinate increases toward the bottom of image. The origin of
%the Global Coordinate System measured in scaled lengths (with R0) is located on the
%baseline at the center with the second coordinate increasing upward.
%Note: distances between two pixels are measured from pixel centers in this program [e.g., dist(i
    =4,i=2)=(4-2)*pixelsize].
%********************************************************************************
%edg_l[r]: Array to store left[right] edge pixels (locus of maximum intensity gradient)
%edg_n: Array to store needle tipe pixels
%edg_x1: x1-coordinate of edge (left or right is chosen by the program) in Global coordinate to be
     used for fitting
%edg_x2: x2-coordinate of edge (left or right is chosen by the program) in Global coordinate to be
     used for fitting
%mns_x1: x1-coordinate of left[right] the fitted meniscus in Global coordinate
%mns_x2: x2-coordinate of left[right] the fitted meniscus in Global coordinate
%mns_l[r]: Array to store left[right] fitted meniscus pixels
%j_s: Column index from which edge detection starts
%j_m: Column index at which the image is splitted into two halfs (roughly at the image middle,
    blonging to the left half)
%j_e: Column index at which edge detection ends
%i_tip_s: Row index from which needle-tip detection starts
%i_tip_e: Row index at which needle-tip detection ends
%i_base_s: Row index from which baseline detection starts
%i_base_e: Row index at which baseline detection ends
%i_tip: Row index of the needle tip
%i_base: Row index of the baseline on the coverslip. Set it to zero for the program to determine
    it. Set it to desired value if it is reliably calculated from another image in the image set.
%i_tl: Row index of top-left corner of the bridge
%i_bl: Row index of bottom-left corner of the bridge
%i_tr: Row index of top-right corner of the bridge
%i_br: Row index of bottom-right corner of the bridge
%j_tl: Column index of top-left corner of the bridge
%j_bl: Column index of bottom-left corner of the bridge
%j_tr: Column index of top-right corner of the bridge
%j_br: Column index of bottom-right corner of the bridge
%hydrophobicity: To be given to the program to help detect the baseline (1-> hydrophobic, -1->
    hydrophilic)
%n_opt_smp: Number pixels to be left be left between sampled pixels used for curve fitting.
%n_row_b: Number of rows of pixels lying between the two baselines
%n_row_needle: Number of rows from the top to measure the needle stem radius
%n_col_tip: Number of columns about the symmetry axis to measure the needle-tip radius
%axis_col: column of the symmetry axis in pixels
%jump_off_tol: Tolerance of jump in pixel along edge to identify off-data points
%tilt_tol: Tolerance in pixels of the difference between j_bl and j_br. If not met, the program
    issues a warning.
%n_off_max: Maximum number of off-data points on each edge, used for memory allocation
```

```
%R00m: Needle stem radius in millimetre
%R00: Needle stem radius in pixels
%R0: Needle tip radius in pixels
%R0m: Needle tip in millimeter
%R1: Contact-line radius in pixels
%R1m: Contact-line radius in millimeter
%RMS: Root mean square of curve fitting
%ar: Aspect ratio (slenderness)
%v: Bridge volume in m^3
%v_: Scaled volume, v_=v/(4 pi R0m^3/3)
%v_cyl: Cylindrical volume, v_cyl=v/(pi R0m^2 h)
%edg_var: Variance measuring the discrepancy between the distance of left and right edges from the
      symmetry axis. This measure non-axial symmetry of the image.
%do_opt: Optimizing switch (0-> it is off, 1-> it is on)
%side_opt: Side the program chooses for curve fitting (1-> left, 2->right). Set it to zero for the
      program to determine. Set it to desired value if it known
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
global n_row_b n_opt_smp ar RMS n_int

clc;
format('long');
strImgPath='/Users/amirakbari/Documents/Amir/PhD - McGill/PhD project/Experiment/Matlab/image
     processing (Contact Drop Dispensing)';
strImgXname='29';
strImgXext='.bmp';
strImgPathX=strcat(strImgPath,'/image sources/',strImgXname,strImgXext);
strImgPathXedg=strcat(strImgPath,'/image edge/',strImgXname,'-edg',strImgXext);
strImgPathXfit=strcat(strImgPath,'/image fitting/',strImgXname,'-fit',strImgXext);

%II=imread(strImgPathX);
%I=rgb2gray(imread(strImgPathX));
I=Gaussian(strImgPathX,[5 5],0.95,'xxx','rgb'); %last argument 'gray' or 'rgb'
[n_row n_col]=size(I);
Iedg=zeros(n_row,n_col,3,'uint8');
Ifit=zeros(n_row,n_col,3,'uint8');
edg_l=zeros(n_row,1,'uint16');
edg_r=zeros(n_row,1,'uint16');
djdi_l=zeros(n_row,1,'double');
djdi_r=zeros(n_row,1,'double');

do_opt=1;
y0=[-1.0264     -1.9651 2.2564  0.8325];
n_int=500; %number of intervals for numerical integration
n_rk=1000; %number of point for Runge-Kutta solution of dzds=f(s) (for drawing fitted edge)
hydrophobicity=1;
j_s=5;
j_m=round(n_col/2);
j_e=n_col-5;
i_tip_s=500;
i_tip_e=630;
i_base_s=950;
i_base_e=n_row-5;
i_base=0;
side_opt=1;
n_opt_smp=5;
n_col_tip=80;
n_row_needle=200;
jump_off_tol=10;    %pixel
tilt_tol=5;         %pixel
n_off_max=20;
R00m=2.07/2;          %mm

iedg_off_l=zeros(n_off_max,2,'uint16');
iedg_off_r=zeros(n_off_max,2,'uint16');
tip=zeros(2*n_col_tip+1,1,'uint16');

% Identifying right and left edges
% 1- predictor:
j_ignore=-1;
for i=1:n_row
    %left
```

```
    if i<3
        djdi_l(i)=0;
    else
        djdi_l(i)=double(edg_l(i-1))-double(edg_l(i-2));
    end
    edg_l(i)=Fun_edgl(I,i,djdi_l(i),n_row,j_s,j_m,j_ignore);
    %right
    if i<3
        djdi_r(i)=0;
    else
        djdi_r(i)=double(edg_r(i-1))-double(edg_r(i-2));
    end
    edg_r(i)=Fun_edgr(I,i,djdi_r(i),n_row,j_e,j_m,j_ignore);
end
%2- refining:
counter_l=0;
edg_test=edg_l(1);
i_test=1; %index of the last correct pixel
for i=2:n_row
    if abs(double(edg_l(i))-double(edg_test))>jump_off_tol
        counter_l=counter_l+1;
        iedg_off_l(counter_l,1)=i;
        iedg_off_l(counter_l,2)=i_test;
        continue;
    end
    i_test=i;
    edg_test=edg_l(i);
end
counter_r=0;
edg_test=edg_r(1);
i_test=1;
for i=2:n_row
    if abs(double(edg_r(i))-double(edg_test))>jump_off_tol
        counter_r=counter_r+1;
        iedg_off_r(counter_r,1)=i;
        iedg_off_r(counter_r,2)=i_test;
        continue;
    end
    i_test=i;
    edg_test=edg_r(i);
end
% extrapolating (linear) the last edge pixel from the last correct pixel:
% this part assumes the darivative at the last correct pixel 'djdi' is +-1
if iedg_off_l(counter_l,1)==n_row
    edg_l(n_row)=edg_l(iedg_off_l(counter_l,2))-hydrophobicity*double(n_row-iedg_off_l(counter_l
     ,2));
    iedg_off_l(counter_l,:)=0;
    counter_l=counter_l-1;
end
if iedg_off_r(counter_r,1)==n_row
    edg_r(n_row)=edg_r(iedg_off_r(counter_r,2))+hydrophobicity*double(n_row-iedg_off_r(counter_r
     ,2));
    iedg_off_r(counter_r,:)=0;
    counter_r=counter_r-1;
end
%3- spline interpolation:
edg_spline_l=zeros(n_row-counter_l,2,'double');
edg_spline_r=zeros(n_row-counter_r,2,'double');

i_l=1;
i_r=1;
for i=1:n_row
    is_accepted=1;
    for j=1:counter_l
        if i==iedg_off_l(j,1)
            is_accepted=0;
            break;
        end
    end
    if is_accepted
        edg_spline_l(i_l,1)=i;
```

```
        edg_spline_l(i_l,2)=edg_l(i);
        i_l=i_l+1;
    end
    is_accepted=1;
    for j=1:counter_r
        if i==iedg_off_r(j,1)
            is_accepted=0;
            break;
        end
    end
    if is_accepted
        edg_spline_r(i_r,1)=i;
        edg_spline_r(i_r,2)=edg_r(i);
        i_r=i_r+1;
    end
end

ii=1:1:n_row;
edg_l(:)=round(spline(edg_spline_l(:,1),edg_spline_l(:,2),ii(:)));
edg_r(:)=round(spline(edg_spline_r(:,1),edg_spline_r(:,2),ii(:)));

% Symmetry axis and needle stem radius
axis_col=0;
for i=1:n_row_needle
    axis_col=axis_col+double(edg_l(i)+edg_r(i));
end
axis_col=axis_col/2/n_row_needle;

R00=0;
for i=1:n_row_needle
    R00=R00+double(edg_r(i)-edg_l(i));
end
R00=R00/n_row_needle/2;

% Identifying needle tip
gradI=zeros(i_tip_e-i_tip_s+1,1,'double');
jj=1;
for j=round(axis_col)-n_col_tip:round(axis_col)+n_col_tip
    for i=i_tip_s:i_tip_e
        Iffff=double(I(i+4,j));
        Ifff=double(I(i+3,j));
        Iff=double(I(i+2,j));
        If=double(I(i+1,j));
        Ib=double(I(i-1,j));
        Ibb=double(I(i-2,j));
        Ibbb=double(I(i-3,j));
        Ibbbb=double(I(i-4,j));
        gradI(i-i_tip_s+1)=abs(-3*Iffff+32*Ifff-168*Iff+672*If-672*Ib+168*Ibb-32*Ibbb+3*Ibbbb)
     /840;
    end
    [g_max,i_max]=max(gradI,[],1);
    tip(jj)=i_max+i_tip_s-1;
    jj=jj+1;
end
i_tip=0;
for j=1:2*n_col_tip+1
    i_tip=i_tip+double(tip(j));
end
i_tip=round(i_tip/(2*n_col_tip+1));
%corners:
j_tl=edg_l(i_tip);
j_tr=edg_r(i_tip);
i_tl=i_tip;
i_tr=i_tip;

% Identifying the baseline
if i_base==0
    edg=zeros(i_base_e-i_base_s+1,1,'uint16');
    for i=i_base_s:i_base_e
        edg(i-i_base_s+1)=edg_l(i);
    end
```

```
    if hydrophobicity==1
        [g_m,i_ml]=max(edg,[],1);
    end
    if hydrophobicity==-1
        [g_m,i_ml]=min(edg,[],1);
    end
    n_bl=0;
    for i=1:i_base_e-i_base_s+1
        if edg(i)==g_m
            n_bl=n_bl+1;
        end
    end

    for i=i_base_s:i_base_e
        edg(i-i_base_s+1)=edg_r(i);
    end
    if hydrophobicity==1
        [g_m,i_mr]=min(edg,[],1);
    end
    if hydrophobicity==-1
        [g_m,i_mr]=max(edg,[],1);
    end
    n_br=0;
    for i=1:i_base_e-i_base_s+1
        if edg(i)==g_m
            n_br=n_br+1;
        end
    end
    %corners
    i_bl=i_ml+round(n_bl/2)+i_base_s-1;
    j_bl=edg_l(i_bl);
    i_br=i_mr+round(n_br/2)+i_base_s-1;
    j_br=edg_r(i_br);
    if abs(double(i_bl)-double(i_br))>tilt_tol
        display('The baseline is tilted beyond the tolerable limit.')
    end
    if side_opt==0
        if n_bl<n_br
            i_base=i_bl;
            side_opt=1;
        else
            i_base=i_br;
            side_opt=2;
        end
    else
        if side_opt==1
            i_base=i_bl;
        end
        if side_opt==2
            i_base=i_br;
        end
    end
end
n_row_b=i_base-i_tip+1;

% Non-axisymmetry variance and needle-tip radius
axis_col=double(j_tr+j_tl)/2; %correction
R0=double(j_tr-j_tl)/2;
if side_opt==1
    R1=axis_col-double(j_bl);
end
if side_opt==2
    R1=double(j_br)-axis_col;
end
R0m=R0/R00*R00m;
edg_var=0;
for i=i_tip:i_base
    r_r=double(edg_r(i))-axis_col;
    r_l=axis_col-double(edg_l(i));
    edg_var=edg_var+(r_r-r_l)^2;
end
```

```
edg_var=sqrt(edg_var/n_row_b);
ar=double(i_base-i_tip)/R0;

% Setting up coordinates
edg_x1=zeros(n_row_b,1,'double');
edg_x2=zeros(n_row_b,1,'double');

if side_opt==1
    for i=1:n_row_b
        edg_x1(i)=(axis_col-double(edg_l(i_base-i+1)))/R0;
        edg_x2(i)=(double(i)-1)/R0;
    end
end
if side_opt==2
    for i=1:n_row_b
        edg_x1(i)=(double(edg_r(i_base-i+1))-axis_col)/R0;
        edg_x2(i)=(double(i)-1)/R0;
    end
end

% Fitting theoritical meniscus profile to experimental data
mns_x1=zeros(n_row_b,1,'double');
mns_x2=zeros(n_row_b,1,'double');
mns_l=zeros(n_row_b,1,'uint16');
mns_r=zeros(n_row_b,1,'uint16');

if (do_opt)
    y=fminsearch(@(x)Error(edg_x1,edg_x2,x), y0);
else
    y=y0;
end
if (y(1)<0)
    dKsidRho_0=-(1+y(4)*cos(y(2)))/sin(y(2))/y(4);
    dKsidRho_1=-(1+y(4)*cos(y(3)))/sin(y(3))/y(4);
else
    dKsidRho_0=(1-(y(4)+2)*cos(y(2)))/sin(y(2))/(y(4)+2);
    dKsidRho_1=(1-(y(4)+2)*cos(y(3)))/sin(y(3))/(y(4)+2);
end
th_m=atan(-abs(y(1))*dKsidRho_0/y(1))*180/pi;
th_cp=atan(abs(y(1))*dKsidRho_1/y(1))*180/pi;
v_=Vol(y);
v=v_*4*pi*R0m^3/3;
v_cyl=4*v_/ar/3;

z=zeros(n_rk,1,'double');
s=zeros(n_rk,1,'double');
dtaw=(y(2)-y(3))/(n_rk-1);
ds=dtaw/abs(y(1));

z(1)=0;
sx=y(3)/abs(y(1));
s(1)=sx;
for i=2:n_rk
    k1=sign(y(1))*(1+y(4)*cos(abs(y(1))*sx))./sqrt(1+y(4)^2+2*y(4)*cos(abs(y(1))*sx));
    sx=sx+ds/2;
    k2=sign(y(1))*(1+y(4)*cos(abs(y(1))*sx))./sqrt(1+y(4)^2+2*y(4)*cos(abs(y(1))*sx));
    k3=sign(y(1))*(1+y(4)*cos(abs(y(1))*sx))./sqrt(1+y(4)^2+2*y(4)*cos(abs(y(1))*sx));
    sx=sx+ds/2;
    k4=sign(y(1))*(1+y(4)*cos(abs(y(1))*sx))./sqrt(1+y(4)^2+2*y(4)*cos(abs(y(1))*sx));
    z(i)=z(i-1)+ds*(k1+2*k2+2*k3+k4)/6;
    s(i)=sx;
end
ss=spline(z,s,edg_x2);
ttaw=ss*abs(y(1));
rho=sqrt(1+y(4)^2+2*y(4)*cos(ttaw));
rr=rho/abs(y(1));

mns_x2(:)=edg_x2(:);
mns_x1(:)=rr(:);

for i=1:n_row_b
```

```
    mns_l(i)=round(axis_col-mns_x1(i)*R0);
    mns_r(i)=round(mns_x1(i)*R0+axis_col);
end

% Outputting results
YY=[edg_var;RMS;y(1);y(2);y(3);y(4);th_m;th_cp;ar;v;v_;v_cyl;R1/R0];
display(YY);


set(0,'defaultaxesfontsize',18);
set(0,'defaulttextfontsize',18);
set(0,'defaultaxeslinewidth',2);
set(0,'defaultlinelinewidth',2);
plot(mns_x1,mns_x2,'-b');
xlabel('$r/R_0$','FontSize',18,'Interpreter','latex');
ylabel('$z/R_0$','FontSize',18,'Interpreter','latex');
%xlim([-inf 1]);
%ylim([0 round(ar*10)/10]);
% set(gca,'FontSize',18,'FontName','Times');
% set(gcf, 'PaperUnits', 'centimeters');
% set(gcf, 'PaperSize', [20 15]);
% set(gcf, 'PaperPosition', [0 0 20 15]);
% print('-dpdf',strPlotPathX);

% Writing new image file indicating the edges
 Iedg(:,:,1)=I;
 Iedg(:,:,2)=I;
 Iedg(:,:,3)=I;
for i=1:n_row
    Iedg(i,edg_l(i),1)=0;
    Iedg(i,edg_l(i),2)=255;
    Iedg(i,edg_l(i),3)=0;
    Iedg(i,edg_r(i),1)=0;
    Iedg(i,edg_r(i),2)=255;
    Iedg(i,edg_r(i),3)=0;
    Iedg(i,round(axis_col),1)=0;
    Iedg(i,round(axis_col),2)=185;
    Iedg(i,round(axis_col),3)=241;
end
for j=j_tl:j_tr
    Iedg(i_tip,j,1)=0;
    Iedg(i_tip,j,2)=255;
    Iedg(i_tip,j,3)=0;
end
for j=1:n_col
    Iedg(i_base,j,1)=0;
    Iedg(i_base,j,2)=255;
    Iedg(i_base,j,3)=0;
end
imwrite(Iedg,strImgPathXedg);

 Ifit(:,:,1)=I;
 Ifit(:,:,2)=I;
 Ifit(:,:,3)=I;
for i=i_tip:i_base
    Ifit(i,mns_l(i_base-i+1),1)=255;
    Ifit(i,mns_l(i_base-i+1),2)=0;
    Ifit(i,mns_l(i_base-i+1),3)=0;
    Ifit(i,mns_r(i_base-i+1),1)=255;
    Ifit(i,mns_r(i_base-i+1),2)=0;
    Ifit(i,mns_r(i_base-i+1),3)=0;
end
% for j=j_tl:j_tr
%     Ifit(i_tip,j,1)=0;
%     Ifit(i_tip,j,2)=255;
%     Ifit(i_tip,j,3)=0;
% end
% for j=j_bl:j_br
%     Ifit(i_base,j,1)=0;
%     Ifit(i_base,j,2)=255;
%     Ifit(i_base,j,3)=0;
```

```
% end
imwrite(Ifit,strImgPathXfit);
return


% Functions
function f=Fun_edgl(I,i,djdi,n_row,j_s,j_m,j_ignore)
%Note: j_ignore must be a column vector
gradI_l=zeros(j_m-j_s+1,1,'double');
[n_ignore,m_ignore]=size(j_ignore);

    for j=j_s:j_m
        if (i<5)
            Ii=double(I(i,j));
            If1=double(I(i+1,j));
            If2=double(I(i+2,j));
            If3=double(I(i+3,j));
            If4=double(I(i+4,j));
            If5=double(I(i+5,j));
            If6=double(I(i+6,j));
            If7=double(I(i+7,j));
            If8=double(I(i+8,j));
            grad_i=-761/280*Ii+8*If1-14*If2+56/3*If3-35/2*If4+56/5*If5-14/3*If6+8/7*If7-If8/8;
        elseif (i>n_row-4)
            Ii=double(I(i,j));
            Ib1=double(I(i-1,j));
            Ib2=double(I(i-2,j));
            Ib3=double(I(i-3,j));
            Ib4=double(I(i-4,j));
            Ib5=double(I(i-5,j));
            Ib6=double(I(i-6,j));
            Ib7=double(I(i-7,j));
            Ib8=double(I(i-8,j));
            grad_i=761/280*Ii-8*Ib1+14*Ib2-56/3*Ib3+35/2*Ib4-56/5*Ib5+14/3*Ib6-8/7*Ib7+Ib8/8;
        else
            Iffff=double(I(i+4,j));
            Ifff=double(I(i+3,j));
            Iff=double(I(i+2,j));
            If=double(I(i+1,j));
            Ib=double(I(i-1,j));
            Ibb=double(I(i-2,j));
            Ibbb=double(I(i-3,j));
            Ibbbb=double(I(i-4,j));
            grad_i=(-3*Iffff+32*Ifff-168*Iff+672*If-672*Ib+168*Ibb-32*Ibbb+3*Ibbbb)/840;
        end
        Jffff=double(I(i,j+4));
        Jfff=double(I(i,j+3));
        Jff=double(I(i,j+2));
        Jf=double(I(i,j+1));
        Jb=double(I(i,j-1));
        Jbb=double(I(i,j-2));
        Jbbb=double(I(i,j-3));
        Jbbbb=double(I(i,j-4));
        grad_j=(-3*Jffff+32*Jfff-168*Jff+672*Jf-672*Jb+168*Jbb-32*Jbbb+3*Jbbbb)/840;
        gradI_l(j-j_s+1)=abs(djdi*grad_i-grad_j)/sqrt(djdi^2+1);
    end
    if j_ignore(1)~=-1
        for ii=1:n_ignore
            gradI_l(j_ignore(ii))=0;
        end
    end
    [g_max,j_max]=max(gradI_l,[],1);
    f=j_max+j_s-1;
return

function f=Fun_edgr(I,i,djdi,n_row,j_e,j_m,j_ignore)
%Note: j_ignore must be a column vector
gradI_r=zeros(j_e-j_m,1,'double');
[n_ignore,m_ignore]=size(j_ignore);

    for j=j_m+1:j_e
```

```
        if (i<5)
            Ii=double(I(i,j));
            If1=double(I(i+1,j));
            If2=double(I(i+2,j));
            If3=double(I(i+3,j));
            If4=double(I(i+4,j));
            If5=double(I(i+5,j));
            If6=double(I(i+6,j));
            If7=double(I(i+7,j));
            If8=double(I(i+8,j));
            grad_i=-761/280*Ii+8*If1-14*If2+56/3*If3-35/2*If4+56/5*If5-14/3*If6+8/7*If7-If8/8;
        elseif (i>n_row-4)
            Ii=double(I(i,j));
            Ib1=double(I(i-1,j));
            Ib2=double(I(i-2,j));
            Ib3=double(I(i-3,j));
            Ib4=double(I(i-4,j));
            Ib5=double(I(i-5,j));
            Ib6=double(I(i-6,j));
            Ib7=double(I(i-7,j));
            Ib8=double(I(i-8,j));
            grad_i=761/280*Ii-8*Ib1+14*Ib2-56/3*Ib3+35/2*Ib4-56/5*Ib5+14/3*Ib6-8/7*Ib7+Ib8/8;
        else
            Iffff=double(I(i+4,j));
            Ifff=double(I(i+3,j));
            Iff=double(I(i+2,j));
            If=double(I(i+1,j));
            Ib=double(I(i-1,j));
            Ibb=double(I(i-2,j));
            Ibbb=double(I(i-3,j));
            Ibbbb=double(I(i-4,j));
            grad_i=(-3*Iffff+32*Ifff-168*Iff+672*If-672*Ib+168*Ibb-32*Ibbb+3*Ibbbb)/840;
        end
        Jffff=double(I(i,j+4));
        Jfff=double(I(i,j+3));
        Jff=double(I(i,j+2));
        Jf=double(I(i,j+1));
        Jb=double(I(i,j-1));
        Jbb=double(I(i,j-2));
        Jbbb=double(I(i,j-3));
        Jbbbb=double(I(i,j-4));
        grad_j=(-3*Jffff+32*Jfff-168*Jff+672*Jf-672*Jb+168*Jbb-32*Jbbb+3*Jbbbb)/840;
        gradI_r(j-j_m)=abs(djdi*grad_i-grad_j)/sqrt(djdi^2+1);
    end
    if j_ignore(1)~=-1
        for ii=1:n_ignore
            gradI_r(j_ignore(ii))=0;
        end
    end
    [g_max,j_max]=max(gradI_r,[],1);
    f=j_max+j_m;
return

function ff=Ksi(y,taw)
global n_int

    dt=taw/n_int;
    ff=0;
    t=0;
    for j=1:n_int
        if (y(1)<0)
            ff=ff+(1+y(4)*cos(t))/sqrt(1+y(4)^2+2*y(4)*cos(t));
            t=t+dt;
            ff=ff+(1+y(4)*cos(t))/sqrt(1+y(4)^2+2*y(4)*cos(t));
        else
            ff=ff+(1-(y(4)+2)*cos(t))/sqrt(1+(y(4)+2)^2-2*(y(4)+2)*cos(t));
            t=t+dt;
            ff=ff+(1-(y(4)+2)*cos(t))/sqrt(1+(y(4)+2)^2-2*(y(4)+2)*cos(t));
        end
    end
    ff=ff*dt/2;
```

```
return

function ff=Rho(y,t)
    if (y(1)<0)
        ff=sqrt(1+y(4)^2+2*y(4)*cos(t));
    else
        ff=sqrt(1+(y(4)+2)^2-2*(y(4)+2)*cos(t));
    end
return

function ff=Vol(y)
global n_int
    dt=(y(3)-y(2))/n_int;
    ff=0;
    t=y(2);
    for j=1:n_int
        if (y(1)<0)
            ff=ff+(1+y(4)*cos(t))*sqrt(1+y(4)^2+2*y(4)*cos(t));
            t=t+dt;
            ff=ff+(1+y(4)*cos(t))*sqrt(1+y(4)^2+2*y(4)*cos(t));
        else
            ff=ff+(1-(y(4)+2)*cos(t))*sqrt(1+(y(4)+2)^2-2*(y(4)+2)*cos(t));
            t=t+dt;
            ff=ff+(1-(y(4)+2)*cos(t))*sqrt(1+(y(4)+2)^2-2*(y(4)+2)*cos(t));
        end
    end
    ff=-0.75*ff*dt/2/y(1)^3;
return

function ff=Dist2(p,y,taw)
    rho=Rho(y,taw);
    ksi=Ksi(y,taw)-Ksi(y,y(3));
    r=rho/abs(y(1));
    z=ksi/y(1);
    ff=(r-p(1))^2+(z-p(2))^2;
return

function ff=Error(edg_x1,edg_x2,y)
global n_row_b n_opt_smp ar RMS

dtaw=(y(3)-y(2))/(n_row_b-1);
ff=0;
i=1;
n_i=0;
while i<=n_row_b
   tawi=y(2)+(i-1)*dtaw;
   p(1)=edg_x1(i);
   p(2)=edg_x2(i);
   taw=fminsearch(@(x)Dist2(p,y,x),tawi);
   ff=ff+Dist2(p,y,taw);
   i=i+n_opt_smp+1;
   n_i=n_i+1;
end
dz=(Ksi(y,y(2))-Ksi(y,y(3)))/y(1);
r0=Rho(y,y(2))/abs(y(1));
r1=Rho(y,y(3))/abs(y(1));
RMS=sqrt((ff+(dz-ar)^2+(r0-edg_x1(n_row_b))^2+(r1-edg_x1(1))^2)/(n_i+3));
ff=RMS;
return

function ff=Error(edg_x1,edg_x2,y)
global n_row_b n_opt_smp ar RMS

dtaw=(y(3)-y(2))/(n_row_b-1);
ff=0;
i=1;
n_i=0;
while i<=n_row_b
   tawi=y(2)+(i-1)*dtaw;
   p(1)=edg_x1(i);
   p(2)=edg_x2(i);
```

```
   taw=fminsearch(@(x)Dist2(p,y,x),tawi);
   ff=ff+Dist2(p,y,taw);
   i=i+n_opt_smp+1;
   n_i=n_i+1;
end
dz=(Ksi(y,y(2))-Ksi(y,y(3)))/y(1);
r0=Rho(y,y(2))/abs(y(1));
r1=Rho(y,y(3))/abs(y(1));
RMS=sqrt((ff+(dz-ar)^2+(r0-edg_x1(n_row_b))^2+(r1-edg_x1(1))^2)/(n_i+3));
ff=RMS;
return

 function MM=Gaussian(im, size, sigma, noiseType,mode)

%  im='31.bmp';
%  size=[5 5];
%  sigma=0.375;
%  noiseType='gaussian';

 % Display the original and gray image
 original = imread(im);
 if strcmp(mode,'rgb')
    grayscale = rgb2gray(original);
 end
 if strcmp(mode,'gray')
     grayscale=original;
 end
 %figure(1);
 %imshow(original);
 %figure(2);
 %imshow(grayscale);

 % Add noise to the grayscale image and display
%  noisyImage = imnoise(grayscale , noiseType);
%  figure(4);
%  imshow(noisyImage);

 % Generate Gaussian matrix
 h = fspecial('gaussian', size, sigma);
 %h = fspecial('sobel');
% Convolve the noised image with the Gaussian kernel
 M = conv2(double(grayscale), double(h));

 % Display the result
% figure(3);
% imshow((M.^2).^0.5, []);
MM=(M.^2).^0.5;

return
```

Appendix J

Image processing code II

```
function SO_LB_04()

clc;

image='so_20_i_26.jpg';

I=imread(image);

figure

imshow(I)

blswitch=1;
length=0.002032;
scale=2.08/384.16666667; %mm per number of pixels

rowstart=10;
rowend=10;

topcutoff=8;
botcutoff=8;

I=double(I);

[row, col]=size(I);

%Extracting the left side boundary
left=zeros(1000,3);
count1=1;
for j=rowstart:row-rowend
    dIleft=zeros(512,3);
    for i=5:512-4
        dIleft(i,1)=((-3*I(j,i+4)+32*I(j,i+3)-168*I(j,i+2)+672*I(j,i+1)-672*I(j,i-1)+168*I(j,i-2)
     -32*I(j,i-3)+3*I(j,i-4))/840)^2;
        dIleft(i,2)=((-3*I(j+4,i)+32*I(j+3,i)-168*I(j+2,i)+672*I(j+1,i)-672*I(j-1,i)+168*I(j-2,i)
     -32*I(j-3,i)+3*I(j-4,i))/840)^2;
        dIleft(i,3)=sqrt(dIleft(i,1)+dIleft(i,2));
    end
    [maxval1, maxindex1]=max(dIleft(:,3));
    left(count1,1)=j-4;
    left(count1,2)=maxindex1;
    left(count1,3)=maxval1;
    count1=count1+1;
end

left(count1:end,:)=[];

%Extracting the right side boundary
right=zeros(1000,3);
count2=1;
```

```
for m=rowstart:row-rowend
    dIright=zeros(1024,3);
    for n=512+4:1019
        dIright(n,1)=((-3*I(m,n+4)+32*I(m,n+3)-168*I(m,n+2)+672*I(m,n+1)-672*I(m,n-1)+168*I(m,n-2)
     -32*I(m,n-3)+3*I(m,n-4))/840)^2;
        dIright(n,2)=((-3*I(m+4,n)+32*I(m+3,n)-168*I(m+2,n)+672*I(m+1,n)-672*I(m-1,n)+168*I(m-2,n)
     -32*I(m-3,n)+3*I(m-4,n))/840)^2;
        dIright(n,3)=sqrt(dIright(n,1)+dIright(n,2));
    end
    [maxval2, maxindex2]=max(dIright(:,3));
    right(count2,1)=m-4;
    right(count2,2)=maxindex2;
    right(count2,3)=maxval2;
    count2=count2+1;
end

right(count2:end,:)=[];


if blswitch==1
    [min_left, min_ind]=min(left(800:end,2));                                       %find the
     left most point for base line
    [max_right, max_ind]=max(right(800:end,2));                                     %find the
     right most point for base line
    MAX_ind=max_ind+800;
    MIN_ind=min_ind+800;
    base_row=round((MAX_ind+MIN_ind)/2)+botcutoff;
else
    [min_left, min_ind]=max(left(800:end,2));                                       %find the
     left most point for base line
    [max_right, max_ind]=min(right(800:end,2));                                     %find the
     right most point for base line
    MAX_ind=max_ind+800;
    MIN_ind=min_ind+800;
    base_row=round((MAX_ind+MIN_ind)/2)+botcutoff;
end


%Extracting needle tip and findind the top of the liquid bridge
mid=zeros(1000,3);
count3=1;

for q=390:650
    dImid=zeros(1024,1);
    %%%%%%%%%%% range for avgtop
    for p=500:600
        dImid(p,1)=((-3*I(p+4,q)+32*I(p+3,q)-168*I(p+2,q)+672*I(p+1,q)-672*I(p-1,q)+168*I(p-2,q)
     -32*I(p-3,q)+3*I(p-4,q))/840)^2;
    end
    [maxval3, maxindex3]=max(dImid(:,1));
    mid(count3,1)=maxindex3;
    mid(count3,2)=q;
    mid(count3,3)=maxval3;
    count3=count3+1;
end

mid(count3:end,:)=[];

avgtop=round(sum(mid(:,1))/(count3-1))-topcutoff;

%Left and right segment w/ points only from the top to the bottom
leftseg=left(avgtop:base_row,:);
rightseg=right(avgtop:base_row,:);

[rowseg colseg]=size(leftseg);
mid=zeros(rowseg,1);

%Find the vertical symmetrical line that splits the bridge in half
for u=1:rowseg
    mid(u,1)=(rightseg(u,2)+leftseg(u,2))/2;
end
```

```
avgmid=sum(mid)/rowseg;

%Find the variance between the left and right section
leftdiff=zeros(rowseg,1);
rightdiff=zeros(rowseg,1);

for count6=1:rowseg
    leftdiff(count6)=abs(leftseg(count6,2)-avgmid);
    rightdiff(count6)=abs(rightseg(count6,2)-avgmid);
end

diff=zeros(rowseg,1);

for count7=1:rowseg
    diff(count7)=(leftdiff(count7)-rightdiff(count7))^2;
end

var=sqrt(sum(diff)/rowseg);

hold on
plot(leftseg(:,2),leftseg(:,1),'color','r','linewidth',2)
hold on
plot(rightseg(:,2),rightseg(:,1),'color','g','linewidth',2)
hold on
plot([leftseg(1,2),rightseg(1,2)],[leftseg(1,1),rightseg(1,1)],'b','linewidth',2)
hold on
plot([leftseg(end,2),rightseg(end,2)],[leftseg(end,1),rightseg(end,1)],'y','linewidth',2)
hold on
plot([avgmid, avgmid],[leftseg(end,1),leftseg(1,1)],'m','linewidth',2)


count4=1;                                                               %transform pixel
     coordinates to actual coordinates for the left half
transleft=double(zeros(base_row-avgtop,2));
for w=avgtop:base_row
    transleft(count4,1)=(-1*(left(w,1)-left(base_row,1)))*scale;
    transleft(count4,2)=(left(w,2)-avgmid)*scale;
    count4=count4+1;
end

count5=1;                                                               %transform pixel
     coordinates to actual coordinates for the right half
transright=double(zeros(base_row-avgtop,2));
for v=avgtop:base_row
    transright(count5,1)=(-1*(right(v,1)-right(base_row,1)))*scale;
    transright(count5,2)=(right(v,2)-avgmid)*scale;
    count5=count5+1;
end

rright=transright(1,2);
transright=transright(:,:)./rright;
rleft=transleft(1,2);
transleft=transleft(:,:)./abs(rleft);

%curve fitting

%initial guesses
yg(1)=-0.9;
yg(2)=4.2;
yg(3)=6.8;
yg(4)=0.88;

global numexp ar

%integration steps
ni=200;
sright=[transright(1:10:end,:);transright(rowseg,:)];

%y(1)=q
%y(2)=tau0
%y(3)=tau1
```

```
%y(4)=a

[numexp, col]=size(sright);

%slenderness=h/r0
ar=(sright(1,1)-sright(numexp,1))/sright(1,2);

[y, RMS]=fminsearch(@(x)Error(x,sright,ni),yg);

dz=(Zeta(y,y(2),ni)-Zeta(y,y(3),ni))/y(1);
r0=Rho(y,y(2))/abs(y(1));
RMS=RMS-abs(dz-ar)+abs(r0-1);

rtheo=zeros(ni+1,1,'double');
ztheo=zeros(ni+1,1,'double');

dt=(y(3)-y(2))/ni;
t=y(2);

for i=1:ni+1
    rtheo(i)=Rho(y,t)/abs(y(1));
    ztheo(i)=(Zeta(y,t,ni)-Zeta(y,y(3),ni))/y(1);
    t=t+dt;
end

VV=Vol(y,ni);
vv=ar*VV/(4/3);

dZetadRho_0=-(1+y(4)*cos(y(2)))/sin(y(2))/y(4);
dZetadRho_1=-(1+y(4)*cos(y(3)))/sin(y(3))/y(4);

%contact angle of top and bottom
th_m=atan(-abs(y(1))*dZetadRho_0/y(1))*180/pi+180;
th_cp=atan(abs(y(1))*dZetadRho_1/y(1))*180/pi+180;

n_dia=1.65e-3;
VVnom=(2e-8)/(pi*((n_dia/2)^2)*length);
vvnom=(2e-8)/((4/3)*pi*(n_dia/2)^3);
arnom=length/(n_dia/2);

figure
plot(transleft(:,2),transleft(:,1),'or'), axis([-3 3 0 5])
hold on
plot(transright(:,2),transright(:,1),'og')
hold on
plot(rtheo,ztheo,'-b','markersize',10)

display(var);
display(RMS);
display(y);
display(VV);
display(VVnom);
display(vv);
display(vvnom);
display(ar);
display(arnom);
display(th_m);
display(th_cp);


%transform fitted curve back to pixel coordinates
rightfit=zeros(ni,2);
for count8=1:ni
    rightfit(count8,1)=ztheo(count8)*rright*(1/scale)*-1+right(base_row,1);
    rightfit(count8,2)=rtheo(count8)*rright*(1/scale)+avgmid;
end

leftfit=zeros(ni,2);
for count9=1:ni
    leftfit(count9,1)=ztheo(count9)*rright*(1/scale)*-1+right(base_row,1);
    leftfit(count9,2)=rtheo(count9)*-1*rright*(1/scale)+avgmid;
```

```
end


figure
I=imread(image);
imshow(I)
hold on
plot(rightfit(:,2),rightfit(:,1),'-m','linewidth',2)
hold on
plot(leftfit(:,2),leftfit(:,1),'-m','linewidth',2)


return



function f=Zeta(y,tau,n_inv)
dt=tau/n_inv;
f=0;
t=0;
for j=1:n_inv
    f=f+(1+y(4)*cos(t))/sqrt(1+y(4)^2+2*y(4)*cos(t));
    t=t+dt;
    f=f+(1+y(4)*cos(t))/sqrt(1+y(4)^2+2*y(4)*cos(t));
end
f=f*dt/2;
return



function f=Rho(y,t)
f=sqrt(1+y(4)^2+2*y(4)*cos(t));
return



function f=Dist(p,y,tau,ni)
rho=Rho(y,tau);
zeta=Zeta(y,tau,ni)-Zeta(y,y(3),ni);
r=rho/abs(y(1));
z=zeta/y(1);
f=(r-p(1))^2+(z-p(2))^2;
return



function F=Error(y, expp,ni)
global numexp ar

dtau=(y(3)-y(2))/(numexp-1);
dtau=double(dtau);
f=0;

for i=1:numexp
    taui=y(2)+(i-1)*dtau;
    p(1)=expp(i,2);
    p(2)=expp(i,1);
    tau=fminsearch(@(x)Dist(p,y,x,ni),taui);
    f=f+Dist(p,y,tau,ni);
end

dz=(Zeta(y,y(2),ni)-Zeta(y,y(3),ni))/y(1);
r0=Rho(y,y(2))/abs(y(1));
F=sqrt(f/numexp)+abs(dz-ar)+abs(r0-1);

return



function f=Vol(y,n)
global ar
```

```
dt=(y(3)-y(2))/n;
f=0;
t=y(2);
for j=1:n
    f=f+(1+y(4)*cos(t))*sqrt(1+y(4)^2+2*y(4)*cos(t));
    t=t+dt;
    f=f+(1+y(4)*cos(t))*sqrt(1+y(4)^2+2*y(4)*cos(t));
end
f=(f*dt/2)/(-ar*y(1)^3);
return
```

Bibliography

Adams, R. A. & Fournier, J. J. F. 2003 *Sobolev spaces*. Academic press.

Akbari, A., Akbari, M. & Hill, R. J. 2013 Effective thermal conductivity of two-dimensional anisotropic two-phase media. *Int. J. Heat Mass Tran.* **63**, 41–50.

Akbari, A., Hill, R. J. & van de Ven, T. G. M. 2015*a* Liquid bridge breakup in contact-drop dispensing: Catenoid stability with a free contact line. *Submitted to SIAM J. Appl. Math.* **xx** (x), xxx–xxx.

Akbari, A., Hill, R. J. & van de Ven, T. G. M. 2015*b* Liquid bridge breakup in contact-drop dispensing: dynamics and self-similarity near pinch-off. *To be submitted.* **xx** (x), xxx–xxx.

Akbari, A., Hill, R. J. & van de Ven, T. G. M. 2015*c* Liquid bridge breakup in contact-drop dispensing: Liquid bridge stability with a free contact line. *Submitted to J. Colloid Interf. Sci.* **xx** (x), xxx–xxx.

Akbari, A., Hill, R. J. & van de Ven, T. G. M. 2015*d* Stability and folds in an elastocapillary system. *Submitted to SIAM J. Appl. Math.* **xx** (x), xxx–xxx.

Akhiezer, N. I. & Glazman, I. M. 1993 *Theory of Linear Operators in Hilbert Space*. Courier Dover Publications.

Alince, B. & van de Ven, T. G. M. 1997 Porosity of swollen pulp fibers evaluated by polymer adsorption. In *The Fundamentals of Papermaking Materials* (ed. B. C. F.), *Transactions of the 11th fundamental research symposium*, vol. 2, pp. 771–788. Cambridge, UK: Pira International, Surrey.

Aristoff, J. M., Duprat, C. & Stone, H. A. 2011 Elastocapillary imbibition. *Int. J. Nonlinear Mech.* **46** (4), 648–656.

Arnold, V. I. 1992 *Catastrophe theory*. Springer.

Bachmann, J., Ellies, A. & Hartge, K. H. 2000 Development and application of a new sessile drop contact angle method to assess soil water repellency. *J. Hydrology* **231**, 66–75.

Banerjee, B. & Datta, S. 1981 A new approach to an analysis of large deflections of thin elastic plates. *Int. J. Nonlinear Mech.* **16** (1), 47–52.

Barber, N. 1968 A theoretical model of shrinking wood. *Holzforschung* **22** (4), 97–103.

Barber, N. & Meylan, B. 1964 The anisotropic shrinkage of wood: A theoretical model. *Holzforschung* **18** (5), 146–156.

Bariska, M. 1992 Collapse phenomena in eucalypts. *Wood Sci. Tech.* **26** (3), 165–179.

Bayramli, E. & van de Ven, T. G. M. 1987 An experimental study of liquid bridges between spheres in a gravitational field. *J. Colloid Interf. Sci.* **116** (2), 503–510.

Bico, J., Roman, B., Moulin, L. & Boudaoud, A. 2004 Adhesion: elastocapillary coalescence in wet hair. *Nature* **432** (7018), 690–690.

Bledzki, A. & Gassan, J. 1999 Composites reinforced with cellulose based fibres. *Prog. Polym. Sci* **24** (2), 221–274.

Bobylov, N. A., Emelyanov, S. V. & Korovin, S. 1999 *Geometrical methods in variational problems*. Springer.

Boudaoud, A., Bico, J. & Roman, B. 2007 Elastocapillary coalescence: Aggregation and fragmentation with a maximal size. *Phys. Rev. E.* **76** (6), 060102.

Brenner, M. P., Lister, J. R. & Stone, H. A. 1996 Pinching threads, singularities and the number 0.0304... *Phys. Fluids* **8** (11), 2827–2836.

Brenner, M. P., Shi, X. D. & Nagel, S. R. 1994 Iterated lnstabilities during droplet fission. *Phys. Rev. Lett.* **73**, 3391–3393.

Burdine, N. T., Gournay, L. S. & Reichertz, P. P. 1950 Pore size distribution of petroleum reservoir rocks. *J. Petrol. Technol.* **2** (07), 195–204.

Campbell, M. D. & Coutts, R. S. P. 1980 Wood fibre-reinforced cement composites. *J. Mater. Sci.* **15** (8), 1962–1970.

Chandra, D. & Yang, S. 2009 Capillary-force-induced clustering of micropillar arrays: is it caused by isolated capillary bridges or by the lateral capillary meniscus interaction force? *Langmuir* **25** (18), 10430–10434.

Chandra, D. & Yang, S. 2010 Stability of high-aspect-ratio micropillar arrays against adhesive and capillary forces. *Accounts Chem. Res.* **43** (8), 1080–1091.

Cheng, J. & Kricka, L. J. 2001 *Biochip technology*. Taylor & Francis.

Cheung, H., Ho, M., Lau, K., Cardona, F. & Hui, D. 2009 Natural fibre-reinforced composites for bioengineering and environmental engineering applications. *Compos. Part B-Eng.* **40** (7), 655–663.

Choi, H. G., Choi, D. S., Kim, E. W., Jung, G. Y., Choi, J. W. & Oh, B. K. 2009 Fabrication of nanopattern by nanoimprint lithography for the application to protein chip. *Biochip J.* **3**, 76–81.

Clarke, F. H. 1990 *Optimization and nonsmooth analysis*, , vol. 5. Siam.

Cohen, A. E. & Mahadevan, L. 2003 Kinks, rings, and rackets in filamentous structures. *P. Natl. Acad. Sci. USA* **100** (21), 12141–12146.

Coman, C. D. & Bassom, A. P. 2007 Boundary layers and stress concentration in the circular shearing of annular thin films. *Proc. R. Soc. London A* **463** (2087), 3037–3053.

Cousins, W. J. 1976 Elastic modulus of lignin as related to moisture content. *Wood Sci. Tech.* **10** (1), 9–17.

Cousins, W. J. 1978 Young's modulus of hemicellulose as related to moisture content. *Wood Sci. Tech.* **12** (3), 161–167.

Cown, D. J. & McConchie, D. L. 1980 Wood property variations in an old-crop stand of radiata pine. *New Zeal. J. For. Sci.* **10** (3), 508–520.

Cristian Neagu, R., Kristofer Gamstedt, E., Bardage, S. L. & Lindström, M. 2006 Ultrastructural features affecting mechanical properties of wood fibres. *Wood Mater. Sci. Eng.* **1** (3-4), 146–170.

Davidovitch, B., Schroll, R. D., Vella, D., Adda-Bedia, M. & Cerda, E. A. 2011 Prototypical model for tensional wrinkling in thin sheets. *P. Natl. Acad. Sci.* **108** (45), 18227–18232.

Deladi, S., Tas, N. R., Berenschot, J. W., Krijnen, G. J. M., De Boer, M. J., De Boer, J. H., Peter, M. & Elwenspoek, M. C. 2004 Micromachined fountain pen for atomic force microscope-based nanopatterning. *Appl. Phys. Lett.* **85** (22), 5361–5363.

Diamond, S. 2000 Mercury porosimetry: An inappropriate method for the measurement of pore size distributions in cement-based materials. *Cement Concrete Res.* **30** (10), 1517 – 1525.

Dodds, S., Carvalho, M. & Kumar, S. 2011 Stretching liquid bridges with moving contact lines: The role of inertia. *Phys. Fluids* **23** (9), 092101.

Dodds, S., da Silveira Carvalho, M. & Kumar, S. 2009 Stretching and slipping of liquid bridges near plates and cavities. *Phys. Fluids* **21**, 092103.

Drmanac, R., Sparks, A. B., Callow, M. J., Halpern, A. L., Burns, N. L., Kermani, B. G., Carnevali, P., Nazarenko, I., Nilsen, G. B., Yeung, G. *et al.* 2010 Human genome sequencing using unchained base reads on self-assembling DNA nanoarrays. *Science* **327** (5961), 78–81.

Duprat, C., Aristoff, J. M. & Stone, H. A. 2011 Dynamics of elastocapillary rise. *J. Fluid Mech.* **679**, 641–654.

Duprat, C., Protiere, S., Beebe, A. Y. & Stone, H. A. 2012 Wetting of flexible fibre arrays. *Nature* **482** (7386), 510–513.

Dury-Brun, C., Jury, V., Guillard, V., Desobry, S., Voilley, A. & Chalier, P. 2006 Water barrier properties of treated-papers and application to sponge cake storage. *Food Res. Int.* **39** (9), 1002–1011.

Eggers, J. 1993 Universal pinching of 3d axisymmetric free-surface flow. *Phys. Rev. Lett.* **71** (21), 3458–3460.

Eggers, J. 1997 Nonlinear dynamics and breakup of free-surface flows. *Rev. Mod. Phys.* **69** (3), 865.

Eggers, J. 2012 Stability of a viscous pinching thread. *Phys. Fluids* **24** (7), 072103.

Eggers, J. & Dupont, T. F. 1994 Drop formation in a one-dimensional approximation of the Navier–Stokes equation. *J. Fluid Mech.* **262**, 205–221.

Eichhorn, S., Baillie, C., Zafeiropoulos, N., Mwaikambo, L., Ansell, M., Dufresne, A., Entwistle, K., Herrera-Franco, P., Escamilla, G. & Groom, L. 2001 Review: current international research into cellulosic fibres and composites. *J. Mater. Sci.* **36** (9), 2107–2131.

Erle, M. A., Gillette, R. D. & Dyson, D. C. 1970 Stability of interfaces of revolution with constant surface tension—the case of the catenoid. *Chem. Eng. J.* **1** (2), 97–109.

Farshid-Chini, S. & Amirfazli, A. 2010 Understanding pattern collapse in photolithography process due to capillary forces. *Langmuir* **26** (16), 13707–13714.

Gadegaard, N., Martines, E., Riehle, M. O., Seunarine, K. & Wilkinson, C. D. W. 2006 Applications of nano-patterning to tissue engineering. *Microelectron. Eng.* **83** (4), 1577–1581.

Gällstedt, M., Brottman, A. & Hedenqvist, M. S. 2005 Packaging-related properties of protein-and chitosan-coated paper. *Packag. Tech. Sci.* **18** (4), 161–170.

García Velarde, M. 1988 *Physicochemical Hydrodynamics: Interfacial Phenomena*, , vol. 174. Springer.

Gelfand, I. M. & Fomin, S. V. 2000 *Calculus of variations*. Dover publications.

Gillette, R. D. & Dyson, D. C. 1971 Stability of fluid interfaces of revolution between equal solid circular plates. *Chem. Eng. J.* **2** (1), 44–54.

Giomi, L. & Mahadevan, L. 2012 Minimal surfaces bounded by elastic lines. *Proc. R. Soc. London A* **468** (2143), 1851–1864.

Groom, L., Mott, L. & Shaler, S. 2002 Mechanical properties of individual southern pine fibers. part i. determination and variability of stress-strain curves with respect to tree height and juvenility. *Wood Fiber Sci.* **34** (1), 14–27.

Hafren, J., Fujino, T. & Itoh, T. 1999 Changes in cell wall architecture of differentiating tracheids of pinus thunbergii during lignification. *Plant Cell Physiol.* **40** (5), 532–541.

Halpin, J. C. & Kardos, J. L. 1976 The halpin-tsai equations: a review. *Polym. Eng. Sci.* **16** (5), 344–352.

Hernández, R. E. & Pontin, M. 2006 Shrinkage of three tropical hardwoods below and above the fiber saturation point. *Wood Fiber Sci.* **38** (3), 474–483.

Hofstetter, K. & Gamstedt, E. K. 2009 Hierarchical modelling of microstructural effects on mechanical properties of wood. a review. *Holzforschung* **63** (2), 130–138.

Hofstetter, K., Hellmich, C. & Eberhardsteiner, J. 2005 Development and experimental validation of a continuum micromechanics model for the elasticity of wood. *Eur. J. Mech. A-Solid.* **24** (6), 1030–1053.

van Honschoten, J. W., Escalante, M., Tas, N. R., Jansen, H. V. & Elwenspoek, M. 2007 Elastocapillary filling of deformable nanochannels. *J. Appl. Phys.* **101** (9), 094310.

Howe, W. 1887 Die Rotations-Flächen welche bei vorgeschriebener Flächengrösse ein möglichst grosses oder kleines Volumen enthalten. PhD thesis, Friedrich Wilhelms Universität zu Berlin.

Huang, J., Juszkiewicz, M., De Jeu, W. H., Cerda, E., Emrick, T., Menon, N. & Russell, T. P. 2007 Capillary wrinkling of floating thin polymer films. *Science* **317** (5838), 650–653.

Huo, F., Zheng, Z., Zheng, G., Giam, L. R., Zhang, H. & Mirkin, C. A. 2008 Polymer pen lithography. *Science* **321** (5896), 1658–1660.

Hwang, J. K., Cho, S., Dang, J. M., Kwak, E. B., Song, K., Moon, J. & Sung, M. M. 2010 Direct nanoprinting by liquid-bridge-mediated nanotransfer moulding. *Nat. Nanotechnol.* **5** (10), 742–748.

Innes, T. 1995 Stress model of a wood fibre in relation to collapse. *Wood Sci. Tech.* **29** (5), 363–376.

Kim, H. & Mahadevan, L. 2006 Capillary rise between elastic sheets. *J. Fluid Mech.* **548**, 141–150.

Kim, J. & Kim, H.-Y. 2012 On the dynamics of capillary imbibition. *J. Mech. Sci. Tech.* **26** (12), 3795–3801.

Kjellgren, H., Gällstedt, M., Engström, G. & Järnström, L. 2006 Barrier and surface properties of chitosan-coated greaseproof paper. *Carbohyd. Polym.* **65** (4), 453–460.

KONG, C. S., KIM, D.-Y., LEE, H.-K., SHUL, Y.-G. & LEE, T.-H. 2002 Influence of pore-size distribution of diffusion layer on mass-transport problems of proton exchange membrane fuel cells. *J. Power Sources* **108** (1-2), 185 – 191.

KWON, H., KIM, H., PUËLL, J. & MAHADEVAN, L. 2008 Equilibrium of an elastically confined liquid drop. *J. Appl. Phys.* **103** (9), 093519.

KWON, M., BEDGAR, D. L., PIASTUCH, W., DAVIN, L. B. & LEWIS, N. G. 2001 Induced compression wood formation in douglas fir in microgravity. *Phytochemistry* **57** (6), 847–857.

LAIVINS, G. V. & SCALLAN, A. M. 1993 The mechanism of hornification of wood pulps. *Prod. Papermaking* **2**, 1235.

LANGBEIN, D. W. 2002 *Capillary surfaces: shape-stability-dynamics, in particular under weightlessness*. Springer.

LEE, K.-B., PARK, S.-J., MIRKIN, C. A., SMITH, J. C. & MRKSICH, M. 2002 Protein nanoarrays generated by dip-pen nanolithography. *Science* **295** (5560), 1702–1705.

LIU, G.-Y., XU, S. & QIAN, Y. 2000 Nanofabrication of self-assembled monolayers using scanning probe lithography. *Accounts Chem. Res.* **33** (7), 457–466.

LOWRY, B. J. & STEEN, P. H. 1995 Capillary surfaces: stability from families of equilibria with application to the liquid bridge. *Proc. R. Soc. London A* **449** (1937), 411–439.

LUCASSEN-REYNDERS, E. H., LUCASSEN, J. & GILES, D. 1981 Surface and bulk properties of mixed anionic/cationic surfactant systems i. equilibrium surface tensions. *J. Colloid Interf. Sci.* **81** (1), 150–157.

MADDOCKS, J. H. 1987 Stability and folds. *Arch. Ratio. Mech. An.* **99** (4), 301–328.

MARR, D. & HILDRETH, E. 1980 Theory of edge detection. *Proc. R. Soc. London B* **207** (1167), 187–217.

MARR-LYON, M. J., THIESSEN, D. B., BLONIGEN, F. J. & MARSTON, P. L. 2000 Stabilization of electrically conducting capillary bridges using feedback control of radial electrostatic stresses and the shapes of extended bridges. *Phys. Fluids* **12**, 986.

MARTÍNEZ, I. & PERALES, J. M. 1986 Liquid bridge stability data. *J. Cryst. Growth* **78** (2), 369–378.

MASTRANGELO, C. H. & HSU, C. H. 1993*a* Mechanical stability and adhesion of microstructures under capillary forces. I. Basic theory. *J. Microelectromech. S.* **2** (1), 33–43.

Mastrangelo, C. H. & Hsu, C. H. 1993*b* Mechanical stability and adhesion of microstructures under capillary forces. II. Experimentsperimentsperiments. *J. Microelectromech. S.* **2** (1), 44–55.

Meseguer, J., Slobozhanin, L. A. & Perales, J. M. 1995 A review on the stability of liquid bridges. *Adv. Space Res.* **16** (7), 5–14.

Michael, D. H. 1981 Meniscus stability. *Annu. Rev. Fluid Mech.* **13** (1), 189–216.

Michell, A., Vaughan, J. & Willis, D. 1976 Wood fibre synthetic polymer composites I laminates of paper and polyethylene. *J. Polym. Sci.* **55**, 143–154.

Monica, E., Gellerstedt, G. & Henriksson, G. 2009 *Pulp and Paper Chemistry and Technology*, 1st edn. Berlin: de Gruyter.

Mukerjee, P. & Mysels, K. J. 1971 Critical micelle concentrations of aqueous surfactant systems. *Tech. Rep.*. DTIC Document.

Müller, F. A., Müller, L., Hofmann, I., Greil, P., Wenzel, M. M. & Staudenmaier, R. 2006 Cellulose-based scaffold materials for cartilage tissue engineering. *Biomaterials* **27** (21), 3955–3963.

Mumford, D. & Shah, J. 1989 Optimal approximations by piecewise smooth functions and associated variational problems. *Commun. Pur. Appl. Math.* **42** (5), 577–685.

Mwaikambo, L. 2006 Review of the history, properties and application of plant fibres. *Afr. J. Sci. Tech.* **7**, 120–133.

Myshkis, A. D., Babskii, V. G., Kopachevskii, N. D., Slobozhanin, L. A., Tyuptsov, A. D. & Wadhwa, R. S. 1987 *Low-gravity fluid mechanics*. Springer-Verlag Berlin.

Nakamura, K., Hatakeyama, T. & Hatakeyama, H. 1981 Studies on bound water of cellulose by differential scanning calorimetry. *Text. Res. J.* **51** (9), 607–613.

Neumann, A. W., David, R. & Zuo, Y. 2012 *Applied surface thermodynamics*. CRC Press.

Orr, F. M., Scriven, L. E. & Rivas, A. P. 1975 Pendular rings between solids: meniscus properties and capillary force. *J. Fluid Mech.* **67** (04), 723–742.

Pang, S. 2002 Predicting anisotropic shringkage of softwood Part 1: Theories. *Wood Sci. Tech.* **36** (1), 75–91.

Park, S., Venditti, R. A., Jameel, H. & Pawlak, J. J. 2006 Changes in pore size distribution during the drying of cellulose fibers as measured by differential scanning calorimetry. *Carbohyd. Polym.* **66** (1), 97–103.

Péraud, J. P. & Lauga, E. 2014 Geometry and wetting of capillary folding. *Phys. Rev. E.* **89** (4), 043011.

Piñeirua, M., Tanaka, N., Roman, B. & Bico, J. 2013 Capillary buckling of a floating annulus. *Soft Matter* **9** (46), 10985–10992.

Piner, R. D., Zhu, J., Xu, F., Hong, S. & Mirkin, C. A. 1999 Dip-pen nanolithography. *Science* **283** (5402), 661–663.

Plateau, J. A. F. 1873 *Statique expérimentale et théorique des liquides soumis aux seules forces moléculaires*. Gauthier-Villars.

Pokroy, B., Kang, S. H., Mahadevan, L. & Aizenberg, J. 2009 Self-organization of a mesoscale bristle into ordered, hierarchical helical assemblies. *Science* **323** (5911), 237–240.

Py, C., Reverdy, P., Doppler, L., Bico, J., Roman, B. & Baroud, C. N. 2007 Capillary origami: spontaneous wrapping of a droplet with an elastic sheet. *Phys. Rev. Lett.* **98** (15), 156103.

Py, C., Reverdy, P., Doppler, L., Bico, J., Roman, B. & Baroud, C. N. 2009 Capillarity induced folding of elastic sheets. *Eur. Phys. J. Special Topics* **166** (1), 67–71.

Qian, B. & Breuer, K. S. 2011 The motion, stability and breakup of a stretching liquid bridge with a receding contact line. *J. Fluid Mech.* **666**, 554–572.

Qian, B., Loureiro, M., Gagnon, D. A., Tripathi, A. & Breuer, K. S. 2009 Micron-scale droplet deposition on a hydrophobic surface using a retreating syringe. *Phys. Rev. Lett.* **102** (16), 164502.

Quirk, J. T. 1984 Shrinkage and related properties of douglas-fir cell walls. *Wood Fiber Sci.* **16** (1), 115–133.

Raccurt, O., Tardif, F., d'Avitaya, F. A. & Vareine, T. 2004 Influence of liquid surface tension on stiction of SOI MEMS. *J. Micromech. Microeng.* **14** (7), 1083.

Rayleigh, L. 1879 On the capillary phenomena of jets. *Proc. R. Soc. London A* **29** (196-199), 71–97.

Rivas, D. & Meseguer, J. 1984 One-dimensional self-similar solution of the dynamics of axisymmetric slender liquid bridges. *J. Fluid Mech.* **138**, 417–429.

Robertson, A. A. 1964 Some observations on the effects of drying papermaking fibres. *Pulp Paper Mag. Can.* **65**, 161–168.

Roman, B. & Bico, J. 2010 Elasto-capillarity: deforming an elastic structure with a liquid droplet. *J. Phys. Condens. Mat.* **22** (49), 493101.

Russo, M. J. & Steen, P. H. 1986 Instability of rotund capillary bridges to general disturbances: Experiment and theory. *J. Colloid Interf. Sci.* **113** (1), 154 – 163.

Salaita, K., Wang, Y. & Mirkin, C. A. 2007 Applications of dip-pen nanolithography. *Nat. Nanotechnol.* **2** (3), 145–155.

Salmén, L. & de Ruvo, A. 1985 A model for the prediction of fiber elasticity. *Wood Fiber Sci.* **17** (3), 336–350.

Seydel, R. U. 2009 *Practical bifurcation and stability analysis*. Springer.

Shen, W., Zhong, H., Neff, D. & Norton, M. L. 2009 NTA directed protein nanopatterning on DNA origami nanoconstructs. *J. Am. Chem. Soc.* **131** (19), 6660–6661.

Shi, X. D., Brenner, M. P. & Nagel, S. R. 1994 A cascade of structure in a drop falling from a faucet. *Science* pp. 219–219.

Shim, W., Braunschweig, A. B., Liao, X., Chai, J., Lim, J. K., Zheng, G. & Mirkin, C. A. 2011 Hard-tip, soft-spring lithography. *Nature* **469** (7331), 516–520.

Sirviö, J. & Kärenlampi, P. 1998 Pits as natural irregularities in softwood fibers. *Wood Fiber Sci.* **30** (1), 27–39.

Sirviö, J. A., Kolehmainen, A., Liimatainen, H., Niinimäki, J. & Hormi, O. E. 2014 Biocomposite cellulose-alginate films: Promising packaging materials. *Food Chem.* **151**, 343–351.

Slobozhanin, L. 1983 Stability of the equilibrium state of a capillary liquid with disconnected free surface. *Fluid Dyn.* **18** (2), 171–180.

Slobozhanin, L. A. & Alexander, J. I. D. 1998 Combined effect of disk inequality and axial gravity on axisymmetric liquid bridge stability. *Phys. Fluids* **10**, 2473–2488.

Slobozhanin, L. A., Alexander, J. I. D. & Patel, V. D. 2002 The stability margin for stable weightless liquid bridges. *Phys. Fluids* **14**, 209–224.

Slobozhanin, L. A., Alexander, J. I. D. & Resnick, A. H. 1997 Bifurcation of the equilibrium states of a weightless liquid bridge. *Phys. of Fluids* **9**, 1893–1905.

Slobozhanin, L. A., Gomez, M. & Perales, J. M. 1995 Stability of liquid bridges between unequal disks under zero-gravity conditions. *Microgravity Sci. Tec.* **8** (1), 23–34.

Slobozhanin, L. A. & Perales, J. M. 1993 Stability of liquid bridges between equal disks in an axial gravity field. *Phys. Fluids A* **5**, 1305.

Slobozhanin, L. A. & Perales, J. M. 1996 Stability of an isorotating liquid bridges between equal disks under zero-gravity conditions. *Phys. Fluids* **8** (9), 2307–2318.

Slobozhanin, L. A. & Tyuptsov, A. D. 1974 Characteristic stability parameter of the axisymmetric equilibrium surface of a capillary liquid. *Fluid Dyn.* **9** (4), 563–571.

Smith, P. G. & van de Ven, T. G. M. 1985 The separation of a liquid drop from a stationary solid sphere in a gravitational field. *J. Colloid Interf. Sci.* **105** (1), 7–20.

Stamm, A. J. 1935 Shrinking and swelling of wood. *Ind. Eng. Chem.* **27** (4), 401–406.

Stone, J. E. & Scallan, A. M. 1968 A structural model for the cell wall of water-swollen wood pulp fibres based on their accessibility to macromolecules. *Cell. Chem. Technol.* **2**, 343–358.

Strube, D. 1992 Stability of a spherical and a catenoidal liquid bridge between two parallel plates in the absence of gravity. In *Microgravity Fluid Mechanics*, pp. 263–269. Springer.

Taroni, M. & Vella, D. 2012 Multiple equilibria in a simple elastocapillary system. *J. Fluid Mech.* **712**, 273–294.

Tejado, A. & van de Ven, T. G. M. 2010 Why does paper get stronger as it dries? *Mater. Today* **13** (9), 42–49.

Thompson, J. M. T. & Hunt, G. W. 1984 *Elastic instability phenomena*. Wiley New York.

Thuvander, F., Kifetew, G. & Berglund, L. A. 2002 Modeling of cell wall drying stresses in wood. *Wood Sci. Tech.* **36** (3), 241–254.

Tiemann, H. D. 1941 Collapse in wood as shown by the microscope. *J. Forestry* **39** (3), 271–282.

Timoshenko, S. P. & Gere, J. M. 2009 *Theory of elastic stability*. Courier Dover Publications.

Timoshenko, S. P. & Goodier, J. N. 1951 *Theory of elasticity*. McGraw-Hill.

Timoshenko, S. P., Woinowsky-Krieger, S. & Woinowsky-Krieger, S. 1959 *Theory of plates and shells*. McGraw-hill New York.

Tjahjadi, M., Stone, H. & Ottino, J. 1992 Satellite and subsatellite formation in capillary breakup. *J. Fluid Mech.* **243** (1), 297–317.

Tomotika, S. 1935 On the instability of a cylindrical thread of a viscous liquid surrounded by another viscous fluid. *Proc. R. Soc. London A* **150** (870), 322–337.

Topgaard, D. & Söderman, O. 2002 Changes of cellulose fiber wall structure during drying investigated using nmr self-diffusion and relaxation experiments. *Cellulose* **9** (2), 139–147.

TORQUATO, S. 2002 *Random heterogeneous materials: microstructure and macroscopic properties*. Springer.

VELLA, D., ADDA-BEDIA, M. & CERDA, E. 2010 Capillary wrinkling of elastic membranes. *Soft Matter* **6** (22), 5778–5782.

VOGEL, T. I. 1989 Stability of a liquid drop trapped between two parallel planes ii: General contact angles. *SIAM J. Appl. Math.* **49** (4), 1009–1028.

VOGEL, T. I. 1996 On constrained extrema. *Pac. J. Math.* **176** (2), 557–561.

VOGEL, T. I. 1999 Non-linear stability of a certain capillary problem. *Dyn. Contin. Discret. I.* **5**, 1–16.

VOGEL, T. I. 2000 Sufficient conditions for capillary surfaces to be energy minima. *Pac. J. Math.* **194** (2), 469–489.

WALKER, J. C. 2006 *Primary wood processing: principles and practice*. Springer.

WALTER, W. 1998 *Ordinary differential equations*. Springer.

WEISE, U., MALONEY, T. & PAULAPURO, H. 1996 Quantification of water in different states of interaction with wood pulp fibres. *Cellulose* **3** (1), 189–202.

YAMAMOTO, H. 1999 A model of the anisotropic swelling and shrinking process of wood. part 1. generalization of barber's wood fiber model. *Wood Sci. Tech.* **33** (4), 311–325.

YAMAMOTO, H., SASSUS, F., NINOMIYA, M. & GRIL, J. 2001 A model of anisotropic swelling and shrinking process of wood. *Wood Sci. Tech.* **35** (1-2), 167–181.

YAN, D. & LI, K. 2008 Measurement of wet fiber flexibility by confocal laser scanning microscopy. *J. Mater. Sci.* **43** (8), 2869–2878.

ZHOU, L. 1997 On stability of a catenoidal liquid bridge. *Pac. J. Math.* **178** (1), 185–197.