2015

Time and Temperature Dependent Surface Tension Measurements of Responsive Protein-Based Polymer Surfactant Solutions

Hakan Celik
Cleveland State University

Recommended Citation
Time and Temperature Dependent Surface Tension Measurements of Responsive Protein-based Polymer Surfactant Solutions

HAKAN CELIK

Bachelor of Science in Physics
Dicle University
July 2005

submitted in partial fulfillment of requirements for the degree
MASTER OF SCIENCE IN BIOMEDICAL ENGINEERING
at the
CLEVELAND STATE UNIVERSITY
August 2015
We hereby approve this thesis for

Hakan Celik

Candidate for the Master of Science in Biomedical Engineering degree for the

Department of Chemical and Biomedical Engineering

and the CLEVELAND STATE UNIVERSITY

College of Graduate Studies

Nolan B. Holland, PhD
Thesis Chairperson
Department of Chemical and Biomedical Engineering

____________________
Date

Sridhar Ungarala, PhD
Department of Chemical and Biomedical Engineering

____________________
Date

Dhananjai B. Shah, PhD
Department of Chemical and Biomedical Engineering

____________________
Date

Student’s Date of Defense: 5 August 2015
ACKNOWLEDGMENTS

I would like to convey my deep appreciation to Dr. Nolan B. Holland for the opportunities he afforded me in the laboratory. His support and guidance inside the laboratory as well as other facets of my education have been invaluable. I am incredibly thankful to Dr. Metin Aytekin for advice, support and encouragement.

I would like to extend my appreciation to my committee members Dr. Dhananjai B. Shah and Dr. Ungarala for the encouragement and advice throughout my studies. The staff and faculty of the Biomedical Engineering Department at Cleveland State University have provided an encouraging and comfortable environment in which to study. I am grateful for all that the department has done to facilitate learning and cooperation. My colleagues at the laboratory provided support for which I extend my gratitude.

Ms. Becky Laird and Ms. Darlene Montgomery have provided invaluable support throughout my studies, many thanks for everything. Thank you to Dr. Jim Cole, Eric Helm, Jack Gavin, and Ryan Martin for all the advice and encouragement in my work. I extend my gratitude to Dr. Orhan Talu and Dr. Joanne Belovich for all the encouragement and advice provided.

Finally, I would like to express my appreciation of Dr. Veysel Celik for all of the academic and moral support. I would like to thank Holly Korovich and Adam Maraschky for being supportive friends and providing language support. My I would like to extend my gratitude for all the support and motivation from my family.
TIME AND TEMPERATURE DEPENDENT SURFACE TENSION
MEASUREMENTS OF RESPONSIVE PROTEIN-BASED POLYMER
SURFACTANT SOLUTIONS
HAKAN CELIK

ABSTRACT

A three-armed star elastin-like polypeptide (ELP-foldon) has thermoreversible character which exhibits phase separation above a transition temperature ($T_t$) in physiologic salt concentrations. At lower salt concentration, the ELP-foldon behaves like a thermoresponsive surfactant, exhibiting micelle formation above its $T_t$. The purpose of this study is characterize the surfactant behavior of the ELP-foldon at air-liquid interface by measuring the surface tension. The surface tension is measured as a function of time for different ELP concentrations from 10 nM to 50 μM and over range of temperatures from 25 to 35 ℃ using the axisymmetric drop shape analysis (ADSA). The ADSA is a method which is based on the analysis of the shape and size of drop or bubble profiles to measure surface tension.

It has been determined that the surface tension is not different between conditions where there are no micelles and where micelle form. Therefore, a critical micelle concentration (c.m.c.) measurement by surface tension is not meaningful. The surface tension exhibits a time-dependent reduction which can be fit with the Hua-Rosen equation. The meso-equilibrium surface pressure is ~23 mN/m and does not vary with the bulk concentration or the temperature. The time to reach the meso-equilibrium does vary with the bulk concentration. These times scale with concentration by a power of -1.2 and -1.3, suggesting that the process is not fully diffusion limited.
# TABLE OF CONTENTS

ABSTRACT ............................................................................................................................... iv

LIST OF TABLES .................................................................................................................. vii

LIST OF FIGURES .............................................................................................................. viii

CHAPTER I ............................................................................................................................ 1

INTRODUCTION AND BACKGROUND ............................................................................. 1

1.1 Introduction ..................................................................................................................... 1

1.2 Protein Adsorption ....................................................................................................... 2

1.3 Polymer Adsorption ..................................................................................................... 7

1.4 Surfactants (surface active agent) ............................................................................... 8

1.4.1 Anionic surfactants ............................................................................................... 9

1.4.2 Cationic surfactant ............................................................................................... 9

1.4.3 Nonionic surfactant ............................................................................................. 10

1.4.4 Zwitterion surfactant .......................................................................................... 10

1.5 Micelle formation ....................................................................................................... 10

1.5.1 Thermodynamic of Micelle Formation ................................................................ 11

1.5.2 Parameters impacting c.m.c .................................................................................. 13

1.5.2.1 Impact of Temperature and Pressure ............................................................ 13

1.5.2.2 Impact of Added Salt .................................................................................... 13

1.5.2.3 Impact of Head Group and Chain Length .................................................... 14

1.5.2.4 Impact of Organic Molecules ...................................................................... 15
LIST OF TABLES

Table 1.1 Sum of the free energy change of the system ........................................... 20
Table 2.1 The sample image’s result ............................................................................. 39
Table 3.1 The solution and the solvent surface tension values ....................................... 50
Table 3.2 Fit Parameters for Linear and Trimer Constructs ............................................ 56
LIST OF FIGURES

Figure 1 Dilute solution of typical proteins’ surface tension change ................................. 4
Figure 2 Surface pressure is a function of the fractional coverage using a Langmuir
adsorption model ............................................................................................................ 7
Figure 3 Surfactant is with a triple structure ....................................................................... 8
Figure 4 The polymer’s arms aggregation with respect to temperature .............................. 16
Figure 5 The surfactant entropy change ............................................................................. 18
Figure 6 General procedure of Axisymmetric Drop Shape Analysis (ADSA) .................... 25
Figure 7 Schematic diagram of the experimental setup of ADSA .................................... 25
Figure 8 Charter to obtained final drop profile ................................................................ 26
Figure 9 Drop image for steps of image analysis process ................................................ 27
Figure 10 The kernels’ horizontal (Gx) and vertical (Gy) derivative approximations ..... 28
Figure 11 Drop profile’s cut-off point ................................................................................ 30
Figure 12 Midpoint of drop profile and midpoint of each curve point .............................. 31
Figure 13 The distribution of the parameters geometrically on the drop ............................ 32
Figure 14 Drop profile analysis is used for the fitting process ........................................ 36
Figure 15 Theoretical and experimental curve are overlapped ........................................ 38
Figure 16 The minimum value of x against the value of b and c are described .................. 39
Figure 17 Comparison of literature values of surface tension of water and experimental
values of water is function of the temperature ............................................................. 41
Figure 18 Measured surface tension is a function of temperature for 10, 31.6, 0.1 μM ... 43
Figure 19 Surface tension is a function of time and temperature for a 0.2 μM ............... 45
Figure 20  Surface tension is a function of time and temperature for a 0.316 μM ........... 46
Figure 24  Surface tension is a function of time and temperature for a 0.1 μM .......... 47
Figure 22  Surface tension and rate of absorption (UV) is a function of time and temperature for a 50 μM ................................................................. 49
Figure 23  The surface pressure is a function of the temperature at 50 μM .............. 50
Figure 24  The surface tension is a function of temperature for all concentrations and a function of concentration at different temperatures ....................................... 51
Figure 25  Experimental data is fitted by Hua-Rosen equation to create theoretical curve for 0.2, 0.316, 1, 50 μM. ................................................................. 53
Figure 26  The half time (t*) is function of concentration for 0.2, 0.316, 1, 50 μM ....... 54
Figure 27  The polymers’ movement to air-aqueous surface ........................................ 56
CHAPTER I
INTRODUCTION AND BACKGROUND

1.1 Introduction

Proteins are large organic molecules formed in cells by binding amino acids to each other to form chains. Proteins are important for living organisms because almost every functional property of living organisms is performed by proteins. The structures of proteins are defined by genes which have an important role in the protein synthesis. Cells use the information in genes to produce all of the different protein structures for living organisms. The genes can be modified to synthesize protein-based polymers in living organisms.

Elastin-like polypeptides (ELP) are one such class of protein-based polymers which can be synthesized using molecular biology techniques to generate recombinant DNA (rDNA) molecules [1]. These protein-based polymers have been synthesized with the desired structure and precision control [1, 2]. The ELP sequence is based on a sequence in the elastin protein which can be found in blood, vessels, lungs, and skin [1]. ELPs are used for many applications, including tissue engineering and pharmaceutical and biomedical sciences since ELPs are generally non-toxic, biocompatible, biodegradable, and have good pharmacokinetics properties [1, 3].
Since ELP is a thermally responsive polymer which exhibits phase separation above a transition temperature ($T_t$), ELP conformation is changed by the temperature. Conformations of polymers are significantly affected by chemical and physical characteristics of the polymers.

Molecular weight and length of polymer chain are significantly important for physical properties of the polymer. To give an example, long polymer chain provides excellent wear resistance and impacts toughness in ultra-high-molecular-weight polyethylene (UHMWPE) [4]. Also, characterization of small polymers characterizations is easier than large polymers [5]. Furthermore, molecular size of polymers affects surface tension and surface tension change process. Surface tension can be defined as the energy which is needed to increase the surface area of a liquid. Surface tension is affected by surfactant, and some polymers are used as surfactants for liquid solutions.

1.2 Protein Adsorption

Polymers have long chain structures that are formed by connecting monomers via chemical bonds [5]. Amino acids are the monomers that form the polymer structure of proteins and polypeptides [5]. An important difference between proteins and polypeptides is that the proteins have a defined conformational folded structure which can be denatured through conformational changes; however, a polypeptide typically does not have a single structure, but rather assumes a random coil conformation. Therefore, adsorption kinetics of proteins and polypeptides are different.

Prior studies on protein solutions show protein film formation at the fluid interface by the adsorption of the proteins to the surface from the bulk solution [6]. The
formation of films causes the reduction of the surface tension, increasing surface pressure $(\Pi)$ defined by

$$\Pi = \gamma_0 - \gamma$$

where, $\gamma_0$ is the solvent surface tension, and $\gamma$ is the solution surface tension. Thereby, the increase in the surface pressure is the result of a decrease of the surface tension during protein adsorption. Protein adsorption occurs in the three steeps (diffusion, adsorption, and rearrangement). Firstly, the proteins diffuse to the interface. After the proteins reach to the interface, the state change of the proteins cause energy reduction in the system, and in this case, the proteins want to further minimize free energy by conformational rearrangement. The adsorption processes affect the time required for the system to reach reduced interfacial tensions.

The reduction of surface tension as a function of time during protein adsorption has been shown to exhibit an s-type curve for many proteins (Figure 1) [6]. The behavior is only observed in dilute protein solution. In higher protein concentrations, it is not observed, because, the proteins which are in the region close to surface reach to the surface quickly to generate high surface coverage [7].
One protein they studied was ovalbumin, a globular protein which has a molecular weight of 42 kDa, one disulfide bond, and an isoelectric point of 4.6 [6]. Denaturation of the ovalbumin by urea resulted in an increased time for the proteins to reach the interface with respect to the urea-free system, since both kinetics of the protein adsorption was reduced by denaturation. It has been suggested that the conformation resulted in higher flexibility of molecules in the interface that had increased rearrangement time [6].

Higher ionic strength of the solvent can affect the regimes by reducing Debye length of charged protein side groups, reducing the repulsion [6, 8] between the proteins. The protein diffusion velocity is increased, resulting in faster interfacial saturation. Since $\beta - casein$ is a protein without disulfide bonds, it has a more disordered structure when denatured, and this gives it flexible properties [6]. Prior studies show that $\beta - casein$ reach the second and third regime faster than ovalbumin since globular protein interfacial

**Figure 1** Dilute solution of typical proteins’ surface tension change as depending on time is illustrated. While the proteins adsorbed to the surface, the system shows three different regimes. Modified from Beverung et al. [6].
unfolding and rearrangement in the solution is slow and the processes take extra time for induction regime [6].

Regime I is an induction regime for the interfacial tension change and it is usually equal to the pure solution interfacial tension at low protein concentrations. A theoretical model for dynamics of interfacial tension (diffusion controlled adsorption kinetics) can was developed by Ward and Tordai [9]. In the model, the effects of diffusion on interfacial concentration $\Gamma(t)$ depends on bulk protein concentration ($C_b$), and diffusion coefficient ($D$), the relation is [9]

$$\Gamma(t) = 2C_b \sqrt{\frac{Dt}{\pi}}$$

where, back diffusion is negligible [6]. Regime I formation can also be explained by Langmuir adsorption isotherm with Gibbs equation [6]

$$\Pi(t) = -k_B T \Gamma_{max} \ln \left( 1 - \frac{\Gamma(t)}{\Gamma_{max}} \right)$$

where $\Pi$ is the surface pressure, $k_B$ is Boltzmann’s constant, $T$ is the temperature, $\Gamma(t)$ is the molecular surface concentration at the time $t$, and $\Gamma_{max}$ the molecular surface concentration at the maximum coverage [6]. To be seen in the equation, the surface pressure is increased by the fractional coverage $\Theta = \frac{\Gamma}{\Gamma_{max}}$ [6]. However, surface pressure also depends on molecular size of the surfactants. In the equation, the area covered by an adsorbing molecule in the interface is described as $\frac{1}{\Gamma_{max}}$, and molecular size of the surfactants can be compared by the value $\left( \frac{1}{\Gamma_{max}} \right)$ [6] and also, the value is biggest for
high-molecular size surfactants. As seen in the Figure 2, at the same fractional coverage point, low-molecular size surfactant surface pressure value is greater than the high-molecular size surfactant. Also, number of the low-molecular size surfactant is more than the high-molecular size surfactant. In the graphic, in the moment when smaller molecules reach the surface, surface pressure rise is observed. However, the effect of the biggest molecules on the surface pressure starts after the molecular coverage reaches to a certain value. The reaching time may cause an induction time to change of the surface tension.

Regime II forms by reduction of the surface tension. In the Regime II, the proteins tend to conformation change at the surface, and it causes spaces between the proteins at the surface. The spaces are filled from the bulk proteins. Thereby, protein concentration is increased at the surface reducing the surface tension. Also, Regime II can be explained that rigid parts of the adsorbed protein are relaxed by the conformational change and desorption of the protein provides new interaction area at the surface [6].
After the surface is more saturated with the proteins, monolayer forms. Bulk proteins can continue to adsorb to the monolayer forming multilayers. According to Douillard and Lefebvre’s studies on the two-layer model of protein adsorption, surface tension is affected only by first monolayer and second layer does not affect to the surface tension [10]. Therefore, after the monolayer formation, the surface tension does not change during multilayer formation resulting in constant surface tension of Regime III. Prior studies exhibited small change in the surface tension; however, the change can depend on the continuous small conformational change in the monolayer proteins [6].

1.3 Polymer Adsorption

Polymers can affect surface tension akin to proteins. However, since polymer structure is different than protein, processes which polymer adsorption to the surface may be different. Firstly, polymer adsorption process begins diffusion toward the surface. The diffusion takes a certain time. After the polymer reaches the surface, the polymer

\[ \frac{1}{\Gamma_{\text{max}}} = 20 \, \text{Å}^2/\text{molecule} \]

\[ \frac{1}{\Gamma_{\text{max}}} = 2000 \, \text{Å}^2/\text{molecule} \]

Figure 2 Surface pressure is a function of the fractional coverage using a Langmuir adsorption model at 298 K [6]. Where, as \( \frac{1}{\Gamma_{\text{max}}} = 20 \, \text{Å}^2/\text{molecule} \) is low-molecular size surfactant and \( \frac{1}{\Gamma_{\text{max}}} = 2000 \, \text{Å}^2/\text{molecule} \) is high-molecular size surfactant defined. Modified from Beverung et al [6].
rearranges itself on the surface in time. Also during the diffusion and the rearrangement, the polymer's desorption can occur to the bulk solvent. These processes result in characteristic time dependent surface tension to reach what is formed the meso-equilibrium point. According to Hua-Rosen study, the lag time is divided four regions (region I is induction time, region II is rapid fall region, region III is meso-equilibrium region, region IV is equilibrium region) [11]. The regions can be explained by diffusion controlled adsorption kinetics [9, 12]. According to Ward and Tordai approach, the relationship between time ($t$) and bulk polymer concentration ($C_b$) is $t \propto C_b^{-2}$ for a certain surface coverage [13].

1.4 Surfactants (surface active agent)

![Surfactant](image)

Figure 3 Surfactant is with a triple structure

Organic surfactants affect the surface tension when dissolved in water or an aqueous solution. Also, surfactants usually decrease the surface tension. The surface tension is basically Gibbs free energy in per unit area of surface. Surfactants are amphiphilic which consists of both hydrophobic tail groups and hydrophilic head groups. Hydrophobic molecules are non-polar molecules, and hydrophilic molecules are polar molecules, and we know that water is a polar molecule. Therefore, according to the
aphorism “like dissolves like”, the hydrophobic tails tend to leave from aqueous solution and the hydrophilic heads tend to go toward the aqueous solution. At the interfaces, these tendencies provide the formation of the adsorption of surfactants. At the interface, because of the adsorption of the surfactant molecules, intermolecular interaction forces between water molecules increase and a diminution occurs in the surface tension of the solution.

1.4.1 Anionic surfactants

If the polar head group is negatively charged, the surfactant is referred to as an anionic surfactant. A hydrophobic group can be bonded to one or two hydrophilic groups, such as sulfate, sulfonate, phosphate, and carboxylates alkyl sulfates. Anionic surfactants are used more than the other kind of surfactants since their production is easier and cheaper [14]. They are used in cleaning products, such as detergents, because solubility is increased in the water and oil by anionic surfactants becoming counter-ion [14].

1.4.2 Cationic surfactant

If the polar head group is positively charged, the surfactant is called a cationic surfactant. A hydrophobic group can be bonded to one or more hydrophilic groups. The majority of cationic surfactant are based on the nitrogen atom carrying the cationic charge [14]. Such as, alkyltrimethylammonium salts, cetylpyridinium chloride (CPC), benzalkonium chloride (BAC), benzethonium chloride (BZT), 5-bromo-5-nitro-1,3-dioxane, dimethyldioctadecylammonium chloride cetrimonium bromide, and dioctadecyldimethylammonium bromide (DODAB). Cationic surfactants are used in surface modifications such as softening, lubricating, corrosion inhibitors, and adhesion.
1.4.3 Nonionic surfactant

Surfactants without charge are usually called nonionic surfactants. The nonionic part of the surfactant has a large number of mostly nitrogen, oxygen and sulfur atoms. In contrast to ionic surfactants, physical properties of nonionic surfactants are not affected by electrolytes significantly [14]. However, nonionic surfactants are affected by temperature, and, in contrast to ionic surfactant, when temperature is increased; solubility of nonionic surfactants is reduced becoming hydrophilic in water [14]. Polyoxyethylene glycol alkyl ethers, polyoxypropylene glycol alkyl ethers, glucoside alkyl ethers, and glycerol alkyl esters are the familiar example of the nonionic surfactants.

1.4.4 Zwitterion surfactant

If a surfactant has both positive and negative functional groups in the polar head group, it is called as a zwitterionic surfactant, or an amphoteric surfactant. According to their structure and ambient conditions, the surfactants may possess anionic and cationic characteristics. Since they cause less damage to the skin and the eyes, they are used in personal hygiene productions, such as hair shampoo, cleansing lotions, and liquid soaps. The solution PH is important for these surfactants since it affects the surfactant charge, and it can cause a change in physicochemical properties of the surfactants, such as foaming, wetting properties, and cleaning effects. Zwitterion surfactants include the surfactant examples, dodesil betain and cocamidopropyl betaine.

1.5 Micelle formation

A micelle form by surfactant molecules form cluster together. In solution, because of their amphiphilic structure, surfactants change the solution physicochemical properties, such as, changing surface tension of the solution [14]. Also, ionic surfactants
manner electrolyte in dilute solution, and in solution, increasing of the surfactant concentration causes breaking down the delicate balance of electrostatic and hydration interactions [14]. In aqueous solution, micelle formation is under the influence of two forces [15], one of them is an attraction force causing molecules integration, and another force is a repulsive force preventing unrestricted growth of the micelle size to become a different macroscopic phase [15].

1.5.1 Thermodynamic of Micelle Formation

The formation of micelles can be explained by Gibbs free energy of mixing [16, 14]. The Gibbs free energy change is considered at constant temperature (T) and pressure (P) [16]. As is known, surfactants are classified with respect to their polar head group and due to their variations, Gibbs free energy would be different, each kind of surfactant is examined in a separate way using different parameters.

According to Pseudo-Phase Separation Model, the chemical potential of the monomer and surfactant in micelle form are equal at equilibrium [14]. The chemical potential of the monomer and surfactant in micelle form are showed as $\mu_s$ and $\mu_m$, respectively [14].

$$\mu_s = \mu_m$$

The chemical potential of monomeric surfactants is given by the following equation:

$$\mu_s = \mu_s^0 + RT\ln x_s$$

where $\mu_s^0$, chemical potential of the monomeric surfactant, is in the optimum state, $x_s$, mole fraction of monomer. Because the micelles are assumed to be in the optimum case [14], $\mu_s^0 = \mu_m$, and the Gibbs energy change resulting from the formation of micelles, $\Delta G_{mix}$ is given as follow; Where $\alpha$, micelles ionization degree
\[ \Delta G_{\text{mic}}^0 = \mu_m^0 - \mu_s^0 \]
\[ = \mu_m - \mu_s + RT\ln x_s \]
\[ = RT\ln x_s \]

The c.m.c value is equal to solubility limit of free monomers [14]. In this case, \( x_s = x_{\text{c.m.c.}} \), and \( \Delta G_{\text{mic}}^0 \) is defined as follows;

\[ \Delta G_{\text{mic}}^0 = RT\ln x_{\text{c.m.c.}} \]

\[ \Delta G_{\text{mix}} = \Delta H_{\text{mix}} - T\Delta S_{\text{mix}} \]

where \( \Delta H_{\text{mix}} \) is enthalpy of mixing, \( \Delta S_{\text{mix}} \) is entropy of mixing and \( T \) is temperature.

Because of the surfactant structure, we have to discuss the surfactant energy changing in different sections, such as tail and head group. The overall system has to be based; hence we have to take the solvent energy changing into account due to Gibbs free energy of mixing.

Consequently, c.m.c depends on the Gibbs free energy of mixing, and when total Gibbs free energy of mixing becomes less than in the beginning condition of the Gibbs free energy state, the surfactants will form micelle. Since the head groups always are inside of the solution, there are small energy change observed due this, and it can be considered negligible.
1.5.2 Parameters impacting c.m.c

1.5.2.1 Impact of Temperature and Pressure

The Krafft point is the minimum temperature at which solubility of the surfactant is equal to c.m.c formation [16, 17]. Effect of the temperature is quite different on ionic and non-ionic surfactants [16, 17]. For ionic surfactants, at the temperature below the Krafft point, the surfactant solubility is significantly lower and micelles does not form [16]. At temperature above the Krafft point, the surfactant solubility rapidly increases and micelle formation occurs [16]. Temperature has the opposite effect on the non-ionic surfactants [16]. When temperature is increased, the surfactants’ solubility is reduced and the solution will become turbid at a point, which is referred to as cloud point [18, 16]. The solution will begin to phase separation [18, 16, 19].

1.5.2.2 Impact of Added Salt

Salt concentration affects the c.m.c formation especially for ionic surfactants [16, 20, 21]. Actually, the salt concentration effect is still small for non-ionic surfactants with respect to ionic surfactants; however, the effect is significant [16]. The effect of the salt concentration on c.m.c is demonstrated as follows [16];

\[
\log(c.m.c) = b_2 + b_3 C \quad \text{(non-ionic)}
\]

\[
\log(c.m.c) = b_4 + b_5 \log C \quad \text{(ionic)}
\]

where, \( b_j \) constants depend on the nature of the electrolyte. For the ionic surfactants, when salt concentration is increased, the repulsive electrostatic force increases between the head groups lowering the c.m.c [16, 14].
For ionic surfactants (E$_{40}$-Foldon head groups have negative charge), when salt concentration is increased, the effective head group size is reduced due to the decrease in Debye length [16]. The shrinking in the head groups causes decreasing micelle surface area compared to volume, which can lead to a changed shape of the micelle such as a cylinder [16].

For nonionic surfactants, the salt acts like an electrolyte in the solution [7]. The effects of the salt for nonionic surfactants are explained by the notion of ‘salting in’ and ‘salting out’ [16, 14, 22]. When the solution contains salt, the water molecules tend to dissolve salt molecules [16]. However, the water molecules are needed to dissolve to the hydrophilic part of the surfactants [16]. In the salted case, the water available for this is reduced [16]. Therefore, the surfactants’ solubility and c.m.c. is reduced by salting out. However, contain salts exhibit which is salting in the opposite behavior and c.m.c. is increased [16].

1.5.2.3 Impact of Head Group and Chain Length

C.m.c is related to chain length of the surfactants, and the relationship is given by the following equation [16, 14, 20]:

\[
\log_{10}\text{c.m.c} = b_0 - b_1m_c
\]

Where $b_0$ and $b_1$ are constant, $m_c$ is the number of carbon atoms in the chain for surfactants which consists of hydrocarbon tails [16]. The previous studies show that the nature of the head group can affect the value of $b_0$ and $b_1$, however, $b_1$ is significantly affected by the head group. Nonionic surfactants generally have larger $b_0$ value than ionic
surfactants, but despite that, nonionic surfactant c.m.c. values are lower than ionic surfactants [16]. Furthermore, variation of the hydrocarbon chain generally affects the c.m.c. formation and this effect usually tends to increase the c.m.c [16, 14]. The variation can be such as introduction branching, or double bonds, or polar functional group along the chain [16].

1.5.2.4 Impact of Organic Molecules

Quite small amounts of organic molecules significantly affect c.m.c. [23, 14, 16], and aqueous solutions of sodium dodecyl sulphate (SDS) can be given as a traditional example for the effect [16]. In aqueous solutions, SDS causes reduction in surface tension because of competing effects of adsorption of dodecanol at the air-water interface and in the SDS micelles [16].

1.6 ELP–Foldon (MGH(GVGVPGEVGP)(GVGVP)_{41}GWP-Foldon)

Elastin-like polypeptides (ELPs) consist of repeats of the pentapeptide (GβGαP)_n, and α which is in the parenthesis can be any of the 20 naturally occurring amino acids, β can be any of those amino acids except for proline, n is the repeated number of the monomer [19]. Since the side chain of the proline is bonded covalently to the nitrogen atom of the peptide backbone, it does not have amide hydrogen to use as a donor in hydrogen bonding [5]. An important characteristic of these polypeptides is their LCST (lower critical solution temperature) behavior, which are thermally responsive polymers exhibit phase separation above a transition temperature (T_t) [19]. The polymer is soluble in water below T_t, and when the temperature is increased the polymer shows aggregation [19].
Figure 4 The figure shows the polymer’s arms aggregation with respect to temperature [19].

Above $T_t$, the coacervate phase, which is viscoelastic and dense, is formed. Coacervation process is reversible, and two components which are water and polypentapeptide form at above $T_t$. Below the $T_t$, the polymer (ELP) is going to be soluble in the water and does not show surfactant properties. And the polymers disperse as a homogeneous in the solution. However, this case can cause changing the surface tension since the solvent is not becoming pure.

Above $T_t$, the polymer is insoluble in the water and shows surfactants properties. The trimer which is aggregated at elevated temperatures gains hydrophobic features. In this case, the polymer going to toward solvent surface, polymer tail which is trimer is located air and head group of the polymer is located at the solvent surface. When the surfactants concentration reaches the c.m.c point, the polymers form micelles in the solution.

1.7 Thermodynamics of ELP-foldon

ELP-foldon consists of three $MGH(GVGPGEGV)(GVGVP)_{41}GWP-GYIPEAPRDGQAYVRKDGEWVLSTFL$ polymers, which are held together though the
trimer forming ELP-foldon head group (foldon) [23]. The ELP chains form the tails, and because of their three tails, they are referred to a three-armed star polypeptide [23, 18]. Since the ELP-foldon tails show hydrophobic properties at elevated temperatures and the head group shows hydrophilic properties, ELP-foldon can be used as a surfactant in liquid [23]. Because of ELP’s thermally responsive features, by increasing temperature, the arms undergo conformational changes and they encourage micelle formations [23]. Prior studies show that micelle formations are observed in low salt above the transition temperature (T_t) of the ELP. However, at physiological salt concentrations above the T_t, turbidity, which is measured with UV-vis spectroscopy occurs [18]. In ELP-foldon system, micelle formations depend on the system pH, salt concentration, polarity, and the ELP molecular length and size [1, 16]. The molecular length and size also affect viscosity of the ELP [1]. Since larger molecules interact with liquid molecules more, the molecular movement becomes slower, and that causes diffusion of the molecules to be slower. In addition, as observed on the previous studies, polymers which have the biggest molecular size affect surface tension of the system when the polymers reach a certain coverage number in the surface [6]. Thereby, occurrence of the ELP adsorption on the surface is time dependent, related to the diffusion coefficient, resulting in time variation between different molecular size ELPs for adsorption of the ELP on the surface. Furthermore, since temperature affects diffusion velocity, time variation can be observed at different temperatures.
In Figure 5, ELP-foldon polymer surfactants are illustrated in three states. Each state is explained below using general thermodynamic properties. Notably, since the head groups and the tails have different chemical properties, they will be examined separately. Sum of head group contributions to Gibbs free energy of mixing is close to zero, so it is assumed to be negligible. Changes of the enthalpy are small between the each state since the intermolecular bonds are not changed. However, the changes of the orientation of the molecules and hydrophobic effects cause changes of entropy which dominant the process.

In State 1, the surfactant tails and head groups are in solution. Entropy ($\Delta S$) of the water molecules which are closer to the tail will be smaller due to the water molecules’ order which is increased. Water molecules are bonded to each other by hydrogen bonds. When the surfactant is located in the water, the surfactant takes the water molecule's place. Thereby, water molecules lose H-bonds and they prefer to gain the bonds again. Otherwise, due to the tail being hydrophobic, water molecules do not H-bond with the tail. In this case, the water molecules are going to bond to each other but, firstly, the water molecules have to be in the appropriate position, and hence they need to

**Figure 5** The surfactant entropy change is explained for each section. 1) In the water, the surfactants are in the free form. 2) The surfactants are in the aqueous solution-air interface. 3) The surfactants are in the micelle form.
be reoriented (conformational change) to bond to each other. The event causes increasing order (decreasing of entropy) in the solvent (water). This is called the hydrophobic effect. Moreover, enthalpy ($\Delta H$) of the solvent molecules which is closer to the tail is increased. Furthermore, the tail of the molecule is located in the bulk, so disorder (entropy) and enthalpy is greater than when located on the surface, since the bulk surfactant has more orientation state than to be located on the surface of the solvent.

The water molecules interact with the head group due to the head group being hydrophilic. Thereby, the head group disorder entropy is increased compared to the beginning conditions. Enthalpy is structural stored energy of the matter. And it is described as the sum of the internal energy and potential energy of the matter. Therefore, enthalpy depends on the molecular bonding energy.

In State 2, the hydrophobic tails of the surfactants are located in the air and hydrophilic head group is located in the solution (Figure 5). The surface tension is reduced because the head groups interact to satisfy the lack a bonding compared to bulk water molecules. Also, the surface molecules because disordered by interacting with the surfactant. Compared State 1, the water molecules that were surrounding the tail will be released to increase disorder, resulting in higher entropy. However, the tail is confined at, the interface decreasing its entropy. The surface molecules have more order (less entropy), but the overall system has less order (more entropy) because of the disordering of the water.

The surfactants assemble as micelle in State 3. There is limited surface available for the surfactants to occupy State 2 and reduce their energy. The concentration at which surface saturation is reached is referred to as the critical micelle concentration (c.m.c).
The formation of micelles is similar to State 2 in that the hydrophobic tails are separated from the water by the hydrophilic head group the molecules become slightly more order than in State 2. It leads to reduce the entropy with respect to surfactants which are in the surface. Since the surface reaches the maximum capacity, when surfactant concentration is increased in the system, the surface tension will be constant. In other words, surface tension is not changed by concentration, and the added surfactants form micelle in the system.

<table>
<thead>
<tr>
<th>S</th>
<th>H</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tail</td>
<td>Head</td>
</tr>
<tr>
<td>Solvent</td>
<td>Molecule</td>
</tr>
<tr>
<td>1 max</td>
<td>Low</td>
</tr>
<tr>
<td>2 surface</td>
<td>High</td>
</tr>
<tr>
<td>3 micelle</td>
<td>High</td>
</tr>
<tr>
<td>For micelle</td>
<td>Favorable</td>
</tr>
</tbody>
</table>

In this study, at different temperatures and different concentrations, the effect of ELP on the surface tension is investigated. Also, according to the ELP’s structure, diffusion of the ELP to the surface and the diffusion’s dependency is viewed. Furthermore, c.m.c of the ELP-foldon and its dependency of temperature are investigated.
CHAPTER II
MATERIALS AND METHOD

2.1 Expression and Purification of ELP-Foldon

The protein-based polymer surfactant used in these experiments is ELP-foldon (MGH(GVGVPPEGVP)(GVGVP)$_{41}$GWP-GYIPEAPRDQAYVRKDGEWVLLSTFL). It consists of 43 pentapeptide repeats, all of them except one are GVGVP. One of the N-terminal pentapeptides is GEGVP. The substitution of the glutamic acid introduces a negative charge at neutral pH to counteract the positive charge of the N-terminus. This allows micelle formation at neutral pH. The ELP-foldon is produced in an E. coli expression system. Cultures are prepared by adding a small sample from a frozen bacterial stock to 10 ml Luria Broth (LB) with 100 μg/mL ampicillin. The culture is left overnight in an incubator which is shaking at 37 °C. LB culture medium is prepared by adding 10 g peptone, 5 g NaCl, and 5 g yeast extract to 1 L purified water. The medium is put in the autoclave at 121 °C for around 60 minutes. After cooling, 100 μg/mL ampicillin is added to the medium. A 1 ml sample is taken from the medium to generate a reference point for optical density (O.D). Then, the overnight culture is transferred to the medium. Until the OD has reached a desirable point which is around 1.0, the medium is kept in the
incubator which is shaking at 37 °C. After this point, the bacteria number has reached a desirable number to induce expression by the addition of 0.24 g/l isopropyl β-D-1-thiogalactopyranoside (IPTG). The culture is kept 4-5 hours in the incubator which is shaking at 37 °C. To harvest the cells, the culture is centrifuged 20 minutes at 2-3 ⁰C, 14000 xg to obtain pelleted bacteria. To purify the ELP from the bacteria, the pellet is resuspended adding phosphate buffered saline (PBS) and the cells are lysed by sonication. Centrifugation is carried out both cold and hot to utilize the thermally responsive behavior of the ELP to purify it. Cold centrifugation is performed 20 minutes at 2-3 ⁰C and 20400 xg and hot centrifugation processes is performed 20 minutes, at 43-45 ⁰C and 7700 xg. After the sonication process, the centrifugation process steps are first cold centrifugation, first hot centrifugation, second cold centrifugation, second hot centrifugation, and third cold centrifugation, resulting with the final protein in the supernatant.

The concentration of the final protein is measured by UV-light absorption at 280 nm using a spectrophotometer. To convert absorbance to concentration, Beer's law is used. Aromatic side chains (tryptophan (W), tyrosine (Y), and cysteine (C)) absorb the UV-light [24]. The absorbance and the concentration are related linearly, through an extinction coefficient as expressed by Beer’s law [24]. The extinction coefficient for the ELP-foldon is 13980 M⁻¹cm⁻¹.

SDS-PAGE (sodium dodecyl sulfate-polyacrylamide gel electrophoresis) is used to verify molecular weight and purity. For the SDS-PAGE process, 15 μl protein sample, 3 μl 6 × dye, and 5 μl marker are used. To observe the trimer formation of the protein, the solution is either boiled or unboiled prior to addition to the gel. The gel is submersed
in buffer solution applying 100 V electrical voltages. After an hour, the gel was removed from the buffer solution, rinsed, and stained with coomassie blue.

2.2 Surface Tension Apparatus

To measure surface tension, several methods are available in the literature such as maximum pull on a rod (Du Noüy-Padday), Wilhelmy plate, Du Noüy ring, spinning drop, bubble pressure, and drop shape methods. We used a pendant drop shape methods. The methods, which do not depend on the contact angle [25], are based on the analysis of drop shape which is obtained from the shape of a sessile drop, pendant drop or captive bubble to determine the liquid–vapor or liquid–liquid interfacial tensions. The shape of a drop is determined by a combination of surface tension and gravity effects [26]. Drop shape methods can be used in many difficult experimental conditions since they have a lot of advantages in comparison to the other techniques [26].

To obtain pendant drop image, a Ramè-Hart Goniometer/Tensiometer is used. The tensiometer consists of temperature controller, CCD camera (home built CCD camera with computer capture is used), fiber optic light source, environmental chamber, chamber cover with stage, elevated temperature syringe, glass syringe, stainless steel needle, film clamps, microsyringe fixture, and base. The temperature controllers provide accurate temperature control on the environmental chamber and the elevated temperature syringe. The environmental chamber also protects the drop from adverse effects. The glass syringe is assembled into the elevated temperature syringe that keeps the glass syringe and its contents at a controlled temperature. The drop is formed by an adjustable screw of the apparatus applying pressure to the plunger. The melting point of the materials is up to 230 °C [27], although we made measurements only up to 75 °C. The
chamber cover can be tilted to align the sample. The microsyringe fixture holds an elevated temperature syringe, and it can be adjusted in all directions. All of the components are assembled on the base. Details about the CCD camera, the fiber optic light source, and the needle are described in the image analysis section 2.3.2. The schematic diagram of the parts is illustrated in Figure 7.

2.3 Axisymmetric Drop Shape Analysis (ADSA)

Surface tension determination by axisymmetric drop shape analysis (ADSA) was first introduced by Bashforth and Adams and it continues to this day [28, 29]. A second generation of ADSA was developed by del Río [29, 30, 31] using the curvature at the apex rather than the radius of curvature and the angle of vertical alignment as optimization parameters [26]. A flowchart (Figure 6) shows the general procedure of ADSA to measure surface tension. ADSA uses drop interface properties which are obtained from the shape of pendant drops or sessile drops found by analyzing the images. The coordinate profile properties (i.e. the experimental profile) of the drops are obtained for use in numerical optimization processing. After that, the experimental properties of the drop and physical properties such as density and gravity are used to fit a series of Laplacian curves to obtain liquid–fluid interfacial tension, contact angle (in the case of sessile drops), drop volume, surface area, radius of curvature at the apex, and the radius of the contact circle between the liquid and solid (in the case of sessile drops) [26]. The images were analyzed using MatLab® codes. The codes were written by Eric Helm loosely based on code found in literature [25].
Figure 6 General procedure of Axisymmetric Drop Shape Analysis (ADSA).

2.3.1 Image Capture

Figure 7 Schematic diagram of the experimental setup of ADSA for analysis of pendant drop. Modified M. Hoorfar et al. [26].
To analyze a drop image, the image is obtained from the experimental setup of ADSA shown in Figure 7. A fiber optic light source is used behind the drop to improve image contrast. A Vivicam 3750 CCD camera was used to obtain images. Each pixel of the images consists of bits which describe gray scale. Mathematically, the relationship between expressions is showed in the following equation:

$$L = 2^k$$

where $L$ is the number of gray levels or the shades of gray and $k$ is bits per pixel (bpp). The camera resolution is $2048 \times 1536$ pixels and 24 bits hence gray levels is $2^{24}$ or 16,777,216 different shades. The picture is analyzed and the appropriate parameters are solved based on the Young-Laplace equation.

2.3.2 Image Analysis
2.3.2.1 Edge Detection

The image analysis process first begins by the software edge detection procedure [26, 32, 33], consisting of three steps (Figure 8) [26]. The pendant drop image is loaded into MatLab® as an original image (Figure 9A). To improve visibility of the drop in the
image, the image is changed to grayscale then binary using a threshold. By this method, an image consisting of black and white pixels is obtained (Figure 9B).

![Original Image](image1.png)  ![Binary Image](image2.png)

![Image After Edge Detection](image3.png)  ![Image After Boundary Trace](image4.png)

**Figure 9** Drop image for steps of image analysis process  
A) The drop image of 1 μM solution of ELP-foldon is at pH of 7.4, 10th minute, and a salt concentration of 25 mM.  
B) Binary image is obtained using threshold.  
C) The image is after edge detection. It is obtained using Canny operator.  
D) Final drop profile image is obtained after boundary trace process.

The original image contains noise and useless information, hence to reduce the noise and useless information preserving important information, filter and edge detection
operators are applied to the image [34]. All edges are detected using the MatLab® edge
detectors, Canny [26]. Figure 9C shows edge points after the edge detection operator is
applied on the drop image.

Sobel edge detection was also attempted. It is one of the most well-known image
processing algorithms [26, 34]. Two convolution kernel algorithms (3 × 3) are used in
the Sobel [26, 34]. While one of them is used to find the horizontal edges, another one is
used to find vertical edges. Basically, these two kernels are perpendicular to each other
(Figure 10) [34]. These kernels help to determine sudden light intensity change within the
image.

\[
G_x = \begin{bmatrix}
-1 & 0 & +1 \\
-2 & 0 & +2 \\
-1 & 0 & +1 \\
\end{bmatrix}
\]

\[
G_y = \begin{bmatrix}
+1 & +2 & +1 \\
0 & 0 & 0 \\
-1 & -2 & -1 \\
\end{bmatrix}
\]

Figure 10 The kernels’ horizontal (\(G_x\)) and vertical (\(G_y\)) derivative
approximations [34].

Basically, the image is divided into (3×3) pixels in size, and then gradients are
calculated applying the kernels separately to the image [34]. The gradients are combined
together to describe exact gradient magnitude [26, 34]. Gradient magnitude is calculated
using the following equation:

\[
|G| = \sqrt{G_x^2 + G_y^2}
\]

The gradient direction is calculated by:

\[
\theta = \arctan\left(\frac{G_y}{G_x}\right)
\]
According to studies, the Canny edge detection technique performs better compared to other techniques for surface tension measurement [26, 34] therefore, we used the Canny edge detection technique. The Canny uses Gaussian filter to eliminate the noise and useless information from the original image using standard convolution methods [26, 34]. Thereby, a smoothed image which is purified from noise is obtained. The gradient intensity and direction are computed for the image [26, 34]. Non-maximum suppression is applied to reduce thick edge responses to thin lines [34, 35]. Hysteresis is used to determine beginning and end points of the edge using two different threshold values, a high and a low [26, 34].

In our diagram, a high resolution image is used. In the high resolution image, the intermediate values exist because of pixel density. Thereby, boundary trace methods are applied directly to the image to obtain drop profile, which is illustrated in Figure 9.D.

Since the unit of the experimental edge value is in pixels, it needs to be converted to millimeters to compute the theoretical Laplace equation [26]. A calibration grid was used to verify this procedure to convert pixels to millimeters. In this process, the diameter of the syringe needle is used as a reference to determine the ratio of millimeters to pixels. The needle diameter was measured as 0.72 mm. The number of pixels across the needle is determined for a row of pixel, yielding the number of pixels for 0.72 mm, the process is applied for 20 rows of pixels, and the average is taken to calculate the conversion to millimeters.

Once the capillary diameter is calculated, the Z-axes height cut-off point is selected. The number of analyzed points is changed with respect to the cutoff point. Thereby, the drop curve, which will be analyzed, is obtained (Figure 11). In our study,
the cut-off point level was selected close to the needle, and when different cut-off points were selected close to the needle, the variability of the surface tension value was insignificant.

![Figure 11](image)

**Figure 11** The red line is shown drop profile after cut-off point is selected.

To observe and correct for the camera and experimental setup errors, vertical symmetric axis of a pendent drop image curve is found, and a midpoint line is formed based on the midpoint of several horizontal values (Figure 12A). The red line which is shown in Figure 12B. is formed basing each horizontal pixel point coordinate of the drop edge as vertical symmetrical axis of a pendent drop image curve. The profile is divided into two parts by the midpoint line. However, if the camera and drop were not in alignment, the midpoint line would not divide the drop curve symmetrically. In other words, the coordinates of pixel points \((X_i, Z_i)\) on the same line are not exactly equidistant
from the midpoint line. When the blue line is not exactly described by the software, it means that the edge coordinate points are not described well and its reason can be light errors or camera’s sharpness. If a red line forms a curve close to the holder, the error is caused from the curvature of the camera.

![Figure 12](image)

**Figure 12** a) The blue line is midpoint of drop profile b) The red line is midpoint of each curve point.
2.3.3 Drop Shape Calculation

The balance between surface tension and external forces is described mathematically using a set of initial parameters which are fit to the drop profile [26] by the Young-Laplace equation of capillarity,

\[ \Delta P = \gamma \left( \frac{1}{R_1} + \frac{1}{R_2} \right) \]  

(1)

where \( R_1 \) and \( R_2 \) are the two principal radii of curvature, \( \gamma \) is surface tension of the drop, and \( \Delta P \) is the pressure difference across the liquid interface. If there is not any external force except gravity in the surrounding environment of the drop, \( \Delta P \) can be expressed as a linear function of the elevation:

\[ \Delta P = \Delta P_0 + (\Delta \rho)gz \]  

(2)

Figure 13 The image shows the distribution of the parameters geometrically on the drop [26, 52].

The coordinate system shows direction of axises.
where \( \Delta P_0 \) is the pressure difference at a reference plane and \( z \) is the vertical coordinate of the drop measured from the reference plane. Also, when the value of \( \gamma \) is given, by inverse calculation, the shape of the drop can be determined [30].

\[
\Delta P_0 + (\Delta \rho)gz = \gamma \left( \frac{1}{R_1} + \frac{1}{R_2} \right)
\]  

(3)

Two principal radii are determined by two planes which are defined at any point of a curved surface \((X_i, Z_i)\) [26]. One of the planes passes through the surface, and a curve is generated between the plane and the surface containing a normal [26], thereby, the first radius of curvature is generated [26]. To describe the second radius of the curvature, another plane is passed through the surface being perpendicular to the first plane [26]. Under the assumption of axial-symmetry (between the interface and \( z \)-axis), the principal radius of curvature, \( R_1 \), is related to the arc length, \( s \), and the angle of inclination of the interface to the horizontal, \( \Phi \), by [26, 30, 36, 37]

\[
\frac{1}{R_1} = \frac{d\Phi}{ds}
\]  

(4)

The second radius of curvature is given by [26]

\[
\frac{1}{R_2} = \frac{\sin \Phi}{x}
\]  

(5)

Figure 13 represents the ADSA coordinate system. In this system, “mean curvature” is described by summing \( \left( \frac{1}{R_1} + \frac{1}{R_2} \right) \) of two principal radii of curvature [26]. The drop’s apex
curvature is defined as “b” [31], and because of the axial-symmetry, at the apex, the b value is constant in all directions and the two principal radii of curvature are equal, i.e.,

\[
\frac{1}{R_1} = \frac{1}{R_2} = \frac{1}{R_0} = b
\]  

(6)

where, \(R_0\) is the radius of curvature [26]. At the apex, the arc length, s, is equal to zero [26]. Thereby, in this point, the pressure difference is expressed using equation 1 as [26]

\[
P_0 = \frac{2\gamma}{b}
\]  

(7)

The following boundary-value problem is obtained as a function of the functions of the arc length, s, using equation 4, 5 and 7 into equation 1 [26]

\[
\frac{d\Phi}{ds} = \frac{2b}{b} + Cz - \frac{\sin \Phi}{x}
\]  

(8)

\[
C = \frac{(\Delta \rho)g}{\gamma}
\]  

(9)

where c is a capillary constant, and because the gravity, g, has positive values for sessile drops and negative values for pendant drops [26].

Equation (8) together with the geometrical relations [26]

\[
\frac{dx}{ds} = \cos \Phi
\]  

(10)

\[
\frac{dz}{ds} = \sin \Phi
\]  

(11)
form a set of first order differential equations for $x$, $z$, and $\phi$ as functions of the arc length, $s$, with the boundary conditions [26]

$$x(0) = z(0) = \Phi(0) = 0 \quad (12)$$

Also, at $s=0$

$$\frac{d\phi}{ds} = b \quad (13)$$

The Laplacian axisymmetric fluid–liquid interface curve was generated by solving these equations numerically [25, 26] using a Runge-Kutta method [36, 38, 39, 40, 41] for given values of $b$ and $c$ [26]. We programmed this using the ODE45 function in MatLab®. Dimensionless parameters were substituted into Equation 8. The values are normalized using apex curvature of the drop, $b$ [25]:

$$\bar{x} = \frac{x}{b}$$

$$\bar{s} = \frac{s}{b} \quad (14)$$

$$\bar{z} = \frac{z}{b}$$

geometric consideration [25]

$$\frac{d\Phi}{d\bar{s}} = 2 + c\bar{z}b^2 - \frac{\sin \Phi}{\bar{x}} \quad (15)$$

$$\frac{d\bar{x}}{d\bar{s}} = \cos \Phi \quad (16)$$

$$\frac{d\bar{z}}{d\bar{s}} = \sin \Phi \quad (17)$$
The initial conditions [25],

\[
\bar{x}(0) = \bar{z}(0) = \phi(0) = 0
\] (18)

A theoretical curve generated using the Young-Laplace equation is illustrated in Figure 14.

![Figure 14](image)

**Figure 14** Drop profile analysis is used for the fitting process. The red curve is a theoretical curve and is generated from the Young-Laplace equation, the black is an experimental curve and is obtained from the picture. The green line is a rotated curve with respect to the original image.

### 2.3.4 Optimization

The values b and c are defined from the experimental profile, and these values are used in the Laplace equation to generate a theoretical profile. The two profiles are mapped and errors are found using error function, \( e_i \). For each experimental data point \((X_i, Z_i)\), the closest to the theoretical curve point \((x_i, z_i)\) is selected, and the distance
between these points, \(d_i\), is calculated [26]. The error function is defined as \(e_i = \frac{1}{2}d_i^2\) [26, 29, 42].

\[
e_i = d_i^2 = (x_i - X_i)^2 + (z_i - Z_i)^2
\]  \hspace{1cm} (19)

The fitting process minimizes the objective function, \(E\), which is described as the sum of the individual errors squared. The function contains the fitting parameter, \(q\), with elements \(q_k, k=1, \ldots, M\). Best fit between the experimental points and a Laplacian curve is obtained finding \(q\) values that minimize \(E\). A point is necessary in order to calculate the objective function and is assumed a minimum value at the point in the maximum \(M\) value. The objective function is a defined function of a set of parameters at following.

\[
\frac{\partial E}{\partial q_k} = \sum_{i=1}^{N} \frac{\partial e_i}{\partial q_k} = 0, \hspace{0.5cm} k = 1, \ldots, M
\]  \hspace{1cm} (20)

The objective function consists of nonlinear algebraic equations, hence an iterative solution is required using numerical solver such as Newton–Raphson method, Levenberg–Marquardt method and Nelder-Mead simplex method [26, 36]. While first generation of ADSA uses Newton–Raphson method, second generation uses Newton–Raphson method and Levenberg–Marquardt method together [26, 36]. In our program, we used the Nelder-Mead method as a numerical solver using MatLab®. Figure 15 shows the final drop curve of after the optimization processes and the residuals of the fit. Nelder-Mead is a simplex method that is used to find a local minimum point of a few variable functions [43]. For two variables, the simplex forms a triangle and it is a method
of comparing the value of the functions in the three vertices of the triangle. The function, where value is the largest peak value is rejected and a new peak value is determined. Thereby, a new triangle is created and the process is continued. The coordinates of the minimum points are found reducing the size of the triangle. The algorithm is created using the simplex term, and the minimum point of the function of N variables will be found by this algorithm in N dimensions.

a. 

Figure 15 a) The final drop curve results, theoretical and experimental curve are overlapped by optimization processes. b) The residual plot is formed by remaining from the difference between the theoretical and the experimental curve.

At the minimum error point, the value of b and c are determined. A graph illustrating error as a function of c and b gives an idea of the error sensitivity to the determined values (Figure 16). Surface tension is calculated substituting c, g and Δρ values into Equation 9. The obtained results for this example are shown in Table 2.1.
Surface tension was measured at different polymer concentrations (from 10 nm to 50 μM) and temperature (from 25 to 35°C). Before we used the ELP-Foldon solution, the solution was filtrated. Each concentration was prepared from highest concentration to lower concentration and the concentration was measured using UV-spectroscopy at 280 nm.
nm to obtain precise concentration values. The solutions were taken from the solution tube using a glass syringe. After the syringe was assembled to the tensiometer, 5 minutes elapsed for solution to reach desired temperature. The tensiometer and the camera were calibrated taking sample drop pictures and analyzing with the MatLab® code. To reduce evaporation from the drop, the humidity in the chamber was increased by placing two drops of solution on the chamber surface. An experimental drop was created immediately after a previous drop was dropped and timer was started. To obtain pictures, the camera utilized auto-shooting mode with a 10 second delay to avoid vibrations. During each drop, a picture was taken every 30 seconds. For each concentration, at the same temperature, the experiment was repeated at least three times.

In order to validate the experimental method, experimental surface tension values of water were compared to values reported by N.B. Vargaftik et al [44] (Figure 17). As is seen from the graph, the experimental values and literature values are close to each other up to 55 °C. For the experiment, drop images were taken by the camera in 10 °C increments. The experimental water surface tension value is obtained by averages of the three images at each temperature point.
Figure 17 The graph shows a comparison of literature values of surface tension of water and experimental values of water as function of the temperature.
CHAPTER III
RESULTS AND DISCUSSION

The surface tension as a function of temperature was measured for seven different concentrations. The behavior varied with concentration resulting in three general cases for lower, intermediate, and higher concentrations.

At the lower concentrations, ($\leq 100 \text{ nM}$) the surface tension was not affected by the protein within the time of the experiment. The samples were measured at times as long as 2 hours with no observed change in the surface tension. It is possible that there was not sufficient time for the protein to diffuse and create a monolayer at the surface. The change of surface tension observed as a function of temperature is equivalent to the surface tension change of the pure solution (Figure 18).
At intermediate concentrations, between 0.2 and 1.0 µM, the surface tension

Figure 18 Measured surface tension as a function of temperature for solutions of different concentrations of ELP-foldon at pH 7.4 and a salt concentration of 25 mM PBS. A) 10 nM, B) 31.6 nM, and C) 0.1 µM.

At intermediate concentrations, between 0.2 and 1.0 µM, the surface tension
varied with time and temperature (Figure 19A, 20A, 21A, 22A). The behavior is similar to what was observed by Hua-Rosen for surfactant adsorption. There was an initial period of small decrease in the surface tension followed by more rapid decrease to a more stable meso-equilibrium value. The initial and meso-equilibrium values varied with temperature comparable to the solvent value (Figure 19, 20, 21, 22) and the times required to reach the meso-equilibrium decreased with increased temperature and concentration.

For example, Figure 19A shows that at 0.2 μM polymer concentration, the surface tension change is a function of time and temperature. At 25 °C, for the first the 10 minutes, the surface tension was approximately equal to the solvent surface tension (~72 mN/m). A surface tension decrease was observed after approximately 10 minutes, and the decrease continued until about 23 minutes. After this, change was not observed. At this time, the surface tension was ~50 ± 1.5 mN/m. At 30 °C, a drop in surface tension began in less than 10 minutes. The reduction continued between 10 and 20 minutes, after which, change was not observed. At this time, the surface tension was determined as ~50 ± 1.5 mN/m. At 35 °C, the surface tension decrease began at the fourth minute and continued until 18 minutes when the surface tension remained constant at ~49 ± 1.5 mN/m.
Figure 19 A) Measured surface tension as a function of time for a 0.2 μM solution of ELP-foldon at pH of 7.4 and a salt concentration of 25 mM PBS at different temperatures. B) Measured surface tension as a function of temperature for a 0.2 μM solution of ELP-foldon at pH of 7.4 and a salt concentration of 25 mM.
Figure 20  A) Measured surface tension as a function of time for a 0.316 μM solution of ELP-foldon at pH of 7.4 and a salt concentration of 25 mM PBS at different temperatures. B) Measured surface tension as a function of temperature for a 0.316 μM solution of ELP-foldon at pH of 7.4 and a salt concentration of 25 mM.
Figure 21 A) Measured surface tension as a function of time for a 0.1 μM solution of ELP-foldon at pH of 7.4 and a salt concentration of 25 mM PBS at different temperatures. B) Measured surface tension as a function of temperature for a 0.1 μM solution of ELP-foldon at pH of 7.4 and a salt concentration of 25 mM.
At higher concentration (50 $\mu M$), the surface tension had reached meso-equilibrium at the first time point measured for all temperatures. The meso-equilibrium surface tension decreases with temperature comparable to the solvent as is indicate by a constant surface pressure (Table 3.1)

Figure 22A was obtained at 50 $\mu M$ polymer concentration, at pH of 7.4 and a salt concentration of 15 $mM$ PBS as a function of time and temperature. The surface tension reached meso-equilibrium quickly between 25 and 50 °C. At this concentration, micelle formation has been observed at 50 °C as shown in Figure 22B, yet no change in surface pressure is observed. The solution and the solvent surface tension values are presented in Table 3.1.

The surface pressure is defined as $\Pi = \gamma_0 - \gamma$, where, $\Pi$ is surface pressure, $\gamma_0$ is solvent surface tension and $\gamma$ is the solution surface tension. The difference between the solution surface tension and the solvent surface tension has been approximately constant for the temperatures studied resulting in a constant surface pressure (Figure 23).
Figure 22  A) Measured surface tension is a function of time for a 50 μM solution of ELP-foldon at pH of 7.4 and a salt concentration of 15 mM PBS at different temperatures. B) Measured surface tension and rate of absorption (UV) is a function of temperature for a 50 μM solution of ELP-foldon at pH of 7.4 and a salt concentration of 15 mM.
Table 3.1 The solution and the solvent surface tension values and surface pressure values at different temperatures for 50 μM.

<table>
<thead>
<tr>
<th>Temperature (℃)</th>
<th>The solution Surface tension (mN/m)</th>
<th>The solvent Surface tension (mN/m)</th>
<th>The surface pressure (γₒ − γ) (mN/m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>25</td>
<td>50.5 ±1.5</td>
<td>72</td>
<td>21.5</td>
</tr>
<tr>
<td>30</td>
<td>49 ±1.5</td>
<td>71.2</td>
<td>22.2</td>
</tr>
<tr>
<td>35</td>
<td>48.5 ±1.5</td>
<td>70.4</td>
<td>21.8</td>
</tr>
<tr>
<td>40</td>
<td>47.8 ±1.5</td>
<td>69.6</td>
<td>21.8</td>
</tr>
<tr>
<td>45</td>
<td>47.2 ±1.5</td>
<td>68.8</td>
<td>21.6</td>
</tr>
<tr>
<td>50</td>
<td>46.3 ±1.5</td>
<td>67.9</td>
<td>21.6</td>
</tr>
</tbody>
</table>

Critical micelle concentration (c.m.c) is a point in which the surface tension is stable and does not change when the concentration and temperature are increased (Table 3.2). For a 50 μM, measurement of rate of absorption (UV) shows micelle formation after 40 °C (Figure 22). Despite micelle formed, surface tension does not change.
Figure 24 In the graph, the surface tension is shown as a function of temperature (°C) for all concentrations. At 10, 31.6, and 100 nM the surface tension did not decrease at the time measured, while at 0.2, 0.316, 1.0, and 50 µM the surface tension had reached meso-equilibrium.
A. 25 °C

Surface Tension (mN/m)

log(time (s))

0.2 uM
0.316 uM
1 uM
50 uM

B. 30 °C

Surface Tension (mN/m)

log(time (s))

0.2 uM
0.316 uM
1 uM
50 uM
**Figure 25** A) Measured surface tension as a function of log(time(s)) for 0.2, 0.316, 1, 50 μM solution of ELP-foldon at pH of 7.4 and a salt concentration of 25 mM PBS at 25 °C. Experimental data is fit by the Hua-Rosen equation to create theoretical curve (—). B) Measured surface tension as a function of time for 0.2, 0.316, 1, 50 μM solution of ELP-foldon at pH of 7.4 and a salt concentration of 25 mM PBS at 30 °C. C) Measured surface tension as a function of time for 0.2, 0.316, 1, 50 μM solution of ELP-foldon at pH of 7.4 and a salt concentration of 25 mM PBS at 35 °C.
At different concentrations and constant temperature, the surface tension depends on time (Figure 25). At 25, 30 and 35 °C, and at 0.2, 0.316 and 1 μM concentrations, formation of the region I, II, III are observed; however, at higher concentration (50 μM), a direct transition to region III is observed. While concentration increases, lag time decreases. When the temperature is increased, the lag time to reach meso-equilibrium surface tension value decreases. As it is seen in the graphs, the meso-equilibrium surface tension value does not change with the concentration or the temperature.

**Figure 26** The half time ($t^*$) as a function of concentration for 0.2, 0.316, 1, 50 μM solution of ELP-foldon at pH of 7.4 and a salt concentration of 25 mM PBS at 25, 30, 35 °C. The slope is measured as -1.2 for graph A and B, and for graph C, is -1.3.
Time dependent surface tension measurement of polymer solution can be fit to find dynamic surface tension parameters (Table 3.2) including initial surface tension \( \gamma_s \) when the drop is formed, meso-equilibrium surface tension \( \gamma_m \), the time the surface tension \( \gamma(t) \) is half-way between \( \gamma_s \) and \( \gamma_m \), \( (t^*) \) and empirical constant \( (n) \) using the Hua-Rosen equation [13, 45]:

\[
\log \frac{\gamma_s - \gamma(t)}{\gamma(t) - \gamma_m} = n \log \frac{t}{t^*}
\]

The \( t^* \) value from the Hua-Rosen equation scales with the bulk concentration with the exponent \( m \) (Figure 26):

\[
t \propto C_b^m
\]

According to Ward and Tordai, for diffusion controlled adsorption kinetics, the scaling exponent is -2 [13]. In our study, the slope obtained for (ELP)_{40}-foldon is -1.2 and -1.3 at 25 and 30 °C and 35 °C respectively (Figure 26). Since our experimental slope is not close to -2, the adsorption kinetics is not exclusively diffusion controlled. The difference between the diffusion controlled slope and experimental slope can be explained by the occurrence of polymer adsorption/desorption barriers [12].

The surface pressure at the meso-equilibrium is found to be \(~23 \text{ mN/m}.\) This surface pressure does not depend on the temperature or the concentration.
Table 3.2 Parameters of the dynamic surface tension of (ELP)-Foldon.

<table>
<thead>
<tr>
<th>Temperature °C</th>
<th>Concentration (μM)</th>
<th>( t^* ) (s)</th>
<th>( Y_s ) (mN/m)</th>
<th>( Y_m ) (mN/m)</th>
<th>( Y_s - Y_m ) (mN/m)</th>
<th>n</th>
</tr>
</thead>
<tbody>
<tr>
<td>25</td>
<td>0.2</td>
<td>927</td>
<td>71.9 ±1.5</td>
<td>49.7 ±1.5</td>
<td>22.3</td>
<td>4.8</td>
</tr>
<tr>
<td></td>
<td>0.316</td>
<td>676</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>177</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>1.8</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>0.2</td>
<td>1046</td>
<td>70.9 ±1.5</td>
<td>49.2 ±1.5</td>
<td>22</td>
<td>4.8</td>
</tr>
<tr>
<td></td>
<td>0.316</td>
<td>452</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>152</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>1.4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>35</td>
<td>0.2</td>
<td>772</td>
<td>69.5 ±1.5</td>
<td>47.5 ±1.5</td>
<td>22.9</td>
<td>4.8</td>
</tr>
<tr>
<td></td>
<td>0.316</td>
<td>429</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>145</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>0.6</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

3.1 ELP-Foldon’s Diffusion

Figure 27 The figures show the movement of the polymer to the air-aqueous surface. A 0.316 μM solution of ELP-foldon at pH 7.4 and a salt concentration of 25 mM PBS below the transition temperature. A) The polymers amount is not enough to reduce surface tension hence the surface tension value is approximately equal to PBS’s surface tension, region I. B) The surface has enough polymers to reduce surface tension, region II. C) The surface has reached enough polymers’ amount to form multilayer, region III.
The polymer moves from one point to another point by diffusion through Brownian motion in the aqueous solution. The polymers accumulate at the air-water interface because the adsorbed protein results in a lower energy system. The movement of the polymer to the interface requires time. The time required is inversely proportional to the diffusion velocity [46]. Figure 27 illustrates the regions (I, II, III) formation of the (ELP)$_{40}$-Foldon at a 0.316 μM solution. Region I is the induction time and at 25 °C is between 0.5 and 5.5 minutes (0 minute is the solution’s first interaction time with air), surface tension is closer to pure water surface tension and since there is not enough, the polymers move to the air-aqueous surface, as illustrated in Figure 27A. Region II is rapid fall region, at 25 °C, between 5.5 and 15.5 minutes, decreasing of the surface tension is observed since enough polymers have been arrived to the surface forming a monolayer, illustrated in Figure 27B. Region III is meso-equilibrium region, after 15.5 minutes, the surface tension change is not observed since the surface reaches enough polymer saturation forming a multilayer, illustrated in Figure 27C.
CHAPTER IV

CONCLUSIONS

Surface tension of the polymer ELP-Foldon in PBS solution was measured as a function of temperature, time, and various concentration using a drop shape tensiometer. At lower concentrations (10, 31.6 and 100 nM), the surface tension was approximately equal to PBS surface tension as a function of temperature. Therefore, effect of the polymer on the surface tension change can be thought negligible at the lower polymer concentration. At higher concentrations (0.2, 0.316, 1 and 50 μM), the surface tension was reduced by the polymer. However, the decrease was observed to be dependent on time, exhibiting three characteristic regions. It is noteworthy that at all the concentrations and temperatures, the surface tension values were approximately equal to each other around 49 ± 1.5 mN/m resulting in a surface pressure of ~23 mN/m that does not vary with concentration or temperature. It is also observed that elapsed time to reach meso-equilibrium surface tension was dependent upon the concentration and temperature. When temperature and the concentration were increased, the time to reach equilibrium decreased since polymer diffusion is increased by temperature, and the probability of the presence of the polymer in the region close to the surface is increased by the
concentration; thus, they are adsorbed more rapidly since distance to the surface is decreased.

The half time (t*) is shown to scale with concentration with an exponent of -1.2 and -1.3 at 25 and 30 °C and 35 °C, respectively. This suggests that the ELP-Foldon adsorption kinetics does not show exclusively diffusion controlled behavior.

The ELP-Foldon does not show c.m.c. formation point since the polymer-PBS solution surface tension is not affected by the polymer concentration. The concentration affects only the region formation. At higher concentration, regime formation is not observed.


[35] T. Q. Pham, "Non-maximum Suppression Using fewer than Two Comparisons per Pixel," *Canon Information Systems Research Australia (CiSRA)*.
[36] A. Kalantarian, "Development Of Axisymmetric Drop Shape Analysis - No Apex (ADSA-NA)," Doctor of Philosophy Graduate Department of Mechanical and Industrial Engineering University of Toronto, Toronto, 2011.


APPENDIX

Experimental images were analyzed by the MatLab® codes to measure the surface tension.

% pd_run.m
% run file for pendant drop analysis

clc;
clear;

% Input parameters for Pendent Drop Analysis

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%% Experimental Image Parameters %%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Image for Analysis (Color or Greyscale)
img = 'IMAGE.jpg';

% edge detection parameters for edge detector
% ed_type = 1 for c10anny
% ed_type = 2 for sobel errors??????
% ed_type = 3 for bwboundaries with 'noholes' errors???
ed_type = 1;
thresh = [];
sigma = 1;

% number of rows to search for start of bwtraceboundary
line_check = 10;

% calculate capillary diameter
% cap_dia_points = # of points to search and average for capillary diameter
cap_dia_points = 20;

% rotate experimental data
% rotation = 1 too use rotated experimental data
rotation = 1;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%% Theoretical Parameters %%%%%%%%%%%%%%

% Parameters
% c_start = -0.11347;            % mm^-2
b_start = 1.31989;              % mm
cap_dia_units = 0.72;          % mm (1.27)(0.72)
del_row = 0.001;               % g/mm3
g = -9806.4;                   % mm/s^2

% choose optimization solver type
% solver_type = 0 to exit without optimization
% solver_type = 1 for fminsearch (Nelder-Mead)
% solver_type = 2 for Levenberg-Marquardt
% maxiter is number of solver loops
solver_type = 1;
maxiter = 1;

% include residual plot
% residual_plot = 0 for No
% residual_plot = 1 for Yes
residual_plot = 1;

% include error surface plot
% err_surface_plot = 0 for No
% err_surface_plot = 1 for Yes
% deltab is the +- distance to vary b
% deltac is the +- distance to vary c
% nop is the number of data points for each delta
err_sur = 0;
deltab = 0.05; % careful changing this, may cause problems
deltac = 0.05; % careful changing this, may cause problems
nop = 5;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% define all parameters
parameters(1) = ed_type;
parameters(2) = sigma;
parameters(3) = line_check;
parameters(4) = solver_type;
parameters(5) = maxiter;
parameters(6) = del_row;
parameters(7) = g;
parameters(8) = cap_dia_points;
parameters(9) = cap_dia_units;
parameters(10) = residual_plot;
parameters(11) = rotation;

% define error surface parameters
err_surface_para(1) = err_sur;
err_surface_para(2) = deltab;
err_surface_para(3) = deltac;
err_surface_para(4) = nop;

% directing function file
colonial(img, c_start, b_start, parameters, thresh, err_surface_para);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Volume_calc.m

function [vol_total] = vol_calc(x_theor_final, z_theor_final, b2)
x_theor_final_dim = x_theor_final.*b2;
z_theor_final_dim = z_theor_final.*b2;
%[num] = xlswrite(’xz_theor_final_dim.xls’, [x_theor_final_dim
z_theor_final_dim]); % write to file
vol_total =
sum((pi.*((x_theor_final_dim(2:end,:).^2))).*(diff(z_theor_final_dim)))
;
return;

%%%unique_stable.m

function val = unique_stable(input_matrix)

% remove duplicate rows
[values index] = unique(input_matrix, ’rows’, ’first’);

% resort data as ’stable’
out = sortrows([index values]);
val = out(:,2:3);
return;

% unique_extract.m

function val = unique_extract(input_matrix)

% remove first bwtraceboundary data
vall(1,:) = input_matrix(1,:); % adds first value to new matrix
i = 2;
while input_matrix(1,1) ~= input_matrix(i,1)
    vall(i,:) = input_matrix(i,:);
    i = i + 1;
end
vall(i,:) = input_matrix(i,:); % adds last value to new matrix

%disp(vall)

% remove duplicate rows
[values index] = unique(vall, ’rows’, ’first’);

% re-sort data as ’stable’
out = sortrows([index values]);
val = out(:,2:3);
return;

% sur_ten.m
% function calculates surface tension from c, del_row, and g

function st = sur_ten(c_value, parameters3)
del_row = parameters3(1);
\[ g = \text{parameters3}(2); \]

\[ st = (\text{del_row} \cdot g) / c\text{\_value}; \]

\text{return;};

\% residual.m
\% function to plot residual

\text{function residual}(c2, b2, x\text{\_new}, z\text{\_new}, \text{ratio}, \text{height}, \text{residual\_plot})

\text{if residual\_plot == 1}
\text{fprintf('Calculating Residual....\n');}
\text{\% plot residual}
\text{V(1) = c2; V(2) = b2; [dum, zx\_residual] = obj\_fun(V, x\_new, z\_new, \text{ratio}, \text{height});}

\text{figure(11); plot(zx\_residual(:,2), zx\_residual(:,1)); xlabel('Drop Position'); ylabel('Residuals'); title('Residual Plot'); fprintf(' Residual Plot Complete....\n\n');}

\text{end}

\text{return};

\text{function [mid\_return, apex, height, cap\_radius, theta, mid] =}
\text{pd\_sym(b\text{\_matrix}, top\_cut\_off, points, sigma)}
\text{\% this function finds the vertical symmetric axis}
\text{\% of a pendant drop image curve}
\text{\% input requires a matrix of [j i] or [y x] along with}
\text{\% the top cutoff value for scale and number of analysis}
\text{\% points}
\text{\% output is of the matrix type [j i] or [y x]}

\text{\% begin by finding approx vertical height}
\text{j\_max = max(b\_matrix(:,1)); height = j\_max - top\_cut\_off; \% in pixels}

\text{\% find midpoint of several horizontal values}
\text{\% construct loop statement to find midpoint values}

\text{i = 0; while (length(b\_matrix(:,2)) - i) > i}
\text{mid(i+1,1) = (b\_matrix(end-i,2) + b\_matrix(i+1,2))/2; i = i + 1;}
\text{end}
\text{mid\_coor = [mid b\_matrix(1:length(mid),1)]; \% [x y]
% check for outliers then eliminate
s = std(mid_coors(:,1));
min_x = mean(mid_coors(:,1)) - sigma*s;
max_x = mean(mid_coors(:,1)) + sigma*s;

sss = mid_coors(:,1) >= min_x & mid_coors(:,1) <= max_x;
mid_sorted = mid_coors(sss,:);

a1 = length(mid_coors(:,1));
a2 = length(mid_sorted(:,1));

% fit a line to remaining data point
p = polyfit(mid_sorted(:,1), mid_sorted(:,2), 1);

% calc fitted values
midX = linspace(min(mid_coors(:,1)) - 1, max(mid_coors(:,1)) + 1, points);
midY = polyval(p, midX);

% format return
mid_return = [midY; midX]';

% calc vertex position
intersect = (j_max - p(2))/p(1);
apex = [intersect j_max];

% calc cap_radius position
cap = (top_cut_off - p(2))/p(1);
cap_apex = [cap top_cut_off];
cap_radius = b_matrix(end,2) - cap;
cap_radius2 = (b_matrix(end,2) - b_matrix(1,2))/2;

% calc theta
theta = (atan(1/(-1*p(1))));

% output results to screen
fprintf('   Results from Symmetric Check\n');
fprintf('   Standard Deviation = %.4f\n', s);
fprintf('   Number of mid-points before sort = %.0f\n', a1);
fprintf('   Number of mid-points after sort = %.0f\n', a2);
fprintf('   Offset Angle = %5.4f degrees\n', (theta*(180/pi)));
fprintf('   Slope = %5.4f\n', (-1*p(1)));
fprintf('   Intercept = %5.4f\n', p(2));
fprintf('   Apex at position [x z] = %5.4f  %5.4f\n', apex(1), apex(2));
fprintf('   Cap Center at position [x z] = %5.4f  %5.4f\n', cap_apex(1), cap_apex(2));
fprintf('   Cap Radius = %5.4f pixels (uses line equ)\n', cap_radius);
fprintf('   Cap Radius = %5.4f pixels\n', cap_radius2);
fprintf('   \nPress any Key to Accept Results and Continue or CTRL-C to exit.....\n');
return;

% optim_solver.m
% includes functions to optimize c and b parameters for curve fitting
% first method is that of Nelder-Mead
% second method is Levenberg-Marquardt solver
% can change TolX and TolFun as required

function [c2, b2] = optim_solver(x_new, z_new, parameters2)

% define parameters
solver_type = parameters2(1);
c_start = parameters2(2);
b_start = parameters2(3);
maxiter = parameters2(4);
ratio = parameters2(5);
height = parameters2(6);
del_row = parameters2(7);
g = parameters2(8);

switch (solver_type)
    case (0)
c2 = c_start;
b2 = b_start;
return;
    case (1)
        % % minimize objective function to find solution
        % % optimization using fminsearch
        fprintf('Press Enter to Continue with Fitting (Nelder-Mead)');
        fprintf('Initial c = %5.4f
', c_start);
        fprintf('Initial b = %5.4f
', b_start);
        in_flag = input(' ');

        results = zeros(maxiter,5);
        options = optimset('Display','iter','TolX',1e-4,'TolFun',1e-4);
        tic; % start timer
        for iter = 1:maxiter

            i_guess = [c_start b_start];
            [c_b, fval] = fminsearch(@(V) obj_fun(V, x_new, z_new, ratio, height), i_guess, options);
            fprintf('
Final c = %5.10f
', c_b(1));
            fprintf('Final b = %5.10f
', c_b(2));
            fprintf('Error Function Value = %5.10f
', fval);

            results(iter,1) = iter; % iteration
            results(iter,2) = c_b(1); % c value
            results(iter,3) = c_b(2); % b value
            results(iter,4) = fval; % error

        % calculate surface tension
        parameters3(1) = del_row;
        parameters3(2) = g;
        results(iter,5) = sur_ten(c_b(1), parameters3);

        % reload
        c_start = c_b(1);
b_start = c_b(2);
end
toc;  % stop timer

case(2)
% Levenberg-Marquardt solver
fprintf('
Continue with Fitting (Levenberg-Marquardt)
');
fprintf('Initial c = %5.4f
', c_start);
fprintf('Initial b = %5.4f
', b_start);
in_flag = input(' ');

results = zeros(maxiter,5);
options1 = optimset('Algorithm','levenberg-
marquardt','ScaleProblem','Jacobian','Display','iter','TolX',1e-
4,'TolFun',1e-4);
tic;  % start timer
for iter = 1:maxiter

    i_guess = [c_start b_start];
    [c_b, resnorm] = lsqnonlin(@(V) obj_funb(V, x_new, z_new,
ratio, height), i_guess, [], [], options1);

    fprintf('
Final c = %5.10f
', c_b(1));
    fprintf('Final b = %5.10f
', c_b(2));
    fprintf('Error Function Value = %5.10f
', resnorm);

    results(iter,1) = iter;
    results(iter,2) = c_b(1);
    results(iter,3) = c_b(2);
    results(iter,4) = resnorm;

% calculate surface tension
parameters3(1) = del_row;
parameters3(2) = g;
results(iter,5) = sur_ten(c_b(1), parameters3);

% reload
    c_start = c_b(1);
    b_start = c_b(2);

end
toc;  % stop timer
end

% display results
fprintf('
results
');
fprintf('
 Iteration   c (mm^-2)  b (mm)  Error    Surface Tension
(\text{mN/m})
');
disp(results);

% show plot of both curves
c2 = c_b(1);
b2 = c_b(2);

% generate plot of final results
[x_new_final, z_new_final] = dim2dimless(x_new, z_new, ratio, b2);
[x_theor_final, z_theor_final] = lap_run(c2, b2, height, ratio);

figure(9);
plot(x_theor_final, z_theor_final,'r.','MarkerSize', 0.1);
xlabel('x Dimensionless');
ylabel('z Dimensionless');
title('Final Results');
axis equal;
hold on
plot(x_new_final, z_new_final,'k.','MarkerSize', 0.1);
legend('Theoretical','Experimental');
hold off

% calc volume
vol = vol_calc(x_theor_final, z_theor_final, b2);
fprintf('
Volume = %5.4f mm3
', vol);

return;

% obj_funb.m
% original objective function
% used for Levenberg-Marquardt method

function obj = obj_funb(V, x_new, z_new, ratio, height)

c = V(1);
b = V(2);

% convert to dimensionless
[x_new_dl, z_new_dl] = dim2dimless(x_new, z_new, ratio, b);

exp_data(:,1) = z_new_dl;
exp_data(:,2) = x_new_dl;

% call function lap_run.m
[x_theor, z_theor] = lap_run(c, b, height, ratio);

theor_data(:,1) = z_theor;
theor_data(:,2) = x_theor;

% calculate shortest distance
results = zeros(length(exp_data(:,1)),4);

for j = 1:length(exp_data(:,1))
term1 = exp_data(j,1) - theor_data(:,1); % z
term2 = exp_data(j,2) - theor_data(:,2); % x
dist = sqrt((term1.^2)+(term2.^2));

[dum, idx] = min(dist); % min distance index value

results(j,1) = exp_data(j,1);
results(j,2) = exp_data(j,2);
results(j,3) = theor_data(idx,1);
results(j,4) = theor_data(idx,2);
distzx(:,1) = (results(:,1) - results(:,3));
distzx(:,2) = (results(:,2) - results(:,4));

obj = [distzx(:,1); distzx(:,2)];
return;

% obj_fun.m

% original objective function
% used for Nelder-Mead method

function [obj, zx_residual] = obj_fun(V, x_new, z_new, ratio, height)

c = V(1);
b = V(2);

% convert to dimensionless [x z]
[exp_data(:,2), exp_data(:,1)] = dim2dimless(x_new, z_new, ratio, b);

% call function lap_run.m
[x_theor, z_theor] = lap_run(c, b, height, ratio);

theor_data(:,1) = z_theor;
theor_data(:,2) = x_theor;

mindist = zeros(length(exp_data(:,1)),1);
zx_residual = zeros(length(exp_data(:,1)),2);

for j = 1:length(exp_data(:,1))
term1 = exp_data(j,1) - theor_data(:,1); % z
term2 = exp_data(j,2) - theor_data(:,2); % x
dist = sqrt((term1.^2)+(term2.^2));
[mindist(j,1), idx] = min(dist); % min distance index value

terma = exp_data(j,1) - theor_data(idx,1);
termb = exp_data(j,2) - theor_data(idx,2);

if terma > 0 && termb < 0
    zx_residual(j,1) = mindist(j,1)*-1;
else
    zx_residual(j,1) = mindist(j,1);
end
zx_residual(j,2) = j;

end

obj = sum((mindist(:,1)).^2);
return;
\texttt{% lap_run.m}

\texttt{function [x2 z2] = lap_run(c, b, height, ratio)}

\texttt{para(1) = c;}
\texttt{para(2) = b;}

\texttt{% conditions for the ODE solver as [x z phi]}
\texttt{s_span = [0:0.001:8];}
\texttt{y_initial = [1e-20, 0, 0];}

\texttt{% call ODE solver as [x z phi]}
\texttt{[S, Y] = ode45(@(lap_equ, s_span, y_initial, [], para);}

\texttt{% sort data}
\texttt{i = 1;}
\texttt{while Y(i, 2) < Y(i+1, 2) && Y(i, 1) < Y(i+1, 1);}
\texttt{i = i + 1;}
\texttt{end}

\texttt{\% Sort Theor. Data}
\texttt{x(:,1) = Y(1:i,1);}
\texttt{z(:,1) = Y(1:i,2);}
\texttt{s(:,1) = S(1:i,1);}
\texttt{Phi(:,1) = Y(1:i,3)*(180/pi);}

\texttt{% match height of experimetal curve}
\texttt{height_dl = height*ratio/b; \% convert to dimensionless height}
\texttt{z_index = z(:,1) <= height_dl;}
\texttt{x2(:,1) = x(z_index,1);}
\texttt{z2(:,1) = z(z_index,1);}

\texttt{return;}

\texttt{% lap_equ.m}
\texttt{\% function file for pendant drop analysis (lap_run.m)}

\texttt{function ydot = lap_equ(s, f, para)}
\texttt{c = para(1);}
\texttt{b = para(2);}

\texttt{x = f(1);}
\texttt{z = f(2);}
\texttt{phi = f(3);}

\texttt{ydot(1) = (cos(phi));}
\texttt{ydot(2) = (sin(phi));}
\texttt{ydot(3) = 2 + c*z*(b^2) - ((sin(phi))/x);}
ydot = ydot';
return;

% image_crop.m

function [d, top_cut_off] = image_crop(BW3, img)

input1 = 0;
while input1 ~= 1
    if input1 == 2
        % clear all data
        clear top_cut_off d;
        clear figure 5;
    end

    % construct menu
    fprintf('
Enter Z-axes Height Cut-off\n');
    fprintf('Press 1 for Known Input in Pixels\n');
    fprintf('Press 2 for Graphical Analysis\n');
    input2 = input(' ');

    if input2 == 1
        top_cut_off = input('
Enter Pixel value for drop height: '); % removes values above top_cut_off
    else
        % insert graphical analysis
        imtool(img);
        top_cut_off = input('
Enter Pixel value for drop height: '); % removes values above top_cut_off
        imtool close all;
    end

d = BW3(BW3(:,1)>=top_cut_off,:); % removes values above top_cut_off

    % verify results
    figure(5);
    imshow(img);
    hold on;
    plot(d(:,2),d(:,1),'r.','MarkerSize', 0.1);
    hold off;

    % verify correct info from user
    fprintf('
Verify Image Cut off is Correct\n');
    fprintf('Press 1 to Continue\n');
    fprintf('Press 2 to Try Again\n');
    fprintf('Press CTRL-C to Exit\n');
    input1 = input(' ');

end
imtool close all;

return;
%image_construct.m

function BW3 = image_construct(img, thresh, sigma, line_check, ed)

%load the image
rgb_img = imread(img);
figure(1);
imshow(rgb_img);
title('Original Image');

% Change image to grayscale then binary using threshold
level = graythresh(rgb_img);
I2 = im2bw(rgb_img);
figure(2);
imshow(I2); % can also look at imtool
title('Binary Image');

% detect all edges using MATLAB edge detector
switch (ed)
    case (1)
        BW = edge(I2, 'canny');
    case (2)
        BW = edge(I2, 'sobel');
    case (3)
        BW = bwboundaries(I2, 'noholes');
end

figure(3);
imshow(BW);
title('Image After Edge Detection');

% find the start of target profile edge for bwtraceboundary script
s=size(BW);
for row = 2:line_check % be careful with this, first row must only
    for col=1:s(2) % contain the start of the curve
        if BW(row, col) == 1
            %disp([BW(row, col) row col]);
            e_switch = 1;
            break;
        end
    end
    if e_switch == 1
        break;
    end
end
%disp([row, col]);

input1 = 0;
while input1 ~= 1
    if input1 == 2
        % clear all data and open BW display with input
clear BW3 BW2;
close figure 4;
% need to obtain [row col] for bwtrace
fprintf('\n Use imtool to Select Start of Trace Function\n');
imtool(BW);
col = input('Input col of Starting Point: '); row = input('Input row of Starting Point: ');
imtool close all;
end

% trace profile
BW2 = bwtraceboundary(BW, [row, col], 'S'); disp(BW2);

% remove duplicate rows
BW3 = unique_extract(BW2);
%BW3 = unique_stable(BW2);

% plot results for review
figure(4)
plot(BW3(:,2),BW3(:,1), 'k.', 'MarkerSize', 0.1);
axis ij;
axis equal;
title('Image After Boundary Trace');

% verify correct info from user
fprintf('\n Verify Boundary Trace is Correct\n');
fprintf('Press 1 to Continue\n');
fprintf('Press 2 for Manual Control\n');
fprintf('Press CTRL-C to Exit\n');
input1 = input( ' ');
end

return;

% err_sur_plot.m
% calculates and plots the error surface

function err_sur_plot(c2, b2, x_new, z_new, err_surf_para)

% define err_surf_para
err_sur = err_surf_para(1);
deltab = err_surf_para(2);
deltac = err_surf_para(3);
nop = err_surf_para(4);
ratio = err_surf_para(5);
height = err_surf_para(6);

if err_sur == 1
    fprintf(' Constructing Error Surface....\n');
    count = 0;
    countout = 0;
    xzerr = zeros(nop^2,3);

    for b_span = linspace(b2-deltab, b2+deltab, nop);
        for c_span = linspace(c2-deltac, c2+deltac, nop);
            % compute error surface
            %...
\[
\begin{align*}
V(1) &= \text{c_span}; \\
V(2) &= \text{b_span}; \\
\end{align*}
\]

% uses Nelder-Mead obj function
[obj_err, dum] = obj_fun(V, x_new, z_new, ratio, height);

% store data
count = count + 1;
xzerr(count,1) = V(2);
xzerr(count,2) = V(1);
xzerr(count,3) = \log10(obj_err);
end

countout = countout + 1;
fprintf(' %i', countout);
end

% reshape data for plotting
x = reshape(xzerr(:,1),nop,nop)';
y = reshape(xzerr(:,2),nop,nop)';
z = reshape(xzerr(:,3),nop,nop)';

fprintf('   Surface Complete....\n');

%plot xzerr surface
figure(10);
surfc(x, y, z);
xlabel('b');
ylabel('c');
zlabel('Log Error');
title('Error Surface Plot');
end

return;

% edge_reposition.m

function [x_new, z_new] = edge_reposs(d, apex)

% reposition drop where apex = [0 0] and extract half for fitting
% extract half the curve for comparison to theory

% idx = (d(:,2)>round(apex(1))); % obtains logical index values
%d_half = d(idx,:);
%
% reposition at apex = [0 0]
% x_new = d_half(:,2) - apex(1); % use for evaluatuion
% z_new = (d_half(:,1) - apex(2)).*-1; % use for evaluation

%try this
zx_zeroed(:,2) = d(:,2) - apex(1); % reposition x
zx_zeroed(:,1) = (d(:,1) - apex(2)).*1; % reposition z

idx_rt = (zx_zeroed(:,2)>=0);
idx_lt = (zx_zeroed(:,2)<=0);
zx_new_rt = zx_zeroed(idx_rt,:);
zx_new_lt = zx_zeroed(idx_lt,:);

x_new = zx_new_rt(:,2);
z_new = zx_new_rt(:,1);

return;

% dim2dimless.m
% converts image pixels into dimensionless units

function [x_new_dl, z_new_dl] = dim2dimless(x_new, z_new, ratio, b)

% convert to dimensionless
z_new_dl = z_new.*ratio./b;
x_new_dl = x_new.*ratio./b;

return;

% colonal.m
% this function serves to call function for pendent drop shade analysis
% also does some printing and screen output functions
% input from pd_run.m

function colonal(img, c_start, b_start, parameters, thresh, err_surf_para)

ed_type = parameters(1);
sigma = parameters(2);
line_check = parameters(3);
solver_type = parameters(4);
maxiter = parameters(5);
del_row = parameters(6);
g = parameters(7);
cap_dia_points = parameters(8);
cap_dia_units = parameters(9);
residual_plot = parameters(10);
rotation = parameters(11);

% image filtering and edge detection analysis
% produce fig 1 - 4
fprintf('Beginning Edge Detection...\n');
BW3 = image_construct(img, thresh, sigma, line_check, ed_type);
fprintf('Edge Detection Complete...\n');

% calculate capillary diameter
fprintf('Calculating Capillary Diameter...\n');
ratio = cap_dia_calc(BW3, cap_dia_points, cap_dia_units, img);
fprintf('   Calculation Complete....\n');

% image crop selection of drop height (top_cut_off)
% produce fig 5
fprintf('\n   Beginning Edge Crop....\n');
[d, top_cut_off] = image_crop(BW3, img);
fprintf('   Edge Crop Complete....\n');

% find vertical symmetric axis call function pd_sym
fprintf('\n   Beginning Symmetric Check....\n');
points = 200;  % number of analysis points to determine sym
[mid_zx, apex, height, cap_radius, theta, mid] = pd_sym(d, top_cut_off, points, sigma);
fprintf('   Symmetric Check Complete....\n');

fprintf('\n   Performing Edge Extraction and Axes Centering....\n');
[x_new, z_new] = edge_reposs(d, apex);
%
% rotate experimental data (pixels)
R = [cos(theta) -sin(theta); sin(theta) cos(theta)];
xz_new_rot = (R*[x_new'; z_new']);

% calc new height from rotation (pixels)
height = max(xz_new_rot(:,2));

% convert original experimental data (pixels) to dimensionless form for plotting
[x_new_dl, z_new_dl] = dim2dimless(x_new, z_new, ratio, b_start);
cap_radius_dl = cap_radius*ratio/b_start;
fprintf('   Repositioning Complete....\n');

% convert rotated experimental data (pixels) to dimensionless form for plotting
[x_new_dl_rot, z_new_dl_rot] = dim2dimless(xz_new_rot(:,1),
xz_new_rot(:,2), ratio, b_start);

figure(6)
imshow(img);
hold on
plot(d(:,2),d(:,1),'r.','MarkerSize', 0.1);
plot(mid_zx(:,2),mid_zx(:,1),'b');
plot(mid, d(1:length(mid)),'- r.','MarkerFaceColor','g','MarkerSize',0.2);
hold off
figure(7)
figure(7)
plot(x_new_dl, z_new_dl,'k.','MarkerSize', 0.1);
xlabel('x Dimensionless');
ylabel('z Dimensionless');
axis equal;
hold on
plot(x_new_dl_rot, z_new_dl_rot,'g.','MarkerSize', 0.1);
title('Drop Profile Analysis Image Used for Fitting');
legend('Original','Rotated');
hold off

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Theor. Profile %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% apex coordinates sent from image apex coordinates sent as [x z]
[x_theor, z_theor] = lap_run(c_start, b_start, height, ratio);

figure(8)
plot(x_theor, z_theor, 'r.', 'MarkerSize', 0.1);
xlabel('x Dimensionless');
ylabel('z Dimensionless');
axis equal;
hold on
plot(x_new_dl, z_new_dl, 'k.', 'MarkerSize', 0.1);
plot(x_new_dl_rot, z_new_dl_rot, 'g.', 'MarkerSize', 0.1);
hold off
title('Initial Profiles');
legend('Theor','Experimental','Rotated');

% optimization methods
% define parameters
parameters2(1) = solver_type;
parameters2(2) = c_start;
parameters2(3) = b_start;
parameters2(4) = maxiter;
parameters2(5) = ratio;
parameters2(6) = height;
parameters2(7) = del_row;
parameters2(8) = g;

% Use rotated Image?
if rotation == 1
    fprintf('n            Using Rotated Experimental Data\n');
    x_new = xz_new_rot(:,1);
    z_new = xz_new_rot(:,2);
end

% calculate error surface plot
% plots fig 10
% call function for error plotting
% err_sur_plot(c2, b2, xz_new_rot(:,1), xz_new_rot(:,2),
% err_surf_para);
% % residual plot
% residual(c2, b2, x_new, z_new, ratio, height, residual_plot)

[c2, b2] = optim_solver(x_new, z_new, parameters2);

% calculate error surface plot
% plots fig 10
% define err_surf_para
err_surf_para(5) = ratio;
err_surf_para(6) = height;

% call function for error plotting
err_sur_plot(c2, b2, x_new, z_new, err_surf_para);
% residual plot
residual(c2, b2, x_new, z_new, ratio, height, residual_plot)

return;

% cap_dia_calc
% calculates the capillary diameter for a selected number of rows then determines the average
function ratio = cap_dia_calc(BW3, cap_dia_points, cap_dia_units, img)

% cap_dia = zeros(cap_dia_points,1);
% for i = 1:cap_dia_points
%     idxBW3 = ismember(BW3(:,1),BW3(i,1));
%     cap_dia(i,1) = diff(BW3(idxBW3,2));
% end
%
% % calculate average of all points
% cap_dia_avg = mean(cap_dia);

%%%%%%%%%%%%%%%%%%%%
% try this way
right = BW3(1:cap_dia_points,:);
left = BW3(end-cap_dia_points+1:end,:);
sorted_data = intersect(right(:,1), left(:,1));
cap_dia = zeros(2,length(sorted_data(:,1)));
for i = 1:length(sorted_data)
    idxright = ismember(right(:,1),sorted_data(i));
    avgright = mean(right(idxright,2));
    cap_dia(1,i) = avgright;

    idxleft = ismember(left(:,1),sorted_data(i));
    avgleft = mean(left(idxleft,2));
    cap_dia(2,i) = avgleft;
end
% calculate average of all points
cap_dia_avg = mean(diff(capDia));

% calculate ratio
ratio = cap_dia_units/cap_dia_avg;

% output results to screen
fprintf('    Results from Capillary Diameter Calculation
');
fprintf('Number of Points Selected = %5.0f points
', cap_dia_points);
fprintf('Average Capillary Diameter = %5.2f pixels
', cap_dia_avg);
fprintf('Calculated Ratio = %5.5f mm/pixel
', ratio);
disp(' ');
fprintf(' Use the Calculated Capillary Diameter?\n');
input2 = 0;
input2 = input('Press 1 for Yes or Press 2 for No: ');

if input2 == 2
    imtool(img);
cap_dia_avg = input('Enter Average Capillary Diameter in Pixels: ');
    imtool close all;
clear ratio;
ratio = cap_dia_units/cap_dia_avg;
fprintf('Calculated Ratio = %5.5f mm/pixel',ratio);
end

return;