4-2016

Secondary atomization of inelastic non-newtonian liquid drops in the bag and multimode regimes

Jonathan Rocha
Purdue University

PURDUE UNIVERSITY
GRADUATE SCHOOL
Thesis/Dissertation Acceptance

This is to certify that the thesis/dissertation prepared

By Jonathan Rocha

Entitled

SECONDARY ATOMIZATION OF INELASTIC NON-NEWTONIAN LIQUID DROPS IN THE BAG AND MULTIMODE REGIMES

For the degree of Master of Science in Mechanical Engineering

Is approved by the final examining committee:

Paul E. Sojka
Chair

Jun Chen

Daniel R. Guildenbecher

To the best of my knowledge and as understood by the student in the Thesis/Dissertation Agreement, Publication Delay, and Certification Disclaimer (Graduate School Form 32), this thesis/dissertation adheres to the provisions of Purdue University's "Policy of Integrity in Research" and the use of copyright material.

Approved by Major Professor(s): Paul E. Sojka

Approved by: Anil K. Bajaj 4/28/2016

Head of the Departmental Graduate Program Date

SECONDARY ATOMIZATION OF INELASTIC NON-NEWTONIAN LIQUID DROPS IN THE BAG AND MULTIMODE REGIMES

A Thesis

Submitted to the Faculty

of

Purdue University

by

Jonathan Rocha

In Partial Fulfillment of the

Requirements for the Degree

of

Master of Science in Mechanical Engineering

May 2016

Purdue University

West Lafayette, Indiana

To my family, friends, and all who supported me up to this time.

ACKNOWLEDGEMENTS

To begin, I thank God for providing the opportunities and paths I have taken. Next, I give thanks to my mother, Alicia Santana, who sacrificed so much to raise two children alone. Nothing I could give would repay her efforts. Also, I thank my older sister, Juliette Rocha, for being there and her guidance during my youth. I am so proud of her for finishing her A.A. degree with a 4.0 while caring for four children! What's more, thanks go to my nephews and niece (Christian, Emilio, Ruben, and Bryanna) for being a part of my life and showing me the joys of watching them grow into the beautiful people that they are.

I would like to thank the TRIO program, especially the director, Mr. Carlton L. Daley, and academic advisor, Ms. Eon Alleyne-McMayo. They supported and believed in me during my A.A. degree and led me toward a Presidential Scholarship for my B.S. degree. Also, I would like to thank Dr. Aglan at Tuskegee University who provided me with one and a half years of undergraduate research experience. Further thanks go to Tuskegee University who offered me such an outstanding scholarship and academic program where otherwise I would not had been able to afford college.

Next, I would like to thank Dr. Sojka first for accepting me as his research student at Purdue University and also all his guidance and patience during the process of research. He assisted me in obtaining the NSF Graduate Research Fellowship and his experience in sprays and atomization has ensured that all my methods and results are accurate and meaningful. He is a mentor that I will acknowledge when I continue in my career. Additionally, thanks go to Dr. Campanella for allowing me to use his rheology lab.

Lastly, I would like to thank my girlfriend, Chuyang Xiao, who has supported and been there with me through my master's degree. Her sacrifices and patience during my studies is much appreciated. She has been my closest friend to enjoy life with and my shoulder to lean on when times were tough.

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

Figure Page

NOMENCLATURE

Symbols	Description
Roman	
a	Constant for surface tension correction factor
a_L	Drop acceleration
a_p	Particle acceleration
A_{probe}	Area covered by probe
b	Constant for surface tension correction factor
C	Dimensionless polymer concentration by mass
c	Constant for surface tension correction factor
C_d	Coefficient of drag
$C_{d,ini}$	Coefficient of drag at initiation time
$C_{d,sphere}$	Coefficient of drag for a sphere
d_{cro}	Drop cross-stream diameter
$d_{cro,ini}$	Drop cross-stream diameter at initiation time
$d^*_{cro,ini}$	Drop dimensionless cross-stream diameter at initiation time
d_{min}	Minimum drop cross-stream length
$d_{min,ini}$	Stream-wise length at initiation time
d	Damping constant in TAB model, or drop diameter
d_o	Initial drop diameter
d_p	Particle diameter
d_{rim}	Drop rim diameter
F	Aerodynamic forces acting on the drop
F_{correc}	Correction factor
g	Acceleration due to gravity
J_{air}	Air momentum
J_{par}	Momentum required for particles to reach gas velocity
k	Spring constant in TAB model

Symbols	Description
<u>Roman</u>	
K	Flow consistency index
L_{bag}	Bag length
L_{cir}	Circumferential length of Du Nouy ring
L	Measured length
L_{Avg}	Average measured length
m	Drop mass
m_{cmc}	Mass of CMC
m_{Total}	Total mass of solution
m_{Water}	Mass of water
Ma	Mach number
n	Flow behavior index
$\dot{n}$	Number of particles per unit time
N	Number of drops, particles, or measurements
N_{RT}	Rayleigh-Taylor Wave number
Oh	*Oh*nesorge number
P_{Avg}	Average tensiometer reading
P_{calc}	Ideal tensiometer reading
P_{read}	Reading from tensiometer
q	Iterated value for velocity in TAB model
r	Du Nouy wire radius
r_{cro}	Drop cross-stream radius
r_0	Initial drop radius
$r_{shortest}$	Shortest distance from centroid to drop boundary
R	Drop or Du Nouy ring radius
Re	Reynolds number
Re_L	Reynolds number using drop viscosity
s	Standard deviation
t	Time
t_{ini}	Initiation time
t_{exp}	Characteristic experimental time
t_p	Particle relaxation time
$t_{\alpha/2}$	Confidence interval coefficient
T	Dimensionless inviscid time

Symbols	Description
Roman	
T_{bag}	Dimensionless time at bag break up
T_{ini}	Dimensionless initiation time
T_{rim}	Dimensionless time at rim break up
T_{Stamen}	Dimensionless time at stamen breakup
T_{total}	Dimensionless total breakup time
u_{lag}	Stokes lag velocity
u_G	Gas phase velocity
u_d	Drop velocity
u_{rel}	Relative velocity between continuous and dispersed phase
u_y	Vertical or cross-stream velocity
U_{ini}	Drop velocity at initiation time
U_{Bag}	Drop velocity at bag breakup
We	Weber number
x	Displacement of drop equator
$\dot{x}$	Velocity of drop equator
$\ddot{x}$	Acceleration of drop equator
x_{ini}	Drop displacement up to initiation time
x_{Bag}	Drop displacement up to bag breakup
y	Dimensionless displacement of drop equator

Symbols	Description
Greek	
$\dot{\gamma}$	Strain rate
$\dot{\gamma}_{Avg}$	Average strain rate
δ	Uncertainty
δ_b	Boundary layer thickness
$\dot{\varepsilon}_{Avg}$	Average strain rate
λ	Constant for Carreau model
λ_c	Critical wave length
λ_{max}	Wave length corresponding to maximum growth rate
μ_0	Viscosity at zero shear
μ_{∞}	Viscosity at infinite shear

Symbols	Description
<u>Greek</u>	
μ_{eff}	Effective viscosity
μ_G	Gas viscosity
μ_L	Liquid viscosity
ρ_G	Gas density
$\rho_{instrument}$	Density measurement
ρ_L	Liquid density
ρ_{mean}	Mean density
ρ_p	Particle density
σ	Surface tension
τ	Shear stress
φ	Relative uncertainty

ABSTRACT

Rocha, Jonathan, M.S.M.E. Purdue University, May 2016. Secondary Atomization of Inelastic Non-Newtonian Liquid Drops in the Bag and Multimode Regimes. Major Professor: Dr. Paul E. Sojka.

Secondary atomization of inelastic shear thinning non-Newtonian liquids in the bag and multimode regimes was studied. Six mixtures were formulated from deionized (DI) water, *Avantor Performance Materials'* USP grade 100% vegetable based glycerin, and Ashland's Carboxymethylcellulose (CMC-7MF or CMC-7HF). The resulting solutions had power law parameters flow behavior index, *n*, between 0.71 and 0.93 and consistency index, *K*, in the range of 0.0464 to 0.37 $Pa \cdot s^n$. The effective viscosity for each mixture was estimated using the power-law model and experimentally measured strain rates up to the initiation time

Secondary atomization was achieved using a continuous jet setup. Breakup events were captured using a Vision Research Phantom v7.3 high speed camera operated at >6600 fps. This typically yielded more than 100 frames for each breakup event. Post processing was performed using an in-house MATLAB code. Breakup was observed to occur in the bag and multimode regimes.

The measurement approach was validated by comparing DI-water values with literature results. The flow conditions and liquid properties varied between: $10 < We < 50$, $0.0021 < Oh < 0.41$, $0.71 \leq n \leq 0.93$, $0.0464 \leq K \leq 0.37\ Pa \cdot s^n$, $990 < \rho_L < 1210\ \frac{kg}{m^3}$, and $0.065 < \sigma < 0.073\ \frac{N}{m}$.

Data obtained using the MATLAB code includes: initiation time, cross-stream diameter, drop displacement, velocity and acceleration, plus the coefficient of drag at initiation time. The bag breakup time was measured, along with the corresponding rim

diameter, bag length, displacement, and velocity. Many of these quantities exhibited peaks in the bag or bag-and-stamen regimes, with magnitudes that varied with liquid properties.

Results from the videos show shear thinning, inelastic drop breakup modes share many morphological features with those for Newtonian liquid drops. The only minor differences are persistent ligaments throughout every stage of breakup and non-uniform bag growth in the bag breakup regime.

The similarity between current and Newtonian drop results means the classical *We* versus *Oh* regime map remains valid for shear thinning, inelastic drops. Two other correlations were found to adequately act as alternate regime maps. These include *We* versus liquid *Re* and *We* versus Rayleigh-Taylor wavenumber.

Finally, the Taylor Analogy Breakup (TAB) model was applied to the current results to determine its accuracy when predicting shear thinning, inelastic drop deformation. The aerodynamic force term was altered to account for the increasing drop projected area and drop viscosity modeled using the approximated strain rate up to initiation time. Data from post processing was used in order to further improve the TAB model The result was quantitative agreement between predictions and experiments to within 29% for initiation time and 36% for drop velocity at initiation time.

CHAPTER 1. INTRODUCTION

When large enough relative velocities exist between a liquid drop and the surrounding gas phase, the drop will deform and fragment. This process is known as secondary atomization. It should be treated as a rate process rather than a jump process due to drop movement between 40 and 100 diameters while breakup is ongoing (Dai and Faeth 2001).

Typical applications where secondary atomization is important include pharmaceutical coating sprays, diesel fuel injection, and air assisted fuel atomization in gas turbine engines (Flock *et al*., 2012). The study of secondary atomization dates to before 1904 (Hinze, 1955).

Figure 1.1 shows several modes by which drops disintegrate. They are often termed oscillatory deformation, bag breakup, multimode breakup, sheet thinning breakup, and catastrophic breakup (Hsiang and Faeth, 1995). Each mode has its own deformation stage prior to breakup. For example, a spherical drop undergoing bag breakup evolves into an ellipsoid (Dai and Faeth, 2001).

There are several experimental methods used to produce drop breakup. The most popular are the shock tube, continuous jet and drop tower (Guildenbecher *et al*., 2009).

For Newtonian liquids, the mode of breakup for a given relative velocity can be predicted using two non-dimensional parameters (Hsiang and Faeth, 1995). These are the Weber number (*We*),

$$We = \frac{\rho_G u_{rel}^2 d_0}{\sigma} \tag{1.1}$$

which is the ratio of aerodynamic to restorative forces, and the *Oh* number,

$$Oh = \frac{\mu_L}{\sqrt{\rho_L \sigma d_0}} \tag{1.2}$$

which includes the effect of liquid viscosity. In Equation (1.1-1.2), ρ_G and ρ_L are the gas and liquid densities, u_{rel} is the initial relative velocity between the gas and the drop, d_o is the initial drop diameter, σ is the surface tension, and μ_L is the liquid viscosity.

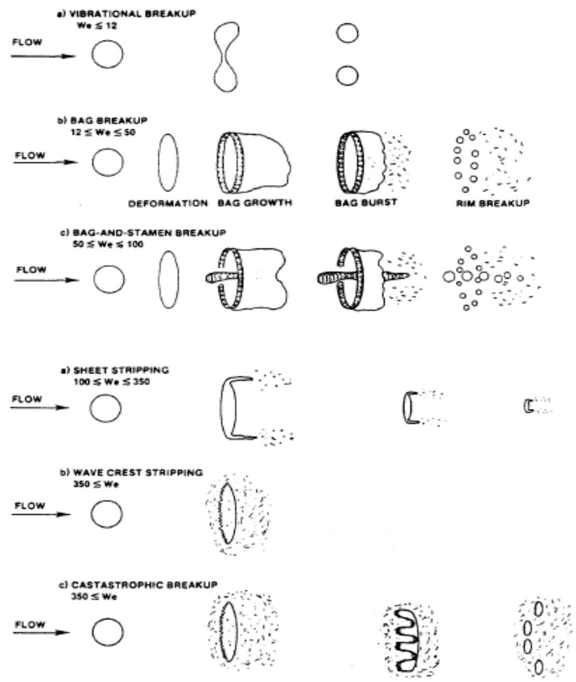


Figure 1.1. Drop breakup regimes (Pilch and Erdman, 1987).

Additional research challenges arose when non-Newtonian liquid breakup was considered (Wilcox *et al.*, 1961). Non-Newtonian liquids have a viscosity that changes with strain rate, possibly depending on the history of the applied strain, and may also have elasticity.

Experimentally, the shear thinning, or pseudo plastic, liquids (Lopez, 2010) and time dependent, or thixotropic, liquids (Snyder, 2015) have been studied successfully. A few applications for non-Newtonian liquids include aerospace propellants, bio-fuels, firefighting liquids, thermal barrier coatings, water-gel explosives, paints, and more (Gao *et al.*, 2014).

Despite their variable viscosity, shear-thinning, inelastic and elastic drop breakup modes are qualitatively similar to those found for Newtonian breakup. The only qualitative differences reported is increased bag growth, ears extruding from the rim, and persistent ligaments (Snyder, 2011).

There is some non-Newtonian drop experimental data, including fragment sizes (Gao *et al*., 2014; Wilcox *et al*., 1961; Ng and Theofanous, 2008), breakup times (Lopez, 2010; Snyder, 2015; Arcoumanis *et al*., 1996), maximum cross-stream dimension (Lopez, 2010; Snyder, 2015); Theofanous, 2011), drop displacement and acceleration (Joseph *et al*., 1999, 2002), plus velocity and drag coefficients (Theofanous, 2011). However, experimental results that include drop rim diameters, and bag lengths have not been reported for these liquids as either temporal histories or as a function of *We*.

To help bridge this gap, secondary breakup for six non-Newtonian liquids are compared to one another, and to a Newtonian liquid. Results include time histories for all the quantities listed. The influence of *We* on these quantities is also presented and discussed.

The remainder of this thesis is organized as follows. The literature for inelastic non-Newtonian drop breakup was reviewed and limitations pointed out. However, a thorough background in Newtonian liquid breakup was established and necessary in order to identify the main differences and similarities between breakup of these two liquid types. Sections are broken up into general non-Newtonian behaviors and then into the bag breakup regime, multimode regime, and sheet thinning regime. Then, the literature review is summarized and provides the most important findings from Newtonian and non-Newtonian secondary atomization works. After that, the experimental methods and corresponding uncertainties are discussed. Subsequently, the results give detailed breakup characteristics followed by experimental measurements at specific points of interest. Finally, the Taylor analogy breakup (TAB) model improvements and result comparisons are presented.

In summary, the purpose of this thesis is to remove gaps in the inelastic non-Newtonian liquid drop secondary atomization literature by focusing on the bag and multimode regimes. Experimental data were acquired, then predictions from a modified version of the TAB model were compared to that data to improve the physical understanding of these important processes.

CHAPTER 2. LITERATURE REVIEW

2.1 Introduction

Drop disintegration requires a relative velocity between the surrounding medium (in this case air) and the drop itself. It is desirable that this velocity be uniform across the drop windward surface for the sake of accurately interpreting breakup behavior, for simplifying computations, and for comparisons between experimental methods. The most popular methods for initiating breakup are the drop tower, shock tube, and continuous jet. The continuous jet was used in this study.

A typical continuous jet setup as shown in Figure 2.1. It uses a converging nozzle to accelerate air while drops fall into it by the action of gravity. Continuous operation allows for higher data rates than shock tubes or drop towers. However, the flow experienced by the drop as it passes through the upper boundary layer is not uniform.

To ensure the boundary layer does not significantly affect drop deformation, it is important to ensure the time it takes for the drop to transit the boundary layer is less than the breakup initiation time (Guildenbecher, 2009),

$$\frac{(d_0+\delta_b)}{u_y}\frac{u_{rel}}{d_0}\sqrt{\frac{\rho_G}{\rho_L}} < T_{ini} \tag{2.1}$$

where δ_b is the boundary layer thickness and the u_y the drop velocity as it enters the boundary layer.

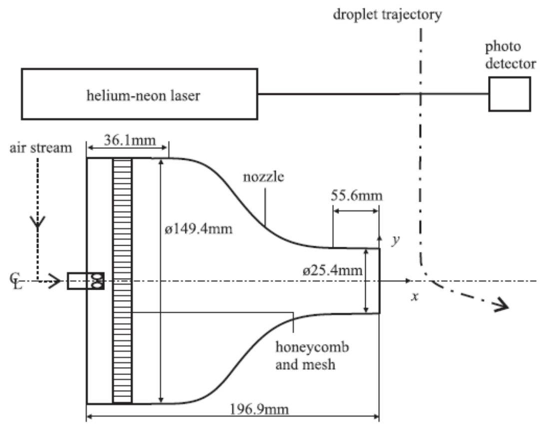


Figure 2.1. Typical continuous jet drop breakup apparatus (Flock *et al*., 2012).

Every system for drop breakup must have a drop generating system. For the low to moderate *Oh* operation considered here a syringe dispenser was used. It is shown in Figure 2.2.

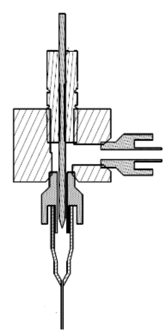


Figure 2.2. Typical syringe type drop dispenser. (Guildenbecher, 2009).

2.2 <u>Non-Newtonian Fluids</u>

Newtonian fluids are those which have a linear relationship between shear stress and strain rate, and whose curve also begins at the origin. All other fluids are considered non-Newtonian (Rao, 2014).

Generally, there are three broad categories of non-Newtonian liquids. Time independent liquids are those whose viscosity is a function of strain rate only and may or may not begin at the origin on a shear stress versus strain rate plot. Time dependent liquids have a shear stress versus strain slope that is dependent on the history of applied shear.

Lastly, viscoelastic fluids have a recovery so continue to deform after shearing has ceased (Holdsworth, 1971). Time independent non-Newtonian liquids were the focus of this study.

Time independent liquids are also termed inelastic. As shown in Figure 2.3 they are subdivided into Newtonian liquids, shear thinning liquids, and shear thickening liquids. Shear thinning liquids are also known as pseudo plastic, have a concave down shear stress versus strain rate curve, and therefore an effective viscosity that decrease with increasing strain rate. Shear thickening liquids are also known as dilatant, have a concave up shear stress versus strain rate curve, and therefore an effective viscosity that increases with an increase in strain rate. Any of these three inelastic liquids can exhibit Bingham plastic behavior—this describes a shear stress versus strain rate curve that does not depart from the origin. This study employed shear thinning liquids exclusively.

Shear thinning behavior may occur due to asymmetric polymeric molecules becoming entangled or randomly oriented. Applied strain causes the long chained entangled molecules or randomly oriented molecules to become less tangled and orient themselves along the strain direction. This process requires a larger force near zero strain and therefore results in a greater effective viscosity. Once the molecules have become untangled and/or oriented at large strain rates the effectively viscosity is reduced and can become constant with higher strain rate.

Shear thinning may also occur when solvated polymer molecules produce layers which interact with neighboring solvated molecules. When strain occurs these layers are sheared away and the effective molecule size drops. This in turn results in smaller intermolecular interactions and a consequent drop in viscosity.

Time independent fluids, except Bingham plastics, may be modeled using the simple power-law, or Ostwald-de Waele, relationship. It can be applied to shear thinning liquids (n<1), shear thickening liquids (n>1), and Newtonian liquids (n=1),

$$\mu_{eff} = K(\dot{\gamma})^{n-1} \tag{2.2}$$

The flow behavior index n determines the extent of non-Newtonian behavior (Metzner and Otto, 1957).

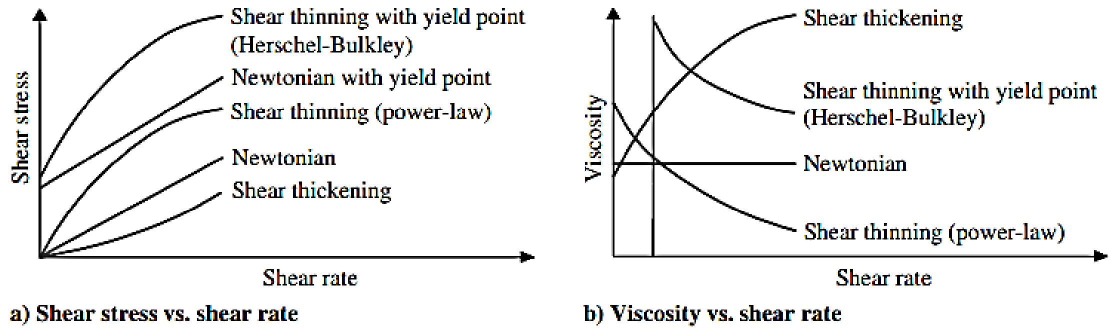


Figure 2.3. Strain rate versus shear stress for various time independent liquids (Arnold *et al*., 2011).

A more complicated viscosity model is that of Carreau. It models the effective viscosity with strain rate for pseudo plastic liquids (Yoon and Ghajar, 1987),

$$\mu_{eff} = \mu_\infty + (\mu_0 - \mu_\infty)[1 + (\lambda\dot{\gamma})^2]^{(n-2)/2} \tag{2.3}$$

here μ_∞ and μ_0 are the zero- and infinite-strain rate viscosities, $\dot{\gamma}$ is the strain rate, λ is a fluid relaxation time and n is some constant.

Regardless of rheological model used, the effective $\dot{\gamma}$ for drops under various aerodynamic loads hasn't been determined. Since it's crucial to calculation of μ_{eff}, a model must be built for its estimation.

One possible method for determining the effective strain rate is to use deformation data (Theofanous *et al*., 2013),

$$\dot{\gamma}_{avg} = \frac{\Delta d}{d_0 \Delta t} = \frac{(d-d_0)}{d_0 \Delta t} = \left(\frac{d}{d_0} - 1\right)\frac{1}{\Delta t} \tag{2.4}$$

where Δd is the change in drop diameter during Δt, d is the drop diameter at time t, and d_0 the drop initial diameter.

Another approach is that of Lopez (2010), who defined an effective strain rate using only initial conditions,

$$\dot{\gamma} = \frac{u_{rel}}{d_0} \tag{2.5}$$

where u_{rel} is the initial relative velocity between the drop and the surrounding gas and d_0 the initial drop diameter.

Regardless of the model for effective strain rate, the spatial variation in deforming drop local $\dot{\gamma}$ causes variations in effective viscosity throughout the drop. This makes determining an effective viscosity difficult.

2.2.1 Weber and Ohnesorge Numbers

The Weber number is defined in Equation (1.1). If *We* is an appropriate criterion for non-Newtonian secondary breakup is up for debate. For instance, Arcoumanis *et al*. (1996) claim that *We* is not appropriate because non-Newtonian surface tension varies with applied strain. This statement is not generally valid because not all inelastic liquids have strain rate dependent surface tension, although it may be the case for solutions containing large concentration of surface active molecules. That is not the case here so Equation (1.1) will be used.

The *Oh*nesorge number (*Oh*) is defined in Equation (1.2), but in the non-Newtonian liquid case an effective viscosity, μ_{eff}, is employed,

$$Oh = \frac{\mu_{eff}}{\sqrt{\rho_L d_0 \sigma}} \tag{2.6}$$

Using Equation (2.2) to approximate the effective viscosity and characterize the inelastic/shear thinning liquids (CMC-Water and CMC-Water-Glycerin), Lopez (2010) found that increasing *K* (flow consistency index) or *n* (flow behavior index) caused *Oh* to increase and required a greater value for *We* to initiate a particular breakup. See Table 2.1. This *Oh* versus *We* is consistent with the conceptual understanding for Newtonian liquids (Hsiang and Faeth, 1995).

Table 2.1. Critical *We* for bag and bag-and-stamen breakup based for a power law liquid (Lopez, 2010).

Liquid	***n***	***K***	***Oh* bag**	**We_c bag**	***Oh* bag-and-stamen**	**We_c bag-and-stamen**
0.8%CMC-7MF	0.83	0.10	0.05	12	0.05	18
0.5% CMC-7HF	0.67	0.36	0.04	12	0.04	19
1.4% CMC-7MF	0.75	0.38	0.09	13	0.09	23
0.06%CMC-7MF /74.96% Glycerin	0.84	0.28	0.14	14	0.14	19
0.06%CMC-7MF /79.95% Glycerin	0.85	0.40	0.24	15	0.23	24
0.05%CMC-7MF /84.96% Glycerin	0.86	0.43	0.27	15	0.26	25

2.2.2 Initiation Time

Non-Newtonian drop breakup initiation time is defined in the same manner as for Newtonian liquids. For the liquids considered here, bag and bag-and-stamen regime initiation times were found to be approximately independent of *We*. In agreement with other findings, the initiation time was found to rise with an increase in *Oh*. In fact, an increase in *K* did increase initiation time, but variations in *n* had no effect (Lopez, 2010). This independence may be due to the Lopez (2010) *We* range of $10 \lesssim We \lesssim 25$.

2.2.3 Maximum cross-stream dimension

The maximum cross-stream dimension is defined as the transverse drop dimension at the initiation time. Lopez (2010) found that maximum cross-stream diameter increased with *We*, while an increase in either *n* or *K* decreased the cross-stream diameter.

2.2.4 Bag breakup results

Lopez (2010) showed that the stages of deformation and breakup for an inelastic/shear thinning liquid resemble those for Newtonian liquids in almost every way. The only differences are significant bag growth and stretching before the bag breaks, along with persistent ligaments. Inelastic/shear thinning liquid deformation was observed to increases with increasing *We*, and more viscous liquids (higher *n* and *K*) deform less at the same *We*.

Since the breakup stages are similar to those found for Newtonian liquids, the initiation time, T_{ini}, is still the time interval from drop initial deformation until the bag begins to form. The bag and total breakup time definitions, T_{bag} and T_{tot}, also remain the same. They are the time when the bag first ruptures and when the original drop and its fragments no longer breakup.

Table 2.2 shows changes in *K* have a larger effect on inelastic/shear thinning liquid Tbag than does *n*. Regardless of whether *K* or *n* is increased, the trend of increasing *Oh* resulting in a longer T_{ini} was also observed by Gel'fand *et al*. (1973) for Newtonian liquids. Furthermore, Lopez (2010) reported that T_{ini} was not affected by increasing *We* in this regime, which is similar to results from Hsiang and Faeth (1992), who found that $T_{ini} \approx 1.6$ when $Oh < 0.1$. Lopez (2010) also noted that increasing *Oh* provided results that were again consistent with Hsiang and Faeth (1992).

Table 2.2. Bag breakup times for inelastic/power law liquids (Lopez, 2010).

Liquid	n	K	T_{ini}	T_{bag}	T_{rim}
0.8%CMC-7MF	0.83	0.1	1.51	3.25	5.08
0.5% CMC-7HF	0.67	0.36	1.73	3.48	5.43
1.4% CMC-7MF	0.75	0.38	1.71	3.66	5.49
0.06%CMC-7MF / 74.96% Glycerin	0.84	0.28	1.71	3.84	6.36
0.06%CMC-7MF / 79.95% Glycerin	0.85	0.4	1.92	4.33	7.63
0.05%CMC-7MF / 84.96% Glycerin	0.86	0.43	1.76	4.13	7.1

For inelastic/shear thinning liquids T_{bag} exhibited more sensitivity to changes in *K* than in *n*. Unlike T_{ini}, T_{bag} did decrease with larger *We*, behavior similar in magnitude to that reported by Dai and Faeth (2001).

Table 2.2 shows that rim breakup time, T_{rim}, for inelastic/shear thinning liquids increases with either increasing *n* or *K*, and decreases with increasing *We* (Lopez, 2010). Since T_{rim} for Newtonian liquids typically coincides with the total breakup time, T_{tot}, they may be compared. Results for Newtonian liquids from Hsiang and Faeth (1992) show that when $Oh<0.1$ T_{tot} is approximately 5 and increases when $Oh>0.1$. However, the highest estimated *Oh* (0.26) considered by Lopez (2010) had $T_{total} \approx 7.6$, which is ~15% larger than predicted by the relation of Gel'fand *et al*. (1973). The difference is likely due to u_{rel} decreasing in time, which causes a reduction in $\dot{\gamma}$ and thus a rise of μ_{eff}. Furthermore, by comparing the dependence of T_{rim} with those for T_{ini} and T_{bag}, the dependence on *n* is seen to increase with time. Consequently, shear thinning behavior becomes more important and initial conditions are of less importance for non-Newtonian drops post breakup initiation.

2.2.5 Bag-and-stamen breakup results

The stages of deformation for inelastic/shear thinning drops undergoing bag-and-stamen breakup are also generally the same as to those for Newtonian liquids—the spherical drop first deforms into an ellipsoid, the bag, rim, and stamen grow, the bag breaks, followed by rim breakup, and finally stamen breakup. The most notable differences are significant bag growth, stretching before bag breakup, and persistent ligaments after breakup (Lopez, 2010). .

As shown in Table 2.3 all inelastic/shear thinning liquids have breakup times that increase with larger *n* or *K*. These results are expected since higher *K* corresponds to a larger viscosity while larger *n* implies less deviation from the zero shear viscosity.

Table 2.3 also shows breakup times decrease with increasing *We*. The total breakup time with *We* relation is contrary to what is found for Newtonian liquids by Hsiang and Faeth (1992) who found the total breakup time is approximately constant with *We* at $T_{total} \approx 5$ which is still of similar magnitude observed for the 0.8%CMC-7MF liquid (*Oh*<0.05). Although, the increase in times with *Oh* is in accordance with his results for Newtonian liquids. However, observations by Gel'fand *et al*. (1974) did support the general decrease in T_{total} with *We* and Krzeczkowski (1980) results were in agreement with both the variation of *We* and *Oh*.

What’s more, comparing the bag breakup times for the inelastic liquids in the bag regime to the bag breakup times in this regime, the average values dropped from an average range of $3.25 < T_{bag} < 4.33$ to a range of $2.64 < T_{bag} < 3.7$, respectively. Yet, the result may not be completely due to an increased strain rate, although likely a contributing factor, but instead due to the stamen reducing the volume of liquid transferred to the bag which is discussed thoroughly by Dai and Faeth (2001). Furthermore, there are two bags present compared to the bag regime which adds further volume reduction for a single bag.

Surprisingly, the average total breakup time did not vary significantly from the bag breakup mode to the bag-and-stamen breakup model (Lopez, 2010).

Table 2.3. Breakup times for various stages within the bag-and-stamen regime for inelastic/shear thinning liquids (Lopez, 2010).

Liquid	n	K	T_{bag1}	T_{bag2}	T_{rim}	T_{stamen}
0.8%CMC-7MF	0.83	0.1	2.64	2.74	3.83	4.78
0.5% CMC-7HF	0.67	0.36	2.8	2.96	3.84	5.12
1.4% CMC-7MF	0.75	0.38	2.95	3.15	4.54	6.28
0.06%CMC-7MF / 74.96% Glycerin	0.84	0.28	3.38	3.48	5.2	6.33
0.06%CMC-7MF / 79.95% Glycerin	0.85	0.4	3.7	3.86	5.93	7.61
0.05%CMC-7MF / 84.96% Glycerin	0.86	0.43	3.5	3.61	5.89	7.41

2.2.6 Sheet Thinning Results

For viscoelastic/ shear thinning liquids, the breakup is similar to Newtonian sheet thinning results. However, the sheets pulled from the periphery are much more persistent so they are pulled much further downstream before being sheared (Theofanous *et al*., 2013). Also, at high polymer concentrations, elastic solutions have been observed to develop ligaments instead of sheets being stripped from the periphery.

The initiation time for this process did not vary significantly with *We*, *Oh*, or polymer concentration and the average initiation times can be seen in (Snyder, 2011).This trend or lack thereof may be the result of strain rate being significantly large such that the effective viscosity has become near to the solvent viscosity. The liquids of this study had a flow behavior index, n, between $0.329 < n < 0.480$ and therefore had a highly shear thinning behavior. As a result, as strain rate is increased (larger We) the viscosity deviates more rapidly from the zero shear viscosity compared to liquids with greater *n*.

2.3 **Summary and Conclusions**

A comprehensive study of Newtonian and Non-Newtonian liquid secondary atomization literature has been completed. This includes the most popular experimental setups and the assumptions which allow results to be comparable to one another. Also, dimensionless groups such as Re, We, *Oh*, Ma, and ε have been sought out for their importance to secondary atomization. Newtonian regime maps have been compared to obtain *We* ranges where a particular mode of breakup can be expected to occur. The most impactful forces acting on a disintegrating drop have been identified and negligible forces noted. C_d for spheres and oblate spheroids as a function of *Re* have been obtained for comparison with the deforming liquid drops.

Furthermore, the effects of contamination have been noted and should always be considered for any natural phenomena investigation. For Newtonian liquids, more literature is available so relationships between breakup modes, maximum cross-stream dimension, fragment sizes, volumes, displacements, velocity, acceleration, and breakup times with *We* and *Oh* have been identified. Additionally, a few Rayleigh-Taylor theory applications have shown to be sufficiently accurate in predicting stamen growth, breakup modes, and regime maps. However, most require some experimental data. In respect to non-Newtonian liquids, mostly breakup modes, maximum cross-stream dimensions and breakup times were found to be available in literature. Therefore, there is much work needed to be done with these complex fluids.

A few of the most important findings from Newtonian and non-Newtonian secondary atomization literature are as follows:

2.3.1 Gas-Liquid Drop Flows

- If the Ma<0.3 the flow can be considered incompressible and is compressible otherwise.
- Virtual mass and Basset history forces are negligible if $\rho_L/\rho_G \gg 1$.
- Density variations do not affect drop deformation, breakup times, and breakup regimes if $\rho_L/\rho_G > 32$.
- Pressure gradient forces are negligible if the flow is uniform.
- Gravitational forces are negligible when a large relative velocity exists since accelerations may approach or exceed 100-200 times that of gravity.
- The major force acting on a drop is the stream-wise drag force which is a composed of form drag and skin friction drag. Form drag is approximately 80-95% of the total drag.

- The coefficient of drag for liquids is slightly less than solids of the same shape due to surface movement and internal circulation.
- *Re* effects are insignificant to deformation and breakup properties when $Re > 100$.
- Drop towers are limited to vibrational and bag breakup regimes.
- Shock tubes' initial shock waves do not significantly affect drop deformation and major deformation and breakup characteristics are the result of post convective flow which allows for comparison between continuous jet and drop tower experimental setups.
- For all regimes considered here (bag to sheet thinning breakup), all drops first deform from spherical to "ellipsoidal." The ellipsoidal shape is an approximation and the drop may actually be flat on the windward or leeward ends.
- Contamination such as plastic, or any unintentional surface active contamination can have a significant affect. The contaminants decrease interfacial movement and cause the coefficient of drag to increase. The impact becomes greater at low viscosity ratios.

2.3.2 Newtonian Liquids

- *We* and *Oh* are the critical parameters which influence deformation, breakup times, breakup regimes, fragments sizes, and all other secondary atomization characterizations.
- Since there are many names given to the same breakup modes, here are some of the various names given in literature (refer to Table 2.4):
- *We* required to initiate any mode of breakup increases with larger *Oh*.

Table 2.4. Breakup Nomenclature Found in the Literature.

Name of Breakup used here	**Vibrational**	**Bag-and-Stamen**	**Bag, Stamen, and Bag**	**Sheet Thinning**
Other names used in literature	Oscillatory	Club, Umbrella, Claviform, and Bag-Jet	Dual Bag	Plume/ Shear, Plume/Sheet Thinning

- There are discrepancies in literature of whether the maximum cross-stream diameter ratio, ${d_{cro,ini}}/{d_0}$, increases or is constant with *We*. This diameter does decrease as *Oh* becomes larger. The ratio ${d_{cro,ini}}/{d_{min,ini}}$ may provide a correlation which is independent of *Oh* and is linear with We.
- Maximum fragment sizes decrease with *We* and are twice the size of the mass median diameter. Average fragment diameters have the same trend with *We* and increase with *Oh*.
- While displacement may have a parabolic profile, velocity is not linear, and acceleration is not a constant. Parent drops can displace up to 50 times the initial diameter over the course of breakup.
- Windward and leeward continuous phase velocity differentials may account for breakup modes. Also, twin vortices occur on the leeward side of the drop in the continuous phase wake region.
- ${C_d}/{C_{d,Sphere}}$ $vs.$ ${d_{cro}}/{d_{min}}$ may provide a correlation which is only dependent on deformation and is independent of *We* and *Oh*.
- C_d has been found to be comparable initially to a sphere ($C_d \approx 0.4$) and to a disk ($C_d \approx 1.2$) at the maximum cross-stream dimension.
- There is discrepancy in literature as to whether the initiation time decreases or is constant with We. However, it does not increase with We. Yet, the initiation time does become greater with larger *Oh*.
- There are large discrepancies with respect to the total breakup time. Authors have observed a decrease in total breakup with *We* and also independence of the total breakup time with *We*. Yet, others even noted an increasing and decreasing trend depending on the *We* interval. However, unanimously the total breakup time becomes larger as *Oh* is increased.
- Rayleigh Tayler Piercing theory has been successfully used to predict the *We* boundaries of bag breakup and dual bag breakup with variation of *Oh*.

For Low *Oh* (*Oh*<0.1):

- *We* required to initiate breakup is independent of *Oh* as long as the condition (*Oh*<0.1) is met.
- The relative velocity at initiation time is approximately 90% of the flow field velocity.
- Simplified regime maps may be made in terms of u_{rel} $versus$ d_0.
- Rayleigh Taylor dispersion relations hold for stamen growth up to 40% of the critical wave length.
- Rayleigh Taylor theory has successfully been used to predict what breakup modes will occur. However, the theory degrades above sheet thinning breakup.
- Regimes begin within the following ranges:

Table 2.5. *We* Number Ranges Where Breakup Can Be Expected for Low Viscosity Liquids (*Oh* < 0.1).

Regime	**Vibrational**	**Bag**	**Bag-and-stamen**	**Dual Bag**	**Sheet Thinning**
We	3-10	10-13	16-18	26-28	32-40

2.3.3 Non-Newtonian Liquids

- All modes of breakup observed for Newtonian liquids are also found for viscoelastic and shear thinning liquids, except for the dual bag breakup mode. However, for coal slurries liquids which exhibit viscoelastic/ yield stress properties, the multimode regimes found for Newtonian liquids were not observed, only a similar mode as bag breakup.
- All non-Newtonian liquids produced ligaments instead of drops after any mode of breakup. The ligaments have been observed to eventually breakup into drops downstream. The size of these ligaments increases with polymer concentration and decreases with larger air velocities. Also, fragments are inherently larger with Non-Newtonian drop breakup.
- Increases in polymer concentration, causes the viscosity to increase and thus *Oh*. This results in larger *We* required to initiate a particular breakup mode and

decreased maximum cross-stream dimension. Also, higher molecular weighted polymers and increased coal concentrations have the same impact.

- Initiation time decreases with *We* for large *We* variation yet within small changes of *We* there isn't large effects on initiation time. Highly shear thinning liquids will exhibit smaller initiation times.
- Increases in polymer concentration result in increased total breakup time.
- Elasticity of a viscoelastic liquid may be negligible in determining the breakup modes which rely more heavily on the viscosity. However, elastic properties influence breakup after the breakup has been initiated (IE: size of nodes, breakup times, fragment sizes, etc.).
- Determining global effective viscosities is difficult, and as a result the *Oh*, due to local strain rate variations. Also, the effective viscosity may actually vary depending on what parameter is of interest (IE: breakup mode, maximum cross-stream dimension, fragment sizes, breakup times, etc.).
- Surfactants may or may not affect viscosity in a Non-Newtonian manner. Their main impact is to be surface active which causes C_d to increase by preventing surface movement and also causes increased deformation. There is a critical value where additional surfactant addition ceases to affect drop dynamics.
- Using zero shear viscosities to calculate *Oh* for a *We* versus *Oh* regime map match that for Newtonian liquids but only for low *Oh*. The *Oh* where this method is no longer applicable depends on the liquids viscosity versus strain rate profile. Also, higher *We* regimes result in sooner deviation from Newtonian *We* versus *Oh* maps as *Oh* is increased using this method.
- Modeling Non-Newtonian liquid viscosity using the Power Law model has been successful in predicting bag breakup mode on a Newtonian *We* versus *Oh* regime map.

CHAPTER 3. EXPERIMENTAL APPATUS AND MATERIALS

In this chapter the experimental setup, liquid properties, and methods of obtaining properties, measurements and their corresponding uncertainties will be presented.

The setup is that of a drop falling into a continuous air jet (Figure 3.1) with *We* varying between $10 < We < 50$. When the air velocity is large enough the drops begin deforming and eventually break up.

The process is illuminated using Newport's Xenon 1000 Watt arc lamp whose beam is expanded and diffused. The behavior of the disintegrating drops is captured using Vision Research's Phantom high speed camera v7.3 or v7.1.\

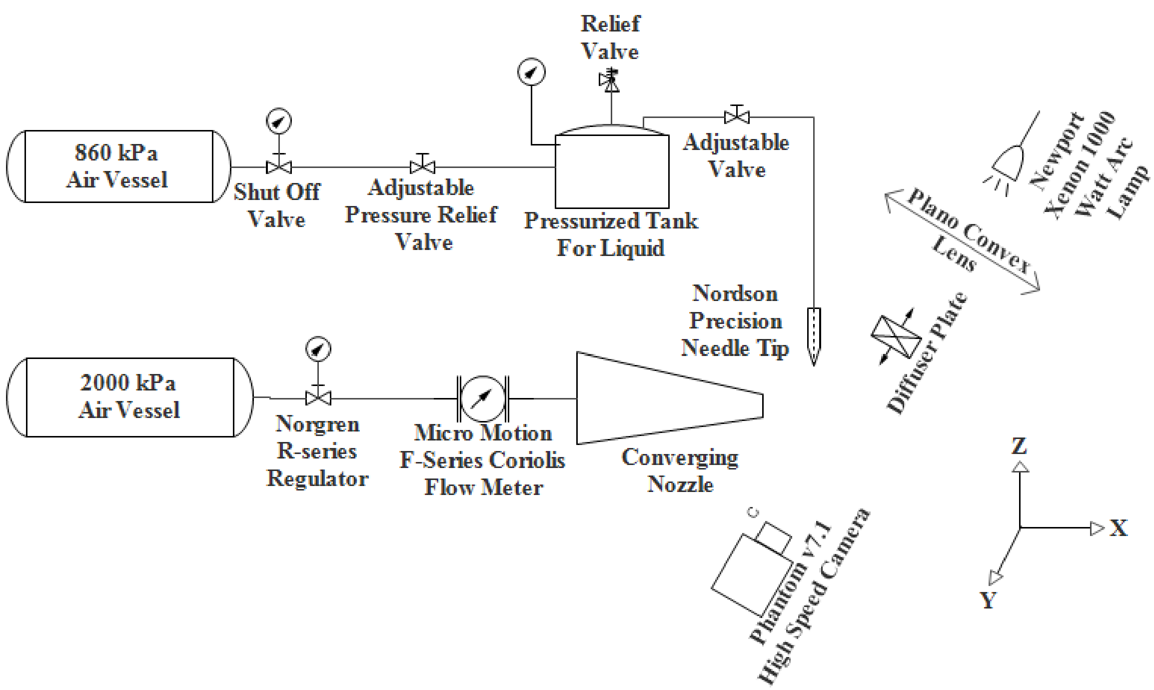


Figure 3.1. Experimental Setup Diagram.

3.1 Air Supply System

Dried and filtered facility air was stored in two separate tanks. One tank stored 2.65 m^3 at a pressure of 2000 kPa which is sufficient for hours of continuous operation. This air, which is used by the converging nozzle, is first passed through a *Norgren* R-17 regulator. The flow rate is monitored using an *Emerson* Micro Motion F-Series flow meter that is accurate to within ±0.1% of the value displayed by an *Emerson* Model 1700 integral mount transmitter. The air then enters the nozzle.

The other air tank supplies 860 kPa to pressurize the liquid supply system. This air stream first passes through a shut off valve and then an adjustable pressure relief valve. The air pressure supplied to the liquid tank is measured with a pressure gauge (50 kPa readability). Tank pressure was kept at 150 kPa (gage) to allow for the more viscous fluids through the syringe tip droplet generator.

3.2 Converging Nozzle

The nozzle, shown in Figure 3.2, is mounted on a *Velmex* 3D translator. *UniSlide* stepper motors and a programmable NF90 controller were used for positioning. As air flows in from the left of Figure 3.3 it passes through a 13 mm od tube and then enters the 15cm od entrance chamber through radially directed ports, which improves flow uniformity. The flow then passes through a 2.54 cm long polycarbonate honey comb having 4 mm cells. Its purpose is to suppress large scale eddies and reduce radial and swirling flow. Next the air flows through a wire mesh having 0.05 mm diameter wires with 0.07 mm spacing. This produces small scale turbulence which dissipates quickly to produce a steady, laminar, one-dimensional flow field. The converging section ends with a nozzle whose exit is 2.54 cm in diameter.

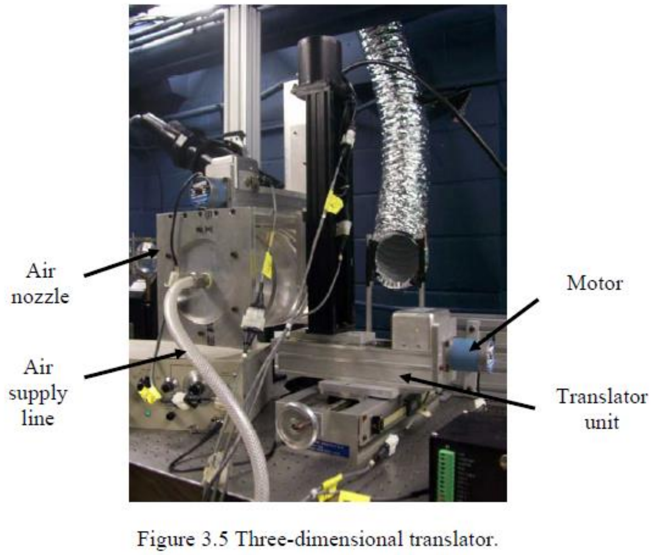


Figure 3.2. Air nozzle and translator setup. (Lopez, 2010).

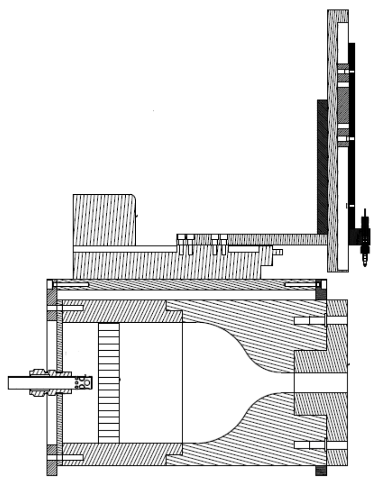


Figure 3.3. Nozzle-Liquid System. (Guildenbecher, 2009).

3.2.1 Nozzle Exit Velocity Profile

For a continuous jet secondary breakup apparatus it is desired to reduce boundary layer thickness and turbulence. To determine the level of these flow effects, the flow out of the converging nozzle was characterized by Guildenbecher (2009) and Lopez (2010). The most important results are summarized here.

A combination of PIV and PDA was used to find the mean air flow velocity, the boundary layer thickness, and level of velocity fluctuations. This required seeding the flow with olive oil drops (ρ = 915 kg/m^3) that had a mass median diameter of 2 μm (confirmed using PDA). To ensure the seed particles would follow the flow, the velocity lag was determined,

$$u_{lag} = \frac{\rho_L d_0^2}{18\mu_G} a_L \tag{3.1}$$

For this relation to be valid the condition Re<1 must be met and this occurs only if $u_{lag} < 7\ m/s$. The lag velocity was calculated using PIV velocity data, (a_L = 150 m/s^2) and found to be $0.002\frac{m}{s} \ll 7\frac{m}{s}$. This value is less than the level of turbulent velocity fluctuations.

As and added check, the momentum transfer from particles to the air flow was checked to ensure it was negligible. The air momentum flux for air is,

$$J_{air} = \rho_G u_G^2 \tag{3.2}$$

while that for the particles is,

$$J_{par} = \frac{\pi}{6} d_0^3 \rho_L u_G \dot{n} \frac{1}{A_{probe}} \tag{3.3}$$

where $\dot{n}$ is the number of particles per unit time as measured using LDV (6700 particles/s), A_{probe} is the probe cross sectional area (1.2 mm^2), and the mean speed u_G was 21 m/s. The resultant momentum ratio, $J_{air}/J_{par} \approx 1 \times 10^6$, demonstrates no effect.

The relaxation time for the seeded particles to attain velocity equilibrium with air was also determined,

$$t_p = d_p^2 \frac{\rho_p}{18\mu_G} \tag{3.4}$$

It was calculated to be 11 μs, which is sufficiently small compared to the experimental characteristic time of 47 μs (drop time interval as it deforms into an ellipsoid).

The 2-d air velocity profile in the radial direction, mean centerline velocity, and the centerline turbulent velocity fluctuations were obtained using a Dantec Dynamics FiberFlow PDA. Signal processing was accomplished using Dantec Dynamics BSA Flow Software version 3.00.00.17. The settings were optimized to achieve 99% data validation on the jet centerline with 10,000 samples being taken at each measurement location (starting at the centerline and moving radially outward).

Figure 3.4 shows that the velocity is uniform from 0 to 11mm for a range of mass flow rates. Also, there is a boundary layer which is approximately independent of flow rate whose size is 3mm. The measurement location and boundary layer dimensions are shown in Figure 3.5.

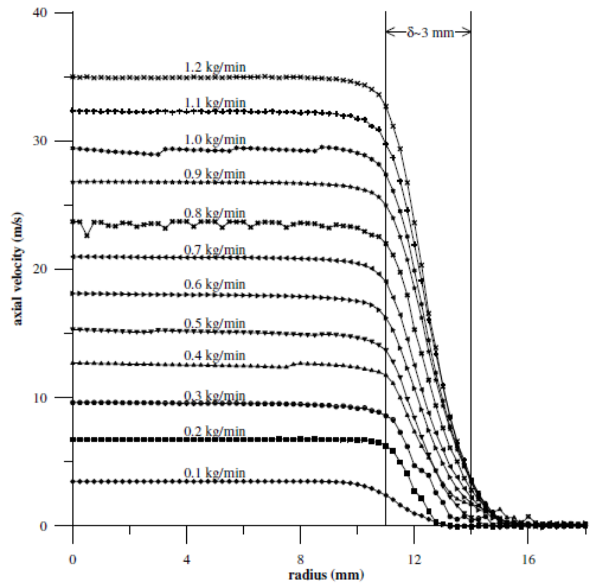


Figure 3.4. LDV axial velocity measurements versus radial distance from the centerline. (Guildenbecher, 2009).

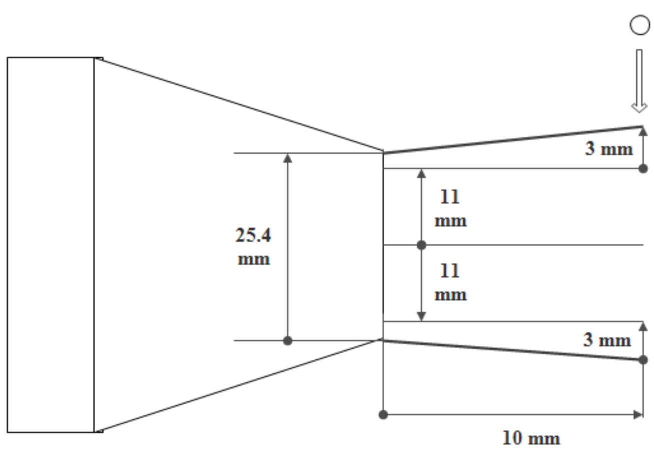


Figure 3.5. Nozzle, boundary layer, and drop locations.

The PDA measurements reported above were compared with values obtained using PIV. Agreement for centerline velocities is within ±0.5%.

3.3 Liquid Drop System

As discussed in the previous section, air is supplied to the liquid pressure vessel, Figure 3.6, Alloy Products Corp. with a maximum rated pressure of 900 kPa at 38°C. The liquid flow rate is controlled using a needle valve before passing to the drop generator. The drop generator (Figure 3.7) has a nylon body with liquid entering perpendicular to where the drops fall. The dispenser tips are *Nordson EFD* part number: 7018336. They have a length of 25.4 mm and id of 0.26 mm. The drop generating system is mounted on a 2-d *Velmex* traverse (model#: MB2506Q1J-S2.5) and the dispensing tip exit was positioned 10cm above the air jet exit.

Figure 3.6. Pressurized Liquid Vessel (Lopez, 2010).

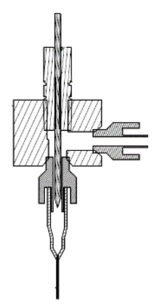


Figure 3.7. Drop Generator (Guildenbecher, 2009).

3.4 High Speed Camera and Lighting System

Drops were backlit using a *Newport* 6269 1000 W Xe arc lamp powered by a *Kratos Analytical Instruments* LPS 255 HR housing. The beam was expanded using a plano-convex lens (1:300mm, dia:152mm, Pyrex) and then diffused by a ground glass diffuser plate. The 2-d shadow produced by a deforming drop is captured by a Nikon AF-Micro Nikkor with a 105 mm focal length and an aperture of f/2.8. It was attached to a *Vision Research* Phantom v7.1 or v7.3 high speed camera. Framing rates of 4,796 or 6,660 fps (for the full 600x800 sensor) provide times between images of 0.21 or 0.15 ms, respectively. The camera images were sampled using Phantom Camera Control version 2.6.749.0 (64 bit) software.

3.5 Liquids

Test liquids were formed from mixtures of DI water and *Avantor Performance Materials'* USP grade 100% vegetable based glycerin (Gly), into which one of two polymers is mixed. They are ASHLAND's PH Sodium Carboxymethylcellulose (NaCMC) 7MF and 7HF (Lot Num. 70640 and 70525, respectively). The solutions are:

- 0.8 wt-% CMC-7MF
- 1.4 wt-%CMC-7MF
- 0.5 wt-%CMC-7HF
- 74.96 wt-% Gly/0.06 wt-% CMC-7MF
- 79.95 wt-%Gly/0.06 wt-% CMC-7MF
- 84.96 wt-%/0.05 wt-% CMC-7MF

in de-ionized (DI) water. The differences in solution viscosity are due to the polymer degree of substitution and molecular weight. High viscosity 7HF CMC has a DP of 3,200 and molecular weight (MW) of 700 kDa while the 7MF CMC has a DP of 1,100 and MW of 250 kDa.

There are four important considerations to ensure proper characterization of these liquids:

- At low frequencies of about 0.2 Hz they may exhibit some elasticity. However, after examination of the loss and storage modulus at the higher frequencies (>0.2Hz) relevant to secondary breakup the liquids have a much larger loss modulus (Mallory, 2012). This allows elasticity to be ignored.

- DI-water must be used for the assumption of inelasticity to hold (Bonferoni *et al*.,1995). As the ionic strength of a CMC solution is increased, the elastic modulus is significantly increased. Solution ionic behavior increases with dissolved mineral content, which is eliminated by using DI water.
- Dolz *et al*., (1991) found that the storage time of these liquids may affect their properties. Thus, all liquids were formulated, had their properties measured, and were used in experiments measured within 48 hrs of mixing, to prevent property variation, and no sooner than 12 hrs to allow the solution to stabilize.
- Lee *et al*. (2012) report that the Gibbs absorption theorem for biopolymers (including CMC) that are not generally surface active at concentrations less than about 2% should have a surface tension similar to that of water (this is observed for the test solutions, as shown in the following section).

3.6 Experimental Procedures, Uncertainties, and Liquid Properties

3.6.1 Lab Equipment Cleanliness and Temperature Control

Prior to use, each beaker or graduated cylinder is rinsed with isopropyl alcohol (*Mallinckrodt Analytical's* Manufacturer's Part No. 3043-10) due to its ability to remove non-polar compounds and dissolve oils. The glassware is then vigorously scrubbed with soapy hot water. After that, the glassware is rinsed 3 times with hot water to ensure that any soap residue is completely removed. Finally, it is placed upside-down in a dish rack to drain excess tap water which may contain minerals/ions. The same procedure is done for the magnetic stirrers.

For the liquid pressure tank, tubing, and nozzle assembly, the pressure tank was scrubbed and hot soapy water was run through the system 2 times for approximately 15 min to remove any residual liquid from the prior tests. Then, only hot water was running through the system 3 times for about 15 min each time to remove any residual soap. Finally, a compressed air gun was used to force excess tap water, which may contain minerals/ions, out through the liquid pressure vessel, the tubing, and the nozzle assembly.

There were some small variations between test temperatures and those for liquid property measurements. On average the test temperatures were 23.7^{o}C. The maximum temperature difference when measuring viscosity from the test temperature was 1.7^{o}C. In respect to Water viscosity, this would result in a maximum variation of about 3.6%. Surface tension and density were measured in the same room within about 1hr so the liquid temperature was the same. The maximum temperature difference for these properties from the test temperature was 2.8^{o}C. For water, this would result in a maximum percent difference in surface tension of 0.6% and 0.07% for density.

3.6.2 Mixing Procedure and Concentration Uncertainty

Before formulating any test liquid, the amounts of CMC, DI water, and glycerol are calculated and weighed. For all the two component solutions (water and CMC), the mass of water was placed into a beaker located atop a magnetic stirrer. The water mass was nominally constant at 700 g. The mass of CMC was measured next, then slowly poured into the water vortex that was established by the stirrer. Finally, the stirring speed is reduced to a point where the surface of the solution is still being disturbed and stirred for at least an hour. This yielded a visibly homogeneous solution.

Since there is uncertainty in measuring each mass there must be uncertainty in the CMC concentration. The absolute is calculated using (Taylor, 1982),

$$\delta C = \sqrt{\left(\frac{\partial C}{\partial m_{CMC}}\delta m_{CMC}\right)^2 + \left(\frac{\partial C}{\partial m_{Water}}\delta m_{Water}\right)^2} \tag{3.5}$$

Noting that $\delta m_{CMC} = \delta m_{Water} = \delta m$ because they are all measured using the same balance, and transforming to a relative composition uncertainty,

$$\frac{\delta C}{C} = \frac{\delta m}{m_{CMC}}\sqrt{1 - 2C + 2C^2} \tag{3.6}$$

The magnitude is less than 0.3% in all cases.

For the three component solutions (water, CMC, and glycerin) the total mass was nominally 600 g. Because these liquids are more difficult to mix a different mixing technique was used. First, separate beakers were partially filled with glycerin and DI-water.

CMC was dissolved in the water as per the two component solutions. The DI-water/CMC solution was poured into swirling glycerin and stirred for at least an hour to ensure homogeneity.

As with the 2 component solutions, these also must have some uncertainty due to the mass balance. Again following Taylor (1982),

$$\delta C = \frac{\delta m}{m_{Total}}\sqrt{1 - 2C + 3C^2} \tag{3.7}$$

and

$$\frac{\delta C}{C} = \frac{\delta m}{m_{CMC}}\sqrt{1 - 2C + 3C^2} \tag{3.8}$$

gives a maximum relative uncertainty below 3.5%.

3.6.3 Density and Uncertainties

Density was computed by measuring the mass of a known volume. A Pioneer Series Analytical model PA1502 balance having an uncertainty of 0.01g was used for mass measurements. Volume were measured using a 25 mL graduated cylinder (No: 3046-25) whose uncertainty is ±0.3 mL.

Table 3.1. Uncertainty in Liquid Density Measurements.

	ρ_{Mean} (kg/m^3)	$\delta\rho_{Instrument}$ (kg/m^3)	$\delta\rho_{Instrument}/\rho$ (%)
0.8% CMC-7MF	995	±12	±1.2
1.4% CMC-7MF	998	±12	±1.2
0.5% CMC-7HF	999	±12	±1.2
74.96% Glycerin/0.6%CMC-7MF	1180	±14	±1.2
79.95%Glycerin/0.06%CMC-7MF	1200	±14	±1.2
84.96%Glycerin/0.05%CMC-7MF	1210	±15	±1.2

These uncertainties are within 1.2%.

3.6.4 Surface Tension and Uncertainties

Test liquid surface tensions were measured using a CSC Precision DuNouy Tensiometer (Part No.70535) having manufacturer-stated repeatability of ± 0.05 mN/m. It was calibrated using,

$$P_{calc} = \frac{m \cdot g}{2 \cdot L_{cir}} \tag{3.9}$$

where P_{calc} is the expected reading, m is the mass to be measured, g is gravity, and L_{cir} is the circumference of the DuNouy ring (provided by the manufacturer to be 6.06491 cm). The calibrated accuracy was within 0.2 %.

Accuracy was further assessed by comparing measured surface tensions (DI water, isopropyl alcohol and glycerin) with literature values. All measurement readings were corrected per Zuidema and Waters (1941),

$$F_{correc} = a + \sqrt{\frac{4bP_{read}}{\pi^2 R^2 (\rho_L - \rho_G)} + c} \tag{3.10}$$

where a, b, and c are constants, P_{read} is the reading from the tensiometer and R is the radius of the DuNouy ring. The value of a is 0.7250, b is 0.0009075, and c is calculated using,

$$c = 0.04534 - 1.679 \frac{r}{R} \tag{3.11}$$

Here R is the radius of the DuNouy ring (0.96526 cm) and r is the radius of the wire (0.01573 cm), both provided by the manufacturer. To prevent contamination of the samples for each set of readings the DuNouy ring is heated to burn off any remaining impurities.

Again applying the procedure given by Taylor (1982), this time to the correction factor, *F*,

$$\delta F_{correc} = \frac{4b}{2\pi^2 R^2 (\rho_L - \rho_G)(F_{correc} - a)} \sqrt{\delta P_{read}^2 + \frac{P_{read}^2}{(\rho_L - \rho_G)^2} (\delta\rho_L^2 + \delta\rho_G^2)} \tag{3.12}$$

where the uncertainties in air densities are based on uncertainties in temperature readings from a Cole-Parmer Thermohygrometer (item#: EW-03313-85) whose accuracy is ±1 ^{0}C. Results are provided in Table 3.2, and show that agreement is within 1.2%.

Table 3.2. Measured σ, uncertainties, and comparison to standard values.

	Water	**Isopropyl Alcohol**	**Glycerin**
P_{Avg} **(mN/m)**	76.99	24.05	69.32
ρ_L **(kg/m^3)**	998	784	1260
$\delta\rho_L$ **(kg/m^3)**	±0.250	±0.836	±0.612
ρ_G **(kg/m^3)**	1.20	1.20	1.20
$\delta\rho_G$**(kg/m^3)**	±0.004	±0.004	±0.004
F_{correc}	0.945	0.898	0.924
δF_{correc}	±0.00012	±0.00012	±0.00010
σ **(mN/m)**	72.77	21.61	64.07
$\delta\sigma$ **(mN/m)**	±0.130	±0.0695	±0.112
$\sigma_{Literature}$ **(mN/m)**	72.64	21.60	63.30
Difference from Literature (%)	0.18	0.018	1.2

Using the same procedure and calculations as discussed above, surface tension measurements for the test fluids are always less than 0.4%. See Table 3.3 and 3.4.

Table 3.3. Measured σ and uncertainties for the two component test solutions.

	0.8% CMC-7MF	**1.4% CMC-7MF**	**0.5% CMC-7HF**
P_{Avg} **(mN/m)**	76.84	77.02	76.51
ρ_L **(kg/m^3)**	995	998	999
$\delta\rho_L$ **(kg/m^3)**	±12	±13	±12
ρ_G **(kg/m^3)**	1.20	1.19	1.20
$\delta\rho_G$ **(kg/m^3)**	±0.0040	±0.0040	±0.0040
F_{correc}	0.945	0.945	0.945
δF_{correc}	±0.00084	±0.00089	±0.00084
σ**(mN/m)**	72.64	72.80	72.28
$\delta\sigma$**(mN/m)**	±0.0855	±0.0954	±0.138
$\delta\sigma/\sigma$ **(%)**	±0.12	±0.13	±0.19

Table 3.4. Measured σ and uncertainties for the three component test solutions.

	74.96% Glycerin/ 0.6%CMC-7MF	**79.95%Glycerin/ 0.06%CMC-7MF**	**84.96%Glycerin/ 0.05%CMC-7MF**
$\boldsymbol{P_{Avg}}$ **(mN/m)**	71.13	70.25	70.04
$\boldsymbol{\rho_L}$ **(kg/m3)**	1182	1198	1209
$\boldsymbol{\delta\rho_L}$ **(kg/m3)**	±18	±14	±15
$\boldsymbol{\rho_G}$ **(kg/m3)**	1.20	1.20	1.19
$\boldsymbol{\delta\rho_G}$ **(kg/m3)**	±0.0040	±0.0040	±0.0040
$\boldsymbol{F_{correc}}$	0.929	0.928	0.927
$\boldsymbol{\delta F_{correc}}$	±0.00090	±0.00071	±0.00068
$\boldsymbol{\sigma}$ **(mN/m)**	66.11	65.19	64.94
$\boldsymbol{\delta\sigma}$ **(mN/m)**	±0.0925	±0.207	±0.0697
$\boldsymbol{\delta\sigma/\sigma}$ **(%)**	±0.14	±0.32	±0.11

3.6.5 Viscosity, Model, and Uncertainties

Rheological characterization was done using TA Instruments AR-G2 Rheometer in the Peltier setup. The experiments were run as a flow sweep from 1000 to 0.1 s^{-1}, choosing five points per decade with three repetitions at each point and a 5% tolerance. Finally, for each experimental liquid at least 3 flow sweep measurements (providing stress, viscosity, normal stress, and torque versus strain rate) were taken. Therefore, at least 9 measurements of viscosity at each strain rate was measured. Additionally, the procedure was evaluated by measuring the viscosities of glycerin/DI water solutions having varying concentrations (75% glycerin/25% water, 80% glycerin/20% water, and 85% glycerin/15% water). Results were compared to values provided by Dorsey (1940). See Figure 3.8.

The percent difference from literature is significant (up to 7%) only when the strain rate is below 0.63 s^{-1}. Above that strain rate the percent difference is at most 1.7%.

Using the aforementioned procedure, viscosity versus strain rate data for the experimental liquids are shown in Figure 3.9.

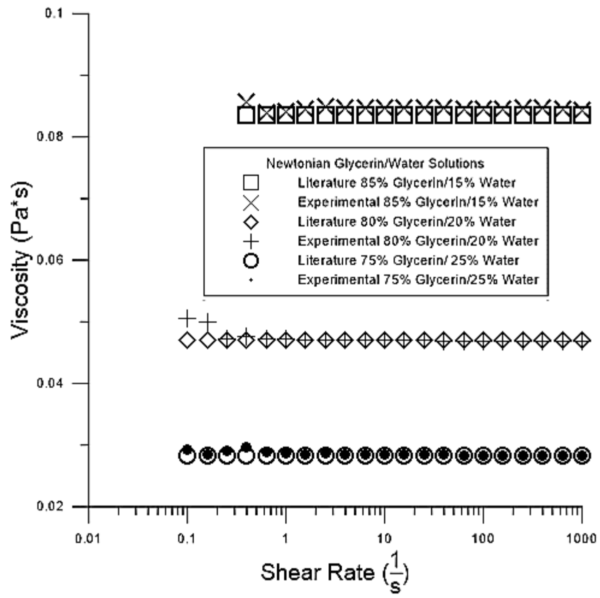


Figure 3.8. Newtonian Glycerin/Water Solutions compared to Literature Values.

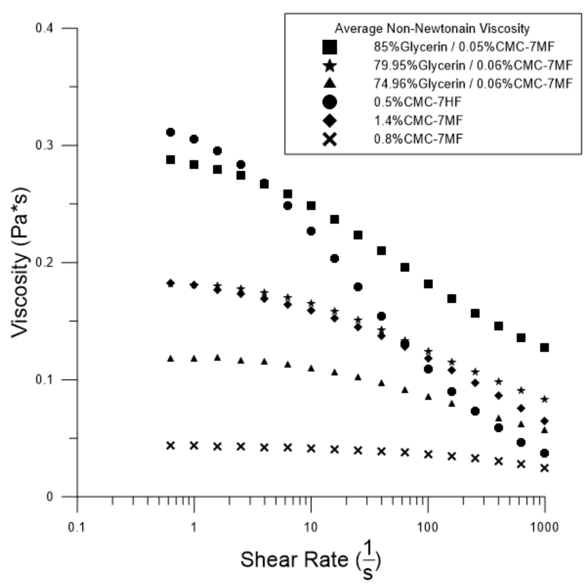


Figure 3.9. Viscosity versus strain rate for Non-Newtonian Liquids.

From here, it is obvious that these liquids are all shear thinning or pseudo plastic. Consequently, the power law or Ostwald-de Waele model is appropriate,

$$\tau = K(\dot{\gamma})^n \qquad (3.13)$$

where τ is the stress (Pa), K is the flow consistency index ($Pa \cdot s^n$), n is the flow behavior index (dimensionless), and $\dot{\gamma}$ is the strain rate (s^{-1}). K and n are extracted by taking the natural log (ln) of both sides.

Using the methods of Taylor (1982), the uncertainties in *K* and *n* are,

$$\delta K = \frac{\partial K}{\partial Ln(K)} \delta ln(K) = e^{ln(K)} \cdot \delta ln(K) \tag{3.14}$$

and

$$\delta ln(K) = t_{\alpha/2} \frac{s}{ln(K)_{Avg}\sqrt{N}} \tag{3.15}$$

The results provided in the table below,

Table 3.5. Results and Uncertainties for *n* and *K*.

	n (*)	$\delta n/n$ (%)	$K(Pa \cdot s^n)$	$\delta K/K$ (%)
0.8% CMC-7MF	0.933	0.23	0.0464	2.8
1.4% CMC-7MF	0.867	0.25	0.200	3.8
0.5% CMC-7HF	0.708	0.26	0.375	2.1
74.96% Glycerin/0.6%CMC-7MF	0.899	0.31	0.128	2.5
79.95%Glycerin/0.06%CMC-7MF	0.891	0.65	0.197	1.7
84.96%Glycerin/0.05%CMC-7MF	0.884	0.48	0.304	7.2

Note that the uncertainties in n are all under 0.7% and those in K are under 7.3%.

The *n* and *K* uncertainties are combined to compute the uncertainty in effective viscosity, μ_{eff},

$$\mu_{eff} = K(\dot{\gamma})^{n-1} \tag{3.16}$$

where the strain rate is approximated using data describing the drop transverse growth,

$$\dot{\gamma} = \frac{\Delta d_{cro}}{\Delta t d_0} = \frac{(d_{cro,ini} - d_0)}{d_0 t_{ini}} = \left(\frac{d_{cro,ini}}{d_0} - 1\right)\frac{1}{t_{ini}} = \frac{(d^*_{cro,ini} - 1)}{t_{ini}} \tag{3.17}$$

3.6.6 Uncertainties in D_{drop} and Non-Dimensional Groups

3.6.6.1 Uncertainties in D_{drop}

Uncertainties for camera images are ±1 pixel. The image of a calibration grid along the nozzle axis with 5 mm squares was used to transform camera array distances in pixels to mm. The corresponding uncertainty is 0.073mm for water and 0.101 mm for the other liquids tested. The difference is due to water breakup fitting within a smaller area (~55mm X 40mm) in contrast to the non-Newtonian liquids (~80mm X 60mm) so a higher magnification was possible for water .

For each dimensional measurement taken, the relative uncertainty can be calculated as,

$$\varphi = \frac{\delta L}{L_{Avg}} \tag{3.18}$$

where L is the length of the dimension taken and δL is the uncertainty related to the ±1 pixel assumption. For water $\delta L = 0.073\ mm$ since the breakup did fit in a smaller window and for the Non-Newtonian liquids $\delta L = 0.101\ mm$. Diameter measurements and uncertainties are as follows,

Table 3.6**.** Drop diameter, D_{drop}, measurements and uncertainties.

	d_o (mm)	δd_0(mm)	$\delta d_0/d_0$ (%)
Water	2.66	0.073	2.7
0.8% CMC-7MF	2.74	0.101	3.7
1.4% CMC-7MF	2.67	0.101	3.8
0.5% CMC-7HF	2.70	0.101	3.7
74.96% Glycerin/0.6%CMC-7MF	2.57	0.101	3.9
79.95%Glycerin/0.06%CMC-7MF	2.50	0.101	4.0
84.96%Glycerin/0.05%CMC-7MF	2.53	0.101	4.0

3.6.6.2 Uncertainties in Weber Number

The uncertainty in *We* (Equation 1.1) varies from liquid to liquid since the uncertainties in properties differ with each liquid. Adopting the methods discussed by Taylor (1982),

$$\frac{\delta We}{We} = \sqrt{\left(\frac{\delta \rho_G}{\rho_G}\right)^2 + \left(\frac{\delta d_0}{d_0}\right)^2 + \left(2\,\frac{\delta u_G}{u_G}\right)^2 + \left(\frac{\delta \sigma}{\sigma}\right)^2} \quad (3.19)$$

Carrying out the calculations, the uncertainties are shown in the Table 3.7 below. Note that the *We* uncertainties are always below 7.4%.

Table 3.7. Uncertainties in *We* for the DI-Water and Non-Newtonian Liquids.

	$\delta d_0/d_0$ (%)	$\delta \rho_G/\rho_G$ (%)	$\delta \sigma/\sigma$ (%)	$2\cdot \delta u_G/u_G$ (%)	$\delta We/We$ (%)
Water	2.75	0.335	0.201	6	6.6
0.8% CMC-7MF	3.69	0.335	0.118	6	7.1
1.4% CMC-7MF	3.79	0.335	0.131	6	7.1
0.5% CMC-7HF	3.74	0.334	0.191	6	7.1
74.96% Glycerin/ 0.6%CMC-7MF	3.93	0.335	0.140	6	7.2
79.95%Glycerin/ 0.06%CMC-7MF	4.05	0.334	0.318	6	7.3
84.96%Glycerin/ 0.05%CMC-7MF	3.99	0.336	0.107	6	7.2

The non-Newtonian *Oh* is defined as,

$$Oh = \frac{K\left(\left(\frac{d_{cro,ini}}{d_0}-1\right)\frac{1}{t_{ini}}\right)^{n-1}}{\sqrt{d_0\sigma\rho_L}} = \frac{K\left(\frac{\left(d^*_{cro,ini}-1\right)}{t_{ini}}\right)^{n-1}}{\sqrt{d_0\sigma\rho_L}}$$

(3.20)

The corresponding uncertainties are,

$$\frac{\delta Oh}{Oh} = \sqrt{\left(\frac{\delta K}{K}\right)^2 + \left(\frac{(n-1)\delta d^*_{cro,ini}}{(d^*_{cro,ini}-1)}\right)^2 + \left(ln\left(\frac{(d^*_{cro,ini}-1)}{t_{ini}}\right)\delta n\right)^2 + \left(\frac{\delta \rho_L}{2\cdot\rho_L}\right)^2 + \left(\frac{\delta\sigma}{2\cdot\sigma}\right)^2 + \left(\frac{\delta d_0}{2\cdot d_0}\right)^2 + \left(\frac{(1-n)}{t_{ini}}\delta t_{ini}\right)^2}$$

(3.21)

Applying gives the results shown in Table 3.8 below, and a maximum uncertainty of 8.1%.

Table 3.8. Uncertainties in *Oh* for Non-Newtonian Liquids.

	0.8% CMC-7MF	**1.4% CMC-7MF**	**0.5% CMC-7HF**	**74.96%Gly/ 0.6% CMC-7MF**	**79.95%Gly/ 0.06% CMC-7MF**	**84.96%Gly/ 0.05% CMC-7MF**
$\frac{\delta K}{K}$ (%)	2.8	3.8	2.1	2.5	1.7	7.2
$\frac{(n-1)\delta d^*_{cro,ini}}{(d^*_{cro,ini}-1)}$ (%)	0.88	1.4	4	1.2	1.1	1.7
$\frac{\delta \rho_L}{2\cdot\rho_L}$ (%)	0.6	0.64	0.6	0.77	0.6	0.6
$\frac{\delta \sigma}{2\cdot\sigma}$ (%)	0.059	0.066	0.096	0.07	0.16	0.054
$\frac{(1-n)\delta t_{ini}}{t_{ini}}$ (%)	0.8	1.9	3.3	0.97	1.5	1.4
$\frac{\delta d_0}{2d_0}$ (%)	1.8	1.9	1.9	2	2	2
$ln\left(\frac{(d^*_{cro,ini}-1)}{t_{ini}}\right)\delta n$ (%)	1.1	1.1	0.98	1.4	3	2.1
$\frac{\delta Oh}{Oh}$ (%)	3.8	5	6.1	3.8	4.5	8.1

For DI water,

$$\frac{\delta Oh}{Oh} = \sqrt{\left(\frac{\delta\mu_L}{\mu_L}\right)^2 + \left(\frac{\delta\rho_L}{2\cdot\rho_L}\right)^2 + \left(\frac{\delta\sigma}{2\cdot\sigma}\right)^2 + \left(\frac{\delta d_0}{2\cdot d_0}\right)^2} \tag{3.22}$$

with results presented in the table below,

Table 3.9. Uncertainty in *Oh* of DI-Water.

	$\frac{\delta\rho_L}{2\cdot\rho_L}$ (%)	$\frac{\delta\sigma}{2\cdot\sigma}$ (%)	$\frac{\delta d_0}{2\cdot d_0}$ (%)	$\frac{\delta\mu_L}{\mu_L}$ (%)	$\frac{\delta Oh}{Oh}$ (%)
De-Ionized Water	0.0125	0.101	1.39	2.16	2.6

Note that the maximum uncertainty is 2.6 %.

3.6.6.3 Uncertainties in Re_L

Re_L is defined as,

$$Re_L = \frac{\rho_G d_0 u_{rel}}{\mu_L} = \frac{\rho_G d_0 u_{rel}}{\left(\left(\frac{d_{cro,ini}}{d_0}-1\right)\frac{1}{t_{ini}}\right)^{n-1}} \tag{3.23}$$

The uncertainties are,

$$\frac{\delta Re_L}{Re_L} = \sqrt{\left(\frac{\delta K}{K}\right)^2 + \left(\frac{(1-n)\delta d^*_{cro,ini}}{(d^*_{cro,ini}-1)}\right)^2 + \left(ln\left(\frac{d^*_{cro,ini}-1}{t_{ini}}\right)\delta n\right)^2 + \left(\frac{\delta\rho_G}{\rho_G}\right)^2 + \left(\frac{\delta d_0}{d_0}\right)^2 + \left(\frac{\delta u_G}{u_G}\right)^2 + \left(\frac{(n-1)\delta t_{ini}}{t_{ini}}\right)^2} \tag{3.24}$$

which gives the values shown below.

Table 3.10. Uncertainty in Re_L for Non-Newtonian Liquids.

	0.8% CMC-7MF	**1.4% CMC-7MF**	**0.5% CMC-7HF**	**74.96% Gly/ 0.6% CMC-7MF**	**79.95%Gly/ 0.06% CMC-7MF**	**84.96%Gly/ 0.05% CMC-7MF**
$\frac{\delta\rho_G}{\rho_G}$ **(%)**	0.34	0.34	0.33	0.34	0.33	0.34
$\frac{\delta u_G}{u_G}$ **(%)**	3	3	3	3	3	3
$\frac{\delta d_0}{d_0}$ **(%)**	3.7	3.8	3.7	4	4	4
$\frac{\delta K}{K}$ **(%)**	2.8	3.8	2.1	2.5	1.7	7.2
$Ln\left(\frac{d^*_{cro,ini}-1}{t_{ini}}\right)\delta n$ **(%)**	1.1	1.1	0.98	1.4	3	2.1
$\frac{(1-n)\delta d^*_{cro,ini}}{(d^*_{cro,ini}-1)}$ **(%)**	0.88	1.4	4	1.2	1.1	1.7
$\frac{(n-1)\delta t_{ini}}{t_{ini}}$ **(%)**	0.8	1.9	3.3	0.97	1.5	1.4
$\frac{\delta Re_L}{Re_L}$ **(%)**	5.7	6.7	7.5	5.9	6.4	9.3

Here the maximum uncertainty is 9.3%. For the DI-Water,

$$\frac{\delta Re_L}{Re_L} = \sqrt{\left(\frac{\delta \mu_L}{\mu_L}\right)^2 + \left(\frac{\delta \rho_G}{\rho_G}\right)^2 + \left(\frac{\delta u_G}{u_G}\right)^2 + \left(\frac{\delta d_0}{d_0}\right)^2} \qquad (3.25)$$

which gives,

Table 3.11. Uncertainties in Re_L for DI-Water.

	$\frac{\delta \rho_G}{\rho_G}$ (%)	$\frac{\delta u_G}{u_G}$ (%)	$\frac{\delta d_0}{d_0}$ (%)	$\frac{\delta \mu_L}{\mu_L}$ (%)	$\frac{\delta Re_L}{Re_L}$ (%)
De-Ionized Water	0.335	3	2.77	2.16	4.63

Note that the uncertainty is under 5 %.

CHAPTER 4. RESULTS AND ANALYSIS

In this study, three breakup regimes were considered including bag, bag-and-stamen, and dual bag breakup modes using seven liquids. Six of these are inelastic non-Newtonian liquids composed of combinations with DI-Water, Glycerin, and Sodium Carboxymethyl Cellulose (CMC) both 7MF and 7HF. The seventh liquid acts as the control of the experiment which is highly studied Newtonian liquid water. Furthermore, the results are based on a MATLAB code developed. The code itself can be found in Appendix A. All results in the plots which follow are averages from 6-12 data points, or in other words, 6-12 videos were post processed in order to obtain one data point in the plot. For uncertainty, besides those discussed in the materials section, are the standard deviation in order to give a statistical measure of the uncertainty for each point. However, based on the number of measurements for each point, the 95% confidence interval would be either 5% larger or 10% smaller for 6 and 12 data points, respectively.

The total number of videos post processed using MATLAB was 382. This means each plot is composed of 382 measurements. Furthermore, since there are about 100 frames per movie then about 38,200 frames were processed. For each frame, the cross-stream dimension, stream-wise/bag dimension, area, and centroids are obtained with time. Therefore, approximately 153,000 total measurements are obtained which is by far impractical to obtain manually and likely less accurate. Actually, more data points are gathered however will not be discussed here.

To obtain those data points accurately while the drop deforms/breakups up, varies on its axis (for asymmetric occurrences), and deal with the significant variations between 3 regimes (bag, bag-and-stamen, and dual bag breakup) the code required significant sophisticated methods. Then, those data points are further processed (IE: velocity, acceleration, coefficient of drag, rate of bag and cross-stream length growth, We, *Oh*, etc.).

Finally, the velocity, cross-stream length, and initial diameter are used with the non-Newtonian TAB first developed by Lopez (2010) and then improved in this study.

Additionally, understanding what is occurring on the molecular and macroscopic levels with these non-Newtonian liquids is valuable in understanding the results to come. In respect to power law parameters, larger consistency index, K, values mean that the zero shear viscosity is larger and flow behavior index, n, values indicate the degree of shear thinning behavior. Therefore, as n decreases ($n<1$) the effective viscosity will deviate at a faster rate from the zero shear viscosity, K, as the liquid is sheared. At the same strain rate, increases in K causes the effective viscosity to rise and decreases in n causes it to drop. In this respect, it may be more convenient to think of the actual effective viscosity having a positive correlation with *K* and *n*.

In terms of physics/rheology, if *K* is larger it is because of a combination of the solvent viscosity being larger, polymer chains being entangled, and hydrogen bonds occurring between the solvent liquid and the polymer. As the liquid is sheared, the effective viscosity decreases due of the entangled polymers becoming disentangled and alignment in the direction of shearing. Additionally, the decrease in n implies that the polymer chains require much less forces to disentangle themselves. Before beginning the numerical results, first it is valuable to visualize the breakup modes and to determine if any significant differences exists between the liquids.

4.1 <u>Visualization</u>

The breakup regimes each occur at various *We* number depending on the liquid chosen and its effective viscosity. Due to this, for bag breakup the images at a *We* number prior to the onset of Bag-and-stamen breakup is shown. Also, for Bag-and-stamen an approximate midpoint within the regime was chosen since these represent more of the ideal image of Bag-and-stamen. Finally, the dual bag regime images are provided. Additionally, the drop diameter, length of bag at bag breakup, and non-dimensional time at specific points are provided in the captions to give a sense of the scale at which this breakup is occurring and the time at which they occur.

4.1.1 Bag Breakup Regime

From Figures 4.1 thru 4.7 below, it is clear that the general morphology within the bag breakup regime are similar for the Newtonian and non-Newtonian Liquids. First, the spherical drop deforms to a disk-like shape. After that, bag growth occurs from the center of the disk in the downstream direction. The bag resembles a thin membrane-like bag with a much thicker rim. Next, the bag breaks up first from the downstream end and gradually toward the basal ring. Afterwards, a series of large nodes which form along the ring. This is followed by breakup of the ring into a circular array of relatively large drops which ends the breakup process.

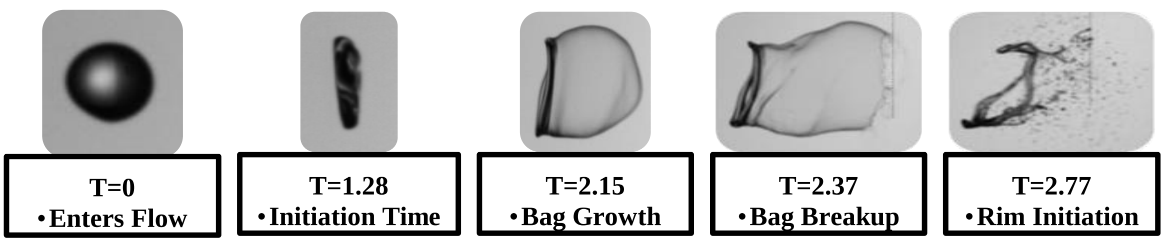


Figure 4.1. DI-Water (n=1.00, K=0.00094) in the Bag Breakup Regime, We=15.
$d_0 = 2.64mm \,; L_{Bag} = 12.78mm$

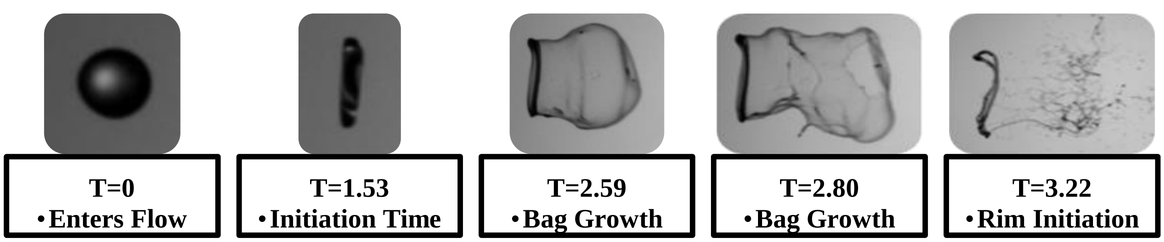


Figure 4.2. 0.8%CMC-7MF (n=0.93, K=0.046) in the Bag Breakup Regime, We=18.
$d_0 = 2.74mm \,; L_{Bag} = 16.1mm$

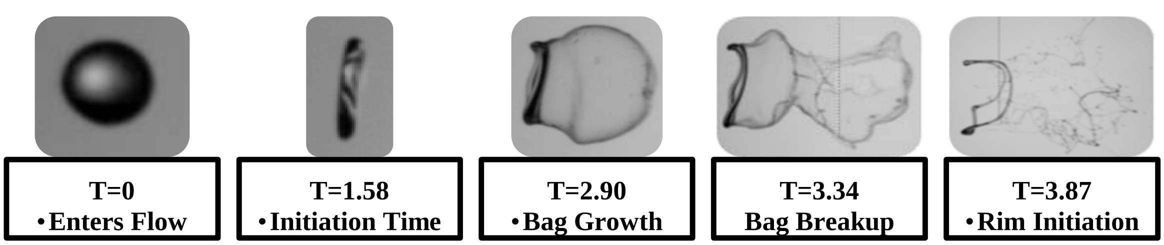


Figure 4.3. 1.4%CMC-7MF (n=0.87, K=0.20) in the Bag Breakup Regime, We=21.
$d_0 = 2.63mm \,; L_{Bag} = 16.8mm$

Figure 4.4. 0.5%CMC-7HF (n=0.71, K=0.37) in the Bag Breakup Regime, We=20. $d_0 = 2.72mm\,; L_{Bag} = 20.2mm$

Figure 4.5. 74.96%Gly/0.06%CMC-7MF (n=0.90, K=0.13) in the Bag Breakup Regime, We=21. $d_0 = 2.54mm\,; L_{Bag} = 18.8mm$

Figure 4.6. 79.95%Gly/0.06%CMC-7MF (n=0.89, K=0.20) in the Bag Breakup Regime, We=24. $d_0 = 2.51mm\,; L_{Bag} = 17.9mm$

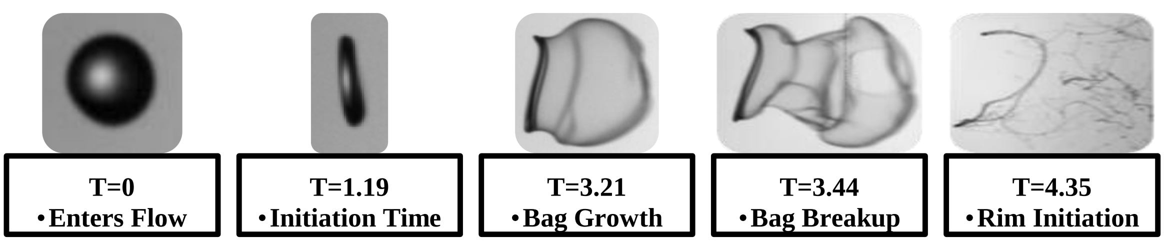


Figure 4.7. 84.96%Gly/0.05%CMC-7MF (n=0.88, K=0.30) in the Bag Breakup Regime, We=27. $d_0 = 2.53mm\,; L_{Bag} = 13.6mm$

There are a few distinct differences when comparing Newtonian and non-Newtonian liquids. After the bag bursts, Newtonian liquids produce only drops. In contrast, non-Newtonian liquids produce very few drops and mostly ligaments which persist further downstream. In respect to the power law parameters, as K is increased the ligaments increase in size and number while the number of drops decreases (refer to Figures 4.5

through 4.7). The affect is not altered by decreasing *n*, as can be seen in Figure 4.4. These ligaments will eventually break apart via an instability along their major axis.

This effect of polymer addition was noted in Wilcox *et al*. (1961) who compared non-Newtonian drop breakup with that for highly viscous Newtonian liquids. In that study, the Newtonian viscosity was increased by a factor of 25 with no ligaments being observed, yet when the non-Newtonian effective viscosity was increased by as little as a factor of 4 there was significant ligament formation. Further support comes from the study of Joseph *et al*. (1999) who observed drop breakup in the shear thinning/catastrophic breakup regime for both highly viscous Newtonian and non-Newtonian liquids and found persistent ligament formation for only the non-Newtonian case.

In addition, the polymer effect on ligament formation is more substantial for solutions containing glycerin. This may be due to the structure of the glycerin molecule, which has 3 hydroxyl groups (*OH*) so is polar and water soluble. Under lower strain rate conditions (when the bag has ruptured), it is possible that hydrogen bonding with the more rigid CMC, which also has 2 or 3 *OH* groups and a carboxymethyl group, may resist shearing and remain linked in small clusters. It's also possible that the clusters link to one another forming ligaments. Furthermore, the CMC-7HF polymer chain lengths are about 3 times those of the CMC-7MF. The individual polymer units have more *OH* and carboxymethyl groups, which may bond to more water molecules and link to neighboring clusters to form ligaments. These types of bonds have been discussed by Yang and Zhu (2007).

An additional distinction between Newtonian and inelastic non-Newtonian drop breakup occurs during bag growth, where Newtonian drop breakup forms a more uniformly rounded bag whereas the non-Newtonian drops produce a disturbed/wave-like bag that grows significantly. This can be observed in Figures 4.8 to 4.14, where it is apparent that disturbed bag growth is absent. Therefore, the difference must reside with the fact that non-Newtonian liquids produce longer bags before breaking.

Finally, an experiment to test whether hydrogen bonding, or polymer entanglement, is responsible for the above observations is to form a solution having a non-polar polymer dissolved into a non-polar solution. The intermolecular bonds would be much weaker than for polar polymer/polar solvent solutions so if different behavior (no ligaments) were observed then hydrogen bonding must play a role.

4.1.2 Bag-and-stamen Breakup Regime

Bag-and-stamen breakup for inelastic non-Newtonian liquids is qualitatively similar to that for Newtonian drop breakup, as can be seen in Figures 4.8 through 4.14. Like their Newtonian counterparts, inelastic non-Newtonian drops that are initially spherical deform into disk-like shapes except that there with a bulge in the center where the stamen will emerge. As the bag grows, a stamen projects parallel to the flow and along the windward direction, it disconnects from the basal ring and remains connected to the bag, then separates from the bag(s) as they break up. After the bag bursts, the basal ring begins to break up into drops, followed by the stamen.

The discrepancies between Newtonian and inelastic non-Newtonian bag breakup are observed for bag-and-stamen breakup: increasing K increases the number and size of ligaments while the number of drops decreases. Again, there is no effect when decreasing n.

Figure 4.8. DI-Water (n=1.00, K=0.00094) in the Bag and Stamen Breakup Regime, We=19. $d_0 = 2.60mm$; $L_{Bag} = 15.4mm$

Figure 4.9. 0.8%CMC-7MF (n=0.93, K=0.046) in the Bag and Stamen Breakup Regime, We=21. $d_0 = 2.75mm$; $L_{Bag} = 13.3mm$

A difference between Newtonian and inelastic non-Newtonian bag-and-stamen occurs after the bag bursts when the ligaments which held the stamen to the bag rim also remain and flow downstream with the stamen. All such ligaments will eventually form nodes on themselves and will break apart due to an axial instability, for reasons discussed in the previous section.

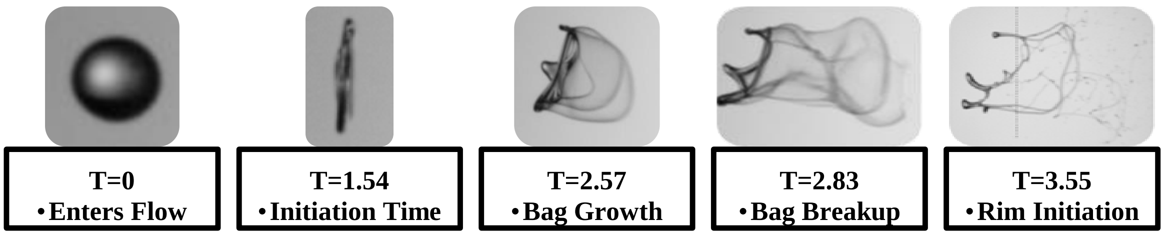


Figure 4.10. 1.4%CMC-7MF (*n*=0.87, *K*=0.20) in the Bag and Stamen Breakup Regime, *We* =30. $d_0 = 2.68mm$; $L_{Bag} = 14.0mm$

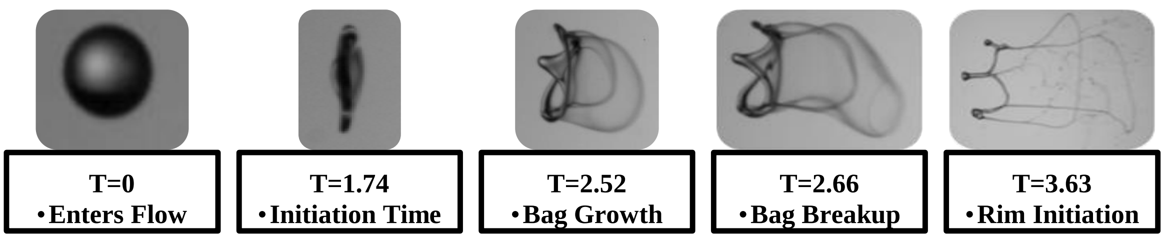


Figure 4.11. 0.5%CMC-7HF (*n*=0.71, *K*=0.37) in the Bag and Stamen Breakup Regime, *We*=27. $d_0 = 2.66mm$; $L_{Bag} = 10.8mm$

Figure 4.12. 74.96%Gly/0.06%CMC-7MF (*n*=0.90, *K*=0.13) in the Bag and Stamen Breakup Regime, *We*=28. $d_0 = 2.59mm$; $L_{Bag} = 11.0mm$

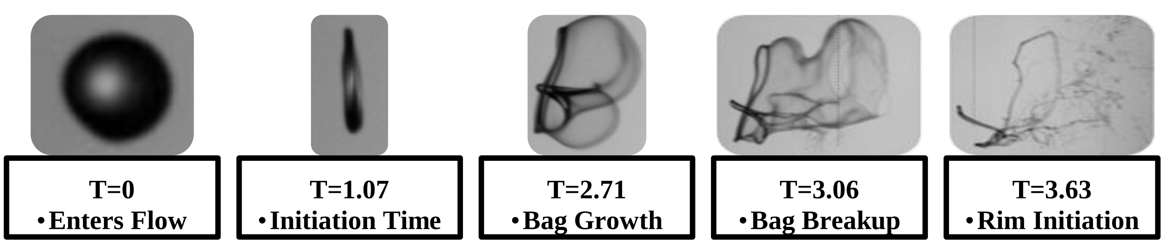


Figure 4.13. 79.95%Gly/0.06%CMC-7MF (*n*=0.89, *K*=0.20) in the Bag and Stamen Breakup Regime, *We*=32. $d_0 = 2.49mm$; $L_{Bag} = 13.9mm$

Figure 4.14. 84.96%Gly/0.05%CMC-7MF n=0.88, K=0.30) in the Bag and Stamen Breakup Regime, We=37. $d_0 = 2.53mm$; $L_{Bag} = 11.2mm$

4.1.3 Dual Bag Breakup Regime

Dual bag breakup morphology for non-Newtonian inelastic drops is qualitatively similar to that for Newtonian drops. See Figures 4.15 through 4.21. In this case the initially spherical drop becomes ellipsoidal, but with the disk-like center feature being more dome-like. Bag-and-stamen formation then occurs with the stamen parallel to the flow and connected to the rim through ligaments. These connections results in two or more bags, the exact number depending on the number of ligaments which are connected to the stamen.

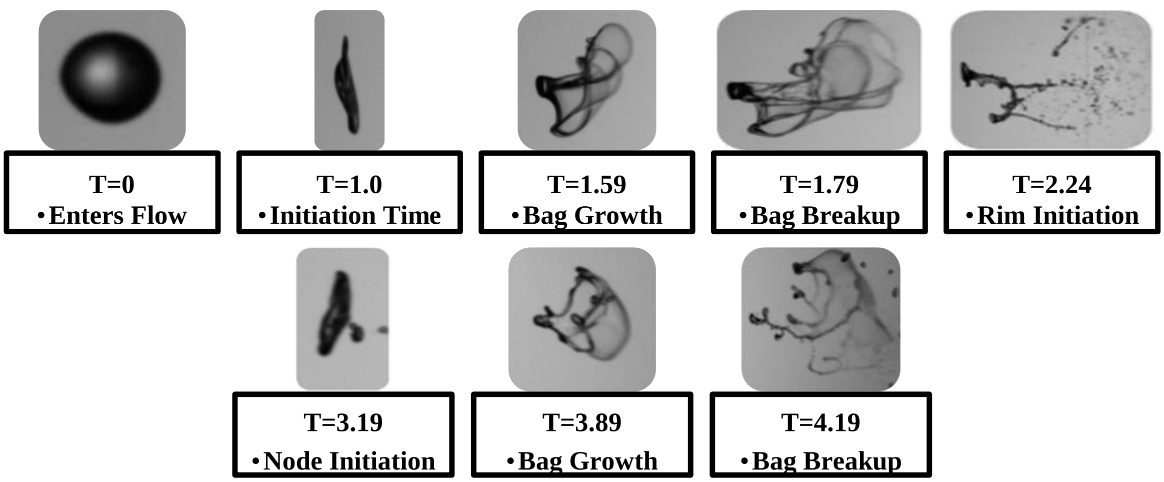


Figure 4.15. DI-Water n=1.00, K=0.00094) in the Dual Bag Breakup Regime, We=30. $d_0 = 2.74mm$; $L_{Bag} = 10.0mm$

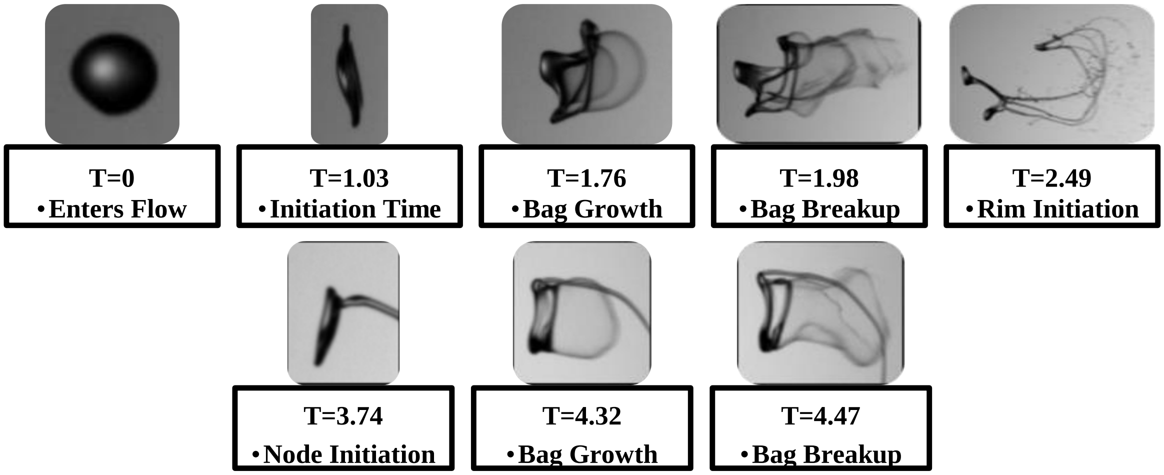


Figure 4.16. 0.8%CMC-7MF (*n*=0.93, *K*=0.046) in the Dual Bag Breakup Regime, *We*=35. $d_0 = 2.73mm$; $L_{Bag} = 11.3mm$

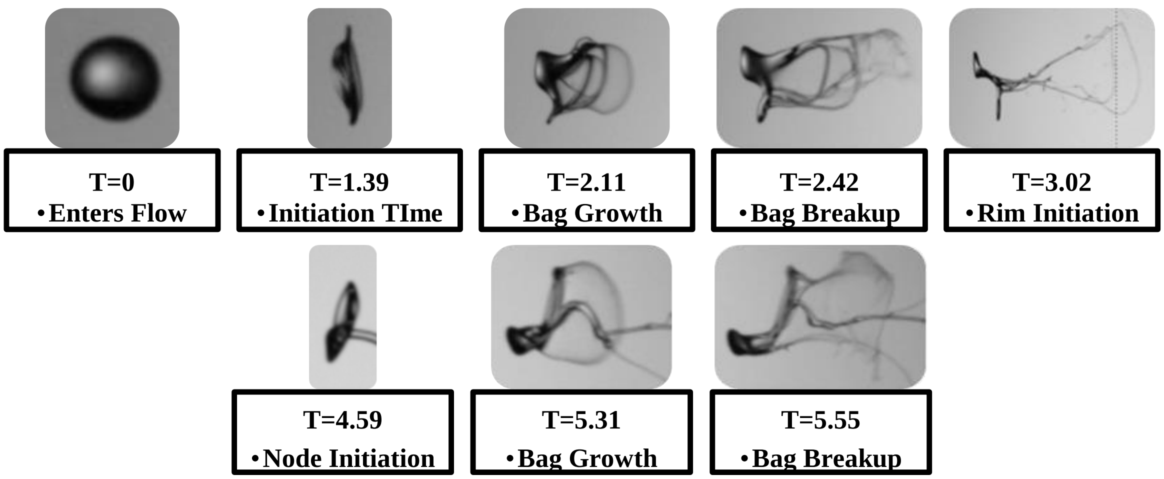


Figure 4.17. 1.4%CMC-7MF (*n*=0.87, *K*=0.20) in the Dual Bag Breakup Regime, *We*=44. $d_0 = 2.73mm$; $L_{Bag} = 11.7mm$

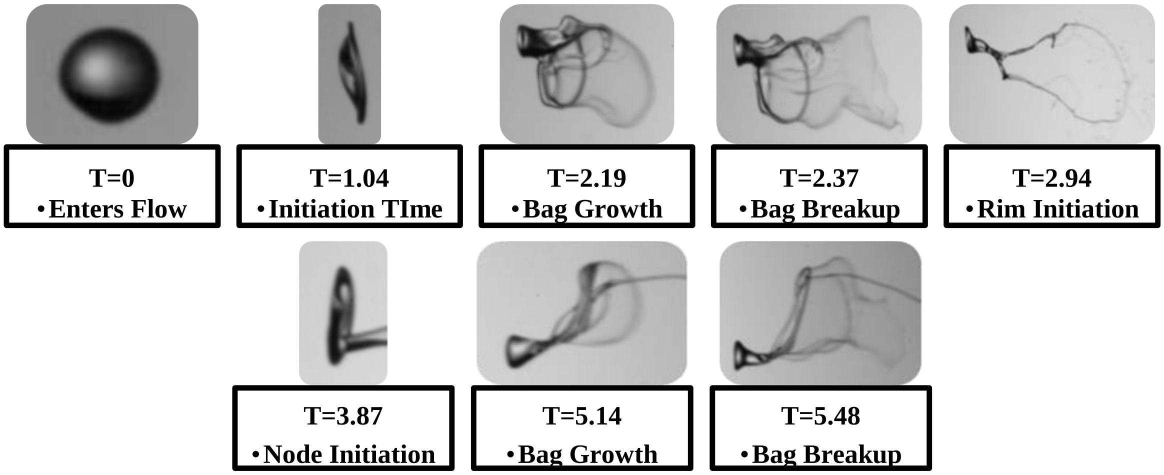


Figure 4.18. 0.5%CMC-7HF (*n*=0.71, *K*=0.37) in the Dual Bag Breakup Regime, *We* 39. $d_0 = 2.70mm\,; L_{Bag} = 10.2mm$

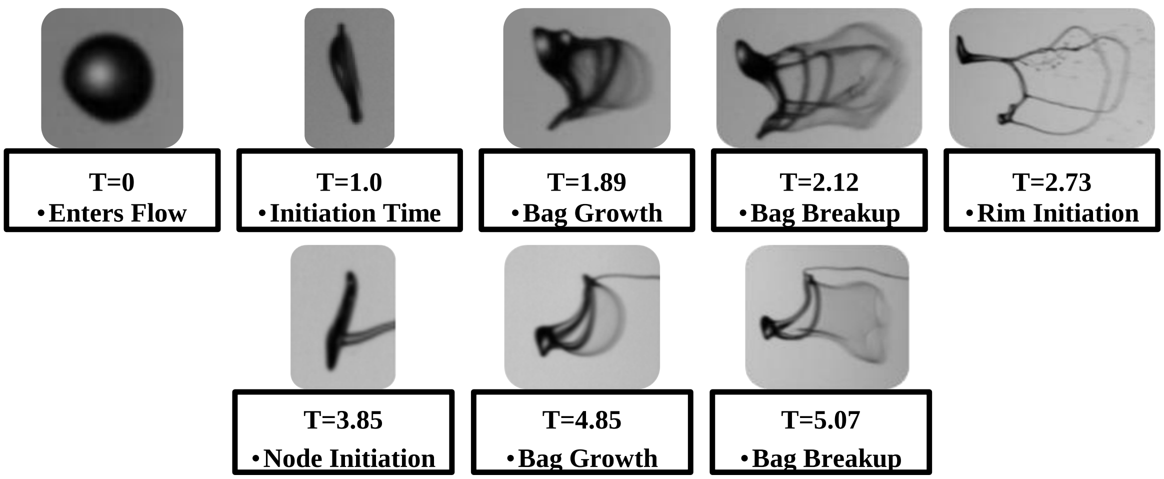


Figure 4.19. 74.96%Gly/0.06%CMC-7MF (*n*=0.90, *K*=0.13) in the Dual Bag Breakup Regime, *We*=41. $d_0 = 2.57mm\,; L_{Bag} = 7.78mm$

After bag-and-stamen growth, the bag ruptures with a large node forming from the stamen upstream. This node evolves into a disk-like shape similar to the bag breakup regime. At this point, additional bags begin to emerge—they are smaller due to the smaller node size. The bag then bursts and sequential breakup of the rim follows.

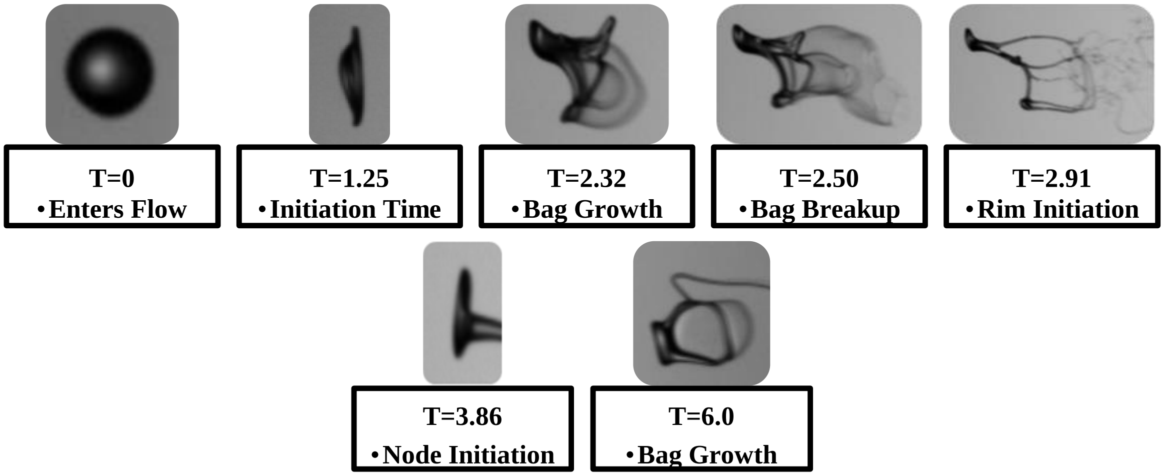


Figure 4.20. 79.95%Gly/0.06%CMC-7MF (*n*=0.89, *K*=0.20) in the Dual Bag Breakup Regime, *We*=43.$d_0 = 2.48mm$; $L_{Bag} = 7.58mm$

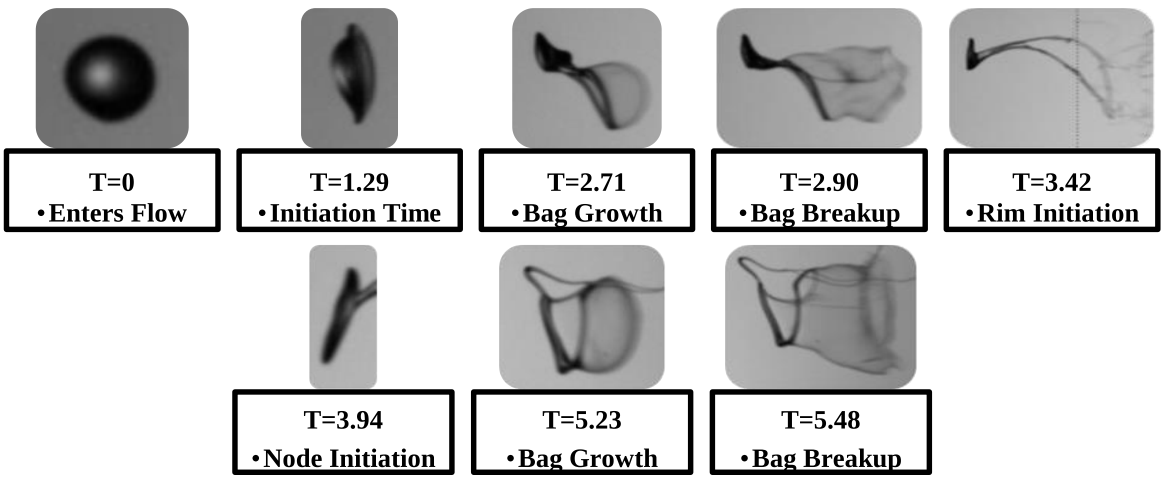


Figure 4.21. 84.96%Gly/0.05%CMC-7MF (*n*=0.88, *K*=0.30) in the Dual Bag Breakup Regime, *We*=55.

As with the previous regimes the non-Newtonian liquid break up produces marked differences. These liquids produce more ligaments than drops and have the same correlation to K and n as previously discussed in the former sections. As with the bag-and-stamen regime discussion, the ligaments attached to the rim/bag remains intact after the rim/bag has broken up. Also, for Newtonian liquids the rim and stamen typically breakup prior to the second bag growth. In contrast, for non-Newtonian liquids during the entire duration of node formation, deformation, 2nd bag growth, and bag bursting the stamen remains attached to the node and sequential 2nd rim produced. All the ligaments produced will indeed breakup by producing drops connected by ligaments, the attaching ligaments breaking apart, and the ligament segments contract onto the drops.

4.2 Regime Maps

Newtonian liquid secondary atomization literature has proven that breakup regimes can be arranged into a *We* versus *Oh* plot. There are however known discrepancies in determining definite standard values due to uncertainties, systemic errors, and the subjective nature of an experimentalist defining the *We* where a particular breakup occurs. Although reasonable approximations can still be made and the breakup mode desired can be obtained with the ease of calculating *We* and *Oh* using the initial conditions.

4.2.1 *We* versus *Oh*.

Difficulty has arisen when experimenters began using Non-Newtonian liquids (Wilcox *et al*., 1961). Viscosity becomes variable and means of determining the effective viscosity has posed a challenge. In this study, a simple model is proposed in order to allow for calculating the *Oh* and give comparable results to what has been the long standing trend of *We* versus *Oh* regime maps.

In Figure 4.22 below, the *We* and *Oh* data corresponding to the end of the bag breakup regime, beginning and end of bag-and-stamen regime, and the beginning of the dual bag breakup regime is plotted. The colors represent the type of liquid and the shapes denote the breakup mode. The *We* was calculated as is traditionally done. However, since these liquids are non-Newtonian they have a varying viscosity with strain rate. Thus, the *Oh* would also vary. The strain rate was approximated using the average strain rate up to the initiation time. Typically, it is assumed that the breakup regimes considered in this study involve a spherical drop deforming to an ellipsoidal shape (defining the initiation time). Other authors such as Zhao *et al*. (2010) have postulated that it is up to this initiation time, which will determine the breakup mechanism which will occur. Thus, the average strain rate up to this initiation time was chosen and is calculated using,

$$\dot{\gamma}_{Avg} \approx \dot{\varepsilon}_{Avg} = \frac{d_{cro,ini}-d_0}{d_0 t_{ini}} = \left(\frac{d_{cro,ini}}{d_0} - 1\right)\frac{1}{t_{ini}} = \left(\frac{d_{cro,ini}}{d_0} - 1\right)\frac{u_{rel}}{T_{ini} d_0}\sqrt{\frac{\rho_G}{\rho_L}} \tag{4.1}$$

This form is valuable since many authors in secondary atomization consider the non-dimensional cross-stream diameter $\left(\frac{d_{cro,ini}}{d_0}\right)$ and non-dimensional initiation time (T_{ini}). The shortcoming of this relation is that it does not use initial conditions for determining the strain rate and requires some experimental data. With strain rate approximated, it is put into the power law model to give,

$$\mu_{eff} = K(\dot{\gamma})^{n-1} = K\left(\left(\frac{d_{cro,ini}}{d_0} - 1\right)\frac{u_{rel}}{T_{ini}d_0}\sqrt{\frac{\rho_G}{\rho_L}}\right)^{n-1} \tag{4.2}$$

The strain rates were between $(127 - 356)s^{-1}$, within the power law model's range of applicability and less than the maximum strain rates tested. *Oh* is then,

$$Oh = \frac{\mu}{\sqrt{\rho_L \sigma d_0}} = \frac{K}{\sqrt{\rho_L \sigma d_0}}\left(\left(\frac{d_{cro,ini}}{d_0} - 1\right)\frac{u_{rel}}{T_{ini}d_0}\sqrt{\frac{\rho_G}{\rho_L}}\right)^{n-1} \tag{4.3}$$

was plotted using this relationship for *Oh*. It is obvious that the method produced the expected trend between *We* and *Oh*. When the *Oh* is increased the *We* required to initiate a particular breakup regime becomes greater. In other words, as the viscosity increases the liquid resists deformation (implied by an increase in *Oh*). With higher resistance, the airflow field must overcome the viscous resistance and the restorative surface tension forces, whose forces tend in the direction of keeping a drop spherical (implied in the We). Therefore, as *Oh* increases the *We* required to produce the same deformation also increases. This trend has been proven accurate throughout secondary atomization Newtonian liquid literature and further supported here. Besides that, the points plotted correspond to approximately the same curve as provided by Zhao *et al*. (2013) where only Newtonian liquids were used. This curve from literature did not have uncertainty values provided however by inspection of the data points in Zhao *et al*. (2013)'s study it is plausible to assume the data points in the figure below are within experimental uncertainties. Since systematic uncertainties within a system usually remain constant and vary from one system to another, a definite method to prove that approximating strain rate using the average strain rate is to develop a Newtonian liquid *We* versus *Oh* regime map.

Then, with the same system test the Non-Newtonian liquid breakup and compare the results using the above relation.

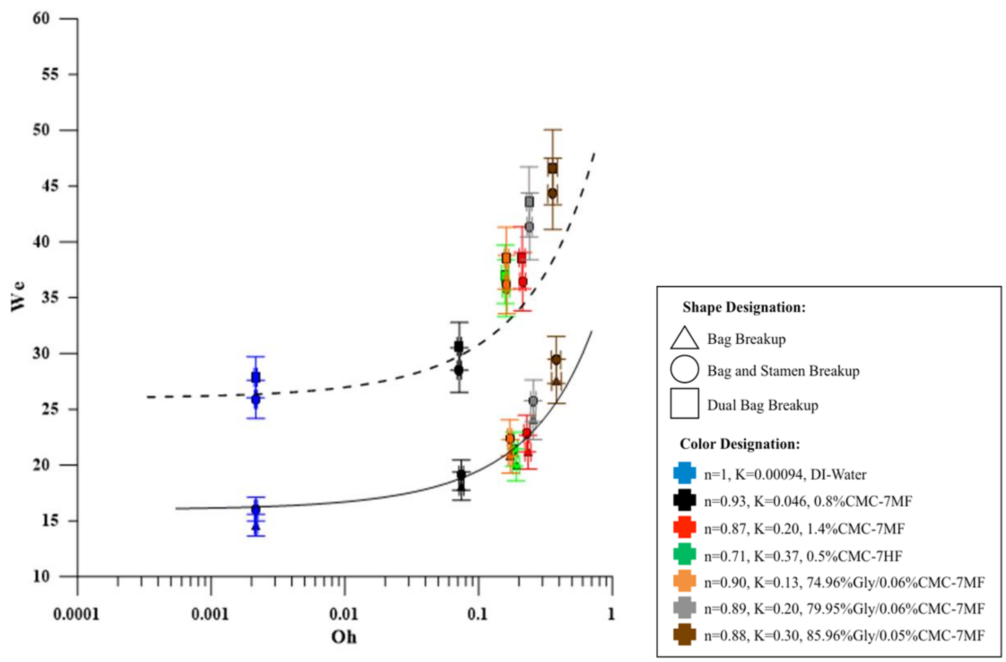


Figure 4.22. *We* versus *Oh* Regime Map with curve provided by Zhao *et al*. (2013).

4.2.2 Liquid *Re* versus *We*

Experimentalists such as Snyder (2011), Lopez (2010), and Theofanous *et al*. (2013) have attempted to find a means for developing a new method using non-dimensional parameters which allows for comparison between Newtonian and non-Newtonian liquids and also provides insight into the physical nature of the problem. Here, using the above relationship for viscosity a new physically meaningful correlation was found.

In Figure 4.23, the *We* and *Re* values where bag breakup ends, bag-and-stamen begins and ends, and dual bag breakup starts are plotted. *Re* is,

$$Re_L = \frac{\rho_G d_0 u_{rel}}{\mu_{eff}} = \frac{\rho_G d_0 u_{rel}}{K}\left(\left(\frac{d_{cro,ini}}{d_0} - 1\right)\frac{u_{rel}}{T_{ini} d_0}\sqrt{\frac{\rho_G}{\rho_L}}\right)^{1-n} \tag{4.4}$$

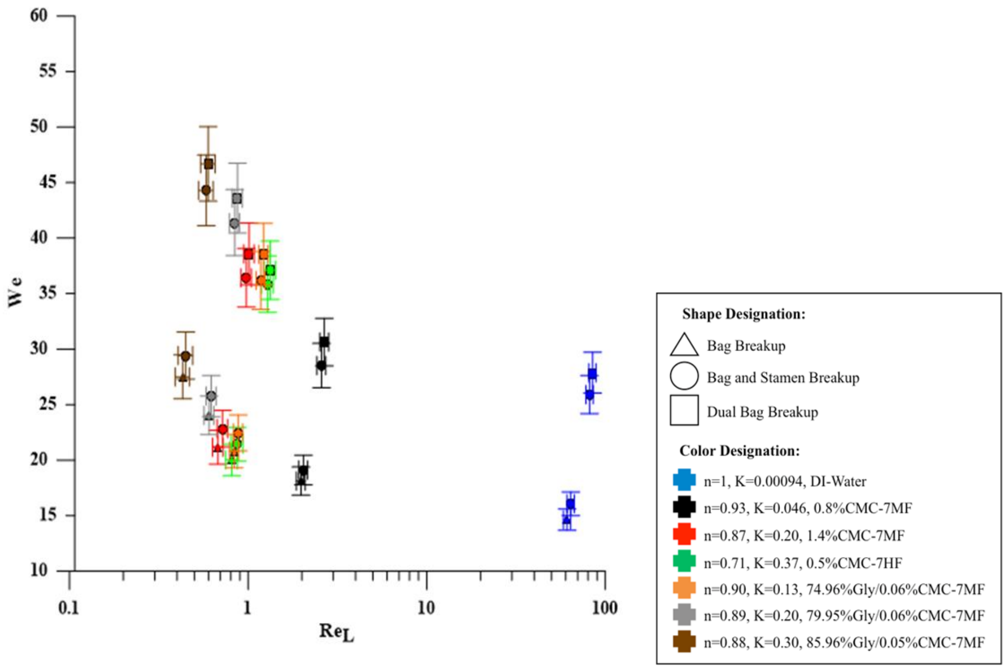


Figure 4.23. *We* versus *Re*.

It is apparent from Figure 4.23 there exists a correlation with *We*. As the Liquid *Re* number increases, the *We* required to initiate a particular breakup decreases. The physical interpretation of the plot is quite revealing. On either axis is a breakup promoting factor in the numerator of the term, inertia force. However, in the denominator for the x-axis is viscous resistive forces while the y-axis denominator has restorative surface tension forces. As the Liquid Reynolds increases, this implies that inertial forces are sufficient to overcome the viscous forces and so more inertial forces are allocated to overcoming surface tension forces. The result is a smaller required inertial force (smaller We) to produce the same type of breakup. Conversely, if the Liquid *Re* decreases, then inertial forces become smaller or comparable to viscous forces so less can be allocated to overcome surface tension forces. Therefore, larger inertial forces are required (higher We) to overcome the larger viscous and the surface tension forces. Additionally, the surface tension, drop diameter, and drop density was varied at most 12%, 10%, and 22%, respectively and the continuous phase used was only incompressible air. Thus, it is still not certain whether this relationship will hold for all cases.

4.2.3 Number of Rayleigh Taylor Waves (N_{RT}) versus *We*

Besides *Re*, another possible regime map comes from Zhao *et al*. (2010). They argue Rayleigh-Taylor instabilities are responsible for secondary breakup, and note that the most unstable wave which causes breakup is,

$$\lambda_{max} = \lambda_c\sqrt{3} = 2\pi\sqrt{\frac{3\sigma}{\rho_L a_L}} \tag{4.5}$$

Furthermore, the number of waves which fit onto the windward side of the drop at initiation time is called the Rayleigh-Taylor wave number (N_{RT}) and is expressed by,

$$N_{RT} = \frac{D_{cro,ini}}{\lambda_{max}} = \frac{D_{cro,ini}}{\lambda_c\sqrt{3}} = \frac{D_{cro,ini}}{2\pi}\sqrt{\frac{\rho_L a_L}{3\sigma}} \tag{4.6}$$

Zhao *et al*. (2010) conjectured that the number of waves which fit onto the windward side of the drop at initiation time is the determining factor for the breakup mode that will commence. Refer to Table 4.1.

Table 4.1. Number of waves on the drop at initiation time and the resulting breakup mode.

Rayleigh Wave Number Range (N_{RT})	**Breakup Type**
$N_{RT} < \frac{1}{\sqrt{3}}$	Vibrational
$\frac{1}{\sqrt{3}} < N_{RT} < 1$	Bag
$1 < N_{RT} < 2$	Bag-and-stamen
$2 < N_{RT} < 3$	Bag, Stamen, and Bag
$N_{RT} > 3$	Sheet Thinning

In Figure 4.24, N_{RT} is plotted versus *We* where each data point represents a liquid and of breakup type. Zhao *et al*. (2010) used very low viscosity Newtonian liquids (maximum of $\mu = 0.0013\ Pa \cdot s$) in the developing their model, but demonstrates it applies remarkably well even for liquids having two orders of magnitude larger viscosity (maximum of $\mu = 0.171\ Pa \cdot s$). It is therefore concluded that the Zhao *et al*. (2010) relationship is a valid alternative to the classical *We* versus *Oh* breakup regime map for inelastic non-Newtonian liquids. There is one obstacle when using this model—it does not use initial conditions. Instead, it requires acceleration and the cross-stream diameter at initiation time.

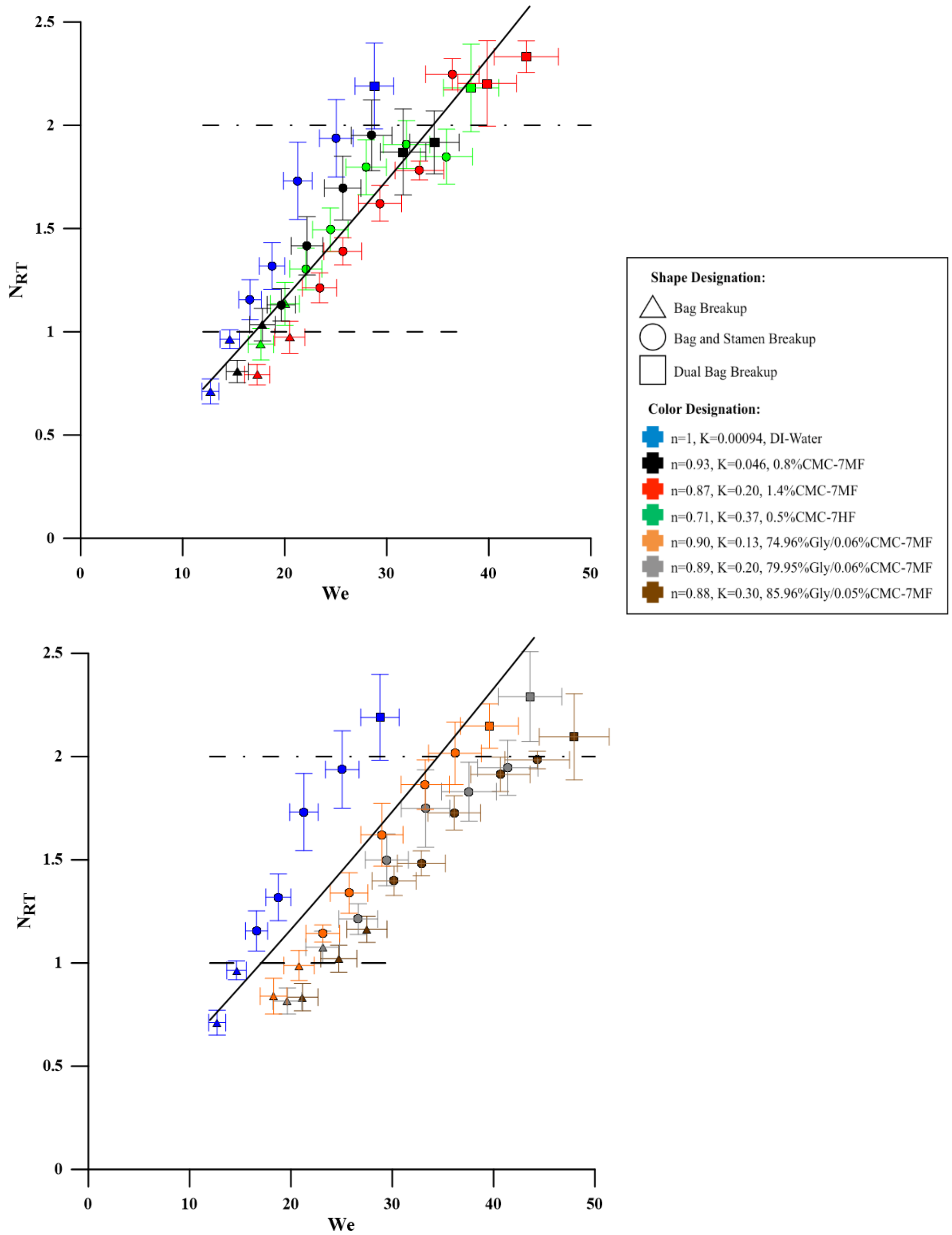


Figure 4.24. Number of Rayleigh-Taylor waves, N_{RT}, versus *We* with correlation from Zhao *et al*. (2010).

4.3 Breakup Times

Breakup times are typically reported in non-dimensional form. The most common relation is,

$$T = \frac{t u_{rel}}{d_0} \sqrt{\frac{\rho_G}{\rho_L}} \tag{4.7}$$

where t is the measured time, d_0 the initial drop diameter, u_{rel} the initial drop relative velocity, ρ_G and ρ_L the gas and liquid density, respectively. While Pilch and Erdman (1987) showed that T is not constant at subsonic velocities, the expression is used abundantly in the literature and was adopted here for comparative purposes.

4.2.4 Initiation Time

The initiation time in the bag regime is defined as the time when the bag begins to form (Pilch and Erdman, 1987). Flock *et al*. (2012) suggested to use the instant when the axial extent on the drop is minimal. The latter choice was adopted here.

Initiation time is measured by first determining the time at which the drop enters the flow. This is taken to be the frame prior to when the ratio of the cross-stream radius and shortest radius from the centroid has increased by 10%. A value of 10% was chosen as it is double that of the largest fluctuation observed in the videos.

From Figure 4.25, it's clear there is qualitative agreement between experimental data presented in this thesis and predictions using the relationship of Pilch and Erdman (1987). Some deviation is expected since Pilch and Erdman (1987) used *We* and *Oh* ranges much larger than were used during this study: $10 \leq We \leq 10^6$ and $0.1 \leq Oh \leq 1.15$ versus $12 \leq We \leq 49$ and $0.002 \leq Oh \leq 0.3$. In addition, DI-water and the 0.8%CMC-7MF solution are outside the Pilch and Erdman range.

Figure 4.25 also shows that as K increases the initiation time increases for any particular *We*. However, if n decreases sufficiently (as with the 0.5%CMC-7HF) the effects of increasing K are counteracted and initiation time drops. Lopez (2010) used similar non-Newtonian liquids and observed the same qualitative behavior.

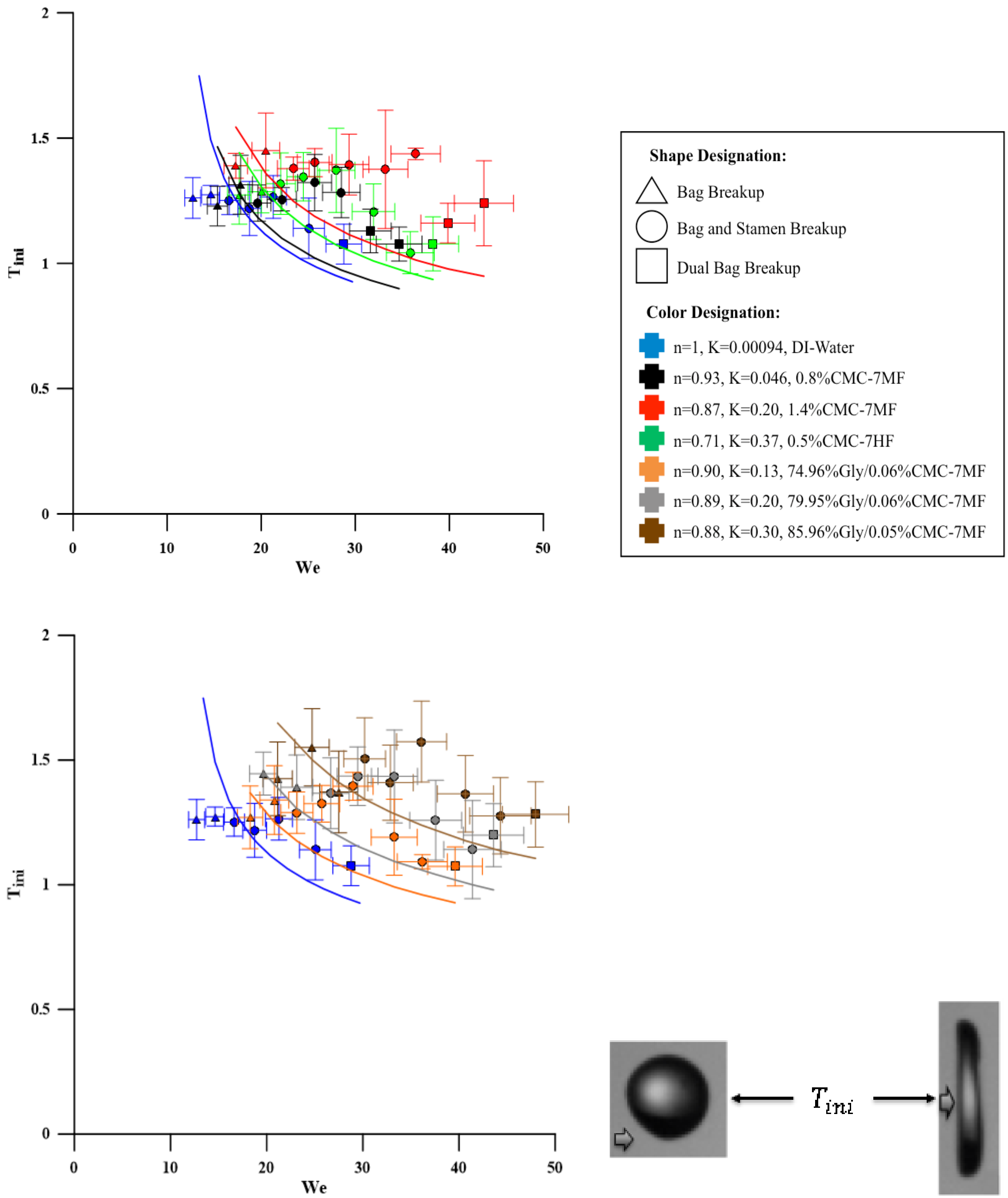


Figure 4.25. Non-Dimensional Initiation Time, T_{ini}, versus *We* with correlations from Pilch and Erdman (1987).

4.2.5 Bag Breakup Time

Bag breakup time is the instant when the bag bursts. It is valuable because it is the time when the smallest fragments are formed, so models for bag breakup cease to apply and additional Lagrangian objects must be tracked. Bag breakup time was measured using a pop up movie and pop up dialogue box for user determination, with subsequent calculations completed using MATLAB.

Figure 4.26 shows bag breakup time is a function of *We* and liquid rheology. As expected, bag breakup time decreases monotonically with increasing *We*. Also as expected, increasing *K* increases bag breakup time while as *n* decreases bag breakup time decreases. The increased bag breakup time with increasing effective viscosity is a compound bags, which require more time to grow. The same behavior was observed by Lopez (2010) using non-Newtonian liquids, and by Dai and Faeth (2001) using Newtonian liquids.

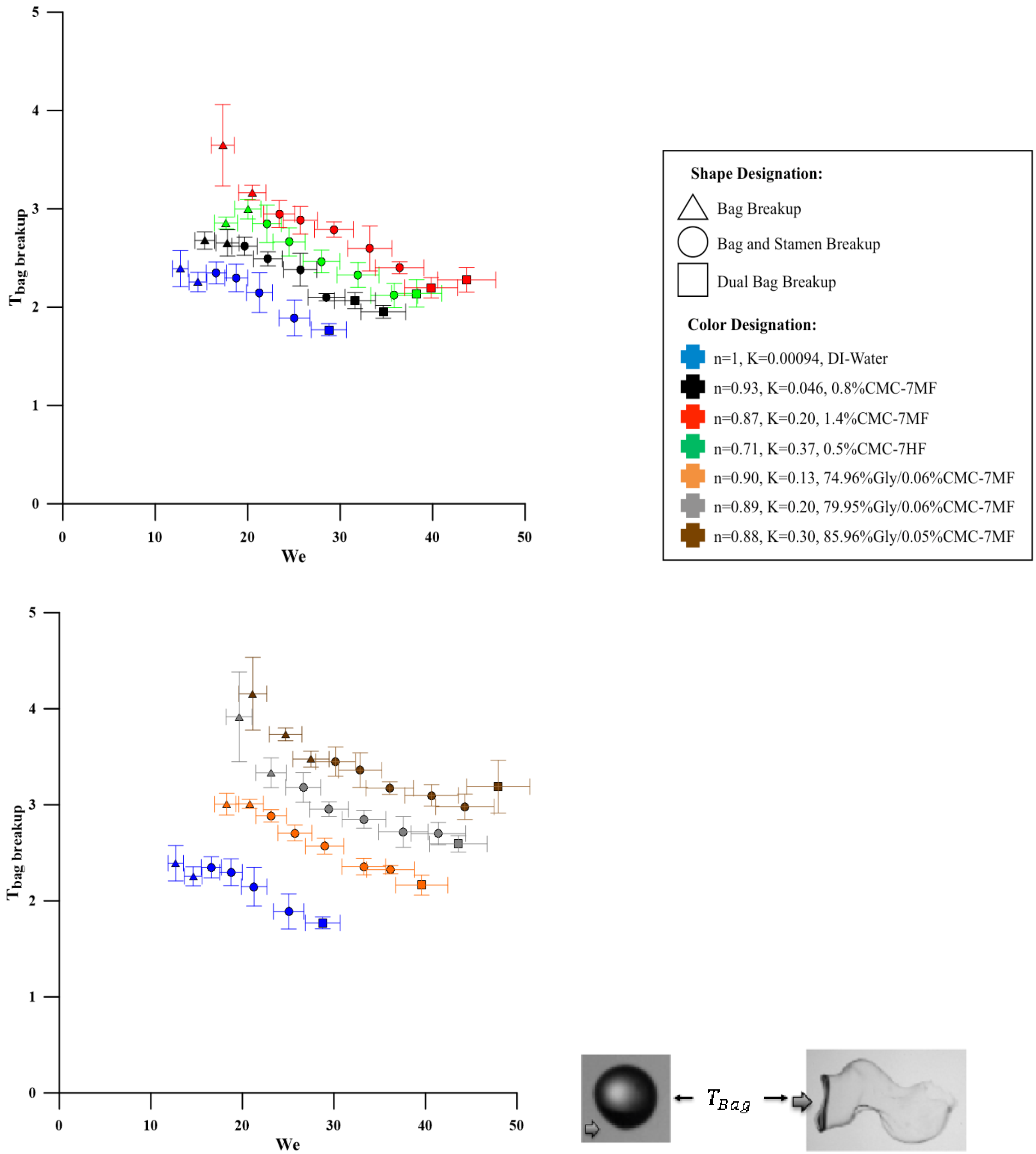


Figure 4.26. Non-Dimensional Initiation Time, T_{Bag}, versus *We*

4.3 Physical Correlations

4.3.1 Cross Stream Diameter at Initiation Time

Many authors (Chou and Faeth, 1998; Gel'fand *et al*., 1974; Zhao *et al*., 2013; Hsiang and Faeth, 1992; Zhao *et al*., 2010); Dai and Faeth, 2001; and Hsiang and Faeth, 1995) have reported the cross-stream diameter at breakup because it is easy to measure and is relevant to drop motion (it is needed for validation of drop spherical-to-ellipsoidal shape sub-models). See Figure 4.27.

All three correlations presented in Fig. 4.27 were developed using $Oh<0.1$. They are therefore expected to best agree with DI-water data.

In all cases, increasing *We* first increases non-dimensional cross-stream diameter and then decreases it, a trend also reported by Zhao *et al*. (2010). The *We* where the decrease occurs is within the bag-and-stamen regime. The reason for the maximum is illustrated in Figures 4.8 through 4.14—once the stamen forms less mass is available for the bag.

Increasing *K* has two effects on the peak cross-stream diameter curve. First, the peak is shifted to higher *We*, where that bag-and-stamen morphology occurs for that liquid. Second, deformation is less so the entire curve shifts to lower *We*. In contrast, if *n* decreases the peak cross-stream diameter shifts to lower *We* with the entire curve shifting upward. A similar trend was noted by Hsiang and Faeth (1992), and by Lopez (2010).

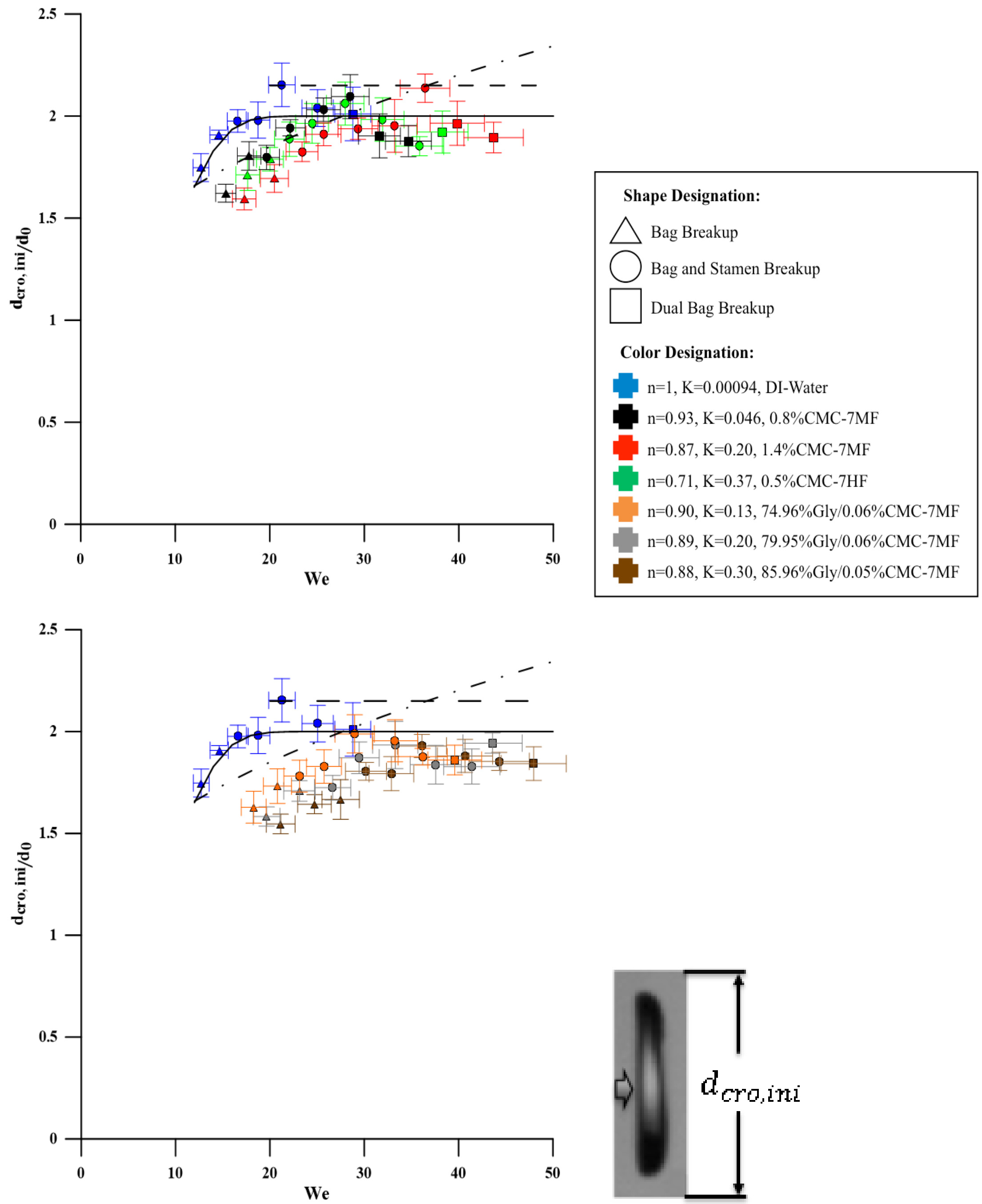


Figure 4.27. Non Dimensional Cross Stream Diameter versus *We*. Curve fits from: - - - - - (Z. Dai and Faeth, 2001); -·-·-·- (Hsiang and Faeth,1992); —— (Hui Zhao, *et al*., 2010).

4.3.2 Rim Diameter When Bag Breaks

Secondary atomization studies have also provided the cross-stream diameter at bag breakup time (Chou and Faeth, 1998; Dai and Faeth, 2001), reported as a function of *We* and time. This dimension is useful for modelers predicting drop deformation with time because it serves as a validation.

As Figure 4.28 shows, both *We* and μ_{eff} control rim diameter at bag breakup. Rim diameter increases to a maximum with increasing *We* in the bag-and-stamen regime, then falls again. In addition, as *We* increases the rim diameter becomes until the bag-and-stamen regime is reached, then decreases. The increase is due to increased static pressure/aerodynamic forces within the rim/bag forcing the rim outward. The subsequent decrease in rim diameter is due to the change in morphology, where the bag and rim become smaller because some mass is allocated to the stamen (Dai and Faeth, 2001). The size of the rim diameter at bag breakup for water with We=19 here ($d_{rim}/d_0 = 4.16$) is similar to the value found by Chou and Faeth (1998) for water at *We*=20($d_{rim}/d_0 \approx 4.00$).

Figure 4.28 also shows the influence of effective viscosity on breakup dynamics. Careful consideration shows that if *K* increases (with all other liquid properties remaining equal) the rim diameter at bag breakup decreases regardless of *We*. In contrast, if *n* decreases the rim diameter at bag breakup curve increases regardless of *We*. Finally, if surface tension is decreased the rim diameter will be larger at bag breakup.

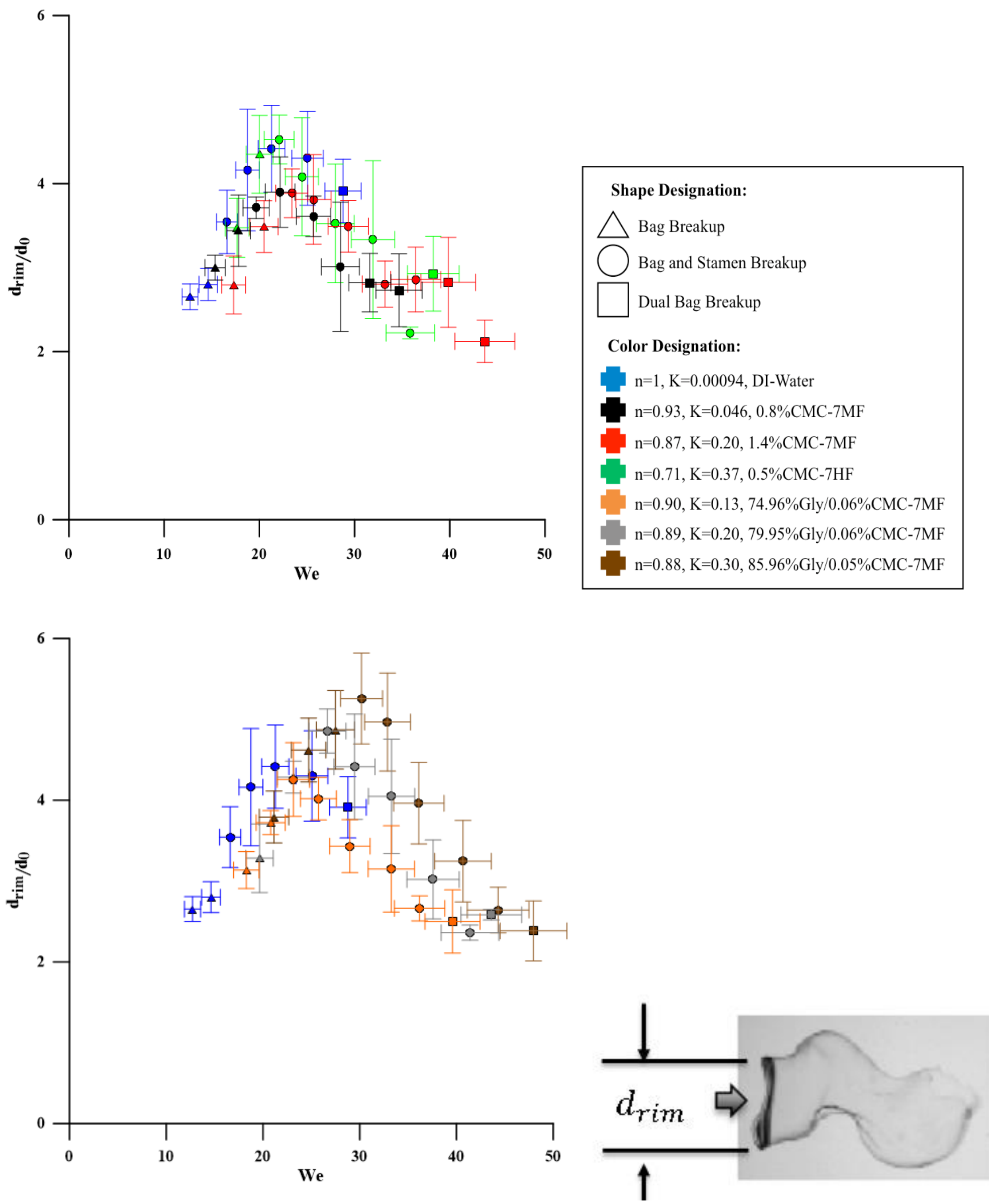


Figure 4.28. The Non-Dimensional Rim Diameter When the Bag Breaks.

4.3.3 Bag Length When Bag Breaks

Snyder (2015) and Lopez (2010) have reported that non-Newtonian bag growth is significantly larger than that for Newtonian liquids. Their comparisons were not quantitative. Kulkarmi (2013) did perform quantitative comparisons for Newtonian liquids, which will also be done here. This experimental data will also be valuable for model validation.

Figure 4.29 indicates that increasing *We* in the bag regime causes the bag length at breakup to first rise, then peak when the bag-and-stamen regime starts. Bag length decreases as *We* rises further, with breakup length a stronger function of *K* than *n*. As Figure 4.29 shows, increasing *K* shifts both the entire curve and peak value to higher *We*. In contrast, decreasing *n* will shift the entire curve to lower *We*, but not decrease the peak magnitude. Comparing the average DI-water bag length from this work at $We = 13$ ($^{L_{Bag}}/_{d_0} = 3.9$) with that of Kulkarmi (2013) ($^{L_{Bag}}/_{d_0} \approx 4.5$) shows agreement to within experimental uncertainties.

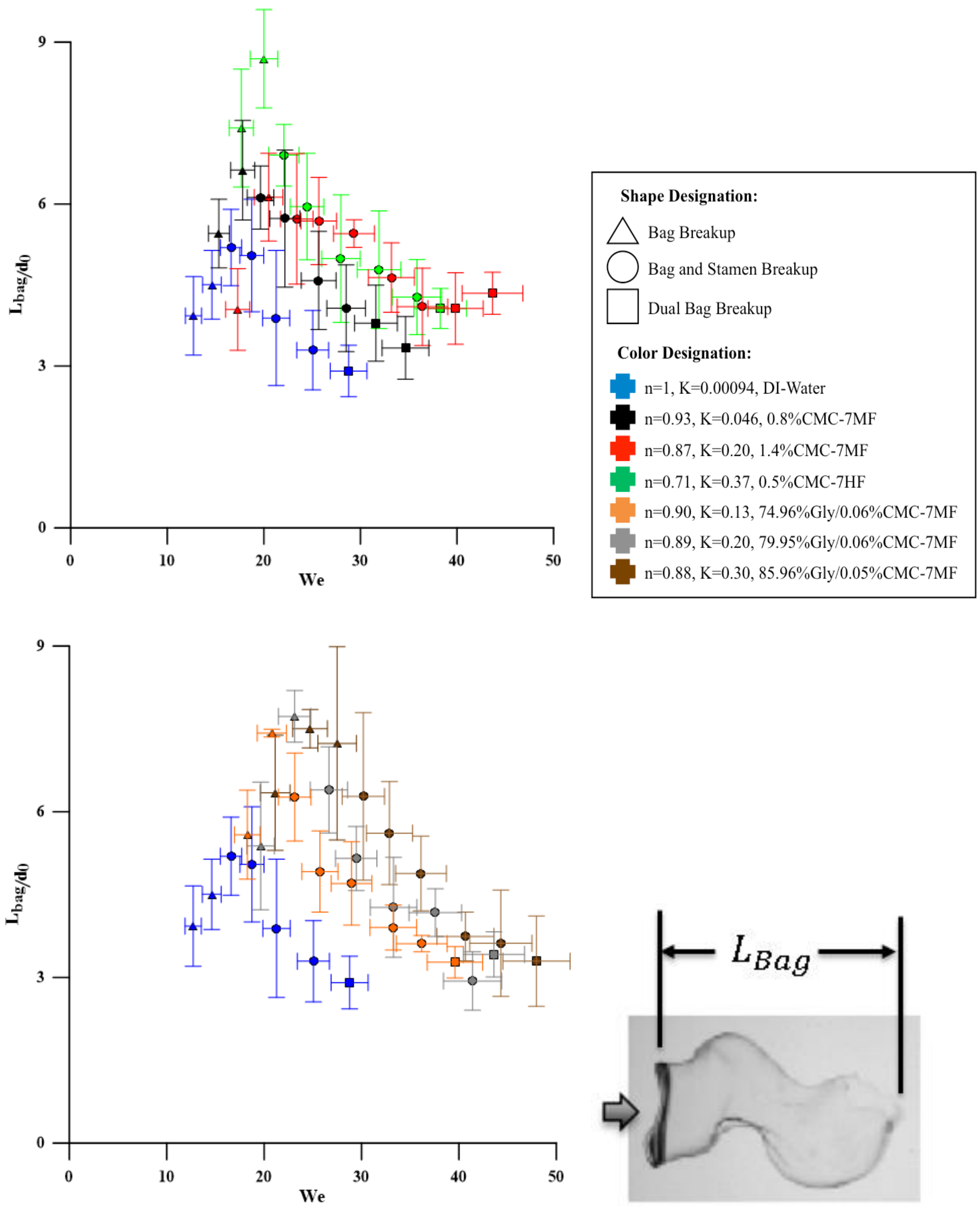


Figure 4.29. Non-Dimensional Bag Length When Bag Breaks versus *We*.

4.4 Drop Dynamics

This section includes displacement, velocity, and the coefficient of drag. In secondary atomization there are a number of opinions as to how drop displacement should be measured on the drop. These issues are thoroughly discussed by Pilch and Erdman (1987). In this thesis the drop position is reported in terms of one of its centroids. Justification for each choice is provided when it is introduced.

4.4.1 Drop Displacement at Initiation Time

The first drop quantity to be discussed is drop displacement. In order to ensure that the mass centroid is being accurately tracked, the MATLAB code takes the area centroid (using image moments) before the bag forms. When the bag emerges the centroid is computed for the rim. Justification for this second choice comes from Arcoumanis *et al*. (1994), Lane (1951), and Gel'fand *et al*. (1974), all of whom have estimated that the ring volume is 70% of the original drop volume. In a more recent study, Zhao *et al*. (2011) found that the ring volume is 85% of the original drop volume.

Figure 4.30 shows that the displacement at initiation time is similar at lower *We*, regardless of liquid. As *We* is increased, drop displacement at initiation time for all liquids increases to a peak in their corresponding bag-and-stamen regimes, at which point the displacement drops as *We* is increased further. If *K* is increased, the magnitude of the peak increases and shifts to larger *We*. In contrast, decreasing *n* shifts the peak back to lower *We*, but has no other effect. The trend with *K* seems to correlate with what occurs at the initiation time. If *K* is larger the liquid has a larger viscous force so the deformation rate is reduced and the drop takes longer to reach the initiation time and is displaced further.

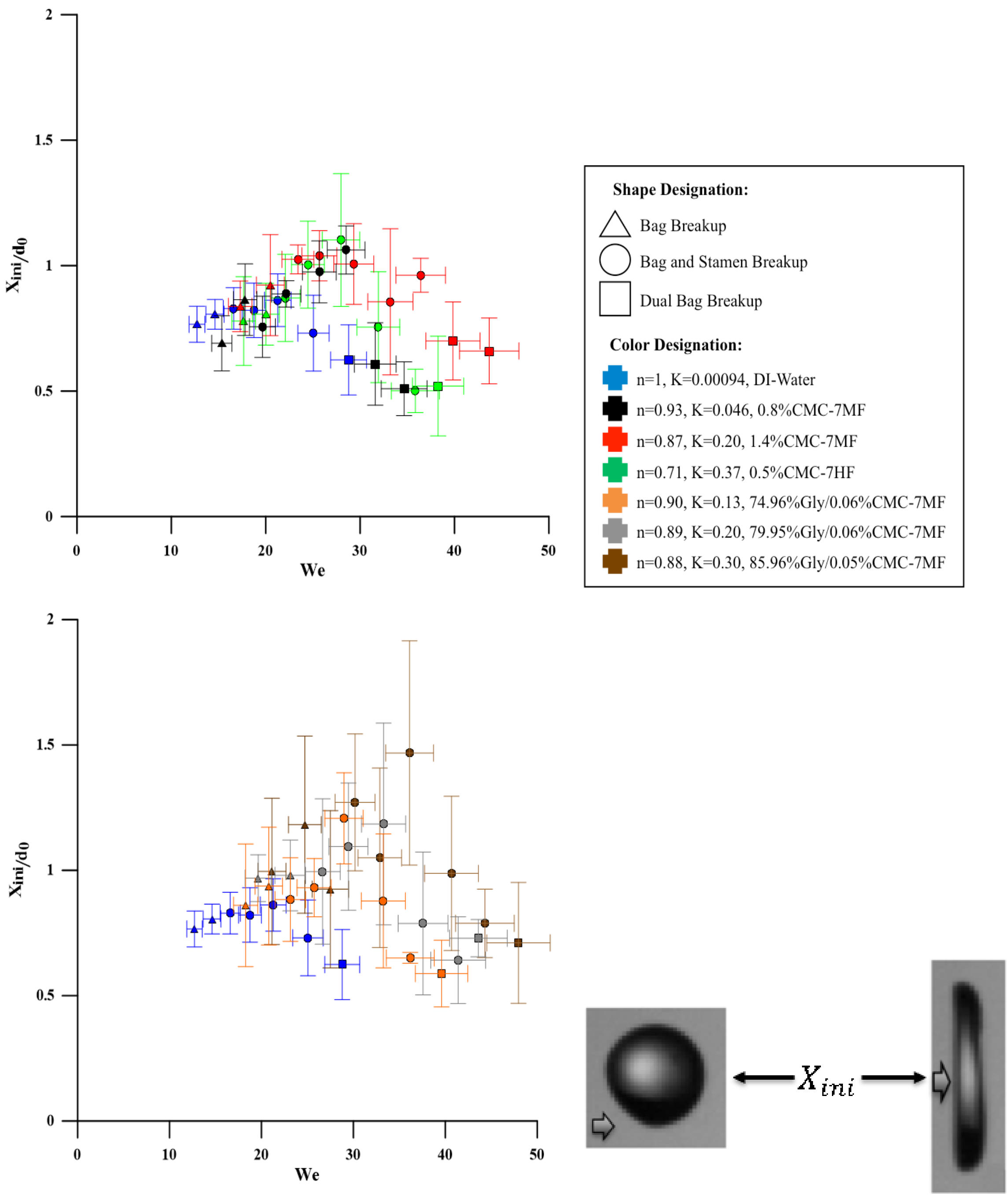


Figure 4.30. Non-Dimensional Displacement at Initiation Time versus *We*.

4.4.2 Drop Displacement When Bag Breaks

A few authors (Chou and Faeth, 1998; Flock *et al*., 2012; Simpkins and Bales, 1972) have studied how drop displacement varies with time. However, in this study the focus was on drop displacement when the bag breaks, and how it depends on *We* and liquid. See Figure 4.31.

The displacement reported in Figure 4.31 is that of the rim, for reasons listed above. Note that as *We* is increased, rim displacement at bag breakup first increases and then decreases when viscosity is low (low *K* or *n*). If *K* is larger, the displacement at bag break up only decreases with increasing *We*. This behavior resembles that of bag breakup time.

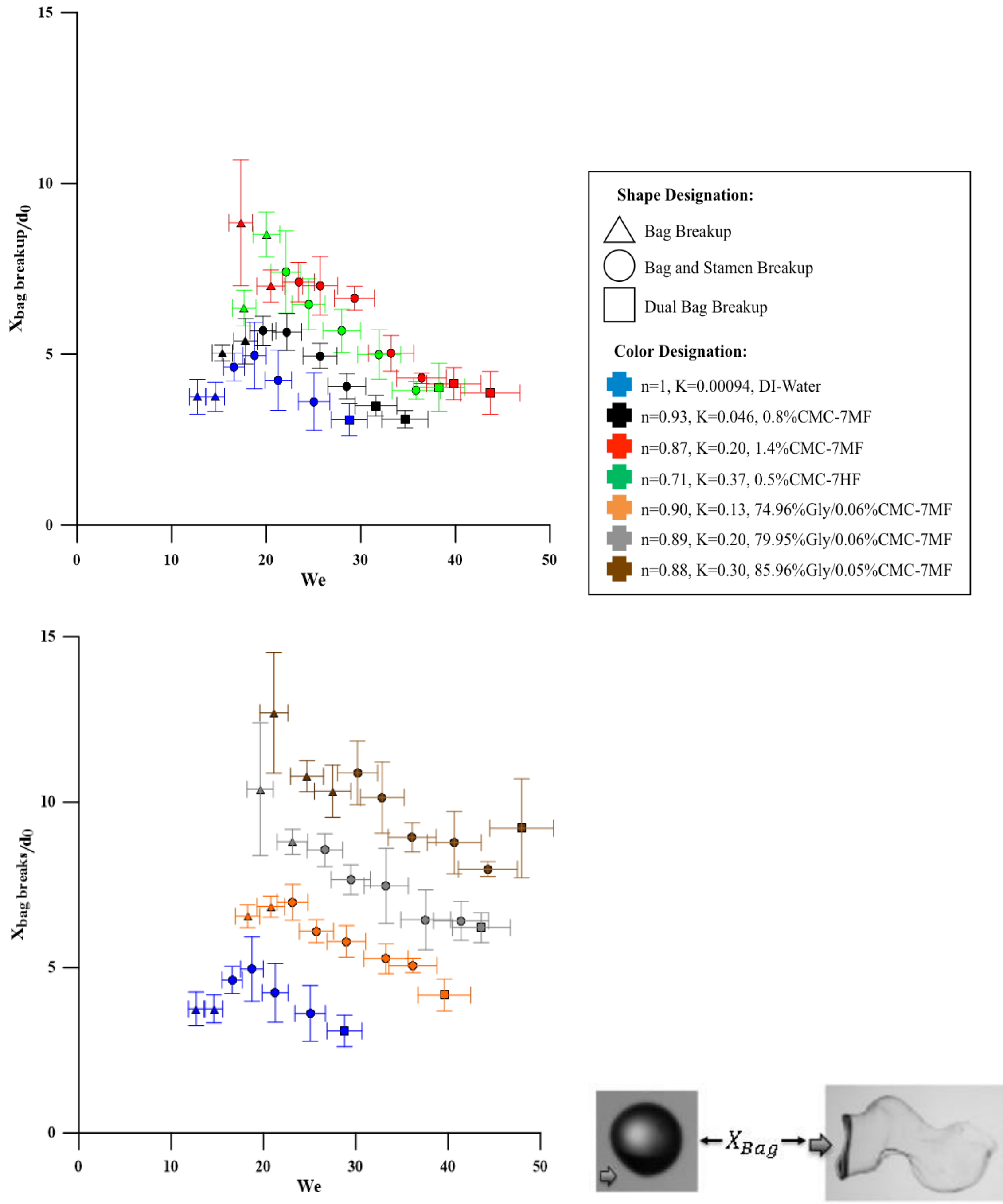


Figure 4.31. Non-Dimensional Drop Displacement When the Bag Breaks versus *We*.

4.4.3 Drop Velocity at Initiation Time

Flock *et al*. (2012) and Dai and Faeth (2001) have studied the drop velocity as a function of time. Zhao *et al*. (2010) reported drop velocity at initiation time. In this thesis, drop velocity at several times is reported; it was obtained via the MATLAB code's central differencing with second order error algorithm. Either a 0.209 or 0.152 ms inter-frame spacing was used.

Figure 4.32 shows that drop velocity at initiation time initially increases as *We* is increased, except for the DI-water case, peaks, and then falls. The peak occurs within the bag-and-stamen regime for each liquid so shifts toward higher *We* when *K* is increased. The peak also increases in magnitude.

Figure 4.32 also shows that the velocity at initiation time decreases for all *We* when *n* is decreased.

At low *We*, the velocity magnitudes at initiation time are approximately equal for all liquids. This may be attributed to the longer initiation times for large *K* and *n*, which result in smaller drop acceleration. At larger *We* the initiation time has a more pronounced effect with larger consistency index liquids having a noticeably longer initiation time. Therefore, those liquids overcome the smaller acceleration and have a greater velocity at initiation time. A similar argument can be made if *n* is decreased while *K* remains large (0.5%CMC-7HF), since this liquid has the largest *K*.

Zhao *et al*. (2010) approximated the drop velocity at initiation time to be approximately 10% of the gas phase velocity. This is consistent with values measured as part of this thesis research (4%-8%).

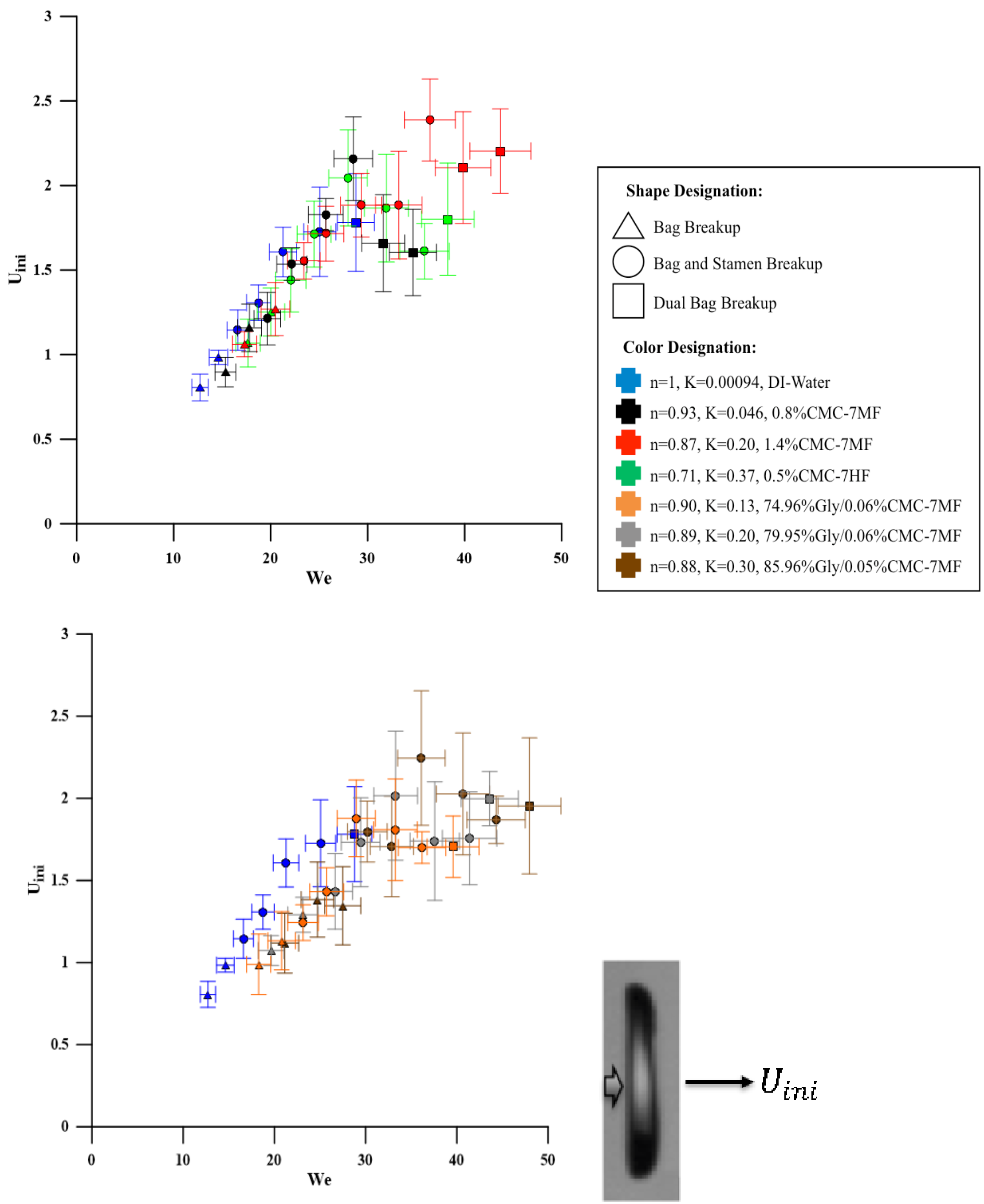


Figure 4.32. Drop Velocity (m/s) at Initiation Time versus *We*.

4.4.4 Drop Velocity When Bag Breaks

Drop velocity when the bag breaks up is computed using a different centroid than the initiation time centroid for the following reason. At initiation time the area centroid is an accurate measurement of the drop position. However, when the bag bursts the centroid is more accurately obtained at the rim axis since it has been estimated in the literature to be 70 to 85% of the drop mass. In addition, when the stamen begins to emerge its position varies between the downstream end of the bag and upstream of the rim. The choice of mass centroid was made based on stamen position. However, as can be seen in the visualizations (see Figure 4.8 through 4.14), the approach using the rim axis still provides a reasonable approximation within the bag-and-stamen regime because the stamen is still small. In contrast, once the drop enters the dual bag regime a larger fraction of the drop volume is concentrated into the stamen as it begins to protrude significantly further from the rim. The rim and bag have correspondingly reduced masses. Thus, in the dual bag regime it may be more accurate to place the centroid on the stamen itself while neglecting the rim and bag completely. This behavior is thoroughly discussed by Dai and Faeth (2001).

In Figure 4.33, as *We* is increased all liquids, except water, exhibit an increase, a peak, and then a decrease in drop velocity. The position of the peak is not dependent on *n* or *K*, although it does occur in the bag or bag-and-stamen breakup regimes. The peak magnitudes do depend on *K*, with peak magnitude increasing with an increase in *K* and exhibiting no dependence on *n*. Water is the exception, exhibiting only has an increase with *We*.

The velocity at bag breakup depends on the history of the drop. For example, the glycerin solutions have the longest bag breakup times, longest bags, and largest rim diameters at bag breakup. Therefore, the glycerin solutions have more time to accelerate and, with a larger rim diameter will allow more air to enter the bag and transfer its momentum to the deforming drop. Additionally, there must be a relationship between U_{ini} and bag length because the 0.5%CMC-7HF peak corresponds to the same peak in the bag length curve. Indeed, an increased bag length must be the result of a larger pressure differential and resistance to rupture so an increased velocity may be expected. To further support this explanation, the rim diameter for the 0.5%CMC-7HF liquid is similar to that for water and the bag breakup time is less than for the 1.4%CMC-7MF case, yet the velocity of the 0.5%CMC at initiation is much larger than either. The only explanation must be influence of another physical dimension, which is bag length.

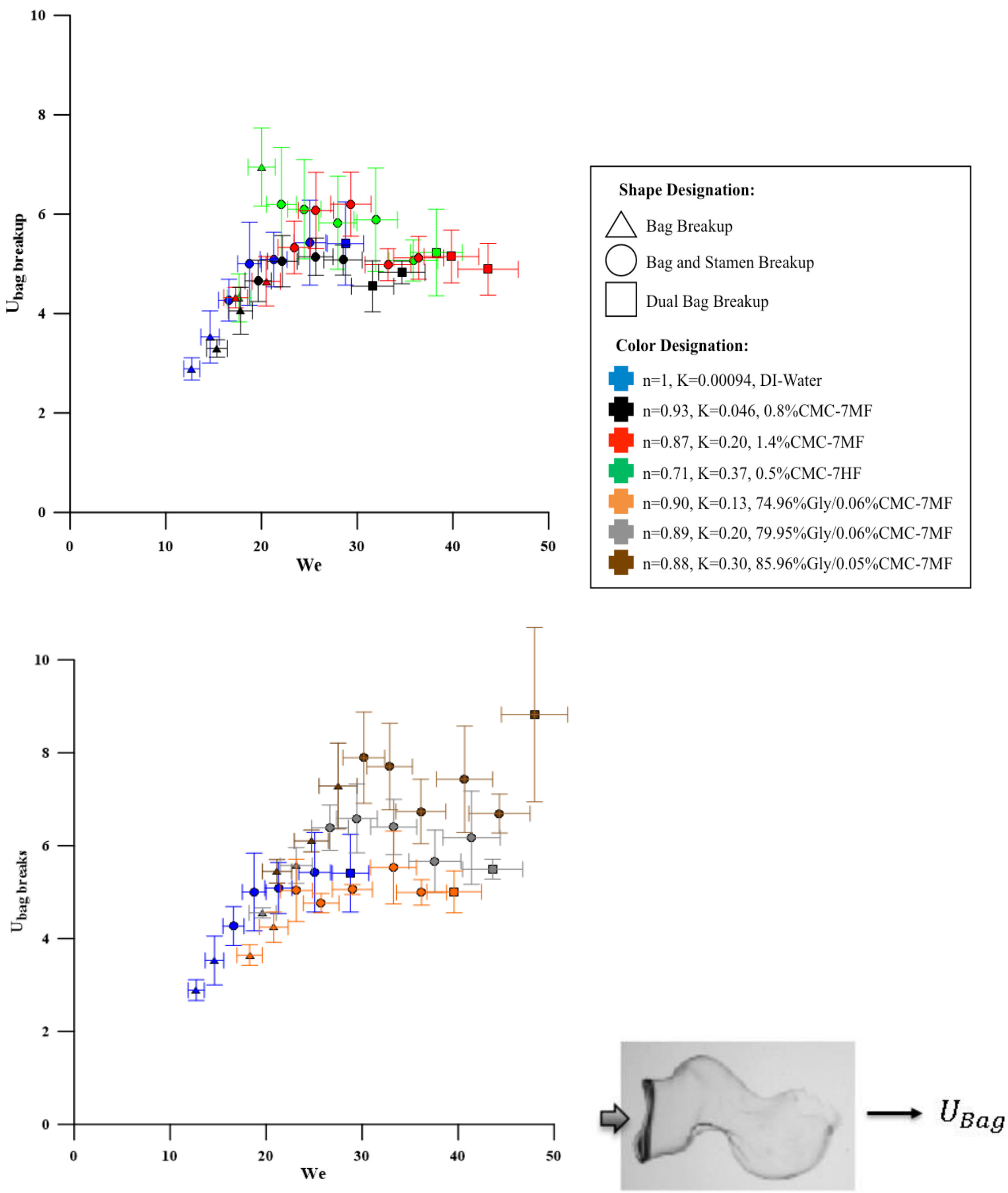


Figure 4.33. Drop Velocity (m/s) When Bag Breaks versus *We*.

4.4.5 Coefficient of Drag at Initiation Time

The final quantity of interest is C_d. This measurement is difficult to obtain since it depends on three simultaneous measurements, acceleration, velocity, and cross-stream diameter. For this reason, it has not been widely reported.

The most common method for computing C_d is to fit the displacement profile with the double integral of Newton's 2^{nd} Law. This was done by Ranger and Nicholls (1969). The inconsistencies related to this procedure are noted by Pilch and Erdman (1987).

Here, position measurements were taken and inserted into the drag relation to obtain C_d. The effective area was approximated as an axisymmetric flat circular disk in the bag breakup regime. However, in the bag-and-stamen and dual bag regimes the flat disk approximation is no longer valid.

Since drop velocity was measured as a function of time, the relative velocity was not assumed constant. Also, the acceleration was obtained by fitting the velocity to a smoothing polynomial and then differentiating it using central differencing with second order errors.

From Figure 4.34, we see that increasing *We* increased C_d for all liquids. Motion correlates with that for a solid spheroid of the same aspect ratio and *Re* range when We≈18 to 20. As *We* increases, C_d becomes a function of liquid rheology with larger *K* having a lower C_d. If *n* is also decreased (with large *K*) C_d increases. The acceleration is the dominant factor as it varies by about 7, while the cross-stream dimension varies by a factor of 1.25, and the relative velocity varies by a factor of 1.14. Although, the acceleration may be related to the cross-stream dimension so the cross-stream dimension may indirectly control the C_d variation. This possibility has been explored by Hsiang and Faeth (1995), who found that such a relationship does exist. Additionally, from Figure 4.34 we see there is a slope change within the bag-and-stamen breakup regime for each liquid. Therefore, as *K* increases the slope change occurs at a larger *We*, and if *n* is decreased the slope change occurs at a lower *We*.

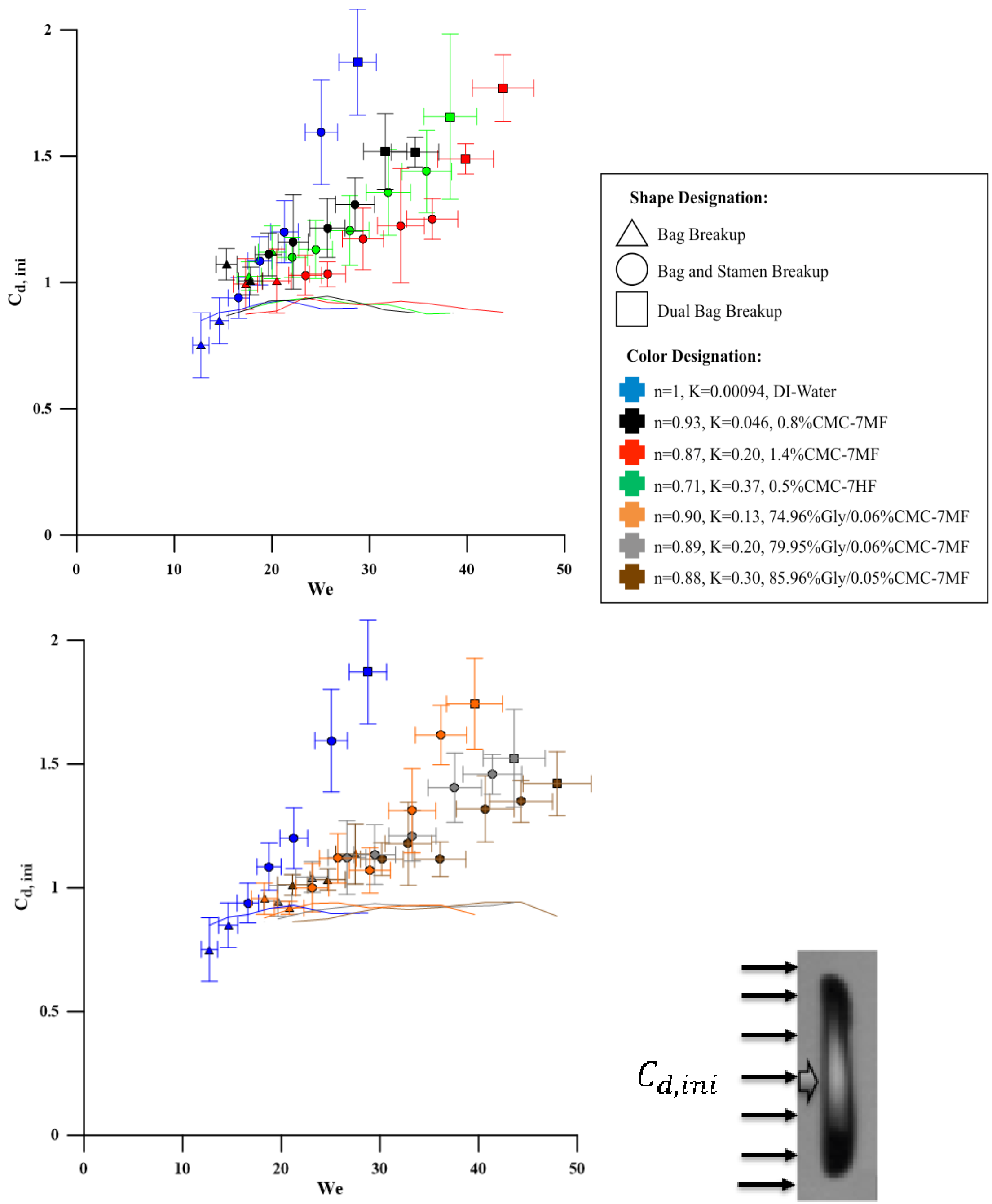


Figure 4.34. Coefficient of Drag at Initiation Time with correlations provided by Clift *et al*. (2005).

4.5 Summary and Conclusions

In this thesis work, six inelastic non-Newtonian liquids were formulated with power law parameters $0.708 \leq n \leq 0.933$ and $0.0464 \leq K \leq 0.3750$. DI-water was also included as a control and for comparisons with literature.

Results are presented as visualization, regime maps, breakup times, physical correlations, and drop dynamics. The MATLAB (2015b) code written for the extraction of data is provided in Appendix A. It was used to make 153,000 measurements, as each data point is constructed from 6 to 12 measurements with the uncertainty approximated using the standard deviation. It deviates from the 95% confidence interval by at most 10% (either 5% smaller or 10% greater depending on whether the average was from 6 or 12 quantities, respectively).

From the visualizations, the inelastic non-Newtonian bag, bag-and-stamen, and dual bag breakup morphologies share similar major breakup characteristics, and a few minor differences. During each type of breakup (bag, rim, and stamen) persistent ligaments form. The size and number of ligaments decreases as aerodynamic forces increase, and increase if *K* is increased. The affect is not diminished if *n* is decreased (with *K* still large). The ligaments do eventually break up by first forming nodes along their major axis, then break up between nodes, and finally form drops. Highly viscous Newtonian liquids do not exhibit this increased ligament formation, it is hypothesized that the ligament development is due to hydrogen bonding between polymer and solvent.

The classical *We* versus *Oh* regime map remains valid if the correct approximation for effective strain rate and corresponding effective viscosity are used. In this study, the strain rate was approximated as the strain rate up to the initiation time. It was in agreement with the convention that increased *Oh* results in increased *We* required to initiate a particular breakup regime, and in agreement with a Newtonian liquid literature correlation. In respect to power law parameters, if K is increased the *We* required to initiate a type of breakup increases, while if *n* is decreased the *We* to initiate a breakup mode decreases. The *Re* versus *We* regime map showed a physically revealing relationship in terms of aerodynamic forces, surface tension, and viscosity. Additionally, a regime map formed using the method developed by Zhao *et al*. (2010), that employs the Rayleigh Taylor Wave number (N_{RT}) to determine the breakup mode, agreed well for non-Newtonian liquids.

Many of the experimental results exhibited a morphological dependence. This dependence either resulted in a peak or a change in slope. The *We* where the peak or slope change occurred shifted to larger or smaller *We*: if *K* increased the peak or change in slope shifted to higher *We,* while if *n* decreased the peak or slope change shifted to smaller *We*.

The behaviors which had a peak in either the bag breakup or bag-and-stamen regimes are the cross-stream dimension at initiation time, rim diameter at bag breakup, length of bag when the bag breaks, displacement at initiation time, displacement when the bag breaks (only for low viscous liquids), and velocity at initiation time and when the bag breaks (expect water). These behaviors imply that when *We* increases they all first increase, peak, and then drop with further increases in *We*.

The correlations which exhibit only a slope change in the bag-and-stamen regime are the initiation time, the drag coefficient at initiation time, velocity of water at initiation time and when the bag breaks up. The initiation time decreases with *We* while C_d increases with *We*. The velocity of DI-water at initiation time and bag breakup increased with *We*. There is also a relationship which did not exhibit any morphological dependence--the bag breakup time (decreases with *We*).

Viscosity plays a large role. The curves which shift to higher *We* values throughout or larger peaks if *K* is increased are displacement when the bag breaks, velocity when the bag breaks, length of bag when the bag breaks, displacement at initiation time, velocity at initiation time, initiation time, and bag breakup time. The relations which shift to lower *We* as *K* increases are the diameter of the rim when the bag breaks, cross-stream dimension at initiation time, and coefficient of drag at initiation time.

The relationships that shift toward higher *We* values throughout or larger peaks by decreasing *n* include the diameter of the rim when the bag breaks, the velocity when the bag breaks, the length of the bag when it breaks, the cross-stream diameter at initiation time, the displacement at initiation time, and the coefficient of drag at initiation time. The relationships that shift toward lower *We* values throughout or lower peaks are the displacement when the bag breaks, the velocity at initiation time, initiation time, and bag breakup time.

The rim diameter scaling remains valid as long as density and surface tension are constant. In this case surface tension is the dominant liquid property for rim diameter at bag breakup. If surface tension drops, an increase in K is overshadowed by a decreased surface tension, which causes the rim diameter curve to shift toward higher values throughout at the time when the bag breaks up.

CHAPTER 5. NON-NEWTONAIN TAB MODEL

The Taylor Analogy Breakup (TAB) model is based on a liquid drop being analogous to a spring mass damper system and the deformation is due to an increase in the amplitude of the drop oscillation. It was developed by O'Rourke and Amsden (1987). The model predicts the deformation of a spherical drop into an ellipsoid. The relationship describing the force balance is,

$$F = kx + d\dot{x} + m\ddot{x} \qquad (5.1)$$

Where F, k, d, and m is the force, spring constant, damping constant, and mass. Due to experience with the original TAB model, the rate of deformation was not large enough. This was the result of either the aerodynamic forces not being large enough or the resistive forces being too large. Through the improved TAB model proposed by Park and Yoon (2002) the aerodynamic force term was altered from the original relation. The alteration to the equation is based on the understanding that the external force should also increase as the drop deforms and is proportional to the squared increase, analogous to area. The final equations are,

$$\frac{F}{m} = C_F \frac{\rho_G u_{rel}^2}{\rho_L r_0}(1 + 0.5y)^2 \qquad (5.2)$$

$$\frac{k}{m} = C_k \frac{\sigma}{\rho_L r_0^3} \qquad (5.3)$$

$$\frac{d}{m} = C_v \frac{\mu_L}{\rho_L r_0^2} \qquad (5.4)$$

Therefore, the applied force due to aerodynamic forces is equal to the restorative surface tension forces, viscous dampening forces, and the force due to acceleration of the equator. This relationship is typically non-dimensionalized using,

$$y = \frac{x}{C_b r_0} \; ; \; \frac{R}{r_0} = (1 + 0.5y) \tag{5.5}$$

Inputting these relations into the Equation (5.1) and solving for the non-dimensional drop acceleration gives a 2nd order ODE of the form.

$$\frac{d^2y}{dt^2} = \frac{C_F}{C_b}\frac{\rho_G u_{rel}^2}{\rho_L r_0^2}(1 + 0.5y)^2 - \frac{C_k \sigma}{\rho_L r_0^3} y - \frac{C_v \mu_{eff}}{\rho_L r_0^2}\frac{dy}{dt} \tag{5.6}$$

Where, the constants have been determined by O'Rourke and Amsden (1987) as,

$$C_k = 8 \; ; \; C_F = \frac{1}{3} \; ; \; C_v = 5 \; ; \; C_b = \frac{1}{2} \tag{5.7}$$

With these constants the final relation is,

$$\frac{d^2y}{dt^2} = \frac{2}{3}\frac{\rho_G u_{rel}^2}{\rho_L r_0^2}(1 + 0.5y)^2 - \frac{8\sigma}{\rho_L r_0^3} y - \frac{5\mu_{eff}}{\rho_L r_0^2}\frac{dy}{dt} \tag{5.8}$$

However, it is noted that further experimental comparisons are required to determine the model constants more precisely Liu *et al*. (1993). Therefore, some work is needed here. Additionally, from Equation (5.8) it becomes physically reasonable to state that in order for the drop equator to accelerate, the aerodynamic forces must be larger and overcome the restorative surface tension forces and the dampening viscous forces.

Also, it was stated by O'Rourke and Amsden (1987) that breakup will occur when y>1 or in other terms, the ratio $\frac{R}{r_0} = 1.5$. From other experimental data and this current work, the condition seems rather a minimum condition instead of the actual deformation during breakup. Furthermore, the cross-stream dimension varies with *We* and with *Oh* and most non-dimensional cross-stream data was larger than 1.5. Therefore, $\frac{R}{r_0} = 1.5$ or y>1 is not a correct condition to estimate the maximum cross-stream dimension. Instead, for each video and set of measurements with MATLAB the cross-stream diameter at initiation time is extracted and used as the criteria for the end condition for that specific flow and liquid properties.

What's more, the model requires the relative velocity to be determined in the case where velocity data is not available. Simply using Newton's 2nd Law and solving for acceleration, the first order ODE for drop velocity is,

$$\frac{du_d}{dt} = \frac{3}{8}\frac{\rho_G}{\rho_L}\frac{R^2}{r_0^3}u_{rel}^2 C_d \tag{5.9}$$

To solve for this velocity, the Cd is needed. Liu *et al*. (1993) provided an approximate linear expression which assumes that the Cd must be between the limits for that of a sphere to that of a disk. The relationship of Cd with non-dimensional equator displacement is,

$$C_d = C_{d,sphere}(1 + 2.632y) \tag{5.10}$$

However, it is discussed by Liu *et al*. (1993) that this relationship requires some experimental validation. After obtaining velocity data with MATLAB, it can be then used to compare the experimental velocity with the model and determine the best coefficient in front of the non-dimensional equator displacement, y. The relation would take the form,

$$C_d = C_{d,sphere}(1 + q \cdot y) \tag{5.11}$$

where q is iterated between 0.01 to 10.0 and $C_{d,sphere} = 0.445$. This drag value was chosen for simplicity and because within the range $750 \le Re \le 3.5 \times 10^5$ (called the "Newton law range") the $C_{d,sphere}$ only varies about ±13% from the value $C_{d,sphere} = 0.445$ (Clift *et al*., 1978). Also, the experiments conducted in this thesis involve a range of *Re* between $2900 < Re < 5400$ and therefore likely varies much less.

Additionally, since for non-Newtonian liquids, the viscosity is not constant then the effective viscosity must also be approximated. The strain rate is estimated as discussed previously,

$$\dot{\gamma}_{Avg} \approx \dot{\varepsilon}_{Avg} = \frac{d_{cro,ini} - d_0}{d_0 t_{ini}} = \left(\frac{d_{cro,ini}}{d_0} - 1\right)\frac{1}{t_{ini}} = \left(\frac{d_{cro,ini}}{d_0} - 1\right)\frac{u_{rel}}{T_{ini} d_0}\sqrt{\frac{\rho_G}{\rho_L}} \tag{5.12}$$

Finally, the velocity and TAB model is solved using the 4th order Runge-Kutta method. The program cannot predict the final cross-stream dimension due to the dimension being a predefined value in order to stop the program. It is possible to compare the temporal cross-stream dimension, however with seven liquids and five *We* tested the number of plots would be 35 which is impractical. Therefore, the value which determines the models accuracy is the initiation time. Also, it is valuable to compare the velocity at T_{ini} to determine whether the method used in determining C_d is accurate.

5.1 Initiation Time

This time is experimentally obtained through a MATLAB code which approximates time zero as when the drop has deformed by 10% (no less because of noise) and then defines the initiation time when the stream-wise (or axial) length is a minimum. The cross-stream dimension and velocity up to this initiation time are then used in the improved non-Newtonian TAB model to obtain an experimentally valid drag coefficient with increases in the non-dimensional displacement of the drop equator. Then, the end condition is determined by the experimental cross-stream dimension. The purpose of this study is actually one phase in improving the TAB model. Before continuing in later studies first it is proved here whether the new TAB model will:

1. With the cross-stream dimension known accurately, say with some dependable correlation as a function of *We* and *Oh*, will the initiation time be accurately predicted? This cross-stream dimension must be known prior to running the code in order to define the stopping criteria (historically y>1 is used but has been proven to be incorrect).
2. With a trustworthy drag relationship, albeit a function of only non-dimensional equator displacement or an added *We* and *Oh* term, will the velocity at initiation time be predicted correctly?

Due to the new strain rate model used in this study being produced after post processing approximately 400 videos, there is only a limited number of model predictions. For each liquid, the tested *We* range for each liquid was broken up into 5 evenly separated We. Thus, with 7 liquids the TAB model was tested 35 times. While it sufficiently shows whether the model is promising, there is still the necessity to complete more runs for statistical validity. Regardless, from Figure 5.1 there does seem to be very reasonable agreement with the experimentally determined initiation time (including uncertainties) to that predicted by the TAB model. However, relative to the average at the same *We* which the TAB model depended on, the maximum percent difference was about 29% while on average the percent difference was 11%. These percent's do not consider the experimental uncertainty and are therefore much lower. While, the magnitudes of the initiation time are approximately accurate the expected trends are not present.

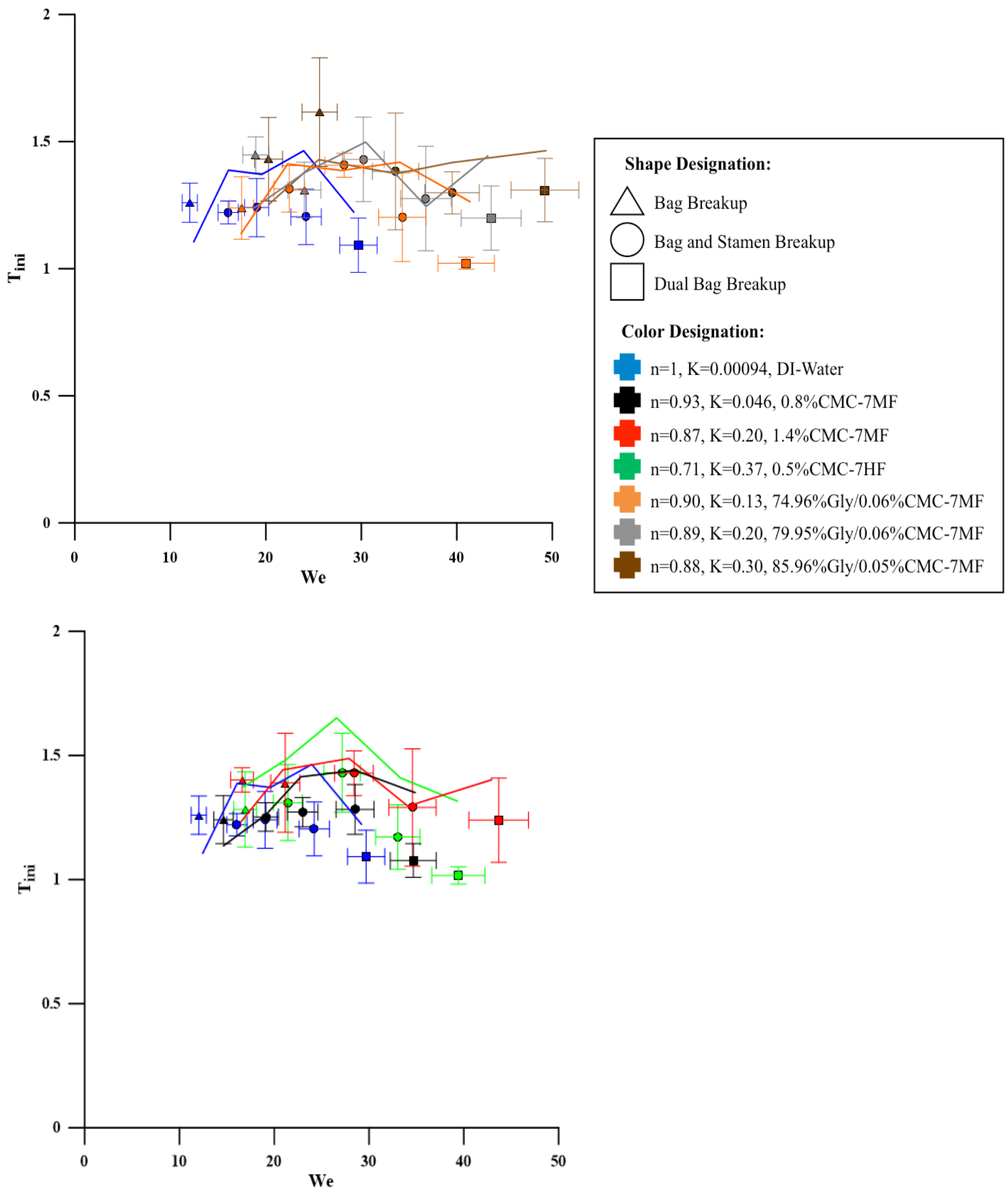


Figure 5.1. Comparing Experimental and Predicted Non-Dimensional Initiation Time.

From the experimental data, it is expected that the more viscous liquids will have a longer initiation time. This trend is not followed by all the liquids. In the next phase of developing this TAB model, with the current data available correlations can be made for the cross-stream diameter and coefficient of drag, and the differences between liquids may emerge with those predictions.

5.2 Velocity at Initiation Time

As discussed previously, to determine an accurate velocity at initiation and velocity prior to that, the TAB model required first determining the C_d which best fits the experimental data. In Figure 5.2, all 35 TAB model runs are included. This C_d comes from Equation 5.11. There is some scatter although the magnitudes are increasing systematically with *We*. This is in accordance with an increased velocity at initiation time found experimentally. The curves added on the plot are a linear relation and the 95% confidence interval. This relationship is,

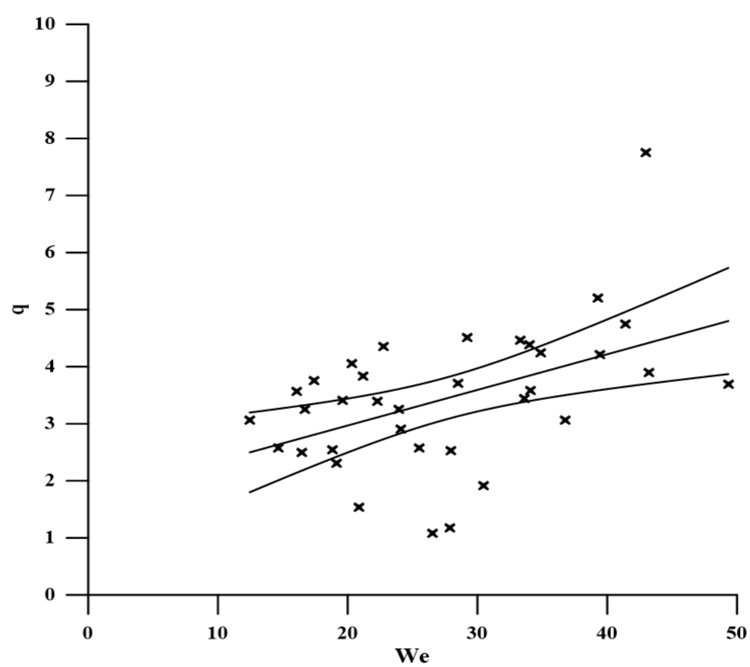


Figure 5.2. The Coefficient Used in Front of the Non-Dimensional Equator Displacement in Equation 5.11.

$$q = 0.0624We + 1.72;\ 12 < We < 49 \tag{5.13}$$

Originally Liu *et al*. (1993) recommended a constant coefficient of 2.632 to use in the drag model. However, this equation above varies between q=2.47 at We=12 and q=4.78 at We=49. The latter value being in agreement with the observation of Park and Yoon (2002) who also found that the coefficient of drag is being underestimated by Liu *et al*. (1993) at We=50. Additionally, this increasing trend is in agreement with the experimental coefficient of drag at initiation time found in this study. It is increasing with increases in aerodynamic forces.

What's more, it should be noted that in the computations of the TAB model, the velocity of the drop does not significantly affect the predictions. Even if we consider the extreme case where drop velocity is zero,

$$\frac{F}{m} \propto u_{rel}^2 = u_G^2 \tag{5.14}$$

Then, we consider if the drop had the maximum value at initiation time found in these experiments,

$$u_d = 0.10u_G \tag{5.15}$$

Plugging that into the above proportionality,

$$\frac{F}{m} \propto u_{rel}^2 = (u_G - u_d)^2 = (0.90u_G)^2 = 0.81u_G^2 \tag{5.16}$$

Therefore, if we assume the drop velocity is zero then the force term would be 23% (1/0.81) larger. This is of course an extreme scenario and the drop is always assumed to have a velocity. The point being that any assumed velocity albeit lower than the actual would still result in an improvement from the 23%.

With C_d determined using experimental data, the velocity at initiation time can be compared. It should be understood that the time dependent experimental velocity is used when determining C_d, and not y the final initiation time velocity. As a result, there will be differences.

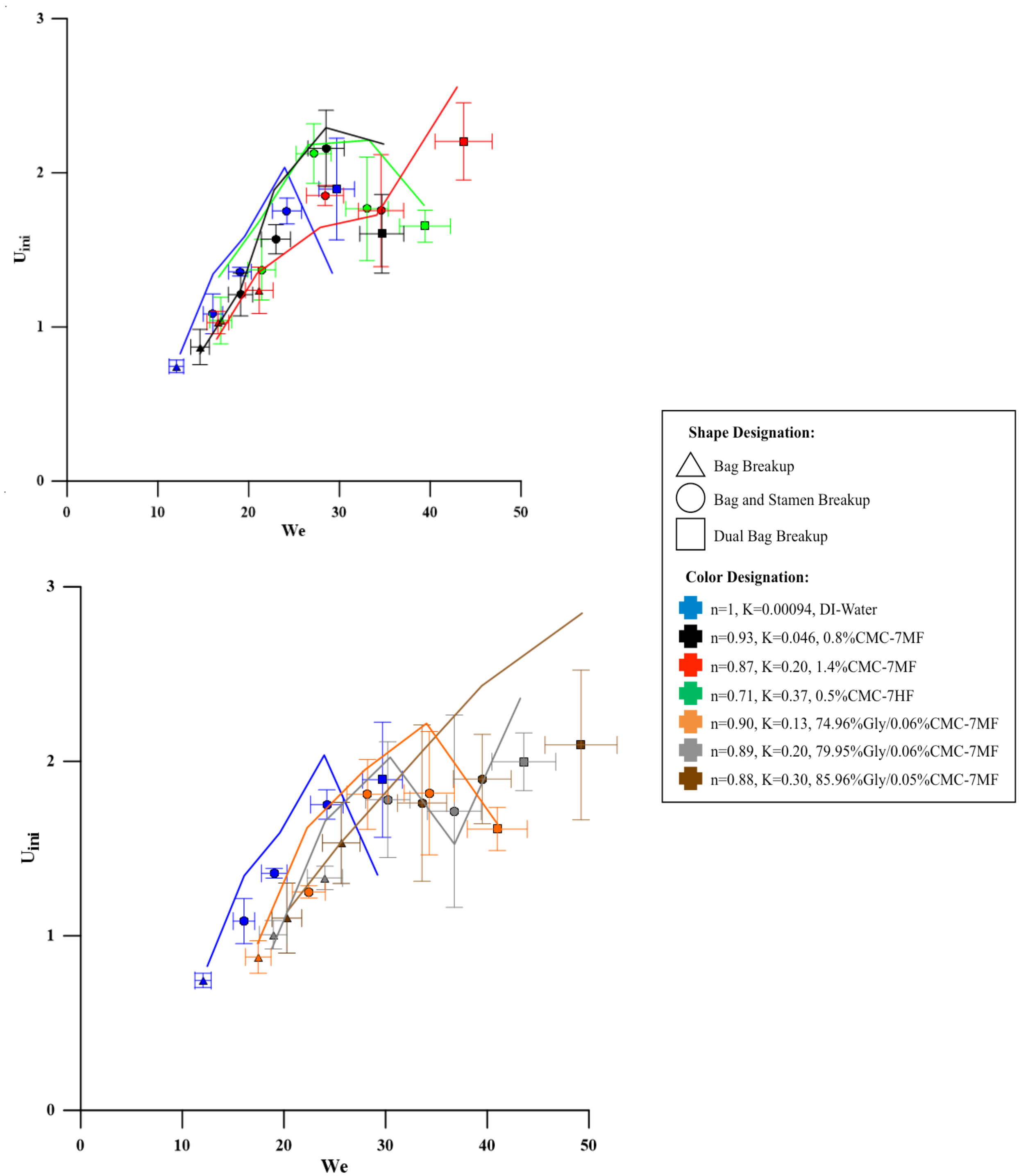


Figure 5.3. Comparing Experimental and Predicted Velocity at Initiation Time.

From Figure 5.3 it is apparent that the prediction approximately follows the trends of each liquid. They all show an increase in velocity as *We* is increased and then those which exhibit a peak will then have a decline as does the experimental values. The maximum percent difference of velocity is 36% and the average is 15%. Of course these percentages are based on differences from the average and do not include the experimental uncertainties. Therefore, the actual percentage would be lower. Additionally, it may improve the model further if the coefficient of drag varied in a parabolic fashion with the increase in non-dimensional displacement, as discussed by Park and Yoon (2002). Furthermore, as can be seen from the experimental data, the velocity at initiation time differs between liquids and therefore the coefficient of drag may require a relation which incorporates variation with *We* and *Oh*. This statement is supported by the experimental results of this work which showed variation in velocity between liquids.

5.3 Summary and Conclusions

The TAB model was originally developed by O'Rourke and Amsden (1987), then refined by Liu *et al*. (1993), followed by being made non-Newtonian by Lopez (2010). Next, through recommendations proposed by Park and Yoon (2002), the model developed by Lopez (2010) was altered to increase the aerodynamic force term. Also, the strain rate model was changed. Furthermore, since it has been noted by the authors above that the velocity is underestimated, an experimentally determined C_d was found; it is an increasing function of *We*. Additionally, the cross-stream dimension has been shown to vary with *We* and *Oh* so an experimentally determined cross-stream dimension were used in lieu of the y>1 condition.

The result was a predicted initiation time and velocity at initiation time of similar magnitude to experimental data. The maximum percent difference from the experimental average cross-stream dimension at initiation time was 29% and on average 11%. The largest percent difference comparing velocity at initiation time was 36% and on average 15%. These percentages to do not consider experimental uncertainties. Therefore, with an appropriate coefficient of drag model to predict drop velocity and an appropriate correlation for cross-stream dimension the initiation time can be predicted with reasonable accuracy using the TAB model.

CHAPTER 6. SUMMARY AND CONCLUSION

6.1 Experimental Summary

A comprehensive study of Newtonian and Non-Newtonian liquid secondary atomization literature was completed. This allowed for comparisons in both trend direction and magnitude with the plethora of experimental data measured in this thesis. Since there exists much more Newtonian liquid literature, the experiments included water as the experimental control liquid.

The conditions of this experiment was incompressible flow since Ma was always less than Ma<0.1 using a continuous jet setup. Also, basset history and virtual mass forces were not important due to ${\rho_L}/{\rho_G} \gg 1$ and pressure gradients were assumed negligible because the drop falls into a uniform air jet. Furthermore, gravity was assumed negligible since experimental data showed that the drops accelerated at least 20 g's at the initiation time. The *We* was varied between $12 < We < 48$. As for the non-Newtonian power law parameters, the flow behavior index, n, was varied between $0.708 \leq n \leq 0.933$ and the consistency index, K, was in the range of $0.0464 \leq K \leq 0.3750 Pa \cdot s^n$. The density had at most 22% ($990 < \rho_L < 1210\ \frac{kg}{m^3}$) variation and the surface tension had at most 12% ($0.0649 < \sigma < 0.0728\ \frac{N}{m}$).

The effective viscosity was estimated using the strain rate up to the initiation time and resulted in an approximate range of ($0.000945 < \mu < 0.182\ Pa \cdot s$). With this estimated viscosity, the Oh varied between ($0.00216 < Oh < 0.4$). Additionally, each liquid was completely tested within the bag-and-stamen regime and overlapped into the bag and dual bag regimes. The data was extracted from high speed black and white videos using a MATLAB code. The time zero was defined as when the drop deforms by 10%, initiation time is when the axial length is minimal and the bag breakup time was user defined through a popup movie and dialogue boxes.

6.2 Conclusions

From the visualizations, the non-Newtonian liquid breakup modes tested here share similar major breakup characteristics with Newtonian liquid drops. There are a few minor differences which involve persistent ligaments throughout every stage of breakup. The ligaments do eventually breakup further into drops. It is hypothesized that the ligament development is due to hydrogen bonding between polymer and solvent. Additionally, there is significant non-uniform bag growth in the bag breakup regime.

The classical *We* versus *Oh* regime map did remain valid even for non-Newtonian liquids. It was also of similar magnitude compared to literature with the estimated viscosity. Two other correlations were found to adequately act as alternate regime maps. These include the liquid Reynolds number versus *We* and the Rayleigh-Taylor Wave number versus We. Additionally, many of the experimental relationships found in this work showed a peak or slope change as the liquid transitions between bag, bag-and-stamen, and dual bag breakup regimes which implies a morphological dependence. However, other relationships were strictly independent of any morphological changes. While most relationships seemed to be dependent on viscosity as the dominant liquid property, the rim diameter at bag breakup showed significant dependence on either surface tension or density. It is hypothesized here that surface tension is the culprit due to obvious physical explanation that reduced surface tension may allow for increased rim diameter growth.

Due to the large amount of data and relations in this thesis the general trends found with the consistency index, *K*, flow behavior index, *n*, and the *We* are tabulated below. Here the "→", "←", "↑", and "↓" means that the curve of the trend generally shifts right, left, up, and down, respectively under the given conditions for the relation of interest. However, this shifting is not to be assumed to mean that all values are necessarily always larger or smaller but is rather the shifting of the peak value or position of the slope change. Also, in the column under "Increase We" it tells how the trend varies with We. The "peak" term implies that the trend increases, reaches a peak, and then decreases. These approximations are necessary since the behavior of these liquids is complex and the explanations would other be severely tedious.

Table 6.1 .Scaling in Terms of *K*, and *n* and *We*.

Relation	**Increase *K***	**Decrease *n***	**Increase *We***
Initiation time	→/↑	←/↓	Decreases
Bag Breakup Time	↑	↓	Decreases
Cross Stream Diameter at Initiation Time	→/↓	←/↑	Peaks
Rim Diameter When Bag Breaks (If Density and Surface Tension Equal)	→/↓	←/↑	Peaks
Bag Length When Bag Breaks	→/↑	←	Peaks
Displacement at Initiation Time	→/↑	←	Peaks
Displacement When Bag Breaks (Low Viscous Liquids)	→/↑	←/↓	Peaks
Velocity at Initiation Time (except water)	→/↑	←/↓	Peaks
Velocity When Bag Breaks (except water)	→/↑	←/↑	Peaks
Coefficient of Drag at Initiation Time	↓	↑	Increases

The final topic of this thesis work was to attempt to improve and determine whether the TAB model was a viable option to accurately predict initiation time and velocity at initiation time for non-Newtonian liquids. First, the strain rate model was added to determine the effective viscosity and then the aerodynamic force term was altered in order to take the increasing equator into consideration. Next, using the data available through post processing, the experimental velocity and cross-stream dimension were used to estimate drop velocity and act as the end condition for the TAB model, respectively. The final result was a maximum percent difference from the experimental average cross-stream dimension at initiation time of 29% and on average 11%. The largest percent difference comparing velocity at initiation time was 36% and on average 15%. Therefore, the TAB model reasonably predicts most of the initiation times and velocity with acceptable accuracy and should be pursued further.

6.3 Future Work

For future work, the current data found in this thesis will be published and correlations made based on variation with *We* and *Oh*. Furthermore, the TAB model will be further pursued as there seems to be significant promise. Moreover, with the MATLAB code developed here, many challenges related to post processing these dynamic deforming drops has already been accomplished. It is possible to make the code more sophisticated by working toward automation for more statistically improved data. Furthermore, it is possible to add coding in order to incorporate measuring more breakup regimes, and also develop a means of taking measurements for later times after the bag has ruptured.

In terms of other possible studies, non-polar polymers and non-polar solutions could be tested in order to verify whether hydrogen bonding or polymer entanglement is the cause for ligaments formation. Non-polar molecules have equally distributed electrons around the entire molecule. Therefore, the strength of the bonds is weaker compared to polar molecules whose negative and positive distribution produces a dipole moment. If the non-polar solution did not form ligaments then the ligaments are formed because of polar bonds and not solely all non-Newtonian liquids. What's more, the difference between non-polar solution, polar solution and Newtonian liquid breakup could be emphasized.

Also, to validate or disprove the strain rate models used thus far, it is necessary to produce a Newtonian *We* versus *Oh* map with the same system which will be used to test the non-Newtonian liquids. First, take about 10 Newtonian liquids with varying but known viscosity, which must be large enough to compare to the non-Newtonian viscosity. Then, run those liquids through the flow rates/*We* and obtain only the *We* and *Oh* regime map. After that, the non-Newtonian liquid can be tested and the strain rate model can be adequately proved, disproved, or further developed.

Additionally, very large viscosity liquids ($\mu \gg 0.171$) either Newtonian or non-Newtonian should be tested (may require a larger nozzle since breakup time will significantly increase) in order to prove whether the Rayleigh Taylor Wave number versus *We* regime map still applies, whether the liquid Reynolds number versus *We* regime map still applies, and if the bag lengths become as significantly long as with non-Newtonian liquids.

LIST OF REFERENCES

LIST OF REFERENCES

Arcoumanis, C., Khezzar, L., Whitelaw, D. S., & Warren, B. C. H. (1994). Breakup of Newtonian and non-Newtonian fluids in air jets. *Experiments in fluids*, *17*(6), 405-414.

Arcoumanis, C., Whitelaw, D. S., & Whitelaw, J. H. (1996). Breakup of droplets of Newtonian and non-Newtonian fluids. *Atomization and Sprays*, *6*(3).

Arnold, R., S. Santos, P. H., Campanella, O. H., & Anderson, W. E. (2011). Rheological and Thermal Behavior of Gelled Hydrocarbon Fuels. *Journal of Propulsion and Power*, *27*(1), 151-161.

Bonferoni, M. C., Rossi, S., Ferrari, F., Bertoni, M., & Caramella, C. (1995). Influence of medium on dissolution-erosion behaviour of Na carboxymethylcellulose and on viscoelastic properties of gels. *International journal of pharmaceutics*, *117*(1), 41-48.

Chou, W. H., & Faeth, G. M. (1998). Temporal properties of secondary drop breakup in the bag breakup regime. *International journal of multiphase flow*, *24*(6), 889-912.

Clift, R., Grace, J. R., & Weber, M. E. (2005). *Bubbles, drops, and particles*. Courier Corporation.

Dai, Z., & Faeth, G. M. (2001). Temporal properties of secondary drop breakup in the multimode breakup regime. *International Journal of Multiphase Flow*, *27*(2), 217-236.

Dolz, M., Roldan-Garcia, C., Herraez-Dominguez, J. V., & Belda-Maximino, R. (1991). Thixotropy of different concentrations of microcrystalline cellulose: sodium carboxymethyl cellulose gels. *Journal of pharmaceutical sciences*, *80*(1), 75-79.

Dorsey, N. E. (1940). Properties of ordinary water-substance.

Flock, A. K., Guildenbecher, D. R., Chen, J., Sojka, P. E., & Bauer, H. J. (2012). Experimental statistics of droplet trajectory and air flow during aerodynamic fragmentation of liquid drops. *International Journal of Multiphase Flow*, *47*, 37-49.

Gao, J., Rodrigues, N. S., Sojka, P. E., & Chen, J. (2014, August). Measurement of Aerodynamic Breakup of Non-Newtonian Drops by Digital In-Line Holography. In *ASME 2014 4th Joint US-European Fluids Engineering Division Summer Meeting collocated with the ASME 2014 12th International Conference on Nanochannels, Microchannels, and Minichannels* (pp. V002T11A009-

Gel'Fand, B. E., Gubin, S. A., & Kogarko, S. M. (1974). Various forms of drop fractionation in shock waves and their special characteristics. *Journal of Engineering Physics and Thermophysics*, *27*(1), 877-882.

Gel'Fand, B. E., Gubin, S. A., Kogarko, S. M., & Komar, S. P. (1973). Singularities of the breakup of viscous liquid droplets in shock waves. *Journal of engineering physics and thermophysics*, *25*(3), 1140-1142.

Guildenbecher, D. R. (2009). Secondary atomization of electrostatically charged drops.

Guildenbecher, D. R., Lopez, C., & Sojka, P. E. (2009). Secondary atomization. *Experiments in Fluids*, *46*(3), 371-402.

Hinze, J. O. (1955). Fundamentals of the hydrodynamic mechanism of splitting in dispersion processes. *AIChE Journal*, *1*(3), 289-295.

Holdsworth, S. D. (1971). Applicability of rheological models to the interpretation of flow and processing behaviour of fluid food products. *Journal of Texture Studies*, *2*(4), 393-418.

Hsiang, L., & Faeth, G. (1994). Drop Deformation and Breakup Due to Shock Wave and Steady Disturbances. *AIAA, 94-0560* .

Hsiang, L. P., & Faeth, G. M. (1995). Drop deformation and breakup due to shock wave and steady disturbances. *International Journal of Multiphase Flow*, *21*(4), 545-560.

Hsiang, L. P., & Faeth, G. M. (1992). Near-limit drop deformation and secondary breakup. *International Journal of Multiphase Flow*, *18*(5), 635-652.

Joseph, D. D., Beavers, G. S., & Funada, T. (2002). Rayleigh–Taylor instability of viscoelastic drops at high Weber numbers. *Journal of Fluid Mechanics*, *453*, 109-132.

Joseph, D. D., Belanger, J., & Beavers, G. S. (1999). Breakup of a liquid drop suddenly exposed to a high-speed airstream. *International Journal of Multiphase Flow*, *25*(6), 1263-1303.

Krzeczkowski, S. A. (1980). Measurement of liquid droplet disintegration mechanisms. *International Journal of Multiphase Flow*, *6*(3), 227-239.

Kulkarni, V. (2013). *An analytical and experimental study of secondary atomization of vibrational and bag breakup modes* (Doctoral dissertation, PURDUE UNIVERSITY).

Lane, W. R. (1951). Shatter of drops in streams of air. *Industrial & Engineering Chemistry*, *43*(6), 1312-1317.

Lee, B. B., Chan, E. S., Ravindra, P., & Khan, T. A. (2012). Surface tension of viscous biopolymer solutions measured using the du Nouy ring method and the drop weight methods. *Polymer bulletin*, *69*(4), 471-489.

Liu, A. B., Mather, D., & Reitz, R. D. (1993). *Modeling the effects of drop drag and breakup on fuel sprays* (No. TP-930072). WISCONSIN UNIV-MADISON ENGINE RESEARCH CENTER.

Lopez, C. (2010). Secondary breakup of inelastic non-Newtonian liquid drops.

Mallory, J. A. (2012). Jet impingement and primary atomization of non-Newtonian liquids.

Metzner, A. B., & Otto, R. E. (1957). Agitation of non-Newtonian fluids. *AIChE Journal*, *3*(1), 3-10.

Ng, C. L., & Theofanous, T. G. (2008, July). Modes of Aero-Breakup with Visco-Elastic Liquids. In *THE XV INTERNATIONAL CONGRESS ON RHEOLOGY: The Society of Rheology 80th Annual Meeting* (Vol. 1027, No. 1, pp. 183-185). AIP Publishing.

O'Rourke, P. J., & Amsden, A. A. (1987). *The TAB method for numerical calculation of spray droplet breakup* (No. 872089). SAE Technical Paper.

Park, J. H., Yoon, Y., & Hwang, S. S. (2002). Improved TAB model for prediction of spray droplet deformation and breakup. *Atomization and Sprays*, *12*(4).

Pilch, M., & Erdman, C. A. (1987). Use of breakup time data and velocity history data to predict the maximum size of stable fragments for acceleration-induced breakup of a liquid drop. *International Journal of Multiphase Flow*, *13*(6), 741-757.

Rao, M. A. (2014). Flow and functional models for rheological properties of fluid foods. In *Rheology of Fluid, Semisolid, and Solid Foods* (pp. 27-61). Springer US.

Simpkins, P. G., & Bales, E. L. (1972). Water-drop response to sudden accelerations. *Journal of Fluid Mechanics*, *55*(04), 629-639.

Snyder, S. (2015). Secondary Atomization of Elastic Non-Newtonian Liquid Drops, PhD thesis, Purdue University.

Tarnogrodzki, A. (2001). Theory of free fall breakup of large drops. *International journal of mechanical sciences*, *43*(4), 883-893.

Taylor, J. R. (1982). An Introduction to Error Analysis: The Study of Uncertainties in Physical Measurements, 327 pp. *Univ. Sci. Books, Mill Valley, Calif*.

Theofanous, T. G. (2011). Aerobreakup of Newtonian and viscoelastic liquids. *Annual Review of Fluid Mechanics*, *43*, 661-690.

Theofanous, T. G., Mitkin, V. V., & Ng, C. L. (2013). The physics of aerobreakup. III. Viscoelastic liquids. *Physics of Fluids (1994-present)*, *25*(3), 032101.

Wilcox, J. D., June, R. K., Brown, H. A., & Kelley, R. C. (1961). The retardation of drop breakup in high-velocity airstreams by polymeric modifiers. *Journal of Applied Polymer Science*, *5*(13), 1-6.

Yang, X. H., & Zhu, W. L. (2007). Viscosity properties of sodium carboxymethylcellulose solutions. *Cellulose*, *14*(5), 409-417.

Yoon, H. K., & Ghajar, A. J. (1987). A note on the Powell-Eyring fluid model. *International communications in heat and mass transfer*, *14*(4), 381-390.

Zhao, H., Liu, H. F., Li, W. F., & Xu, J. L. (2010). Morphological classification of low viscosity drop bag breakup in a continuous air jet stream. *Physics of Fluids*, *22*(11), 4103.

Zhao, H., Liu, H. F., Xu, J. L., & Li, W. F. (2011). Experimental study of drop size distribution in the bag breakup regime. *Industrial & Engineering Chemistry Research*, *50*(16), 9767-9773.

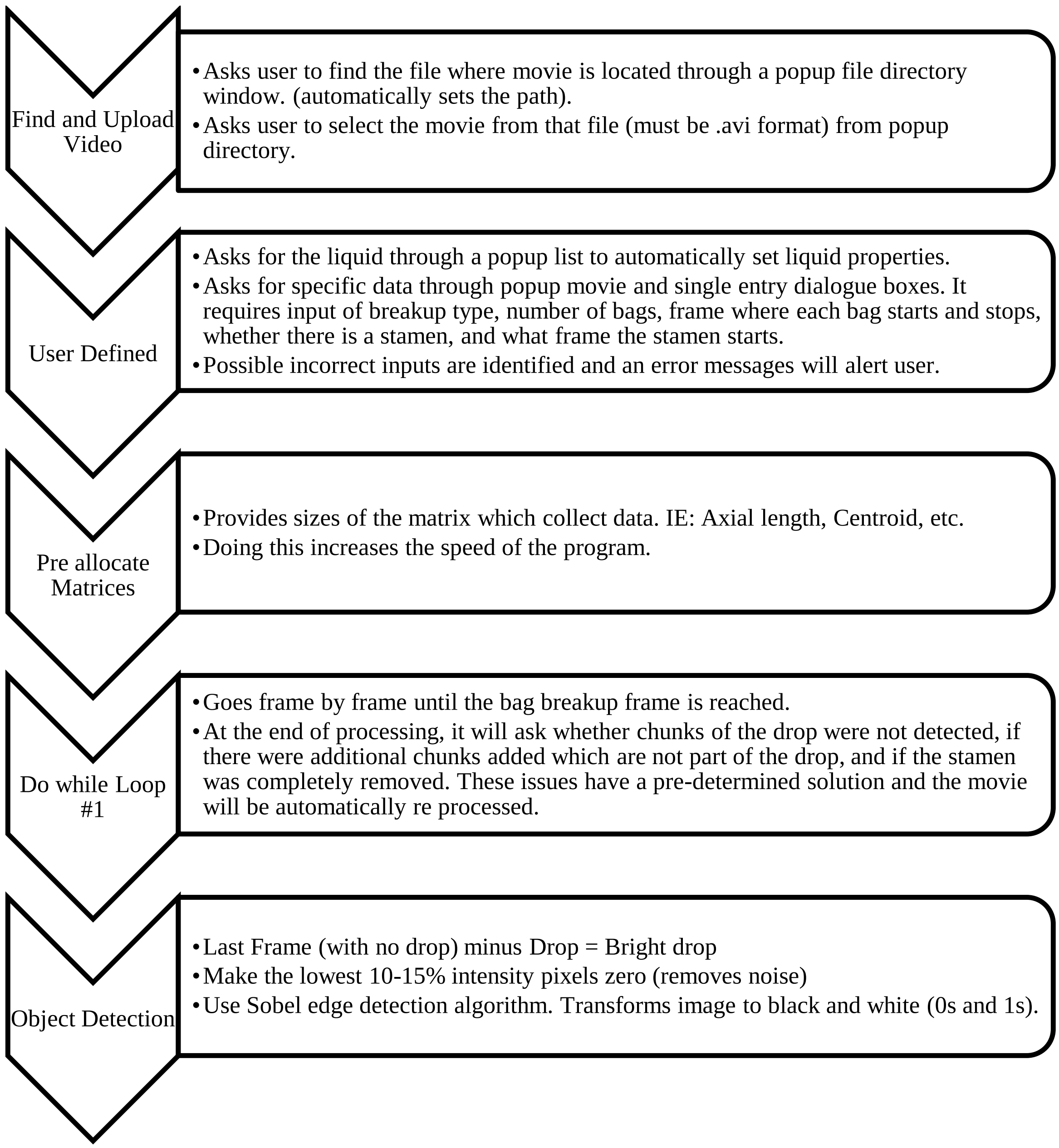
Zhao, H., Liu, H. F., Xu, J. L., Li, W. F., & Lin, K. F. (2013). Temporal properties of secondary drop breakup in the bag-stamen breakup regime. *Physics of Fluids (1994-present)*, *25*(5), 054102.

Zuidema, H., & Waters, G. (1941). Ring Method for Determination of Intrfacial Tension. *Industrial & Engineering Chemistry Analytical Edition*, *13*(5), 312-313.

APPENDIX

APPENDIX

Appendix A. Flow Chart of MATLAB Program:



Do While Loop #2

- While the number of objects detected is greater than 1.
- Remove pixels area less than 4+i
- Erosion and dilation.
- Fill drop.
- Check number of objects, If > 1, i=i+1. else, end.

Cross Stream Length

- Methods depend on whether the drop has deformed by 10%, if the bag has emerged, and if a stamen is present.
- The methods include simply choosing the column with the maximum number of white pixels, choosing top and bottom bounding box extrema, gathering boundary points and adding and subtracting x and y coordinates to get the maximum and minimum values, and removing the bag completely before performing operations.

Stream Wise Length

- Methods depend on whether the bag has formed, if a stamen is present, and if one or two bags is present.
- Methods include obtaining the row with the maximum number of white pixels and removing stamen.
- If there are two bags, then those same maximums are obtained above and below the average point. Also, when each bag begins and ends becomes important.

Centroids

- If the bag hasn't emerged, then the area centroid is used.
- Once the bag emerges then the centroid is the average between end points of the column length and is physically the rim.

Plots

- As each frame is processed, images from each step in drop identification are presented for data verification and trouble shooting.
- Steps include background minus drop, Sobel edge detection, noise removal, dilation, erosion, boundary fill, and then all data gathered is plotted onto the original video.

Data of Interest

- For initiation time, 2 criteria were found to obtain similar and accurate results. These are the minimum axial length and minimum bounding box length.
- Also, many of the time varying data are non-dimensionalized, specific data points are non-dimensionalized, and non-dimensional parameters are calculated.
- Furthermore, data is extracted at initiation time and bag breakup time from the time varying data.

Further Processing

- With the time varying displacement, it is possible to calculate the velocity, acceleration, and the coefficient of drag.
- Forward differencing is used at the first point, then central differencing, followed by backward differencing at the end point.
- Acceleration required first curve fitting the velocity to reduce noise.

Plots

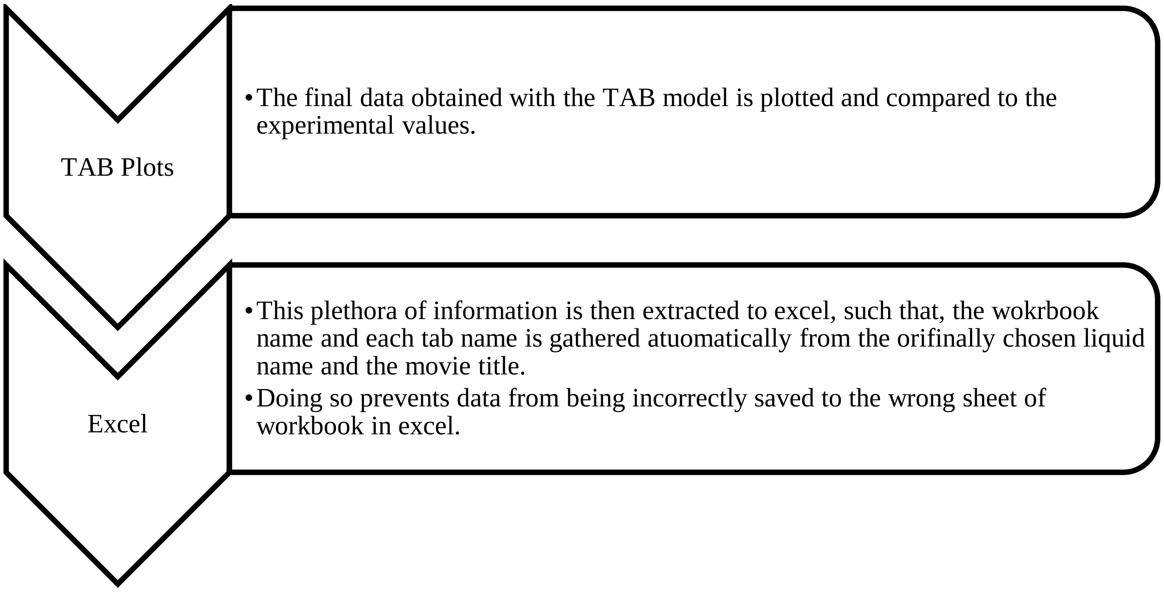
- The time varying information is then plotted to determine if there are any significant variations.
- Data plotted include displacement, velocity, acceleration, drop area, axial length, cross stream length, stamen length protruding, residuals from curve fitting velocity, and vertical velocity (to validate fps and length/pixel info).

Images

- To ensure that the correct time zero, initiation time, bag start time, bag breakup time, and stamen start time are being inputted by the user, the program creates a figure window with each of these instantaneous images from the video.

TAB Model

- The TAB model is ran except with varying a constant related to the coefficient of drag from 0 to 10 in increments of 0.01. Then the velocity profile from each run is compared to the experimental value by obtaining an R squared value. Finally, the constant with the maximum R squared value is taken as the final constant.
- After this, the TAB model is run again with the final constant and time varying velocity, cross stream length, coefficient of drag, and finally the initiation time is extracted.



Appendix B. MATLAB Program:

% Developed with MATLAB 2015b

```
clear % Clear all Variables.
clc
close all
prompt_folder= {'Are you ready to select the folder with the movies?
yes or no (input y or n)'};%Asks if user is ready to choose a folder.
title_folder = 'Select Folder';
answer_folder=inputdlg(prompt_folder,title_folder);
    if strcmp('y',answer_folder);% If user is ready, then he/she is
directed to the folder where the movies are.
        dname = uigetdir('E:\Current Research\Movies_cine\MATLAB
Videos\avi_0.8%CMC_7MF');% Must be changed to where the folder of
movies is
        addpath(dname);
        if strcmp('C:\',dname);
            msgbox('You did not select a folder. Please restart program
and try again','Error');
        end
    elseif (strcmp('n',answer_folder));
        msgbox('Program stoped, please know and have access to folder
where the movies are located and restart program','Error');
    else
        msgbox('Invalid entry please restart program and try
again','Error');
    end
    [FileName,PathName] = uigetfile('.avi','Please Select Movie
File',dname);% Allows the user to select the movie from the previously
chosen file. The movie must be in .avi format
    addpath(PathName);% Automatically makes the MATLAB path to the
folder
```

```
    Current_movie = FileName;
    Only_flow_rate=strtok(Current_movie,'_');
    mass_flow_rate = str2double(Only_flow_rate); %Obtain mass flowrate
by subtracting everything but the first numbers. My file formats were
MassFlowRate_Liquid_distance from nozzle. If the format is different
then this function will have an error.
% For mulitple movies use uifetfile('MultiSelect', 'on')
% for num_movies = 1:length(FileName);
%     Current_movie=FileName{num_movies};
    clc; % Clear the command windows
%clearvars -except FileName
% Load movie and get image
    warning('off','MATLAB:aviread:FunctionToBeRemoved');%Removes
unneccessary warnings
    warning('off','MATLAB:aviinfo:FunctionToBeRemoved');
    my_movie = VideoReader(Current_movie);
    lastFrame = read(my_movie, inf);
    numFrames = my_movie.NumberOfFrames;
    %movie_info = aviinfo(Current_movie);
    backg_im = read(my_movie,numFrames);
% Obtain Size of Rows and Columns
    size_image = size(backg_im);
    num_rows=size_image(1);
    num_columns=size_image(2);
    flow_behavior_index_n =0;
    flow_consistency_index_k=0;
    S = {'0.8%CMC-7MF','1.4%CMC-7MF','0.5%CMC-7HF','74.96%Gly/0.06%CMC-
7MF','79.95%Gly/0.06%CMC-7MF','84.96%Gly/0.05%CMC-7MF','DI-Water'};
    [Selection_liquid,ok] = listdlg('PromptString',{'Select a
Liquid','Movie
Title:',Current_movie},'SelectionMode','single','ListSize',[160
100],'ListString',S);
    if Selection_liquid==1; % 0.8%CMC-7MF
        Liquid = '0.8%CMC-7MF';
        density_drop = 995; % (kg/m^3)
        surface_tension = 0.0726; % (N/m)
        flow_behavior_index_n = 0.933; % (dimensionless)
        flow_consistency_index_k = 0.0464; % (Pa*s^n)
        dyn_viscosity_air = 0.0000184; % (Pa*s)
        density_air = 1.19; % (kg/m^3)
        frame_speed = 4796.163; % (frames/sec)
        calibration = 0.101; % (mm/sec)
        uncertainty_We = 7.0; % (percent)
        uncertainty_Oh = 3.8; % (percent)
        uncertainty_Re_L = 5.8; %(percent)
    elseif Selection_liquid==2; % 1.4%CMC-7MF
        Liquid = '1.4%CMC-7MF';
        density_drop = 998; % (kg/m^3)
        surface_tension = 0.0728; % (N/m)
        flow_behavior_index_n = 0.867; % (dimensionless)
        flow_consistency_index_k = 0.200; % (Pa*s^n)
        dyn_viscosity_air = 0.0000183; % (Pa*s)
        density_air = 1.19; % (kg/m^3)
        frame_speed = 6600.66; % (frames/sec)
        calibration = 0.101; % (mm/sec)
```

```
    uncertainty_We = 7.2; % (percent)
    uncertainty_Oh = 4.6; % (percent)
    uncertainty_Re_L = 6.5; %(percent)
elseif Selection_liquid==3; % 0.5%CMC-7HF
    Liquid = '0.5%CMC-7HF';
    density_drop = 999; % (kg/m^3)
    surface_tension = 0.0723; % (N/m)
    flow_behavior_index_n = 0.708; % (dimensionless)
    flow_consistency_index_k = 0.375; % (Pa*s^n)
    dyn_viscosity_air = 0.0000184; % (Pa*s)
    density_air = 1.19; % (kg/m^3)
    frame_speed = 6600.66; % (frames/sec)
    calibration = 0.101; % (mm/sec)
    uncertainty_We = 7.1; % (percent)
    uncertainty_Oh = 3.0; % (percent)
    uncertainty_Re_L = 5.5; %(percent)
elseif Selection_liquid==4; % 74.96%Gly/0.06%CMC-7MF
    Liquid = '74.96%Gly-0.06%CMC-7MF';
    density_drop = 1180; % (kg/m^3)
    surface_tension = 0.0661; % (N/m)
    flow_behavior_index_n = 0.899; % (dimensionless)
    flow_consistency_index_k = 0.128; % (Pa*s^n)
    dyn_viscosity_air = 0.0000184; % (Pa*s)
    density_air = 1.19; % (kg/m^3)
    frame_speed = 6600.66; % (frames/sec)
    calibration = 0.101; % (mm/sec)
    uncertainty_We = 7.2; % (percent)
    uncertainty_Oh = 4.0; % (percent)
    uncertainty_Re_L = 6.0; %(percent)
elseif Selection_liquid==5; % 79.95%Gly/0.06%CMC-7MF
    Liquid = '79.95%Gly-0.06%CMC-7MF';
    density_drop = 1200; % (kg/m^3)
    surface_tension = 0.0652; % (N/m)
    flow_behavior_index_n = 0.891; % (dimensionless)
    flow_consistency_index_k = 0.197; % (Pa*s^n)
    dyn_viscosity_air = 0.0000184; % (Pa*s)
    density_air = 1.19; % (kg/m^3)
    frame_speed = 6600.66; % (frames/sec)
    calibration = 0.101; % (mm/sec)
    uncertainty_We = 7.2; % (percent)
    uncertainty_Oh = 6.0; % (percent)
    uncertainty_Re_L = 7.5; %(percent)
elseif Selection_liquid==6; % 84.96%Gly/0.05%CMC-7MF
    Liquid = '84.96%Gly-0.05%CMC-7MF';
    density_drop = 1210; % (kg/m^3)
    surface_tension = 0.0649; % (N/m)
    flow_behavior_index_n = 0.884; % (dimensionless)
    flow_consistency_index_k = 0.304; % (Pa*s^n)
    dyn_viscosity_air = 0.0000184; % (Pa*s)
    density_air = 1.19; % (kg/m^3)
    frame_speed = 6600.66; % (frames/sec)
    calibration = 0.101; % (mm/sec)
    uncertainty_We = 7.2; % (percent)
    uncertainty_Oh = 8.5; % (percent)
    uncertainty_Re_L = 9.6; %(percent)
```

```
    elseif Selection_liquid==7; % DI-Water
        Liquid = 'DI-Water';
        density_drop = 998; % (kg/m^3)
        surface_tension = 0.0725; % (N/m)
        Viscosity = 0.000945; % (Pa*s)
        dyn_viscosity_air = 0.0000183; % (Pa*s)
        density_air = 1.19; % (kg/m^3)
        frame_speed = 6600.66; % (frames/sec)
        calibration = 0.073; % (mm/sec)
        uncertainty_We = 6.6; % (percent)
        uncertainty_Oh = 2.6; % (percent)
        uncertainty_Re_L = 4.6; %(percent)
    elseif ok==0;
        msgbox('Invalid selection please restart program and choose a
liquid','Error');
    end
% Obtain Information from Movie
    implay(read(my_movie));
    set(findall(0,'tag','spcui_scope_framework'),'position',[50 50 1200
600]);
    h=msgbox({'CAUTION: Press ok only when all information is
obtained.' 'Note: Frame number is in the bottom right. You will later
be prompted for this information' '' ...
              '1.What type of breakup is this?' '' '2. How many bags
develop in 2D and 3D? (2D: 0,1, or 2);(3D: 0-4)' '' '3. What frame
number does each bag start and stop?' 'Differentiate between the top
and bottom bags.' '(IE: Top Start: 45, Top End: 60, Bottom Start: 50,
Bottom End: 65)' ''...
              '4. Does a stamen protrude from the rim before the bag
breaks up? (y/n)' 'If yes, what frame number does it emerge from the
bag?' '' '5.Please find the frame number when only the rim is present
and the bag(s) have completely broken up'}, Current_movie,'help');
    waitfor(h);
    R = {'Vibrational Breakup','Bag Breakup','Bag-and-stamen
Breakup','Dual Bag Breakup'};
    [Selection_type_of_breakup,ok] = listdlg('PromptString','Select the
Type of Breakup:','SelectionMode','single','ListSize',[160
100],'ListString',R);
    if Selection_type_of_breakup==1;
        Type_of_breakup='Vibrational Breakup';
    elseif Selection_type_of_breakup==2;
        Type_of_breakup='Bag Breakup';
    elseif Selection_type_of_breakup==3;
        Type_of_breakup='Bag-and-stamen Breakup';
    elseif Selection_type_of_breakup==4;
        Type_of_breakup='Dual Bag Breakup';
    else
        msgbox('The selection is invalid. Please watch the video and
determine the breakup type','Error');
    end
% Initial Conditions
    stamen_present='n';
    num_bags=0;
    zero_bags_ends=0;
    only_one_bag_starts=0;
```

```
    only_one_bag_ends=0;
    one_bag_with_stamen_starts=0;
    one_bag_with_stamen_ends=0;
    stamen_with_one_bag_starts=0;
    first_bag_of_two_no_stamen_starts= 0;
    second_bag_of_two_no_stamen_starts= 0;
    first_bag_of_two_no_stamen_ends= 0;
    second_bag_of_two_no_stamen_ends= 0;
    first_bag_of_two_with_stamen_starts= 0;
    second_bag_of_two_with_stamen_starts= 0;
    stamen_with_two_bags_starts= 0;
    first_bag_of_two_with_stamen_ends= 0;
    second_bag_of_two_with_stamen_ends = 0;
    start_recording_stamen=0;
    num_bags_3D = 0;
    breakup_frame_3D_bag = 0;
    breakup_frame_3D_bag_2 = 0;
% Input Logic
% Number of Bags during Breakup and Stamen Protrusion
    num_bags_stamen_prompt= {'How many bags are produced during the
breakup in 2D? (0,1,2)','How many bags are produced during the breakup
in 3D?','Does a Stamen protrude from the rim before the bag
breaks?(y/n)'};
    num_lines=1;
    answer_num_bags_stamen
=inputdlg(num_bags_stamen_prompt,Current_movie,num_lines);
    if strcmp('n',answer_num_bags_stamen{3})==1 ||
strcmp('y',answer_num_bags_stamen{3})==1;
        stamen_present = answer_num_bags_stamen{3};
        if str2double(answer_num_bags_stamen{1})<3 &&
str2double(answer_num_bags_stamen{1})>=0;
            num_bags_3D = str2double(answer_num_bags_stamen{2});
            num_bags=str2double(answer_num_bags_stamen{1});
% For Zero Bags during Breakup
        if num_bags==0;
            zero_bags_prompt = {'Zero bags implies only 1 axial
dimension to be measured (vibrational breakup). Please input the frame
number where measurments are to end. (IE: 60)'};
            zero_bags_title = 'Frame Number to Stop Program';
            answer_zero_bags
=inputdlg(zero_bags_prompt,zero_bags_title,num_lines);
            if str2double(answer_zero_bags{1})<numFrames &&
(str2double(answer_zero_bags{1}))>0;
                zero_bags_ends= str2double(answer_zero_bags{1});
                end_frame = zero_bags_ends;
            else
                msgbox('The frame number is not within the number of
frames in the movie','Error');
            end
        end
% For One bag and no Stamen
        if num_bags==1 && strcmp('n',answer_num_bags_stamen{3})==1;
            one_bag_no_stamen_prompt = {'Please input the frame number
where the bag emerges. (IE: 45)','Input the frame number where the bag
breaks up. (IE: 60)'};
```

```
            one_bag_no_stamen_title = 'Frame Number to Stop Program';
            answer_one_bag_no_stamen
=inputdlg(one_bag_no_stamen_prompt,one_bag_no_stamen_title,num_lines);
        % Check if bag breaks up before it emerges
            if
str2double(answer_one_bag_no_stamen{1})>str2double(answer_one_bag_no_st
amen{2});
                 msgbox('Please check inputted values, the bag cannot
breakup before it emerges','Error');
            end
        % Get inpuuted values if they are within the number of frames
in the loaded movie
            if str2double(answer_one_bag_no_stamen{1})<numFrames &&
(str2double(answer_one_bag_no_stamen{1}))>0;
                only_one_bag_starts=
str2double(answer_one_bag_no_stamen{1});
                start_rim_data= only_one_bag_starts;
            else
                msgbox('The inputted frame number at which the bag
emerges not within the number of frames in the movie','Error');
            end
            if str2double(answer_one_bag_no_stamen{2})<numFrames &&
(str2double(answer_one_bag_no_stamen{2}))>0;
                only_one_bag_ends=
str2double(answer_one_bag_no_stamen{2});
                end_frame = only_one_bag_ends;
            else
                msgbox('The inputted frame number is at which the bag
breaks up is not within the number of frames in the movie','Error');
            end
        end
% For One bag and a stamen
        if num_bags==1 && strcmp('y',answer_num_bags_stamen{3})==1;
            one_bag_and_stamen_prompt = {'Please Input the frame number
where the bag emerges. (IE: 45)','Input the frame number where the bag
breaks up.(IE: 60)','Input the frame number where the stamen
emerges.(IE: 50)'};
            one_bag_and_stamen_title = 'Frame Number to Stop Program';
            answer_one_bag_and_stamen
=inputdlg(one_bag_and_stamen_prompt,one_bag_and_stamen_title,num_lines)
;
        % Check if bag breaks up before it emerges
            if
str2double(answer_one_bag_and_stamen{2})<str2double(answer_one_bag_and_
stamen{1});
                msgbox('Please check your input, the bag cannot breakup
before it emerges','Error');
            end
        % Assign values as long as the inputted values are within the
number of frames in loaded movie
            if str2double(answer_one_bag_and_stamen{1})<numFrames &&
(str2double(answer_one_bag_and_stamen{1}))>0;
                 one_bag_with_stamen_starts=
str2double(answer_one_bag_and_stamen{1});
                 start_rim_data= one_bag_with_stamen_starts;
```

```
            else
                msgbox('The frame number for bag breakup emerging is
not within the number of frames of the movie','Error');
            end
            if str2double(answer_one_bag_and_stamen{2})<numFrames &&
(str2double(answer_one_bag_and_stamen{2}))>0;
                one_bag_with_stamen_ends=
str2double(answer_one_bag_and_stamen{2});
                end_frame = one_bag_with_stamen_ends;
            else
                msgbox('The frame number for bag breakup ending is not
within the number of frames of the movie','Error');
            end
            if str2double(answer_one_bag_and_stamen{3})<numFrames &&
(str2double(answer_one_bag_and_stamen{3}))>0;
                stamen_with_one_bag_starts=
str2double(answer_one_bag_and_stamen{3});
                start_recording_stamen = stamen_with_one_bag_starts;
            else
                msgbox('The frame number for bag breakup ending is not
within the number of frames of the movie','Error');
            end
        end
% Two bags and no Stamen
        if num_bags==2 && strcmp('n',answer_num_bags_stamen{3})==1;
            two_bags_no_stamen_prompt = {'Please input the frame number
where the top bag emerges. (IE: 45)','Input the frame number where the
top bag breaks up. (IE: 60)','Input the frame number where the bottom
bag emerges. (IE: 50)','Input the frame number where the bottom bag
breaks up. (IE: 65)'};
            two_bags_no_stamen_title = 'Frame Number to Stop Program';
            answer_two_bags_no_stamen
=inputdlg(two_bags_no_stamen_prompt,two_bags_no_stamen_title,num_lines)
;
        % Check if breakup occurs before bag emerges
            if
str2double(answer_two_bags_no_stamen{2})<str2double(answer_two_bags_no_
stamen{1});
                msgbox('Please check your input because the first bag
cannot breakup before it emerges','Error');
            end
            if
str2double(answer_two_bags_no_stamen{4})<str2double(answer_two_bags_no_
stamen{3});
                msgbox('Please check your input because the second bag
cannot breakup before it emerges','Error');
            end
        % Assign values as long as the inputted values are within the
number of frames in loaded movie
            if str2double(answer_two_bags_no_stamen{1})<numFrames &&
(str2double(answer_two_bags_no_stamen{1}))>0;
                first_bag_of_two_no_stamen_starts=
str2double(answer_two_bags_no_stamen{1});
            else
```

```
                msgbox('The frame number for the first bag starting is
not within the number of frames of the movie','Error');
            end
            if str2double(answer_two_bags_no_stamen{2})<numFrames &&
(str2double(answer_two_bags_no_stamen{2}))>0;
                first_bag_of_two_no_stamen_ends=
str2double(answer_two_bags_no_stamen{2});
            else
                msgbox('The frame number for the first bag breaking up
is not within the number of frames of the movie','Error');
            end
            if str2double(answer_two_bags_no_stamen{3})<numFrames &&
(str2double(answer_two_bags_no_stamen{3}))>0;
                second_bag_of_two_no_stamen_starts=
str2double(answer_two_bags_no_stamen{3});
            else
                msgbox('The frame number for the second bag starting is
not within the number of frames of the movie','Error');
            end
            if str2double(answer_two_bags_no_stamen{4})<numFrames &&
(str2double(answer_two_bags_no_stamen{4}))>0;
                second_bag_of_two_no_stamen_ends=
str2double(answer_two_bags_no_stamen{4});
            else
                msgbox('The frame number for the second bag breaking up
is not within the number of frames of the movie','Error');
            end
            if
first_bag_of_two_no_stamen_starts<=second_bag_of_two_no_stamen_starts;
                start_rim_data= first_bag_of_two_no_stamen_starts;
            end
            if
first_bag_of_two_no_stamen_starts>second_bag_of_two_no_stamen_starts;
                start_rim_data= second_bag_of_two_no_stamen_starts;
            end
            if
first_bag_of_two_no_stamen_ends<=second_bag_of_two_no_stamen_ends;
                end_frame = second_bag_of_two_no_stamen_ends;
            end
            if
first_bag_of_two_no_stamen_ends>second_bag_of_two_no_stamen_ends;
                end_frame = first_bag_of_two_no_stamen_ends;
            end
        end
% Two bags and a Stamen
        if num_bags==2 && strcmp('y',answer_num_bags_stamen{3})==1;
            two_bags_and_stamen_prompt = {'Please input the frame
number where the top bag emerges. (IE: 45)','Input the frame number
where the top bag breaks up. (IE: 60)','Input the frame number where
the bottom bag emerges. (IE: 50)','Input the frame number where the
bottom bag breaks up. (IE: 65)','Input the frame number where the
stamen emerges. (IE: 48)'};
            two_bags_and_stamen_title = 'Frame Number to Stop Program';
```

```
            answer_two_bags_and_stamen
=inputdlg(two_bags_and_stamen_prompt,two_bags_and_stamen_title,num_line
s);
        % Check if bags breakup before they emerge
            if
str2double(answer_two_bags_and_stamen{2})<str2double(answer_two_bags_an
d_stamen{1});
                msgbox('Please check your input because the first bag
cannot breakup before it emerges','Error');
            end
            if
str2double(answer_two_bags_and_stamen{4})<str2double(answer_two_bags_an
d_stamen{3});
                msgbox('Please check your input because the second bag
cannot breakup before it emerges','Error');
            end
        % Apply values if the inputted numbers are within the number of
frames in the movie
            if str2double(answer_two_bags_and_stamen{1})<numFrames &&
(str2double(answer_two_bags_and_stamen{1}))>0;
                first_bag_of_two_with_stamen_starts=
str2double(answer_two_bags_and_stamen{1});
            else
                msgbox('The frame number for the first bag emerging is
not within the number of frames of the movie','Error');
            end
            if str2double(answer_two_bags_and_stamen{2})<numFrames &&
(str2double(answer_two_bags_and_stamen{2}))>0;
                first_bag_of_two_with_stamen_ends=
str2double(answer_two_bags_and_stamen{2});
            else
                msgbox('The frame number for the first bag breaking up
is not within the number of frames of the movie','Error');
            end
            if str2double(answer_two_bags_and_stamen{3})<numFrames &&
(str2double(answer_two_bags_and_stamen{3}))>0;
                second_bag_of_two_with_stamen_starts=
str2double(answer_two_bags_and_stamen{3});
            else
                msgbox('The frame number for the second bag emerging is
not within the number of frames of the movie','Error');
            end
            if str2double(answer_two_bags_and_stamen{4})<numFrames &&
(str2double(answer_two_bags_and_stamen{4}))>0;
                second_bag_of_two_with_stamen_ends=
str2double(answer_two_bags_and_stamen{4});
            else
                msgbox('The frame number for the second bag breaking is
not within the number of frames of the movie','Error');
            end
            if str2double(answer_two_bags_and_stamen{5})<numFrames &&
(str2double(answer_two_bags_and_stamen{5}))>0;
                stamen_with_two_bags_starts=
str2double(answer_two_bags_and_stamen{5});
                start_recording_stamen = stamen_with_two_bags_starts;
```

```
            else
                msgbox('The frame number for the stamen emerging is not
within the number of frames of the movie','Error');
            end
            if
first_bag_of_two_with_stamen_starts<=second_bag_of_two_with_stamen_star
ts;
                start_rim_data= first_bag_of_two_with_stamen_starts;
            end
            if
first_bag_of_two_with_stamen_starts>second_bag_of_two_with_stamen_start
s;
                start_rim_data= second_bag_of_two_with_stamen_starts;
            end
            if
first_bag_of_two_with_stamen_ends<=second_bag_of_two_with_stamen_ends;
                end_frame = second_bag_of_two_with_stamen_ends;
            end
            if
first_bag_of_two_with_stamen_ends>second_bag_of_two_with_stamen_ends;
                end_frame =first_bag_of_two_with_stamen_ends;
            end
        end
        if num_bags>0 && num_bags<3;
            end_bag_breakup_prompt = {'Please input the frame number
where only the rim is present and the bag(s) has completely broken
up'};
            end_bag_breakup_title = 'Frame Number to Stop Program';
            answer_end_bag_breakup
=inputdlg(end_bag_breakup_prompt,end_bag_breakup_title,num_lines);
            if str2double(answer_end_bag_breakup{1})<numFrames &&
(str2double(answer_end_bag_breakup{1}))>0;
                frame_num_only_rim =
str2double(answer_end_bag_breakup{1});
            end
            if num_bags ~= num_bags_3D;
                if num_bags_3D==2 && num_bags==1;
                    breakup_time_3D_bag_prompt_2_1 = {'Please input the
frame number where the other bag in 3D breaks up'};
                    breakup_time_3D_bag_title_2_1 = '3D Bag Times';
                    answer_breakup_time_3D_bag_2_1
=inputdlg(breakup_time_3D_bag_prompt_2_1,breakup_time_3D_bag_title_2_1,
num_lines);
                    if
str2double(answer_breakup_time_3D_bag_2_1{1})<numFrames &&
(str2double(answer_breakup_time_3D_bag_2_1{1}))>0;
                        breakup_frame_3D_bag =
str2double(answer_breakup_time_3D_bag_2_1{1});
                    else
                        msgbox('The frame number entered is not within
the number of movies frames. Please Restart Program and Try
Again','Error');
                    end
                end
                if num_bags_3D==3 && num_bags==1;
```

```
                    breakup_time_3D_bag_prompt_3_1 = {'Please input the
frame number where one of the other bags in 3D breaks up','Please input
the frame number where the last of the 3D bags breaks up'};
                    breakup_time_3D_bag_title_3_1 = '3D Bag Times';
                    answer_breakup_time_3D_bag_3_1
=inputdlg(breakup_time_3D_bag_prompt_3_1,breakup_time_3D_bag_title_3_1,
num_lines);
                    if
str2double(answer_breakup_time_3D_bag_3_1{1})<numFrames &&
(str2double(answer_breakup_time_3D_bag_3_1{1}))>0;
                        breakup_frame_3D_bag =
str2double(answer_breakup_time_3D_bag_3_1{1});
                    else
                        msgbox('The frame number entered is not within
the number of movies frames. Please Restart Program and Try
Again','Error');
                    end
                    if
str2double(answer_breakup_time_3D_bag_3_1{2})<numFrames &&
(str2double(answer_breakup_time_3D_bag_3_1{2}))>0;
                        breakup_frame_3D_bag_2 =
str2double(answer_breakup_time_3D_bag_3_1{2});
                    else
                        msgbox('The frame number entered is not within
the number of movies frames. Please Restart Program and Try
Again','Error');
                    end
                end
                if num_bags_3D==3 && num_bags==2;
                    breakup_time_3D_bag_prompt_3_2 = {'Please input the
frame number where the last of the 3D bags breaks up'};
                    breakup_time_3D_bag_title_3_2 = '3D Bag Times';
                    answer_breakup_time_3D_bag_3_2
=inputdlg(breakup_time_3D_bag_prompt_3_2,breakup_time_3D_bag_title_3_2,
num_lines);
                    if
str2double(answer_breakup_time_3D_bag_3_2{1})<numFrames &&
(str2double(answer_breakup_time_3D_bag_3_2{1}))>0;
                        breakup_frame_3D_bag =
str2double(answer_breakup_time_3D_bag_3_2{1});
                    else
                        msgbox('The frame number entered is not within
the number of movies frames. Please Restart Program and Try
Again','Error');
                    end
                end
            end
        end
        else
            msgbox('This is not a valid entry for the number of bags in
2D. The number of bag which this program can process is between 0 and
2.','Error');
        end
    else
```

```
        msgbox('The is not a valid input for stamen protrusion. Answer
either yes(y) or no(n)','Error');
    end
%%
% Preallocate Matrix Rows and Columns
    close all
    Frame_Number = zeros(end_frame-1,1);
    Drop_info_centroid_axial = zeros(end_frame-1,1);
    Drop_info_centroid_transverse = zeros(end_frame-1,1);
    Drop_info_Area = zeros(end_frame-1,1);
    Image_info_num_objects  = zeros (end_frame-1, 1);
    frame_num_from_time_zero= zeros(end_frame-1,1);
    peak_rows_and_location=zeros(end_frame-1,2);
    peak_columns_and_location=zeros(end_frame-1,5);
    num_non_zero_pixels_each_row = zeros(num_rows,1);
    num_non_zero_pixels_each_column = zeros(1,num_columns);
    stamen_length_protruding=zeros(end_frame-1,1);
    peak_rows_and_location_top = zeros(end_frame-1,2);
    peak_rows_and_location_bottom = zeros(end_frame-1,2);
    noise_fill_bw_im_contrast_no_stamen = zeros(num_rows,num_columns);
    top_left_point = zeros(end_frame-1,2);
    bottom_left_point = zeros(end_frame-1,2);
    min_point = zeros(end_frame-1,2);
    max_point = zeros(end_frame-1,2);
    percent_bag_to_keep_iteration = zeros(end_frame-1,1);
    stamen_length_protruding_from_image = zeros(end_frame-1,1);
    axial_centroid_based_on_basal_ring = zeros(end_frame-1,1);
    transverse_centroid_based_on_basal_ring = zeros(end_frame-1,1);
    volume_ratio = zeros(end_frame-1,1);
    ratio = zeros(end_frame-1,1);
    orientation = zeros(end_frame-1,1);
    top_area_image = zeros(num_rows,num_columns);
    bottom_area_image = zeros(num_rows,num_columns);
    surface_area_ratio = zeros(end_frame-1,1);
    E = zeros(end_frame-1,1);
    bounding_box_1 = zeros(end_frame-1,1);
    bounding_box_2 = zeros(end_frame-1,1);
    bounding_box_3 = zeros(end_frame-1,1);
    bounding_box_4 = zeros(end_frame-1,1);
    scnsize = get(0,'Screensize');
    figure(1)
    hFig = figure(1);
    set(hFig,'Position',[1 -scnsize(2) scnsize(3) scnsize(4)]);
    reference_figure = getframe(gcf);
    size_figure=size(reference_figure.cdata);
    allTheFrames = cell(1,end_frame);
    allTheFrames(:) = {zeros(size_figure(1),size_figure(2), 3,
'uint8')};
    allTheColorMaps = cell(1,end_frame);
    final_movie = struct('cdata', allTheFrames, 'colormap',
allTheColorMaps);
% Initial Conditions for do while loop
    end_check_noise_removal = 0;
    end_check_structered_element_size=0;
    percent_stamen_end_check=0;
```

```
% Do while Loop: Iteration depends whether the program has obtained the
data properly
    end_check_complete='not complete';
    while (strcmp('not complete',end_check_complete));
% Create Loop to Process All Images in Video
    % Initial Condition for Ratio in for loop
        stop_computing_ratio='no';
        tic
        for i = 1:end_frame;
            drop_im_1 = read(my_movie,i);
        % Turn into Black and white image
            im_contrast = backg_im - drop_im_1;
            max_im_contrast_10_percent = 0.15*max(im_contrast(:));
            for o = 1:num_columns;
                for p = 1:num_rows;
                    if (im_contrast(p,o)<max_im_contrast_10_percent);
                        im_contrast(p,o)=0;
                    end
                end
            end
        % Detect Edges
            bw_im_contrast = edge(im_contrast,'sobel');
        % Initial condition for iteration of noise removal
            iterate_noise_removal = 0;
            Image_info_num_objects(i,1) = 2;
            u=0;
            structured_element_size=0;
            removed_area =0;
% Remove Noise
        % Do while Loop: Iteration depends on the number of objects
detected
            while (Image_info_num_objects (i,1)>1);
            % Noise Removal: done by removing a certain number of
connected components
                removed_area = 4 + iterate_noise_removal +
end_check_noise_removal;
                noise_bw_im_contrast = bwareaopen(bw_im_contrast,
removed_area);
            % Structured element defined
                if strcmp(stop_computing_ratio,'no');
                    structure_size_initial = 5;
                elseif
strcmp(stop_computing_ratio,'yes')&&i<start_rim_data;
                    structure_size_initial = 15;
                else
                    structure_size_initial = 15;
                end
                structured_element_size = structure_size_initial +
end_check_structered_element_size;
                structured_element =  strel('disk',
structured_element_size);
            % Dilation using the structured element
                pre_pre_pre =
imdilate(noise_bw_im_contrast,structured_element);
            % Errosion using the structured element
```

```
                pre_pre_morph=imerode(pre_pre_pre,structured_element);
            % Filling the closed boundary with white pixels
                noise_fill_bw_im_contrast =
imfill(pre_pre_morph,'holes');
            % Obtain the number of detected objects in image
                num_obj_noise_fill_bw_im_contrast = bwconncomp
(noise_fill_bw_im_contrast, 4);
                Image_info_num_objects (i,1) =
num_obj_noise_fill_bw_im_contrast.NumObjects;
                iterate_noise_removal= 1 + iterate_noise_removal;
            end
            for e=1:num_rows;
                if (noise_fill_bw_im_contrast(e,1))>0;
                    u=1+u;
                    break;
                end
            end
            for e=1:num_columns;
                if (noise_fill_bw_im_contrast(1,e))>0;
                    u=1+u;
                    break;
                end
            end
            if u>0;
                msgbox('The Structured Element Size is Too Large and is
Causing Inclusion of the Boundaries Please Reduce the Size and Restart
Program','Error');
                break
            end
            end_frame_two_bags_no_stamen=0;
            end_frame_two_bags_with_stamen=0;
        % Output final frame and when ratio is 1.1
            if i==end_frame;
                figure(2);
                hFig = figure(2);
                set(hFig, 'Position', get(0,'Screensize'));
                subplot(1,2,1),imshow(drop_im_1),...
                h=title (['Frame#:', num2str(Frame_Number(i-1,1)+1), '
of file: ', Current_movie, '  No Data Obtained From This
Frame']);set(h,'FontSize',8, 'FontWeight','bold', 'Interpreter',
'none');
                subplot (1,2,2),
imshow(read(my_movie,time_zero_frame_num)),s=title('Time Zero
Image');set(s,'FontSize',8, 'FontWeight','bold', 'Interpreter',
'none');
                break;
            end
        % Remove Stamen and Record value
        % Start measurement if condition is satisfied
            im_complement = imcomplement(noise_fill_bw_im_contrast);
            percent_stamen_removed = 0.5 + percent_stamen_end_check;
            percent_node_removed = 0.5;
            percent_bag_to_keep = 0.1;
            percent_column_to_compare =1.00;
```

```
%/////////////////////////////////////////////////////////////////////////
//////////////////////////////////////////////////////////////////////
% Remove stamen
            if strcmp('y',stamen_present) && start_recording_stamen<=i;
                g=0;
                image_no_stamen = noise_fill_bw_im_contrast;
                for c=1:num_columns
                    column_vector = im_complement(1:num_rows,c);
                    dist_from_non_zero= bwdist(column_vector);
                % Remove column if there is only one point is present
                    find_how_many_dist_are_one =
find(dist_from_non_zero==1);
                    if nnz(find_how_many_dist_are_one)==1;
                        stamen_length_protruding_from_image(i,1) =
1+stamen_length_protruding_from_image(i,1);
                        image_no_stamen(1:num_rows,c)=0;
                % If computer reads 2 points then remove if the height
of those points is less than a percent of the previous column and exit
the while
                % loop if the percent is greater than some percent of
the previous column
                    elseif nnz(find_how_many_dist_are_one)==2;
                        if g>0;
                            break
                        end
                        if
nnz(noise_fill_bw_im_contrast(1:num_rows,c))<percent_stamen_removed*pea
k_columns_and_location(i-1,1);
                            image_no_stamen(1:num_rows,c)=0;
                            stamen_length_protruding_from_image(i,1) =
1+stamen_length_protruding_from_image(i,1);
                        else
                            break;
                        end
                % Remove column if the number of points is 3
                    elseif nnz(find_how_many_dist_are_one)==3;
                        image_no_stamen(1:num_rows,c)=0;
                        stamen_length_protruding_from_image(i,1) =
1+stamen_length_protruding_from_image(i,1);
                % Remove the first set of points when the computer
reads 4 points along a column
                    elseif nnz(find_how_many_dist_are_one)==4;

first_point_to_remove=find_how_many_dist_are_one(1);

second_point_to_remove=find_how_many_dist_are_one(2);
                        stamen_length_protruding_from_image(i,1) =
1+stamen_length_protruding_from_image(i,1);

image_no_stamen(first_point_to_remove:second_point_to_remove,c)=0;
                        g=1+g;
                    elseif nnz(find_how_many_dist_are_one)==5;
                        image_no_stamen(1:num_rows,c)=0;
                        stamen_length_protruding_from_image(i,1) =
1+stamen_length_protruding_from_image(i,1);
```

```
                    elseif nnz(find_how_many_dist_are_one)==6;

first_point_to_remove=find_how_many_dist_are_one(3);

second_point_to_remove=find_how_many_dist_are_one(4);
                        stamen_length_protruding_from_image(i,1) =
1+stamen_length_protruding_from_image(i,1);

image_no_stamen(first_point_to_remove:second_point_to_remove,c)=0;
                        g=1+g;
                    end
                end
                structured_element_for_stamen =  strel('disk',5);
            % Dilation using the structured element
                image_no_stamen =
imdilate(image_no_stamen,structured_element_for_stamen);
            % Errosion using the structured element
                image_no_stamen =
imerode(image_no_stamen,structured_element_for_stamen);
% Remove Bag from removed Stamen
                stats_no_stamen = regionprops(image_no_stamen,
'Centroid','Extrema','BoundingBox');
                Axial_centroid = stats_no_stamen.Centroid(1);
                Transverse_centroid = stats_no_stamen.Centroid(2);
                if start_rim_data<=i;
                        num_non_zero_pixels_in_no_stamen_image =
zeros(num_columns,1);
                    for t=1:num_columns;
                        num_non_zero_pixels_in_no_stamen_image(t,1) =
nnz(image_no_stamen(1:num_rows,t));
                    end
                % Obtain first and last point along centroid without
stamen
                    find_first_in_image =
find(num_non_zero_pixels_in_no_stamen_image,1,'first');
                    find_last_in_image =
find(num_non_zero_pixels_in_no_stamen_image,1,'last');
                    percent_bag_to_keep_iteration(i,1) = 0;
                    peak_columns_and_location(i,1)=0;
                    while
(peak_columns_and_location(i,1)<percent_column_to_compare*peak_columns_
and_location(i-1,1));
                        Image_info_num_objects_no_stamen_no_bag=2;
                        while
(Image_info_num_objects_no_stamen_no_bag>1);
                            point_start_removing =
find_first_in_image+round((percent_bag_to_keep +
percent_bag_to_keep_iteration(i,1))*(find_last_in_image-
find_first_in_image));
                            image_no_stamen_no_bag = image_no_stamen;

image_no_stamen_no_bag(1:num_rows,point_start_removing:num_columns) =
0;
                            image_no_stamen_no_bag =
bwareaopen(image_no_stamen_no_bag, 4);
```

```
                            num_obj_no_stamen_no_bag =
bwconncomp(image_no_stamen_no_bag, 4);

Image_info_num_objects_no_stamen_no_bag=num_obj_no_stamen_no_bag.NumObj
ects;
                            percent_bag_to_keep_iteration(i,1) =
percent_bag_to_keep_iteration(i,1)+0.01;
                            if percent_bag_to_keep_iteration(i,1)>0.9;
                                break
                            end
                        end
                        if percent_bag_to_keep_iteration(i,1)>0.9;
                            break
                        end
                % Column length when stamen is present, bag has
emerged, stamen has emerged, and ratio greater than 1.1
                            stats_no_stamen_no_bag =
regionprops(image_no_stamen_no_bag,'Extrema','Centroid','BoundingBox');
                            top_left_point(i,1)
=((stats_no_stamen_no_bag.Extrema(1,2)+stats_no_stamen_no_bag.Extrema(2
,2))/2)+0.5;
                            top_left_point(i,2)
=((stats_no_stamen_no_bag.Extrema(1,1)+stats_no_stamen_no_bag.Extrema(2
,1))/2);
                            %bottom_left_point(i,1)
=((stats_no_stamen_no_bag.Extrema(5,2)+stats_no_stamen_no_bag.Extrema(6
,2))/2)-0.5;
                            %bottom_left_point(i,2)
=((stats_no_stamen_no_bag.Extrema(5,1)+stats_no_stamen_no_bag.Extrema(6
,1))/2);
                            Boundary =
bwboundaries(image_no_stamen_no_bag);
                            Data_rows = Boundary{1}(:,1);
                            Data_columns = Boundary{1}(:,2);
                            data_add = Data_rows + Data_columns;
                            data_subtract = Data_rows - Data_columns;
                            inverse_add = 1.00*max(data_add)-data_add;
                            max_inv_add = max(inverse_add);
                            max_subtract = max(data_subtract);
                            [top_left_max,top_left_index] =
findpeaks(inverse_add,'MINPEAKHEIGHT',max_inv_add-1, 'NPEAKS',1);
                            [bottom_left_max,bottom_left_index] =
findpeaks(data_subtract,'MINPEAKHEIGHT',max_subtract-1,'NPEAKS',1);
                            if isempty(top_left_max);
                                top_left_index =
find(inverse_add>(max_inv_add-1),1,'first');
                            end
                            if isempty(bottom_left_index);
                                bottom_left_index =
find(data_subtract>(max_subtract-1),1,'last');
                            end
                            %top_left_point (i,1) =
Data_rows(top_left_index,1);
                            %top_left_point (i,2) =
Data_columns(top_left_index,1);
```

```
                            bottom_left_point (i,1) =
Data_rows(bottom_left_index,1);
                            bottom_left_point (i,2) =
Data_columns(bottom_left_index,1);
                            peak_columns_and_location(i,1) =
sqrt((top_left_point(i,1)-bottom_left_point(i,1))^2+(top_left_point
(i,2)-bottom_left_point (i,2))^2)+1;
                    end
                end
            end
% Get column/length of rim
        % Measure Properties of Connected Objects
            stats = regionprops(noise_fill_bw_im_contrast, 'all');
        % Column length when ratio is less than 1.1 (still spherical)
for all cases.
            if strcmp(stop_computing_ratio,'no');
                for t=1:num_columns;
                    num_non_zero_pixels_each_column(t,1) =
nnz(noise_fill_bw_im_contrast(1:num_rows,t));
                end

max_column_initial=max(num_non_zero_pixels_each_column(1:num_rows,1));
                [peak_columns,location_peak_columns] =
findpeaks(num_non_zero_pixels_each_column(1:num_columns,1),'MINPEAKHEIG
HT',max_column_initial-1,'NPEAKS',1);
                find_first_in_peak_column =
find(noise_fill_bw_im_contrast(1:num_rows,location_peak_columns),1,'fir
st');
                find_last_in_peak_column =
find(noise_fill_bw_im_contrast(1:num_rows,location_peak_columns),1,'las
t');
                top_left_point(i,1)= find_first_in_peak_column;
                top_left_point(i,2)= location_peak_columns;
                bottom_left_point (i,1)= find_last_in_peak_column;
                bottom_left_point (i,2)= location_peak_columns;
            end
        % Column length when the bag hasnt emerged, ratio is greater
than 1.1 (no longer spherical), and no stamen
            if (i<start_rim_data && strcmp(stop_computing_ratio,'yes')
&& strcmp('n',stamen_present));
                top_left_point(i,1)
=((stats.Extrema(1,2)+stats.Extrema(2,2))/2)+0.5;
                top_left_point(i,2)
=((stats.Extrema(1,1)+stats.Extrema(2,1))/2);
                bottom_left_point(i,1)
=((stats.Extrema(5,2)+stats.Extrema(6,2))/2)-0.5;
                bottom_left_point(i,2)
=((stats.Extrema(5,1)+stats.Extrema(6,1))/2);
            end
        % Column length when the bag hasnt emerged, ratio is greater
than 1.1, and stamen hasnt emerged
            if (i<start_rim_data && strcmp(stop_computing_ratio,'yes')
&& start_recording_stamen>i);
                top_left_point(i,1)
=((stats.Extrema(1,2)+stats.Extrema(2,2))/2)+0.5;
```

```
                top_left_point(i,2)
=((stats.Extrema(1,1)+stats.Extrema(2,1))/2);
                bottom_left_point(i,1)
=((stats.Extrema(5,2)+stats.Extrema(6,2))/2)-0.5;
                bottom_left_point(i,2)
=((stats.Extrema(5,1)+stats.Extrema(6,1))/2);
            end
        % Column length when there is a stamen present, stamen has
protruded, ratio is greater than 1.1, and bag hasnt emerged
            if strcmp('y',stamen_present) && start_recording_stamen<=i
&& strcmp(stop_computing_ratio,'yes') && i<start_rim_data;
                top_left_point(i,1)
=((stats_no_stamen.Extrema(1,2)+stats_no_stamen.Extrema(2,2))/2)+0.5;
                top_left_point(i,2)
=((stats_no_stamen.Extrema(1,1)+stats_no_stamen.Extrema(2,1))/2);
                bottom_left_point(i,1)
=((stats_no_stamen.Extrema(5,2)+stats_no_stamen.Extrema(6,2))/2)-0.5;
                bottom_left_point(i,2)
=((stats_no_stamen.Extrema(5,1)+stats_no_stamen.Extrema(6,1))/2);
                peak_columns_and_location(i,1) =
sqrt((top_left_point(i,1)-bottom_left_point(i,1))^2+(top_left_point
(i,2)-bottom_left_point (i,2))^2)+1;
            end
        % Remove nodes from original image
            if i>=start_rim_data;
                noise_fill_bw_im_contrast_no_nodes =
noise_fill_bw_im_contrast;
                for c=1:num_columns
                    num_non_zero_no_nodes =
nnz(noise_fill_bw_im_contrast_no_nodes(1:num_rows,c));
                    if
num_non_zero_no_nodes<=(percent_node_removed*peak_columns_and_location(
i-1,1));

noise_fill_bw_im_contrast_no_nodes(1:num_rows,c)=0;
                    else
                        break;
                    end
                end
            end
            if strcmp('y',stamen_present) && start_recording_stamen<=i
&& i<start_rim_data;
                noise_fill_bw_im_contrast_no_nodes =
noise_fill_bw_im_contrast;
                for c=1:num_columns
                    num_non_zero_no_nodes =
nnz(noise_fill_bw_im_contrast_no_nodes(1:num_rows,c));
                    if
num_non_zero_no_nodes<=(percent_node_removed*peak_columns_and_location(
i-1,1));

noise_fill_bw_im_contrast_no_nodes(1:num_rows,c)=0;
                    else
                        break;
                    end
```

```
                end
            end
        % Column legnth when bag has emerged and no stamen is present
            if i>=start_rim_data && strcmp('n',stamen_present);
                Boundary =
bwboundaries(noise_fill_bw_im_contrast_no_nodes);
                Data_rows = Boundary{1}(:,1);
                Data_columns = Boundary{1}(:,2);
                data_add = Data_rows + Data_columns;
                data_subtract = Data_rows - Data_columns;
                inverse_add = 1.00*max(data_add)-data_add;
                max_inv_add = max(inverse_add);
                max_subtract = max(data_subtract);
                [top_left_max,top_left_index] =
findpeaks(inverse_add,'MINPEAKHEIGHT',max_inv_add-1, 'NPEAKS',1);
                [bottom_left_max,bottom_left_index] =
findpeaks(data_subtract,'MINPEAKHEIGHT',max_subtract-1,'NPEAKS',1);
                if isempty(top_left_max);
                    top_left_index = find(inverse_add>(max_inv_add-
1),1,'first');
                end
                if isempty(bottom_left_index);
                    bottom_left_index =
find(data_subtract>(max_subtract-1),1,'last');
                end
                top_left_point (i,1) = Data_rows(top_left_index,1);
                top_left_point (i,2) = Data_columns(top_left_index,1);
                bottom_left_point (i,1) =
Data_rows(bottom_left_index,1);
                bottom_left_point (i,2) =
Data_columns(bottom_left_index,1);
            end
        % Column length when stamen is present, bag has emerged, stamen
hasnt emerged, and ratio greater than 1.1
            if strcmp('y',stamen_present) && i>=start_rim_data &&
start_recording_stamen>i && strcmp(stop_computing_ratio,'yes');
                Boundary =
bwboundaries(noise_fill_bw_im_contrast_no_nodes);
                Data_rows = Boundary{1}(:,1);
                Data_columns = Boundary{1}(:,2);
                data_add = Data_rows + Data_columns;
                data_subtract = Data_rows - Data_columns;
                inverse_add = 1.00*max(data_add)-data_add;
                max_inv_add = max(inverse_add);
                max_subtract = max(data_subtract);
                [top_left_max,top_left_index] =
findpeaks(inverse_add,'MINPEAKHEIGHT',max_inv_add-1, 'NPEAKS',1);
                [bottom_left_max,bottom_left_index] =
findpeaks(data_subtract,'MINPEAKHEIGHT',max_subtract-1,'NPEAKS',1);
                if isempty(top_left_max);
                    top_left_index = find(inverse_add>(max_inv_add-
1),1,'first');
                end
                if isempty(bottom_left_index);
```

```
                    bottom_left_index =
find(data_subtract>(max_subtract-1),1,'last');
                end
                top_left_point (i,1) = Data_rows(top_left_index,1);
                top_left_point (i,2) = Data_columns(top_left_index,1);
                bottom_left_point (i,1) =
Data_rows(bottom_left_index,1);
                bottom_left_point (i,2) =
Data_columns(bottom_left_index,1);
            end
            axial_centroid_based_on_basal_ring(i,1) = (top_left_point
(i,2)+bottom_left_point (i,2))/2;
            transverse_centroid_based_on_basal_ring(i,1) =
(top_left_point (i,1)+bottom_left_point (i,1))/2;
            peak_columns_and_location(i,1) = sqrt((top_left_point(i,1)-
bottom_left_point(i,1))^2+(top_left_point (i,2)-bottom_left_point
(i,2))^2)+1;
            peak_columns_and_location(i,2) = top_left_point (i,1);
            peak_columns_and_location(i,3) = top_left_point (i,2);
            peak_columns_and_location(i,4) = bottom_left_point (i,1);
            peak_columns_and_location(i,5) = bottom_left_point (i,2);
% Get Row Data/bag length
        % Bag hasnt emerged and no stamen
            if i<start_rim_data && strcmp('n',stamen_present);
                reference_image = noise_fill_bw_im_contrast;
            end
        % Bag emerged and no stamen
            if i>=start_rim_data && strcmp('n',stamen_present);
                reference_image = noise_fill_bw_im_contrast_no_nodes;
            end
        % Bag nor stamen has emerged
            if i<start_rim_data && strcmp('y',stamen_present) &&
start_recording_stamen>i;
                reference_image = noise_fill_bw_im_contrast;
            end
        % Bag hasnt emerged, but stamen has
            if i<start_rim_data && strcmp('y',stamen_present) &&
start_recording_stamen<=i;
                reference_image=noise_fill_bw_im_contrast_no_nodes;
                %for c=1:num_columns;
                %if
nnz(noise_fill_bw_im_contrast(1:num_rows,c))<percent_stamen_removed*pea
k_columns_and_location(i-1,1);
                %    reference_image(1:num_rows,c)=0;
                %    if
nnz(noise_fill_bw_im_contrast(1:num_rows,c))>=1;
                 %       stamen_length_protruding_from_image(i,1) =
1+stamen_length_protruding_from_image(i,1);
                  %  end
                %else
                 %   break
                %end
                %end
                            %reference_image = image_no_stamen;
            end
```

```
        % Bag has emerged but stamen hasnt
            if i>=start_rim_data && strcmp('y',stamen_present) &&
start_recording_stamen>i;
                reference_image = noise_fill_bw_im_contrast_no_nodes;
            end
        % Bag-and-stamen has emerged
            if i>=start_rim_data && strcmp('y',stamen_present) &&
start_recording_stamen<=i;
                reference_image=noise_fill_bw_im_contrast_no_nodes;
                %for c=1:num_columns;
                %if
nnz(noise_fill_bw_im_contrast(1:num_rows,c))<percent_stamen_removed*pea
k_columns_and_location(i-1,1);
                 %   reference_image(1:num_rows,c)=0;
                  %  if
nnz(noise_fill_bw_im_contrast(1:num_rows,c))>=1;
                   %     stamen_length_protruding_from_image(i,1) =
1+stamen_length_protruding_from_image(i,1);
                    %end
                %else
                 %   break
                %end
                %end
                                    %reference_image = image_no_stamen;
            end
            for t=1:num_rows; %Rows
                num_non_zero_pixels_each_row(t,1) =
nnz(reference_image(t,1:num_columns));
            end
            for t=1:num_columns;
                num_non_zero_pixels_each_column(t,1) =
nnz(reference_image(1:num_rows,t));
            end
        % Obtain first and last point along centroid without stamen
            min_point(i,1) =
find(num_non_zero_pixels_each_column,1,'first');
            min_point(i,2) =
find(reference_image(1:num_rows,min_point(i,1)),1,'first');
            max_point(i,1) =
find(num_non_zero_pixels_each_column,1,'last');
            max_point(i,2) =
find(reference_image(1:num_rows,max_point(i,1)),1,'first');
            max_row=max(num_non_zero_pixels_each_row);
        % For Zero or One Bag
            if num_bags ==0 || num_bags ==1;
                [peak_rows,location_peak_rows] =
findpeaks(num_non_zero_pixels_each_row(1:num_rows,1),'MINPEAKHEIGHT',ma
x_row-1,'NPEAKS',1);
                peak_rows_and_location(i,1) = peak_rows;
                peak_rows_and_location(i,2) = location_peak_rows;
            end
        % For Two Bags
            if num_bags == 2;
                if first_bag_of_two_no_stamen_starts>0 &&
second_bag_of_two_no_stamen_starts>0 &&
```

```
first_bag_of_two_no_stamen_ends>0 &&
second_bag_of_two_no_stamen_ends>0;
                    first_bag_starts =
first_bag_of_two_no_stamen_starts;
                    second_bag_starts =
second_bag_of_two_no_stamen_starts;
                    first_bag_ends = first_bag_of_two_no_stamen_ends;
                    second_bag_ends = second_bag_of_two_no_stamen_ends;
                end
                if first_bag_of_two_with_stamen_starts >0 &&
second_bag_of_two_with_stamen_starts>0 &&
first_bag_of_two_with_stamen_ends>0 &&
second_bag_of_two_with_stamen_ends>0 ;
                    first_bag_starts =
first_bag_of_two_with_stamen_starts ;
                    second_bag_starts =
second_bag_of_two_with_stamen_starts;
                    first_bag_ends = first_bag_of_two_with_stamen_ends;
                    second_bag_ends =
second_bag_of_two_with_stamen_ends;
                end
                find_last_row_nnz = find
(num_non_zero_pixels_each_row,1,'last');
                find_first_row_nnz = find
(num_non_zero_pixels_each_row,1,'first');
                avg_first_last =
(find_last_row_nnz+find_first_row_nnz)/2;
            % Top Bag Starts First
                if first_bag_starts < second_bag_starts;
                % Until Bottom Bag Starts
                    if second_bag_starts>i;
                        [peak_rows,location_peak_rows] =
findpeaks(num_non_zero_pixels_each_row(1:num_rows,1),'MINPEAKHEIGHT',ma
x_row-1,'NPEAKS',1);
                        peak_rows_and_location_top(i,1) = peak_rows;
                        peak_rows_and_location_top(i,2) =
location_peak_rows;
                    % Obtain an estimate of bottom bag until there is
some curvature
                        if first_bag_starts<=i;
                            inbetween_peak_row_top_and_bottom =
round((avg_first_last+find_last_row_nnz)/2);
                            peak_rows_and_location_bottom(i,1) =
num_non_zero_pixels_each_row(inbetween_peak_row_top_and_bottom);
                            peak_rows_and_location_bottom(i,2) =
inbetween_peak_row_top_and_bottom;
                            if
peak_rows_and_location_top(i,2)>avg_first_last;
                                inbetween_peak_row_top_and_bottom =
round((avg_first_last+find_first_row_nnz)/2);
                                peak_rows_and_location_top(i,1) =
num_non_zero_pixels_each_row(inbetween_peak_row_top_and_bottom);
                                peak_rows_and_location_top(i,2) =
inbetween_peak_row_top_and_bottom;
                            end
```

```
                        end
                    end
                % After Bottom Bag Emerges and before either bag
breaksup
                    if  second_bag_starts<=i &&  first_bag_ends>i &&
second_bag_ends>i;
                        [peak_rows,location_peak_rows] =
findpeaks(num_non_zero_pixels_each_row(1:num_rows,1),'MINPEAKHEIGHT',ro
und(max_row/3),'MINPEAKDISTANCE',round(peak_columns_and_location(i,1)/2
),'NPEAKS',2);
                        peak_rows_and_location_top(i,1) = peak_rows(1);
                        peak_rows_and_location_top(i,2) =
location_peak_rows(1);
                        if length(peak_rows)==1;
                            inbetween_peak_row_top_and_bottom =
round((avg_first_last+find_last_row_nnz)/2);
                            peak_rows_and_location_bottom(i,1) =
num_non_zero_pixels_each_row(inbetween_peak_row_top_and_bottom);
                            peak_rows_and_location_bottom(i,2) =
inbetween_peak_row_top_and_bottom;
                        else
                            peak_rows_and_location_bottom(i,1) =
peak_rows(2);
                            peak_rows_and_location_bottom(i,2) =
location_peak_rows(2);
                        end
                        if
peak_rows_and_location_top(i,2)>avg_first_last;
                            inbetween_peak_row_top_and_bottom =
round((avg_first_last+find_first_row_nnz)/2);
                            peak_rows_and_location_top(i,1) =
num_non_zero_pixels_each_row(inbetween_peak_row_top_and_bottom);
                            peak_rows_and_location_top(i,2) =
inbetween_peak_row_top_and_bottom;
                        end
                        if
peak_rows_and_location_bottom(i,2)<avg_first_last;
                            inbetween_peak_row_top_and_bottom =
round((avg_first_last+find_last_row_nnz)/2);
                            peak_rows_and_location_bottom(i,1) =
num_non_zero_pixels_each_row(inbetween_peak_row_top_and_bottom);
                            peak_rows_and_location_bottom(i,2) =
inbetween_peak_row_top_and_bottom;
                        end
                    end
                end
            % Bottom Bag Starts First
                if first_bag_starts > second_bag_starts;
                    if first_bag_starts>i; % Until first bag starts
                        [peak_rows,location_peak_rows] =
findpeaks(num_non_zero_pixels_each_row(1:num_rows,1),'MINPEAKHEIGHT',ma
x_row-1,'NPEAKS',1);
                        peak_rows_and_location_bottom(i,1) = peak_rows;
                        peak_rows_and_location_bottom(i,2) =
location_peak_rows;
```

```
                        if second_bag_starts<=i;
                            inbetween_first_nnz_and_bottom_peak =
round((avg_first_last+find_first_row_nnz)/2);
                            peak_rows_and_location_top(i,1) =
num_non_zero_pixels_each_row(inbetween_first_nnz_and_bottom_peak);
                            peak_rows_and_location_top(i,2) =
inbetween_first_nnz_and_bottom_peak;
                            if
peak_rows_and_location_bottom(i,2)<avg_first_last;
                                inbetween_peak_row_top_and_bottom =
round((avg_first_last+find_last_row_nnz)/2);
                                peak_rows_and_location_bottom(i,1) =
num_non_zero_pixels_each_row(inbetween_peak_row_top_and_bottom);
                                peak_rows_and_location_bottom(i,2) =
inbetween_peak_row_top_and_bottom;
                            end
                        end
                    end
                % After Top Bag Emerges and before either bag breaksup
                    if  first_bag_starts<=i &&  first_bag_ends>i &&
second_bag_ends>i;
                        [peak_rows,location_peak_rows] =
findpeaks(num_non_zero_pixels_each_row(1:num_rows,1),'MINPEAKHEIGHT',ro
und(max_row/3),'MINPEAKDISTANCE',round(peak_columns_and_location(i,1)/2
),'NPEAKS',2);
                        if length(peak_rows)==1;
                            peak_rows_and_location_bottom(i,1) =
peak_rows(1);
                            peak_rows_and_location_bottom(i,2) =
location_peak_rows(1);
                            inbetween_first_nnz_and_bottom_peak =
round((avg_first_last+find_first_row_nnz)/2);
                            peak_rows_and_location_top(i,1) =
num_non_zero_pixels_each_row(inbetween_first_nnz_and_bottom_peak);
                            peak_rows_and_location_top(i,2) =
inbetween_first_nnz_and_bottom_peak;
                        else
                            peak_rows_and_location_top(i,1) =
peak_rows(1);
                            peak_rows_and_location_top(i,2) =
location_peak_rows(1);
                            peak_rows_and_location_bottom(i,1) =
peak_rows(2);
                            peak_rows_and_location_bottom(i,2) =
location_peak_rows(2);
                        end
                        if
peak_rows_and_location_bottom(i,2)<avg_first_last;
                            inbetween_peak_row_top_and_bottom =
round((avg_first_last+find_last_row_nnz)/2);
                            peak_rows_and_location_bottom(i,1) =
num_non_zero_pixels_each_row(inbetween_peak_row_top_and_bottom);
                            peak_rows_and_location_bottom(i,2) =
inbetween_peak_row_top_and_bottom;
                        end
```

```
                        if
peak_rows_and_location_top(i,2)>avg_first_last;
                            inbetween_peak_row_top_and_bottom =
round((avg_first_last+find_first_row_nnz)/2);
                            peak_rows_and_location_top(i,1) =
num_non_zero_pixels_each_row(inbetween_peak_row_top_and_bottom);
                            peak_rows_and_location_top(i,2) =
inbetween_peak_row_top_and_bottom;
                        end
                    end
                end
            % Bags start at the same frame
                if  (first_bag_starts == second_bag_starts) &&
first_bag_ends>i && second_bag_ends>i; %After Top Bag Emerges and
before either bag breaksup
                    if first_bag_starts>i;
                        [peak_rows,location_peak_rows] =
findpeaks(num_non_zero_pixels_each_row(1:num_rows,1),'MINPEAKHEIGHT',ma
x_row-1,'NPEAKS',1);
                        peak_rows_and_location_top(i,1) = peak_rows;
                        peak_rows_and_location_top(i,2) =
location_peak_rows;
                    end
                    if first_bag_starts<=i;
                        [peak_rows,location_peak_rows] =
findpeaks(num_non_zero_pixels_each_row(1:num_rows,1),'MINPEAKHEIGHT',ro
und(max_row/3),'MINPEAKDISTANCE',round(peak_columns_and_location(i,1)/2
),'NPEAKS',2);
                        if length(peak_rows)==1;
                    % Find peaks is obtaining only the bottom bag
                            if location_peak_rows>avg_first_last;
                                peak_rows_and_location_bottom(i,1) =
peak_rows(1);
                                peak_rows_and_location_bottom(i,2) =
location_peak_rows(1);
                                inbetween_first_nnz_and_bottom_peak =
round((avg_first_last+find_first_row_nnz)/2);
                                peak_rows_and_location_top(i,1) =
num_non_zero_pixels_each_row(inbetween_first_nnz_and_bottom_peak);
                                peak_rows_and_location_top(i,2) =
inbetween_first_nnz_and_bottom_peak;
                            end
                        % Find peaks is obtaining only the top bag
                            if location_peak_rows<avg_first_last;
                                peak_rows_and_location_top(i,1) =
peak_rows(1);
                                peak_rows_and_location_top(i,2) =
location_peak_rows(1);
                                inbetween_peak_row_top_and_bottom =
round((avg_first_last+find_last_row_nnz)/2);
                                peak_rows_and_location_bottom(i,1) =
num_non_zero_pixels_each_row(inbetween_peak_row_top_and_bottom);
                                peak_rows_and_location_bottom(i,2) =
inbetween_peak_row_top_and_bottom;
                            end
```

```
                        else
                            peak_rows_and_location_top(i,1) =
peak_rows(1);
                            peak_rows_and_location_top(i,2) =
location_peak_rows(1);
                            peak_rows_and_location_bottom(i,1) =
peak_rows(2);
                            peak_rows_and_location_bottom(i,2) =
location_peak_rows(2);
                        end
                        if
peak_rows_and_location_bottom(i,2)<avg_first_last;
                            inbetween_peak_row_top_and_bottom =
round((avg_first_last+find_last_row_nnz)/2);
                            peak_rows_and_location_bottom(i,1) =
num_non_zero_pixels_each_row(inbetween_peak_row_top_and_bottom);
                            peak_rows_and_location_bottom(i,2) =
inbetween_peak_row_top_and_bottom;
                        end
                        if
peak_rows_and_location_top(i,2)>avg_first_last;
                            inbetween_peak_row_top_and_bottom =
round((avg_first_last+find_first_row_nnz)/2);
                            peak_rows_and_location_top(i,1) =
num_non_zero_pixels_each_row(inbetween_peak_row_top_and_bottom);
                            peak_rows_and_location_top(i,2) =
inbetween_peak_row_top_and_bottom;
                        end
                    end
                end
            % Bottom bag ends first and second bag has ended
                if first_bag_ends > second_bag_ends &&
second_bag_ends<=i
                    [peak_rows,location_peak_rows] =
findpeaks(num_non_zero_pixels_each_row(1:num_rows,1),'MINPEAKHEIGHT',ro
und(max_row/2),'NPEAKS',1);
                    peak_rows_and_location_top(i,1) = peak_rows(1);
                    peak_rows_and_location_top(i,2) =
location_peak_rows(1);
                    if peak_rows_and_location_top(i,2)>avg_first_last;
                        inbetween_peak_row_top_and_bottom =
round((avg_first_last+find_first_row_nnz)/2);
                        peak_rows_and_location_top(i,1) =
num_non_zero_pixels_each_row(inbetween_peak_row_top_and_bottom);
                        peak_rows_and_location_top(i,2) =
inbetween_peak_row_top_and_bottom;
                    end
                end
            % Top bag ends first and first bag has ended
                if first_bag_ends < second_bag_ends &&
first_bag_ends<=i
                    [peak_rows,location_peak_rows] =
findpeaks(num_non_zero_pixels_each_row(1:num_rows,1),'MINPEAKHEIGHT',ro
und(max_row/2));
                    find_peak_bottom = find(peak_rows, 1, 'last');
```

```
                    peak_rows_and_location_bottom(i,1) =
peak_rows(find_peak_bottom);
                    peak_rows_and_location_bottom(i,2) =
location_peak_rows(find_peak_bottom);
                    if
peak_rows_and_location_bottom(i,2)<avg_first_last;
                        inbetween_peak_row_top_and_bottom =
round((avg_first_last+find_last_row_nnz)/2);
                        peak_rows_and_location_bottom(i,1) =
num_non_zero_pixels_each_row(inbetween_peak_row_top_and_bottom);
                        peak_rows_and_location_bottom(i,2) =
inbetween_peak_row_top_and_bottom;
                    end
                end
            end
% Obtain Centroid
            Drop_info_centroid_axial(i,1) = stats.Centroid (1);
            Drop_info_centroid_transverse(i,1) = stats.Centroid (2);
        % Output frame number and time corresponding to infomation
            Frame_Number(i,1) = i;
            max_radius=peak_columns_and_location(i,1)/2;
            dist_bw_everywhere=bwdist(im_complement);

min_radius=dist_bw_everywhere(round(Drop_info_centroid_transverse
(i,1)),round(Drop_info_centroid_axial(i,1)));
            ratio(i,1) = max_radius/min_radius;
            if strcmp('no',stop_computing_ratio) && i~=1 && i~=2;
                if ratio(i,1)>=1.1;
                    time_zero_frame_num=i-1;
                    stop_computing_ratio='yes';
                end
            end
            if strcmp('yes',stop_computing_ratio)
                frame_num_from_time_zero(i,1) =
frame_num_from_time_zero(i-1,1) + 1;
            end
        % Obtain Area of breaking up drop
            Drop_info_Area(i,1) = stats.Area;
            orientation(i,1) = stats.Orientation;
            bounding_box_3(i,1) = stats.BoundingBox(3);
        % Obtain Volume of revolution
            if start_recording_stamen>=i || strcmp('n',stamen_present);
            for center_rev = 1:num_rows;
                if center_rev <=
round(Drop_info_centroid_transverse(i,1));
                    top_area_image(center_rev,1:num_columns) =
reference_image(center_rev,1:num_columns);%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                    bottom_area_image(center_rev,1:num_columns) = 0;
                end
                if center_rev >
round(Drop_info_centroid_transverse(i,1));
                    top_area_image(center_rev,1:num_columns) = 0;
```

```
                    bottom_area_image(center_rev,1:num_columns) =
reference_image(center_rev,1:num_columns);%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                end
            end
            top_area_image = bwareaopen(top_area_image, 30);
            bottom_area_image = bwareaopen(bottom_area_image, 30);
            top_area_props =
regionprops(top_area_image,'Centroid','Area','Perimeter');
            x_axis_top_area_centroid = top_area_props.Centroid(1);
            y_axis_top_area_centroid = top_area_props.Centroid(2);
            find_last_point_in_top_image =
find(top_area_image(1:num_rows,round(x_axis_top_area_centroid)),1,'last
')+0.5;
            centroid_distance_top=find_last_point_in_top_image-
y_axis_top_area_centroid;
            volume_top =
pi*top_area_props.Area*(find_last_point_in_top_image-
y_axis_top_area_centroid);
            bottom_area_props =
regionprops(bottom_area_image,'Centroid','Area','Perimeter');
            x_axis_bottom_area_centroid =
bottom_area_props.Centroid(1);
            y_axis_bottom_area_centroid =
bottom_area_props.Centroid(2);
            find_first_point_in_bottom_image =
find(bottom_area_image(1:num_rows,round(x_axis_bottom_area_centroid)),1
,'first')-0.5;
            volume_bottom =
pi*bottom_area_props.Area*(y_axis_bottom_area_centroid-
find_first_point_in_bottom_image);
            centroid_distance_bottom=y_axis_bottom_area_centroid-
find_first_point_in_bottom_image;
            volume_ratio(i,1) =
(volume_top+volume_bottom)/(4/3*pi*(peak_columns_and_location(1,1)/2)^3
);
        % Obtain surface area ratio
            total_perimeter_top = bwmorph(top_area_image,'remove');
            find_last_point_in_top_image_nc =
find(total_perimeter_top(1:num_rows,round(x_axis_top_area_centroid)),1,
'last');
            find_first_point_along_last =
find(total_perimeter_top(find_last_point_in_top_image_nc,1:num_columns)
,1,'first');
            find_last_point_along_last =
find(total_perimeter_top(find_last_point_in_top_image_nc,1:num_columns)
,1,'last');

total_perimeter_top(find_last_point_in_top_image_nc,(find_first_point_a
long_last+1):(find_last_point_along_last-1)) = 0;
            total_perimeter_top = bwareaopen(total_perimeter_top, 10);
            props_perimeter_top =
regionprops(total_perimeter_top,'Centroid','Area','Perimeter');
            arc_length_top = nnz(total_perimeter_top)+2;
```

```
            x_axis_top_perimeter_centroid =
props_perimeter_top.Centroid(1);
            y_axis_top_perimeter_centroid =
props_perimeter_top.Centroid(2);
            distance_perimeter_centroid_top =
find_last_point_in_top_image_nc-y_axis_top_perimeter_centroid+1;
            surface_area_top =
pi*distance_perimeter_centroid_top*arc_length_top;
            total_perimeter_bottom =
bwmorph(bottom_area_image,'remove');
            find_first_point_in_bottom_image_nc =
find(total_perimeter_bottom(1:num_rows,round(x_axis_bottom_area_centroi
d)),1,'first');
            find_first_point_along_first =
find(total_perimeter_bottom(find_first_point_in_bottom_image_nc,1:num_c
olumns),1,'first');
            find_last_point_along_first =
find(total_perimeter_bottom(find_first_point_in_bottom_image_nc,1:num_c
olumns),1,'last');

total_perimeter_bottom(find_first_point_in_bottom_image_nc,(find_first_
point_along_first+1):(find_last_point_along_first-1)) = 0;
            total_perimeter_bottom = bwareaopen(total_perimeter_bottom,
10);
            props_perimeter_bottom =
regionprops(total_perimeter_bottom,'Centroid','Area','Perimeter');
            arc_length_bottom = nnz(total_perimeter_bottom)+2;
            x_axis_bottom_perimeter_centroid =
props_perimeter_bottom.Centroid(1);
            y_axis_bottom_perimeter_centroid =
props_perimeter_bottom.Centroid(2);
            distance_perimeter_centroid_bottom =
y_axis_bottom_perimeter_centroid-find_first_point_in_bottom_image_nc+1;
            surface_area_bottom =
pi*distance_perimeter_centroid_bottom*arc_length_bottom;

surface_area_ratio(i,1)=(surface_area_top+surface_area_bottom)/(4*pi*(p
eak_columns_and_location(1,1)/2)^2);
            end
            if num_bags==1 || num_bags==0;
                E(i,1) =
peak_rows_and_location(i,1)/peak_columns_and_location(i,1);
            end
            if num_bags==2
                if peak_rows_and_location_top(1,1)>0;
                    E(i,1) =
peak_rows_and_location_top(i,1)/peak_columns_and_location(i,1);
                end
                if peak_rows_and_location_bottom(1,1)>0;
                    E(i,1) =
peak_rows_and_location_bottom(i,1)/peak_columns_and_location(i,1);
                end
            end
% Show all images and boundaries
%///////////// PLOTS
```

```
        % Create new figure window
            figure(1)
                hFig = figure(1);
                scnsize = get(0,'Screensize');
                set(hFig,'Position',[1 -scnsize(2) scnsize(3)
scnsize(4)]);
                subplot (2,5,1), imshow (im_contrast),
h=title({['Frame#:' num2str(Frame_Number(i,1))];['  of file: '
Current_movie]}); set(h,'Interpreter','none','FontSize',6),hold
'on',axis on;
                    axis([stats.BoundingBox(1,1)-5
stats.BoundingBox(1,1)+stats.BoundingBox(1,3)+5 stats.BoundingBox(1,2)-
5 stats.BoundingBox(1,2)+stats.BoundingBox(1,4)+5] ),hold 'off';
                subplot (2,5,2), imshow (bw_im_contrast), h=title
('Edge Detection'); set(h,'FontSize',6),hold 'on',axis on;
                    axis([stats.BoundingBox(1,1)-5
stats.BoundingBox(1,1)+stats.BoundingBox(1,3)+5 stats.BoundingBox(1,2)-
5 stats.BoundingBox(1,2)+stats.BoundingBox(1,4)+5] ),hold 'off';
                subplot (2,5,3), imshow (noise_bw_im_contrast), h=title
({['Remove Areas less then than:
',num2str(removed_area)]});set(h,'FontSize',6),hold 'on',axis on;
                    axis([stats.BoundingBox(1,1)-5
stats.BoundingBox(1,1)+stats.BoundingBox(1,3)+5 stats.BoundingBox(1,2)-
5 stats.BoundingBox(1,2)+stats.BoundingBox(1,4)+5] ),hold 'off';
                subplot (2,5,4), imshow (pre_pre_pre), h=title
({['Dilation Element Size:
',num2str(structured_element_size)]});set(h,'FontSize',6),hold
'on',axis on;
                    axis([stats.BoundingBox(1,1)-
structured_element_size-5
stats.BoundingBox(1,1)+stats.BoundingBox(1,3)+structured_element_size+5
stats.BoundingBox(1,2)-structured_element_size-5
stats.BoundingBox(1,2)+stats.BoundingBox(1,4)+structured_element_size+5
] ),hold 'off';
                subplot (2,5,5), imshow (pre_pre_morph), h=title
({['Erosion Element of Size:
',num2str(structured_element_size)]});set(h,'FontSize',6),hold
'on',axis on;
                    axis([stats.BoundingBox(1,1)-5
stats.BoundingBox(1,1)+stats.BoundingBox(1,3)+5 stats.BoundingBox(1,2)-
5 stats.BoundingBox(1,2)+stats.BoundingBox(1,4)+5] ),hold 'off';
                subplot (2,5,6), imshow (noise_fill_bw_im_contrast),
                    if num_bags ==1 || num_bags==0;
                        h=title({['Fill Boundary'];['Axial
Dimension:',num2str(peak_rows_and_location(i,1))];['Transverse
Dimension: ',num2str(peak_columns_and_location(i,1))];['Drop Area:
',num2str(Drop_info_Area(i,1))];['Ratio of Max Radius to Min: ',
num2str(ratio(i,1))]});set(h,'FontSize',6),hold 'on',axis on;
                    end
                    if num_bags ==2;
                        h=title ({['Fill Boundary'];['Axial Dimension
Top: ',num2str(peak_rows_and_location_top(i,1))];['Axial Dimension
Bottom: ',num2str(peak_rows_and_location_bottom(i,1))];['Transverse
Dimension: ',
```

```
num2str(peak_columns_and_location(i,1))]});set(h,'FontSize',6),hold
'on',axis on;
                    end
                    axis([stats.BoundingBox(1,1)-5
stats.BoundingBox(1,1)+stats.BoundingBox(1,3)+5 stats.BoundingBox(1,2)-
5 stats.BoundingBox(1,2)+stats.BoundingBox(1,4)+5] )
                % Plot arrow showing direction of movement
                    if i>1;

a=quiver(Drop_info_centroid_axial(i,1),Drop_info_centroid_transverse(i,
1),(Drop_info_centroid_axial(i,1)-Drop_info_centroid_axial(i-
1,1)),(Drop_info_centroid_transverse(i,1)-
Drop_info_centroid_transverse(i-1,1)),3,'r','LineWidth',3);
                    end
                    plot(min_point(i,1),min_point(i,2),'ob');
                    plot(max_point(i,1),max_point(i,2),'ob'),hold
'off';
                subplot (2,5,7), imshow (drop_im_1),
                    h=title ({['Plot Data on Original'];['Axial Area
Centroid: ',num2str(Drop_info_centroid_axial(i,1))];['Transverse Area
Centroid: ',num2str(Drop_info_centroid_transverse(i,1))];['Axial Basal
Centroid: ',
num2str(axial_centroid_based_on_basal_ring(i,1))];['Transverse Basal
Centroid: ',
num2str(transverse_centroid_based_on_basal_ring(i,1))]});set(h,'FontSiz
e',6),hold 'on',axis on;
                    axis([stats.BoundingBox(1,1)-5
stats.BoundingBox(1,1)+stats.BoundingBox(1,3)+5 stats.BoundingBox(1,2)-
5 stats.BoundingBox(1,2)+stats.BoundingBox(1,4)+5] )
                % Plot Boundaries
                    boundary_plot = rectangle('Position',
stats.BoundingBox, 'LineStyle', '--' );
                    set (boundary_plot, 'EdgeColor', [ 0.75 0 0]);
                    if strcmp('y',stamen_present) &&
start_recording_stamen<=i;
                        boundary_plot_no_stamen = rectangle('Position',
stats_no_stamen.BoundingBox, 'LineStyle', '--' );
                        set (boundary_plot, 'EdgeColor', [ 0 0 0.75 ]);
                    end
                % Plot Centroids
                    if strcmp(stop_computing_ratio,'no');

plot(Drop_info_centroid_axial(i,1),Drop_info_centroid_transverse(i,1),'
y*');
                    end
                    if strcmp(stop_computing_ratio,'yes');
                        if start_rim_data>i;

plot(Drop_info_centroid_axial(i,1),Drop_info_centroid_transverse(i,1),'
r*');
                        end
                        if start_rim_data<=i;

plot(axial_centroid_based_on_basal_ring(i,1),transverse_centroid_based_
on_basal_ring(i,1),'r*');
```

```
                        end
                    end
                % Plot Etrema
                    extrema_box = cat(1, stats.Extrema);
                    plot(imgca,extrema_box(:,1), extrema_box(:,2),
'b*');
                % Plot Object Boundaries
                    objectBoundries =
bwboundaries(noise_fill_bw_im_contrast);
                    for thisObject = 1:length(objectBoundries);
                        boundary = objectBoundries{thisObject};
                        plot(boundary(:,2), boundary(:,1),'b');
                    end
                % Plot peak rows (bag length)
                    if num_bags ==0 || num_bags ==1;
                        start_plot_row =
find(reference_image(location_peak_rows,1:num_columns),1,'first');
                        plot([start_plot_row (start_plot_row+peak_rows-
1)],[peak_rows_and_location(i,2)
peak_rows_and_location(i,2)],'Color','g');
                    end
                    if num_bags ==2;
                        if peak_rows_and_location_top(i,1)>0;
                            start_plot_row =
find(reference_image(peak_rows_and_location_top(i,2),1:num_columns),1,'
first');
                            plot([start_plot_row
(start_plot_row+peak_rows_and_location_top(i,1)-
1)],[peak_rows_and_location_top(i,2)
peak_rows_and_location_top(i,2)],'Color','w');
                        end
                        if peak_rows_and_location_bottom(i,1)>0;
                            start_plot_row =
find(reference_image(peak_rows_and_location_bottom(i,2),1:num_columns),
1,'first');
                            plot([start_plot_row
(start_plot_row+peak_rows_and_location_bottom(i,1)-
1)],[peak_rows_and_location_bottom(i,2)
peak_rows_and_location_bottom(i,2)],'Color','g');
                        end
                    end
                % Plot column length (rim length)
                    plot ([bottom_left_point(i,2)
top_left_point(i,2)],[bottom_left_point(i,1)
top_left_point(i,1)],'Color','g'),hold 'off';
                % Plot image used for surface area and volume ratios
                if start_recording_stamen>=i ||
strcmp('n',stamen_present);
                    subplot (2,5,8), hold 'on', imshow
(top_area_image),
plot(x_axis_top_area_centroid,y_axis_top_area_centroid,'*r',x_axis_top_
area_centroid,find_last_point_in_top_image,'*g'),axis on;
                        h=title ({['Surface Area Ratio
',num2str(surface_area_ratio(i))]});set(h,'FontSize',6),axis([stats.Bou
ndingBox(1,1)-5 stats.BoundingBox(1,1)+stats.BoundingBox(1,3)+5
```

```
stats.BoundingBox(1,2)-5
stats.BoundingBox(1,2)+stats.BoundingBox(1,4)+5] ),hold 'off';
                 % Plot bottom area used for surface area and volume
ratios
                    subplot (2,5,9), hold 'on', imshow
(bottom_area_image),plot(x_axis_bottom_area_centroid,y_axis_bottom_area
_centroid,'*r',x_axis_bottom_area_centroid,find_first_point_in_bottom_i
mage,'*g');
                        axis on,axis([stats.BoundingBox(1,1)-5
stats.BoundingBox(1,1)+stats.BoundingBox(1,3)+5 stats.BoundingBox(1,2)-
5 stats.BoundingBox(1,2)+stats.BoundingBox(1,4)+5] )
                        h=title ({['Volume Ratio
',num2str(volume_ratio(i))]});set(h,'FontSize',6),hold 'off';
                end
                % Plot stamen removed image if present
                    if strcmp('y',stamen_present) &&
start_recording_stamen<=i;
                        subplot (2,5,9), hold 'on', imshow
(reference_image),axis on;
                            axis([stats.BoundingBox(1,1)-5
stats.BoundingBox(1,1)+stats.BoundingBox(1,3)+5 stats.BoundingBox(1,2)-
5 stats.BoundingBox(1,2)+stats.BoundingBox(1,4)+5] )
                            plot (Axial_centroid,Transverse_centroid,
'*r'),h=title ('Image with Stamen Removed');set(h,'FontSize',6),hold
'off';
                    % Plot bag-and-stamen removed image if present
                            if start_rim_data<=i
                                %if i>200
                                subplot (2,5,10), hold 'on',
imshow(image_no_stamen_no_bag),axis on;
                                    axis([stats.BoundingBox(1,1)-5
stats.BoundingBox(1,1)+stats.BoundingBox(1,3)+5 stats.BoundingBox(1,2)-
5 stats.BoundingBox(1,2)+stats.BoundingBox(1,4)+5] ),

plot(Axial_centroid,Transverse_centroid,'*r'),h=title ('Image with No
Stamen, nor Bag');set(h,'FontSize',6),hold off;
                                %end
                            end
                    end
            final_movie(i)=getframe(gcf);
            bounding_box_1(i,1) = stats.BoundingBox(1,1);
            bounding_box_2(i,1) = stats.BoundingBox(1,2);
            bounding_box_3(i,1) = stats.BoundingBox(1,3);
            bounding_box_4(i,1) = stats.BoundingBox(1,4);
        end
        toc
    % End check for while statement
        end_check_prompt= {'Does the program miss out significant
"chuncks" of the drop? (y/n)','Does the program have extra significant
"chunks" added to the drop? (y/n)','Is the Stamen Length Protruding
completely measured? (y/n)'};
        end_check_title = 'End of Program Check';
        answer_end_check
=inputdlg(end_check_prompt,end_check_title,num_lines);
        if strcmp('y',answer_end_check(1));
```

```
            end_check_structered_element_size = 5 +
end_check_structered_element_size ;
        elseif (strcmp('n',answer_end_check(1)));
            end_check_structered_element_size = 0 +
end_check_structered_element_size;
        else
            msgbox('This is not a valid entry. The program will
restart.','Help');
        end
        if strcmp('y',answer_end_check(2));
            end_check_noise_removal = 4 + end_check_noise_removal;
        elseif (strcmp('n',answer_end_check(2)));
            end_check_noise_removal = 0 + end_check_noise_removal;
        else
            msgbox('This is not a valid entry. The program will
restart.','Help');
        end
        if strcmp('n',answer_end_check(3));
            end_check_percent_stamen_prompt= {'Sould the amount of drop
removed be increased or decreased? (i/d)'};
            end_check_percent_stamen_title = 'Alter Drop Percentage
Removed';
            answer_end_check_percent_stamen
=inputdlg(end_check_percent_stamen_prompt,end_check_percent_stamen_titl
e,num_lines);
            if strcmp('i',answer_end_check_percent_stamen(1));
                percent_stamen_end_check = 0.1 +
percent_stamen_end_check;
            elseif strcmp('d',answer_end_check_percent_stamen(1));
                percent_stamen_end_check = -0.1 +
percent_stamen_end_check;
            else
                msgbox('This is not a valid entry. The program will
restart.','Help');
            end
        elseif strcmp('y',answer_end_check(3));
            percent_stamen_end_check = 0 + percent_stamen_end_check;
        else
            msgbox('This is not a valid entry. The program will
restart.','Help');
        end
        if
(strcmp('n',answer_end_check(2))&&strcmp('n',answer_end_check(1))&&strc
mp('y',answer_end_check(3)));
            end_check_complete='done';
        end
    end
% Output Movie
    [pathstr, name, ext]=fileparts(Current_movie);
    s = strcat('movie_',name,'.avi');
    movie2avi(final_movie, s,'compression','None','fps',2);
%%
% Obtain Dimensional and Non_dimensional Data
    num_data_points = nnz(Drop_info_centroid_axial);
    Axial_length=zeros(num_data_points,1);
```

```
    bag_growth_rate=zeros(num_data_points,1);
    rim_growth_rate=zeros(num_data_points,1);
    Column_length = zeros(num_data_points,1);
    bounding_box = zeros(num_data_points,1);
    min_extent = zeros(num_data_points,1);
    max_extent = zeros(num_data_points,1);
    num_stamen_points = nnz(stamen_length_protruding_from_image);
    Drop_Area=zeros(num_data_points,1);
    num_data_points_top_bag = nnz(peak_rows_and_location_top)/2;
    find_first_top_bag = find(peak_rows_and_location_top,1,'first');
    num_data_points_bottom_bag = nnz(peak_rows_and_location_bottom)/2;
    find_first_bottom_bag =
find(peak_rows_and_location_bottom,1,'first');
    inst_Oh_num = zeros(num_data_points,1);
    inst_Galilei_num = zeros(num_data_points,1);
    inst_Liquid_Re_num = zeros(num_data_points,1);
    inst_New_non_dimensional_num = zeros(num_data_points,1);
    inst_Non_dimensional_data_x_axial=zeros(num_data_points,1);
    inst_We_number = zeros(num_data_points,1);
    inst_Galilei_num_based_on_inst_acc = zeros(num_data_points,1);
    if peak_rows_and_location_top(1,1)>0;
        find_first_top_bag = time_zero_frame_num;
    end
    if peak_rows_and_location_bottom(1,1)>0;
        find_first_bottom_bag = time_zero_frame_num;
    end
    Axial_length_top_bag=zeros(num_data_points,1);
    bag_growth_rate_top_bag=zeros(num_data_points,1);
    Axial_length_bottom_bag=zeros(num_data_points,1);
    bag_growth_rate_bottom_bag=zeros(num_data_points,1);
    l=0;
% Find Frame When Bounding Box is Minimum
    for g=1:length(bounding_box_3);
        if bounding_box_3(g,1)>0;
            bounding_box(g,1) = bounding_box_3(g,1);
        end
    end
    inv_bounding_box = max(bounding_box)-bounding_box;
    max_inv_bounding_box = max(inv_bounding_box);
    [max_inv_bounding_box,location_max_bounding_box] =
findpeaks(inv_bounding_box,'MINPEAKHEIGHT',max_inv_bounding_box-0.1);
    last_max_inv_bounding_box =
find(location_max_bounding_box,1,'last');
    location_max_inv_bounding_box_based_on_last_max =
find(inv_bounding_box==max_inv_bounding_box(last_max_inv_bounding_box),
1,'last');
% Calculate Initial Relative Velocity, Initial Diameter, and  Initial
Area
    Initial_relative_velocity = mass_flow_rate*28.493 +
0.9852; %(mm/ms)same as (m/s)
    diameters = peak_columns_and_location(1:(time_zero_frame_num-
5),1); % (pixels)
    initial_diameter = mean(diameters)*calibration; %(mm)
    initial_area = pi*(initial_diameter)^2/4; %(mm^2)
% Dimensionalize Area, Axial Length, Column Length, Stamen Length
```

```
    for b = time_zero_frame_num:num_data_points;
        Drop_Area(b,1) = Drop_info_Area(b,1)*calibration^2; % (mm^2)
        Axial_length(b,1) =
peak_rows_and_location(b,1)*calibration; %(mm)
        Column_length(b,1) =
peak_columns_and_location(b,1)*calibration; %(mm)
        min_extent(b,1) = (min_point(b,1)-
min_point(time_zero_frame_num,1))*calibration/initial_diameter; %(*)
        max_extent(b,1) = (max_point(b,1)-
min_point(time_zero_frame_num,1))*calibration/initial_diameter; %(*)
    end
    for b = find_first_top_bag:(num_data_points);
        Axial_length_top_bag(b,1) =
peak_rows_and_location_top(b,1)*calibration; %(mm)
    end
    for b = find_first_bottom_bag:(num_data_points);
        Axial_length_bottom_bag(b,1) =
peak_rows_and_location_bottom(b,1)*calibration; %(mm)
    end
    find_first_in_stamen =
find(stamen_length_protruding_from_image,1,'first');
    for f=1:num_stamen_points;
        stamen_length_protruding(find_first_in_stamen+f-
1,1)=stamen_length_protruding_from_image(find_first_in_stamen+f-
1,1)*calibration; %(mm)
    end
% Non-Dimensional Stamen Length, Drop Area, Axial Length, and Column
Length
    Non_dimensional_stamen_length_protruding =
stamen_length_protruding/initial_diameter;
    Non_dimensional_drop_area = Drop_Area/initial_area;
    Non_dimensional_axial_length = Axial_length/initial_diameter;
    Non_dimensional_axial_length_top_bag =
Axial_length_top_bag/initial_diameter;
    Non_dimensional_axial_length_bottom_bag =
Axial_length_bottom_bag/initial_diameter;
    Non_dimensional_column_length = Column_length/initial_diameter;
    if num_bags ==1 || num_bags==0;
        s=0;
        for t=time_zero_frame_num:length(Axial_length);
            if Axial_length(t,1)>0;
                s=s+1;
                axial_length_positive(s,1) = Axial_length(t,1);
            end
        end
    % Obtain Minimum Axial Length
        inv_axial_length = max(axial_length_positive)-
axial_length_positive;
        max_inv_axial_length = max(inv_axial_length);
        [max_inv_axial_length,location_max_inv_axial_length] =
findpeaks(inv_axial_length,'MINPEAKHEIGHT',max_inv_axial_length-0.001);
    % First Minimum
        location_max_inv_axial_length =
location_max_inv_axial_length(find(location_max_inv_axial_length,1,'las
t'));
```

```
        location_max_inv_axial_length = location_max_inv_axial_length +
time_zero_frame_num - 1;
        min_axial_length_based_on_first_min =
Axial_length(location_max_inv_axial_length);
        min_axial_length_based_on_first_min =
min_axial_length_based_on_first_min(find(min_axial_length_based_on_firs
t_min,1,'last'),1);
       % Furthest Minimum
        location_min_axial_length_based_on_last_min =
find(Axial_length==min_axial_length_based_on_first_min,1,'last');
        min_axial_length_based_on_last_min =
Axial_length(location_min_axial_length_based_on_last_min);
        Non_dimensional_min_axial_length_based_on_last_min =
Non_dimensional_axial_length(location_min_axial_length_based_on_last_mi
n);
       % Axial Length Based on minimum bounding box
        min_axial_length_based_on_bounding_box =
Axial_length(location_max_inv_bounding_box_based_on_last_max);
        Non_dimensional_min_axial_length_based_on_bounding_box =
Non_dimensional_axial_length(location_max_inv_bounding_box_based_on_las
t_max);
    end
    if num_bags==2;
        s=0;
        if Axial_length_top_bag(time_zero_frame_num,1)>0;
            for t=time_zero_frame_num:length(Axial_length_top_bag);
                if Axial_length_top_bag(t,1)>0;
                    s=s+1;
                    axial_length_positive (s,1) =
Axial_length_top_bag(t,1);
                end
            end
        % Obtain Minimum Axial Length
            inv_axial_length = max(axial_length_positive)-
axial_length_positive;
            max_inv_axial_length = max(inv_axial_length);
            [max_inv_axial_length,location_max_inv_axial_length] =
findpeaks(inv_axial_length,'MINPEAKHEIGHT',max_inv_axial_length-0.001);
        % First Minimum
            location_max_inv_axial_length =
location_max_inv_axial_length(find(location_max_inv_axial_length,1,'las
t'));
            location_max_inv_axial_length =
location_max_inv_axial_length + time_zero_frame_num - 1;
            min_axial_length_based_on_first_min =
Axial_length_top_bag(location_max_inv_axial_length);
            min_axial_length_based_on_first_min =
min_axial_length_based_on_first_min(find(min_axial_length_based_on_firs
t_min,1,'last'),1);
        % Furthest Minimum
            location_min_axial_length_based_on_last_min =
find(Axial_length_top_bag==min_axial_length_based_on_first_min,1,'last'
);
            min_axial_length_based_on_last_min =
Axial_length_top_bag(location_min_axial_length_based_on_last_min);
```

```
            Non_dimensional_min_axial_length_based_on_last_min =
Non_dimensional_axial_length_top_bag(location_min_axial_length_based_on
_last_min);
        % Axial Length Based on minimum bounding box
            min_axial_length_based_on_bounding_box =
Axial_length_top_bag(location_max_inv_bounding_box_based_on_last_max);
            Non_dimensional_min_axial_length_based_on_bounding_box =
Non_dimensional_axial_length_top_bag(location_max_inv_bounding_box_base
d_on_last_max);
        end
        if Axial_length_bottom_bag(time_zero_frame_num,1)>0;
            for t=time_zero_frame_num:length(Axial_length_bottom_bag);
                if Axial_length_bottom_bag(t,1)>0;
                    s=s+1;
                    axial_length_positive (s,1) =
Axial_length_bottom_bag(t,1);
                end
            end
        % Obtain Minimum Axial Length
            inv_axial_length = max(axial_length_positive)-
axial_length_positive;
            max_inv_axial_length = max(inv_axial_length);
            [max_inv_axial_length,location_max_inv_axial_length] =
findpeaks(inv_axial_length,'MINPEAKHEIGHT',max_inv_axial_length-0.001);
        % First Minimum
            location_max_inv_axial_length =
location_max_inv_axial_length(find(location_max_inv_axial_length,1,'las
t'));
            location_max_inv_axial_length =
location_max_inv_axial_length + time_zero_frame_num - 1;
            min_axial_length_based_on_first_min =
Axial_length_bottom_bag(location_max_inv_axial_length);
            min_axial_length_based_on_first_min =
min_axial_length_based_on_first_min(find(min_axial_length_based_on_firs
t_min,1,'last'),1);
        % Furthest Minimum
            location_min_axial_length_based_on_last_min =
find(Axial_length_bottom_bag==min_axial_length_based_on_first_min,1,'la
st');
            min_axial_length_based_on_last_min =
Axial_length_bottom_bag(location_min_axial_length_based_on_last_min);
            Non_dimensional_min_axial_length_based_on_last_min =
Non_dimensional_axial_length_bottom_bag(location_min_axial_length_based
_on_last_min);
        % Axial Length Based on minimum bounding box
            min_axial_length_based_on_bounding_box =
Axial_length_bottom_bag(location_max_inv_bounding_box_based_on_last_max
);
            Non_dimensional_min_axial_length_based_on_bounding_box =
Non_dimensional_axial_length_bottom_bag(location_max_inv_bounding_box_b
ased_on_last_max);
        end
    end
% Central Differencing
% Preallocating Values
```

```
    data_y_axial=zeros(num_data_points,1);
    data_x_axial=zeros(num_data_points,1);
    velocity_axial_from_data = zeros(num_data_points,1);
    acceleration_axial_from_poly = zeros(num_data_points,1);
    data_y_transverse = zeros(num_data_points,1);
    data_x_transverse = zeros(num_data_points,1);
    velocity_transverse_from_data = zeros(num_data_points,1);
    Cd_from_data = zeros(num_data_points,1);
    axial_velocity_residuals = zeros(num_data_points,1);
    transverse_over_axial_velocity = zeros(num_data_points,1);
    Non_dimensional_viscous_time = zeros(num_data_points,1);
% Finding Axial Displacement of Centroid (mm), time in (ms)
    for q=(time_zero_frame_num:num_data_points);
        if q<location_max_inv_bounding_box_based_on_last_max;
        % Using Area Centroid
            data_y_axial(q,1)=(Drop_info_centroid_axial(q,1)-
Drop_info_centroid_axial(time_zero_frame_num,1))*calibration; %(mm)
            data_x_axial(q,1)=(q-
time_zero_frame_num)*1000./frame_speed; %(ms)
        end
        if q>=location_max_inv_bounding_box_based_on_last_max;
        % Using Basal Axis
            data_y_axial(q,1)=(axial_centroid_based_on_basal_ring(q,1)-
Drop_info_centroid_axial(time_zero_frame_num,1))*calibration;%(mm)
            data_x_axial(q,1)=(q-
time_zero_frame_num)*1000./frame_speed; %(ms)
        end
    end
% Vertical displacement to check accuracy of inputted data
    for q=(1:num_data_points);
% Using Area Centroid
        data_y_transverse(q,1)=-1*(Drop_info_centroid_transverse(q,1)-
Drop_info_centroid_transverse(1,1))*calibration; %(mm)
        data_x_transverse(q,1)=(q-1)*1000./frame_speed;%(ms)
    end
    for differ_i=(1:num_data_points);
        if differ_i==1;
            velocity_transverse_from_data(differ_i,1) =
(data_y_transverse(differ_i+1)-
data_y_transverse(differ_i))/(data_x_transverse(differ_i+1)-
data_x_transverse(differ_i));%(mm/ms) same as (m/s)
        end
        if differ_i>1 && differ_i<num_data_points;
            velocity_transverse_from_data(differ_i,1)
=(data_y_transverse(differ_i+1)-data_y_transverse(differ_i-
1))/(2*(data_x_transverse(differ_i+1)-
data_x_transverse(differ_i)));%(mm/ms) same as (m/s)
        end
        if differ_i==num_data_points;
            velocity_transverse_from_data(differ_i,1)
=(data_y_transverse(differ_i-1)-
data_y_transverse(differ_i))/(data_x_transverse(differ_i-1)-
data_x_transverse(differ_i));%(mm/ms) same as (m/s)
        end
    end
```

```
%///////////////////////////////////////////////////////////////////////
////////////////////////////////////////
    poly_axial_velocity_order=6;
% Finding Velocity of Axial Centroid (mm/ms)
    for differ_i=(time_zero_frame_num:num_data_points);
        if differ_i==time_zero_frame_num;
            velocity_axial_from_data(differ_i,1) =
(data_y_axial(differ_i+1)-
data_y_axial(differ_i))/(data_x_axial(differ_i+1)-
data_x_axial(differ_i));%(mm/ms) same as (m/s)
        end
        if differ_i>time_zero_frame_num && differ_i<num_data_points;
            velocity_axial_from_data(differ_i,1) =
(data_y_axial(differ_i+1)-data_y_axial(differ_i-
1))/(2*(data_x_axial(differ_i+1)-data_x_axial(differ_i)));%(mm/ms) same
as (m/s)
        end
        if differ_i==num_data_points;
            velocity_axial_from_data(differ_i,1) =
(data_y_axial(differ_i-1)-
data_y_axial(differ_i))/(data_x_axial(differ_i-1)-
data_x_axial(differ_i));  %(mm/ms) same as (m/s)
        end
    end
    velocity_fraction = (Initial_relative_velocity-
velocity_axial_from_data)/Initial_relative_velocity;
    for d=time_zero_frame_num:num_data_points;
        transverse_over_axial_velocity(d,1) =
abs(velocity_transverse_from_data(d,1))/velocity_axial_from_data(d,1);
    end
    polynomial_velocity_function =
polyfit(data_x_axial(time_zero_frame_num:num_data_points,1),velocity_ax
ial_from_data(time_zero_frame_num:num_data_points,1),poly_axial_velocit
y_order);
    polynomial_velocity =
polyval(polynomial_velocity_function,data_x_axial);
    polynomial_velocity(1:(time_zero_frame_num-1),1) = 0;
    sum_of_squared_axial_residuals=0;
    sum_of_squared_axial_diff_from_mean=0;
    mean_axial_velocity =
mean(velocity_axial_from_data(time_zero_frame_num:num_data_points,1));
    for r=(time_zero_frame_num:num_data_points);
        axial_velocity_residuals(r,1) = velocity_axial_from_data(r,1)-
polynomial_velocity(r,1);
        sum_of_squared_axial_residuals =
(velocity_axial_from_data(r,1)-
polynomial_velocity(r,1))^2+sum_of_squared_axial_residuals;
        sum_of_squared_axial_diff_from_mean =
(velocity_axial_from_data(r,1)-
mean_axial_velocity)^2+sum_of_squared_axial_diff_from_mean;
    end
    R_squared_axial_velocity = 1 -
sum_of_squared_axial_residuals*(num_data_points-
time_zero_frame_num)/(sum_of_squared_axial_diff_from_mean*(num_data_poi
nts-time_zero_frame_num-poly_axial_velocity_order));
```

```
% Finding Acceleration of Axial Centroid (mm/ms^2)
    for differ_i=(time_zero_frame_num:num_data_points);
        if differ_i==time_zero_frame_num;

acceleration_axial_from_poly(differ_i,1)=(polynomial_velocity(differ_i+
1)-polynomial_velocity(differ_i))/(data_x_axial(differ_i+1)-
data_x_axial(differ_i));
        end
        if differ_i>time_zero_frame_num && differ_i<num_data_points;
            acceleration_axial_from_poly(differ_i,1)
=(polynomial_velocity(differ_i+1)-polynomial_velocity(differ_i-
1))/(2*(data_x_axial(differ_i+1)-data_x_axial(differ_i)));
        end
        if differ_i==num_data_points;
            acceleration_axial_from_poly(differ_i,1)
=(polynomial_velocity(differ_i-1)-
polynomial_velocity(differ_i))/(data_x_axial(differ_i-1)-
data_x_axial(differ_i));
        end
    end
    time_based_on_min_bounding_box =
data_x_axial(location_max_inv_bounding_box_based_on_last_max);
    Non_dimensional_transverse_based_on_min_bounding_box =
Non_dimensional_column_length(location_max_inv_bounding_box_based_on_la
st_max);
% Calculate Effective Viscosity
    if Selection_liquid == 7;
        Newtonian_or_not = 'n';
    else
        Viscosity =
flow_consistency_index_k*((Non_dimensional_transverse_based_on_min_boun
ding_box-
1)*1000/time_based_on_min_bounding_box)^(flow_behavior_index_n-1); %
(Pa*s)
        Newtonian_or_not = 'nn';
    end
    Oh_number =
Viscosity/sqrt(density_drop*(initial_diameter/1000)*surface_tension);
    We_number =
density_air*(Initial_relative_velocity)^2*initial_diameter/(surface_ten
sion*1000);
    Cd_spheroid_near_critical = 0.445*(1+1.63*(1-
E(1:num_data_points)));
    Cd_spheroid = 0.445*(1+1.63*(1-E(1:num_data_points)).^2);
    Galilei_num =
9.81*(initial_diameter/1000)^3*density_drop^2/Viscosity^2;
    Liquid_Re_num =
(initial_diameter/1000)*Initial_relative_velocity*density_air/Viscosity
;
    New_non_dimensional_num =
(density_air/density_drop)*(dyn_viscosity_air/Viscosity)*We_number;
        for i = time_zero_frame_num:num_data_points;
            inst_Oh_num(i,1) =
Viscosity/sqrt(density_drop*(Column_length(i,1)/1000)*surface_tension);
```

```
            inst_Galilei_num(i,1) =
9.81*(Column_length(i,1)/1000)^3*density_drop^2/Viscosity^2;
            inst_Galilei_num_based_on_inst_acc(i,1) =
acceleration_axial_from_poly(i,1)*1000*(Column_length(i,1)/1000)^3*dens
ity_drop^2/Viscosity^2;
            inst_Liquid_Re_num(i,1) =
(Column_length(i,1)/1000)*(Initial_relative_velocity-
velocity_axial_from_data(i,1))*density_air/Viscosity;
            inst_We_number(i,1) =
density_air*(Initial_relative_velocity-
velocity_axial_from_data(i,1))^2*Column_length(i,1)/(surface_tension*10
00);
            inst_New_non_dimensional_num(i,1) =
(density_air/density_drop)*(dyn_viscosity_air/Viscosity)*inst_We_number
(i,1);

inst_Non_dimensional_data_x_axial(i,1)=data_x_axial(i,1)*(Initial_relat
ive_velocity-
velocity_axial_from_data(i,1))*sqrt(density_air/density_drop)/(Column_l
ength(i,1));
        end
% Rim Growth Rate
    if num_bags==1;
        for differ_i=(time_zero_frame_num:num_data_points);
            if differ_i==time_zero_frame_num;
                rim_growth_rate(differ_i,1) =
(Column_length(differ_i+1)-
Column_length(differ_i))/(data_x_axial(differ_i+1)-
data_x_axial(differ_i));
            end
            if differ_i>time_zero_frame_num &&
differ_i<num_data_points;
                rim_growth_rate(differ_i,1) =
(Column_length(differ_i+1)-Column_length(differ_i-
1))/(2*(data_x_axial(differ_i+1)-data_x_axial(differ_i)));
            end
            if differ_i==num_data_points;
                rim_growth_rate(differ_i,1) = (Column_length(differ_i-
1)-Column_length(differ_i))/(data_x_axial(differ_i-1)-
data_x_axial(differ_i));
            end
        end
    end
% Bag Growth Rate for One bag
    if num_bags==1;
        for differ_i=(time_zero_frame_num:num_data_points);
            if differ_i==time_zero_frame_num;
                bag_growth_rate(differ_i,1) =
(Axial_length(differ_i+1)-
Axial_length(differ_i))/(data_x_axial(differ_i+1)-
data_x_axial(differ_i));
            end
            if differ_i>time_zero_frame_num &&
differ_i<num_data_points;
```

```
                bag_growth_rate(differ_i,1) =
(Axial_length(differ_i+1)-Axial_length(differ_i-
1))/(2*(data_x_axial(differ_i+1)-data_x_axial(differ_i)));
            end
            if differ_i==num_data_points;
                bag_growth_rate(differ_i,1) = (Axial_length(differ_i-
1)-Axial_length(differ_i))/(data_x_axial(differ_i-1)-
data_x_axial(differ_i));
            end
        end
    end
% Bag Growth Rate for two bags
    if num_bags==2;
        if Axial_length_top_bag(time_zero_frame_num,1)>0;
            frame_start_top_bag = time_zero_frame_num;
            frame_start_bottom_bag = find_first_bottom_bag;
        end
        if Axial_length_bottom_bag(time_zero_frame_num,1)>0;
            frame_start_top_bag = find_first_top_bag;
            frame_start_bottom_bag = time_zero_frame_num;
        end
        for differ_i=(frame_start_top_bag:num_data_points);
            if differ_i==frame_start_top_bag;
                bag_growth_rate_top_bag(differ_i,1) =
(Axial_length_top_bag(differ_i+1)-
Axial_length_top_bag(differ_i))/(data_x_axial(differ_i+1)-
data_x_axial(differ_i));
            end
            if differ_i>frame_start_top_bag &&
differ_i<num_data_points;
                bag_growth_rate_top_bag(differ_i,1) =
(Axial_length_top_bag(differ_i+1)-Axial_length_top_bag(differ_i-
1))/(2*(data_x_axial(differ_i+1)-data_x_axial(differ_i)));
            end
            if differ_i==num_data_points;
                bag_growth_rate_top_bag(differ_i,1) =
(Axial_length_top_bag(differ_i-1)-
Axial_length_top_bag(differ_i))/(data_x_axial(differ_i-1)-
data_x_axial(differ_i));
            end
        end
        for differ_i=(frame_start_bottom_bag:num_data_points);
            if differ_i==frame_start_bottom_bag;
                bag_growth_rate_bottom_bag(differ_i,1) =
(Axial_length_bottom_bag(differ_i+1)-
Axial_length_bottom_bag(differ_i))/(data_x_axial(differ_i+1)-
data_x_axial(differ_i));
            end
            if differ_i>frame_start_bottom_bag &&
differ_i<num_data_points;
                bag_growth_rate_bottom_bag(differ_i,1) =
(Axial_length_bottom_bag(differ_i+1)-Axial_length_bottom_bag(differ_i-
1))/(2*(data_x_axial(differ_i+1)-data_x_axial(differ_i)));
            end
            if differ_i==num_data_points;
```

```
                bag_growth_rate_bottom_bag(differ_i,1) =
(Axial_length_bottom_bag(differ_i-1)-
Axial_length_bottom_bag(differ_i))/(data_x_axial(differ_i-1)-
data_x_axial(differ_i));
            end
        end
    end
    Non_dimensional_data_y_axial=data_y_axial/initial_diameter;

Non_dimensional_data_x_axial=data_x_axial*Initial_relative_velocity*sqr
t(density_air/density_drop)/(initial_diameter);
    for d=time_zero_frame_num:num_data_points;
        Non_dimensional_viscous_time(d,1) =
data_x_axial(d,1)*nthroot(density_drop*(acceleration_axial_from_poly(d,
1)*1000)^2/Viscosity,3);
    end

Non_dimensional_data_y_transverse=data_y_transverse/initial_diameter;

Non_dimensional_data_x_transverse=data_x_transverse*Initial_relative_ve
locity*sqrt(density_air/density_drop)/(initial_diameter);

Non_dimensional_velocity_axial_from_data=velocity_axial_from_data*sqrt(
density_drop/density_air)/Initial_relative_velocity;

Non_dimensional_velocity_axial_from_poly=polynomial_velocity*sqrt(densi
ty_drop/density_air)/Initial_relative_velocity;
    Non_dimensional_velocity_transverse_from_data =
velocity_transverse_from_data*sqrt(density_drop/density_air)/Initial_re
lative_velocity;
    Non_dimensional_acceleration_axial_from_poly =
acceleration_axial_from_poly*initial_diameter*(density_drop/density_air
)/((Initial_relative_velocity)^2);
    Non_dimensional_bag_growth_rate =
bag_growth_rate*sqrt(density_drop/density_air)/Initial_relative_velocit
y;
    Non_dimensional_bag_growth_rate_top_bag =
bag_growth_rate_top_bag*sqrt(density_drop/density_air)/Initial_relative
_velocity;
    Non_dimensional_bag_growth_rate_bottom_bag =
bag_growth_rate_bottom_bag*sqrt(density_drop/density_air)/Initial_relat
ive_velocity;
    Non_dimensional_rim_growth_rate =
rim_growth_rate*sqrt(density_drop/density_air)/Initial_relative_velocit
y;
    density_ratio = density_drop/density_air;
    viscosity_ratio = Viscosity/dyn_viscosity_air;
    mach_number = Initial_relative_velocity/346.3;
% Obtaining the Coeffecient of drag
    for drag_int = time_zero_frame_num:num_data_points;
        Cd_from_data(drag_int,1) =
4*density_drop*initial_diameter^3*(acceleration_axial_from_poly(drag_in
t,1))/(3*density_air*(Column_length(drag_int,1))^2*(Initial_relative_ve
locity-polynomial_velocity(drag_int,1))^2);
    end
```

```
% Non-Dimensional Plots
    figure(3);
        hFig = figure(3);
        set(hFig,'Position', get(0,'Screensize'));
        subplot(2,3,1),
plot(Non_dimensional_data_x_axial(time_zero_frame_num:num_data_points),
Non_dimensional_data_y_axial(time_zero_frame_num:num_data_points),'ob')
,h=title('Axial Centroid Displacement versus
Time');set(h,'FontSize',6),grid on;
            xlabel('T = t*(pho_g/pho_L)^1^/^2 * U_r_e_l /
D_o'),ylabel('x/D_o');
        subplot
(2,3,2),plot(Non_dimensional_data_x_axial(time_zero_frame_num:num_data_
points),Non_dimensional_velocity_axial_from_data(time_zero_frame_num:nu
m_data_points),'sg',Non_dimensional_data_x_axial(time_zero_frame_num:nu
m_data_points),Non_dimensional_velocity_axial_from_poly(time_zero_frame
_num:num_data_points),'-r'),h=title('Axial Centroid Velocity versus
Time');set(h,'FontSize',6),grid on;
            xlabel('T = t*(pho_g/pho_L)^1^/^2 * U_r_e_l /
D_o'),ylabel('V*(pho_L/pho_g)^1^/^2 /Vrel_o'),h=legend('Axial
Velocity','Fitted
Velocity','Location','NorthWest');set(h,'FontSize',8);
        subplot
(2,3,3),plot(Non_dimensional_data_x_axial(time_zero_frame_num:num_data_
points),Non_dimensional_acceleration_axial_from_poly(time_zero_frame_nu
m:num_data_points),'-r'),h=title('Axial Centroid Acceleration versus
Time');set(h,'FontSize',6),grid on;
            xlabel('T = t*(pho_g/pho_L)^1^/^2 * U_r_e_l /
D_o'),ylabel('Acc*D_o*pho_L/((Urel_o)^2*pho_g)'),h=legend('Axial
Acceleration','Location','NorthWest');set(h,'FontSize',10);
        subplot
(2,3,4),plot(Non_dimensional_data_x_axial(time_zero_frame_num:num_data_
points),Non_dimensional_drop_area(time_zero_frame_num:num_data_points),
'.c'),h=title('Area of Drop versus Time');set(h,'FontSize',6),grid on;
            xlabel('T = t*(pho_g/pho_L)^1^/^2 * U_r_e_l /
D_o'),ylabel('Area/A_o');
        if num_bags ==1 || num_bags ==0;
            subplot (2,3,5), hold 'on',

plot(Non_dimensional_data_x_axial(time_zero_frame_num:num_data_points),
Non_dimensional_axial_length(time_zero_frame_num:num_data_points),'*y',
Non_dimensional_data_x_axial(time_zero_frame_num:num_data_points),Non_d
imensional_column_length(time_zero_frame_num:num_data_points),'pb'),gri
d on,
                xlabel('T = t*(pho_g/pho_L)^1^/^2 * U_r_e_l /
D_o'),ylabel('(Length)/D_o');
                if strcmp('y',stamen_present);
                    h=title('Bag Length, Stamen Length Protruding, and
Cross Stream Dimension versus Time');set(h,'FontSize',6);
                    find_first_in_stamen =
find(stamen_length_protruding,1,'first');
                    length_stamen_data = nnz(stamen_length_protruding);
                    plot
(Non_dimensional_data_x_axial(find_first_in_stamen:(find_first_in_stame
n+length_stamen_data-
```

```
1)),Non_dimensional_stamen_length_protruding(find_first_in_stamen:(find
_first_in_stamen+length_stamen_data-1)),'hg'),grid on,
                    h=legend ('Length of Bag','Cross stream
Dimension','Length of Stamen
Protruding','Location','NorthWest');set(h,'FontSize',6);
                end
                if strcmp('n',stamen_present);
                    h=title('Bag Length and Cross Stream diameter
versus Time');set(h,'FontSize',6);
                    h=legend ('Length of Bag','Cross stream
Dimension','Location','NorthWest');set(h,'FontSize',6),grid on;
                end
                hold 'off';
        end
        if num_bags==2;
            subplot (2,3,5), hold 'on',
                find_start_top_bag =
find(Axial_length_top_bag,1,'first');
                nnz_top_bag = nnz(Axial_length_top_bag);
                find_start_bottom_bag =
find(Axial_length_bottom_bag,1,'first');
                nnz_bottom_bag = nnz(Axial_length_bottom_bag);
                if Axial_length_top_bag(1,1)>0;
                    find_start_top_bag = time_zero_frame_num;
                end
                if Axial_length_bottom_bag(1,1)>0;
                    find_start_bottom_bag = time_zero_frame_num;
                end

plot(Non_dimensional_data_x_axial(find_start_top_bag:(find_start_top_ba
g+nnz_top_bag-1)),
Non_dimensional_axial_length_top_bag(find_start_top_bag:(find_start_top
_bag+nnz_top_bag-
1)),'or',Non_dimensional_data_x_axial(find_start_bottom_bag:(find_start
_bottom_bag+nnz_bottom_bag-1)),
Non_dimensional_axial_length_bottom_bag(find_start_bottom_bag:(find_sta
rt_bottom_bag+nnz_bottom_bag-
1)),'sk',Non_dimensional_data_x_axial,Non_dimensional_column_length,'pb
'),grid on;
                xlabel('T = t*(pho_g/pho_L)^1^/^2 * U_r_e_l /
D_o'),ylabel('Length/D_o');
                if strcmp('y',stamen_present);
                    h=title('Bag Length, Stamen Length Protruding, and
Cross Stream diameter versus Time');set(h,'FontSize',6)
                    find_first_in_stamen =
find(stamen_length_protruding,1,'first');
                    length_stamen_data = nnz(stamen_length_protruding);
                    plot
(Non_dimensional_data_x_axial(find_first_in_stamen:(find_first_in_stame
n+length_stamen_data-
1)),Non_dimensional_stamen_length_protruding(find_first_in_stamen:(find
_first_in_stamen+length_stamen_data-1)),'hg'),
                    h=legend ('Length of Top Bag','Length of Bottom
Bag','Cross stream Diameter','Length of Stamen
Protruding','Location','NorthWest');set(h,'FontSize',6),grid on;
```

```
                end
                if strcmp('n',stamen_present);
                    h=title('Bag Lengths and Cross Stream diameter
versus Time');set(h,'FontSize',6)
                    h=legend ('Length of Top Bag','Length of Bottom
Bag','Cross stream
Dimension','Location','NorthWest');set(h,'FontSize',6),grid on;
                end
                hold 'off';
        end
        subplot (2,3,6), hold 'on',
g=plot(Non_dimensional_data_x_axial(time_zero_frame_num:num_data_points
),Cd_from_data(time_zero_frame_num:num_data_points),'db');

plot(Non_dimensional_data_x_axial(time_zero_frame_num:location_max_inv_
bounding_box_based_on_last_max),Cd_spheroid(time_zero_frame_num:locatio
n_max_inv_bounding_box_based_on_last_max),'-m','MarkerSize',20);

plot(Non_dimensional_data_x_axial(time_zero_frame_num:location_max_inv_
bounding_box_based_on_last_max),Cd_spheroid_near_critical(time_zero_fra
me_num:location_max_inv_bounding_box_based_on_last_max),'-
g','MarkerSize',20);

plot(Non_dimensional_data_x_axial,0.445,'+c',Non_dimensional_data_x_axi
al,1.17,'+c');
            h=legend('Cd from Polynomial','Cd for Spheroid','Cd for
Spheroid near critical','Cd for a Sphere','Cd for a
Disk','Location','SouthEast');set(h,'FontSize',6);
            xlabel('T = t*(pho_g/pho_L)^1^/^2 * U_r_e_l /
D_o'),ylabel('Cd'),h=title ('Coefficient of Drag versus Non-Dimensional
Time');set(h,'FontSize',6),hold 'off';grid on;
            s = strcat('plots_Non_Dimensional',name,'.png');

set(gcf,'PaperUnits','inches','PaperSize',[15,6],'PaperPosition',[0 0
15 6])
            print('-dpng','-r200',s);
% Dimensional Plots
    figure(4);
        hFig = figure(4);
        set(hFig,'Position', get(0,'Screensize'));
        subplot(2,3,1),
plot(data_x_axial(time_zero_frame_num:num_data_points),data_y_axial(tim
e_zero_frame_num:num_data_points),'ob'),h=title('Axial Centroid
Displacement versus Time');set(h,'FontSize',6),grid on;
            xlabel('Time (ms)'),ylabel('Displacment (mm)');
        subplot (2,3,2), hold 'on',
plot(data_x_axial(time_zero_frame_num:num_data_points),velocity_axial_f
rom_data(time_zero_frame_num:num_data_points),'sg',data_x_axial(time_ze
ro_frame_num:num_data_points),polynomial_velocity(time_zero_frame_num:n
um_data_points),'-r'),h=title('Axial Centroid Velocity versus
Time');set(h,'FontSize',6),grid on;
            xlabel('Time (ms)'),ylabel('Velocity
(mm/ms)'),h=legend('Axial Velocity','Fitted
Velocity','Location','NorthWest');set(h,'FontSize',10);
```

```
        subplot (2,3,3), hold 'on',
plot(data_x_axial(time_zero_frame_num:num_data_points),acceleration_axi
al_from_poly(time_zero_frame_num:num_data_points),'-r'), grid
on,h=title('Axial Centroid Acceleration versus
Time');set(h,'FontSize',6);
            xlabel('Time (ms)'),ylabel('Acceleration (mm/ms^2)');
        subplot
(2,3,4),plot(data_x_axial(time_zero_frame_num:num_data_points),Drop_Are
a(time_zero_frame_num:num_data_points),'.c'),h=title('Area of Drop
versus Time');set(h,'FontSize',6),grid on;
            xlabel('Time (ms)'),ylabel('Area (mm^2)');
        if num_bags ==1 || num_bags ==0;
            subplot (2,3,5),hold 'on';

plot(data_x_axial(time_zero_frame_num:num_data_points),Axial_length(tim
e_zero_frame_num:num_data_points),'*y',data_x_axial(time_zero_frame_num
:num_data_points),Column_length(time_zero_frame_num:num_data_points),'p
b');
                xlabel('Time (ms)'),ylabel('Length (mm)'),grid on
                if strcmp('y',stamen_present);
                    find_first_in_stamen =
find(stamen_length_protruding,1,'first');
                    length_stamen_data = nnz(stamen_length_protruding);
                    plot
(data_x_axial(find_first_in_stamen:(find_first_in_stamen+length_stamen_
data-
1)),stamen_length_protruding(find_first_in_stamen:(find_first_in_stamen
+length_stamen_data-1)),'hg'),grid on,
                    h=title('Bag Length, Cross Stream Dimension and
Stamen Length Protruding versus Time');set(h,'FontSize',6);
                    h=legend('Length of Bag','Cross stream
Dimension','Length of Stamen
Protruding','Location','NorthWest');set(h,'FontSize',6);
                end
                if strcmp('n',stamen_present);
                    h=title('Bag Length and Cross Stream Diameter
versus Time');set(h,'FontSize',6),grid on
                    h=legend('Length of Bag','Cross stream
Dimension','Location','NorthWest');set(h,'FontSize',6);
                end
                hold 'off';
        end
        if num_bags==2;
            subplot (2,3,5), hold 'on',
                find_start_top_bag =
find(Axial_length_top_bag,1,'first');
                nnz_top_bag = nnz(Axial_length_top_bag);
                find_start_bottom_bag =
find(Axial_length_bottom_bag,1,'first');
                nnz_bottom_bag = nnz(Axial_length_bottom_bag);
                if Axial_length_top_bag(time_zero_frame_num)>0;
                    find_start_top_bag = time_zero_frame_num;
                end
                if Axial_length_bottom_bag(time_zero_frame_num)>0;
                    find_start_bottom_bag = time_zero_frame_num;
```

```
                end

plot(data_x_axial(find_start_top_bag:(find_start_top_bag+nnz_top_bag-
1)),
Axial_length_top_bag(find_start_top_bag:(find_start_top_bag+nnz_top_bag
-
1)),'or',data_x_axial(find_start_bottom_bag:(find_start_bottom_bag+nnz_
bottom_bag-1)),
Axial_length_bottom_bag(find_start_bottom_bag:(find_start_bottom_bag+nn
z_bottom_bag-
1)),'sk',data_x_axial(time_zero_frame_num:num_data_points),Column_lengt
h(time_zero_frame_num:num_data_points),'pb'),grid on;
                xlabel('Time (ms)'),ylabel('Length (mm)'),grid on;
                if strcmp('y',stamen_present);
                    find_first_in_stamen =
find(stamen_length_protruding,1,'first');
                    length_stamen_data=
nnz(stamen_length_protruding);set(h,'FontSize',5);
                    plot
(data_x_axial(find_first_in_stamen:(find_first_in_stamen+length_stamen_
data-
1)),stamen_length_protruding(find_first_in_stamen:(find_first_in_stamen
+length_stamen_data-1)),'hg'),
                    h=title ('Length of Bags, Cross Stream Dimension,
and Length of Stamen Protruding versus Time');set(h,'FontSize',6), grid
on;
                    h=legend('Length of Top Bag','Length of Bottom
Bag','Cross Stream Diameter','Length of Stamen
Protruding','Location','NorthWest');set(h,'FontSize',6);
                end
                if strcmp('n',stamen_present);
                    h=title ('Length of Bags and Cross Stream Dimension
versus Time');set(h,'FontSize',6),grid on;
                    h=legend('Length of Top Bag','Length of Bottom
Bag','Cross stream
Diameter','Location','NorthWest');set(h,'FontSize',6);
                end
                hold 'off';
        end
        subplot (2,3,6), hold 'on',
plot(data_x_axial(time_zero_frame_num:num_data_points),Cd_from_data(tim
e_zero_frame_num:num_data_points),'db');

plot(data_x_axial(time_zero_frame_num:location_max_inv_bounding_box_bas
ed_on_last_max),Cd_spheroid(time_zero_frame_num:location_max_inv_boundi
ng_box_based_on_last_max),'-m','MarkerSize',20),

plot(data_x_axial(time_zero_frame_num:location_max_inv_bounding_box_bas
ed_on_last_max),Cd_spheroid_near_critical(time_zero_frame_num:location_
max_inv_bounding_box_based_on_last_max),'-g','MarkerSize',20);
            plot(data_x_axial,0.445,'+c',data_x_axial,1.17,'+c');
            xlabel('Time (ms)'),ylabel('Cd'),grid on,h=title
('Coefficient of Drag versus Time');set(h,'FontSize',6),grid on;
```

```
            h=legend('Cd from Polynomial','Cd for Spheroid','Cd for a 
Sphere','Cd for a 
Disk','Location','SouthEast');set(h,'FontSize',6),hold off;
        s = strcat('plots_Dimensional',name,'.png');

set(gcf,'PaperUnits','inches','PaperSize',[15,6],'PaperPosition',[0 0 
15 6]);
        print('-dpng','-r200',s);
 %%
% Residuals and Transverse Displacement and Velocity
    figure(5);
        hFig = figure(5);
        set(hFig,'Position', get(0,'Screensize'));
        subplot
(2,3,1),plot(data_x_axial(time_zero_frame_num:num_data_points),axial_ve
locity_residuals(time_zero_frame_num:num_data_points),'ob'),h=title({['
Axial Velocity Residuals versus Time'],['R Squared Axial Velocity: ',
num2str(R_squared_axial_velocity)]});set(h,'FontSize',6),grid on;
            xlabel('Time (ms)'),ylabel('Velocity Residuals (mm/ms)');
        subplot(2,3,2),plot(data_x_transverse,data_y_transverse);
            xlabel('Time (ms)'),ylabel('Transverse Displacement 
(mm)'),grid on,h=title('Transverse Displacement versus 
Time');set(h,'FontSize',6),grid on;

subplot(2,3,3),plot(data_x_transverse,velocity_transverse_from_data);
            xlabel('Time (ms)'),ylabel('Transverse Velocity 
(mm/ms)'),grid on,h=title('Transverse Velocity versus Time');grid 
on,set(h,'FontSize',6);

subplot(2,3,4),plot(Non_dimensional_data_x_transverse,Non_dimensional_d
ata_y_transverse);
            xlabel('T = t*(pho_g/pho_L)^1^/^2 * U_r_e_l / 
D_o'),ylabel('y/d0'),grid on,h=title('Non-Dimensional Transverse 
Displacement versus Time');set(h,'FontSize',6),grid on;
        s=
strcat('plots_Residuals_and_Transverse_displacement_and_velocity',name,
'.png');

set(gcf,'PaperUnits','inches','PaperSize',[15,6],'PaperPosition',[0 0 
15 6]);
        print('-dpng','-r200',s);
% Screen shots of specific important frames
    figure(6);
        hFig = figure(6);
        set(hFig,'Position', get(0,'Screensize'));
        subplot(2,5,1), imshow(read(my_movie,time_zero_frame_num)), 
h=title({['Drop at Time Zero'],['Frame Num: ', 
num2str(time_zero_frame_num)]});set(h,'FontSize',6);
            axis([bounding_box_1(time_zero_frame_num,1)-5 
bounding_box_1(time_zero_frame_num,1)+bounding_box_3(time_zero_frame_nu
m,1)+5 bounding_box_2(time_zero_frame_num,1)-5 
bounding_box_2(time_zero_frame_num,1)+bounding_box_4(time_zero_frame_nu
m,1)+5] )
        subplot(2,5,2), 
imshow(read(my_movie,location_min_axial_length_based_on_last_min)),
```

```
h=title({['Initiation with Min Axial Length'],['Frame Num: ',
num2str(location_min_axial_length_based_on_last_min)]});set(h,'FontSize
',6);

axis([bounding_box_1(location_min_axial_length_based_on_last_min,1)-5
bounding_box_1(location_min_axial_length_based_on_last_min,1)+bounding_
box_3(location_min_axial_length_based_on_last_min,1)+5
bounding_box_2(location_min_axial_length_based_on_last_min,1)-5
bounding_box_2(location_min_axial_length_based_on_last_min,1)+bounding_
box_4(location_min_axial_length_based_on_last_min,1)+5] )
        subplot(2,5,3),
imshow(read(my_movie,location_max_inv_bounding_box_based_on_last_max)),
h=title({['Initiation with Min Bounding Box'],['Frame Num: ',
num2str(location_max_inv_bounding_box_based_on_last_max)]});set(h,'Font
Size',6);

axis([bounding_box_1(location_max_inv_bounding_box_based_on_last_max,1)
-5
bounding_box_1(location_max_inv_bounding_box_based_on_last_max,1)+bound
ing_box_3(location_max_inv_bounding_box_based_on_last_max,1)+5
bounding_box_2(location_max_inv_bounding_box_based_on_last_max,1)-5
bounding_box_2(location_max_inv_bounding_box_based_on_last_max,1)+bound
ing_box_4(location_max_inv_bounding_box_based_on_last_max,1)+5] )
        if num_bags == 0;
            subplot(2,5,4),
imshow(read(my_movie,zero_bags_ends)),h=title({['Drop When Measurements
Stop'],['Frame Num: ', num2str(zero_bags_ends)]});set(h,'FontSize',6);
                axis([bounding_box_1(zero_bags_ends-1,1)-5
bounding_box_1(zero_bags_ends-1,1)+bounding_box_3(zero_bags_ends-1,1)+5
bounding_box_2(zero_bags_ends-1,1)-5 bounding_box_2(zero_bags_ends-
1,1)+bounding_box_4(zero_bags_ends-1,1)+5] )
        end
        if num_bags == 1;
            if strcmp('n',stamen_present);
                subplot(2,5,4),hold on,
imshow(read(my_movie,only_one_bag_starts)), h=title({['Drop When Bag
Starts'],['Frame Num: ',
num2str(only_one_bag_starts)]});set(h,'FontSize',6);
                    axis([bounding_box_1(only_one_bag_starts,1)-5
bounding_box_1(only_one_bag_starts,1)+bounding_box_3(only_one_bag_start
s,1)+5 bounding_box_2(only_one_bag_starts,1)-5
bounding_box_2(only_one_bag_starts,1)+bounding_box_4(only_one_bag_start
s,1)+5] ), hold off;
                subplot(2,5,5),hold on,
imshow(read(my_movie,only_one_bag_ends)), h=title({['Drop When Bag
Breaks'],['Frame Num: ',
num2str(only_one_bag_ends)]});set(h,'FontSize',6);
                    axis([bounding_box_1(only_one_bag_ends-1,1)-50
bounding_box_1(only_one_bag_ends-1,1)+bounding_box_3(only_one_bag_ends-
1,1)+50 bounding_box_2(only_one_bag_ends-1,1)-50
bounding_box_2(only_one_bag_ends-1,1)+bounding_box_4(only_one_bag_ends-
1,1)+50] ), hold off;
                subplot(2,5,6),hold on,
imshow(read(my_movie,frame_num_only_rim)),h=title({['Drop When Only Rim
```

```
is Present'],['Frame Num: ',
num2str(frame_num_only_rim)]});set(h,'FontSize',6);
                    axis([bounding_box_1(only_one_bag_ends-1,1)-300
bounding_box_1(only_one_bag_ends-1,1)+bounding_box_3(only_one_bag_ends-
1,1)+300 bounding_box_2(only_one_bag_ends-1,1)-300
bounding_box_2(only_one_bag_ends-1,1)+bounding_box_4(only_one_bag_ends-
1,1)+300] ), hold off;
            end
            if strcmp('y',stamen_present);
                subplot(2,5,4),
imshow(read(my_movie,one_bag_with_stamen_starts)), h=title({['Drop When
Bag Starts'],['Frame Num: ',
num2str(one_bag_with_stamen_starts)]});set(h,'FontSize',6);
                    axis([bounding_box_1(one_bag_with_stamen_starts,1)-
5
bounding_box_1(one_bag_with_stamen_starts,1)+bounding_box_3(one_bag_wit
h_stamen_starts,1)+5 bounding_box_2(one_bag_with_stamen_starts,1)-5
bounding_box_2(one_bag_with_stamen_starts,1)+bounding_box_4(one_bag_wit
h_stamen_starts,1)+5] )
                subplot(2,5,5),
imshow(read(my_movie,stamen_with_one_bag_starts)), h=title({['Drop When
Stamen Emerges'],['Frame Num: ',
num2str(stamen_with_one_bag_starts)]});set(h,'FontSize',6);
                    axis([bounding_box_1(stamen_with_one_bag_starts,1)-
5
bounding_box_1(stamen_with_one_bag_starts,1)+bounding_box_3(stamen_with
_one_bag_starts,1)+5 bounding_box_2(stamen_with_one_bag_starts,1)-5
bounding_box_2(stamen_with_one_bag_starts,1)+bounding_box_4(stamen_with
_one_bag_starts,1)+5] )
                subplot(2,5,6),
imshow(read(my_movie,one_bag_with_stamen_ends)), h=title({['Drop When
Bag Breaks'],['Frame Num: ',
num2str(one_bag_with_stamen_ends)]});set(h,'FontSize',6);
                    axis([bounding_box_1(one_bag_with_stamen_ends-1,1)-
50 bounding_box_1(one_bag_with_stamen_ends-
1,1)+bounding_box_3(one_bag_with_stamen_ends-1,1)+50
bounding_box_2(one_bag_with_stamen_ends-1,1)-50
bounding_box_2(one_bag_with_stamen_ends-
1,1)+bounding_box_4(one_bag_with_stamen_ends-1,1)+50] )
                subplot(2,5,7),
imshow(read(my_movie,frame_num_only_rim)),h=title({['Drop When Only Rim
is Present'],['Frame Num: ',
num2str(frame_num_only_rim)]});set(h,'FontSize',6);
                    axis([bounding_box_1(one_bag_with_stamen_ends-1,1)-
300 bounding_box_1(one_bag_with_stamen_ends-
1,1)+bounding_box_3(one_bag_with_stamen_ends-1,1)+300
bounding_box_2(one_bag_with_stamen_ends-1,1)-300
bounding_box_2(one_bag_with_stamen_ends-
1,1)+bounding_box_4(one_bag_with_stamen_ends-1,1)+300] )
            end
        end
        if num_bags == 2;
            if strcmp('n',stamen_present);
                subplot(2,5,4),
imshow(read(my_movie,first_bag_of_two_no_stamen_starts)),
```

```
h=title({['Drop When Top Bag Starts'],['Frame Num: ',
num2str(first_bag_of_two_no_stamen_starts)]});set(h,'FontSize',6);

axis([bounding_box_1(first_bag_of_two_no_stamen_starts,1)-5
bounding_box_1(first_bag_of_two_no_stamen_starts,1)+bounding_box_3(firs
t_bag_of_two_no_stamen_starts,1)+5
bounding_box_2(first_bag_of_two_no_stamen_starts,1)-5
bounding_box_2(first_bag_of_two_no_stamen_starts,1)+bounding_box_4(firs
t_bag_of_two_no_stamen_starts,1)+5] )
                subplot(2,5,5),
imshow(read(my_movie,second_bag_of_two_no_stamen_starts)),h=title({['Dr
op When Bottom Bag Starts'],['Frame Num: ',
num2str(second_bag_of_two_no_stamen_starts)]});set(h,'FontSize',6);

axis([bounding_box_1(second_bag_of_two_no_stamen_starts,1)-5
bounding_box_1(second_bag_of_two_no_stamen_starts,1)+bounding_box_3(sec
ond_bag_of_two_no_stamen_starts,1)+5
bounding_box_2(second_bag_of_two_no_stamen_starts,1)-5
bounding_box_2(second_bag_of_two_no_stamen_starts,1)+bounding_box_4(sec
ond_bag_of_two_no_stamen_starts,1)+5] )
                if
first_bag_of_two_no_stamen_ends>second_bag_of_two_no_stamen_ends;
                    reference_point = first_bag_of_two_no_stamen_ends;
                else
                    reference_point = second_bag_of_two_no_stamen_ends;
                end
                subplot(2,5,6),
imshow(read(my_movie,first_bag_of_two_no_stamen_ends)), h=title({['Drop
When Top Bag Breaks'],['Frame Num: ',
num2str(first_bag_of_two_no_stamen_ends)]});set(h,'FontSize',6);
                    axis([bounding_box_1(reference_point-1,1)-50
bounding_box_1(reference_point-1,1)+bounding_box_3(reference_point-
1,1)+50 bounding_box_2(reference_point-1,1)-50
bounding_box_2(reference_point-1,1)+bounding_box_4(reference_point-
1,1)+50] )
                subplot(2,5,7),
imshow(read(my_movie,second_bag_of_two_no_stamen_ends)),
h=title({['Drop When Bottom Bag Ends'],['Frame Num: ',
num2str(second_bag_of_two_no_stamen_ends)]});set(h,'FontSize',6);
                     axis([bounding_box_1(reference_point-1,1)-50
bounding_box_1(reference_point-1,1)+bounding_box_3(reference_point-
1,1)+50 bounding_box_2(reference_point-1,1)-50
bounding_box_2(reference_point-1,1)+bounding_box_4(reference_point-
1,1)+50] )
                subplot(2,5,8),
imshow(read(my_movie,frame_num_only_rim)), h=title({['Drop When Only
Rim is Present'],['Frame Num: ',
num2str(frame_num_only_rim)]});set(h,'FontSize',6);
                    axis([bounding_box_1(reference_point-1,1)-300
bounding_box_1(reference_point-1,1)+bounding_box_3(reference_point-
1,1)+300 bounding_box_2(reference_point-1,1)-200
bounding_box_2(reference_point-1,1)+bounding_box_4(reference_point-
1,1)+300] )
            end
            if strcmp('y',stamen_present);
```

```
                subplot(2,5,4),
imshow(read(my_movie,first_bag_of_two_with_stamen_starts)),
h=title({['Drop When Top Bag Starts'],['Frame Num: ',
num2str(first_bag_of_two_with_stamen_starts)]});set(h,'FontSize',6);

axis([bounding_box_1(first_bag_of_two_with_stamen_starts,1)-5
bounding_box_1(first_bag_of_two_with_stamen_starts,1)+bounding_box_3(fi
rst_bag_of_two_with_stamen_starts,1)+5
bounding_box_2(first_bag_of_two_with_stamen_starts,1)-5
bounding_box_2(first_bag_of_two_with_stamen_starts,1)+bounding_box_4(fi
rst_bag_of_two_with_stamen_starts,1)+5] )
                subplot(2,5,5),
imshow(read(my_movie,second_bag_of_two_with_stamen_starts)),
title('');h=title({['Drop When Bottom Bag Starts'],['Frame Num: ',
num2str(second_bag_of_two_with_stamen_starts)]});set(h,'FontSize',6);

axis([bounding_box_1(second_bag_of_two_with_stamen_starts,1)-5
bounding_box_1(second_bag_of_two_with_stamen_starts,1)+bounding_box_3(s
econd_bag_of_two_with_stamen_starts,1)+5
bounding_box_2(second_bag_of_two_with_stamen_starts,1)-5
bounding_box_2(second_bag_of_two_with_stamen_starts,1)+bounding_box_4(s
econd_bag_of_two_with_stamen_starts,1)+5] )
                subplot(2,5,6),
imshow(read(my_movie,stamen_with_two_bags_starts)), h=title({['Drop
When Stamen Emerges'],['Frame Num: ',
num2str(stamen_with_two_bags_starts)]});set(h,'FontSize',6);

axis([bounding_box_1(stamen_with_two_bags_starts,1)-5
bounding_box_1(stamen_with_two_bags_starts,1)+bounding_box_3(stamen_wit
h_two_bags_starts,1)+5 bounding_box_2(stamen_with_two_bags_starts,1)-5
bounding_box_2(stamen_with_two_bags_starts,1)+bounding_box_4(stamen_wit
h_two_bags_starts,1)+5] )
                if
first_bag_of_two_with_stamen_ends>second_bag_of_two_with_stamen_ends;
                    reference_point =
first_bag_of_two_with_stamen_ends;
                else
                    reference_point =
second_bag_of_two_with_stamen_ends;
                end
                subplot(2,5,7),
imshow(read(my_movie,first_bag_of_two_with_stamen_ends)),
h=title({['Drop When Top Bag Breaks'],['Frame Num: ',
num2str(first_bag_of_two_with_stamen_ends)]});set(h,'FontSize',6);
                     axis([bounding_box_1(reference_point-1,1)-50
bounding_box_1(reference_point-1,1)+bounding_box_3(reference_point-
1,1)+50 bounding_box_2(reference_point-1,1)-50
bounding_box_2(reference_point-1,1)+bounding_box_4(reference_point-
1,1)+50] )
                subplot(2,5,8),
imshow(read(my_movie,second_bag_of_two_with_stamen_ends)),
h=title({['Drop When Bottom Bag Ends'],['Frame Num: ',
num2str(second_bag_of_two_with_stamen_ends)]});set(h,'FontSize',6);
                     axis([bounding_box_1(reference_point-1,1)-50
bounding_box_1(reference_point-1,1)+bounding_box_3(reference_point-
```

```
1,1)+50 bounding_box_2(reference_point-1,1)-50
bounding_box_2(reference_point-1,1)+bounding_box_4(reference_point-
1,1)+50] )
                subplot(2,5,9),
imshow(read(my_movie,frame_num_only_rim)), h=title({['Drop When Only
Rim is Present'],['Frame Num: ',
num2str(frame_num_only_rim)]});set(h,'FontSize',6);
                    axis([bounding_box_1(reference_point-1,1)-300
bounding_box_1(reference_point-1,1)+bounding_box_3(reference_point-
1,1)+300 bounding_box_2(reference_point-1,1)-300
bounding_box_2(reference_point-1,1)+bounding_box_4(reference_point-
1,1)+300] )
            end
        end
        s = strcat('Images of Specific Frames in Movie',name,'.png');

set(gcf,'PaperUnits','inches','PaperSize',[15,6],'PaperPosition',[0 0
15 6]);
        print('-dpng','-r200',s);
    time_break_up_3D_bag =0;
    time_break_up_3D_bag_2 =0;
    Non_dimensional_time_break_up_3D_bag =0;
    Non_dimensional_time_break_up_3D_bag_2 =0;
% 3D bag times
    if num_bags ~= num_bags_3D;
        if num_bags_3D==2 && num_bags==1;
            time_break_up_3D_bag = (breakup_frame_3D_bag-
time_zero_frame_num)*1000/frame_speed;%(ms)
            Non_dimensional_time_break_up_3D_bag
=(breakup_frame_3D_bag-
time_zero_frame_num)*1000/frame_speed*Initial_relative_velocity*sqrt(de
nsity_air/density_drop)/(initial_diameter);
        end
        if num_bags_3D==3 && num_bags==1;
            time_break_up_3D_bag = (breakup_frame_3D_bag-
time_zero_frame_num)*1000/frame_speed;
            time_break_up_3D_bag_2 = (breakup_frame_3D_bag_2-
time_zero_frame_num)*1000/frame_speed;
            Non_dimensional_time_break_up_3D_bag
=(breakup_frame_3D_bag-
time_zero_frame_num)*1000/frame_speed*Initial_relative_velocity*sqrt(de
nsity_air/density_drop)/(initial_diameter);
            Non_dimensional_time_break_up_3D_bag_2
=(breakup_frame_3D_bag_2-
time_zero_frame_num)*1000/frame_speed*Initial_relative_velocity*sqrt(de
nsity_air/density_drop)/(initial_diameter);
        end
        if num_bags_3D==3 && num_bags==2;
            time_break_up_3D_bag = (breakup_frame_3D_bag-
time_zero_frame_num)*1000/frame_speed;
            Non_dimensional_time_break_up_3D_bag
=(breakup_frame_3D_bag-
time_zero_frame_num)*1000/frame_speed*Initial_relative_velocity*sqrt(de
nsity_air/density_drop)/(initial_diameter);
        end
```

```
    end
% Initiation time based on last min axial length
    time_based_on_last_min_axial_length =
data_x_axial(location_min_axial_length_based_on_last_min);
    Non_dimensional_time_based_on_last_min_axial_length =
Non_dimensional_data_x_axial(location_min_axial_length_based_on_last_mi
n);
% Initiation time based on min bounding box
    time_based_on_min_bounding_box =
data_x_axial(location_max_inv_bounding_box_based_on_last_max);
    Non_dimensional_time_based_on_min_bounding_box =
Non_dimensional_data_x_axial(location_max_inv_bounding_box_based_on_las
t_max);
    Tbag =0;
    Tbag_top=0;
    Tbag_bottom=0;
    length_stamen_protruding_when_bag_breaks=0;
    ND_length_stamen_protruding_when_bag_breaks=0;
    Non_dimensional_Tbag=0;
% Bag Breakup Time
    if num_bags==0;
        Tbag = (zero_bags_ends-time_zero_frame_num)*1000/frame_speed;
        Non_dimensional_Tbag = (zero_bags_ends-
time_zero_frame_num)*1000/frame_speed*Initial_relative_velocity*sqrt(de
nsity_air/density_drop)/(initial_diameter);
        when_bag_ends = zero_bags_ends-1;
    end
    if num_bags ==1;
        if strcmp('n',stamen_present);
            Tbag = (only_one_bag_ends-
time_zero_frame_num)*1000/frame_speed;
            Non_dimensional_Tbag = (only_one_bag_ends-
time_zero_frame_num)*1000/frame_speed*Initial_relative_velocity*sqrt(de
nsity_air/density_drop)/(initial_diameter);
            when_bag_ends = only_one_bag_ends-1;
        end
        if strcmp('y',stamen_present);
            Tbag = (one_bag_with_stamen_ends-
time_zero_frame_num)*1000/frame_speed;
            Non_dimensional_Tbag = (one_bag_with_stamen_ends-
time_zero_frame_num)*1000/frame_speed*Initial_relative_velocity*sqrt(de
nsity_air/density_drop)/(initial_diameter);
            when_bag_ends = one_bag_with_stamen_ends-1;
            length_stamen_protruding_when_bag_breaks =
stamen_length_protruding(when_bag_ends);
            ND_length_stamen_protruding_when_bag_breaks=
Non_dimensional_stamen_length_protruding(when_bag_ends);
        end
    end
    if num_bags ==2;
        if strcmp('n',stamen_present);
            Tbag_top = (first_bag_of_two_no_stamen_ends-
time_zero_frame_num)*1000/frame_speed;
            Non_dimensional_Tbag_top =
(first_bag_of_two_no_stamen_ends-
```

```
time_zero_frame_num)*1000/frame_speed*Initial_relative_velocity*sqrt(de
nsity_air/density_drop)/(initial_diameter);
            Tbag_bottom = (second_bag_of_two_no_stamen_ends-
time_zero_frame_num)*1000/frame_speed;
            Non_dimensional_Tbag_bottom =
(second_bag_of_two_no_stamen_ends-
time_zero_frame_num)*1000/frame_speed*Initial_relative_velocity*sqrt(de
nsity_air/density_drop)/(initial_diameter);
            when_top_bag_ends = first_bag_of_two_no_stamen_ends-1;
            when_bottom_bag_ends = second_bag_of_two_no_stamen_ends-1;
            if
first_bag_of_two_no_stamen_ends<second_bag_of_two_no_stamen_ends;
                when_bag_ends=second_bag_of_two_no_stamen_ends-1;
            else
                when_bag_ends=first_bag_of_two_no_stamen_ends-1;
            end
        end
        if strcmp('y',stamen_present);
            Tbag_top = (first_bag_of_two_with_stamen_ends-
time_zero_frame_num)*1000/frame_speed;
            Non_dimensional_Tbag_top =
(first_bag_of_two_with_stamen_ends-
time_zero_frame_num)*1000/frame_speed*Initial_relative_velocity*sqrt(de
nsity_air/density_drop)/(initial_diameter);
            Tbag_bottom = (second_bag_of_two_with_stamen_ends-
time_zero_frame_num)*1000/frame_speed;
            Non_dimensional_Tbag_bottom =
(second_bag_of_two_with_stamen_ends-
time_zero_frame_num)*1000/frame_speed*Initial_relative_velocity*sqrt(de
nsity_air/density_drop)/(initial_diameter);
            when_top_bag_ends = first_bag_of_two_with_stamen_ends-1;
            when_bottom_bag_ends = second_bag_of_two_with_stamen_ends-
1;
            if
first_bag_of_two_with_stamen_ends<second_bag_of_two_with_stamen_ends;
                when_bag_ends=second_bag_of_two_with_stamen_ends-1;
            end
            if
first_bag_of_two_with_stamen_ends>=second_bag_of_two_with_stamen_ends;
                when_bag_ends=first_bag_of_two_with_stamen_ends-1;
            end
            length_stamen_protruding_when_bag_breaks =
stamen_length_protruding(when_bag_ends);
            ND_length_stamen_protruding_when_bag_breaks=
Non_dimensional_stamen_length_protruding(when_bag_ends);
        end
    end
% Time when a bag is no longer present
    Tend_bag = (frame_num_only_rim-
time_zero_frame_num)*1000/frame_speed;
    Non_dimensional_Tend_bag =(frame_num_only_rim-
time_zero_frame_num)*1000/frame_speed*Initial_relative_velocity*sqrt(de
nsity_air/density_drop)/(initial_diameter);
% Axial length when Bag breaks
    if num_bags==0 || num_bags==1;
```

```
        min_axial_length_when_bag_breaks = Axial_length(when_bag_ends);
        Non_dimensional_min_axial_length_when_bag_breaks =
Non_dimensional_axial_length(when_bag_ends);
    else
        min_axial_length_when_bag_breaks = 0;
        Non_dimensional_min_axial_length_when_bag_breaks = 0;
    end
    if num_bags==2;
        min_axial_length_when_top_bag_breaks =
Axial_length_top_bag(when_top_bag_ends);
        Non_dimensional_min_axial_length_when_top_bag_breaks =
Non_dimensional_axial_length_top_bag(when_top_bag_ends);
        min_axial_length_when_bottom_bag_breaks =
Axial_length_bottom_bag(when_bottom_bag_ends);
        Non_dimensional_min_axial_length_when_bottom_bag_breaks =
Non_dimensional_axial_length_bottom_bag(when_bottom_bag_ends);
    end
% Transverse length when Bag breaks
    transverse_length_when_bag_breaks = Column_length(when_bag_ends);
    Non_dimensional_transverse_length_when_bag_breaks =
Non_dimensional_column_length(when_bag_ends);
% Displacement axial when Bag breaks
    displacement_axial_when_bag_breaks = data_y_axial(when_bag_ends);
    Non_dimensional_displace_axial_when_bag_breaks =
Non_dimensional_data_y_axial(when_bag_ends);
% Displacement transverse when Bag breaks
    displacement_transverse_when_bag_breaks =
data_y_transverse(when_bag_ends);
    Non_dimensional_dis_transverse_when_bag_breaks =
Non_dimensional_data_y_transverse(when_bag_ends);
% Velocity axial when Bag breaks
    velocity_axial_when_bag_breaks =
polynomial_velocity(when_bag_ends);
    Non_dimensional_velocity_axial_when_bag_breaks =
Non_dimensional_velocity_axial_from_poly(when_bag_ends);
% Velocity transverse when Bag breaks
    velocity_transverse_when_bag_breaks =
velocity_transverse_from_data(when_bag_ends);
    Non_dimensional_velocity_transverse_when_bag_breaks =
Non_dimensional_velocity_transverse_from_data(when_bag_ends);
% Acceleration axial when Bag breaks
    acceleration_axial_when_bag_breaks =
acceleration_axial_from_poly(when_bag_ends);
    Non_dimensional_acc_axial_when_bag_breaks =
Non_dimensional_acceleration_axial_from_poly(when_bag_ends);
% Cd when Bag breaks
    Cd_when_bag_breaks = Cd_from_data(when_bag_ends);
% Area when Bag breaks
    area_when_bag_breaks = Drop_Area(when_bag_ends);
    Non_Dimensional_area_when_bag_breaks =
Non_dimensional_drop_area(when_bag_ends);
% Transverse length based on last min axial length
    transverse_based_on_last_min_axial_length =
Column_length(location_min_axial_length_based_on_last_min);
```

```
    Non_dimensional_transverse_based_on_last_min_axial_length =
Non_dimensional_column_length(location_min_axial_length_based_on_last_m
in);
% Transverse length based on min bounding box
    transverse_based_on_min_bounding_box =
Column_length(location_max_inv_bounding_box_based_on_last_max);
    Non_dimensional_transverse_based_on_min_bounding_box =
Non_dimensional_column_length(location_max_inv_bounding_box_based_on_la
st_max);
% Displacement axial based on last min axial length
    displacement_axial_based_on_last_min_axial_length =
data_y_axial(location_min_axial_length_based_on_last_min);
    Non_dimensional_displace_axial_based_on_last_min_axial_length =
Non_dimensional_data_y_axial(location_min_axial_length_based_on_last_mi
n);
% Displacement axial based on min bounding box
    displacement_axial_based_on_min_bounding_box=
data_y_axial(location_max_inv_bounding_box_based_on_last_max);
    Non_dimensional_displacement_axial_based_on_min_bounding_box=
Non_dimensional_data_y_axial(location_max_inv_bounding_box_based_on_las
t_max);
% Displacement transverse based on last min axial length
    displacement_transverse_based_on_last_min_axial_length =
data_y_transverse(location_min_axial_length_based_on_last_min);
    Non_dimensional_dis_transverse_based_on_last_min_axial_length =
Non_dimensional_data_y_transverse(location_min_axial_length_based_on_la
st_min);
% Displacement transverse based on min bounding box
    displacement_transverse_based_on_min_bounding_box =
data_y_transverse(location_max_inv_bounding_box_based_on_last_max);
    Non_dimensional_dis_transverse_based_on_min_bounding_box =
Non_dimensional_data_y_transverse(location_max_inv_bounding_box_based_o
n_last_max);
% Velocity axial based on last min axial length
    velocity_axial_based_on_last_min_axial_length =
polynomial_velocity(location_min_axial_length_based_on_last_min);
    Non_dimensional_velocity_axial_based_on_last_min_axial_length =
Non_dimensional_velocity_axial_from_poly(location_min_axial_length_base
d_on_last_min);
% Velocity axial based on min bounding box
    velocity_axial_based_on_min_bounding_box =
polynomial_velocity(location_max_inv_bounding_box_based_on_last_max);
    Non_dimensional_velocity_axial_based_on_min_bounding_box =
Non_dimensional_velocity_axial_from_poly(location_max_inv_bounding_box_
based_on_last_max);
% Velocity transverse based on last min axial length
    velocity_transverse_based_on_last_min_axial_length =
velocity_transverse_from_data(location_min_axial_length_based_on_last_m
in);
    Non_dimensional_velocity_transverse_last_min_axial_length =
Non_dimensional_velocity_transverse_from_data(location_min_axial_length
_based_on_last_min);
% Velocity transverse based on min bounding box
```

```
    velocity_transverse_based_on_min_bounding_box =
velocity_transverse_from_data(location_max_inv_bounding_box_based_on_la
st_max);
    Non_dimensional_velocity_transverse_based_on_min_bounding_box =
Non_dimensional_velocity_transverse_from_data(location_max_inv_bounding
_box_based_on_last_max);
% Acceleration axial based on last min axial length
    acceleration_axial_based_on_last_min_axial_length =
acceleration_axial_from_poly(location_min_axial_length_based_on_last_mi
n);
    Non_dimensional_acc_axial_based_on_last_min_axial_length =
Non_dimensional_acceleration_axial_from_poly(location_min_axial_length_
based_on_last_min);
% Acceleration axial based on min bounding box
    acceleration_axial_based_on_min_bounding_box =
acceleration_axial_from_poly(location_max_inv_bounding_box_based_on_las
t_max);
    Non_dimensional_acceleration_axial_based_on_min_bounding_box =
Non_dimensional_acceleration_axial_from_poly(location_max_inv_bounding_
box_based_on_last_max);
% Rayleigh Taylor Wave Number based on Min Axial Length
    RT_number_based_on_min_axial_length =
(transverse_based_on_last_min_axial_length/(2000*pi))*sqrt(density_drop
*acceleration_axial_based_on_last_min_axial_length*1000/(3*surface_tens
ion));
% Rayleigh Taylor Wave Number based on Min Box
    RT_number_based_on_bounding_box =
(transverse_based_on_min_bounding_box/(2000*pi))*sqrt(density_drop*acce
leration_axial_based_on_min_bounding_box*1000/(3*surface_tension));
% RT number correlation
    RT_num_correlation =
0.076*(1+0.19*sqrt(We_number))^2*sqrt(We_number);
% Cd based on last min axial length
    Cd_based_on_last_min_axial_length =
Cd_from_data(location_min_axial_length_based_on_last_min);
    Cd_spheroid_on_last_min_axial_length =
Cd_spheroid(location_min_axial_length_based_on_last_min);
    Cd_spheroid_near_critical_min_axial_length =
Cd_spheroid_near_critical(location_min_axial_length_based_on_last_min);
% Cd based on min bounding box
    Cd_based_on_min_bounding_box =
Cd_from_data(location_max_inv_bounding_box_based_on_last_max);
    Cd_spheroid_on_min_bounding_box =
Cd_spheroid(location_max_inv_bounding_box_based_on_last_max);
    Cd_spheroid_near_critical_min_bounding_box =
Cd_spheroid_near_critical(location_max_inv_bounding_box_based_on_last_m
ax);
% Area based on last min axial length
    area_based_on_last_min_axial_length =
Drop_Area(location_min_axial_length_based_on_last_min);
    Non_Dimensional_area_based_on_last_min_axial_length =
Non_dimensional_drop_area(location_min_axial_length_based_on_last_min);
% Area based on min bounding box
    area_based_on_min_bounding_box =
Drop_Area(location_max_inv_bounding_box_based_on_last_max);
```

```
    Non_Dimensional_area_based_on_min_bounding_box =
Non_dimensional_drop_area(location_max_inv_bounding_box_based_on_last_m
ax);
% Instantaneous Ohnesorge number
    inst_Oh_num_min_axial =
inst_Oh_num(location_min_axial_length_based_on_last_min);
    inst_Oh_num_min_box =
inst_Oh_num(location_max_inv_bounding_box_based_on_last_max);
    inst_Oh_num_when_bag_breaks = inst_Oh_num(when_bag_ends);
% Instantaneous Galilei number with acceleration from gravity
    inst_Galilei_num_min_axial =
inst_Galilei_num(location_min_axial_length_based_on_last_min);
    inst_Galilei_num_min_box =
inst_Galilei_num(location_max_inv_bounding_box_based_on_last_max);
    inst_Galilei_num_when_bag_breaks = inst_Galilei_num(when_bag_ends);
% Instananeous Galilei number with instant drop acceleration
    inst_Galilei_num_based_on_inst_acc_min_axial =
inst_Galilei_num_based_on_inst_acc(location_min_axial_length_based_on_l
ast_min);
    inst_Galilei_num_based_on_inst_acc_min_box =
inst_Galilei_num_based_on_inst_acc(location_max_inv_bounding_box_based_
on_last_max);
    inst_Galilei_num_based_on_inst_acc_when_bag_breaks =
inst_Galilei_num_based_on_inst_acc(when_bag_ends);
% Instantaneous liquid reynolds number
    inst_Liquid_Re_num_min_axial =
inst_Liquid_Re_num(location_min_axial_length_based_on_last_min);
    inst_Liquid_Re_num_min_box =
inst_Liquid_Re_num(location_max_inv_bounding_box_based_on_last_max);
    inst_Liquid_Re_num_when_bag_breaks =
inst_Liquid_Re_num(when_bag_ends);
% Instantaneous Weber number
    inst_We_number_min_axial =
inst_We_number(location_min_axial_length_based_on_last_min);
    inst_We_number_min_box =
inst_We_number(location_max_inv_bounding_box_based_on_last_max);
    inst_We_number_when_bag_breaks = inst_We_number(when_bag_ends);
% Instantaneous New non-dimensional number
    inst_New_non_dimensional_num_min_axial =
inst_New_non_dimensional_num(location_min_axial_length_based_on_last_mi
n);
    inst_New_non_dimensional_num_min_box =
inst_New_non_dimensional_num(location_max_inv_bounding_box_based_on_las
t_max);
    inst_New_non_dimensional_num_when_bag_breaks =
inst_New_non_dimensional_num(when_bag_ends);
% Instantaneous New non-dimensional number
    inst_Non_dimensional_data_x_axial_min_axial =
inst_Non_dimensional_data_x_axial(location_min_axial_length_based_on_la
st_min);
    inst_Non_dimensional_data_x_axial_min_box =
inst_Non_dimensional_data_x_axial(location_max_inv_bounding_box_based_o
n_last_max);
    inst_Non_dimensional_data_x_axial_when_bag_breaks =
inst_Non_dimensional_data_x_axial(when_bag_ends);
```

```
% Find the Coefficient of Drag which fit velocity data
    clear y Ud y_rate t tstar R R_squared_velocity_and_program
R_squared_radius_and_program R_over_R0
    R_0=initial_diameter/(2000);% drop initial spherical radius [m]
    Cd_sphere=0.445; % drag coefficient for a sphere forRe>=500
    dt=1/(frame_speed); % step size
    R_squared_velocity_and_program=zeros(10.0*100,1);
    R_squared_radius_and_program=zeros(10.0*100,1);
    for g=0.01:0.01:10.0;
    % Defining initial conditions
        t(1,1)=0; % initial time [s]
        y(1,1)=0; % initial dimensionless displacement
        y_rate(1,1)=0; % initial dimensionless velocity
        Ud(1,1)=0; % initial drop velocity [m/s]
        R(1,1)=R_0; % initial drop cross-stream radius[m]
        tstar(1,1)=0; % initial dimensionless breakup time
        i=1;
        p=0;
    % Calculating drop velocity using 4th order Runge-Kutta method
        while
y(i,1)<(2*(Non_dimensional_transverse_based_on_min_bounding_box-1));
        % Velocity Approximation 1
            Cd = Cd_sphere*(1+g*y(i,1));
            Ud_int=Ud(i,1);

U_1=dt*(3/8)*(density_air/density_drop)*(Initial_relative_velocity-
Ud_int)^2*(R(i,1)^2/R_0^3)*Cd;
        % Velocity Approximation 2
            Ud_int=Ud(i,1)+U_1/2;

U_2=dt*(3/8)*(density_air/density_drop)*(Initial_relative_velocity-
Ud_int)^2*(R(i,1)^2/R_0^3)*Cd;
        % Velocity Approximation 3
            Ud_int=Ud(i,1)+U_2/2;

U_3=dt*(3/8)*(density_air/density_drop)*(Initial_relative_velocity-
Ud_int)^2*(R(i,1)^2/R_0^3)*Cd;
        % Velocity Approximation 4
            Ud_int=Ud(i,1)+U_3;

U_4=dt*(3/8)*(density_air/density_drop)*(Initial_relative_velocity-
Ud_int)^2*(R(i,1)^2/R_0^3)*Cd;
        % Velocity Sum Approximation
            Ud(i+1,1)=Ud(i,1)+(1/6)*(U_1+2*U_2+2*U_3+U_4);
        %Computing non-dimensional deformation (y) using 4th order
using Runge-Kutta method
        % Non Dimensional Displacement Approximation 1
            y_int=y(i,1);
            y_rate_int=y_rate(i,1);
            y_1=dt*y_rate_int;
            y_rate_1=dt*(2*density_air*(Initial_relative_velocity-
Ud(i,1))^2*(1+0.5*y_int)^2/(3*density_drop*R(i,1)^2)-
8*surface_tension*y_int/(density_drop*R(i,1)^3)-
5*Viscosity*y_rate_int/(density_drop*R(i,1)^2));
        % Non Dimensional Displacement Approximation 2
```

```
            y_int=y(i,1)+y_1/2;
            y_rate_int=y_rate(i,1)+y_rate_1/2;
            y_2=dt*y_rate_int;
            y_rate_2=dt*(2*density_air*(Initial_relative_velocity-
Ud(i,1))^2*(1+0.5*y_int)^2/(3*density_drop*R(i,1)^2)-
8*surface_tension*y_int/(density_drop*R(i,1)^3)-
5*Viscosity*y_rate_int/(density_drop*R(i,1)^2));
        % Non Dimensional Displacement Approximation 3
            y_int=y(i,1)+y_2/2;
            y_rate_int=y_rate(i,1)+y_rate_2/2;
            y_3=dt*y_rate_int;
            y_rate_3=dt*(2*density_air*(Initial_relative_velocity-
Ud(i,1))^2*(1+0.5*y_int)^2/(3*density_drop*R(i,1)^2)-
8*surface_tension*y_int/(density_drop*R(i,1)^3)-
5*Viscosity*y_rate_int/(density_drop*R(i,1)^2));
        % Non Dimensional Displacement Approximation 4
            y_int=y(i,1)+y_3;
            y_rate_int=y_rate(i,1)+y_rate_3;
            y_4=dt*y_rate_int;
            y_rate_4=dt*(2*density_air*(Initial_relative_velocity-
Ud(i,1))^2*(1+0.5*y_int)^2/(3*density_drop*R(i,1)^2)-
8*surface_tension*y_int/(density_drop*R(i,1)^3)-
5*Viscosity*y_rate_int/(density_drop*R(i,1)^2));
        % Non Dimensional Displacement Sum Approximation
            t(i+1,1)=t(i,1)+dt;

tstar(i+1,1)=t(i+1,1)*Initial_relative_velocity*sqrt(density_air/densit
y_drop)/(initial_diameter/1000);
            y(i+1,1)=y(i,1)+(1/6)*(y_1+2*y_2+2*y_3+y_4);

y_rate(i+1,1)=y_rate(i,1)+(1/6)*(y_rate_1+2*y_rate_2+2*y_rate_3+y_rate_
4);
            R(i+1,1)=R_0*(1+0.5*y(i+1));
            if p==num_data_points+1;
                break
            end
            p=1+p;
            i=i+1;
        end
    % Calculate Residuals between experimental and predicted velocities
        sum_of_squared_velocity_program_residuals=0;
        sum_of_squared_velocity_program_diff_from_mean=0;
        if length(Ud)<(location_max_inv_bounding_box_based_on_last_max-
time_zero_frame_num+1);
            reference_length = length(Ud)+time_zero_frame_num-1;
        elseif
length(Ud)>=(location_max_inv_bounding_box_based_on_last_max-
time_zero_frame_num+1) && length(Ud)<=nnz(polynomial_velocity);
            reference_length =
location_max_inv_bounding_box_based_on_last_max;
        elseif length(Ud)>nnz(polynomial_velocity);
            reference_length =
location_max_inv_bounding_box_based_on_last_max;
        end
```

```
        mean_velocity_program =
mean(polynomial_velocity(time_zero_frame_num:reference_length,1));
        for r=(time_zero_frame_num:reference_length);
            sum_of_squared_velocity_program_residuals =
(polynomial_velocity(r,1)-Ud((r-
time_zero_frame_num+1),1))^2+sum_of_squared_velocity_program_residuals;
            sum_of_squared_velocity_program_diff_from_mean =
(polynomial_velocity(r,1)-
mean_velocity_program)^2+sum_of_squared_velocity_program_diff_from_mean
;
        end
        R_squared_velocity_and_program(int16(g*100),1) = 1 -
sum_of_squared_velocity_program_residuals/sum_of_squared_velocity_progr
am_diff_from_mean;
    % Calculate Residuals between experimental and predicted radius
        sum_of_squared_radius_program_residuals=0;
        sum_of_squared_radius_program_diff_from_mean=0;
        mean_radius_program =
mean(Column_length(time_zero_frame_num:reference_length,1))/2000;
        for r=(time_zero_frame_num:reference_length);
            sum_of_squared_radius_program_residuals =
(Column_length(r,1)/2000-R(r-
time_zero_frame_num+1,1))^2+sum_of_squared_radius_program_residuals;
            sum_of_squared_radius_program_diff_from_mean =
(Column_length(r,1)/2000-
mean_radius_program)^2+sum_of_squared_radius_program_diff_from_mean;
        end
        R_squared_radius_and_program(int16(g*100),1) = 1 -
sum_of_squared_radius_program_residuals/sum_of_squared_radius_program_d
iff_from_mean;
    end
% Find best coefficient of drag by comparing to experimental data
    max_R_squared_velocity = max(R_squared_velocity_and_program);
    [peak_R_squared_Velocity,location_peak_R_squared_Velocity] =
findpeaks(R_squared_velocity_and_program,'MINPEAKHEIGHT',max_R_squared_
velocity-0.1,'NPEAKS',1);
    R_squared_program_velocity = peak_R_squared_Velocity;
    Coefficient_for_CD_using_velocity =
location_peak_R_squared_Velocity/100;
    max_R_squared_radius = max(R_squared_radius_and_program);
    [peak_R_squared_radius,location_peak_R_squared_radius] =
findpeaks(R_squared_radius_and_program,'MINPEAKHEIGHT',max_R_squared_ra
dius-0.1,'NPEAKS',1);
    R_squared_program_radius = peak_R_squared_radius;
    Coefficient_for_CD_using_radius =
location_peak_R_squared_radius/100;
    clear y Ud y_rate t tstar R Cd R_over_R0
    i=1;
    p=0;
% Calculate Velocity using 4th Order Runge-Kutta Method
% Defining initial conditions
    t(1,1)=0; %initial time [s]
    y(1,1)=0; %initial dimensionless displacement
    y_rate(1,1)=0; %initial dimensionless velocity
    Ud(1,1)=0; %initial drop velocity [m/s]
```

```
    R(1,1)=R_0; %initial drop cross-stream radius[m]
    R_over_R0(1,1)=(1+0.5*y(1,1));
    tstar(1,1)=0; %initial dimensionless breakup time
    dt=1/(10*frame_speed);
    while
y(i,1)<(2*(Non_dimensional_transverse_based_on_min_bounding_box-1));
    % Velocity Approximation 1
        Cd(i,1) =
Cd_sphere*(1+Coefficient_for_CD_using_velocity*y(i,1));
        Ud_int=Ud(i,1);

U_1=dt*(3/8)*(density_air/density_drop)*(Initial_relative_velocity-
Ud_int)^2*(R(i,1)^2/R_0^3)*Cd(i,1);
    % Velocity Approximation 2
        Ud_int=Ud(i,1)+U_1/2;

U_2=dt*(3/8)*(density_air/density_drop)*(Initial_relative_velocity-
Ud_int)^2*(R(i,1)^2/R_0^3)*Cd(i,1);
    % Velocity Approximation 3
        Ud_int=Ud(i,1)+U_2/2;

U_3=dt*(3/8)*(density_air/density_drop)*(Initial_relative_velocity-
Ud_int)^2*(R(i,1)^2/R_0^3)*Cd(i,1);
    % Velocity Approximation 4
        Ud_int=Ud(i,1)+U_3;

U_4=dt*(3/8)*(density_air/density_drop)*(Initial_relative_velocity-
Ud_int)^2*(R(i,1)^2/R_0^3)*Cd(i,1);
    % Velocity Sum Approximation
        Ud(i+1,1)=Ud(i,1)+(1/6)*(U_1+2*U_2+2*U_3+U_4);
    %Computing non-dimensional deformation (y) using 4th order Runge-
Kutta method
    % Non Dimensional Displacement Approximation 1
        y_int=y(i,1);
        y_rate_int=y_rate(i,1);
        y_1=dt*y_rate_int;
        y_rate_1=dt*(2*density_air*(Initial_relative_velocity-
Ud(i,1))^2*(1+0.5*y_int)^2/(3*density_drop*R(i,1)^2)-
8*surface_tension*y_int/(density_drop*R(i,1)^3)-
5*Viscosity*y_rate_int/(density_drop*R(i,1)^2));
    % Non Dimensional Displacement Approximation 2
        y_int=y(i,1)+y_1/2;
        y_rate_int=y_rate(i,1)+y_rate_1/2;
        y_2=dt*y_rate_int;
        y_rate_2=dt*(2*density_air*(Initial_relative_velocity-
Ud(i,1))^2*(1+0.5*y_int)^2/(3*density_drop*R(i,1)^2)-
8*surface_tension*y_int/(density_drop*R(i,1)^3)-
5*Viscosity*y_rate_int/(density_drop*R(i,1)^2));
    % Non Dimensional Displacement Approximation 3
        y_int=y(i,1)+y_2/2;
        y_rate_int=y_rate(i,1)+y_rate_2/2;
        y_3=dt*y_rate_int;
        y_rate_3=dt*(2*density_air*(Initial_relative_velocity-
Ud(i,1))^2*(1+0.5*y_int)^2/(3*density_drop*R(i,1)^2)-
```

```
8*surface_tension*y_int/(density_drop*R(i,1)^3)-
5*Viscosity*y_rate_int/(density_drop*R(i,1)^2));
    % Non Dimensional Displacement Approximation 4
        y_int=y(i,1)+y_3;
        y_rate_int=y_rate(i,1)+y_rate_3;
        y_4=dt*y_rate_int;
        y_rate_4=dt*(2*density_air*(Initial_relative_velocity-
Ud(i,1))^2*(1+0.5*y_int)^2/(3*density_drop*R(i,1)^2)-
8*surface_tension*y_int/(density_drop*R(i,1)^3)-
5*Viscosity*y_rate_int/(density_drop*R(i,1)^2));
    % Non Dimensional Displacement Sum Approximation
        t(i+1,1)=t(i,1)+dt;

tstar(i+1,1)=t(i+1,1)*Initial_relative_velocity*sqrt(density_air/densit
y_drop)/(initial_diameter/1000);
        y(i+1,1)=y(i,1)+(1/6)*(y_1+2*y_2+2*y_3+y_4);

y_rate(i+1,1)=y_rate(i,1)+(1/6)*(y_rate_1+2*y_rate_2+2*y_rate_3+y_rate_
4);
        R(i+1,1)=R_0*(1+0.5*y(i+1,1));
        Cd(i+1,1)=Cd(i,1);
        R_over_R0(i+1,1)=(1+0.5*y(i+1,1));
        if p>1000;
            break
        end
        p=1+p;
        i=i+1;
    end
% Plot Experimental and Predicted Velocity, Radius, and Cd
    figure(7);
        hFig = figure(7);
        set(hFig,'Position', get(0,'Screensize'));
        max_Ud = max(Ud);
        find_max_in_data =
find(velocity_axial_from_data>max_Ud,1,'first');
        if isempty(find_max_in_data);
            find_max_in_data =
nnz(polynomial_velocity)+time_zero_frame_num-1;
        end
    subplot (2,2,1), hold on,
plot(data_x_axial(time_zero_frame_num:(find_max_in_data)),velocity_axia
l_from_data(time_zero_frame_num:find_max_in_data),'or');
    plot(t*1000,Ud,'b-'),xlabel('Time (ms)'),ylabel('Drop Velocity
(mm/ms)'),h=legend('Velocity from Data','Predicted
Velocity','Location','NorthWest');set(h,'FontSize',10), title('Velocity
versus Time'); hold off
    subplot (2,2,2), hold on,
plot(data_x_axial(time_zero_frame_num:find_max_in_data),Non_dimensional
_column_length(time_zero_frame_num:find_max_in_data),'or');
    plot(t*1000,R/R_0,'b-'),xlabel('Time (ms)'),ylabel('Non Dimensonal
Cross Stream Dimension (L/d_0)'),h=legend('Length from Data','Predicted
Length','Location','NorthWest');set(h,'FontSize',10),title('Cross
Stream Dimension versus Time'), hold off;
```

```
    subplot(2,2,3), hold on,
plot(data_x_axial(time_zero_frame_num:find_max_in_data),Cd_from_data(ti
me_zero_frame_num:find_max_in_data),'or');
    plot(t(1:length(Cd))*1000,Cd,'b-'),xlabel('Time
(ms)'),ylabel('Coefficient of Drag'),h=legend('Cd from Data','Predicted
Cd','Location','NorthWest');set(h,'FontSize',10), title('Coefficient of
Drag versus Time'), hold off;
    s = strcat('Model Plots',name,'.png');
    set(gcf,'PaperUnits','inches','PaperSize',[15,6],'PaperPosition',[0
0 15 6]);
    print('-dpng','-r200',s);
    s = strcat(Liquid,'.xls');
    idx=strfind(Current_movie,'.');
    sheet_name=Current_movie(1:idx(3)-4);
    Program_ND_initiation_time =tstar(length(tstar));

Find_position_in_program_velocity=find(tstar>=Non_dimensional_time_base
d_on_min_bounding_box,1,'first');
    if isempty(Find_position_in_program_velocity)
        Find_position_in_program_velocity=length(Ud);
    end
    Program_Velocity_at_exp_ND_Ini =
Ud(Find_position_in_program_velocity);
    Specific_points ={...
        'Movie Name(.avi)',...
        'Type of Breakup',...
        'Mass Flow Rate (kg/min)',...
        'Calibration (Length/Pixel)',...
        'Frame Speed (fps)',...
        'Air Density (kg/m^3)'...
        'Dynamic Air Viscosity (Pa*s)',...
        'Initial Relative Velocity (m/s)',...
        'Initial Diameter (mm)',...
        'Initial Area (mm^2)',...
        'Surface Tension (N/m)',...
        'Density Drop (kg/m^3)',...
        'Newtonian Liquid or Not (n/nn)',...
        'Flow Behav. Index n (*)',...
        'Flow Consis. Index K (Pa*s^n)',...
        'Viscosity (Pa*s)',...
        'We Number (*)',...
        'Uncertainty We (%)',...
        'Oh Number (*)',...
        'Uncertainty Oh (%)',...
        'Liquid Re num (*)',...
        'Uncertainty ReL (%)',...
        'Galilei Number (*)',...
        'Density Ratio (*)',...
        'Viscosity Ratio (*)',...
        'Mach Number (*)',...
        'New ND number (*)',...
        'Num Bags 2D',...
        'Num Bags 3D',...
        'Breakup frame of 3D_bag',...
        'Breakup frame of other 3D bag',...
```

```
        'Time break up 3D bag',...
        'Time break up of other 3Dbag',...
        'Non Dimensional Time Break Up 3D Bag',...
        'Non Dimensional Time Break Up of Other 3D Bag',...
        'Does a Stamen Protrude (y/n)',...
        'Frame when zero bags break (frame)',...
        'Frame when one bag no stamen starts (frame)',...
        'Frame when one bag no stamen ends (frame)',...
        'Frame when one bag with stamen ends (frame)',...
        'Frame when one bag with stamen starts (frame)',...
        'Frame when stamen with one bag protrudes (frame)',...
        'Frame when top bag no stamen starts (frame)',...
        'Frame when top bag no stamen ends (frame)',...
        'Frame when bottom bag no stamen starts (frame)',...
        'Frame when bottom bag no stamen ends (frame)',...
        'Frame when top bag with stamen starts (frame)',...
        'Frame when top bag with stamen ends (frame)',...
        'Frame when bottom bag with stamen starts (frame)',...
        'Frame when bottom bag with stamen ends (frame)',...
        'Frame when stamen with two bags protrudes (frame)',...
        'Last Frame where no data is obtained (frame)',...
        'Frame when only rim is present and bag(s) have broken
(frame)',...
        'Frame when drop enters flow(frame)',...
        'Frame When Last Min Axial Length (frame)',...
        'Frame When Box is Min (frame)',...
        'Initiation Time When Last Min Axial Length (ms)',...
        'Initiation Time When Box is Min (ms)',...
        'ND Initiation Time Based on Min Axial Length (*)',...
        'ND Initiation Time Based on Min Box (*)',...
        'Time From Start When Zero or One Bag Ends (ms)',...
        'ND Time From Start When Zero or One Bag Ends (*)',...
        'Time From Start When Only Rim is Present (ms)',...
        'ND Time From Start When Only Rim is Present (*)',...
        'Min Axial Length Based on Last Min (mm)',...
        'Min Axial Length Based on Min Box (mm)',...
        'ND Min Axial Length Based on Last Min(*)',...
        'ND Min Axial Length Based on Min Box(*)',...
        'Max Cross Stream Dimension Based on Last Min (mm)',...
        'Max Cross Stream Dimension Based on Min Box (mm)',...
        'ND Max Cross Stream Dimension Based on Last Min (*)',...
        'ND Max Cross Stream Dimension Based on Min Box (*)',...
        'Axial Displacement Based on Last Min (mm)',...
        'Axial Displacement Based on Min Box (mm)',...
        'ND Axial Displacement Based on Min Axial Length (*)',...
        'ND Axial Displacement Based on Min Box (*)',...
        'Transverse Displacement Based on Last Min Axial (mm)'...
        'Transverse Displacement Based Min Box (mm)',...
        'ND Transverse Displacement Based on Last Min Axial (*)',...
        'ND Transverse Displacement Based on Min Box (*)',...
        'R squared Axial Velocity (*)',...
        'Axial Velocity Based on Last min (m/s)',...
        'Axial Velocity Based on Min Box (m/s)',...
        'ND Axial Velocity Based on Last Min (*)',...
        'ND Axial Velocity Based on Min Box (*)',...
```

```
'Transverse Velocity Based on Last Min (m/s)',...
'Transverse Velocity Based on Min Box (m/s)',...
'ND Transverse Velocity Based on Last Min (*)',...
'ND Transverse Velocity Based on Min Box (*)',...
'Axial Acceleration Based on Last Min (mm/ms^2)',...
'Axial Acceleration Based on Min Box (mm/ms^2)',...
'ND Axial Acceleration Based on Last Min (*)',...
'ND Axial Acceleration Based on Min Box (*)',...
'Area Based on Last Min (mm^2)',...
'Area Based on Min Box (mm^2)',...
'ND Area Based on Last Min (*)',...
'ND Area Based on Min Box (*)',...
'Cd Based on Min Axial Length (*)',...
'Cd Based on Min Bounding Box (*)',...
'Cd Spheroid Based on Last Min Axial Length (*)',...
'Cd Spheroid Based on Min Bounding Box (*)',...
'Cd Spheroid at Critical on Min Axial (*)',...
'Cd Spheroid at Critical on Min Box (*)',...
'RT num at Min Axial (*)',...
'RT num at Min Box (*)',...
'RT Correlation (*)',...
'Axial Length When Bag Breaks (mm)',...
'ND Axial Length When Bag Breaks (*)',...
'Transverse Length When Bag Breaks (mm)',...
'ND Transverse Length When Bag Breaks (*)',...
'Length Stamen Protruding When Bag Breaks (mm)',...
'ND Length Stamen Protruding When Bag Breaks (*)',...
'Displacement Axial When Bag Breaks (mm)',...
'ND Displacement Axial When Bag Breaks (*)',...
'Displacement Transverse When Bag Breaks (mm)',...
'ND Displacement Transverse When Bag Breaks (*)',...
'Velocity Axial When Bag Breaks (m/s)',...
'ND Velocity Axial When Bag Breaks (*)',...
'Velocity Transverse When Bag Breaks (m/s)',...
'ND Velocity Transverse When Bag Breaks (*)',...
'Acceleration Axial When Bag Breaks (mm/ms^2)',...
'ND Accceleration Axial When Bag Breaks (*)',...
'Cd When Bag Breaks (*)',...
'Area When Bag Breaks (mm^2)',...
'ND Area When Bag Breaks (*)',...
'Inst Oh num at Min Axial (*)',...
'Inst Oh num at Min Box (*)',...
'Inst Ga num at Min Axial (*)',...
'Inst Ga num at Min Box (*)',...
'Inst Ga using Inst Acc num at Min Axial (*)',...
'Inst Ga using Inst Acc num at Min Box (*)',...
'Inst Liquid Re num at Min Axial (*)',...
'Inst Liquid Re num at Min Box (*)',...
'Inst We num at Min Axial (*)',...
'Inst We num at Min Box (*)',...
'Inst New ND num at Min Axial (*)',...
'Inst New ND num at Min Box (*)',...
'Inst ND Time at Min Axial (*)',...
'Inst ND Time at Min Box (*)',...
'Inst Oh num When Bag Breaks (*)',...
```

```
'Inst Ga num When Bag Breaks (*)',...
'Inst Ga num using Inst Acc When Bag Breaks (*)',...
'Inst Liquid Re num When Bag Breaks (*)',...
'Inst We num When Bag Breaks (*)',...
'Inst New ND When Bag Breaks (*)',...
'Inst ND Time When Bag Breaks (*)',...
'Cd for Program using Velocity (*)',...
'R^2 for Program using Velocity (*)',...
'Cd for Program using Radius (*)',...
'R^2 for Program using Radius (*)',...
'Program_ND_initiation_time(*)',...
'Program_Velocity_at_exp_ND_Ini(*)';...
Current_movie,...
Type_of_breakup,...
mass_flow_rate,...
calibration,...
frame_speed,...
density_air,...
dyn_viscosity_air,...
Initial_relative_velocity,...
initial_diameter,...
initial_area,...
surface_tension,...
density_drop,...
Newtonian_or_not,...
flow_behavior_index_n,...
flow_consistency_index_k,...
Viscosity,...
We_number,...
uncertainty_We,...
Oh_number,...
uncertainty_Oh,...
Liquid_Re_num,...
uncertainty_Re_L,...
Galilei_num,...
density_ratio,...
viscosity_ratio,...
mach_number,...
New_non_dimensional_num,...
num_bags,...
num_bags_3D,...
breakup_frame_3D_bag,...
breakup_frame_3D_bag_2,...
time_break_up_3D_bag,...
time_break_up_3D_bag_2,...
Non_dimensional_time_break_up_3D_bag,...
Non_dimensional_time_break_up_3D_bag_2,...
stamen_present,...
zero_bags_ends,...
only_one_bag_starts,...
only_one_bag_ends,...
one_bag_with_stamen_starts,...
one_bag_with_stamen_ends,...
stamen_with_one_bag_starts,...
first_bag_of_two_no_stamen_starts,...
```

```
        first_bag_of_two_no_stamen_ends,...
        second_bag_of_two_no_stamen_starts,...
        second_bag_of_two_no_stamen_ends,...
        first_bag_of_two_with_stamen_starts,...
        first_bag_of_two_with_stamen_ends,...
        second_bag_of_two_with_stamen_starts,...
        second_bag_of_two_with_stamen_ends,...
        stamen_with_two_bags_starts,...
        end_frame,...
        frame_num_only_rim,...
        time_zero_frame_num,...
        location_min_axial_length_based_on_last_min,...
        location_max_inv_bounding_box_based_on_last_max,...
        time_based_on_last_min_axial_length,...
        time_based_on_min_bounding_box,...
        Non_dimensional_time_based_on_last_min_axial_length,...
        Non_dimensional_time_based_on_min_bounding_box,...
        Tbag,...
        Non_dimensional_Tbag,...
        Tend_bag,...
        Non_dimensional_Tend_bag,...
        min_axial_length_based_on_last_min,...
        min_axial_length_based_on_bounding_box,...
        Non_dimensional_min_axial_length_based_on_last_min,...
        Non_dimensional_min_axial_length_based_on_bounding_box,...
        transverse_based_on_last_min_axial_length,...
        transverse_based_on_min_bounding_box,...
        Non_dimensional_transverse_based_on_last_min_axial_length,...
        Non_dimensional_transverse_based_on_min_bounding_box,...
        displacement_axial_based_on_last_min_axial_length,...
        displacement_axial_based_on_min_bounding_box,...

Non_dimensional_displace_axial_based_on_last_min_axial_length,...

Non_dimensional_displacement_axial_based_on_min_bounding_box,...
        displacement_transverse_based_on_last_min_axial_length,...
        displacement_transverse_based_on_min_bounding_box,...

Non_dimensional_dis_transverse_based_on_last_min_axial_length,...
        Non_dimensional_dis_transverse_based_on_min_bounding_box,...
        R_squared_axial_velocity,...
        velocity_axial_based_on_last_min_axial_length,...
        velocity_axial_based_on_min_bounding_box,...

Non_dimensional_velocity_axial_based_on_last_min_axial_length,...
        Non_dimensional_velocity_axial_based_on_min_bounding_box,...
        velocity_transverse_based_on_last_min_axial_length,...
        velocity_transverse_based_on_min_bounding_box,...
        Non_dimensional_velocity_transverse_last_min_axial_length,...

Non_dimensional_velocity_transverse_based_on_min_bounding_box,...
        acceleration_axial_based_on_last_min_axial_length,...
        acceleration_axial_based_on_min_bounding_box,...
        Non_dimensional_acc_axial_based_on_last_min_axial_length,...
```

```
Non_dimensional_acceleration_axial_based_on_min_bounding_box,...
        area_based_on_last_min_axial_length,...
        area_based_on_min_bounding_box,...
        Non_Dimensional_area_based_on_last_min_axial_length,...
        Non_Dimensional_area_based_on_min_bounding_box,...
        Cd_based_on_last_min_axial_length,...
        Cd_based_on_min_bounding_box,...
        Cd_spheroid_on_last_min_axial_length,...
        Cd_spheroid_on_min_bounding_box,...
        Cd_spheroid_near_critical_min_axial_length,...
        Cd_spheroid_near_critical_min_bounding_box,...
        RT_number_based_on_min_axial_length,...
        RT_number_based_on_bounding_box,...
        RT_num_correlation,...
        min_axial_length_when_bag_breaks,...
        Non_dimensional_min_axial_length_when_bag_breaks,...
        transverse_length_when_bag_breaks,...
        Non_dimensional_transverse_length_when_bag_breaks,...
        length_stamen_protruding_when_bag_breaks,...
        ND_length_stamen_protruding_when_bag_breaks,...
        displacement_axial_when_bag_breaks,...
        Non_dimensional_displace_axial_when_bag_breaks,...
        displacement_transverse_when_bag_breaks,...
        Non_dimensional_dis_transverse_when_bag_breaks,...
        velocity_axial_when_bag_breaks,...
        Non_dimensional_velocity_axial_when_bag_breaks,...
        velocity_transverse_when_bag_breaks,...
        Non_dimensional_velocity_transverse_when_bag_breaks,...
        acceleration_axial_when_bag_breaks,...
        Non_dimensional_acc_axial_when_bag_breaks,...
        Cd_when_bag_breaks,...
        area_when_bag_breaks,...
        Non_Dimensional_area_when_bag_breaks,...
        inst_Oh_num_min_axial,...
        inst_Oh_num_min_box,...
        inst_Galilei_num_min_axial,...
        inst_Galilei_num_min_box,...
        inst_Galilei_num_based_on_inst_acc_min_axial,...
        inst_Galilei_num_based_on_inst_acc_min_box,...
        inst_Liquid_Re_num_min_axial,...
        inst_Liquid_Re_num_min_box,...
        inst_We_number_min_axial,...
        inst_We_number_min_box,...
        inst_New_non_dimensional_num_min_axial,...
        inst_New_non_dimensional_num_min_box,...
        inst_Non_dimensional_data_x_axial_min_axial,...
        inst_Non_dimensional_data_x_axial_min_box,...
        inst_Oh_num_when_bag_breaks,...
        inst_Galilei_num_when_bag_breaks,...
        inst_Galilei_num_based_on_inst_acc_when_bag_breaks,...
        inst_Liquid_Re_num_when_bag_breaks,...
        inst_We_number_when_bag_breaks,...
        inst_New_non_dimensional_num_when_bag_breaks,...
        inst_Non_dimensional_data_x_axial_when_bag_breaks,...
```

```
    Coefficient_for_CD_using_velocity,...
    R_squared_program_velocity,...
    Coefficient_for_CD_using_radius,...
    R_squared_program_radius,...
    Program_ND_initiation_time,...
    Program_Velocity_at_exp_ND_Ini};
    xlswrite (s,Specific_points,sheet_name,'A1');
Title_multiple_data_points ={...
    'Time (ms)',...
    'ND Time (*)',...
    'ND Viscous Time (*)',...
    'Axial Length One Bag(mm)',...
    'ND Axial Length One Bag (*)',...
    'Cross Stream Dimension (mm)',...
    'ND Cross Stream Dimension (*)',...
    'Bag Growth Rate (m/s)',...
    'Rim Growth Rate (m/s)',...
    'Drop Area (mm^2)',...
    'ND Drop Area (*)',...
    'Axial Displacement(mm)',...
    'ND Axial Displacement (*)',...
    'Axial Velocity (m/s)',...
    'ND Axial Velocity (*)',...
    'Polynomial Axial Velocity (m/s)',...
    'ND Polynomial Axial Velocity (*)',...
    'Residuals for Axial Velocity (m/s)',...
    'Velocity Fraction (Vg-Vd/Vg)',...
    'Transverse over Axial Velocity (Vtrans/Vaxial)',...
    'Axial Acceleration (mm/ms^s)',...
    'ND Axial Acceleration (*)',...
    'Transverse Displacement (mm)',...
    'ND Transverse Displacement(*)',...
    'Transverse Velocity (m/s)',...
    'ND Transverse Velocity (*)',...
    'Cd of a Spheroid (*)',...
    'Cd Spheroid near critical (*)',...
    'CD(*)',...
    'Inst. Oh (*)',...
    'Inst Ga (*)',...
    'Inst ReL (*)',...
    'Inst New ND (*)',...
    'Inst We (*)',...
    'Inst Ga using Acc (*)',...
    'Inst ND Time (*)',...
    'Frame Number (frames)',...
    'Frames Number From 10% Deformed (Frame)',...
    'Axial Geometric Centroid Position (Pixels)',...
    'Transverse Geometric Centroid Position (Pixels)',...
    'Axial Basal Ring Centroid Position (Pixels)',...
    'Transverse Basal Ring Centroid Position (Pixles)',...
    'Ratio Cross Stream to Min Distance from Centroid (*)',...
    'Ratio Pappus Volume to Original (*)',...
    'Ratio Pappus Surface Area to Original (*)',...
    'Ratio of Major to Minor Axis (*)',...
    'Pixel Area (Pixels^2)',...
```

```
    'Stream Wise Box Dimension (Pixels)',...
    'Orientation of Ellipse (Angle)',...
    'Cross Stream Dimension (Pixels)',...
    'Top Point Row Number (Pixels)',...
    'Top Point Columns Number (Pixles)',...
    'Bottom Point Row Number (Pixles)',...
    'Bottom Point Columns Number (Pixels)',...
    'One Bag-Length (Pixels)',...
    'Row Number for One Bag-Length (Pixles)',...
    'Top Bag Length (Pixels)',...
    'Row Number For Top Bag Length (Pixles)',...
    'Bottom Bag Length (Pixels)',...
    'Row Number For Bottom Bag Length (Pixles)',...
    'Min Extent (*)',...
    'Min Axial Point (Pixels)',...
    'Min Vertical Point (Pixels)',...
    'Max Extent (*)',...
    'Max Axial Point (Pixels)',...
    'Max Vertical Point (Pixels)',...
    'Length of Protruding Stamen (Pixles)',...
    'Length of Protruding Stamen (mm)',...
    'ND Length of Protruding Stamen (*)'};
    xlswrite (s,Title_multiple_data_points,sheet_name,'EX1');
Multiple_data_points=horzcat(...
    data_x_axial,...
    Non_dimensional_data_x_axial,...
    Non_dimensional_viscous_time,...
    Axial_length,...
    Non_dimensional_axial_length,...
    Column_length,...
    Non_dimensional_column_length,...
    bag_growth_rate,...
    rim_growth_rate,...
    Drop_Area,...
    Non_dimensional_drop_area,...
    data_y_axial,...
    Non_dimensional_data_y_axial,...
    velocity_axial_from_data,...
    Non_dimensional_velocity_axial_from_data,...
    polynomial_velocity,...
    Non_dimensional_velocity_axial_from_poly,...
    axial_velocity_residuals,...
    velocity_fraction,...
    transverse_over_axial_velocity,...
    acceleration_axial_from_poly,...
    Non_dimensional_acceleration_axial_from_poly,...
    data_y_transverse,...
    Non_dimensional_data_y_transverse,...
    velocity_transverse_from_data,...
    Non_dimensional_velocity_transverse_from_data,...
    Cd_spheroid,...
    Cd_spheroid_near_critical,...
    Cd_from_data,...
    inst_Oh_num,...
    inst_Galilei_num,...
```

```
        inst_Liquid_Re_num,...
        inst_New_non_dimensional_num,...
        inst_We_number,...
        inst_Galilei_num_based_on_inst_acc,...
        inst_Non_dimensional_data_x_axial,...
        Frame_Number,...
        frame_num_from_time_zero,...
        Drop_info_centroid_axial,...
        Drop_info_centroid_transverse,...
        axial_centroid_based_on_basal_ring,...
        transverse_centroid_based_on_basal_ring,...
        ratio,...
        volume_ratio,...
        surface_area_ratio,...
        E,...
        Drop_info_Area,...
        bounding_box_3,...
        orientation,...
        peak_columns_and_location,...
        peak_rows_and_location,...
        peak_rows_and_location_top,...
        peak_rows_and_location_bottom,...
        min_extent,...
        min_point,...
        max_extent,...
        max_point,...
        stamen_length_protruding_from_image,...
        stamen_length_protruding,...
        Non_dimensional_stamen_length_protruding);
        xlswrite (s,Multiple_data_points,sheet_name,'EX2');
    Titles_program ={...
        'Program Time (s)',...
        'Program ND Time (*)',...
        'Program ND Displacement of Equator (*)',...
        'Program ND Displacement Rate of Equator (*/s)',...
        'Program Radius Cross Stream (m)',...
        'Program ND Cross Stream Length (*)',...
        'Program Cd(*)',...
        'Program Drop Velocity (*)'};
        xlswrite (s, Titles_program,sheet_name,'HP1');
    Data_program = horzcat(...
        t,...
        tstar,...
        y,...
        y_rate,...
        R,...
        R_over_R0,...
        Cd,...
        Ud);
        xlswrite (s,Data_program,sheet_name,'HP2');
% Data that needs specific treatment
        if num_bags==2;
            title_two_bags = {...
                'Length Top Bag (mm)',...
                'ND Length Top Bag (*)',...
```

```
                    'Growth Rate top Bag (m/s)',...
                    'ND Growth Rate Top Bag (*)',...
                    'Length Bottom Bag(mm)',...
                    'ND Length Bottom Bag(*)',...
                    'Growth Rate Bottom Bag (m/s)',...
                    'ND Growth Rate Bottom Bag (*)',...
                    'Time When Top Bag Ends (ms)',...
                    'ND Time When Top Bag Ends (*)',...
                    'Time When Bottom Bag Ends (ms)',...
                    'ND Time When Bottom Bag Ends (*)',...
                    'Axial Length When Top Bag Breaks (mm)',...
                    'ND Axial Length When Top Bag Breaks (*)',...
                    'Axial Length When Bottom Bag Breaks (mm)',...
                    'ND Axial Length When Bottom Bag Breaks (*)'};
                    xlswrite(s,title_two_bags,sheet_name,'HX1');
                info_top_bag = horzcat(...
                    Axial_length_top_bag,...
                    Non_dimensional_axial_length_top_bag,...
                    bag_growth_rate_top_bag,...
                    Non_dimensional_bag_growth_rate_top_bag);
                    xlswrite(s,info_top_bag,sheet_name,'HX2');
                info_bottom_bag = horzcat(...
                    Axial_length_bottom_bag,...
                    Non_dimensional_axial_length_bottom_bag,...
                    bag_growth_rate_bottom_bag,...
                    Non_dimensional_bag_growth_rate_bottom_bag);
                    xlswrite(s,info_bottom_bag,sheet_name,'IB2');
                info_2_bag_break_up_time = horzcat(...
                    Tbag_top,...
                    Non_dimensional_Tbag_top,...
                    Tbag_bottom,...
                    Non_dimensional_Tbag_bottom,...
                    min_axial_length_when_top_bag_breaks,...

Non_dimensional_min_axial_length_when_top_bag_breaks,...
                    min_axial_length_when_bottom_bag_breaks,...

Non_dimensional_min_axial_length_when_bottom_bag_breaks);
                    xlswrite(s,info_2_bag_break_up_time,sheet_name,'IF2');
        end
```