Effects of Liquid Superheat on Droplet Disruption in a Supersonic Stream

by

Logan M. Yanson

A Thesis

Submitted to the faculty

of the

WORCESTER POLYTECHNIC INSTITUTE

in partial fulfillment of the requirements for the

Degree of Master of Science

in

Mechanical Engineering

by

_____

March 2005

APPROVED:

_____

Dr. James Hermanson, Major Advisor

_____

Dr. David Olinger, Committee Member

_____

Dr. John Blandino, Committee Member

_____

Dr. Mark Richman, Graduate Committee Representative

Abstract

The effects of liquid superheat on the disruption of liquid droplets accelerated in a supersonic flow were examined experimentally in a drawdown supersonic wind tunnel. Monodisperse 60 μm diameter droplets of two test fluids (methanol and ethanol) were generated upstream of the entrance to the tunnel and accelerated with the supersonic flow such that their maximum velocities relative to the air flow were transonic. Droplets were imaged by shadowgraphy and by multiple-exposure direct photography using planar laser sheet illumination. In addition to providing information on droplet lifetime, the latter technique allows measurement of the droplet downstream distance versus time, from which the velocity and acceleration during disruption can be inferred. All droplets were unheated upon injection. Depending on the vapor pressure of the liquid, the droplets achieved varying levels of liquid superheat as they experienced low static pressure in the supersonic flow. Histograms of the droplet population downstream of the supersonic nozzle throat indicate that the lifetime of droplets in supersonic flow decreases with an increasing amount of droplet superheat. The shorter lifetime occurs even as the droplet Weber number (based on initial droplet size) decreases initially due to the lower relative velocity of the methanol droplets to that of ethanol droplets. This is due to a higher acceleration than ethanol droplets of comparable initial size. This is consistent with the more rapid disruption and the faster decrease in mass for the methanol droplets. The droplets, depending on the level of superheating, in some cases underwent disruption modes different than those expected for the corresponding values of Weber number.

Acknowledgements

The author would like to thank Dr. James C. Hermanson for advising and supporting this research endeavor and also for being patient over the past several years. The completion of this research was as much a product of his knowledge as it was his excellent advice throughout my tenure at WPI and the University of Washington. The author also needs to thank Dr. David Olinger and Dr. John Blandino, and Dr. Mark Richman for serving on my graduate committee. The author would also like to acknowledge the financial support of the National Science Foundation under the career grant number CTS-9733830.

Special thanks goes to all my friends for their support and the entertaining distractions they provided me that not only enabled me to complete this research, but to complete it with a small shred of the sanity with which I went into this. The author would like the thank the following people for their assistance and support: Elham Maghami, Matthew Munyon, Ben Nawrath, Ian DeBarros, Dr. Hamid Johari, Ms. Janice Dresser, Ms. Barbara Furhman, Ms. Barbara Edilberti, Ms. Pam St. Louis, Adam Young, Edwin Martin, Bruce Lavoie, Lieutenant Mark Phariss, Lisa Kajitani, Sanjoy Som, Rahul Mahajan, Ariane von Heyden, Pablo Navarro-Bullock, Joe Giordano, Olivia Dawson, Jon Lee, Byron Dickerson, Major Brad Johnson, Dr. Mitsuru Kurosaka, Ms. Wanda Frederick and Ms. Marlo Anderson. The author would also like to thank Ken Desabrais and Derek Bond for sharing their knowledge of MATLAB and, Mssrs. Dennis Pederson, Jim Johnston and Steve Derosier for sharing their knowledge of design and machine tools.

And lastly, a very special thanks to my family and especially my parents for all their love and support. I would not have been able to complete this work without them.

Table of Contents

List of Figures

List of Tables

Nomenclature

a	acceleration
D	droplet diameter
F	drag
K	drag coefficient
La	Laplace number
M	Mach number
Oh	Ohnesorge number
p	pressure
R	gas constant
t	time
T	temperature
v	velocity
V	volume
We	Weber number
x	position

Greek Symbols

γ	ratio of specific heats
μ	absolute viscosity
ρ	density
σ	surface tension

Subscripts

b	boiling point
d	deformation
D	droplet
F	fragmentation
∞	freestream
i	initial
r	relative

s	static
t	stagnation, total
v	vapor

1 Introduction

There are substantial benefits to be realized through the utilization of non-cryogenic hydrocarbons, rather than liquid hydrogen, to fuel supersonic combustion ramjets (scramjets). The advantages of hydrocarbon fuels include a higher volumetric energy density, lower relative cost, and ease of handling compared to liquid hydrogen. The successful use of any fuel in scramjets is critically dependent upon the ability to promote rapid, loss-effective fuel injection, mixing and combustion in supersonic flow[1]. Hydrocarbon fuel would normally be injected into a supersonic combustion chamber in a prevaporized state. This, however, may not be possible during a "cold start" phase of flight, where supersonic flow exists in the combustor but sufficient heat input is not yet available from the vehicle structure to prevaporize the fuel prior to injection, thus possibly requiring injection of incompletely vaporized fuel containing droplets.

Specific issues of interest include effects of liquid superheat and compressibility (as indicated by the relative droplet Mach number) on the droplet lifetime, the nature of their disruption, and any changes in droplet drag and acceleration.

1.1 Literature Review

The bulk of studies on the deformation and fragmentation of liquid droplets in gas over the past 50 years were conducted in shock tubes (a summary of some of these studies is shown in Tables 1-1 & 1-2), but occasionally in low speed wind tunnels or stagnant air. These studies have shown that the mode of deformation and fragmentation undergone by a droplet is primarily governed by two non-dimensional parameters, the Weber number and the Ohnesorge number.[2-6] The Weber number is the ratio of inertial

forces to surface tension forces and the Ohnesorge number characterizes the relationship between viscous and surface tension forces:

$$We = \rho_\infty v_r^2 D/\sigma_D \qquad 1.1$$

$$Oh = \mu_D/(\rho_D \sigma_D D)^{1/2} \qquad 1.2$$

Unlike the Weber number, the parameters in the Ohnesorge number depend only on the droplet properties, and not on the flow conditions.

In some literature the Laplace number, abbreviated by *La*, is used instead of the Ohnesorge number. The two are directly related, as follows:

$$Oh = La^{1/2} \qquad 1.3$$

The Laplace number is typically of an order of magnitude greater than unity; the Ohnesorge number is typically of order less than unity.

A Weber-number-breakup-mechanisms correlation was proposed by Pitch & Erdman[2] (see Fig. 1.1). The first breakup regime will occur when the droplet Weber number is less than 12 and is characterized by a flattening of the droplet perpendicular to the flow direction, followed in some cases by vibration. The flattening can become so extreme that the droplet is pinched off into multiple parts. From a droplet Weber number of 12 to 50, droplets undergo a "bag" style breakup in which the droplet is first flattened like the vibrational mode of breakup. However, unlike the vibrational mode, the middle of the flattened droplet begins to bow out, gradually forming into the shape of a thin bag or parachute. Eventually the bag portion of the deforming droplet tears into a fine mist, which is followed by a similar breakup of the remaining ring portion of the droplet. From a droplet Weber number of 50 to 100, droplets deform in a manner similar to the previous mode, but with the appearance of a protruding structure at the center of the bag.

This structure, referred to in most literature as a "stamen" because of its flowerlike appearance, stretches in a direction generally upstream of the droplet ring with the thick bag. Once the bag begins to fragment, all connections between the stamen and the ring are severed and the stamen and ring begin to be torn into a mist. For Weber numbers of 100 to 350, droplets first flatten in a manner similar to the other breakup regimes. Instead of deforming further, however, the flattened disk continually sheds a fine mist from its perimeter, shrinking until it is entirely fragmented. The final mode of droplet deformation, which occurs at Weber number in excess of 350, is called catastrophic or explosive breakup. This flattened droplet starts to pinch off in many places at once quickly reducing the droplet into a fine mist. It should be stressed that these ranges of Weber numbers should be used as guidelines, as the actual breakup modes can vary considerably depending upon experimental method and facility used (specifically, shock-tubes versus the direct injection employed here).

According to Krzeczkowski,[3] the effect of viscosity on time to droplet fragmentation is negligible for Ohnesorge numbers in excess of 0.0316. By contrast, Hirahara & Kawahashi[4] found that when the Ohnesorge number exceeds this value, the increased influence of viscosity does elevate the Weber number ranges for the onset of each breakup regime.

Given the small number of studies involving the deformation and breakup of droplets injected directly into supersonic flow, the known critical times for droplet deformation and fragmentation to date stem mostly from studies of droplets impacted by shock waves. The time to the onset of an appreciable degree of droplet deformation due to the impact of a shock wave is a function of relative droplet velocity, droplet diameter

and a ratio of the droplet density to the freestream static density.[5] The time to the catastrophic breakup of a droplet that struck by a shock wave is a function of Weber number only.[6]

This discussion thus far has involved understanding the deformation and fragmentation of unheated droplets due to aerodynamic forces only. However a number of studies[7-15] have examined the explosive disruption of superheat droplets. Vapor explosion, the rapid vaporization of a fluid at the superheat limit has been observed in neat, or pure, fluid droplets. This has been studied by creating a droplet at the bottom of a heated, stably stratified liquid column, and relying upon buoyant force to carry the droplet towards the surface. As the droplet rises, it experiences an increase in temperature. This temperature increase elevates the level of superheat of the droplet, which in turn causes a very rapid boiling of the test droplet. In an emulsion consisting of immiscible fluids such as water in oil, rapid boiling is triggered when the interior droplet temperature reaches the superheat limit of the more volatile component. The increased pressure gradient between the two phases and rapid bubble growth yields a fragmentation of the droplet.[7,12-14] For a 100 μm droplet, the typical time scale of disruption is of the order of 100 μs for the case of emulsions, whereas the characteristic time for thermal diffusion for the same size droplet is an order of magnitude larger. This difference is even greater for neat fluids where the disruption time scales can be of the order of 20 μs.

The objective of this research was to examine in compressible flow the disruption of droplets consisting of neat fluids with different vapor pressures (hence different levels of superheat). The variation in vapor pressures allows the examination of how changes in

the level of liquid superheat effect a change in the time to, and severity of, droplet deformation and fragmentation, independent of Weber number.

1.2 Objectives

The objectives of the research were to investigate the deformation and fragmentation of smoothly and rapidly accelerated droplets in a supersonic freestream and how changes in the level of liquid superheat affect the disruption modes experienced by the droplets, the time to disruption, and the droplet drag and acceleration. A review of the available literature shows that knowledge of the deformation and fragmentation of discrete droplets is essentially limited to abruptly accelerated (i.e. in a shock tube) droplets, as opposed to direct injection into supersonic flow. In addition, little is known about how changes in level of superheat affect the mode of deformation and fragmentation undergone by droplets. The aims of the current study can be summarized as follows:

- Examine droplet disruption for fluids with different vapor pressures, hence different levels of superheat, but with relatively constant surface tensions and densities,
- Measure the velocities and accelerations of droplets along the tunnel centerline,
- Obtain droplet Weber numbers and drag forces associated with these measurements,
- Examine whether changes in the level of liquid superheat affect a change in the time to, and severity of, droplet deformation and fragmentation.

To accomplish these objectives, a small-scale supersonic wind tunnel was constructed for testing in the WPI and University of Washington supersonic wind tunnel facilities. A

droplet image velocimetry technique was employed to measure droplet velocities and accelerations along the centerline of the wind tunnel. These measurements were used to determine the Weber numbers and drag forces experienced by the droplets. High-magnification images of droplets were taken to determine whether changing the level of droplet superheat affects a change in the mode of droplet deformation and fragmentation.

Author (Year)	Fluids	Weber Number	Diameter [μm]
Simpkins (1972)	Water	10 - 5.8×10^4	1200 – 2800
Wierzba (1988)	Water	600 – 7600	1000 – 4300
Waldman (1972)	Water	$10^5 – 10^6$	500 – 2500
Yoshida (1990)	Ethanol	1500	5600
Ranger (1969)	Water	$10^3 – 10^5$	750 – 4000
Joseph (1999)	Water Glycerol Various silicon oils	$10^4 – 1.7 \times 10^5$	2500 – 2900
Hsiang (1995)	Water n-Heptane Mercury Glycerol	0.5 – 1000	500 – 1500
Hanson (1963)	Water Methanol Various Silicon Oils	7 – 20	400 - 1000

Table 1-1 Summary of Investigation of Droplet Breakup in Shock Tubes.

Fluid	ρ [kg/m^3]	μ [mPa/s]	σ [mN/m]	T_{boil} @ 100kPa [°C]
Water	1000	0.890	71.99	99.6
Ethanol	789.2	1.074	21.97	78.0
Methanol	791.3	0.544	22.07	64.2
Glycerol	1261.3	934	62.5	287
Mercury		1.526	485.48	355.9
n-Heptane	683.7	0.387	19.65	98

Table 1-2 Fluid Properties for Investigations of Droplet Breakup in Shock Tubes. Density, viscosity and surface tension at standard temperature and pressure.

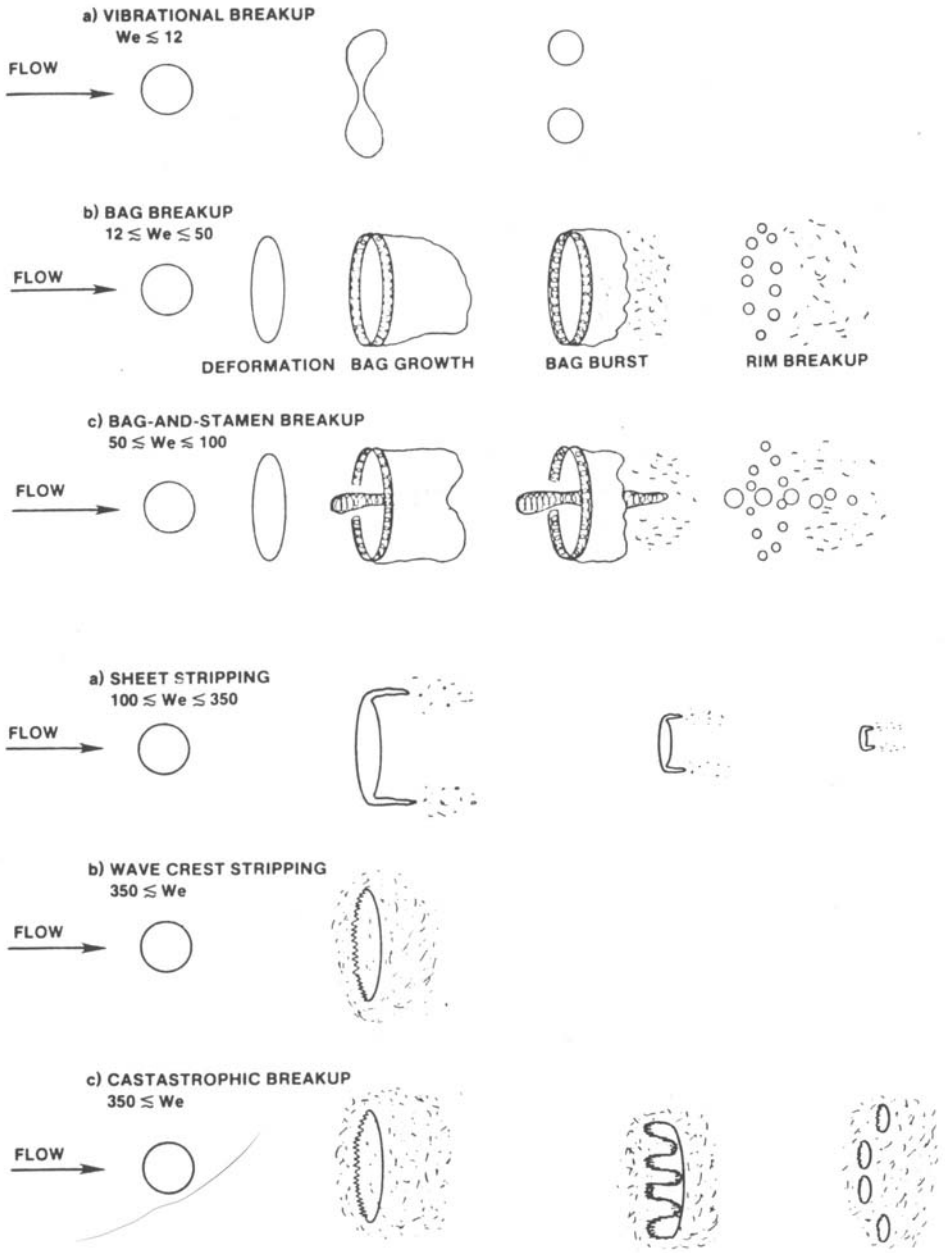

Figure 1.1 Disruption modes and Weber numbers.[2]

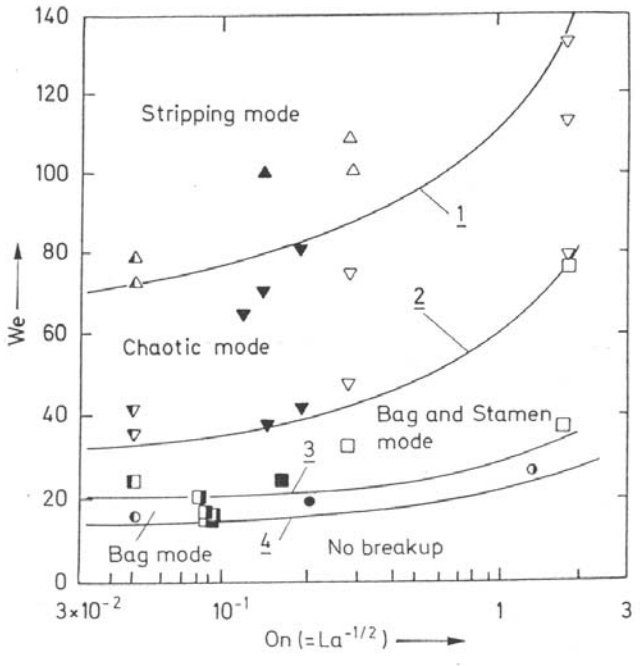

Figure 1.2 Effects of viscosity on the onset of disruption modes.[4]

2 Experimental Setup

2.1 Wind Tunnel

The experiments were conducted in the WPI and University of Washington draw-down supersonic wind tunnel facilities. Photographs of the high-magnification imaging configuration (discussed in section 2.3.1) and of the multiple-exposure imaging configuration (discussed in section 2.3.2) are shown in Figs. 2.1 and 2.2. Two quartz windows, that run the entire 2 inch (50 mm) length of the aluminum tunnel, diverge from the centerline at an angle of 5.6°. The nominal Mach number at the end of the tunnel was $M = 2$. The small scale of the supersonic tunnel was chosen to provide for short droplet residence times and to maximize the velocity difference between the droplets and the supersonic flow. The Mach number and static pressure (calculated from the pitot pressure measured on the tunnel centerline by a Pitot probe inserted through the bottom of the tunnel) as a function of position downstream of the tunnel throat are shown in Figs. 2.3 and 2.4. Two Lexan viewing ports are flush mounted into the aluminum sides of the tunnel to facilitate an uninterrupted view of the tunnel centerline from the throat to the end of the tunnel.

2.2 Droplet Generation Apparatus

A MicroFab Droplet-on-Demand (DoD) dispensing device, shown in Fig. 2.5, was used to create monodisperse, repeatable diameter droplets with a variety of fluids. When used in conjunction with a MicroJet III controller and a syringe pump, the DoD was capable of generating 60 μm droplets at frequencies from 10 to 3000 Hz. Any fluid can be used as long as its absolute viscosity lies between 0.5 and 20 cP, and its surface tension is between 0.02 and 0.7 N/m at the tip of the generator.

Prior to operation, it was imperative that the fluid meniscus was flush with the tip of the DoD dispensing device. With a non-flush meniscus the DoD would not generate droplets. The location of the fluid meniscus was not only important before generating droplets, but also during operation. To ensure a constant position of the meniscus, the syringe pump was programmed to supply the DoD with a constant flow rate of the test fluid. This flow rate was dependent upon the diameter of the syringe and frequency of droplet generation.

A Graphical User Interface (GUI) supplied by MicroFab was used to vary the waveform from the controller used to drive the DoD. While varying the input waveform changed the velocity and frequency of generated droplets, any significant change in droplet diameter requires a change in the diameter of the tip of the DoD.

For all experimental runs, the DoD was placed such that its centerline was aligned along the tunnel centerline and that its tip was 2 inches (50 mm) above the tunnel entrance. This distance between the DoD tip and the entrance was necessary so that fluid was ejected from the tip of the generator, rather than drawn out due to its proximity to the tunnel entrance.

2.3 Imaging System

2.3.1 High-magnification Imaging

In order to examine the droplet disruption in detail it was necessary to obtain high-magnification images of the droplets. To accomplish this, a VZM 300 microscope lens was attached to the C-mount of a Panasonic model WV-CP474 CCD camera. This optical arrangement gave a maximum system magnification of 3x at the plane of the CD array. This magnification and the 480x640 resolution of the camera yielded a minimum

resolvable dimension of 0.04 mm. The small field of view made it necessary to move the camera, in the direction of droplet travel, as many as seven times (dependent upon the test fluid) in order to view droplets at all downstream locations in the supersonic tunnel.

A Xenon Corp. model N-787B Nanopulse system capable of generating 500 mJ, 10 ns pulses at up to 60 Hz was used to effectively "freeze" the droplets in space. For droplet velocities of up to 400 m/s the droplet displacement during the illumination interval amounted to 4 μm. This amounts to 6.7% of the nominal injected droplet diameter 60 μm. A Mu-Tech MV1000 PCI frame grabber card was installed in a PC to which the CCD camera was connected. Through the use of the frame grabber card, 480x640 256 color grayscale tif image files were acquired and stored for post-processing (discussed in section 4.1).

2.3.2 Multiple Exposure Imaging

The multiple exposure imaging setup is composed of three main components: a Stanford Computer Optics (SCO) 4 Quik 05A series ICCD camera, a 35 mm MVO Double Gauss macro imaging lens and a Spectra Physics Model 2020 CW Argon-Ion laser. Whereas only a small area of the wind tunnel was viewable at any given time using the high magnification configuration, the combination of laser and sheet-forming optics in this configuration illuminated the entire length of the tunnel (plus a small length prior to the tunnel entrance) with a 1.5 W, 2.25 inch (57.15 mm) tall and 0.19 inch (4.83 mm) wide sheet. The ability to reduce the gate time of the SCO down to a few nanoseconds was used to "freeze" droplets in this case. Another capability of the SCO that was used was used was its ability to be programmed to take multiple exposures on a single frame. This feature made it possible, not only to track a droplet from the entrance of the tunnel

to that point where it was no longer recognizable as a droplet, but also to determine the velocity and acceleration of the droplets as they traverse the tunnel (the process to determine these numbers is described in Chapter 4, Image Processing).

Figure 2.1 Photograph of high-magnification imaging configuration. Top left corner: Nanopulser. Bottom right corner: VZM lens and Panasonic CCD camera.

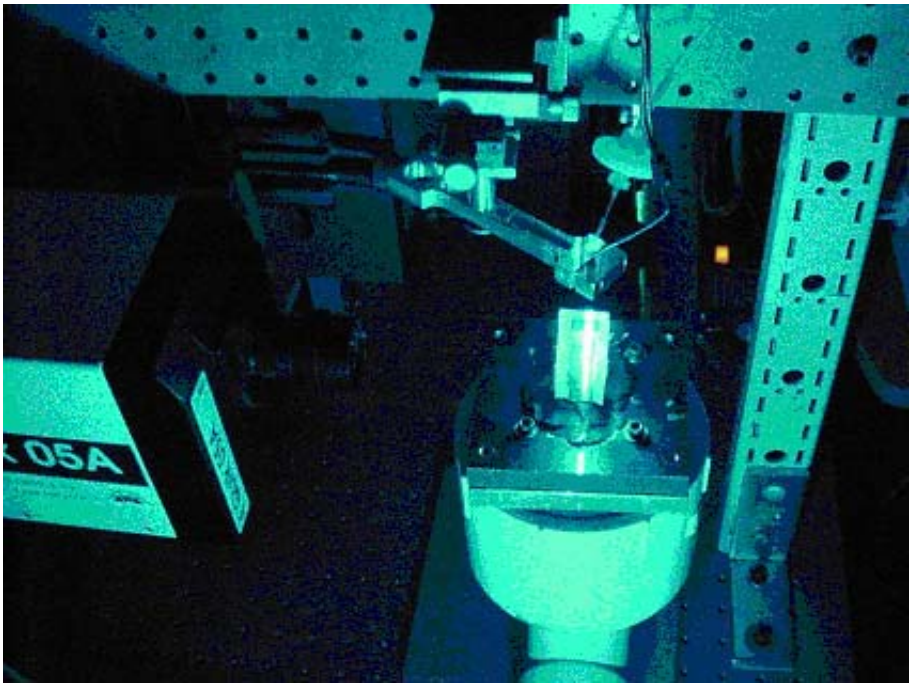

Figure 2.2 Photograph of multiple-exposure imaging configuration. Left hand side: SCO ICCD camera. Top: DoD generator. The laser sheet is the bright line bisecting the wind tunnel.

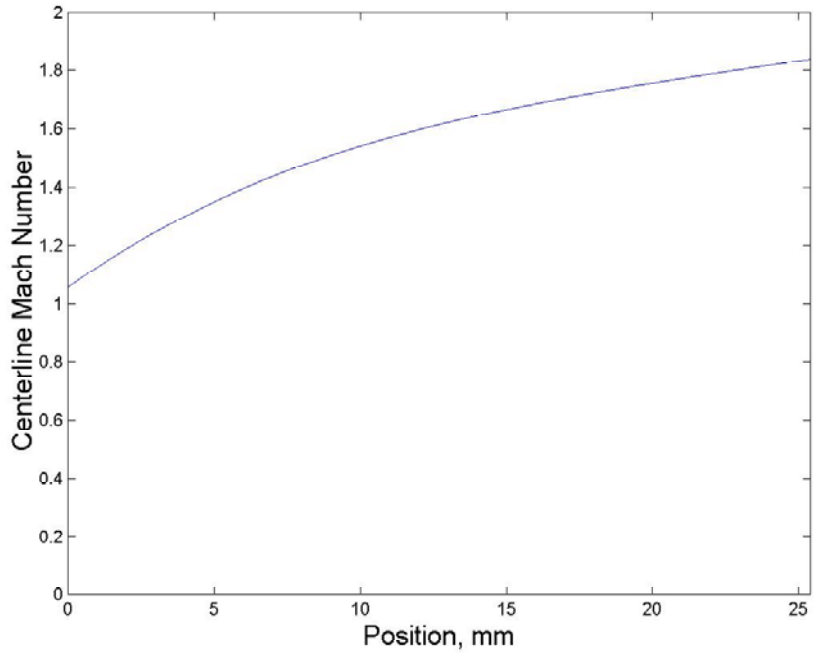

Figure 2.3 Centerline Mach number as a function of Position downstream of the tunnel throat.

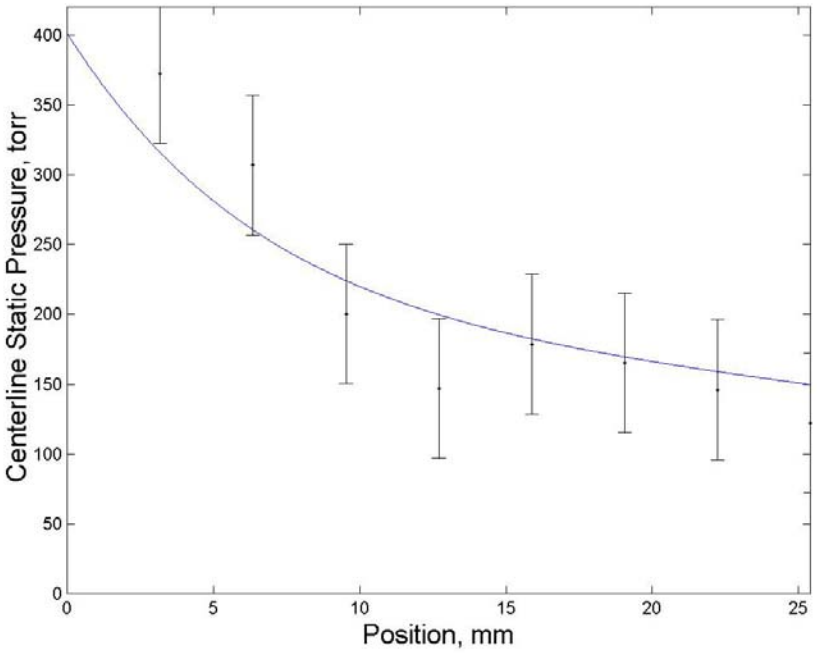

Figure 2.4 Centerline Static Pressure as a function of position downstream of the tunnel throat.

Figure 2.5 Photograph of DoD Dispensing Device. The fluid supply is attached to the metal nib on the right hand side of the image and droplets are ejected from the glass tip on the left hand side of the image. The dispensing device is 0.5 inch (12.7 mm) long.

3 Computational Model

Prior to the design the experiment, it was imperative to generate an estimate of the conditions experienced by droplets as they passed through the wind tunnel in an effort to have a better idea of what modes of disruptions droplets will undergo.

3.1 Model Structure

The Weber and Ohnesorge numbers are the most critical parameters involved in the determination of the disruption modes undergone by droplets. Due to the small diameter of droplet examined in this study, the Ohnesorge has little bearing on the droplet disruption mode.

The Weber number is comprised of the freestream air density, the droplet diameter and surface tension, and the velocity of the droplet relative to the freestream. Whereas measurement of the freestream velocity can lead to a reasonable estimate of the freestream air density and the surface tension of the unheated droplets is nearly constant along the centerline of the tunnel, estimation of the droplet velocity is more complex.

To estimate the droplet acceleration, an estimate of the drag coefficient under compressible conditions is required. Through studying the trajectories of small metal spheres 0.28 – 0.5 in (7.1 – 12.7 mm) in diameter fired from a smooth bore gun at Mach numbers ranging from 0.29 to 3.96, Charters & Thomas generated a series of data points for the drag coefficient, K_D, of small spheres as a function of Mach number. A series of curves, the equations of which are shown below, fit the data in the subsonic, transonic and supersonic regimes:

subsonic:	$0 < M < 0.5$	$K_D = 0.192$
transonic:	$0.5 < M < 1.4$	$K_D = 0.3173 + 0.2711 \cdot (M - 1)$

supersonic: $1.4 < M < 4.0$ $K_D = 0.3812 - 0.0140 \cdot (M - 2.75)$

For the case of droplets accelerating down the center of the wind tunnel, the Mach numbers in the equations listed above are Mach numbers relative to the freestream, M_r, rather than absolute droplet Mach numbers.

The usage of the Charters & Thomas drag coefficient established the assumptions used in the computational model that the droplets are spherical, have a constant diameter and remain undeformed.

The Charters & Thomas drag coefficient relates to the droplet drag, F_D, through the following equation:

$$F_D = K_D \cdot \rho_\infty \cdot D^2 \cdot v_r^2 = K_D \cdot \rho_\infty \cdot D^2 \cdot (v_\infty - v_D)^2$$

Where ρ_∞ is the freestream air density. Using Newton's law, this equation can be rewritten to solve for droplet acceleration as follows:

$$a_D = K_D \cdot (\rho_\infty \cdot D^2 / \rho_f V_D) \cdot (v_\infty - v_D)^2 = C \cdot (v_\infty - v_D)^2$$

With the assumption that from point of calculation to point of calculation the acceleration of the droplet is constant and that the velocity of the droplet at the initial point of calculation is known, the position of the droplet after a time step, Δt, is defined as:

$$x = x_i + v_{\infty i} \cdot \Delta t + (1/C) \cdot ln[1/(C \cdot \Delta t/(v_\infty - v_D) + 1/(v_\infty - v_D)^2)]$$

Similarly the droplet velocity after a time step is defined as:

$$v_D = v_{\infty i} - 1/[C \cdot \Delta t + 1/(v_\infty - v_{Di})]$$

Where v_{Di} is the velocity of the droplet at the beginning of the time step, and v_D is the velocity of the droplet at the end of the time step.

3.2 Model Results

Upon its completion, the computational model was run for 60 μm diameter droplets. In all cases the droplets are considered here to be rigid spheres, not subject to aerodynamic or thermodynamic disruption. Due to the nearly identical results for all test fluids (the only property variation is the slight change in fluid density), only the results for ethanol droplets are presented here.

Calculated droplet Mach numbers relative to laboratory coordinates and relative to the compressible stream are shown in Figs. 3.1 and 3.2, respectively. As can be seen in Fig. 3.2, the model predicts that the droplets remain at a transonic Mach number throughout the length of the supersonic tunnel. The slightly increasing computed relative Mach number is primarily due to the decrease in the local sound speed with static temperature; the relative velocity actually decreases somewhat. The corresponding variation in drag coefficient based on the Charter & Thomas data is shown in Fig. 3.3. A maximum Weber number at the tunnel throat of We = 153, as shown in Fig. 3.4, suggests that droplets of all test fluids observed in the tunnel should undergo a stripping type of fragmentation. For droplets of this size, the Ohnesorge number remains at a constant value of 0.338. This is not totally unexpected because the formulation of the Ohnesorge number includes droplet properties that, for the model, are assumed to remain constant along with position. At 0.0338, the Ohnesorge number is nearly identical to the 0.0316 value mentioned in the literature as a critical value for an increasing effect of viscosity on droplet breakup. As a result, it is assumed that viscosity has little or no effect on the breakup mode of the droplets. Fig 3.6 shows that droplet Reynolds numbers (relative to the compressible stream) peak at approximately 680. This value falls well outside the range (93,000 < Re < 1,300,000) of the experiments of Charters & Thomas,[22] suggesting

that the actual drag coefficient experienced by the droplets in this investigation may be different. Drag coefficients for spheres in incompressible flow[23] indicate little variation in the drag coefficient over the range of Reynolds number predicted by the computational model; the incompressible drag coefficients are roughly twice those of the compressible flow values of Charters & Thomas.

The Weber, Ohnesorge and Reynolds number results suggest that all the test fluid droplets will undergo similar modes of fragmentation with similar times to the onset of deformation. For the case of droplets with significant liquid superheating, however, the thermodynamic instability and flash vaporization may lead to deformation times considerably shorter than those due to aerodynamic disruption alone.

The static temperature on the tunnel centerline is considerably lower than ambient, as shown in Fig. 3.7. The high droplet velocity, combined with the thermal diffusivity of the liquid and the droplet diameter, suggests a representative value of the Fourier number of approximately $Fo = 4\alpha t/D^2 \approx 0.01$, suggesting negligible cooling of the droplet bulk fluid in the absence of fragmentation and vaporization.

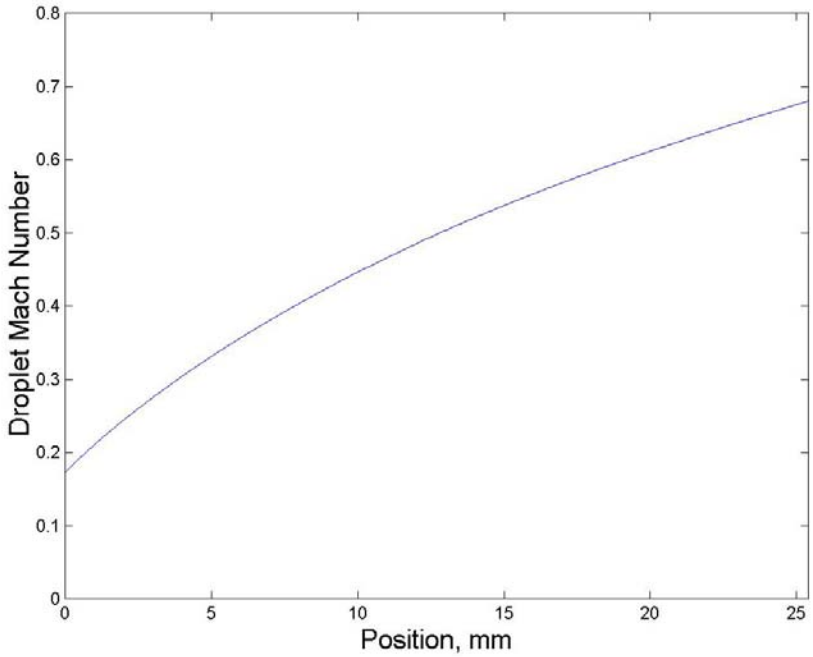

Figure 3.1 Droplet Mach number in laboratory-fixed coordinates as a function of position downstream of the tunnel throat.

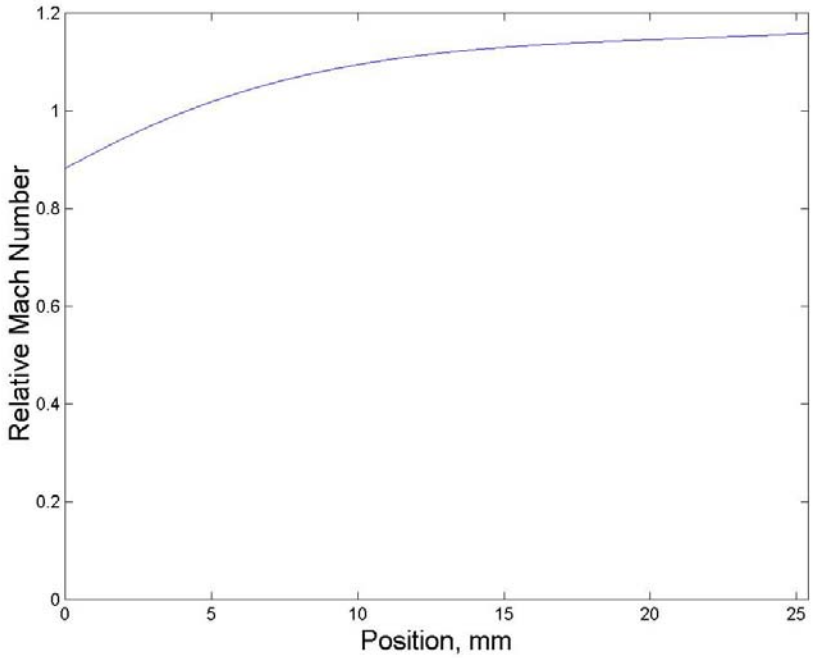

Figure 3.2 Relative droplet Mach number as a function of position downstream of the tunnel throat.

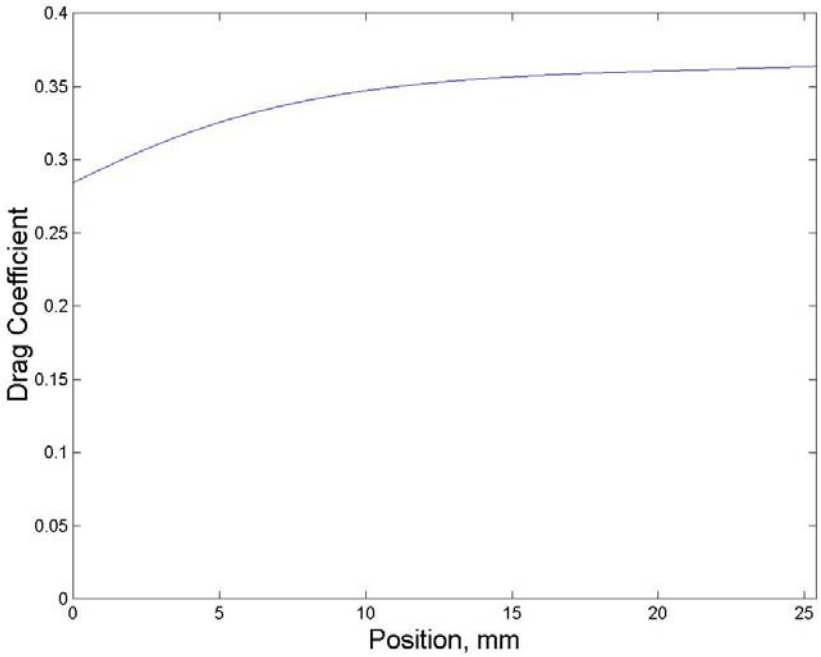

Figure 3.3 Drag Coefficient as a function of position downstream of the tunnel throat.

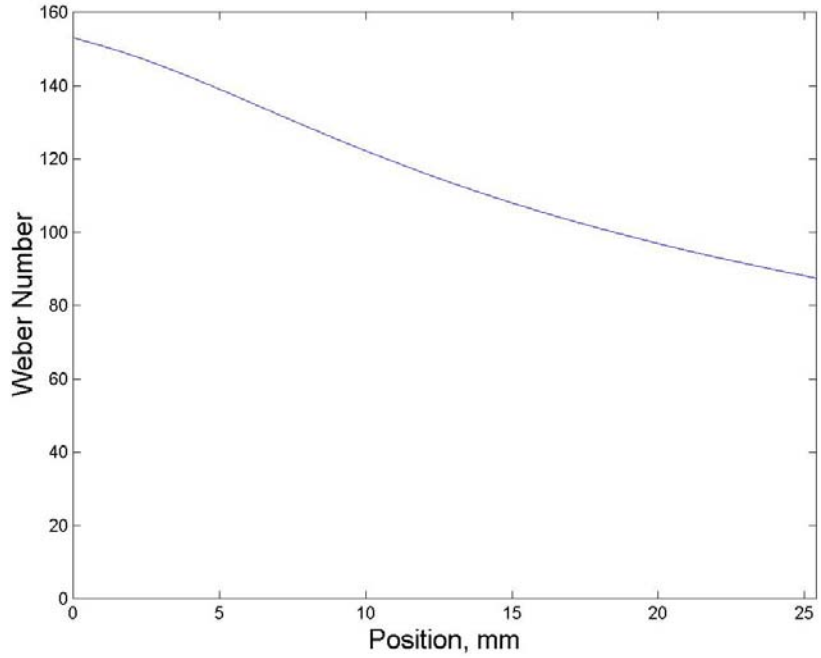

Figure 3.4 Droplet Weber number as a function of position downstream of the tunnel throat.

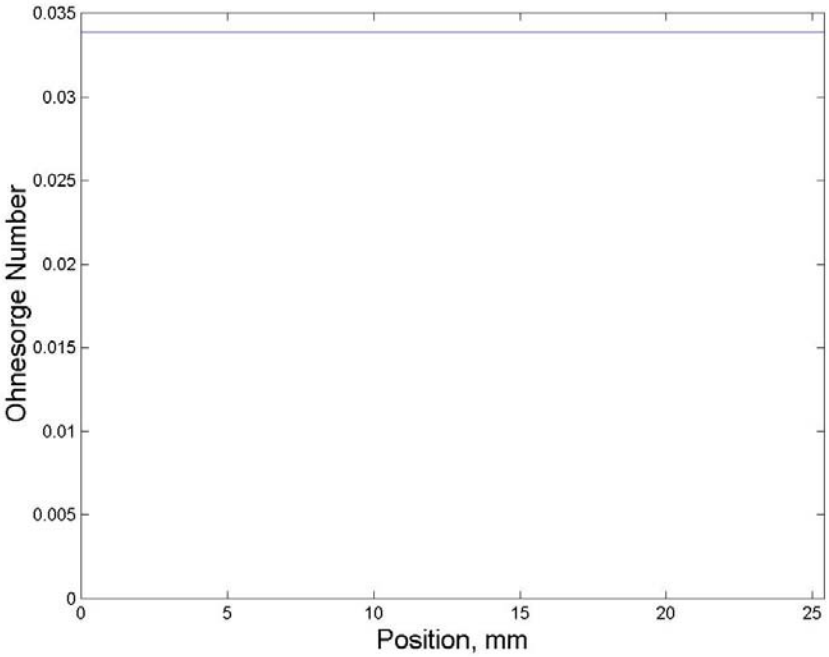

Figure 3.5 Ohnesorge Number as a function of position downstream of the tunnel throat.

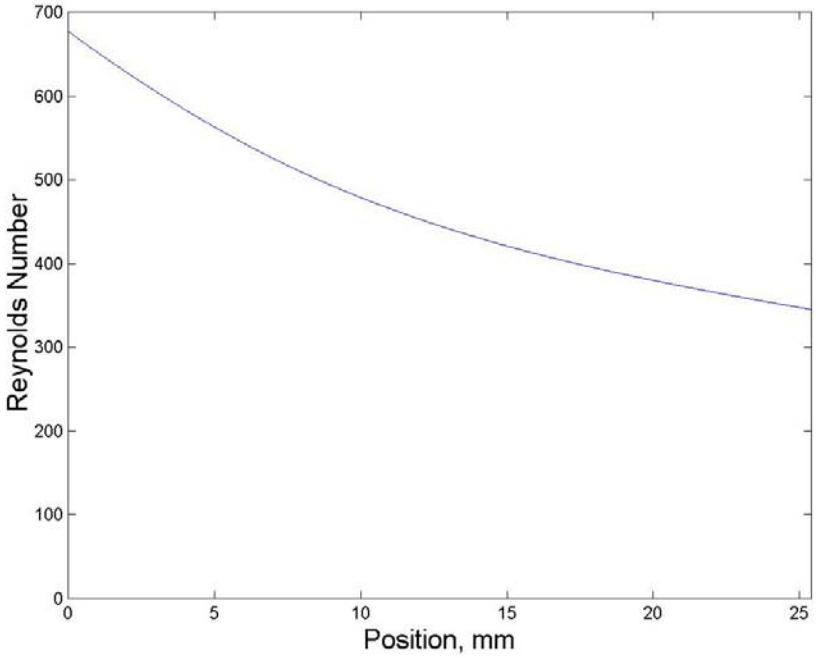

Figure 3.6 Reynolds Number as a function of position downstream of the tunnel throat.

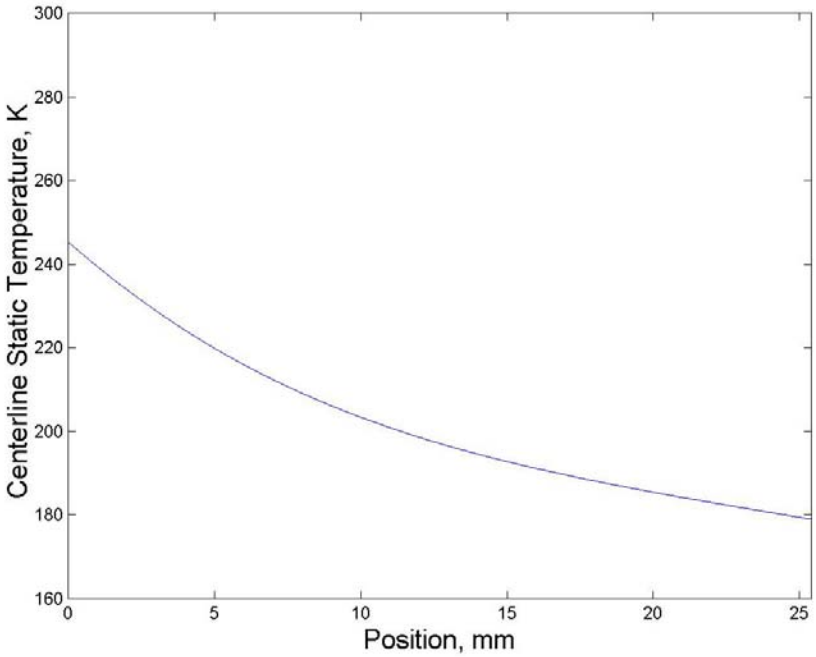

Figure 3.7 Static Centerline Temperature as a function of position downstream of the tunnel throat.

4 Image Processing

4.1 High-magnification Image Processing

To enable easier identification of fragmenting droplets and to ascertain the modes of droplet disruption, a MATLAB program along with four functions was written to process droplet images taken with the Panasonic CCD camera.

The images were first post-processed by performing a simple background subtraction to remove false "objects" due to imperfections in the optical system. Thirty droplet-free images were taken without the tunnel in position for this purpose. The mean intensity of each of these images was calculated and compared to the mean intensity of the image to be processed. The droplet-free image with the closest mean intensity was subtracted from the images to be processed. The original sample image is shown in Fig. 4.1.

Extraneous "objects" due to blemishes on the wind tunnel windows were removed using a slightly different technique, as dictated by the fact that these "objects" moved from image to image due to pulse-to-pulse variations in the location of the spark within the nanopulser. In this case background images were taken with the tunnel in place. The borders of the blank image were then cropped, and the sum of the absolute value of the pixels of the difference between the droplet image and the new cropped image was taken for each position. The cropped image was then "uncropped" (black pixels were used to replace the previously cropped pixels because of their zero contribution to any subsequent matrix addition or subtraction) using the location of the minimum of the difference calculation. The difference between the droplet images and the "uncropped"

image resulted in a reduction in the size of the "objects" due to blemishes on the windows.

Lastly, a scale was added to each image based on the distance in pixels between threads of a 0-80 machine screw placed in the tunnel in the absence of flow. A final processed image of a droplet is shown in Fig. 4.2.

4.2 Multiple Exposure Imaging

To quickly extract data from multiple exposure images of deforming droplets, a pair of MATLAB codes was written that created a droplet population histogram and a droplet velocity as a function of position downstream of the tunnel entrance. These codes are shown in Appendix B.

An example of the multiple exposure images captured with the SCO camera is shown in Fig. 4.3. In this image the droplet appears as seven bright spots oriented in a vertical line down the center of the wind tunnel.

Prior to extracting data from the multiple-exposure image, a background subtract was performed using a droplet-free image in order to remove any imperfections from the tunnel windows.

To decrease the processing time, the number of exposures was input and an imaginary box was drawn around the exposures. A summation was performed for each row within the imaginary box and the center of the fragmenting droplets was said to be the location of the highest value. The center positions for each exposure and image was used to generate a droplet population histogram. With a known time between exposures and an assumption that between exposures that droplets accelerate at a constant rate, a plot of droplet velocity as a function of position could be created.

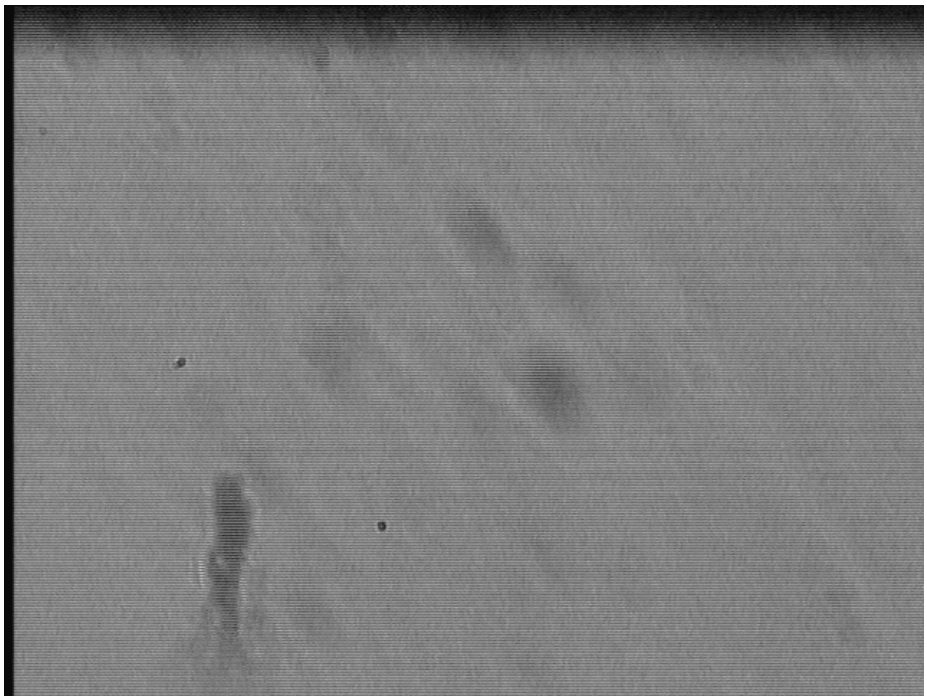

Figure 4.1 Unprocessed image of a 1-propanol droplet.

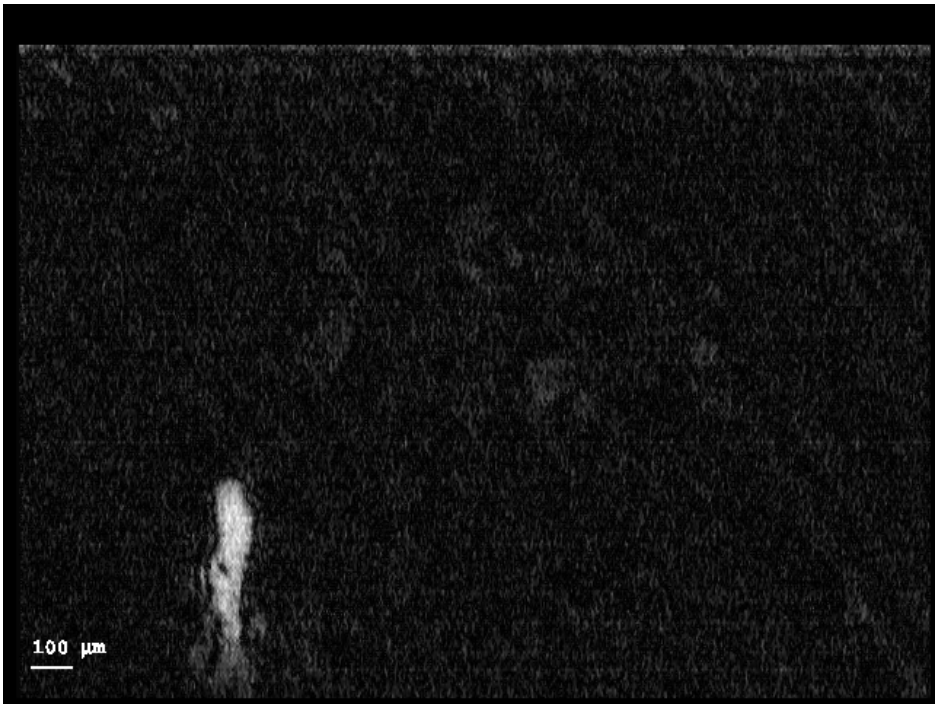

Figure 4.2 Processed image of a 1-propanol droplet.

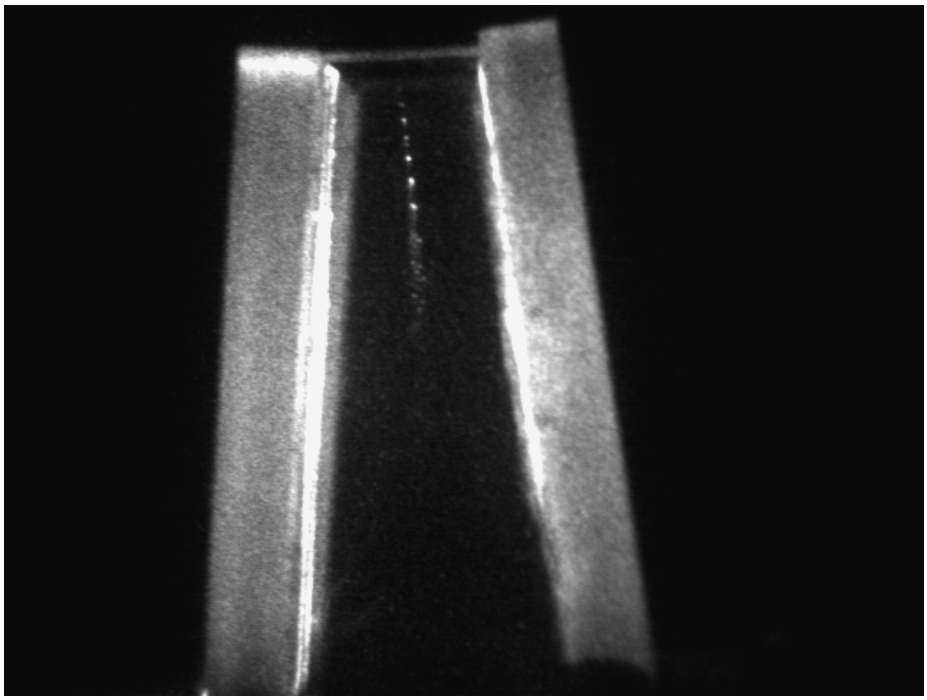

Figure 4.3 Multiple exposure image of ethanol droplet.

5 Results

High-magnification images of deforming and fragmenting droplets were captured for 1-propanol and ethanol. Although the Weber numbers for all droplets, based on the initial droplet diameters, were similar at downstream locations, the droplet disruption patterns were, in some cases, different. This is illustrated by Figs. 5.1 and 5.2 which show representative disrupting droplets of 1-propanol and ethanol, respectively. While a stamen-type of breakup evidently occurs for the 1-propanol droplets, the ethanol droplets appear to exhibit more of a stripping-type of disruption. Increasingly severe modes of droplet disruption (e.g. stripping instead of stamen) appear to occur with an increase in liquid superheat (see Fig. 5.3) as the vapor pressure increases relative to the decreasing static pressure. Table 5-1 indicates that ethanol droplets are the more volatile of the two, with a higher vapor pressure than that of 1-propanol.

The increased severity of the mode of disruption and level of liquid superheat of the droplets also yielded a decrease in the distance traveled by droplets prior to the location at which they were completely disrupted and no longer discernable in the high-magnification images. The observed droplet population (normalized with the droplet population over the first interval) as a function of position downstream of the tunnel throat is shown in the histograms in Fig. 5.4 and Fig. 5.5 for ethanol and methanol droplets, respectively. Through examination of these histograms it can be seen that a substantial decrease in the population of methanol droplets occurs more rapidly than that for the case of ethanol droplets.

For the creation of the histograms, droplets were considered to be disrupting if the center of the droplet was considerably brighter than its surrounding, otherwise, the

droplet was considered a fragment and not counted. Uncertainties brought about by differentiating between disrupting droplets and droplet fragments resulted in the normalized droplet populations not reaching zero. The more violent disruption of methanol droplets increased the error in determining the condition of droplets. This gave rise to the greater minimum population of methanol droplets when compared to ethanol.

This decrease in droplet lifetime was also evident in the multiple exposure images described previously. Examples of the multiple exposure images captured with the SCO camera is shown in Fig. 5.6. In these images the disrupting droplets appear as a series of bright spots oriented in a vertical line down the center of the wind tunnel. The multiple exposure images of deforming and fragmenting droplets for ethanol and methanol were used to determine the droplet position versus time along the tunnel centerline. All measurements were taken using the throat as a reference. Variations in intensity between successive exposures due to the non-uniform intensity over the height of the laser sheet did no preclude accurate determination of droplet positions.

The velocity of methanol droplets is only slightly greater than that of the ethanol droplets. The resulting plot of droplet velocity as a function of position downstream of the tunnel throat is shown in Fig. 5.7. The droplet velocity data can be fit with the following functions respectively for the ethanol and methanol droplets:

$$v_{ethanol} = 54x^{0.539}$$

$$v_{methanol} = 73x^{0.436}$$

At the throat the velocity of the ethanol droplets is approximately 100 m/s, whereas the velocity of the methanol droplets is 25% higher at 125 m/s. The difference between the velocities of ethanol droplets and methanol droplets decrease until the

velocities match at a location 2 cm downstream of the tunnel entrance. However, the actual location of this velocity match is somewhat unclear due to the limited number of data points far downstream of the throat.

The curve fits for the ethanol and methanol droplet velocities and known centerline freestream velocities were used to calculate the actual relative droplet Mach number as a function of position downstream of the tunnel entrance, as shown in Fig. 5.8. Flow Mach number was based on the curve fit from the static centerline pressure in Fig. 2.4. The trends in relative Mach number observed experimentally are seen to differ significantly from those predicted by the simple, 1-D computational model based on rigid-sphere droplets discussed previously (Fig. 3.2). Not only do the experimental results show a subsonic relative droplet Mach number instead of the supersonic Mach number predicted by the computational model, but they also indicate a decrease in the relative droplet Mach number as a function of position. The decrease in relative droplet Mach number means the droplet velocity is approaching the flow velocity. The difference between the model and the experimental results can be attributed to the combined effects of aerodynamic disruption and disruption due to the droplet superheating as well as the uncertainty in the drag coefficient of the undeformed droplets mentioned previously.

The difference in relative droplet Mach numbers determined experimentally implies a modest difference in the Weber numbers, as shown in Fig. 5.9 (the Weber number is here again based on the diameter of the undeformed droplets). The measured velocities suggest that the maximum Weber number experienced by ethanol and methanol droplets were 88 and 75, respectively, based on the initial droplet diameter,

compared to the analytical rigid-sphere prediction of $We = 139$. This decrease in Weber number would, in the absence of rapid vaporization, be expected to shift the disruption regime from a stripping-type towards a stamen-type disruption. It should be pointed out that, in addition to changes in the relative velocity of the droplets, the actual Weber number also changes with time and position due to changes in the droplet diameter due to disruption.

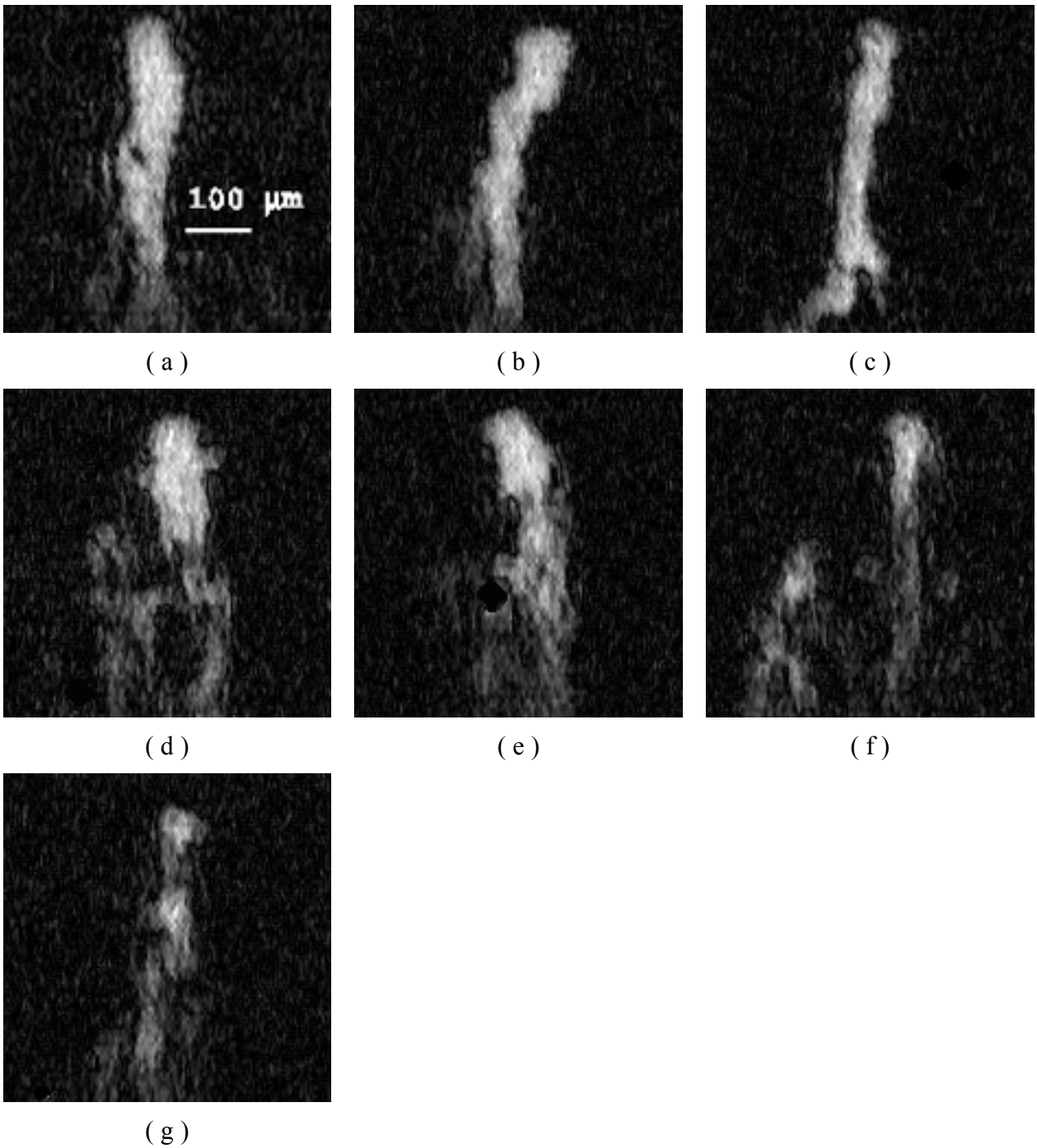

Figure 5.1 High magnification images of various deforming and fragmenting 1-propanol droplets. Positions of droplets in millimeters are as follows: (a) 1.28, (b) 2.20, (c) 3.06, (d) 3.50, (e) 3.92, (f) 4.43, and (g) 5.78.

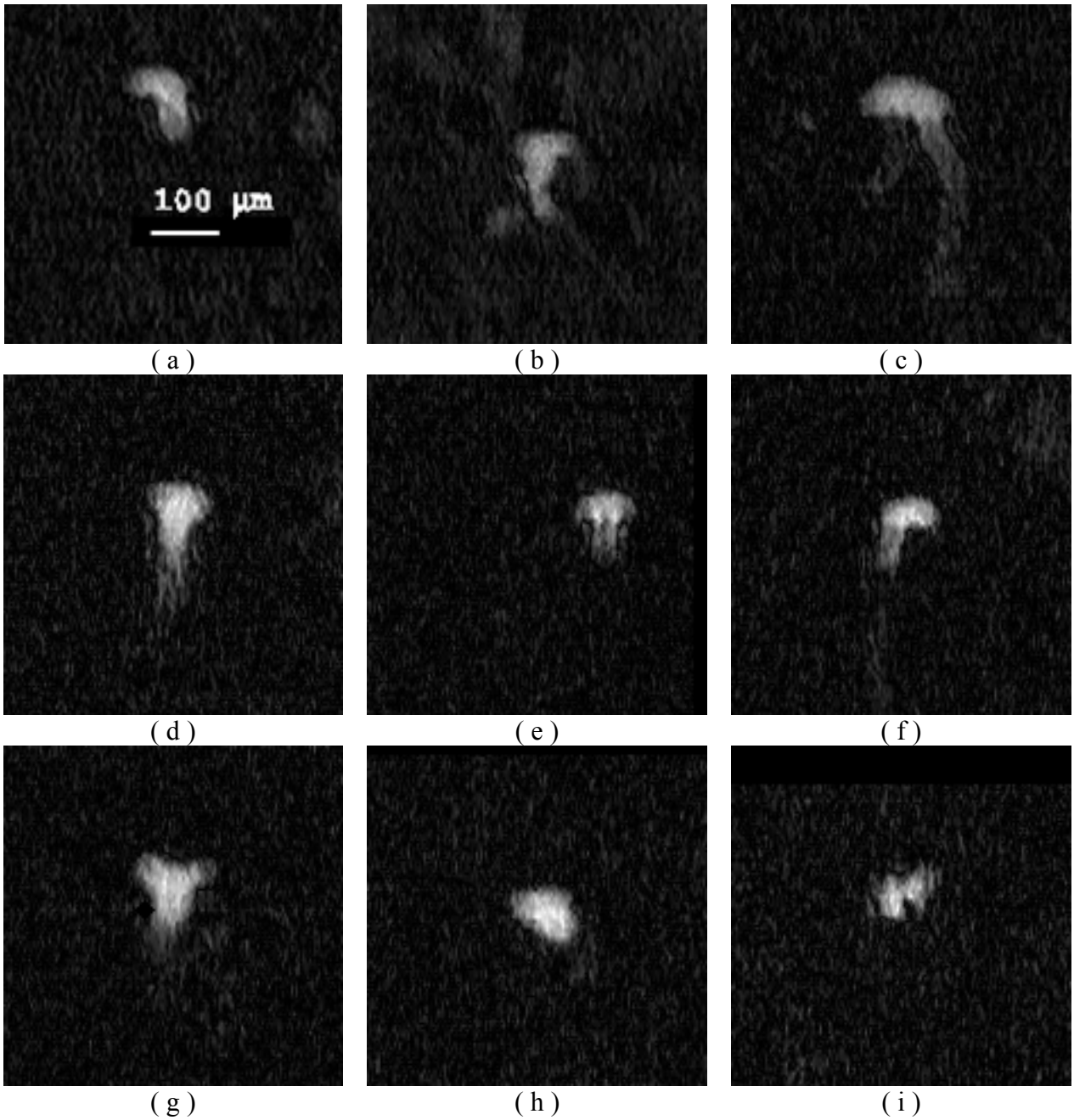

Figure 5.2 High magnification images various of deforming and fragmenting ethanol droplets. Positions of droplets in millimeters as follows: (a) 0.76, (b) 0.43, (c) 0.32, (d) 2.17, (e) 2.60, (f) 2.73, (g) 3.45, (h) 3.54, and (i) 3.54.

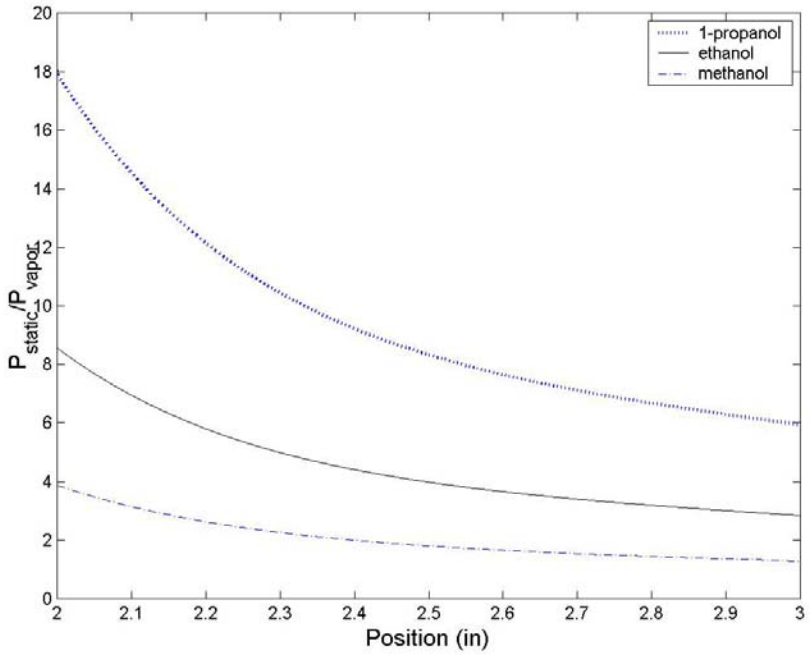

Figure 5.3 Static pressure-vapor pressure ratio as a function of position downstream of the tunnel entrance.

Fluid	Formula	p_v [torr]	ρ [kg/m^3]	μ [cP]	σ [μm]
1-propanol	C_3H_7OH	20	799.7	20	23.32
ethanol	C_2H_5OH	43.9	789.2	43.9	22.32
methanol	CH_3OH	97	791.3	97	22.55

Table 5-1 Properties of test fluids.[26]

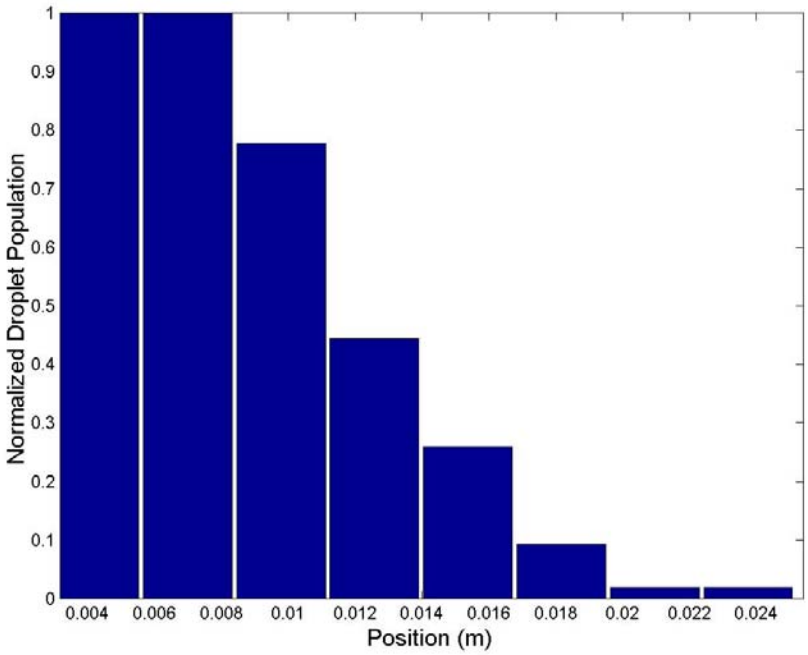

Figure 5.4 Normalized ethanol droplet population as a function of position downstream of the tunnel throat.

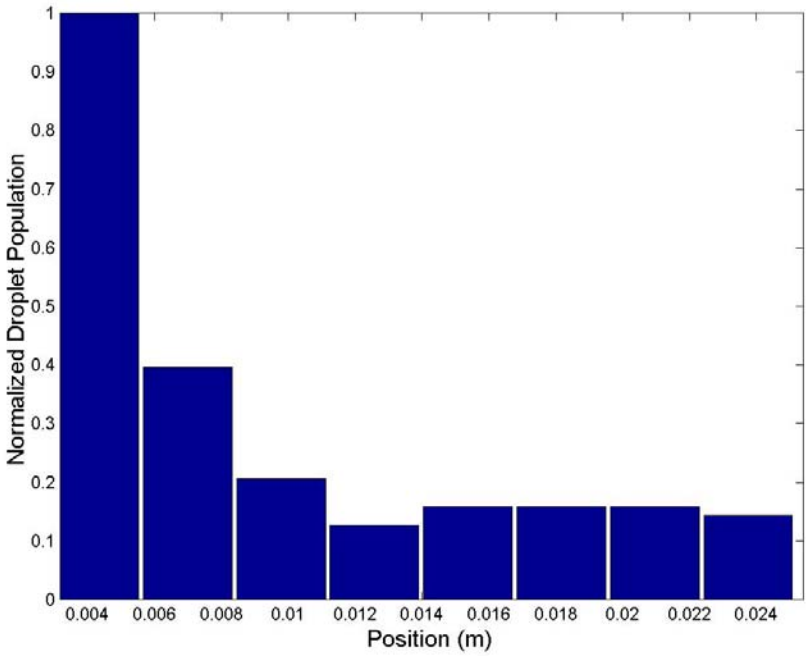

Figure 5.5 Normalized methanol droplet population as a function of position downstream of the tunnel throat.

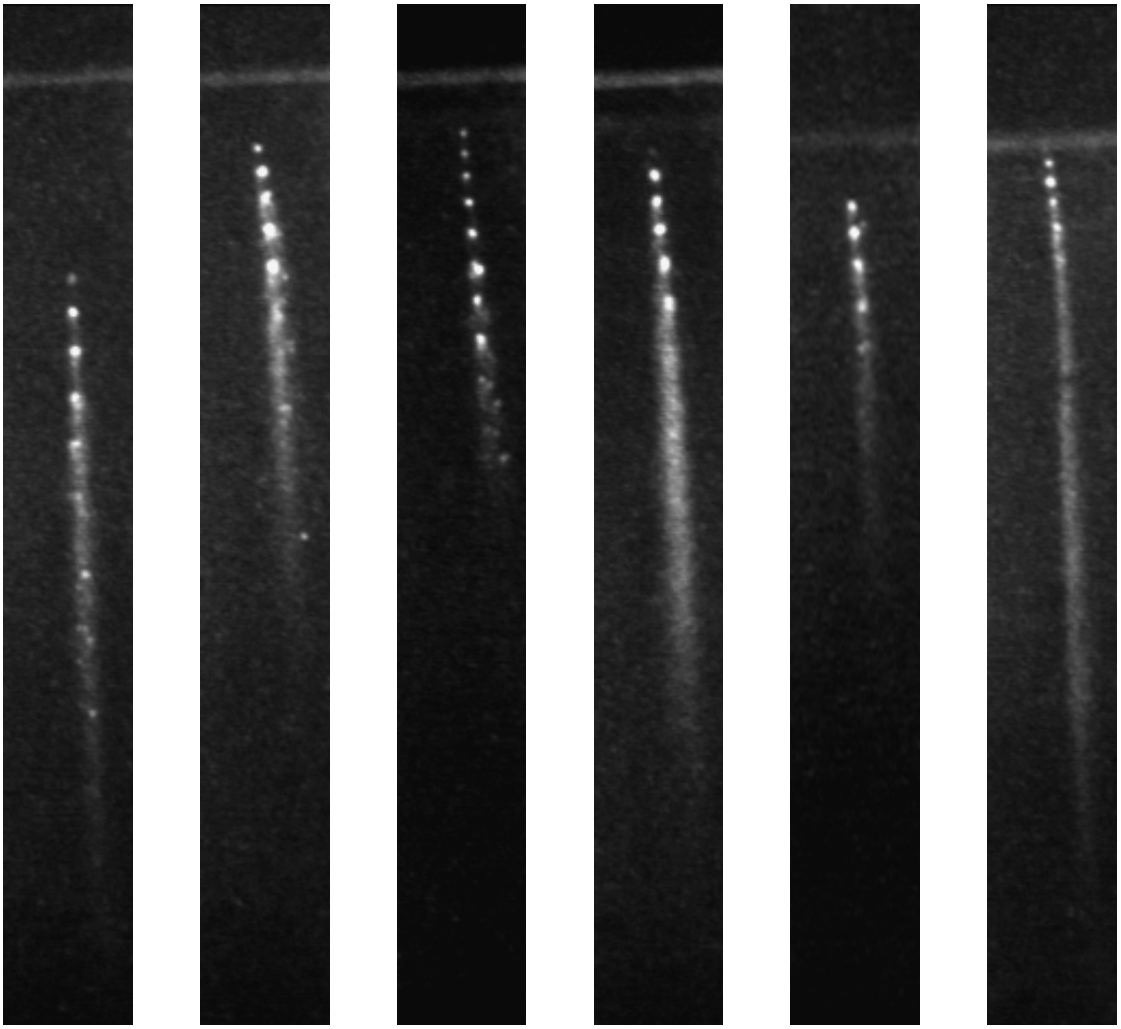

Figure 5.6 Sample multiple exposure images deforming and fragmenting droplets. Ethanol a-d and methanol e & f.

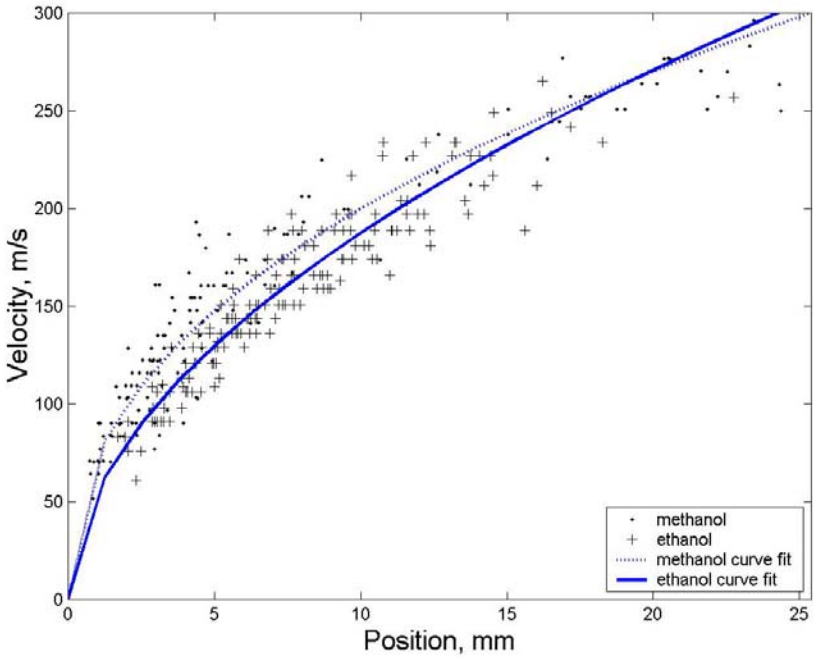

Figure 5.7 Ethanol and methanol droplet velocity as a function of position downstream of the tunnel throat.

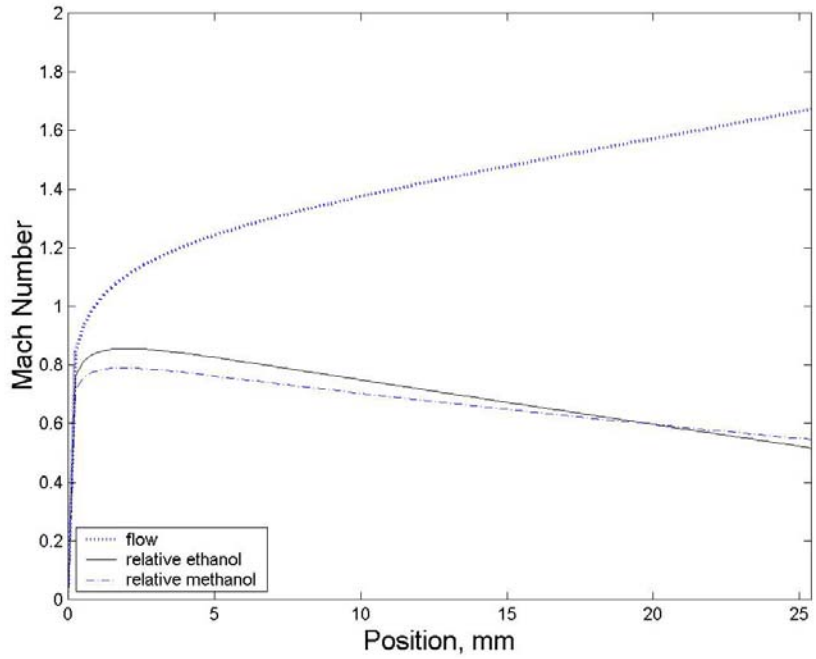

Figure 5.8 Relative ethanol and methanol droplet Mach number as a function of position downstream of the tunnel entrance.

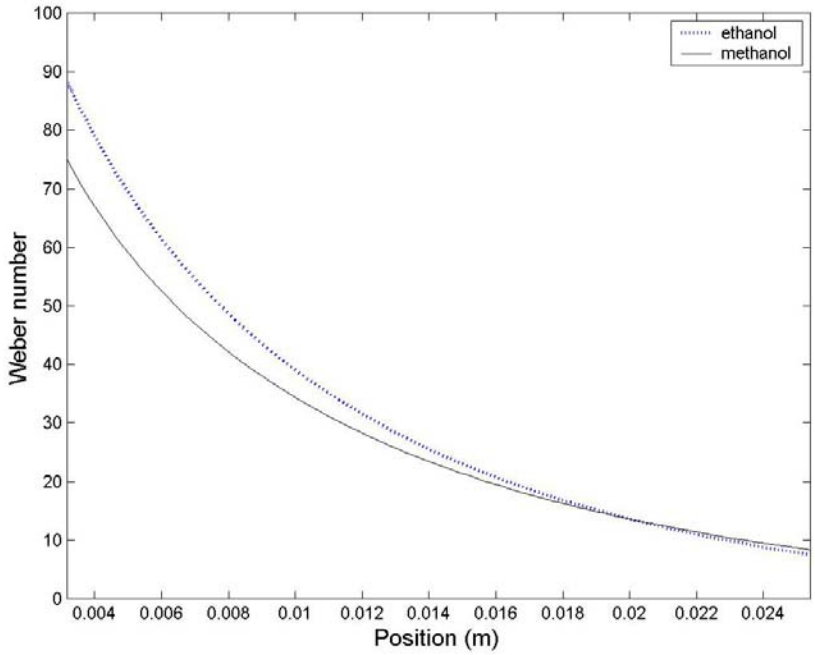

Figure 5.9 Ethanol and methanol droplet Weber number as a function of position downstream of the tunnel throat.

6 Summary and Conclusions

The effects of fluid superheat on the deformation and fragmentation behavior of droplets smoothly accelerated in a supersonic flow were examined by injecting unheated droplets into a supersonic wind tunnel. The decrease in static pressure experienced by the droplets elevated the level of liquid superheat of the droplets. Although the Weber numbers for all three test fluids (1-propanol, ethanol and methanol) were comparable, the droplet disruption modes were not identical, but appeared rather to change as the level of liquid superheat increased, in the direction of a more catastrophic mode of disruption. The relative population of detectable drops decreased more rapidly with downstream distance for the case of increasing superheat due to the more rapid droplet disruption. The measured velocity and relative Mach number of droplets increased slightly with level of superheat, consistent with a decrease in droplet mass due to accelerated vaporization. This increase in droplet velocity affected an increase in the droplet Weber number; however, this increase in Weber number does not by itself appear to account for the observed changes in the severity of droplet disruption.

7 Recommendations for Future Research

The deformation and fragmentation behavior of more test fluids, with a wider range vapor pressures, needs to be examined in order to more clearly establish the relationship between volatility and type of deformation and time to fragmentation.

Mixtures of two of more fluids need to be examined to determine the effects of adding a small quantity of a volatile fluid (acetone for example) to a less volatile fluid (1-propanol). A relation between time to and severity of deformation and fragmentation can be established for volumetric and vapor pressure ratios of fluids.

Planar laser-induced fluorescence (PLIF) using a pulsed Nd:YAG laser should be used to obtain detailed concentration, mixing and flow visualization information. However, because few fluids will fluoresce in the visible spectrum when illuminated with the ultraviolet light from the Nd:YAG laser, it is possible that different tunnel configurations will need to be manufactured to better match levels of liquid superheat achieved by the non-fluorescing test fluids.

A new tunnel configuration will not only lend itself to more representative data achieved through PLIF, but it will also allow for examination of deformation and fragmentation behavior over a wider range of relative velocities and therefore a wider range of Weber numbers. A supersonic wind tunnel even more compact than the configuration employed here could make use of an under-expanded jet to achieved a more aggressive acceleration of freestream, and therefore a greater velocity lag (perhaps even supersonic) between the freestream and droplets.

8 References

[1]Kay, I.W., Peshke, W.T. and Guile, R.N., "Hydrocarbon-Fueled Scramjet Combustor Investigation," *Journal of Propulsion and Power.* March-April 1992, Vol. 8, No. 2, pp. 507-512.

[2]Pilch, M. and Erdman, C.A. "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.* 1987, Vol. 13, No. 6, pp. 741-757.

[3]Krzeczkowski, Stefan A. "Measurement of Liquid Droplet Disintegration Mechanisms," *International Journal of Multiphase Flow.* 1980, Vol. 6, pp. 227-239.

[4]Hirahara, H. and Kawahashi, M. "Experimental Investigation of Viscous Effects Upon a Breakup of Droplets in High-Speed Air Flow," *Experiments in Fluids.* 1992, No. 13, pp. 423-428.

[5]Yoshida, T. and Takayama, K. "Interaction of Liquid Droplets with Planar Shock Waves," *Journal of Fluids Engineering.* December 1990, Vol. 112, pp. 481-486.

[6]Waldman, George D. and Reinecke, William G. "Raindrop Breakup in the Shock Layer of a High-Speed Vehicle," *AIAA Journal.* September 1972, Vol. 10, No. 9, pp. 1200-1204.

[7]Owen, I. and Jalil, J.M., "Heterogeneous flashing in water drops," *International Journal of Multiphase Flow*. 1991, Vol. 17, No. 5, pp. 653-660.

[8]Edwards, J.C. and Perlee, H.E. 1982 "Superheated Drop Vaporization: Halon 1301," *US Bureau of Mines Rpt. 8658.* 1982.

[9]McCann, H., Clarke, L.J. and Masters, A.P., "An experimental study of vapor growth at the superheat limit temperature," *International Journal of Heat and Mass Transfer,* 1989, Vol. 32, No. 6, pp. 1077-1093

[10]Shepard, J.E. and Sturtevant, B., "Rapid evaporation at the superheat limit," 1982, Vol. 121, pp. 379-4027.

[11]Xie, J.G., Ruekgauer, T.E., Armstrong, R.L. and Pinnick, R.G., "Evaporative Instability in Pulsed Laser-Heated Droplets," *Physical Review Letters*. 1991, Vol. 66, No. 23, pp. 2988-2991.

[12]Yap, L.T., Kennedy, I.M. and Dryer, F.L., "Disruptive and Micro-explosive Combustion of Free Droplets in Highly Convective Environments," *Combustion Science and Technology.* 1984, Vol. 41, pp. 291-313.

[13]Tsue, M., Yamasaki, H., and Kadota, T., "Microexplosion of an emulsified fuel droplet under microgravity," *Transactions of the Japan Society of Mechanical Engineers.* 1995, Part B Vol. 61, No. 587, pp. 2712-2717.

[14]Lasheras, J.C., Yap, L.T. and Dryer, F.L., "Effects of the Ambient Pressure on the Explosive Burning of Emulsified and Multicomponent Fuel Droplets," *20th Symposium (International) on Combustion.* 1984, pp. 1761-1772.

[15]Meier, G.E.A. and Thompson, P.A., "Real Gas Dynamics of Fluids with High Specific Heat, " in *Lecture Notes in Physics 235, Flow of Real Fluids.* Meier & Obermeier, Eds., Springer, 1985, pp. 103-114.

[16]Simpkins, P.G. and Bales, E.L. "Water-drop Response to Sudden Accelerations," *Journal of Fluid Mechanics.* 1972, Vol. 55, No. 4, pp. 629-639.

[17]Wierzba, A. and Takayama, K. "Experimental Investigation of the Aerodynamic Breakup of Liquid Drops," *AIAA Journal.* 1988, Vol. 26, No. 11, pp. 1329-1335.

[18]Ranger, A.A. and Nicholls, J.A. "Aerodynamic Shattering of Liquid Drops," *AIAA Journal.* February 1969, pp. 285-290.

[19]Joseph, D.D.; Bellanger, J. and Beavers, G.S. "Breakup of a Liquid Drop Suddenly Exposed to a High-Speed Airstream," *International Journal of Multiphase Flow.* 1999, Vol. 25, pp. 1263-1303.

[20]Hsiang, L.P. and Faeth, G.M. "Drop Deformation and Breakup due to Shock Wave Steady Disturbances," *International Journal of Multiphase Flow.* 1995, Vol. 21, No. 4, pp. 545-560.

[21]Hanson, A.R.; Domich, E.G. and Adams, H.S. "Shock Tube Investigation of the Breakup of Drops by Air Blasts," *The Physics of Fluids.* August 1963, Vol. 6, No. 8, pp. 1070-1080.

[22]Charters, A.C. and Thomas, R.N., "The Aerodynamic Performance of Small Spheres from Subsonic to High Supersonic Velocities," *Journal of the Aeronautical Sciences.* October 1945, pp. 468-476.

[23]Fox, R.W., and McDonald, A.T., *Introduction to Fluid Mechanics,* 5th ed., John Wiley & Sons, Inc., New York, 1998, pp. 250.

[24]Hodge, B.K. and Koenig, K., *Compressible Fluid Dynamics with Personal Computer Applications,* Prentice Hall, New Jersey, 1995.

[25]Thompson, P.A., *Compressible-fluid Dynamics,* McGraw-Hill Book Company, New York, 1972.

26 *CRC Handbook of Chemistry and Physics,* CRC Press, Cleveland, Ohio, 79th ed. 1998.

APPENDIX A: Image Processing Codes for Zoom Imaging

A-1 imageprocess.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                                                   %
% Droplet deformation and fragmentation research.                                   %
%                                                                                   %
% Filename: imageprocess.m                                                          %
%                                                                                   %
% Image processing code to:                                                         %
%           1) deinterlace images                                                   %
%           2) eliminate dirt on the lens from images                               %
%           3) eliminate dirt on the tunnel windows from images                     %
%           4) increase contrast (make black and white) to more easily identify droplets %
%                                                                                   %
% Inputs are:                                                                       %
%           1) scrw.tif - image of screws at various position in the tunnel         %
%                     used for determining location in the tunnel and image scaling %
%                     located in //image on HELMET                                  %
%           2) path.tif - black image with text ('100 micron') written in the bottom lefthand corner %
%                     used to place legend on the bottom of each processed image and %
%                     mark location of main image directory                         %
%                     located in //image on HELMET                                  %
%           3) path.txt - empty text file                                           %
%                     used to mark a location for saved processed images            %
%                     located in //processed on SHIELD                              %
%           4) lens.tif - blank images taken at a position above the wind tunnel    %
%                     used to subtract lens dirt from images                        %
%                     located in //image/Lens on HELMET                             %
%           5) wndw.tif - blank images taken at various positions in the wind tunnel %
%                     used to subtract window dirt from images                      %
%                     located in //image/P##/Window on HELMET                       %
%           6) ####.tif - droplet images taken at various position in the wind tunnel %
%                     images to be processed                                        %
%                     located in //image/P##/Run## on HELMET                        %
%                                                                                   %
% Calls the following function as needed:                                          %
%           1) deinterlace.m                                                        %
%           2) lensfunction.m                                                       %
%           3) matchintensity.m                                                     %
%           4) motionfunction.m                                                     %
%                                                                                   %
% Code written by Logan M. Yanson and Mark R. Phariss                               %
%                                                                                   %
% Last modified Monday, August 31st, 2004                                           %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear all;
                                                        % Clear memory

[fname3, pname3] = uigetfile('*.tif', 'Open tiff of screw');    % Open image of screw to determine scale.
file3 = [pname3 fname3];
```

```
screw = imread(file3, 'tiff');                          % Fill screw image matrix.
screw = double(screw);                                  % Double precision of screw image matrix.
screw = deinterlace(screw);                             % Deinterlace screw image.
colormap(gray(255));                                    % Change colormap to 0-255 gray scale.
image(screw);                                           % Display screw image.

% Display GUI that determines the scaling of the images.
prompt = {'Threads per inch of the screw','Peak to peak spacing (pixels)'};
title = 'Scaling Information';                          % Set title of GUI.
lines = 1;                                              % Number of lines in each input of the GUI.
def = {'80','85'};                                      % Default values of the GUI.
hold = inputdlg(prompt,title,lines,def);
hold = char(hold);                                      % Change from string to ASCII.
hold = str2num(hold);                                   % Change from ASCII to number.
threads = 25400/hold(1);                                % Spacing of threads in microns.
pixels = hold(2);                                       % Number of pixels.

scale = pixels*100/threads;                             % 100 microns on the image.

close;

cd (pname3);
[fname1, pname1] = uigetfile('*.tif', 'Open tiff file for file source path determination');
[fname2, pname2] = uigetfile('*.txt', 'Open txt file for file destination path determination');

t1 = fix(clock);                                        % Start the stopwatch

file1 = [pname1 fname1];
legend = imread(file1,'tiff');
legend = double(legend);
legend(455:456,20:20+round(scale)) = 255;

cd (pname1);                                            % Move to //image on HELMET.
directorynames = dir;                                   % List directories.
directorynames = char({directorynames.name});           % Turn directory names to characters.
directorynames = sortrows(directorynames);              % Alphabetize directory names.

cd Lens;                                                % Move to //image/Lens on HELMET.
filenames = dir('*.tif');                               % List only the tif files.
filenames = char({filenames.name});                     % Turn file names to characters.
filenames = sortrows(filenames);                        % Alphabetize file names.

lensspots = zeros(480,640)+255;                         % Create a white image named lenspots.

for pic = 1:size(filenames,1);
  temp = imread(filenames(pic,:),'tiff');               % Read in blank images above tunnel.
  temp = double(temp);                                  % Double precision of images.
  temp = deinterlace(temp);                             % Call deinterlace function.
  temp = 255-temp;                                      % Take the negative of the images.
  mtemp = mean(mean(temp));                             % Calculate mean of the images.
  h = find(temp > mtemp);                               % Find pixels higher than the mean.
  Z = zeros(size(temp));                                % Create black image, Z.
  Z(h) = 255;                                           % Turn Z matrix white at locations h.
  Z = (Z+lensspots)/2;                                  % Average lensspots and Z matrices
  h = find(Z < 255);                                    % Find nonwhite locations in Z.
  lensspots(h) = 0;                                     % Blackout locations h in lensspots matrix.
```

```
end;

for pixel = 1:4
  lensspots(:,1:pixel) = 0;
  lensspots(:,641-pixel:640) = 0;
  h = find(lensspots > 0);
  lensspots(h+1) = 255;
  lensspots(h-1) = 255;
  lensspots(h+480) = 255;
  lensspots(h-480) = 255;
end;

cd ..;                                          % Move back to //image on HELMET.
for position = 5:(size(directorynames)-2)
  cd (pname2);                                  % Move to //processed on SHIELD.
  mkdir (directorynames(position,:));           % Make P## directory on SHIELD.
  cd (pname1);                                  % Move to //image on HELMET.
  cd (directorynames(position,:));              % Move to //image/P# on HELMET.
  subdirectorynames = dir;                      % List subdirectory names.
  subdirectorynames = char({subdirectorynames.name});  % Change subdirectory names to characters
  subdirectorynames = sortrows(subdirectorynames);     % Alphabetize subdirectory names.
  cd Window;                                    % Move to //image/P##/Window on
HELMET.
  filenames = dir('*.tif');                     % List only the tif files.
  filenames = char({filenames.name});           % Turn file names to characters.
  filenames = sortrows(filenames);              % Alphabetize file names.
        intensities = size(filenames,1);        % Create vector as big as the number of
files.
  for picture = 1:size(filenames,1);
    temp = imread(filenames(picture,:),'tiff');
      % Read blank window file to matrix.
    temp = double(temp);                        % Double precision of blank window
matrix.
    temp = deinterlace(temp);                   % Call function to deinterlace image.
    temp = lensfunction(temp,lensspots);        % Call function to eliminate lens spots.
    intensities(picture) = mean(mean(temp));    % Fill vector with mean of blank window
matrix.
  end;
  cd ..;                                        % Move to //image on HELMET.
  for run = 3:(size(subdirectorynames)-1)
    cd (pname2);                                % Move to //processed on SHIELD.
    cd (directorynames(position,:));            % Move to //processed/P## on SHIELD.
    mkdir (subdirectorynames(run,:));           % Make P## directory on SHIELD.
    cd (pname1);                                % Move to //image on HELMET.
    cd (directorynames(position,:));            % Move to //image/P## on HELMET.
    cd (subdirectorynames(run,:));              % Move to //image/P##/Run## on
HELMET.
    picfilenames = dir('*.tif');                % List only the tif files.
    picfilenames = char({picfilenames.name});   % Turn file names to characters.
    picfilenames = sortrows(picfilenames);      % Alphabetize file names.
    for picture = 1:size(picfilenames)
      pic = imread(picfilenames(picture,:),'tiff');  % Fill pic matrix with selected image.
      pic = double(pic);                        % Double precision of pic matrix.
      pic = deinterlace(pic);                   % Call function to deinterlace image.
      pic = lensfunction(pic,lensspots);        % Call function to eliminate lens spots.
      picintensity = mean(mean(pic));           % Determine mean intensity of image.
```

```
        h = matchintensity(intensities,picintensity);          % Call function to match image intensity.
        cd ..;                                                 % Move to //image on HELMET
        cd Window;                                             % Move to //image/Window on HELMET
        winpic = imread(filenames(h,:), 'tiff');               % Read droplet file to matrix.
        winpic = double(winpic);                               % Double precision of droplet matrix.
        winpic = deinterlace(winpic);                          % Call function to deinterlace image.
        winpic = lensfunction(winpic,lensspots);               % Call function to eliminate lens spots.
        cd ..;                                                 % Move to //image on HELMET.
        cd (pname2);                                           % Move to //processed on SHIELD.
        cd (directorynames(position,:));                       % Move to //processed/P## on SHIELD.
        cd (subdirectorynames(run,:));                         % Move to //processed/P##/Run## on
SHIELD.
        winpic = winpic*(picintensity/intensities(h));         % Normalize pixel values of window image.
        pic = motionfunction(pic,winpic,16,6);                 % Call function to correct to moving spark.
        pic = pic + legend;                                    % Add legend image to processed image.
        pic = uint8(pic);                                      % Adjust pixel values of processed image.
        imwrite(pic, picfilenames(picture,:), 'tiff');         % Write modified image to tiff.
        picfilenames(picture,:)                                % Output processed image file name to
screen.
        cd (pname1);                                           % Move to //image on HELMET.
        cd (directorynames(position,:));                       % Move to //image/P## on HELMET.
        cd (subdirectorynames(run,:));                         % Move to //image/P##/Run## on
HELMET
      end;
      cd ..;                                                   % Move to //image on HELMET.
   end;
   cd ..;                                                      % Move to //image on HELMET.
end;

t2 = fix(clock);                                               % Stop the stopwatch.
days = t2(3)-t1(3);
hours = t2(4)-t1(4);
minutes = t2(5)-t1(5);
seconds = t2(6)-t1(6);
if seconds < 0
   seconds = 60+seconds;
   minutes = minutes-1;
end;
if minutes < 0
   minutes = 60+minutes;
   hours = hours-1;
end;
if hours < 0
   hours = 24+hours;
   days = days-1;
end;
msgbox(sprintf('Image processing complete.\n\nTotal elapsed runtime was
%g:%g:%g.',hours,minutes,seconds));
```

A-2 deinterlace.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                                                    %
% Droplet deformation and fragmentation research.                                    %
%                                                                                    %
% Filename: deinterlace.m                                                            %
%                                                                                    %
% Image processing function to:                                                      %
%       1) linearly deinterlace input images and output the image of highest intensity %
%                                                                                    %
% Inputs are:                                                                        %
%       1) image to be deinterlaced                                                  %
%                                                                                    %
% Function written by Logan M. Yanson and Mark R. Phariss                            %
%                                                                                    %
% Last modified Monday, August 31st, 2004                                            %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function out = deinterlace(in);

odd = in;                                   % Fill odd matrix with input image.
even = in;                                  % Fill even matrix with input image.

rows = size(in,1);

for count = 2:rows-1;
  if max(2 == factor(count)) == 1
    odd(count,:) = (odd(count - 1,:) + odd(count + 1,:))/2;
  else
    even(count,:) = (even(count - 1,:) + even(count + 1,:))/2;
  end
end

even(1,:) = even(2,:);

if max(2 == factor(rows)) == 1
  odd(rows,:) = odd(rows - 1,:);
else
  even(rows,:) = even(rows - 1,:);
end

if mean(mean(even)) > mean(mean(odd))       % Output image with higher mean intensity.
  out = even;
else
  out = odd;
end
```

A-3 lensfunction.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                                                 %
% Droplet deformation and fragmentation research.                                 %
%                                                                                 %
% Filename: lensfunction.m                                                        %
%                                                                                 %
% Image processing code to:                                                       %
%               1) remove blemishes on the CCD chip from images                   %
%                                                                                 %
% Inputs are:                                                                     %
%               1) scrw.tif - image of screws at various position in the tunnel   %
%                               used for determining location in the tunnel and image scaling %
%                               located in //image on HELMET                      %
%               2) path.tif - black image with text ('100 micron') written in the bottom lefthand corner %
%                               used to place legend on the bottom of each processed image and %
%                               mark location of main image directory             %
%                               located in //image on HELMET                      %
%                                                                                 %
% Function written by Logan M. Yanson and Mark R. Phariss                         %
%                                                                                 %
% Last modified Monday, August 31st, 2004                                         %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function OUT = lensfunction(IN,spots)

in = mean(mean(IN(241:460,21:620)));
IN = IN-spots;
h = find(IN<0);
IN(h) = in;

OUT = IN;
```

A-4 matchintensity.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                                                 %
% Droplet deformation and fragmentation research.                                 %
%                                                                                 %
% Filename: matchintensity.m                                                      %
%                                                                                 %
% Image processing function to:                                                   %
%                 1) deinterlace images                                           %
%                 2) eliminate dirt on the lens from images                       %
%                 3) eliminate dirt on the tunnel windows from images             %
%                 4) increase contrast (make black and white) to more easily identify droplets %
%                                                                                 %
% Inputs are:                                                                     %
%                 1) scrw.tif - image of screws at various position in the tunnel %
%                                 used for determining location in the tunnel and image scaling %
%                                 located in //image on HELMET                    %
%                 2) path.tif - black image with text ('100 micron') written in the bottom lefthand corner %
%                                 used to place legend on the bottom of each processed image and %
%                                 mark location of main image directory           %
%                                 located in //image on HELMET                    %
%                 3) path.txt - empty text file                                   %
%                                 used to mark a location for saved processed images %
%                                 located in //processed on SHIELD                %
%                 4) lens.tif - blank images taken at a position above the wind tunnel %
%                                 used to subtract lens dirt from images          %
%                                 located in //image/Lens on HELMET               %
%                 5) wndw.tif - blank images taken at various positions in the wind tunnel %
%                                 used to subtract window dirt from images        %
%                                 located in //image/P##/Window on HELMET         %
%                 6) ####.tif - droplet images taken at various position in the wind tunnel %
%                                 images to be processed                          %
%                                 located in //image/P##/Run## on HELMET          %
%                                                                                 %
% Function written by Logan M. Yanson and Mark R. Phariss                         %
%                                                                                 %
% Last modified Monday, August 31st, 2004                                         %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function out = matchintensity(intensitiesvector,intensityscalar)

a = ones(size(intensitiesvector))*intensityscalar;
out = find(min(abs(a-intensitiesvector))==(abs(a-intensitiesvector)));
```

A-5 motionfunction.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                                                  %
% Droplet deformation and fragmentation research.                                  %
%                                                                                  %
% Filename: motionfunction.m                                                       %
%                                                                                  %
% Image processing function to:                                                    %
%                   1) deinterlace images                                          %
%                   2) eliminate dirt on the lens from images                      %
%                   3) eliminate dirt on the tunnel windows from images            %
%                   4) increase contrast (make black and white) to more easily identify droplets %
%                                                                                  %
% Inputs are:                                                                      %
%                   1) scrw.tif - image of screws at various position in the tunnel %
%                                       used for determining location in the tunnel and image scaling %
%                                       located in //image on HELMET               %
%                   2) path.tif - black image with text ('100 micron') written in the bottom lefthand corner %
%                                       used to place legend on the bottom of each processed image and %
%                                       mark location of main image directory      %
%                                       located in //image on HELMET               %
%                   3) path.txt - empty text file                                  %
%                                       used to mark a location for saved processed images %
%                                       located in //processed on SHIELD           %
%                   4) lens.tif - blank images taken at a position above the wind tunnel %
%                                       used to subtract lens dirt from images     %
%                                       located in //image/Lens on HELMET          %
%                   5) wndw.tif - blank images taken at various positions in the wind tunnel %
%                                       used to subtract window dirt from images   %
%                                       located in //image/P##/Window on HELMET    %
%                   6) ####.tif - droplet images taken at various position in the wind tunnel %
%                                       images to be processed                     %
%                                       located in //image/P##/Run## on HELMET     %
%                                                                                  %
% Function written by Logan M. Yanson and Mark R. Phariss                          %
%                                                                                  %
% Last modified Monday, August 31st, 2004                                          %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function out = motionfunction(in,B,rowcrop,columncrop)

C = B(rowcrop+1:480-rowcrop,columncrop+1:640-columncrop);

for r = 1:2*rowcrop;
  for c = 1:2*columncrop;
    S(r,c) = sum(sum(abs(in(r:479-(2*rowcrop)+r,c:639-(2*columncrop)+c)-C)));
  end;
end;

h = min(min(S));
[i,j] = find(S==h);

B = zeros(size(in));
B(i:i+479-(2*rowcrop),j:j+639-(2*columncrop)) = C;
```

```
A = zeros(size(in));
A(i:i+479-(2*rowcrop),j:j+639-(2*columncrop)) = in(i:i+479-(2*rowcrop),j:j+639-(2*columncrop));
out = abs(A-B);
mout = max(max(out));
out = 255/mout*out;
```

APPENDIX B: Image Processing Codes for Multiple Exposure Imaging

B-1 maindroplet.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                                                                   %
% Droplet deformation and fragmentation research.                                                   %
%                                                                                                   %
% Filename: maindroplet.m                                                                           %
%                                                                                                   %
% Image processing code to:                                                                         %
%               1) deinterlace images                                                               %
%               2) eliminate dirt on the lens from images                                           %
%               3) eliminate dirt on the tunnel windows from images                                 %
%               4) increase contrast (make black and white) to more easily identify droplets        %
%                                                                                                   %
% Inputs are:                                                                                       %
%               1) scrw.tif - image of screws at various position in the tunnel                     %
%                               used for determining location in the tunnel and image scaling       %
%                               located in //image on HELMET                                        %
%               2) path.tif - black image with text ('100 micron') written in the bottom lefthand corner %
%                               used to place legend on the bottom of each processed image and      %
%                               mark location of main image directory                               %
%                               located in //image on HELMET                                        %
%               3) path.txt - empty text file                                                       %
%                               used to mark a location for saved processed images                  %
%                               located in //processed on SHIELD                                    %
%               4) lens.tif - blank images taken at a position above the wind tunnel                %
%                               used to subtract lens dirt from images                              %
%                               located in //image/Lens on HELMET                                   %
%               5) wndw.tif - blank images taken at various positions in the wind tunnel            %
%                               used to subtract window dirt from images                            %
%                               located in //image/P##/Window on HELMET                             %
%               6) ####.tif - droplet images taken at various position in the wind tunnel           %
%                               images to be processed                                              %
%                               located in //image/P##/Run## on HELMET                              %
%                                                                                                   %
% Code written by Logan M. Yanson                                                                   %
%                                                                                                   %
% Last modified Tuesday, October 2nd, 2003                                                          %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear all; clc;                                         % Clearing of memory.

global imagenum position velocity velocityposition...   % Globalize important variables
tempsizepos tempsizevelpos tempsizevel imagecounter

% Display GUI that determines the number of images to be examined.
prompt = {'Enter number of images to be examined '};
title = 'Image number?';
lines = 1;                                              % Number of lines in each input of the GUI.
def = {'2'};                                            % Default values in the GUI.
imagenum = inputdlg(prompt,title,lines,def);
```

```
imagenum = char(imagenum);                          % Change from string to ASCII.
imagenum = str2num(imagenum);                       % Change from ASCII to number.

% Run droplet.m for each image to be examined.
% Fill position matrix.
for imagecounter = 1:imagenum;
  droplet;                                          % Run droplet.m.
  temp=size(pos);
  sizepos=temp(1);                                  % Size of pos matrix from droplet.m.
  temp=size(vel);
  sizevel=temp(1);                                  % Size of vel matrix from droplet.m.
  temp=size(velpos);
  sizevelpos=temp(1);                               % Size of velpos matrix from droplet.m.
  j = 1;                                            % Initialize counter for pos matrix.
  k = 1;                                            % Initialize counter for vel and velpos
matricies.
  if imagecounter == 1;
    tempsizepos = 1;
    tempsizevel = 1;
    tempsizevelpos = 1;
  end;
  for i = tempsizepos:tempsizepos+sizepos-1;
    position(i) = pos(j);                           % Fill position matrix location i.
    j = j+1;                                        % Increment location in pos matrix.
  end;
  for i = tempsizevel:tempsizevel+sizevel-1;
    velocity(i) = vel(k);                           % Fill velocity matrix location i.
    velocityposition(i) = velpos(k);                % Fill velocityposition matrix location i.
    k = k+1;                                        % Increment location in vel and velpos
matricies.
  end;
  tempsizepos = tempsizepos+sizepos;                % Determine start of next position matrix
filling.
  tempsizevel = tempsizevel+sizevel;                % Determine start of next velocity matrix
filling.
  tempsizevelpos = tempsizevelpos+sizevelpos;       % Determine start of next velocityposition
matrix filling.
end;
position = transpose(position);
velocity = transpose(velocity);
velocityposition = transpose(velocityposition);
tempsizepos = size(position);                       % Determine size of position matrix.
sizeposition = tempsizepos(1);

% Setup for histogram.
% Display GUI that determines some parameters for the histogram.
prompt = {'Enter number of bars:', 'Enter x min:', 'Enter x max'};
title = 'Histogram Parameters';
lines = 1;                                          % Number of lines in each input of the GUI.
def = {'10','2','3'};                               % Default values in the GUI.
histparameters = inputdlg(prompt,title,lines,def);
histparameters = char(histparameters);              % Change from string to ASCII.
histparameters = str2num(histparameters);           % Change from ASCII to number.
numberofbars = histparameters(1);                   % Reassign number of bars variable.
xmin = histparameters(2);                           % Resassign minimum x variable.
xmax = histparameters(3);                           % Reassign maximum x variable.
```

```
barwidth = (xmax-xmin)/numberofbars;                    % Width of histogram bars.
barposition(1) = xmin;
for i = 2:numberofbars+1;
        barposition(i) = barposition(i-1)+barwidth;     % Location of beginning and end of bars.
end;
barposition = transpose(barposition);

population = zeros(numberofbars,1);                     % Initialize droplet population matrix.
for i = 2:numberofbars+1;
  for j = 1:sizeposition;
    if ((position(j) > barposition(i-1)) & (position(j) < barposition(i)));
      population(i) = population(i)+1;                  % Filling of droplet population matrix.
    end;
  end;
end;

population = population/max(population);                % Normalize droplet population
distribution.

for i = 1:numberofbars
  barcenter(i) = (barposition(i)+barposition(i+1))/2;   % Determine the center of each bar.
end;
barcenter=transpose(barcenter);

% Output data from position, and velocity and velocityposition to a pair of files.
% Output velocity output file is space delimited.  Use excel to put into columns.
[fname1,pname1] = uiputfile('*.txt','Enter filename for position datafile');
file1 = [pname1,fname1];
pos = [transpose(position)];
fid = fopen(file1,'w');
fprintf(fid, '%6.5f\n',pos);
fclose(fid);

[fname1,pname2] = uiputfile('*.txt','Enter filename for velocity datafile');
file2 = [pname2,fname2];
vel = [transpose(velocity); transpose(velocityposition)];
fid = fopen(file2,'w');
fprintf(fid,'%6.5f %6.5f\n',vel);
fclose(fid);

% Plot histogram and ask to save.
bar(barcenter,population,1)
        % Plot histogram.
axis([xmin xmax 0 max(population)+1]);
xlabel('Position (in)', 'FontSize', 16);
ylabel('Normalized Droplet Population', 'FontSize', 16);
button = questdlg('Save histogram?','Save','Yes','No','Yes');
if strcmp(button, 'Yes');
  [fname3,pname3] = uiputfile('*.jpg','Enter filename of histogram');
  file3 = [pname3,fname3];
  saveas(gcf,file3,'jpg');
end;

% Plot velocity vs. position and ask to save.
plot(velocityposition,velocity,'.k');                   % Plot scatter plot of velocity vs. position.
axis([xmin xmax 0.85*min(velocity) 1.15*max(velocity)]);
```

```
xlabel('Position (in)', 'FontSize', 16);
ylabel('Velocity (m/s)', 'FontSize', 16);
button = questdlg('Save plot?','Save','Yes','No','Yes');
if strcmp(button, 'Yes');
        [fname4,pname4] = uiputfile('*.jpg','Enter filename of plot');
  file4 = [pname4,fname4];
  saveas(gcf,file4,'jpg');
end;
```

B-2 droplet.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                                                                                       %
% Droplet deformation and fragmentation research.                                           %
%                                                                                                                       %
% Filename: droplet.m                                                                                        %
%                                                                                                                       %
% Image processing code to:                                                                             %
%                  1) clean multiple exposure image                                                 %
%                  2) turn image from grayscale to black and white                            %
%                  3) find pixel location of droplets and convert into inches           %
%                  4) calculate droplet velocities                                                     %
%                                                                                                                       %
% Inputs are:                                                                                                   %
%                  1) multiple exposure image                                                         %
%                  2) two blank images                                                                     %
%                  3) resolution of image                                                                  %
%                  4) bounds of exposed droplets (upper, lower, left and right)          %
%                  5) number of useful exposures                                                    %
%                  6) real position of top of the image (inches)                               %
%                  7) scale (inch/pixel)                                                                    %
%                  8) time between exposures (microseconds)                               %
%                                                                                                                       %
% Code written by Logan M. Yanson                                                                %
%                                                                                                                       %
% Last modified Tuesday, September 30th, 2003                                                %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Input three deinterlaced images.  One multiple exposure images and two blank images.
% Input a variety of information about the image and camera exposure settings.
% Code outputs a new black and white image file, and the location of the droplet and its velocity.

clear variables;                                            % Clear nonglobal variables.

global imagenum position velocity velocityposition...       % Globalize important variables
tempsizepos tempsizevelpos tempsizevel imagecounter

% Input images
[fname1, pname1] = uigetfile('*.tif', 'Open multiple exposure droplet image');
file1 = [pname1 fname1];
[fname2, pname2] = uigetfile('*.tif', 'Open blank image #1');
file2 = [pname2 fname2];
[fname3, pname3] = uigetfile('*.tif', 'Open blank image #2');
file3 = [pname3 fname3];

% Create image matrices and double precision them.
multiple = imread(file1, 'tiff');
multiple = double(multiple);
blank1 = imread(file2, 'tiff');
blank1 = double(blank1);
blank2 = imread(file3, 'tiff');
blank2=double(blank2);

% Subtract multiple exposure droplet image from the average of the blank images.
```

```
diff = abs((blank1+blank2)/2 - multiple);

% Create a new black and white image.
% When diff is greater than 50, turn that pixel black.
% Droplets should turn black, everything else white.
blackwhite = zeros(size(multiple));                    % Create zero blackwhite matrix, same size
as multiple matrix.
i = find(diff > 50);                                   % Determine locations in diff matrix greater
than 50.
blackwhite(i) = 255;                                   % Fill 255 in blackwhite matrix at locations
in i matrix.

% Save the black and white image and display it on the screen.
[fname4, pname4] = uiputfile('*.tif', 'Enter black and white droplet image filename');
file4 = [pname4 fname4];
imwrite(blackwhite, file4, 'tiff');
colormap(gray);                                        % Change color scheme of image display to
grayscale.
image(blackwhite);                                     % Display black and white image.

% Display GUI that determines the resolution of the images.
prompt = {'Pixels in horizontal direction','Pixels in vertical direction'};
title = 'Exposure resolution';
lines = 1;                                             % Number of lines in each input of the GUI.
def = {'640','480'};
                    % Default resolution in the GUI.
resolution = inputdlg(prompt,title,lines,def);
resolution = char(resolution);                         % Change from string to ASCII.
resolution = str2num(resolution);                      % Change from ASCII to number.

% Display GUI that determines the box that surrounds the multiple exposure droplets and the number of
exposures.
prompt = {'Enter left of centerline','Enter right of centerline','Enter bottom of last exposure',...
    'Enter top of first exposure','Enter number of exposures'};
title = 'Exposure number and position';
lines = 1;                                             % Number of lines in each input of the GUI.
def = {'275','325','100','250','6'};                   % Default values in the GUI.
dropletbox = inputdlg(prompt,title,lines,def);
dropletbox = char(dropletbox);                         % Change from string to ASCII.
dropletbox = str2num(dropletbox);                      % Change from ASCII to number.
totalexposurenumber = dropletbox(5);                   % Change varible name (number of
exposures in the frame).

% Question box appears that asks if you want to resize the axis to look at only the droplets.
button = questdlg('Do you want to resize the axis?','Resize','Yes','No','No');
if strcmp(button,'Yes');
  axis([dropletbox(1) dropletbox(2) dropletbox(3) dropletbox(4)]);
end;

% Display GUI that determines the position at the of the top of the image, scaling and time between
exposures.
prompt = {'Enter the position of the top of the image (inches)','Enter the scale of the image (inch/pixel)',...
  'Enter the time between exposures (microseconds)'};
title = 'Various';
lines = 1;                                             % Number of lines in each input of the GUI.
def = {'2','0.003','10'};                              % Default values in the GUI.
```

```
frameinformation = inputdlg(prompt,title,lines,def);
frameinformation = char(frameinformation);                    % Change from string to ASCII.
frameinformation = str2num(frameinformation);                 % Change from ASCII to number.
positionoftop = frameinformation(1);                          % Change variable name (position of top of
image).
scale = frameinformation(2);                                  % Change variable name (scale of image).
timebetweenexposures = frameinformation(3);                   % Change variable name (time in between
exposures).

% Determination of line of resolution locations of droplet centers.
summation=zeros(resolution(2));                               % Zero summation matrix.
for r=dropletbox(3):dropletbox(4);                            % Increment pixels in vertical direction.
  for c=dropletbox(1):dropletbox(2);                          % Increment pixels in horizontal direction.
    if blackwhite(r,c) == 0;                                  % If droplet present (blackwhite=0),
increment summation 1.
      summation(r) = summation(r)+1;                          % Greater value of summation, greater
chance of droplet center.
    end;
  end;
end;
j = find(summation>0);                                        % Finding location of nonzero elements of
summation matrix.
temp = size(j);                                               % Holder for size of j matrix.
sizej = temp(1);                                              % Final variable name for size of j matrix.
count = 1;                                                    % Initialize counter for lines of resolution of
droplet.
exposure = 1;                                                 % Initialize droplet exposure number.
total = 0;                                                    % Initialize total resolution lines.
for k = 2:sizej;
  if j(k)-j(k-1) == 1;                                        % Check if location of resolution lines are
nonconsecutive.
    if count == 1;                                            % Make sure to include first resolution line
of each droplet.
      total = j(k-1);
    end;
    total = total+j(k);                                       % Add resolution line to running total.
    count = count+1;                                          % Increment count by one.
  else;
    respos(exposure,1) = total/count;                         % Calculate average location.
    count = 1;                                                % Reinitialize resolution line counter.
    total = 0;                                                % Reinitialize total resolution lines.
    exposure = exposure+1;                                    % Increment droplet exposure number.
  end;
  if (k == sizej) & (j(k)-j(k-1) > 1);                        % Quick fix for when the last resolution line
is its own droplet.
    respos(totalexposurenumber) = j(sizej);
  end;
end;

% Convert from lines of resolution position to actual position.
for p = 1:exposure;
  pos(p) = scale*respos(p)+positionoftop;                     % Position in inches of droplet at exposure
p.
end;
pos = transpose(pos);
```

```
% Determine droplet velocities.
for q = 1:exposure-1;
   vel(q) = (pos(q+1)-pos(q))/timebetweenexposures;          % Velocity of droplet.
   velpos(q) = (pos(q+1)+pos(q))/2;                          % Position of the droplet velocity.
end;
vel = transpose(vel);
velpos = transpose(velpos);
```

APPENDIX C: Computational Model Code

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                                                             %
% Droplet deformation and fragmentation research.                             %
%                                                                             %
% Filename: euler.m                                                           %
%                                                                             %
% Image processing code to:                                                   %
%         1) deinterlace images                                               %
%         2) eliminate dirt on the lens from images                           %
%         3) eliminate dirt on the tunnel windows from images                 %
%         4) increase contrast (make black and white) to more easily identify droplets %
%                                                                             %
% Inputs are:                                                                 %
%         1) scrw.tif - image of screws at various position in the tunnel     %
%                  used for determining location in the tunnel and image scaling %
%                  located in //image on HELMET                               %
%         2) path.tif - black image with text ('100 micron') written in the bottom lefthand corner %
%                  used to place legend on the bottom of each processed image and %
%                  mark location of main image directory                      %
%                  located in //image on HELMET                               %
%         3) path.txt - empty text file                                       %
%                  used to mark a location for saved processed images         %
%                  located in //processed on SHIELD                           %
%         4) lens.tif - blank images taken at a position above the wind tunnel %
%                  used to subtract lens dirt from images                     %
%                  located in //image/Lens on HELMET                          %
%         5) wndw.tif - blank images taken at various positions in the wind tunnel %
%                  used to subtract window dirt from images                   %
%                  located in //image/P##/Window on HELMET                    %
%         6) ####.tif - droplet images taken at various position in the wind tunnel %
%                  images to be processed                                     %
%                  located in //image/P##/Run## on HELMET                     %
%                                                                             %
% Code written by Logan M. Yanson                                             %
%                                                                             %
% Last modified Tuesday, August 26, 2003                                      %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Euler method code with variable temporal grid spacing

clear all; clc; clf;                              % clear the memory

npoints = 10000;                                  % number of data points

gamma = 1.4;                                      % ratio of specific heats for air
R = 287;                                          % gas constant for air

diameterdroplet = 60e-6;                          % initial droplet diameter
volumedroplet = (4/3)*pi*(diameterdroplet/2)^3;   % initial droplet diameter
rhodroplet = 789.2;                               % density of droplet
sigma = 0.02232;                                  % surface tension of droplet
rhot =1.225;                                      % stagnation density of air [kg/m^3]
Tt = 300;                                         % stagnation temperature of air [K]
```

```
pt = 760;                                   % stagnation pressure of air [torr]

% Tunnel geometric constants
xentrance = 2;                              % distance from tip of DoD to entrance [in].
length = 1 + xentrance;                     % length from DoD tip to end of tunnel [in].

%Time information
TimeTotal = 0.0179;                         % Total run time of the model [s]
TimeStep = TimeTotal/npoints;               % Time step [s]
% Filling of the Time array
for points = 1:npoints+1;
  Time(points,1) = TimeStep*points;
end;

% Zeroing of variables
x = zeros(npoints,1);                       % Downstream position
M = zeros(npoints,1);                       % Centerline Mach number
T = zeros(npoints,1);                       % Centerline Static Temperature
p = zeros(npoints,1);                       % Centerline Static Pressure
rho = zeros(npoints,1);                     % Centerline Static Density
v = zeros(npoints,1);                       % Centerline velocity
vd = zeros(npoints,1);                      % Droplet Velocity
Md = zeros(npoints,1);                      % Droplet Mach number
Weber = zeros(npoints,1);                   % Droplet Weber number
kd = zeros(npoints,1);                      % Drag coefficient
c = zeros(npoints,1);

% Initialization of variables
vd(1,1) = 2;
T(1,1) = Tt;
p(1,1) = pt;
rho(1,1) = rhot;

for points = 1:npoints;
  % Setting of TimeStep
  Md(points,1) = vd(points,1)/sqrt(gamma*R*T(points,1));
  Mrel(points,1) = (M(points,1)-Md(points,1));
  vrel(points,1) = Mrel(points,1)*sqrt(gamma*R*Tt*(1+0.5*(gamma-1)*M(points,1)^2));
  Weber(points,1) = rhodroplet*vrel(points,1)^2*diameterdroplet/sigma;
  % Determination of Charters and Thomas drag coefficient
  if Mrel(points,1) < 0.5;
    kd(points,1) = 0.192;
  else
    kd(points,1) = 0.3198+0.2987*(Mrel(points,1)-1)-0.0809...
      *(Mrel(points,1)-1)^2-0.3606*(Mrel(points,1)-1)^3;
  end;
  % Determination of constant C, which isn't really constant
  c(points,1) = kd(points,1)*rho(points,1)...
    *diameterdroplet^2/(rhodroplet*volumedroplet);
  x(points+1,1) = x(points,1)+v(points,1)...
    *TimeStep+(1/c(points,1))*log((1/(v(points,1)...
    -vd(points,1)))/(c(points,1)*TimeStep...
    +1/(v(points,1)-vd(points,1))));
  holder = 39.37*x(points+1,1);             % temporary conversion from meters to
inches.
  % Mach number down the centerline of the tunnel
```

```
  if holder <= xentrance;
    M(points+1,1) = (1.0552/xentrance^12)*(holder^12);
  else;
    M(points+1,1) = -0.2089*holder^4+2.7252...
      *holder^3-13.227*holder^2+28.718*holder-21.932;
  end;
  T(points+1,1) = Tt*(1+(gamma-1)/2*M(points+1,1)^2)^-1;
  p(points+1,1) = pt*(1+(gamma-1)/2*M(points+1,1)^2)^(gamma/(1-gamma));
  rho(points+1,1) = rhot*(1+(gamma-1)/2*M(points,1)^2)^(1/(1-gamma));
  v(points+1,1) = M(points+1,1)*sqrt(gamma*R*T(points+1,1));
  vd(points+1,1) = v(points,1)...
    -1/(c(points,1)*TimeStep+1/(v(points,1)-vd(points,1)));
end;

% Convert droplet positions from meters to inches
x = 39.37*x;

% Data plots
% Plot of centerline Mach number vs. position
subplot(2,2,1);
plot(x(1:npoints,1), M(1:npoints,1),'.k');
xlabel('position');
ylabel('Centerline Mach Number');
axis([0 3 0 2]);

% Plot of droplet Mach number vs. position
subplot(2,2,2);
plot(x(1:npoints,1), Md(1:npoints,1),'.k');
xlabel('position');
ylabel('Droplet Mach Number');
axis([0 3 0 0.8]);

% Plot of relative Mach number vs. position
subplot(2,2,3);
plot(x(1:npoints,1), Mrel(1:npoints,1),'.k');
xlabel('position');
ylabel('Relative Mach Number');
axis([0 3 0 1.5]);

% Plot of drag coefficiant vs. position
subplot(2,2,4);
plot(x(1:npoints,1), kd(1:npoints,1), '.k');
xlabel('position');
ylabel('kd');
axis([0 3 0.15 0.4]);
```