Contact Angle Measurement Technique for Rough Surfaces

By

Russell Stacy

A THESIS

Submitted in partial fulfillment of the requirements

for the degree of

Master of Science in Mechanical Engineering

MICHIGAN TECHNOLOGICAL UNIVERSITY

2009

This thesis, "Contact Angle Measurement Technique for Rough Surfaces" is hereby approved in partial fulfillment of the requirements of the Degree of Master of Science in Mechanical Engineering.

Department of Mechanical Engineering - Engineering Mechanics

Advisor: ____________________
Dr. Jeffrey S. Allen

Committee Member: ____________________
Dr. Chang Kyoung Choi

Committee Member: ____________________
Dr. Jaroslaw W. Drelich

Department Chair: ____________________
Professor William W. Predebon

Date: ____________________

ABSTRACT

Increased transportation fuel costs and environmental awareness has led to an acceleration of research into alternative energy production such as low-temperature fuel cells for automotive use. One of the largest remaining problems facing low-temperature fuel cells is water management. Improvement of water management in low-temperature fuel cells has led to the investigation of the wettability characteristics of fuel cell components, namely gas diffusion layers, as quantified by the contact angle. Current measurement techniques are not suitable for making accurate contact angle measurements on rough surfaces due to poor optical resolution at the contact line. A technique for accurately measuring contact angles on rough surfaces has been developed using a sessile drop method. The technique requires a drop profile image which is processed to extract the solid surface and drop interface data. The Laplace-Young equation is numerically integrated to generate a Laplacian curve which matches the upper portion of the drop profile. The contact angle is then extracted from the Laplacian curve. This technique is advantageous because it removes the dependence of the contact angle measurement from imaging of the contact line.

ACKNOWLEDGMENTS

I would like to thank my wife for her continued support and understanding, my friends and family for believing in me, and my advisor for his wisdom and patience.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

High fuel costs and environmental concerns have led to an acceleration of research into alternative energies for automotive applications, namely fuel cells. Polymer electrolyte membrane (PEM) fuel cells are a leading fuel cell technology for automotive applications due to their low operating temperature and their ability to resist corrosion and contamination [1]. One of the largest remaining problems facing PEM fuel cells is poor water management.

In the case of a basic PEM fuel cell (Figure 1.1), hydrogen and oxygen react to form water, electricity, and heat. The fuel cell membrane must remain fully hydrated and excess water must be removed for proper fuel cell operation. Water is formed near the membrane, typically on the cathode side, and excess water travels through the gas diffusion layer (GDL) to form a drop on the GDL surface. The drops that form on the GDL surface are removed from the cathode channels by the reactant flow. Excess water can form plugs if not properly removed from the fuel cell channels. These water plugs can significantly reduce both fuel cell performance, by reducing the number of reacting sites available in the fuel cell, as well as lifetime of the fuel cell, by increasing thermal gradients and therefore thermal stresses in fuel cell components [1, 2].

An understanding of GDL wettability corresponding to surface roughness, pore size, pore distribution, and surface coating will allow for GDLs to be designed for optimal water management. GDL wettability will be characterized by contact angle measurements, unfortunately current contact angle measurement techniques are not well defined for rough, porous materials. Therefore a technique for measuring contact angles on rough, porous materials has been developed.

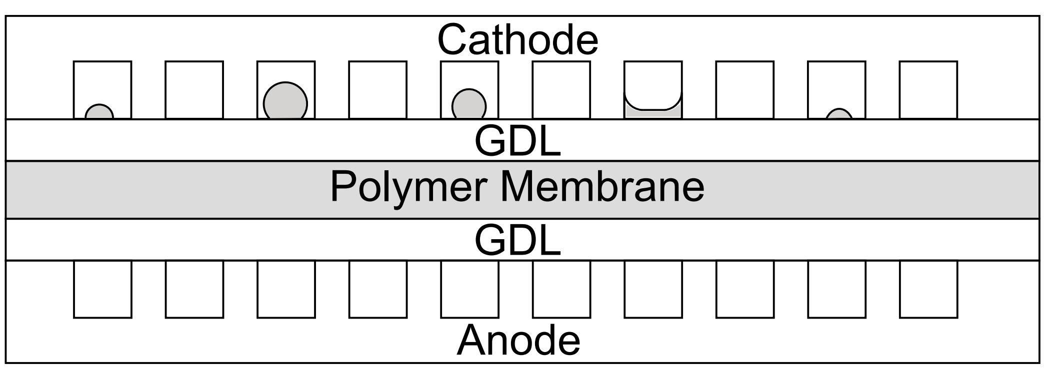


Figure 1.1. Cross sectional view of a basic PEM fuel cell

A definition of contact angle is needed before discussing contact angle measurement techniques. The contact angle can be defined in several ways. Qualitatively, a contact angle is the macroscopic representation of microscopic phenomena. Microscopic characteristics such as surface roughness, surface energies of the materials involved, and surface coatings play a role in the wettability of a material for a given fluid. Quantitatively, a contact angle is the interior angle formed by the substrate being used and the tangent to the drop interface at the apparent intersection of all three interfaces. This intersection is called the contact line. Figure (1.2) illustrates the tangent line and contact angle of a liquid drop on a surface. Historically a static contact angle on a flat surface is defined by the Young Equation (1.1) [3] using interfacial surface tensions between solid and liquid, σ_{SL}, solid and vapor, σ_{SV}, and liquid and vapor, σ_{LV}. Young's equation is essentially a force balance in the horizontal direction. The contact angle may also be directly measured to calculate the ratio of interfacial surface tensions if the interfacial surface tensions are unknown.

$$\sigma_{LV} \;\cos\theta = \sigma_{SV} - \sigma_{SL} \tag{1.1}$$

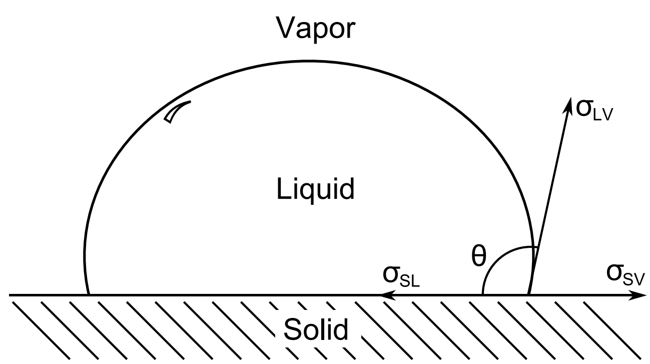


Figure 1.2. Young's model showing the relationship between the three interfacial tensions (solid and liquid, σ_{SL}, solid and vapor, σ_{SV}, and liquid and vapor, σ_{LV}) and the contact angle, θ

2. TECHNIQUES FOR MEASURING CONTACT ANGLES

There are several methods of determining contact angles. On a thin, flat substrate such as a GDL, the two principle methods available are Wilhelmy plate method and goniometry.

2.1 Wilhelmy Plate Method

Tensiometry was originally developed to find surface tension of liquids. A specific form of tensiometry, the Whilhelmy plate method, can be adapted to calculate contact angles of liquids with known surface tensions using a force balance [3]. The Whilhelmy plate method uses a bulk fluid which is raised towards the plate until the plate is submerged in the fluid. The method is illustrated in Figure (2.1). Using the general force balance Equation (2.1) for tensiometry, the contact angle, θ, can be calculated using the recorded change in weight, ΔW, the wetted object perimeter, p, and surface tension, σ.

$$\sigma \cos\theta = \frac{\Delta W}{p} \tag{2.1}$$

The roughness and pores of GDLs make it difficult to find a perimeter and also may cause wicking of the fluid into the material and give inaccurate weight measurements and therefore inaccurate contact angle results.

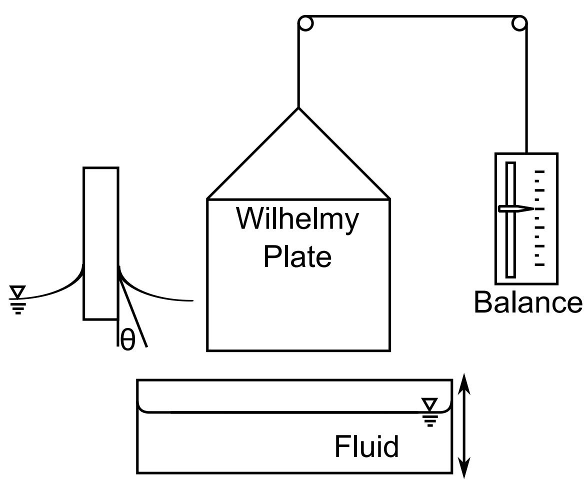


Figure 2.1. Measuring contact angle via Wilhelmy plate tensiometry.

2.2 Goniometry

Goniometry uses a profile image of a drop to find contact angles. Contact angles can be directly measured using direct inspection. Other goniometer methods model the drop interface in order to find the contact angle using one of three approaches; approximations, curve fitting, or interface modeling.

Direct inspection is the easiest goniometer technique to perform. An image of a drop profile is printed and the substrate surface and tangent line to the drop interface at the contact line are drawn using a straight edge. The contact angle is then directly measured using a protractor. An alternate approach would be to use a digital image and a drawing software to draw the surface and tangent lines and measure the contact angle. This technique, although easy to perform, is prone to large inaccuracies in contact angle measurements. User interpretation of the tangent line and surface as well as improper imaging and poor images lead to large variance in contact angle measurements, as illustrated in Figure (2.2).

Two approximation methods are the spherical cap approximation and the small slope approach. As the name implies, the spherical cap approximation models a drop interface as a spherical cap. This approximation may only be used accurately when the drop characteristics meet a specific criteria; the drop must be small enough that gravitational effects are minimal. This criteria may be quantified using the Bond number, Bo$= \rho g l^2/\sigma$, which is a ratio of capillary effects to gravitational effects, where ρ is density of the liquid, g is the acceleration due to gravity, l is the characteristic length scale, and σ is the surface tension of the liquid. The capillary length, $L_c = \sqrt{\sigma/\rho g}$, otherwise known as the Laplace constant, may also be used to quantify the relative effects of gravity verses capillarity. A Bo $= 1$ implies that gravitational effects and capillary effects are equal, with Bo > 1 meaning gravitation effects start to

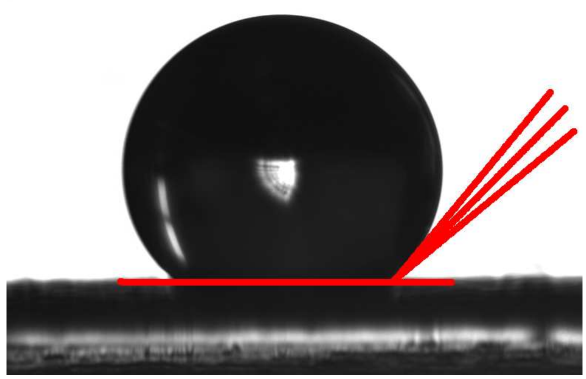


Figure 2.2. Illustration of variance in contact angle using the direct inspection method. This image is of water on GDL 6 and has a drop height of 0.155 centimeters.

dominate and Bo < 1 meaning capillary effects start to dominate. The characteristic length scale is the radius at the drop equator for non-wetting drops (contact angles greater than 90°) and the wetted radius for wetting drops (contact angles less than 90°).

The small slope approach uses a simplified Laplace-Young equation to model a drop interface. This approach does not have a size constraint like the spherical cap approximation. However, a constraint on contact angle, which must be less than 30° [4], is imposed.

Curve fitting techniques typically model the drop interface without constraints on size or contact angle. Most curve fitting techniques require the drop interface to be defined using individual data points so a curve fitting routine or algorithm may be used. One approach is to use a polynomial fitting technique to fit the drop interface. The contact angle is determined using the tangent to the drop interface at the contact line, which is defined as the slope of the polynomial expression. Inaccuracies in contact angle measurements using a polynomial fitting technique come from inaccurately defining the tangent to the drop interface at the contact line.

A more accurate approach is to model the drop interface using the Laplace-Young equation (2.2) which relates the total change in pressure across a curved liquid surface, ΔP, to the two principle radii of curvature, $1/R_1$ and $1/R_2$, and surface tension, σ.

$$\left(\frac{1}{R_1} + \frac{1}{R_2}\right)\sigma = \Delta P \tag{2.2}$$

Bashforth and Adams [5] were the first to develop a numerical solution to the Laplace-Young equation and published solutions to the equation in the form of tables. Hartland and Hartley [6] later modified the Laplace-Young equation to study axisymmetric fluid-liquid interfaces and also published solutions to the Laplace-Young equation in the form of tables. Li et al. [7] numerically integrate the Laplace-Young equation using a Runga-Kutta method coded in Fortran, called Axisymmetric Drop Shape Analysis (ADSA), to find contact angles. Another approach is to use a finite element method (FEM) to numerically integrate the Laplace-Young equation [8].

2.3 Development of the Laplace-Young Model for Sessile Drops

The general form of the Laplace-Young equation is shown below with R_1 and R_2 as the primary radii of curvature and σ as the surface tension. The pressure term is divided into two separate components.

$$\left(\frac{1}{R_1} + \frac{1}{R_2}\right)\sigma = \Delta P_{\mathrm{g}} + \Delta P_{\sigma} \tag{2.3}$$

where ΔP_{g} is the change in hydrostatic pressure and ΔP_{σ} is the change in pressure due to the curvature of the drop.

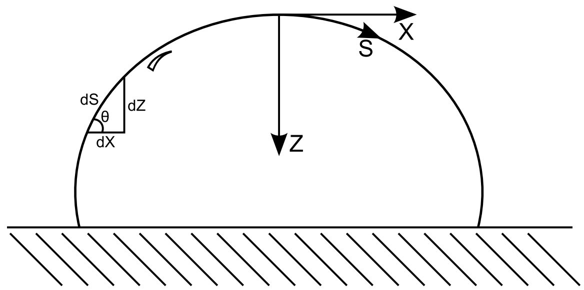


Figure 2.3. Coordinate system used to solve the Laplace-Young equation showing the relationship between X, Z, S, and θ

Since sessile drops do not always have constant curvature over the entire interface, the pressure and curvature terms need to be expressed in a local coordinate system. Cartesian coordinates are convenient when the contact angle is less than 90°. [4] For the gas diffusion media in PEM fuel cells, the contact angle is always greater than 90°. Therefore, a spherical coordinate system is used to describe local curvatures and pressures. Figure (2.3) illustrates the coordinate system with origin located at the apex of the drop.

The hydrostatic pressure drop from the apex of the drop is expressed as

$$\Delta P_{\mathrm{g}} = \rho g z \tag{2.4}$$

At the apex of the drop with boundary conditions $\Delta P_{\mathrm{g}} = 0$ and for an axisymmetric drop, $R_1 = R_2 = b$. Thus, the Laplace-Young equation at $z = 0$ reduces to

$$\frac{2\sigma}{b} = \Delta P_{\sigma} \tag{2.5}$$

Therefore at any z, the pressure balance can be expressed as

$$\left(\frac{1}{R_1} + \frac{1}{R_2}\right)\sigma = \frac{2\sigma}{b} + \rho g z \tag{2.6}$$

The local curvature, $1/R_1 + 1/R_2$, can be expressed in spherical coordinates by recognizing that [3]

$$\frac{1}{R_1} = \frac{d\theta}{ds} \tag{2.7}$$

$$\frac{1}{R_2} = \frac{\sin\theta}{x} \tag{2.8}$$

Equation (2.6) is transformed to

$$\frac{d\theta}{ds} = \frac{2}{b} + \frac{\rho g z}{\sigma} - \frac{\sin\theta}{x} \tag{2.9}$$

where $d\theta/ds$ is the radius of curvature, R_1, in spherical coordinates, $2/b$ is the apex curvature term, $\rho gz/\sigma$ relates gravitational effects to capillary effects (notice $\rho g/\sigma$ is ${L_c}^{-2}$), $\sin\theta/x$ is the radius of curvature, R_2, in spherical coordinates. The final step is to non-dimensionalize equation (2.9) using the ratio of physical properties of the fluid; defined as c, which has dimensions of 1/length^2.

$$c = \frac{\rho g}{\sigma} \tag{2.10}$$

It should be noted that c is related to the capillary length, L_c, in the form of $L_c = c^{-1/2}$. The dimensionless coordinates and apex curvature are defined as

$$X = xc^{\frac{1}{2}} \tag{2.11}$$

$$Z = zc^{\frac{1}{2}} \tag{2.12}$$

$$S = sc^{\frac{1}{2}} \tag{2.13}$$

$$B = bc^{\frac{1}{2}} \tag{2.14}$$

Applying this non-dimensionalization to Equation (2.9) and adding two geometric relationships results in a set of first-order, ordinary differential equations whose solution fully defines a drop profile,

$$\frac{d\theta}{dS} = \frac{2}{B} + Z - \frac{\sin\theta}{X} \tag{2.15}$$

$$\frac{dX}{dS} = \cos\theta \tag{2.16}$$

$$\frac{dZ}{dS} = \sin\theta \tag{2.17}$$

2.4 Predicting Contact Angles on Rough Surfaces

Wenzel [9] developed Equation (2.18) to calculate the theoretical contact angle, θ_{W}, on rough surfaces. The theoretical contact angle is calculated using the contact angle on a smooth surface of the same material, θ, and the roughness factor, ζ, which is the ratio of true surface area to projected surface area.

$$\cos\theta_{\mathrm{W}} = \zeta \ \cos\theta \tag{2.18}$$

Later, Cassie [10] developed two equations to calculate theoretical contact angles on heterogeneous surfaces and porous materials. Equation (2.19) calculates the theoretical contact angle, θ_{C}, on heterogeneous surfaces with two different contact angles, θ_1 and θ_2, with fractional areas of each surface under a drop, f_1 and f_2. Equation (2.20) calculates the theoretical contact angle, θ_{C}, on porous surfaces where f_1 is the fractional area solid-liquid interfaces and f_2 is the fractional area of liquid-air interfaces or pores when a drop is placed on the surface.

$$\cos\theta_{\mathrm{C}} = f_1 \ \cos\theta_1 + f_2 \ \cos\theta_2 \tag{2.19}$$

$$\cos\theta_{\mathrm{C}} = f_1 \ \cos\theta_1 - f_2 \tag{2.20}$$

Wenzel's approach is difficult to apply to a non-patterned, rough, porous material because of the need to know the true surface area of the substrate. Cassie's approach can be used for porous materials, however it becomes difficult to apply this approach when using a material that is porous with a non-patterned roughness. Both approaches require knowledge of the contact angle on a smooth surface of the same material, which is impossible to obtain when dealing with typical GDLs which are coated with PTFE.

2.5 Axisymmetric Drop Model

The models and techniques for measuring contact angles described in this chapter are not applicable to the measurement of contact angles on rough porous materials such as GDLs, therefore a technique based on the sessile drop method using the Laplace-Young equation has been developed. The general steps for measuring contact angles using the developed technique include capturing a profile image of a sessile drop on the desired substrate, processing the drop image to obtain the drop interface data and substrate surface data, and finding a solution to the set of first-order, non-linear differential equations which produces a Laplacian curve which matches the drop interface. The solution is found by minimizing the distance between the Laplacian curve and the drop interface data at a point above the contact line. These steps are discussed in greater detail in the following chapter.

3. CONTACT ANGLE MEASUREMENTS ON GDLS

This chapter explains in detail the process of making contact angle measurements on GDLs. An illustration of the experimental apparatus used to take profile images of drops is shown in Figure (3.1). Components of the setup include the backlight (1), enclosure and drop stage (2), long working distance microscope (3), CCD camera (4), and workstation (5). Details on the experimental apparatus are provided in Appendix (D).

The process of making contact angle measurements begins with capturing a profile image of a sessile drop and an image of an optical scale using the experimental apparatus. The images are processed to extract the drop interface and surface data using a code developed in MATLAB®. A solution that produces a Laplacian curve which matches the drop interface is determined by numerically integrating the Laplace-Young equation using MATLAB®. The solution to the Laplace-Young equation fully defines a drop interface from the apex, where $\theta = 0$, to the maximum theoretical drop height for that given solution, where $\theta = 180°$.

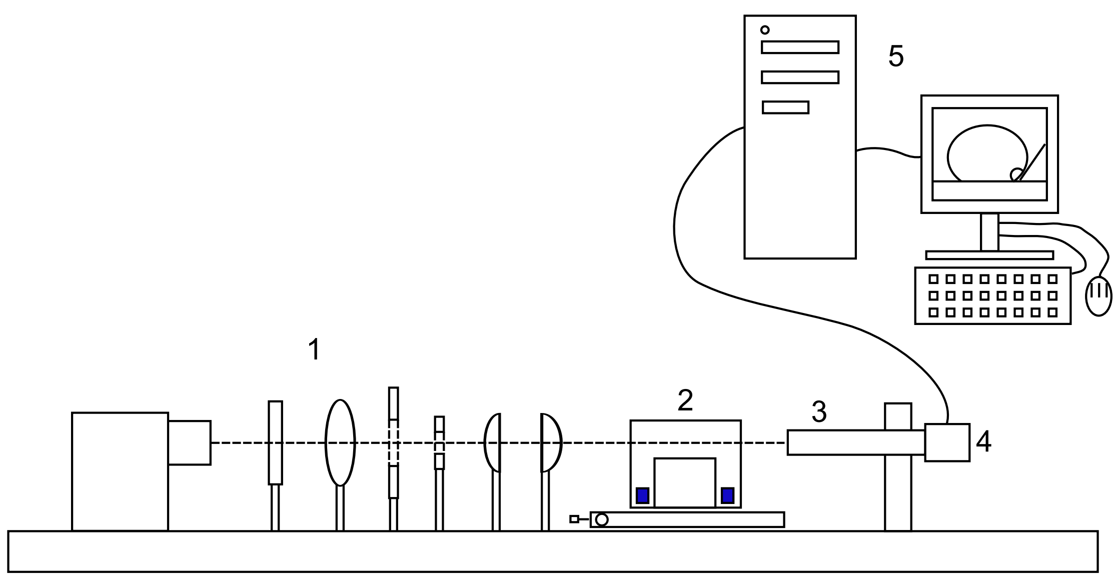


Figure 3.1. Diagram of experimental setup. 1: Köhler illumination, 2: Enclosure and Drop stage, 3: Long working distance microscope, 4: CCD Camera, 5: Workstation

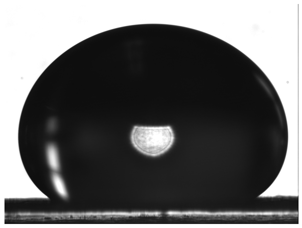
There are several steps to properly capture a profile image of a sessile drop. The temperature of the enclosure must first be set and allowed to stabilize and the enclosure must be completely humidified by filling the containers inside of the enclosure with water and allowing the vapor pressure to equilibrate. Köhler illumination is used as the backlight which provides an equal intensity light beam and improves contrast between the drop interface and the background of the drop image. Before the substrate is placed on the drop stage, the stage is leveled front to rear and side to side. The substrate is then placed on the drop stage.

In the case of the FC-721 coated glass, the coated glass slide is simply placed on the drop stage. GDL samples are cut from a larger sheet and then taped to a glass slide to eliminate problems with the GDL moving on the drop stage as well as problems with the GDL edges rolling up and interfering with imaging.

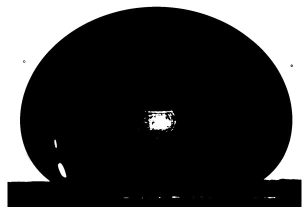
Once the substrate is placed on the drop stage, drops of water are manually placed on the substrate using a syringe and needle. The drops are then brought into the field of view with the substrate surface visible. The entire drop must fit into the image with at least some of the substrate visible on each side of the drop. The long working distance microscope is adjusted to focus the image. An image of the drop is then captured using the software that controls the camera (EPIX XCAP). Once the first image is captured, the x-y translation stage is used to focus subsequent drop images on the same substrate. If there is more than one drop on the substrate, one drop is brought into view at a time and drop images are captured individually. When all images are captured, the substrate is removed from the drop stage and an optical micrometer (Klarmann Rulings KR-812) is placed on the stage. The micrometer is aligned in the field of view and an image of the micrometer is captured. This image is then used to calculate a scale factor relating pixels to centimeters.

The optical scale image is opened in MATLAB® and using a built-in function, `ginput`, a user selects two scale marks which correlates to the pixel position on the image, and enters the number of millimeters between the two marks. MATLAB® uses the pixel information as well as the distance between the two marks to calculate a scale factor in the form of pixels/centimeters. Now an image of a drop profile is loaded into MATLAB®, which is converted to black and white using the built-in function, `im2bw` . A built-in function, `bwtraceboundary`, is performed on the black and white image to extract the substrate surface data and drop interface data. The entire image conversion process and data extraction is shown in Figure (3.2). The `bwtraceboundary` function requires two inputs, a starting point and direction to start, which is defined as origin of the image or the upper left corner of the image and starts in the downward direction. This function looks for and stores the points where pixels change from white to black or black to white, finding the substrate surface location and the drop interface data.

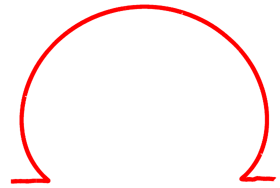
The data that is found is in units of pixels and using the calculated scale factor the data is converted to centimeters. The coordinate system is transformed so the apex coincides with the origin and then non-dimensionalized in the manner of Hartland



(a) Original Image



(b) Black and White Image



(c) Drop Boundary

Figure 3.2. Process to find drop interface data starting with the original image, converted to a black and white image, using a tracing function to find the drop interface. Image shows water on GDL 6 with a drop height of 0.358 centimeters.

and Hartley [6] using $c = \rho g/\sigma$, where ρ is density, g is gravitational acceleration, and σ is the surface tension of the liquid. Alternatively, the capillary length, $L_c = \sqrt{\sigma/\rho g}$ may be used to non-dimensionalize the data. The non-dimensionalization and transformation of the coordinate system are necessary for physical data to be readily comparable to the theoretical data. The surface of the substrate is calculated to be an average of the data found on each side of the drop. If the surface location from the data on the left side of the drop is significantly different, roughly 25 microns, than the surface location on the right, the substrate is considered not level and that image is not used. The camera may also be tilted which may make it difficult to obtain accurate substrate surface data. Leveling of the camera takes place every time a new set of data is captured to ensure accurate substrate surface data is obtained.

The theoretical data is the solution to the Laplace-Young equation which matches or defines the physical data. A solution to the Laplace-Young equation for a given b and c value is found using the built in ordinary differential equation solver in MATLAB®, `ode45`, which is based on a fourth order Runga-Kutta method. The solution can be varied by changing the boundary conditions, b and c.

Figure (3.3) illustrates how Laplacian curves change while varying b with a fixed c value of 14, roughly the c value of water. Drop profiles produced using small b values tend to appear small and spherical in nature, whereas drop profiles produced using large b values tend to appear larger and more like a puddle. This is attributed to the fact that b is the radius of curvature at the apex, and therefore a small b value produces a small drop and a large b value produces a drop with an increasingly flat top. Figure (3.4) illustrates a change in Laplacian curves for a fixed b value of 1 and varying c values. The Laplacian curve with a c value of 14 approximates a drop interface shape of water for $b = 1$, assuming a density of 1000 kg/m^3, gravitational acceleration of 9.8 m/s^2, and a surface tension of 70 dynes/cm. Assuming a fixed density and gravitational acceleration, a c value of 6 would provide a surface tension of approximately 163 dynes/cm and a c value of 22 would provide a surface tension of approximately 45 dynes/cm. The shape and size of the drop profiles in Figure (3.4) do not appear to change significantly for the range of c values shown.

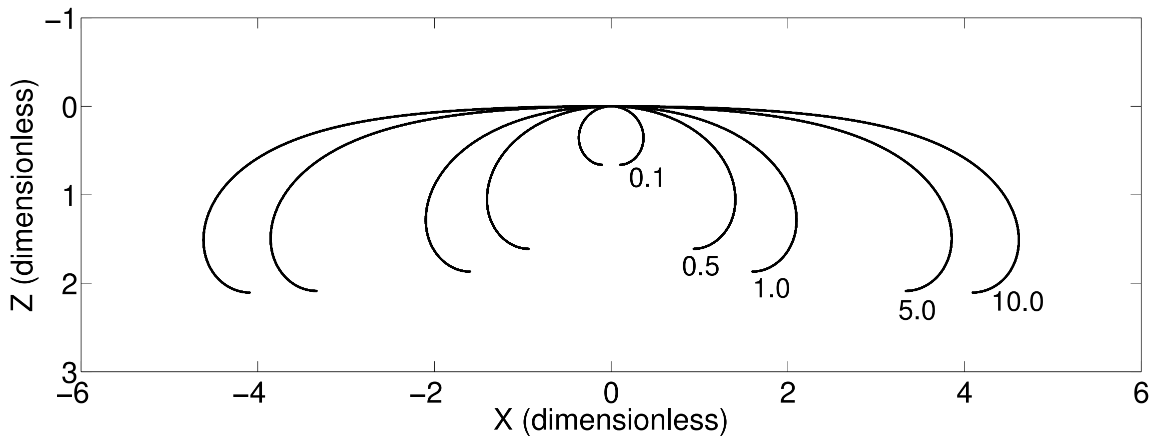


Figure 3.3. Laplacian Curves generated by varying b for $c = 14$

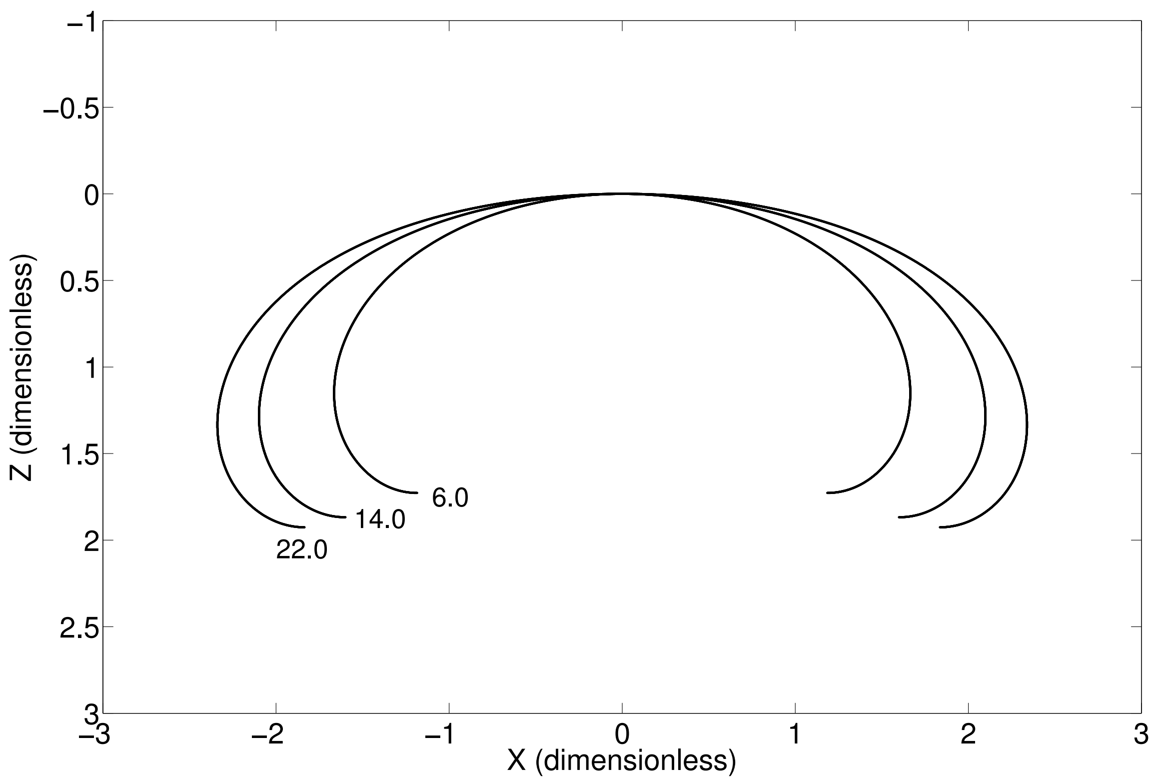


Figure 3.4. Laplacian Curves generated by varying c for $b = 1$

A fitting routine is used to find the solution to the Laplace-Young equation which matches the physical data. This is done by reducing the horizontal distance between the physical data and the theoretical data at a distance above the substrate surface. The purpose of not fitting the theoretical data at the contact line is the poor resolution of the images at the contact line, which can be seen in Figure (3.5). For large Bo drops, b and c can be solved for simultaneously to find the surface tension, σ, and contact angle, θ. However, c should be fixed for small Bo drops due to the relative insignificance of c for small Bo drops as previously mentioned. When fitting a Laplacian curve to the physical data, b is varied to minimize the horizontal distance between the physical data and Laplacian curve at a point roughly 5% of the drop height above the substrate surface. For large Bo drops, b and c may be varied simultaneously. In this case, b is varied to minimize the horizontal distance between the physical data and Laplacian curve at a point roughly 5% of the drop height above the substrate surface and c is varied to minimize the horizontal distance between the physical data and Laplacian curve at a point roughly half way between the drop apex and the substrate surface. The contact angle measurement program, the scale factor program, variation in b plot program, and variation in c plot program are listed in Appendices (E) through (H), respectively.

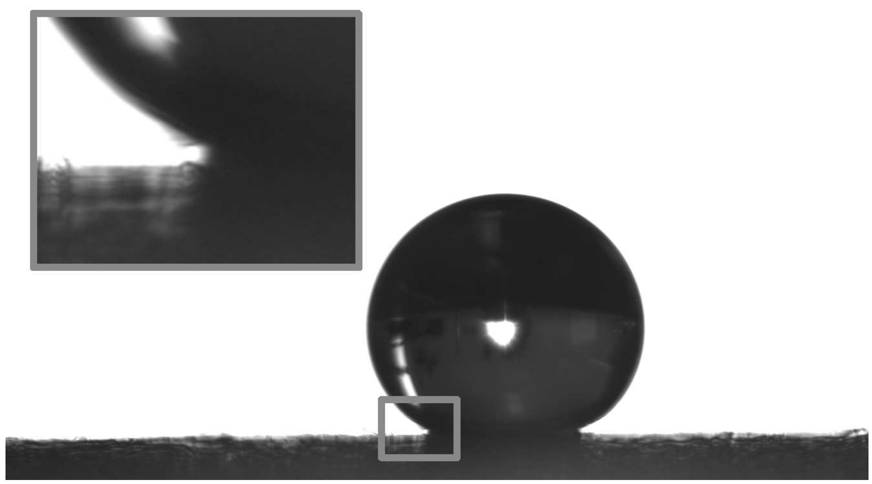


Figure 3.5. Water on GDL 11 with inset of the poor resolution at the contact line. Drop height is 0.201 centimeters.

4. DISCUSSION

4.1 Code Verification

The first step in verifying the code was to test the convergence of the code for a single drop image. A large drop (Figure 4.1) on an FC-721 coated glass slide was captured and processed while varying the initial surface tension value, which in effect varies the initial c value. Table 4.1 shows the results from this drop image that was processed with varying the starting value of σ between 50 dynes/cm to 80 dynes/cm. The results show a convergence of the contact angle to be 115.45°± 0.15° and surface tension to be 69.34 dynes/cm ± 0.34 dynes/cm. This test demonstrates the program can resolve contact angle and surface tension simultaneously regardless of the starting σ, or c, value.

This measurement technique can be used to measure a contact angle and surface tension simultaneously. This would only be possible for large Bo drops, though. Small Bo drops have little dependence on gravity, therefore the interface shape of a small

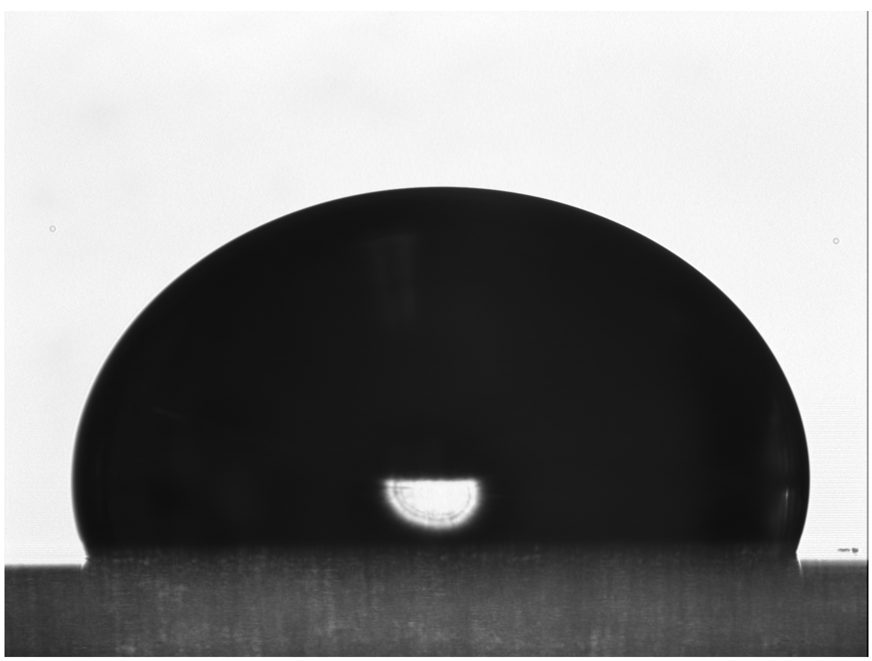


Figure 4.1. Image of water on FC-721 coated glass slide used for convergence study. Drop height is 0.324 centimeters.

Table 4.1. Convergence test data for water on FC-721 (Figure 4.1). The initial value for surface tension was varied for each test run. This, in turn, resulted in a unique initial value of c. Both b and c were allowed to vary while matching the Laplace curve to the drop profile. Final values after successful matching are shown for the Bond number, c [1/cm^2], contact angle (θ) [deg.], and surface tension (σ) [dynes/cm]. The final value of the surface tension is derived from c assuming a fixed value for the density and gravitational acceleration.

test	initial value	final values			
run	σ	σ	c	Bo	θ
1	55	69.00	14.191	1.457	115.6
2	60	69.40	14.109	1.448	115.5
3	65	69.67	14.054	1.443	115.4
4	70	69.55	14.079	1.445	115.3
5	75	69.22	14.147	1.452	115.5
6	80	69.29	14.131	1.451	115.4

drop becomes invariant to c.

4.2 Contact Angle Measurements on FC-721

After the code was verified for convergence on a single drop, several drops of varying size were then placed on the FC-721 coated glass slide and imaged. The results, Figure (4.3), show a variance in contact angle as a function of the drop size presented as the non-dimensionless ratio of equator radius, r, over capillary length, L_c. This ratio is $\sqrt{\text{Bo}}$. The results show contact angle changes with drop size, which has been reported by Drelich et al. [11], Drelich [12], and Vafaei and Podowski [13]. Figure (4.3) shows that contact angles range within 10° for a given size and shows a general linear increase in contact angle when plotted as θ vs $\log(r/L_c)$.

Possible causes of this trend could be related to drop deposition [11], line tension [14], or contact line pinning. In the case of small drops, the drop is formed on the end of the needle and placed on the substrate surface and the needle is removed. Some of the drop volume may be removed on the tip of the needle during this process after the drop has pinned to the surface, resulting in a smaller contact angle than anticipated. Large drops can not be formed at the end of the needle without detaching prematurely. Therefore the drop is deposited on the surface and volume is added to obtain a large drop. The contact line of the drop could be pinning as fluid is added resulting in a larger contact angle than anticipated.

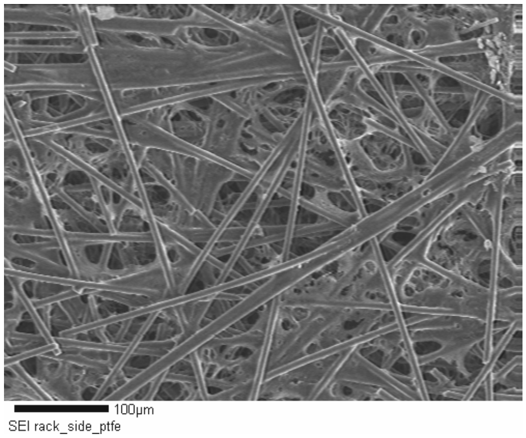
4.3 Contact Angle Measurements on GDL Surfaces

The GDL manufacturer and relative composition can be seen in Table 4.2 with SEM images of the GDLs shown in Figure (4.2). GDL 4 and GDL 6 are from the same manufacturer with the only difference being that GDL 6 has a microporous layer (MPL) coating on one side. The PTFE coating can be seen as a webbing between the individual carbon fibers in GDL 4 and GDL 6. The PTFE coating on GDL 8 is not as visible as the other GDLs and the carbon fibers appear to be longer and are intertwined in what is called a non-woven fashion. GDL 11 has a similar PTFE coating, carbon fiber size, and orientation as GDL 4 and GDL 6 but appears to have a larger average pore size. Similar contact angle results were expected for GDL 4 and GDL 6, but the contact angle appears to have a steeper slope for GDL 6 than GDL 4 when plotted against $\log(r/L_c)$. GDL 4 and GDL 8 have a similar trend in contact angle plotted against drop size when plotted as θ vs $\log(r/L_c)$. GDL 11 is the only sample to show θ decreasing for increasing drop size for a small range of drop sizes.

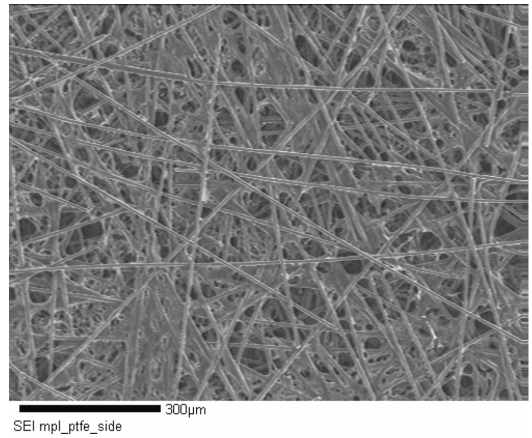
Table 4.2. GDL samples tested.

GDL 4	Mitshubishi MRC 105 9% PTFE (weight)
GDL 6	Mitshubishi MRC 105 9% PTFE (weight) with MPL
GDL 8	Freudenberg Plain H2315 nonwoven
GDL 11	Toray TFP-H-060 (7% PTFE)

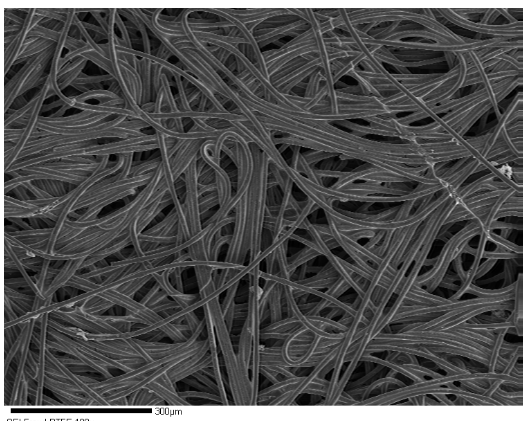
Contact angles of water on several GDLs were measured and results are shown in Figure (4.4) through (4.7). The contact angle measurements on the GDLs are higher, in the range of 130° to 170°, than the contact angles on FC-721, in the range of 100° to 120°. Although data is limited, a similar trend of θ linearly increasing when plotted as θ vs $\log(r/L_c)$ is seen for GDL 4, GDL 6, and GDL 8. The trend appears to change for GDL 11 where θ appears to decrease slightly when plotted as θ vs $\log(r/L_c)$ for a small range of drop sizes.



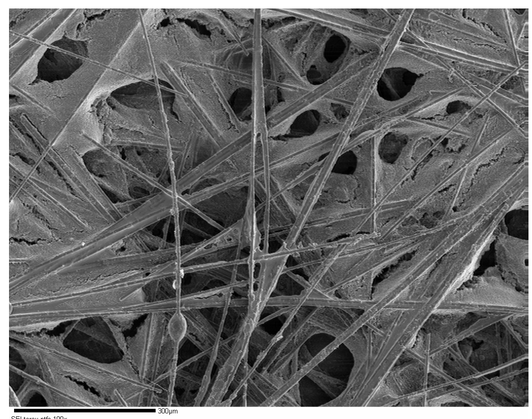
(a) GDL 4



(b) GDL 6



(c) GDL 8



(d) GDL 11

Figure 4.2. SEM images of the GDLs used for contact angle measurements [15]

4.4 Contact Angle vs Drop Size

The trend of contact angles changing with drop size on the GDLs could be explained by drop deposition and contact line pinning similar to the data obtained for FC-721, but the contact angle appears to increase with increasing drop size more rapidly on GDL 4, GDL 6, and GDL 8. GDL 11 does not show the same trend of increasing contact angle with increasing drop size as with GDL 4, GDL 6, and GDL 8. This may be explained by a number differences between the GDLs, including differences in pore size, pore size distribution, coating quality and composition, fiber size and orientation, handling protocol during production, or having a small range of drop sizes. The surface roughness and porosity of the GDLs could increase the effect of contact line pinning when compared to the contact angle data on FC-721.

Automating the drop deposition method may not eliminate the problems associated with manual drop deposition, but would significantly reduce the contact angle dependence associated with moving the contact line of a drop from shaking the needle while depositing the drop.

A few ideas for eliminating or significantly reducing the problem of pinning are to vibrate the drop to relax the contact line to its equilibrium state or add a term to the Laplace-Young equation to account for this "extra energy". Drop vibration could help relax the contact line to an equilibrium state but may be difficult to find the right frequency and time to vibrate the drop to fully relax the contact line. The idea of adding an "extra energy" term to the models used to find contact angle is not new and has been previously discussed by Drelich et al. [11].

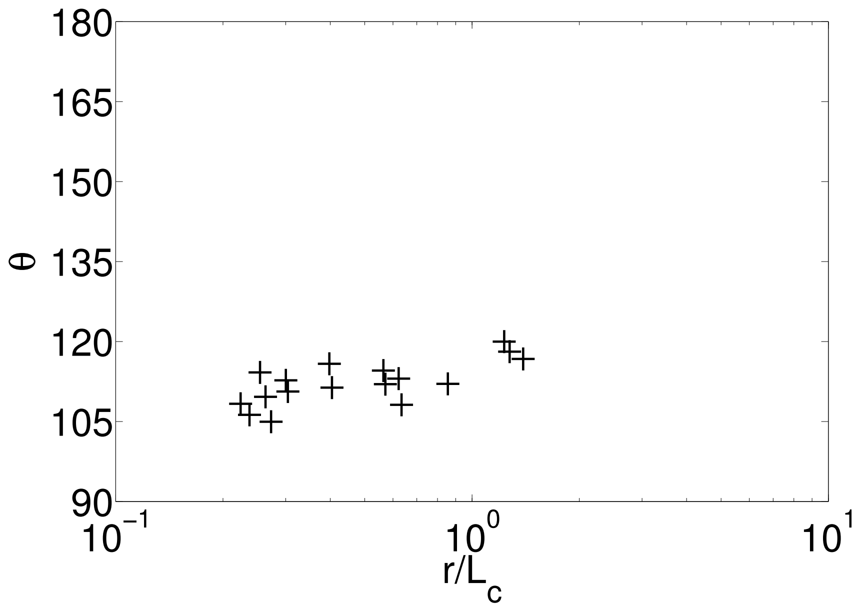


Figure 4.3. Contact angle of water on FC-721 as a function of the ratio of equator radius, r, to Laplace constant, L_c

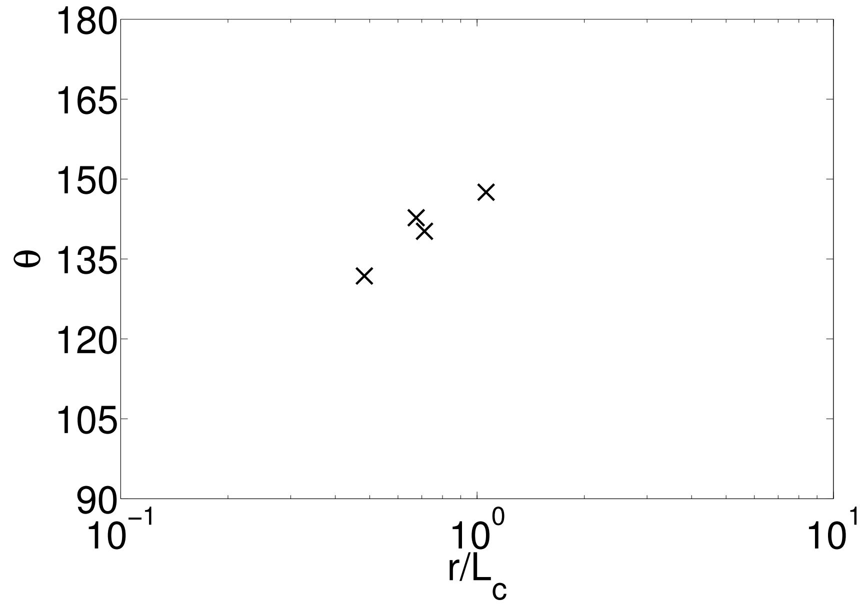


Figure 4.4. Contact angle of water on GDL 4 as a function of the ratio of equator radius, r, to Laplace constant, L_c

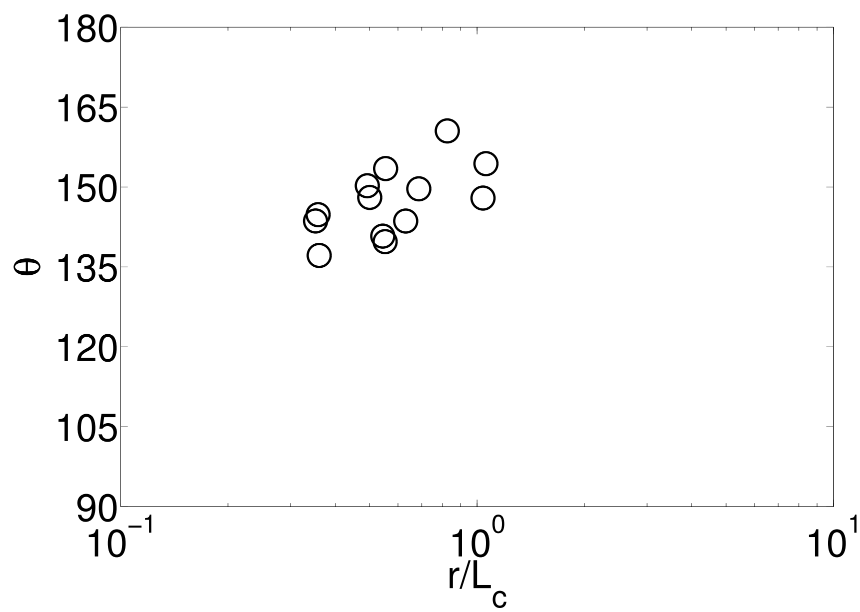


Figure 4.5. Contact angle of water on GDL 6 as a function of the ratio of equator radius, r, to Laplace constant, L_c

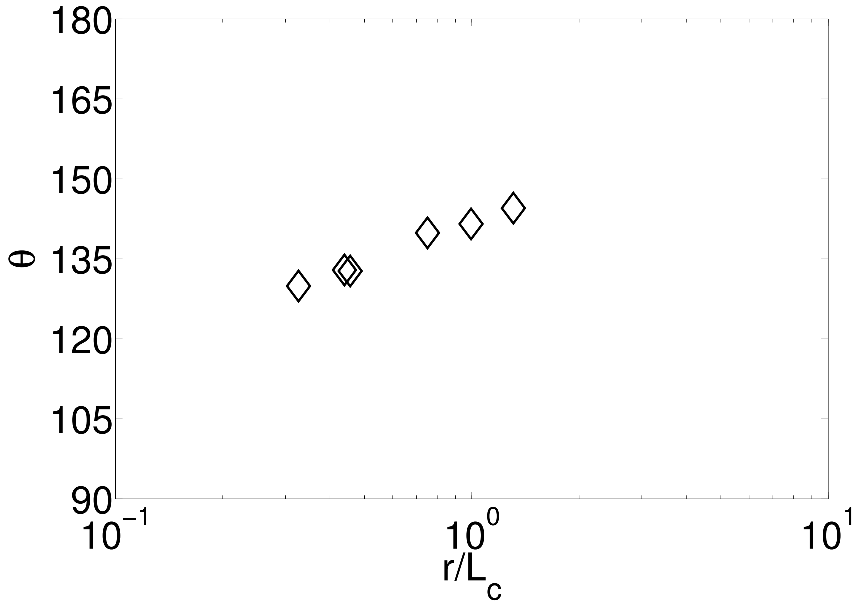


Figure 4.6. Contact angle of water on GDL 8 as a function of the ratio of equator radius, r, to Laplace constant, L_c

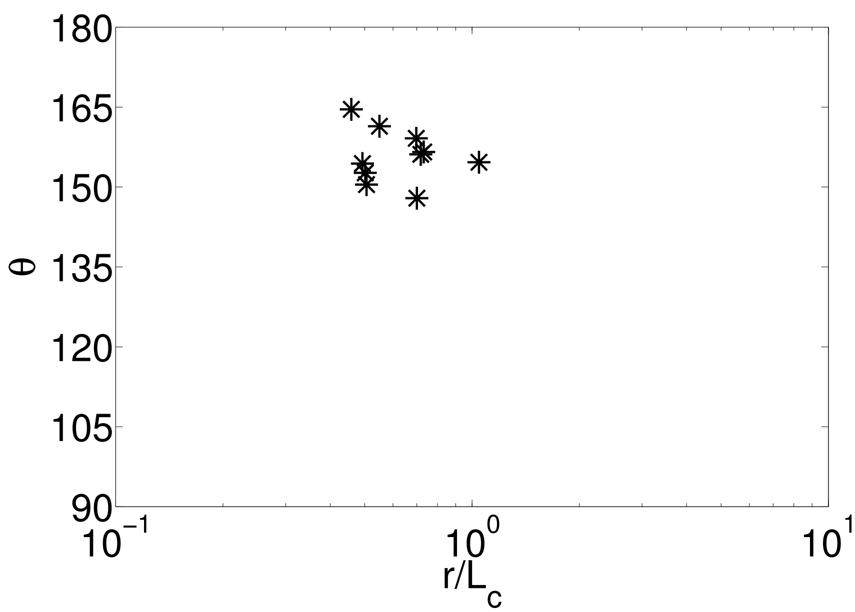


Figure 4.7. Contact angle of water on GDL 11 as a function of the ratio of equator radius, r, to Laplace constant, L_c

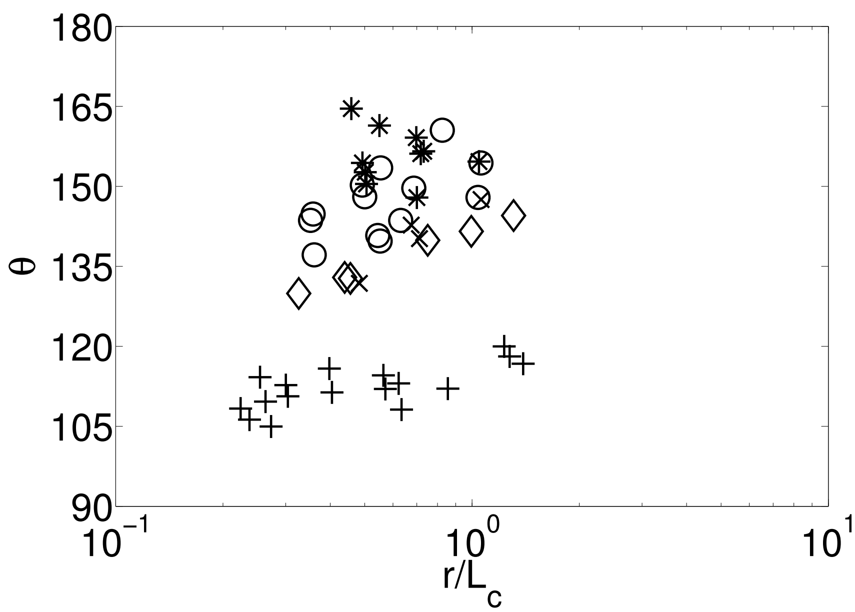


Figure 4.8. Contact angle of water on all tested substrates as a function of the ratio of equator radius, r, to Laplace constant, L_c. FC-721 (+) GDL 4 (×) GDL 6 (∘) GDL 8 (⋄) GDL 11 (∗)

5. CONCLUSION

A contact angle measurement apparatus has been designed, fabricated, and tested. The apparatus includes a backlight, enclosure to control temperature and humidity, a temperature controllable drop stage, imaging hardware including a long working distance microscope and CCD camera, and processing hardware including a framegrabber and workstation. The contact angles are determined using a MATLAB® code developed for use with rough surface. The code is based on the Laplace-Young equation. The contact angles are found after matching the correct Laplacian curve to the drop profile. This contact angle measurement apparatus, including the MATLAB® program, is suitable for making contact angle measurements of water on PEM fuel cell GDLs at various temperatures and for various drop sizes.

A limited set of data on FC-721 coated glass and four different GDLs has been collected. The results show a contact angle dependence on drop size for water on FC-721 coated glass and the four GDLs. Results are displayed showing contact angle, θ, as a function of drop size using the (r/L_c), or $\sqrt{\text{Bo}}$. In the case of the FC-721 on glass and GDL 4, GDL 6, and GDL 8 the contact angle tends to decrease as Bo decreases and increase as Bo increases. This trend is difficult to see in the GDL 11 results due to a large variance in the contact angle for low Bond numbers and the limited range of drop sizes.

The wettability of fuel cell components, namely GDLs, play a crucial role in proper water management for increased fuel cell performance and lifetime. Contact angles of drops must be measured for a range of drop sizes that are desired to obtain meaningful contact angles. The radial and height dimensions of the current set of data exceed the current dimensions of fuel cell channels, approximately 500-700 microns square. Therefore, data for drops with equatorial radii and drop heights in the range of the dimensions of current fuel cell channels is needed. The current data set provides information on the trends of contact angles compared to drop sizes but are not meaningful considering contact angles of drop sizes which may fit in current fuel cell channels without plugging the channel. The reporting of contact angle data should always include key information such as drop width, either equatorial radius or wetted radius, and drop height.

The continuation of the wettability study on GDLs will require the development of an automatic drop deposition method to significantly reduce user error from manual drop deposition. More data is needed for a larger range of Bo numbers at room temperature as well as data for temperatures between 0° and 80° Celsius to look at the effect of temperature on contact angles.

APPENDICES

A. NOMENCLATURE

ADSA	Axisymmetric Drop Shape Analysis
b	radius of curvature at apex
B	non-dimensionalized radius of curvature at apex
Bo	Bond number
c	ratio of physical properties of fluid
f_1	fractional area of surface with contact angle, θ_1
f_2	fractional area of surface with contact angle, θ_2, or fractional area of pores in material
FEM	Finite Element Method
g	gravitational acceleration
GDL	Gas Diffusion Layer
H	non-dimensionalized drop height
l	characteristic length in centimeters
l_{px}	characteristic length in pixels
l_{nd}	non-dimensionalized characteristic length scale
L_c	capillary length
MPL	Microporous Layer
p	perimeter
ΔP	change in pressure
ΔP_g	hydrostatic pressure drop
ΔP_σ	change in pressure across curved interface
PEM	Polymer Electrolyte Membrane or Proton Exchange Membrane
PTFE	polytetrafluoroethylene
r	equator radius
R	difference between wetted radius and radius at 0.05 H above the substrate surface
R_1	first primary radius of curvature
R_2	second primary radius of curvature
s	arc length of drop interface
S	non-dimensionalized arc length of drop interface
Sc	scale length in centimeters
Sc_{px}	scale length in pixels
SF	scale factor
TEC	Thermoelectric cooler

ΔW	change in weight
w_{Bo}	Uncertainty in Bo
w_c	Uncertainty in c
w_g	Uncertainty in gravitational acceleration
w_H	Uncertainty in the non-dimensionalized drop height
w_l	Uncertainty in the characteristic length
$w_{l_{px}}$	Uncertainty in the characteristic length in pixels
w_ρ	Uncertainty in density
w_{Sc}	Uncertainty in the scale
$w_{Sc_{px}}$	Uncertainty in the pixels in the scale
w_{SF}	Uncertainty in scale factor
w_σ	Uncertainty in surface tension
x	horizontal distance
X	non-dimensionalized horizontal distance
z	vertical distance
Z	non-dimensionalized vertical distance

greek symbols

ϵ_b	non-dimensionalized horizontal minimization coefficient
ρ	density
σ	surface tension
θ	contact angle
ζ	roughness factor

superscripts and subscripts

c	Cassie's contact angle
lv	liquid-vapor
sl	solid-liquid
sv	solid-vapor
w	Wenzel's contact angle

B. UNCERTAINTY ANALYSIS

An uncertainty analysis is performed for the Bond number and θ. The Bond number analysis is derived through independent analysis on c, l, and the scale factor, SF.

$$\mathrm{Bo} = \frac{\rho g l^2}{\sigma} \tag{B.1}$$

and by substituting in c,

$$\mathrm{Bo} = c l^2 \tag{B.2}$$

The uncertainty analysis on θ is performed with the assumption that $\tan(\theta) \approx (0.05H)/(R)$, where H is the non-dimensionalized drop height and R is the difference between the non-dimensionalized wetted radius and the non-dimensionalized radius at 0.05 H above the substrate surface.

B.1 Uncertainty in c

Uncertainty in c is a combination of uncertainties in the fluid properties ρ with units of [g/cm^3] and σ with units of [mN/m] as well as g with units of [cm/s^2].

$$c = \frac{\rho g}{\sigma} \tag{B.3}$$

$$w_c = \left[\left(\frac{\partial c}{\partial \rho} w_\rho \right)^2 + \left(\frac{\partial c}{\partial g} w_g \right)^2 + \left(\frac{\partial c}{\partial \sigma} w_\sigma \right)^2 \right]^{1/2} \tag{B.4}$$

$$\frac{\partial c}{\partial \rho} = \frac{g}{\sigma} \tag{B.5}$$

$$\frac{\partial c}{\partial g} = \frac{\rho}{\sigma} \tag{B.6}$$

$$\frac{\partial c}{\partial \sigma} = \frac{-\rho g}{\sigma^2} \tag{B.7}$$

Substituting (B.5 - B.7) into (B.4),

$$w_c = \left[\left(\frac{g}{\sigma} w_\rho \right)^2 + \left(\frac{\rho}{\sigma} w_g \right)^2 + \left(\frac{-\rho g}{\sigma^2} w_\sigma \right)^2 \right]^{1/2} \tag{B.8}$$

Factoring out a c from (B.8) gives,

$$\frac{w_c}{c} = \left[\left(\frac{w_\rho}{\rho}\right)^2 + \left(\frac{w_g}{g}\right)^2 + \left(\frac{w_\sigma}{\sigma}\right)^2 \right]^{1/2} \tag{B.9}$$

$$w_\rho = 0.002 \text{ [g/cm}^3\text{]} \tag{B.10}$$

$$w_g = 2 \text{ [cm/s}^2\text{]} \tag{B.11}$$

$$w_\sigma = 5 \text{ [mN/m]} \tag{B.12}$$

$$\rho = 0.998 \text{ [g/cm}^3\text{]} \tag{B.13}$$

$$g = 981.2 \text{ [cm/s}^2\text{]} \tag{B.14}$$

$$\sigma = 69 \text{ [mN/m]} \tag{B.15}$$

Substituting in the values for ρ, g, σ, w_ρ, w_g, and w_σ,

$$\frac{w_c}{c} = 0.0725 \tag{B.16}$$

B.2 Uncertainty in the scale factor, SF

In the analysis of the uncertainty of the scale factor, SF, Sc_{px} is the distance between the two points picked by the user with units of [pixels] and Sc is the distance between the two points picked by the user with units of [cm].

$$SF = \frac{Sc_{px}}{Sc} \tag{B.17}$$

$$w_{SF} = \left[\left(\frac{\partial SF}{\partial Sc_{px}} w_{Sc_{px}}\right)^2 + \left(\frac{\partial SF}{\partial Sc} w_{Sc}\right)^2 \right]^{1/2} \tag{B.18}$$

$$\frac{\partial SF}{\partial p_{SF}} = \frac{1}{Sc} \tag{B.19}$$

$$\frac{\partial SF}{\partial Sc} = \frac{-Sc_{px}}{Sc^2} \tag{B.20}$$

$$w_{SF} = \left[\left(\frac{1}{Sc} w_{Sc_{px}}\right)^2 + \left(\frac{-Sc_{px}}{Sc^2} w_{Sc}\right)^2 \right]^{1/2} \tag{B.21}$$

$$\frac{w_{SF}}{SF} = \left[\left(\frac{w_{Sc_{px}}}{Sc_{px}} \right)^2 + \left(\frac{w_{Sc}}{Sc} \right)^2 \right]^{1/2} \tag{B.22}$$

$w_{px_{Sc}}$ and w_{Sc} are fixed while Sc_{px} and w_{Sc} vary between each data point.

$$w_{Sc_{px}} = 2 \text{ [pixels]} \tag{B.23}$$

$$w_{Sc} = 0.00006 \text{ [cm]} \tag{B.24}$$

B.3 Uncertainty in the characteristic length, l

In the analysis of the uncertainty of the characteristic length scale, l, l_{px} is the number of pixels of the characteristic length and SF is the scale factor with units of [pixels/centimeters].

$$l = \frac{l_{px}}{SF} \tag{B.25}$$

$$w_l = \left[\left(\frac{\partial l}{\partial l_{px}} w_{l_{px}} \right)^2 + \left(\frac{\partial l}{\partial SF} w_{SF} \right)^2 \right]^{1/2} \tag{B.26}$$

$$\frac{\partial l}{\partial l_{px}} = \frac{1}{SF} \tag{B.27}$$

$$\frac{\partial l}{\partial SF} = \frac{-l_{px}}{SF^2} \tag{B.28}$$

$$w_l = \left[\left(\frac{1}{SF} w_{l_{px}} \right)^2 + \left(\frac{-l_{px}}{SF^2} w_{SF} \right)^2 \right]^{1/2} \tag{B.29}$$

$$\frac{w_l}{l} = \left[\left(\frac{w_{l_{px}}}{l_{px}} \right)^2 + \left(\frac{w_{SF}}{SF} \right)^2 \right]^{1/2} \tag{B.30}$$

$w_{l_{px}}$ is fixed while l_{px} varies for each data point. w_{SF}/SF has already been discussed and varies for each data point.

$$w_{l_{px}} = 2 \text{ [pixels]} \tag{B.31}$$

B.4 Uncertainty in Bo

The analysis of the Bo is a combination of c and l. The analysis of c is essentially constant for every data point because it is based on fluid properties. The analysis of l changes for every drop based on the characteristic length scale or equator radius in this case.

$$w_{\mathrm{Bo}} = \left[\left(\frac{\partial \mathrm{Bo}}{\partial c} w_c \right)^2 + \left(\frac{\partial \mathrm{Bo}}{\partial l} w_l \right)^2 \right]^{1/2} \tag{B.32}$$

$$\frac{\partial \mathrm{Bo}}{\partial c} = l^2 \tag{B.33}$$

$$\frac{\partial \mathrm{Bo}}{\partial l} = 2cl \tag{B.34}$$

$$w_{\mathrm{Bo}} = \left[\left(l^2 w_c \right)^2 + \left(2clw_l \right)^2 \right]^{1/2} \tag{B.35}$$

$$\frac{w_{\mathrm{Bo}}}{\mathrm{Bo}} = \left[\left(\frac{w_c}{c} \right)^2 + \left(2\frac{w_l}{l} \right)^2 \right]^{1/2} \tag{B.36}$$

B.5 Uncertainty in contact angle, θ

The analysis of θ is performed assuming $\tan(\theta) \approx (0.05H)/(R)$. Looking at the maximum θ in this data set, 165°, R can be approximated as 0.19 H.

$$\tan(\theta) = \frac{0.05H}{R} \tag{B.37}$$

$$\theta = \tan^{-1}\left(\frac{0.05H}{R} \right) \tag{B.38}$$

Using the expansion for $\tan^{-1}$, the uncertainty in θ is calculated as;

$$\frac{w_\theta}{\theta} = \left[\left(\frac{w_H}{H} \right)^2 + \left(\frac{w_R}{R} \right)^2 \right]^{1/2} \tag{B.39}$$

Substituting in for R and w_R, where is approximately 2 ϵ_b;

$$\frac{w_\theta}{\theta} = \left[\left(\frac{w_H}{H} \right)^2 + \left(\frac{2\epsilon_b}{.19H} \right)^2 \right]^{1/2} \tag{B.40}$$

ϵ_b is the non-dimensionalized horizontal minimization coefficient. The drop height is known within ± 5 pixels, therefore w_H in non-dimensionalized form is;

$$w_H = \frac{5}{SF\ c^{1/2}} \tag{B.41}$$

The uncertainty in θ is presented in the test matrix.

C. TEST MATRIX

The test matrix presents the data collected for this presentation. Data is presented in order of test date and substrate. The contact angle, θ, is found from the solution to the Laplace-Young equation which produces a Laplacian curve which matches the drop interface and is reported in degrees (°). Drop height and equator radius are found from the original drop image and are reported in units of centimeters. The Bond number, Bo, is calculated using the final c value and the characteristic length scale, in this case the equator radius. The values of c and b are final values after running the program and have units of [cm^{-2}] and [cm], respectively. The c^* values represent final c values when c was iterated on while solving the Laplace-Young equation. Remaining c values were not iterated on while solving the Laplace-Young equation and are based on a surface tension of 70 [mN/m], density of water 0.998 [$\mathrm{g/cm^3}$], density of air of 0.0012 [$\mathrm{g/cm^3}$], and acceleration due to gravity of 981.7 [$\mathrm{cm/s^2}$].

Date	Substrate	θ	Drop Height	Equator Radius	Bo	r/L_c	c	b	Comments
02/04	GDL 8	129.9 ± 1.3	0.159	0.102	0.11	0.33	10.27*	0.104	Fiber protruding from the surface on the right and poor imaging of the substrate surface.
02/04	GDL 8	139.9 ± 0.8	0.286	0.204	0.56	0.75	13.57*	0.223	
02/04	GDL 8	132.7 ± 1.2	0.178	0.117	0.21	0.46	15.17*	0.121	Some fibers protruding from the surface to the right of the drop.
02/04	GDL 8	132.9 ± 1.3	0.167	0.109	0.19	0.44	16.17*	0.113	Drop not centered horizontally in image, not much substrate surface information on the left of the drop
02/04	GDL 11	152.7 ± 1.2	0.210	0.128	0.25	0.50	15.47*	0.134	
02/04	GDL 11	150.5 ± 1.1	0.218	0.134	0.26	0.51	14.27*	0.140	
02/04	GDL 11	156.1 ± 0.9	0.291	0.194	0.52	0.72	13.67*	0.210	
02/04	GDL 11	154.4 ± 1.1	0.216	0.130	0.24	0.49	14.27*	0.136	
02/10	GDL 6	148.1 ± 1.2	0.199	0.123	0.25	0.50	16.57*	0.128	
02/10	GDL 6	139.8 ± 1.1	0.201	0.132	0.31	0.55	17.57*	0.138	Substrate or camera may not be completely level.
02/10	GDL 6	143.6 ± 0.9	0.246	0.164	0.40	0.63	14.77*	0.175	Substrate or camera may not be completely level.
02/10	GDL 6	140.8 ± 1.0	0.217	0.141	0.30	0.54	14.97*	0.149	
02/15	GDL 4	140.2 ± 1.4	0.271	0.190	0.51	0.71	14.07*	0.207	
02/15	GDL 4	142.7 ± 1.5	0.254	0.173	0.46	0.68	15.27*	0.187	Small fiber protruding from surface on the right of the drop.
02/15	GDL 4	131.8 ± 2.0	0.182	0.121	0.23	0.48	15.97*	0.126	
02/15	GDL 4	147.5 ± 1.1	0.358	0.285	1.12	1.06	13.87*	0.336	
02/15	GDL 4	78.4 ± 1.1	0.192	0.132	0.24	0.49	13.97*	0.322	Drop placed too close to edge where fibers and coating disturbed by cutting process, possibly destroyed the non-wetting properties. Data not used.

Date	Substrate	θ	Drop Height	Equator Radius	Bo	r/L_c	c	b	Comments
02/15	GDL 6	160.5 ± 1.3	0.321	0.222	0.68	0.83	13.77*	0.248	
02/15	GDL 6	153.5 ± 1.7	0.240	0.149	0.31	0.55	13.77*	0.157	
02/15	GDL 6	149.7 ± 1.4	0.281	0.188	0.47	0.69	13.37*	0.203	
02/15	GDL 6	150.2 ± 2.0	0.207	0.126	0.24	0.49	15.27*	0.132	
02/15	GDL 6	154.4 ± 1.1	0.368	0.285	1.12	1.06	13.77*	0.341	
02/15	GDL 8	141.6 ± 1.1	0.350	0.276	0.99	1.00	12.97*	0.320	
02/15	GDL 8	144.5 ± 1.0	0.404	0.356	1.71	1.31	13.47*	0.461	Large drop, not much surface on both sides to find an average surface location. Looks like small fiber on the right side of the drop by the base.
02/15	GDL 11	156.6 ± 1.4	0.300	0.201	0.54	0.73	13.27*	0.219	
02/15	GDL 11	164.6 ± 2.2	0.201	0.117	0.21	0.46	15.47*	0.121	
02/15	GDL 11	159.1 ± 1.5	0.285	0.187	0.49	0.70	13.97*	0.202	
02/15	GDL 11	154.7 ± 1.1	0.367	0.284	1.09	1.05	13.57*	0.335	Substrate surface does not look level.
02/15	GDL 11	161.4 ± 1.9	0.230	0.140	0.30	0.55	15.37*	0.148	Substrate surface does not look level.
02/15	GDL 11	147.9 ± 1.4	0.277	0.187	0.49	0.70	14.07*	0.203	Substrate surface does not look level.
03/02	FC-721	108.1 ± 1.2	0.195	0.168	0.40	0.63	14.27	0.179	
03/02	FC-721	111.4 ± 1.8	0.137	0.107	0.16	0.41	14.27	0.110	
03/02	FC-721	113.1 ± 1.2	0.202	0.165	0.18	0.42	14.27	0.175	
03/04	FC-721	114.2 ± 2.7	0.092	0.067	0.07	0.25	14.27	0.068	Contact line region is distorted, possibly from lighting source.
03/04	FC-721	115.8 ± 1.8	0.143	0.105	0.16	0.40	14.27	0.109	Contact line region is distorted, possibly from lighting source.
03/04	FC-721	112.7 ± 2.3	0.107	0.080	0.09	0.30	14.27	0.081	Some reflection on the left and right of the drop image, substrate may not be level.
03/04	FC-721	112.1 ± 1.0	0.253	0.227	0.73	0.86	14.27	0.253	Some reflection on the left of the drop image, substrate may not be level.

Date	Substrate	θ	Drop Height	Equator Radius	Bo	r/L_c	c	b	Comments
03/04	GDL 6	144.8 ± 2.2	0.143	0.084	0.13	0.36	18.17*	0.087	Spots on image, lens appears to be dirty.
03/04	GDL 6	143.6 ± 2.4	0.133	0.079	0.12	0.35	19.67*	0.081	Spots on image, lens appears to be dirty.
03/04	GDL 6	137.2 ± 1.9	0.155	0.096	0.13	0.36	14.17*	0.098	Spots on image, lens appears to be dirty.
03/04	GDL 6	147.9 ± 0.9	0.358	0.281	1.08	1.04	13.67*	0.330	Spots on image, lens appears to be dirty. Large drop, not much surface on both sides to find an average surface location.
03/09	FC-721	105.0 ± 1.9	0.089	0.072	0.07	0.27	14.27	0.073	
03/09	FC-721	114.6 ± 1.0	0.189	0.149	0.32	0.56	14.27	0.156	
03/09	FC-721	108.3 ± 2.3	0.077	0.059	0.05	0.22	14.27	0.060	
03/09	FC-721	106.3 ± 2.2	0.079	0.063	0.06	0.24	14.27	0.064	
03/09	FC-721	110.6 ± 1.7	0.105	0.081	0.09	0.30	14.27	0.082	
03/09	FC-721	109.6 ± 1.9	0.092	0.070	0.07	0.26	14.27	0.071	
03/09	FC-721	112.0 ± 1.0	0.187	0.151	0.33	0.57	14.27	0.160	
03/12	FC-721	120.0 ± 1.0	0.338	0.326	0.41	0.64	14.27	0.408	
03/12	FC-721	118.1 ± 0.9	0.339	0.338	0.43	0.65	14.27	0.427	
03/12	FC-721	116.8 ± 0.9	0.352	0.369	1.94	1.39	14.27	0.486	Large drop, not much surface on both sides to find an average surface location.

D. EXPERIMENTAL APPARATUS

D.1 Enclosure

In the beginning of this research, evaporation of the drops placed on the substrates proved to be a problem. An enclosure built for this research improves control over humidity and temperature in the immediate vicinity of the drop and also houses the drop stage (appendix D.5). Humidity within the enclosure is passively controlled by allowing water filled containers to evaporate over a period of a few hours to overnight depending on the relative humidity in the room. The walls of the enclosure are made of polycarbonate which allows imaging and backlighting through the enclosure. The entire enclosure is mounted on an x-y translation stage (Velmex AXY2509W1). The enclosure measures 11 inches high $\times$ 11 inches wide $\times$ 11 inches deep.

D.2 Imaging

A 10-bit, monochrome, progressive scan, 2/3 inch CCD camera (Pulnix TM-1325CL) with 1392$\times$1040 resolution which can capture up to 30fps is used to capture images and video. This cameara is mounted to a long working distance microscope (Infinity K2/S) which has a field of view ranging from 7.5mm $\times$ 5.6mm at a working distance of 214mm to 1.9mm $\times$ 1.4 mm at a working distance of 140mm. The camera is connected to a framegrabber (EPIX EL1DB) through a cameralink connection and the framegrabber is mounted in the workstation (appendix D.3).

D.3 Workstation

An IBM Intellistation Z Pro (6223-7BU) workstation is used to control the camera for capturing images using a framegrabber (EPIX EL1DB) through a cameralink connection. The contact angle measurement program is written in MATLAB®, which is loaded on the workstation. The computer has a 3.8GHz processor with 2MB L2 Cache, ability to be upgraded to a dual processor, 2 $\times$ 1024MB PC2-3200 DDR2 RAM upgraded to 2 $\times$ 2048MB PC2-4200 DDR2 RAM, 160GB HDD, DVD/CD-RW drive, gigabit ethernet, and Matrox Millenium P690 PCI. A portion of the RAM is dedicated to image and video capturing. The video card was upgraded to open the PCI-Express slot which was needed for the EPIX framegrabber and allows two DVI monitors to be used. The workstation originally used Windows XP 64 bit, however a lack of 32 bit software compatibility required a change to Windows XP 32 bit.

D.4 Köhler Illumination

Köhler Illumination is a backlight source that provides equal intensity, collimated light. This backlight source is used because it provides a crisp, clear image. Köhler Illumination is typically used in chemical or biological fields in a vertical configuration. For this research, Köhler Illumination is created using a series of lenses and apertures in a horizontal configuration. A projector is used as the light source (D.1).

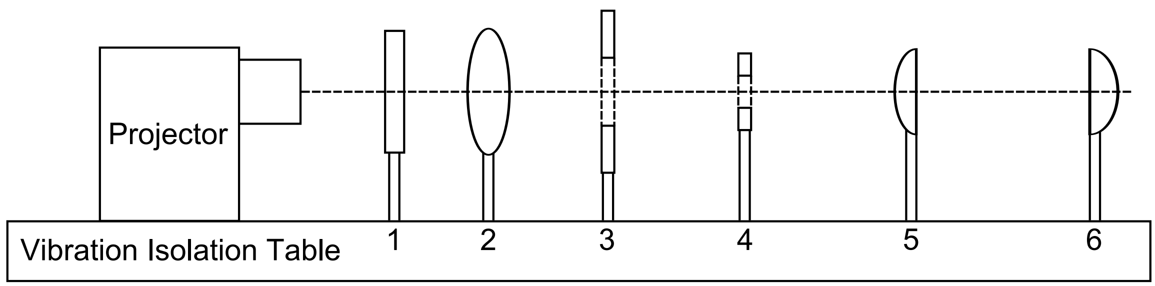


Figure D.1. Köhler Illumination setup

The projector uses a 300 watt light bulb. Lens (1) is a diffuser which scatters the light to eliminate the problem of focusing the filament of the bulb . Lens (2) is a biconvex lens. Positions (3) and (4) are apertures with variable openings of size 1.5 millimeters to 25 millimeters and 1 millimeter to 11 millimeters, respectively. The apertures block the light that comes through edges of the lens where spherical aberrations distort the light. Lenses (5) and (6) have focal lengths of 38.1 millimeters and 75.6 millimeters, respectively. The combination of the lenses and aperatures provide collimated, equal intensity light and can be adjusted to change the light diameter and intensity.

D.5 Stage

The drop stage consists of several components. The top of the stage is a copper block which allows for even heating when doing temperature dependent studies. There are four thermal electric coolers (Marlow Industries DT12-6-01L), TECs, mounted between the copper block and the aluminum fin arrangement. The TECs can be heated or cooled by applying a voltage across them. The aluminum fin arrangement acts as a heat sink to add heat to the TECs when heating the copper block and a remove heat from the TECs when cooling the copper block. The entire drop stage is mounted to a lab jack (THORLABS L200) which allows the height of the drop stage to be adjusted.

E. CONTACT ANGLE MEASUREMENT PROGRAMS

E.1 Contact Angle Program for varying c and b simultaneously

```
% Program written by Russ Stacy
%
% 11_12_08
%
% Based off of contact angle program written by Derek Fultz and
% Russ Stacy
% ca_4.m
%
% Fourth version of the Contact Angle program
%
% CA_4 finds contact angle, drop height, drop diameter at wetted
% area, drop diameter at equator, bond number, and curvature at
% apex while allowing c to vary and find theta and sigma
% simultaneously. This should only be used for relatively large
% drops.
%
% Must run external scale_1.m file to obtain a scale factor for
% this program.
%
% Type help functionname for help with indidual functions used in
% this program.

%% clear and close all opened material

clear
clc
close

%% Scale

% Run scale_1.m to create sc.mat containing the scale factor for
% the image to be processed

load sc.mat;

%% Preload Variables
```

```
S_span=(0:.001:8); % S_span is the step variable for ode45 solver

eps_b=0.000005;    % error for b values
eps_c=0.000005;    % error for c values

ii=1; % counter for error_b loop to passthrough if eps_b limit is not
     met
jj=1; % counter for error_c loop to passthrough if eps_c limit is not
     met

% Predefine error_c and d_cs as row vector for minimazation loop
error_c=zeros(1,10);
d_cs=zeros(1,10);

%% User Inputs (added 01_23_2009_restacy)

[time,substrate,fluid,temperature,excel]=user_input;

%% Fluid Properties

[rho_liquid,rho_air,surface_tension,grav,c,laplace_constant]=
   fluid_prop;

%% Image import

[I,IC,folder,file]=image_input;

%% Edge detection with original image

[E,BW,J]=image_analysis(IC);

%% Boundary Tracing

[boundary]=edge(BW,J);

%% Drop Apex

[apex_x,apex_z]=apex(boundary);

%% Data transform and scaling

[x_dat,z_dat,x_dat_cm,z_dat_cm,x_dat_nd,z_dat_nd]=trans_scale(apex_x,
   apex_z,boundary,scale,c);

%%% VARIABLES CHANGED ON 12/9/08
% x_plot_px to x_dat
% x_plot_cm to x_dat_cm
% x_nd to x_dat_nd
% y_plot_px to z_dat
% y_plot_cm to z_dat_cm
% y_nd to z_dat_nd
```

```
% y values changed to z to reflect z-plane notation

%% Z-Plane

[z_plane,z_plane_cm,z_plane_nd,ind_l,ind_r]=plane(boundary,z_dat,
    apex_z,scale,c);

%% Drop Height and Drop Interface(minus z-plane data)

[drop_height,drop_height_cm,drop_height_nd,x_dat_cut,z_dat_cut,
    x_dc_cm,z_dc_cm,x_dc_nd,z_dc_nd,equator_radius,bond,x_equator_left
    ,x_equator_right,z_equator_left,z_equator_right]=height_interface(
    z_plane,scale,c,z_dat,x_dat,ind_l,ind_r);

%% Spherical Cap approximation data

[x_center,z_center,sphere_radius]=sphere_cap(x_equator_left,
    z_equator_left,x_equator_right,z_equator_right);

%% Circle function

[X_circle,Z_circle]=circle([x_center z_center],sphere_radius,1000,'r.
    ');

%% Contact Angle with Spherical Cap Approximation

[theta_sc]=contact_angle_sphere_cap(x_center,z_center,X_circle,
    Z_circle,drop_height);

%% C Cutoff Data

[cut_c]=c_data(drop_height_nd,z_dat_nd);

%% Laplace-Young Equation

% Derek Fultz and Russ Stacy
% 8_10_07

% This program solves 3 simultaneous differential equations. One
% is the Laplace-Young equation, the other two are geometric
% relationships.

jjj=1;
LYsoln=0;
while (LYsoln==0);

b1=5;
b2=.01;

bb=[b1 b2]; % range of b values to use
```

```
% This is a large loop that solves the Laplace-Young equation for
% several values of b to find the solution.

for j=1:2;

b=bb(j); % set b as individual values of bb

[S,Y] = ode45(@laplace,S_span,[0 1e-100 0],[],b,c); % may need to
    increase second input if curve does not loop

Z=Y(:,1); % the first column of Y is the Z data. Needs to be defined
    to find cutoff point below.

% This loop stores the value, i, where the Laplace-Young data
% starts to 'loop' out of control.

i=1;
while Z(i)<Z(i+1); % finds where contact angle is 180 degrees (cutoff
     point)
    i=i+1;
end;

ll(j)=i;

% Output Variables

x_ly(1:i,j)=Y(1:i,2);            % Drop height dimensionless
z_ly(1:i,j)=Y(1:i,1);            % Drop x dimension dimensionless
S_ly(1:i,j)=S(1:i,1);            % Total arc length dimensionless
Phi_ly(1:i,j)=Y(1:i,3)*(180/pi); % Contact angle in degrees

% Defines laplace curves for both sides of apex.
%
% x_plot_lap_full=[-x_plot_lap;x_plot_lap];
% y_plot_lap_full=[y_plot_lap;y_plot_lap];

end;

%% Error

% If the data at the apex fits and the data at the end of the
% trim data fits and all of the fluid properties are correct,
% solution should be found. This allows error to only be based on
% the last point of the trim data.

% ERROR FOR B

[d1,z1]=min(abs(z_dc_nd(end)-z_ly(:,1))); % find indice of laplace
    data that closely matches raw data
error_b1=x_dc_nd(end)-x_ly(z1,1);          % finds relative horizontal
     error of end of trim data
```

```

[d2,z2]=min(abs(z_dc_nd(end)-z_ly(:,2))); % find indice of laplace data that closely matches raw data
error_b2=x_dc_nd(end)-x_ly(z2,2);         % finds relative horizontal error of end of trim data

if error_b1*error_b2>0 && error_b1<0;
    fprintf('choose smaller b2 value');
elseif error_b1*error_b2>0 && error_b1>0;
    fprintf('choose larger b1 value');
    return
end;

error_b=min(abs([error_b1 error_b2]));

%% Minimization

c_change=.1;
b=(b1+b2)/2;

    b3=b;
        while (abs(error_b)>eps_b);
            clear x_ly z_ly S_ly Phi_ly Z
%             fprintf('ode call\n') % debugging tag
            [S,Y] = ode45(@laplace,S_span,[0 1e-100 0],[],b3,c); % may need to increase span of S if curve does not loop
%             fprintf('out of ode call\n') % debugging tag

            Z=Y(:,1); % the first column of Y is the Z data. Needs to be defined to find cutoff point below.

            % This loop stores the value, i, where the
            % Laplace-Young data starts to 'loop' out of control.

            i=1;
            while Z(i)<Z(i+1); % finds where contact angle is 180 degrees (cutoff point)
               i=i+1;
            end;

            % Output Variables

            x_ly_1=zeros(i,1);   % define as column vector
            z_ly_1=zeros(i,1);   % define as column vector
            S_ly_1=zeros(i,1);   % define as column vector
            Phi_ly_1=zeros(i,1); % define as column vector

            x_ly_1(1:i)=Y(1:i,2);             % Drop height dimensionless
            z_ly_1(1:i)=Y(1:i,1);             % Drop width dimensionless
```

```
            S_ly_1(1:i)=S(1:i,1);             % Arc length dimensionless
            Phi_ly_1(1:i)=Y(1:i,3)*(180/pi); % Contact angle in degrees

            [d3,z3]=min(abs(z_dc_nd(end)-z_ly_1)); % find indice of laplace data that closely matches raw data
            error_b3=x_dc_nd(end)-x_ly_1(z3);       % finds relative horizontal error of end of trim data

            if error_b3*error_b1>0;
                b1=b3;
                error_b1=error_b3;
                error_b=error_b1;
            else
                b2=b3;
                error_b2=error_b3;
                error_b=error_b2;
            end;
            b=b3;
            b3=(b1+b2)/2;

            % If eps_b limit is never met, passthrough when b=b3
            % for 10 iterations. r_b is b rounded to 15 decimal
            % places, r_b3 is b3 rounded to 15 decimal places.
            % d_rs is the difference between r_b and r_b3. When
            % d_rs=0, r_b=r_b3 and when this happens 10 times, a
            % minimum has been reached even if it isn't within
            % eps_b.

            r_b=round2(b,15);
            r_b3=round2(b3,15);

            d_rs=r_b-r_b3;

            if d_rs==0;
                ii=ii+1;
            end;

            if ii==10;
                eps_b=abs(error_b);
                ii=1;
            end;

            fprintf('b=%15.15f error_b=%15.15f d_rs=%f ii=%f eps_b=%f \n',b,error_b,d_rs,ii,eps_b);

        end;

    % Reset eps_b back to original value
    eps_b=.000005;
```

```
% cut_c is predefined in function (c_data). z_ly comes from the
% solution to the LY equations using b3. error_c is the
% horizontal distance between the actual drop interface and the
% LY curve at the point point cut_c.

    [dc,zc]=min(abs(z_dat_nd(cut_c)-z_ly_1(:,1))); % find indice of
        laplace data that closely matches raw data
    error_c=x_dat_nd(cut_c)-x_ly_1(zc,1);          % finds relative
        horizontal error of end of trim data

    if abs(error_c)>eps_c;
        sign=error_c/abs(error_c);
        if sign==1;
            c=c+(c_change);
        else
            c=c-(c_change);
        end;

        [drop_height,drop_height_cm,drop_height_nd,x_dat_cut,
            z_dat_cut,x_dc_cm,z_dc_cm,x_dc_nd,z_dc_nd,equator_radius,
            bond,x_equator_left,x_equator_right,z_equator_left,
            z_equator_right]=height_interface(z_plane,scale,c,z_dat,
            x_dat,ind_l,ind_r);
        [x_dat_nd,z_dat_nd,x_dat_cm,z_dat_cm]=rend(x_dat,z_dat,scale,
            c);

        fprintf('c=%f\nerror_c=%f\n',c,error_c)

        [S,Y] = ode45(@laplace,S_span,[0 1e-100 0],[],b,c); % may
            need to increase span of S if curve does not loop

        Z=Y(:,1); % the first column of Y is the Z data. Needs to be
            defined to find cutoff point below.

        % This loop stores the value, i, where the Laplace-Young
        % data starts to starts to 'loop' out of control.

        i=1;
        while Z(i)<Z(i+1); % finds where contact angle is 180 degrees
             (cutoff point)
           i=i+1;
        end;

        % Output Variables

        x_ly_1=zeros(i,1);   % define as column vector
        z_ly_1=zeros(i,1);   % define as column vector
        S_ly_1=zeros(i,1);   % define as column vector
        Phi_ly_1=zeros(i,1); % define as column vector
```

```
        x_ly_1(1:i)=Y(1:i,2);             % Drop height dimensionless
        z_ly_1(1:i)=Y(1:i,1);             % Drop width dimensionless
        S_ly_1(1:i)=S(1:i,1);             % Arc length dimensionless
        Phi_ly_1(1:i)=Y(1:i,3)*(180/pi); % Contact angle in degrees

        [d3,z3]=min(abs(z_dc_nd(end)-z_ly_1)); % find indice of
            laplace data that closely matches raw data
        error_b=x_dc_nd(end)-x_ly_1(z3);      % finds relative
            horizontal error of end of trim data

    else
        LYsoln=1;
    end;

end;
b=b3;

%% Figures

% Define lines to be used in the plots
z_data_c_cutoff=[z_dat_nd(cut_c);z_dat_nd(cut_c)];
z_data_cutoff=[z_dc_nd(end);z_dc_nd(end)];
x_data_cutoff=[-4;4];
z_plane_1=[drop_height_nd;drop_height_nd];
X_circle_nd=(X_circle/scale)*c^.5;
Z_circle_nd=(Z_circle/scale)*c^.5;

% Figure with Laplace-Young data, c-minimization plane,
% b-minimization plane, Spherical approximation, Calculated
% z_plane, Drop interface data from image. Comment/Uncomment
% any data set you do not want to plot and take out corresponding
% legend value.
figure;
% Laplace-Young data
plot(x_ly_1(:,1),z_ly_1(:,1),'linewidth',2);
hold all
% c minimization plane
plot(x_data_cutoff,z_data_c_cutoff)
% b minimization plane
plot(x_data_cutoff,z_data_cutoff)
% Spherical approximation
plot(X_circle_nd,Z_circle_nd,'linewidth',2)
% Calculated z_plane
plot(x_data_cutoff,z_plane_1,'linewidth',2)
% Drop interface data from image
plot(x_dat_nd,z_dat_nd,'linewidth',2)
% Change axis limits and pbaspect to fit specific drop
axis([-5 5 -.1 4.9]);
set(gca,'YDir','reverse');
pbaspect([2 1 1]);
xlabel('X-Direction');
```

```
ylabel('Z-Direction');
legend('Laplace-Young data','c minimization plane','b minimization
   plane','Spherical approximation','calculated z-plane','Raw Data')
title('Non-dimensionalized data comparison')
hold off

% Following figures used for check if problems arise

%%% Figure of raw data reset to apex at origin

% figure
% plot(x_dat,y_dat,'.')
% set(gca,'ydir','reverse')
% axis([-800 800 0 800])
% pbaspect([2 1 1])

%%% Figure of trimmed data with apex at origin

% plot(x_dat_cut,y_dat_cut,'.')
% set(gca,'ydir','reverse')
% axis([-800 800 0 800])
% pbaspect([2 1 1])

%% Find Contact Angle

dif=z_ly_1-drop_height_nd;
adif=abs(dif);
low=min(adif);
index=find(adif==low);
contact_angle=Phi_ly_1(index);

%% Data Save

[s]=dat_xls(bond,b,error_b,c,error_c,contact_angle,theta_sc,
   equator_radius,drop_height_cm,time,substrate,fluid,temperature,
   excel,scale,file,folder,eps_c);

% eof
```

E.2 Contact Angle Program for varying b

```
% Program written by Russ Stacy
%
% 11_12_08
%
% Based off of contact angle program written by Derek Fultz and
% Russ Stacy ca_fixed.m
%
% Fixed c value version of the Contact Angle program
```

```
%
% CA_FIXED finds contact angle, drop height, drop diameter at
% wetted area, drop diameter at equator, bond number, and
% curvature at apex using a fixed c value.
%
% Must run external scale_1.m file to obtain a scale factor for
% this program.
%
% Type help functionname for help with indidual functions used in
% this program.

%% clear and close all opened material

clear
clc
close


%% Scale

% Run scale_1.m to create sc.mat containing the scale factor for
% the image to be processed

load sc.mat;

%% Preload Variables

S_span=(0:.001:8); % S_span is the step variable for ode45 solver

eps_b=0.000005;     % error for b values
eps_c=0.000005;     % error for c values

ii=1; % counter for error_b loop to passthrough if eps_b limit is not
    met
jj=1; % counter for error_c loop to passthrough if eps_c limit is not
    met

% Predefine error_c and d_cs as row vector for minimazation loop
error_c=zeros(1,10);
d_cs=zeros(1,10);

%% User Inputs (added 01_23_2009_restacy)

[time,substrate,fluid,temperature,excel]=user_input;

%% Fluid Properties

c=14.11914462;
laplace_constant=1/c^.5;

%% Image import
```

```
[I,IC,folder,file]=image_input;

%% Edge detection with original image

[E,BW,J]=image_analysis(IC);

%% Boundary Tracing

[boundary]=edge(BW,J);

%% Drop Apex

[apex_x,apex_z]=apex(boundary);

%% Data transform and scaling

[x_dat,z_dat,x_dat_cm,z_dat_cm,x_dat_nd,z_dat_nd]=trans_scale(apex_x,
    apex_z,boundary,scale,c);

%%% VARIABLES CHANGED ON 12/9/08
% x_plot_px to x_dat
% x_plot_cm to x_dat_cm
% x_nd to x_dat_nd
% y_plot_px to z_dat
% y_plot_cm to z_dat_cm
% y_nd to z_dat_nd
% y values changed to z to reflect z-plane notation

%% Z-Plane

[z_plane,z_plane_cm,z_plane_nd,ind_l,ind_r]=plane(boundary,z_dat,
    apex_z,scale,c);

%% Drop Height and Drop Interface(minus z-plane data)

[drop_height,drop_height_cm,drop_height_nd,x_dat_cut,z_dat_cut,
    x_dc_cm,z_dc_cm,x_dc_nd,z_dc_nd,equator_radius,bond,x_equator_left
    ,x_equator_right,z_equator_left,z_equator_right]=height_interface(
    z_plane,scale,c,z_dat,x_dat,ind_l,ind_r);

%% Spherical Cap approximation data

[x_center,z_center,sphere_radius]=sphere_cap(x_equator_left,
    z_equator_left,x_equator_right,z_equator_right);

%% Circle function

[X_circle,Z_circle]=circle([x_center z_center],sphere_radius,1000,'r.
    ');
```

```
%% Contact Angle with Spherical Cap Approximation

[theta_sc]=contact_angle_sphere_cap(x_center,z_center,X_circle,
    Z_circle,drop_height);

%% Laplace-Young Equation

% Derek Fultz and Russ Stacy
% 8_10_07

% This program solves 3 simultaneous differential equations. One
% is the Laplace-Young equation, the other two are geometric
% relationships.

jjj=1;

b1=5;
b2=.01;

bb=[b1 b2]; % range of b values to use

% This is a large loop that solves the Laplace-Young equation for
% several values of b to find the solution.

for j=1:2;

b=bb(j); % set b as individual values of bb

[S,Y] = ode45(@laplace,S_span,[0 1e-100 0],[],b,c); % may need to
    increase second input if curve does not loop

Z=Y(:,1); % the first column of Y is the Z data. Needs to be defined
    to find cutoff point below.

% This loop stores the value, i, where the Laplace-Young data
% starts to 'loop' out of control.

i=1;
while Z(i)<Z(i+1); % find where contact angle is 180 degrees (cutoff
    point)
     i=i+1;
end;

ll(j)=i;

% Output Variables

x_ly(1:i,j)=Y(1:i,2);            % Drop height dimensionless
z_ly(1:i,j)=Y(1:i,1);            % Drop x dimension dimensionless
S_ly(1:i,j)=S(1:i,1);            % arc length dimensionless
Phi_ly(1:i,j)=Y(1:i,3)*(180/pi); % contact angle in degrees
```

```
% Defines laplace curves for both sides of apex.
%
% x_plot_lap_full=[-x_plot_lap;x_plot_lap];
% y_plot_lap_full=[y_plot_lap;y_plot_lap];
end;

%% Error

% If the data at the apex fits and the data at the end of the
% trim data fits and all of the fluid properties are correct,
% solution should be found. This allows error to only be based on
% the last point of the trim data.

% ERROR FOR B

[d1,z1]=min(abs(z_dc_nd(end)-z_ly(:,1))); % find indice of laplace
    data that closely matches raw data
error_b1=x_dc_nd(end)-x_ly(z1,1);         % finds relative horizontal
     error of end of trim data

[d2,z2]=min(abs(z_dc_nd(end)-z_ly(:,2))); % find indice of laplace
    data that closely matches raw data
error_b2=x_dc_nd(end)-x_ly(z2,2);         % finds relative horizontal
     error of end of trim data

if error_b1*error_b2>0 && error_b1<0;
    fprintf('choose smaller b2 value');
elseif error_b1*error_b2>0 && error_b1>0;
    fprintf('choose larger b1 value');
    return
end;

error_b=min(abs([error_b1 error_b2]));

%% Minimization

% c_change=.1; % not used in fixed c program
b=(b1+b2)/2;

    b3=b;
        while (abs(error_b)>eps_b);
            clear x_ly z_ly S_ly Phi_ly Z
%             fprintf('ode call\n') % debugging tag
            [S,Y] = ode45(@laplace,S_span,[0 1e-100 0],[],b3,c); %
               may need to increase span of S if curve does not loop
%             fprintf('out of ode call\n') % debugging tag

            Z=Y(:,1); % the first column of Y is the Z data. Needs to
                be defined to find cutoff point below.
```

```
% This loop stores the value, i, where the
% Laplace-Young data starts to 'loop' out of control.

i=1;
while Z(i)<Z(i+1); % finds where contact angle is 180 degrees (cutoff point)
   i=i+1;
end;

% Output Variables

x_ly_1=zeros(i,1);   % define as column vector
z_ly_1=zeros(i,1);   % define as column vector
S_ly_1=zeros(i,1);   % define as column vector
Phi_ly_1=zeros(i,1); % define as column vector

x_ly_1(1:i)=Y(1:i,2);            % Drop height dimensionless
z_ly_1(1:i)=Y(1:i,1);            % Drop width dimensionless
S_ly_1(1:i)=S(1:i,1);            % Arc length dimensionless
Phi_ly_1(1:i)=Y(1:i,3)*(180/pi); % Contact angle in degrees

[d3,z3]=min(abs(z_dc_nd(end)-z_ly_1)); % find indice of laplace data that closely matches raw data
error_b3=x_dc_nd(end)-x_ly_1(z3);      % finds relative horizontal error of end of trim data

if error_b3*error_b1>0;
    b1=b3;
    error_b1=error_b3;
    error_b=error_b1;
else
    b2=b3;
    error_b2=error_b3;
    error_b=error_b2;
end;
b=b3;
b3=(b1+b2)/2;

% If eps_b limit is never met, passthrough when b=b3
% for 10 iterations. r_b is b rounded to 15 decimal
% places, r_b3 is b3 rounded to 15 decimal places.
% d_rs is the difference between r_b and r_b3. When
% d_rs=0, r_b=r_b3 and when this happens 10 times, a
% minimum has been reached even if it isn't within
% eps_b.

r_b=round2(b,15);
```

```
            r_b3=round2(b3,15);

            d_rs=r_b-r_b3;

            if d_rs==0;
                ii=ii+1;
            end;

            if ii==10;
                eps_b=abs(error_b);
                ii=1;
            end;

            fprintf('b=%15.15f error_b=%15.15f d_rs=%f ii=%f eps_b=%f
               \n',b,error_b,d_rs,ii,eps_b);

        end;

b=b3;

%% Figures

% Define lines to be used in the plots
z_data_cutoff=[z_dc_nd(end);z_dc_nd(end)];
x_data_cutoff=[-4;4];
z_plane_1=[drop_height_nd;drop_height_nd];
X_circle_nd=(X_circle/scale)*c^.5;
Z_circle_nd=(Z_circle/scale)*c^.5;

% Figure with Laplace-Young data, b-minimization plane,
% Spherical approximation, Calculated z_plane, Drop interface
% data from image. Comment/Uncomment any data set you do not want
% to plot and take out corresponding legend value.
figure;
% Laplace-Young data
plot(x_ly_1(:,1),z_ly_1(:,1),'linewidth',2);
hold all
% b minimization plane
plot(x_data_cutoff,z_data_cutoff)
% Spherical approximation
plot(X_circle_nd,Z_circle_nd,'linewidth',2)
% Calculated z_plane
plot(x_data_cutoff,z_plane_1,'linewidth',2)
% Drop interface data from image
plot(x_dat_nd,z_dat_nd,'linewidth',2)
% Change axis limits and pbaspect to fit specific drop
axis([-5 5 -.1 4.9]);
set(gca,'YDir','reverse');
pbaspect([2 1 1]);
xlabel('X-Direction');
ylabel('Z-Direction');
```

```
legend('Laplace-Young data','b minimization plane','Spherical
   approximation','calculated z-plane','Raw Data')
title('Non-dimensionalized data comparison')
hold off

% Following figures used for check if problems arise

%%% Figure of raw data reset to apex at origin

% figure
% plot(x_dat,y_dat,'.')
% set(gca,'ydir','reverse')
% axis([-800 800 0 800])
% pbaspect([2 1 1])

%%% Figure of trimmed data with apex at origin

% plot(x_dat_cut,y_dat_cut,'.')
% set(gca,'ydir','reverse')
% axis([-800 800 0 800])
% pbaspect([2 1 1])

%% Find Contact Angle

dif=z_ly_1-drop_height_nd;
adif=abs(dif);
low=min(adif);
index=find(adif==low);
contact_angle=Phi_ly_1(index);

%% Data Save

[s]=dat_xls(bond,b,error_b,c,error_c,contact_angle,theta_sc,
   equator_radius,drop_height_cm,time,substrate,fluid,temperature,
   excel,scale,file,folder,eps_c);

% eof
```

E.3 Functions

Here is a list of all of the functions used in the contact angle measurement program in alphabetical order.

E.3.1 apex

```
function [x,z]=apex(bound)

% APEX finds the x and z values for the apex to be used to reset
```

```
% the data to have an apex at (0,0).

z=min(bound(:,2));                               % Gives z position of apex
x=bound(round(mean(find(bound(:,2)==z))),1); % Gives x position of apex

end

% eof
```

E.3.2 c_data

```
function [p2]=c_data(h,z)

% C_DATA finds indice of point half-way between z_plane and apex
% on drop interface for comparing c values.

c_cutoff=h*.5; % set point to minimize error for c, .5 means error is found at the half way point

p1=find(min(abs(z-c_cutoff))==abs(z-c_cutoff)); % set up data set to find minimum of the difference of cutoff and data

p2=p1(end); % last indice in set, wanted for right hand side of drop

end

% eof
```

E.3.3 circle

```
function [X,Y,H]=circle(center,radius,NOP,style)
%---------------------------------------------------------------
% H=CIRCLE(CENTER,RADIUS,NOP,STYLE)
% This routine draws a circle with center defined as
% a vector CENTER, radius as a scaler RADIS. NOP is
% the number of points on the circle. As to STYLE,
% use it the same way as you use the rountine PLOT.
% Since the handle of the object is returned, you
% use routine SET to get the best result.
%
%   Usage Examples,
%
%   circle([1,3],3,1000,':');
%   circle([2,4],2,1000,'--');
%
%   Zhenhai Wang <zhenhai@ieee.org>
%   Version 1.00
```

```
%   December, 2002
%------------------------------------------------------------------

if (nargin <3),
 error('Please see help for INPUT DATA.');
elseif (nargin==3)
    style='b-';
end;
THETA=linspace(0,2*pi,NOP);
RHO=ones(1,NOP)*radius;
[X,Y] = pol2cart(THETA,RHO);
X=X+center(1);
Y=Y+center(2);
% H=plot(X,Y,style);
% axis square;

end

% eof
```

E.3.4 contact_angle_sphere_cap

```
function [tsc]=contact_angle_sphere_cap(xc,zc,Xc,Zc,dh)

% CONTACT_ANGLE_SPHERE_CAP finds the contact angle of the drop
% using the spherical cap approximation.

value1=Zc-dh;
value2=abs(value1);
value3=min(value2);

index=find(value2==value3);

xp=Xc(index);
zp=Zc(index);

xd=xc-xp;
zd=zc-zp;

theta_a=atan(zd/xd);

theta_a=abs(theta_a);

tsc_rad=theta_a+(pi/2);

dtr=180/pi;

tsc=tsc_rad*dtr;

end
```

```
%eof
```

E.3.5 dat_xls

```
function [w]=dat_xls(bo,b,err_b,c,err_c,theta,tsc,equ_rad,dh,d,ss,f,
    temp,xls,sc,filename,foldername,c_lim)

% DAT_XLS saves data to an xls file.

all_data={[foldername,filename],'','','';'' '' '' '';...
     '','','','';'' '' '' '';...
     'test run','substrate','fluid','temperature';[d,'-',filename] ss
        f temp;...
     '','','','';'' '' '' '';...
     'scale factor',sc,'c limit',c_lim;'b',b,'','';...
     'error b',err_b,'','';'error c',err_c,'','';...
     'equatorial radius',equ_rad,'','';'drop height',dh,'','';...
     'bond',bo,'','';'theta - LY',theta,'','';...
     'c',c,'','';'theta - SC',tsc,'',''};

w=xlswrite([d,xls],all_data,filename,'A1');
end

% eof
```

E.3.6 edge

```
function [flip_bound]=edge(Black_White,Cropped_Image)

% EDGE finds the boundary of a black and white image using
% bwtraceboundary.
% BWTRACEBOUNDARY has form: bwtraceboundary(image,[x,y],
% 'direction') with [x,y] being the starting point and 'direction'
% being N,S,E,W for north, south, east, west.
% The boundary includes the edge of the image and must be trimmed
% to only include z-plane and drop interface data.

boundary_full=bwtraceboundary(Black_White,[1,1],'S');  % find total
    boundary of image

bound1=boundary_full(find(boundary_full(:,2)==(length(Black_White
    (1,:))-1), 1, 'last' ):length(boundary_full(:,1))-max(
    boundary_full(:,1)),1); % trim the full boundary of column one to
    eliminate the image edge

bound2=boundary_full(find(boundary_full(:,2)==(length(Black_White
    (1,:))-1), 1, 'last' ):length(boundary_full(:,1))-max(
    boundary_full(:,1)),2); % trim the full boundary of column two to
```

```
eliminate the image edge

bound=[bound2,bound1];    % Recombine the two colums to form a vector of values [x,y]
flip_bound=flipud(bound); % The "bound" vector is in reverse order of data points from right to left. flipud() flips the vectors to follow data points from left to right

end

% eof
```

E.3.7 fluid_prop

```
function [rho_w,rho_a,sig,g,c,l_c]=fluid_prop

% FLUID_PROP allows users to input specific fluid properties of
% fluid density, surrounding fluid density (typically air),
% surface tenstion. This also defines gravitational acceleration
% and calculates c, which is a ratio of physical properties.

% Density of the fluid used
rho_w=input('Density of fluid in (g/cm^3). water¬0.998 (g/cm^3):');
if isempty(rho_w)
    rho_w=.998;
end;
% Density of surround fluid, typically air
rho_a=input('Density of surrounding fluid in (g/cm^3). air¬0.0012 (g/cm^3):');
if isempty(rho_a)
    rho_a=.0012;
end;
% Gravitational Acceleration
g=981.7; % cm/s^2
% Surface Tension of fluid
sig=input('Surface Tension of fluid in (dyne/cm or mN/m). water¬70 (dyne/cm):');
if isempty(sig)
    sig=70.10;
end;

% To get c=11, sig=89.02 dynes/cm.
% To get c=13.96(14), sig=70.10 dynes/cm.
% To get c=17, sig=57.60 dynes/cm.

% Definition of c: c=((rho_w-rho_a)*g)/sigma_dim in units of 1/cm^2
c=((rho_w-rho_a)*g)/sig;
z=c^.5;
l_c=1/z;
```

```
end

% eof
```

E.3.8 height_interface

```
function [dh,dh_cm,dh_nd,x_cut,z_cut,x_cm,z_cm,x_nd,z_nd,er,beta,
    x_equ_point_l,x_equ_point_r,z_equ_point_l,z_equ_point_r]=
    height_interface(p,s,const,z,x,index_left,index_right)

% HEIGHT_INTERFACE finds the drop height and the drop interface
% data without the z-plane data. 'dh' is the drop height, which
% is then scaled to centimeters and then non-dimensionalized.
% 'cutoff' finds the top 95% of the drop as currently defined,
% user can change this to whatever percentage of the drop data
% they would like to retain. 'x_cut,z_cut' are the x and z data
% for the top 95% of the drop, which are scaled to centimeters
% and then non-dimensionalized.

dh=p;                  % in pixels
dh_cm=dh/s;            % in cm
dh_nd=dh_cm*const^.5; % non-dimensionalized

percent=.85;           % Percent drop height where b errors are
    calculated

cutoff=round(percent*dh);

idx=(find(z<cutoff));
first=idx(1);
last=idx(end);

x_cut=x(first:last);    % trimmed data in pixels
z_cut=z(first:last);    % trimmed data in pixels

x_cm=x_cut./s;          % converted to cm
z_cm=z_cut./s;          % converted to cm

x_nd=x_cm*const^(1/2); % converted to non-dimensional
z_nd=z_cm*const^(1/2); % converted to non-dimensional

equator_right=max(x_cm);
equator_left=min(x_cm);

er=(abs(equator_right)+abs(equator_left))/2;

beta=er^2*const;

% To find wetted radius, the index of the contact line on the
% left and right side of the drop was found in the plane function.
```

```
% Now use the index values to find the x and z values at both
% points to obtain a distance. Divide distance by 2 to obtain
% wetted radius.
%
% Problems finding wetted radius

% x_l=x(index_left);
% x_r=x(index_right);
%
% z_l=z(index_left);
% z_r=z(index_left);
%
% dist=((x_r-x_l)^2+(abs(z_l-z_r))^2)^.5;
%
% radius=dist/2;

% Spherical cap approximations need the values at the equator
% radius for the left and right hand side in pixel format to be
% converted.

x_equ_point_l=min(x_cut);
x_equ_point_r=max(x_cut);

index_left=find(x_cut==x_equ_point_l);
index_right=find(x_cut==x_equ_point_r);

z_equ_point_l=mean(z_cut(index_left));
z_equ_point_r=mean(z_cut(index_right));

end

% eof
```

E.3.9 image_analysis

```
function [Edge_image,Black_White,Trimmed_image]=image_analysis(
    Cropped_Image)

% IMAGE_ANALYSIS finds an image trimmed by the value 'co'. The
% BW1 image uses a threshold defined by 'thresh'. E1 finds the
% perimeter of BW1 using built in function 'bwperim'. The
% Trimmed_image is the original image trimmed by 'co', Black_White
% is the black/white image trimmed by 'co', Edge_image is the
% perimeter data trimmed by 'co'.

% co is cutoff for top and bottom and left and right of image
co=5;

% thresh is the threshold of the image for converting to black
% and white.
```

```
thresh=0.9;

BW1=im2bw(Cropped_Image,thresh);

E1=bwperim(BW1);

Trimmed_image=Cropped_Image(co:length(Cropped_Image(:,1))-co,co:
    length(Cropped_Image(1,:))-co);

Black_White=BW1(co:length(BW1(:,1))-co,co:length(BW1(1,:))-co);

Edge_image=E1(co:length(E1(:,1))-co,co:length(E1(1,:))-co);

end

% eof
```

E.3.10 image_input

```
function [Image,Cropped_Image,filepath,filename]=image_input

% IMAGE_INPUT imports image to be processed. 'decision' decides
% if image needs cropped or not, y for yes, n for no.
% 'Cropped_Image' is the output image to be used for the program.
% 'Image' is the original image.

% Opens filepath of previously opened folder
oldpath=char(textread('filepath.dat','%s','whitespace',''));
% Lets user select full filepath of image graphically
[filename,filepath]=uigetfile([oldpath,'*.tif'],'open file:');

% This routine changes the filepath if the user looks in a new folder
fid=fopen('filepath.dat','w');
fprintf(fid,'%s',filepath);
fclose(fid);

% Saves image to 'I' from specific filepath as selected above
Image=imread([filepath,filename]);

% If image needs cropped select 'y', if not select 'n'.
decision=input('Does this image need cropped (y,n):','s');

if isempty(decision);
    Cropped_Image=imcrop(Image);
elseif decision == 'y';
    Cropped_Image=imcorp(Image);
elseif decision == 'n';
    Cropped_Image=Image;
end;
```

```
end

% eof
```

E.3.11 laplace

```
%Derek Fultz and Russ Stacy
%7-24-07
%
% LAPLACE defines the ordinary differential equations to be solved.
% z=drop height
% x=distance from axis to drop interface
% phi=contact angle
% 1/b=radius of curvature at apex
% s=arc length
%
% NON-DIMENSIONALIZE Z,X,S,B USING C^(1/2)
% B=b*c^(1/2)
% X=x*c^(1/2)
% Z=z*c^(1/2)
% S=s*c^(1/2)
%
% No need to define X,Z,S for equations
%
% KEY OF VARIABLE DEFINITIONS FOR SYSTEM
% Z=y(1); Z'=dy(1); Z' is with respect to S
% X=y(2); X'=dy(2); X' is with respect to S
% phi=y(3); phi'=dy(3); phi' is with respect to S
%
% Z'=sin(phi)
% X'=cos(phi)
% phi'=2/B+Z-(sin(phi)/X)

function [dy]=laplace(s,y,m,n)
dy = zeros(3,1); % a column vector

B=m*n^.5;         % non-dimensionalized curvature at apex

dy(1)=sin(y(3));
dy(2)=cos(y(3));
dy(3)=(2/B)+y(1)-(sin(y(3))/y(2));

end

% eof
```

E.3.12 plane

```
function [p,p_cm,p_nd,index_left,index_right]=plane(bound,z,apex_z,s,
    const)

% PLANE finds the z-plane for the image. The z-plane is defined
% as the surface in which the drop sets.
%
% The x indice of the drop apex must be found first using x_ind
% in order to isolate the left and right planes individually.
%
% plane_ind_left are the indices that are within +/- 1 pixel of
% the first element on the left. plane_ind_right are the indeces
% that are within +/- 1 pixel of the last element on the right.
% left_plane_ind are just those indices that fall on the left
% side of the apex and right_plane_ind are just those indices
% that fall on the right side of the apex.
%
% left_plane is the values at the indices for the left,
% right_plane is the values at the indices for the right.
% left_avg and right_avg are the averages of the two vectors. An
% average of these to values is used as the z_plane being the
% mean of both sides of the drop.

x_ind=round(mean(find(bound(:,2)==apex_z))); % x indice of apex

left=z(1);     % plane_1=surface on left of drop
right=z(end);  % plane_2=surface on right of drop

plane_limit=15; % compare left and right sides of z-plane. If not
    within limit, end program and print "Image is not level."

% This is to ensure that the left and right z-planes are within a
% specific range of each other to guard against being unlevel.
if abs(left-right)>plane_limit;
     fprintf('Image is not level! \n');
     fprintf('left=%f\n',left);
     fprintf('right=%f\n',right);
end;

plane_ind_left=find(z≥left-1 & z≤left+1);
plane_ind_right=find(z≥right-1 & z≤right+1);
left_plane_ind=plane_ind_left(find(plane_ind_left<x_ind));
right_plane_ind=plane_ind_right(find(plane_ind_right>x_ind));

left_plane=z(left_plane_ind);
right_plane=z(right_plane_ind);

left_avg=mean(left_plane);
right_avg=mean(right_plane);

lr=[left_avg right_avg];
```

```
p=round(mean(lr));    % z_plane in pixels
p_cm=p/s;             % z_plane in cm
p_nd=p_cm*const^.5;   % z_plane non-dimensionalized

% To find wetted radius, find the maximum of the
% left_plane_ind(x value) and the minimum of the
% right_plane_ind(x value) and evalute the x and z data at that
% index. The evaluation takes place in height_interface function.

index_left=max(left_plane_ind);
index_right=min(right_plane_ind);

end

% eof
```

E.3.13 sphere_cap

```
function [xc,zc,r]=sphere_cap(xl,zl,xr,zr)

% SPHERE_CAP finds the data required to find the sphrerical cap
% approximation for sessile drops. This is used to compare to
% contact angle data collected using the laplace-young equation.
% The apex data is (0,0) and the two equator points are used
% (x_equator_left,z_equator_left) and (x_equator_right,
% z_equator_right) to find a circle using three points.


ma=zl/xl; % slope for left line between apex and left point.
mb=zr/xr; % slope for right line between apex and right point.

xc=(ma*mb*(zr-zl)+mb*(xl)-ma*(xr))/(2*(mb-ma));

zc=((-1/ma)*(xc-(xl/2)))+(zl/2);

r=(xc^2+zc^2)^.5;

end

% eof
```

E.3.14 trans_scale

```
function [x,z,x_cm,z_cm,x_nd,z_nd]=trans_scale(x_apex,z_apex,bound,s,
    const)

% TRANS_SCALE transforms the drop data such that the apex is at
% the origin (0,0) and then scales the data from pixels (x,z) to
% centimeters (x_cm,z_cm) to non-dimensional form (x_nd,z_nd).
```

```
% x is the x drop data, z is the z drop data, x_apex is the
% original x  position of the apex, z_apex is the original z
% position of the apex, x_cm is the x drop data in centimeters,
% z_cm is the z drop data in centimeters, s is the scale to
% convert pixels to centimeters, x_nd is the non-dimensionalized
% x data, z_nd is the non-dimensionalized z data, and const is a
% constant related to the laplace constant (const is c in main
% program).

x=bound(:,1)-x_apex;  % raw x data with origin at apex still in
    pixels
z=bound(:,2)-z_apex;  % raw z data with origin at apex still in
    pixels

x_cm=x./s;  % converted to cm
z_cm=z./s;  % converted to cm

x_nd=x_cm*const^(1/2);  % converted to non-dimensional
z_nd=z_cm*const^(1/2);  % converted to non-dimensional

%%% VARIABLES CHANGED ON 12/9/08
% x_plot_px to x_dat
% x_plot_cm to x_dat_cm
% x_nd to x_dat_nd

end

% eof
```

F. PROGRAM WHICH CALCULATES THE SCALE FACTOR FOR IMAGES

```
% Russ Stacy
% 10_13_08
%
% SCALE_1 creates sc.mat which contains the scale factor for the
% given set of data to be processed. Load scale factor image for
% specific data through graphical user interface.

%% Scale Factor
clear
clc

% Import image to be processed. 'decision' decides if image needs
% cropped or not, y for yes, n for no. 'Cropped_Image' is the
% output image to be used for the program. 'Image' is the
% original image. Must have filepath.dat file in the folder
% with this file.

% Opens filepath of previously opened folder
oldpath=char(textread('filepath.dat','%s','whitespace',''));
% Lets user select full filepath of image graphically
[filename,filepath]=uigetfile([oldpath,'*.tif'],'open file:');

% This routine changes the filepath if the user looks in a new
% folder
fid=fopen('filepath.dat','w');
fprintf(fid,'%s',filepath);
fclose(fid);

% Saves image to 'I' from specific filepath as selected above
scale=imread([filepath,filename]);

figure, imshow(scale, [])

[m,n]=ginput(2);
pixels=m(2)-m(1);
mm=input('Enter size of scale in mm: ');
cm=mm/10;
```

```
scale=abs(pixels/cm);

save sc.mat scale;

% eof
```

G. PROGRAM USED TO PLOT A RANGE OF LAPLACIAN CURVES BY VARYING B

This code plots Laplacian curves for a fixed c value while varying b.

```
% Derek Fultz and Russ Stacy
% 8_10_07
%
% B_VARY solves 3 simultaneous differential equations and plots
% the Laplacian curves for varying b values and a fixed c value.
% The equations are the Laplace-Young equation and two geometric
% relationships.

clear
clc

% Define text and line sizes for plotting

textsize=24;
titlesize=26;
linesize=2;

% Solve Laplace-Young equation for varying b values

S_span=(0:.001:8); % define step for ode solver
c=14; % approximately the value for water

% Put bb in the form of bb={b1 b2 b3 b4 ...};

bb=[.1 .5 1 5 10]; % range of b values to use

% This is a large loop that solves the Laplace-Young equation for
% several values of b to find the solution.

for j=1:length(bb);

b=bb(j); % set b as individual values of bb

[S,Y] = ode45(@laplace,S_span,[0 1e-100 0],[],b,c); % may need to
   increase second input if curve does not loop

```

```
Z=Y(:,1); % the first column of Y is the Z data. Needs to be defined
    to find cutoff point below.

% This loop stores the value, i, where the Laplace—Young data starts
    to
% 'loop' out of control.

i=1;
while Z(i)<Z(i+1); % find where contact angle is 180 degrees (cutoff
    point)
     i=i+1;
end;

ll(j)=i;

% Output Variables

x_ly(1:i,j)=Y(1:i,2);             % Drop height dimensionless
z_ly(1:i,j)=Y(1:i,1);             % Drop x dimension dimensionless
S_ly(1:i,j)=S(1:i,1);             % Arc length dimensionless
Phi_ly(1:i,j)=Y(1:i,3)*(180/pi); % Contact angle in degrees

end;

% Defines laplace curves for both sides of apex.

x_ly_f_1=[-x_ly(:,1);x_ly(:,1)];
z_ly_f_1=[z_ly(:,1);z_ly(:,1)];

x_ly_f_2=[-x_ly(:,2);x_ly(:,2)];
z_ly_f_2=[z_ly(:,2);z_ly(:,2)];

x_ly_f_3=[-x_ly(:,3);x_ly(:,3)];
z_ly_f_3=[z_ly(:,3);z_ly(:,3)];

x_ly_f_4=[-x_ly(:,4);x_ly(:,4)];
z_ly_f_4=[z_ly(:,4);z_ly(:,4)];

x_ly_f_5=[-x_ly(:,5);x_ly(:,5)];
z_ly_f_5=[z_ly(:,5);z_ly(:,5)];

% Plot the range of Laplacian curves for varying b values

figure;
plot(x_ly_f_1,z_ly_f_1,'k.','LineWidth',linesize);
hold on
plot(x_ly_f_2,z_ly_f_2,'k.','LineWidth',linesize);
plot(x_ly_f_3,z_ly_f_3,'k.','LineWidth',linesize);
plot(x_ly_f_4,z_ly_f_4,'k.','LineWidth',linesize);
plot(x_ly_f_5,z_ly_f_5,'k.','LineWidth',linesize);
```

```
% title('Laplacian Curves for Varying b','FontSize',30,'FontWeight','
   b');
set(gca,'YDir','reverse');
% grid minor
set(gca,'FontSize',titlesize);
axis([-6 6 -1 3])
pbaspect([3 1 1])
xlabel('X (dimensionless)','FontSize',titlesize);
ylabel('Z (dimensionless)','FontSize',titlesize);
% legend('b=.1','b=.5','b=1','b=5','b=10','b=25','Location','NW');
text(.2,.8,'0.1','FontSize',textsize);
text(.8,1.8,'0.5','FontSize',textsize);
text(1.6,2.05,'1.0','FontSize',textsize);
text(3.25,2.3,'5.0','FontSize',textsize);
text(4.3,2.2,'10.0','FontSize',textsize);
hold off

% eof
```

H. PROGRAM USED TO PLOT A RANGE OF LAPLACIAN CURVES BY VARYING C

This code plots Laplacian curves for a fixed b value while varying c.

```
% Derek Fultz and Russ Stacy
% 8_10_07
%
% C_VARY solves 3 simultaneous differential equations and plots
% the Laplacian curves for varying c values and a fixed b value.
% The equations are the Laplace-Young equation and two geometric
% relationships.

clear
clc

% Define text and line sizes for plotting

textsize=24;
titlesize=26;
linesize=2;

% Solve Laplace-Young equation for varying c values

S_span=(0:.001:8); % define step for ode solver
b=1; % constant value for b

% Put cc in the form of cc={c1 c2 c3 c4 ...};

cc=[6 14 22]; % range of c values to use

% This is a large loop that solves the Laplace-Young equation for
% several values of b to find the solution.

for j=1:length(cc);

c=cc(j); % set c as individual values of cc

[S,Y] = ode45(@laplace,S_span,[0 1e-100 0],[],b,c); % may need to increase second input if curve does not loop

```

```
Z=Y(:,1); % the first column of Y is the Z data. Needs to be defined to find cutoff point below.

% This loop stores the value, i, where the Laplace-Young data
% starts to 'loop' out of control.

i=1;
while Z(i)<Z(i+1); % find where contact angle is 180 degrees (cutoff point)
    i=i+1;
end;

ll(j)=i;

% Output Variables

x_ly(1:i,j)=Y(1:i,2);            % Drop height dimensionless
z_ly(1:i,j)=Y(1:i,1);            % Drop x dimension dimensionless
S_ly(1:i,j)=S(1:i,1);            % Arc length dimensionless
Phi_ly(1:i,j)=Y(1:i,3)*(180/pi); % Contact angle in degrees

end;

% Defines laplace curves for both sides of apex.

x_ly_f_1=[-x_ly(:,1);x_ly(:,1)];
z_ly_f_1=[z_ly(:,1);z_ly(:,1)];

x_ly_f_2=[-x_ly(:,2);x_ly(:,2)];
z_ly_f_2=[z_ly(:,2);z_ly(:,2)];

x_ly_f_3=[-x_ly(:,3);x_ly(:,3)];
z_ly_f_3=[z_ly(:,3);z_ly(:,3)];

% x_ly_f_4=[-x_ly(:,4);x_ly(:,4)];
% z_ly_f_4=[z_ly(:,4);z_ly(:,4)];
%
% x_ly_f_5=[-x_ly(:,5);x_ly(:,5)];
% z_ly_f_5=[z_ly(:,5);z_ly(:,5)];

% Plot the range of Laplacian curves for varying c values

figure;
plot(x_ly_f_1,z_ly_f_1,'k.','LineWidth',linesize);
hold on
plot(x_ly_f_2,z_ly_f_2,'k.','LineWidth',linesize);
plot(x_ly_f_3,z_ly_f_3,'k.','LineWidth',linesize);
% plot(x_ly_f_4,z_ly_f_4,'k.','LineWidth','linesize');
% plot(x_ly_f_5,z_ly_f_5,'k.','LineWidth','linesize');
% title('Laplacian Curves for Varying c','FontSize',30,'FontWeight','b');
```

```
set(gca,'YDir','reverse');
% grid minor
set(gca,'FontSize',titlesize);
axis([-3 3 -1 3])
pbaspect([(3/2) 1 1])
xlabel('X (dimensionless)','FontSize',titlesize);
ylabel('Z (dimensionless)','FontSize',titlesize);
% legend('c=6','c=14','c=22','location','NW');
text(-1.1,1.75,'6.0','FontSize',textsize);
text(-1.55,1.875,'14.0','FontSize',textsize);
text(-2.0,2.05,'22.0','FontSize',textsize);
hold off

% eof
```

REFERENCES

[1] James Larminie and Andrew Dicks. *Fuel Cell Systems Explained*. Chapter 4. Wiley, 2003.

[2] Wolf. Vielstich, Arnold. Lamm, and Hubert A. Gasteiger. *Handbook of Fuel Cells: Fundamentals, Technology, and Applications*, Volume 3: Fuel Cell Technology and Applications. Chapter 46. Wiley, Chichester, England; Hoboken, N.J., 2003.

[3] Arthur W. Adamson. *Physical Chemistry of Surfaces*. Wiley, 1990.

[4] Jeffrey S. Allen. An analyitical solution for determination of small contact angles from sessile drops of arbitrary size. *Journal of Colloid and Interface Science*, 261: 481–489, 2003.

[5] F. Bashforth and J.C. Adams. *An Attempt to Test The Theories of Capillary Action by Comparing the Theoretical and Measured Forms of Drops of Fluid*. Cambridge Univ. Press, London, 1883.

[6] S. Hartland and R.W. Hartley. *Axisymmetric FluidLiquid Interfaces*. Elsevier Scientific, Amsterdam/Oxford/New York, 1976.

[7] D. Li, P. Cheng, and A.W. Neumann. Contact angle measurement by axisymmetric drop shape analysis (ADSA). *Advances in Colloid and Interface Science*, 39:347–382, 1992.

[8] N. M. Dingle and M. T. Harris. A robust algorithm for the simultaneous parameter estimation of interfacial tension and contact angle from sessile drop profiles. *Journal of Colloid and Interface Science*, 286:670–680, 2005.

[9] Robert N. Wenzel. Resistance of solid surfaces to wetting by water. *Industrial & Engineering Chemistry*, 28(8):988–994, 1936.

[10] A. B. D. Cassie and S. Baxter. Wettability of porous surfaces. *Transactions of the Faraday Society*, 40:546–551, 1944.

[11] Jaroslaw Drelich, Jan D. Miller, and Jan Hupka. The effect of drop size on contact angle over a wide range of drop volumes. *Journal of Colloid and Interface Science*, 155(2):379 – 385, 1993. ISSN 0021-9797.

[12] Jaroslaw Drelich. The effect of drop (bubble) size on contact angle at solid surfaces. *Journal of Adhesion*, 63:31–51, 1997.

[13] S. Vafaei and M. Z. Podowski. Analysis of the relationship between liquid droplet size and contact angle. *Advances in Colloid and Interface Science*, 113(2-3):133–146, 2005.

[14] Jaroslaw Drelich and Jan D. Miller. The line/pseudo - line tension in three-phase systems. *Particulate Science and Technology*, 10:1–20, 1992.

[15] GDL samples provided courtesy of Paul Nicotera, General Motors Corporation, Fuel Cell Activities, Honeoye Falls, New York