APPLICATION OF ADVANCED LASER AND OPTICAL DIAGNOSTICS TOWARDS NON-THERMOCHEMICAL EQUILIBRIUM SYSTEMS

A Dissertation

by

ANDREA GRACE HSU

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

May 2009

Major Subject: Chemistry

APPLICATION OF ADVANCED LASER AND OPTICAL DIAGNOSTICS TOWARDS NON-THERMOCHEMICAL EQUILIBRIUM SYSTEMS

A Dissertation

by

ANDREA GRACE HSU

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Approved by:

Chair of Committee,	Simon North
Committee Members,	Rodney Bowersox
	Jaan Laane
	Dong Hee Son
Head of Department,	David Russell

May 2009

Major Subject: Chemistry

ABSTRACT

Application of Advanced Laser and Optical Diagnostics Towards Non-Thermochemical Equilibrium Systems. (May 2009)

Andrea Grace Hsu, B.S., The University of Texas at Austin
Chair of Advisory Committee: Dr. Simon North

The Multidisciplinary University Research Initiative (MURI) research at Texas A&M University is concerned with the experimental characterization of non-thermal and non-chemical equilibrium systems in hypersonic (Mach>5) flowfields using experimental diagnostics, and is an interdisciplinary collaboration between the Chemistry and Aerospace Engineering departments. Hypersonic flight conditions often lead to non-thermochemical equilibrium (NTE) state of air, where the timescale of reaching a single (equilibrium) Boltzmann temperature is much longer than the timescale of the flow, meaning that certain molecular modes such as vibrational modes, may be much more excited than the translational or rotational modes of the molecule leading to thermal-nonequilibrium. A nontrivial amount of energy is therefore contained within the vibrational mode, and this energy cascades into the flow as thermal energy, affecting flow properties through the process of various vibrational-vibrational (V-V) and vibrational-translational (V-T) energy exchanges between the flow species. The research is a fundamental experimental study of these NTE systems and involves the application of advanced laser and optical diagnostics towards hypersonic flowfields. The research is broken down into two main categories: the application and adaptation of existing laser and optical techniques towards characterization of NTE, and the development of new molecular tagging velocimetry techniques which have been demonstrated in an NTE flowfield, but may be extended towards a variety of flowfields.

ACKNOWLEDGEMENTS

I would like to thank my committee chair, Dr. Simon North, and my committee members, Dr. Rodney Bowersox, Dr. Jaan Laane, and Dr. Dong Hee Son for their guidance and support during my time here at Texas A&M. I would especially like to thank Dr. North for his constant mentorship and patience. I have learned a great deal the last few years and I owe it all to his leadership and motivation.

I would also like to thank my friends and colleagues for their support. In particular, I would like to thank my fellow group members. I was very fortunate to be a part of and to be able to work closely with two groups of graduate students. To the North group, thank you to former graduate students Jiho Park and Hahkjoon Kim for their mentorship, and to fellow students Kate Perkins, Laura Ruebush, Kristin Dooley, Buddhadeb Ghosh, Justine Geidosch, Carrie Dean, Jacob Dean, and Rodrigo Sanchez-Gonzales. Even though sometimes it seemed like all of our research topics were so different, I am very appreciative of all the help, support, and friendship they offered. To the Bowersox group, thank you to former graduate students Isaac Ekoto, Jason Garcia, and Dipankar Sahoo, who taught me valuable lessons on working at the wind tunnel facilities. Also, thanks to current students Mike Semper, Nathan Tichenor, Scott Peltier, and TJ Fuller for their friendship, support, and muscle when I needed a hand. I owe most of what I know about wind tunnels to information osmosis from working side by side with them on a daily basis. I would also like to thank the low speed wind tunnel staff, particularly Will Seward and Carl Johnson, for their patience and guidance on learning the basics of machining. Last but definitely not least, I'd like to thank Monica Gonzales and Colleen Leatherman for making everything I needed to do related to purchasing and paperwork so much easier.

Thanks also go to friends I've met outside of my studies for making my time at Texas A&M University a great experience. I came to realize that the people at A&M are very special; they welcomed me into their family even though I was a former UT graduate and I will never forget the memories of midnight yells, ring dunkings, sand volleyball, and football games. I’d especially like to thank my boyfriend, Shane Schouten, for his patience and support. He was always there to bring me late night coffees and dinners, or to just keep me company at the lab. Today I’m proud to call myself an Aggie thanks to all of these people.

And of course I would like to thank my family, especially my mother and my brother for their love and support.

TABLE OF CONTENTS

Page

CHAPTER Page

CHAPTER Page

CHAPTER Page

LIST OF TABLES

LIST OF FIGURES

FIGURE Page

FIGURE Page

FIGURE Page

FIGURE Page

FIGURE Page

FIGURE Page

FIGURE Page

CHAPTER I

INTRODUCTION

A. Background and Motivation

The proposed Multidisciplinary University Research Initiative (MURI) research at Texas A&M University is concerned with the experimental characterization and theoretical modeling of non-thermal and non-chemical equilibrium systems in hypersonic (Mach>5) shear layers using several experimental diagnostics, and is an interdisciplinary collaboration between the Chemistry and Aerospace Engineering departments. It is well known that hypersonic flight conditions can result in non-thermochemical equilibrium (NTE) state of air, where the timescale of reaching a single (equilibrium) Boltzmann temperature is much longer than the timescale of the flow. As air travels over a hypersonic vehicle and through shocks, the translational and rotational temperatures are excited rapidly to several thousands of K, while the vibrational temperature lags behind. This trend is due to the fact that vibrational energy transfer requires orders of magnitude more collisions (typically 10^2 to 10^4) than translational or rotational energy transfers, which require on the order of only 10-100 collisions. Therefore, behind the shock, the translational and rotational temperatures are much higher than the vibrational temperatures, and the translational and rotational temperatures decay, while the vibrational temperature experiences a slight increase as all three molecular modes equilibrate on a timescale which is significant compared to the bulk flow velocity.

In the present research, NTE is generated where the vibrational mode is isolated and studied; high vibrational temperatures are produced (~2000K) with cold translational and rotational temperatures (~300K). In this case, the vibrational temperature decays, while the translational and rotational temperatures increase during equilibration,

This dissertation follows the style of *American Institute of Aeronautics and Astronautics*.

opposite to what is encountered in true hypersonic flight as the air passes through a shock. The purpose of this experimental setup is twofold. First, it would represent a real situation where the post-shock equilibrated air experiences an expansion, causing the vibrational temperature to freeze, and the rotational and translational temperatures to decay rapidly. Second, the research is a fundamental study of vibrational energy transfer and cascade, where the manner in which this energy cascade couples to flow properties is not well understood. In this research, we expect that only a small amount (typically on the order of a few percent) of the total molecules may exist in the excited vibrational states. However, the energetic spacing between vibrational levels is larger (by roughly an order of magnitude) compared to rotational spacings. A nontrivial amount of energy, about 10-20% of the total flow enthalpy, is therefore contained within the vibrational mode, and this energy cascades into the flow, affecting flow properties through the process of various vibrational-vibrational (V-V) and vibrational-translational (V-T) energy exchanges between the flow species. It is then useful to model the V-V and V-T energy exchanges to understand the mechanisms of the vibrational energy flow. In addition, chemical reactions may take place on a timescale where steady-state is in invalid assumption, so the kinetics of these reactions must also be considered within the model. As a result, recently there has been increased interest in the characterization of non-thermal equilibrium systems [e.g., see Cheng (1995), Scalabrin (2005), Candler (1989), Osipov (2006), Roy (2006)]. At Texas A&M, the research is part of a larger effort in which the experimental data collected will serve as a benchmark to which fluid dynamics modeling such as that discussed in Bowersox (2008) can be validated.

The research experimental goals involve generation of the NTE encountered during hypersonic flight in a controlled environment within the laboratory using two methods: capacitively-coupled radio-frequency (CCRF) plasma discharge and laser photodissociation. In both cases, the NTE produced cause the molecules to exist in a highly vibrationally excited state (thousands of K), while maintaining “cold” (nearly room temperature, 300 + /- 10 to 30 K) rotational and translational temperatures (low-temperature, low fluctuation). The NTE is then characterized using a suite of laser and optical diagnostics. The research is broken down into two main categories: the application and adaptation of existing laser and optical techniques towards characterization of NTE, and the development of new techniques which have been demonstrated in an NTE flowfield, but may be extended towards a variety of flowfields.

The first chapter will give background and motivation for the project, as well as literature reviews for the highlighted laser diagnostics discussed in the remainder of the text. These will include mainly nitric oxide (NO) planar laser-induced fluorescence (PLIF) and molecular tagging velocimetry (MTV). Other diagnostics of more minor importance are discussed briefly in their respective sections. The theoretical background for NO PLIF and MTV will also be discussed in the introduction. The second chapter comprises the hardware and diagnostics system, and serves as the test bed on which the laser and optical diagnostics are applied. This chapter will discuss the design of the chambers to be used, and the generation of the CCRF plasma in the test facilities. The third chapter contains the fundamental theories and experimental application of NO PLIF including discussion on LIF excitation scanning, fluorescence signal modeling, saturation and fluorescence linearity studies, and quenching. Also, to understand the vibrational energy cascade post-CCRF, a section is included on vibrational kinetics modeling. The modeling will be compared to experimental results using vibrational broadband N_2 CARS and the agreement between the model and experiment will be discussed. The effects of impurities such as ambient CO_2 and H_2O on vibrational relaxation will be explored. Resolution and experimental determination of the imaging system modulation transfer function will be included. Lastly, a computer-based study laying the groundwork for NO PLIF application towards low temperature, low fluctuation flowfields such as those expected to occur in the flow post-CCRF will be discussed with experimental validation. This study covers issues such as image processing and appropriate choice of probed rotational lines for rotational temperature mapping using NO PLIF.

Chapters IV, V, and VI cover major experiments. Chapter IV is the characterization of the CCRF plasma, and verification that the plasma vibrationally excites the molecules while imparting only minor gas heating and negligible molecular dissociation using both broadband and narrowband emission. Chapter V comprises two experiments. Both utilize the photodissociation of NO_2 to form a local non-thermoequilibrium distribution of NO in a highly underexpanded axisymmetric free jet flowfield. The first experiment introduces the use of probing $NO_{v=1}$ (formed from photodissociation) for single component (x or streamwise direction) velocimetry, and discusses the advantages of this technique over an equivalent technique probing $NO_{v=0}$ in the case of flowfields with naturally occurring NO. The application of the technique and the advantages and disadvantages of probing either $NO_{v=1}$ or $NO_{v=0}$ as a function of temperature (below 300K to 20,000K) are discussed. The second experiment involves the simultaneous probing of $NO_{v=1}$ and $NO_{v=0}$ to extract a vibrational temperature map. The focus of this experiment is the understanding of the chemical kinetic mechanisms which drive the vibrational relaxation of $NO_{v=1}$ through comparison of kinetic modeling and computational fluid dynamic (CFD) simulations with experimental results. Chapter VI is an extension of the first experiment from Chapter V, and explores two main MTV setups and their application in low and high quenching environments towards two-component velocity mapping. Chapters V and VI then discuss the development of new laser diagnostics techniques, and the application of these techniques towards other flowfields is discussed. Single shot image analysis is included to show the utility of these techniques in short run-time facilities such as blow-down wind tunnels and/or shock tubes (runtimes in the range of single laser shot to a few minutes) or in statistical analyses of dynamic flowfields.

In summary, the research goals were: 1) generation of the NTE flowfield; 2) characterization of the NTE flowfield using laser and optical diagnostics; 3) development of software and image processing programs to acquire and analyze these data; 4) development of new molecular tagging laser diagnostic techniques for single-component and two-component velocity mapping; 4) modeling and understanding of the chemical kinetic mechanisms for vibrational decay in both CCRF flowfield and NO_2 photodissociation studies.

B. LITERATURE REVIEW

B.1 PLIF Pioneering Studies

PLIF is a powerful laser diagnostic technique which can be used to yield a number of flow parameters due to the dependence on the fluorescence signal to flowfield pressure, number density, temperature, and velocity (through use of the Doppler shift). However, the discussion of PLIF in this research will be limited to its discussion towards rotational and vibrational temperature mapping. In this scheme, two molecular states are excited nearly simultaneously to yield either rotational or vibrational temperature maps depending on the two states selected. Planar laser-induced fluorescence originated in the 1980's. Pioneering work in this area include work by Cattolica (1981), which discusses the development of two-line OH PLIF applied in a combusting flowfield. The first reported single-shot PLIF measurements appeared in Applied Optics [see Kychakoff (1982)] where a 100x100 intensified photodiode array was used for OH visualization in a combusting methane/air flowfield; suggestions for probing of NO in such a flowfield for temperature mapping was reported in Kychakoff (1984) due to increased sensitivity of fluorescence signal to temperature compared to OH. This suggestion was carried out in a separate paper published later in the same year by Kychakoff (1984), where the first NO PLIF images were shown for visualization of NO number density again in a methane/air flame.

Throughout the 1980's, the development of NO PLIF progressed and the technique was applied towards a variety of flowfields. Hanson (1988) provides an excellent review of PLIF works completed in the early and mid 1980's. Temperature imaging using PLIF is discussed, with reference to both one and two-line strategies, as well as the progress of detectors. Molecular velocimetry measurements using NO PLIF by utilizing broadband Doppler-shifted fluorescence was first demonstrated in an underexpanded jet flowfield (in both averaged and single-shot images) in Paul (1989) and Palmer (1993). Because NO can be generated naturally in a shock tube through dissociation of O_2 and N_2, NO PLIF could be easily applied towards these flowfields. The first known application of NO PLIF for single-shot vibrational temperature mapping through the D←X (0,1) transition for observation of the post-shock vibrational decay of NO in a shock was shown in McMillin (1990). Nearly concurrently, NO PLIF was used for single-shot two-line rotational temperature imaging in a shock tube in McMillin (1991). This study, although the single-shot images were not instantaneous (since the images were obtained from separate experimental runs), discussed in detail the aspects of fluorescence linearity and temperature sensitivity, which are especially important towards this research. Temperature maps in the range of 900 to 1850K were obtained in this study, which quoted single-shot temperature fluctuations of 16%. Palmer (1992) reported single-shot combined rotational temperature and velocity Doppler-shifted fluorescence measurements in an underexpanded jet flowfield generated in a shock-tube. Again, in this study, the image pairs needed for the two-line NO PLIF technique were not acquired simultaneously. Additional sources which include excellent discussions of NO PLIF theory and background include Lee (1993) and McMillin (1993). In the early 1990's, NO PLIF grew in popularity. Lachney (1998) discussed the imaging of mean temperature in the range of 50-300K and pressure in a supersonic bluff wake. Pressure maps were obtained through the linear relationship between fluorescence signal and pressure (valid for low-quenching environments, such as N_2).

The first simultaneously obtained, single-shot two-line NO PLIF measurements were reported in Palmer (1993) in a vibrationally nonequilibrium underexpanded jet (produced in a shock tube facility) by excitation of four different transitions ((0,0) Q_1 + $P_{21}(5)$, (0,0) Q_2 + $R_{12}(6)$, (0,1) Q_1 + $P_{21}(5)$, (0,1) P_2 + $Q_{12}(6)$) using NO PLIF. Vibrational temperatures in this flow were very high, ranging from 1000-2000K, due to freeze-out after the gas exits the nozzle, while rotational temperatures dropped to very low values (~100K) within the expansion. Single-shot vibrational temperature maps were calculated, as were frame-average rotational temperature maps; these temperature maps were compared with method of characteristics (MOC) simulations which included a simplified model for vibrational relaxation with reasonable agreement.

B.2 PLIF Recent Studies

Over the years, NO PLIF has continued to grow in popularity and is today used in a variety of flowfields for many purposes. Some interesting applications will be included in this section, although the breadth of the literature involving recent NO PLIF applications greatly outnumber those mentioned here. The use of NO PLIF has recently emerged into four main types of studies: qualitative and quantitative visualization studies, scanned studies, high-speed studies, and combination studies.

Within the visualization studies, Jacobsen (2008) reported the use of NO PLIF for imaging of naturally occurring NO in a plasma torch for visualization of plasma-assisted ignition phenomena applicable towards a scramjet engine. Slices of NO PLIF data, where the laser sheet was swept across the volume of interest have been reconstructed in software to yield 3-D maps in Danehy (2006 and 2008). This method of data visualization was applied towards reaction control systems (RCS) jets in a hypersonic flow towards application in capsule entry vehicles. In addition, thick probe laser sheets were used in Danehy (2008) and 3-D flowfield visualization was created via cross-eyed views and red/blue anaglyphs. Byrne (2006) reported the use of NO PLIF to qualitatively visualize and address flow uniformity in a hypersonic nozzle. NO PLIF was also applied towards solid propellant studies in Ryan (2006) to study the interaction between electrothermal plasma radiation and JA2 solid propellant. In this study, the NO was naturally occurring as one of the solid propellant decomposition products, and the distribution of NO was probed at various times after plasma discharge to study the progression of JA2 propellant decomposition. Murugappan (2006) utilized NO PLIF towards towards visualization of mixing enhancement of CSSI (controlled supersonic swirling injector) in SCRAMJET engines. NO PLIF has also been applied towards high pressure (1-60 bar) systems in Lee (2006) with extensive discussion on the minimization of signal-degrading aspects of these environments such as pressure broadening, laser attenuation/high quenching, and interference from other flame natural species.

Studies using a scanning NO technique sweep the probe laser wavelength over at least one single NO absorption transition in order to fit the fluorescence images (pixel-by-pixel, in the case of PLIF) to a known NO LIF spectra in order to extract flow properties; these techniques are time-averaged techniques. Kronemayer (2005) and Lee (2007) demonstrated that this technique, when used for a point-wise (LIF, not PLIF) measurement, yielded a highly accurate temperature measurement (300K + /-1K). Naik (2008) reported the use of this technique for simultaneous, but averaged, velocity, pressure, and temperature mapping. The velocity was determined from Doppler shifted fluorescence, pressure was obtained from the linewidth due to its temperature insensitivity, and temperature was determined from overall intensity. The results showed good agreement with CFD simulations and N_2 CARS measurements. This technique requires the careful correction of laser sheet intensity, but is highly sensitive and in general yields very accurate measurements. High-speed techniques focus on the development of "burst" lasers which can be used in conjunction with NO PLIF for examination of dynamics in unsteady, evolving flowfields. Jiang (2008) reported the demonstration of such a high-speed "burst" laser system which could produce train of pulses with energies of ~0.5 mJ/pulse at a rate of 100 kHz.

PLIF has been combined with other techniques, the most prevalent being the combination of PLIF with either Rayleigh scattering or Particle Imaging Velocimetry (PIV). NO PLIF and Rayleigh scattering [see Sutton (2006)] were applied simultaneously to a turbulent flame for measurements of mixture fraction, scalar dissipation, temperature, and fuel consumption rate by tailoring flame stochiometry to preserve NO as a passive scalar (i.e. a nonreactive molecule which is not consumed in the flame). PLIF/PIV studies are numerous. For example, Lemaire (2004) reported the combination of CH PLIF/PIV using silicon dioxide particles and n-heptane droplets in a flame. Hishida (2000) reported the use combined PLIF/PIV in an aqueous flowfield for simultaneous temperature and velocity measurements. These measurements are used to calculate turbulent heat flux. The temperature measurements are achieved through temperature-dependent fluorescence of Rhodamine B dye, and velocity measurements are achieved through tracing of 5 um polystyrene particles. Lastly, Frank (1999) demonstrated the use of combined OH PLIF/PIV in a turbulent premixed natural-gas/air flame to measure conditional velocities. In this study, submicron alumina particles were seeded which could withstand the high temperatures within the flame for the PIV measurements. Other studies documenting the combination of OH PLIF with PIV in combustion flowfields include Meyer (2004) and Rehm (1997). Acetone PLIF/PIV was reported in Tsurikov (1999).

As seen from the literature survey, the number of recent NO PLIF papers addressing vibrational temperature mapping is limited, although Palmer (1993) provides an excellent fundamental discussion of the application of NO PLIF towards vibrational temperature mapping. In order to couple NO PLIF with other techniques such as PIV, the NO PLIF NTE characterization must be conducted on a single-shot basis. Unfortunately, no studies were found which documented the use of combined NO PLIF/PIV. Also, no studies were found which address the application of NO PLIF towards single-shot, instantaneous, two-line rotational temperature mapping where the expected temperature variations are less than 300 + /- 10 to 30K (low-temperature, low-temperature-fluctuation). This research will address the application of NO PLIF towards vibrational temperature mapping in a NTE flowfield. Also, a separate chapter will specifically address the application of NO PLIF towards low temperature, low temperature fluctuation flows such as those encountered in NTE.

B.3 Molecular Tagging

Velocimetry techniques are varied and consist of two main categories: probe-based (intrusive) and laser-based techniques (non-intrusive). Three widely used nonintrusive techniques include laser Doppler velocimetry (LDV), particle image velocimetry (PIV), and molecular tagging velocimetry (MTV). Of these techniques, PIV and MTV are planar techniques, and can provide an instantaneous velocity map of the flowfield. Table 1.1 lists the advantages and disadvantages of PIV and MTV:

Table 1.1 Comparison of PIV and MTV

PIV	
Advantages	**Disadvantages**
Signal rich	Requires uniform particle seeding
Excellent resolution, limited by particle seeding density	Vibrational relaxation effects difficult to characterize due to particle size distribution
Software commercially available	Does not track shocks well
Well established technique	Scattering near walls or surfaces can be problematic
Less extensive setup	
MTV	
Advantages	**Disadvantages**
Tracks flow exactly	Software not commercially available
Easy uniform seeding	Generally signal-poor, resolution dependent on image processing, i.e. Smoothing
Vibrational relaxation can be modeled (no size distribution)	Generally requires more extensive setup, depending on probed molecule
Scattering can be temporally eliminated	

PIV relies on tracking particles through Mie scattering, and thus is wavelength independent, making the selection of lasers simple. Usually, green 532 nm light (second harmonic of Nd:YAG) is used because the light is in the visible range and therefore easier to align. Double-pulsed Nd:YAG lasers are available specifically for PIV applications relatively inexpensively. Also, Mie scattering is a few orders of magnitude larger in intensity than molecular fluorescence, and since the scattered light is at the same frequency (532 nm), a fast-gated visible camera can be used for image acquisition. Interline (double frame) cameras specifically for PIV are also commercially available (Cooke Corp.). These cameras are very advantageous because they eliminate the need to warp the images from two separate cameras to the same field of view. Because PIV is signal-rich, its resolution is excellent, and is only limited by particle seeding density. Software is commercially available for PIV image processing, and contain very sophisticated procedures for particle location through correlation functions for sub-pixel accuracy. However, PIV has some disadvantages. PIV diagnostics requires the use of particles, and although these particles can be extremely small, they cannot follow the flow as precisely as molecules, and deviations become large in the presence of strong shocks [see Huffman (2009)]. In this paper, a study of the ability of particles to accurately track flow in a supersonic, underexpanded jet was conducted by comparing PIV with MTV measurements. The study reported 5% errors near the jet exit using the smallest particles (100 nm). However, near the Mach disk, the particles did not accurately track the flow and worsened as particle size was increased from 100 nm to 1 um. Fig. 1.1 was adapted from Huffman (2009) to illustrate this point.

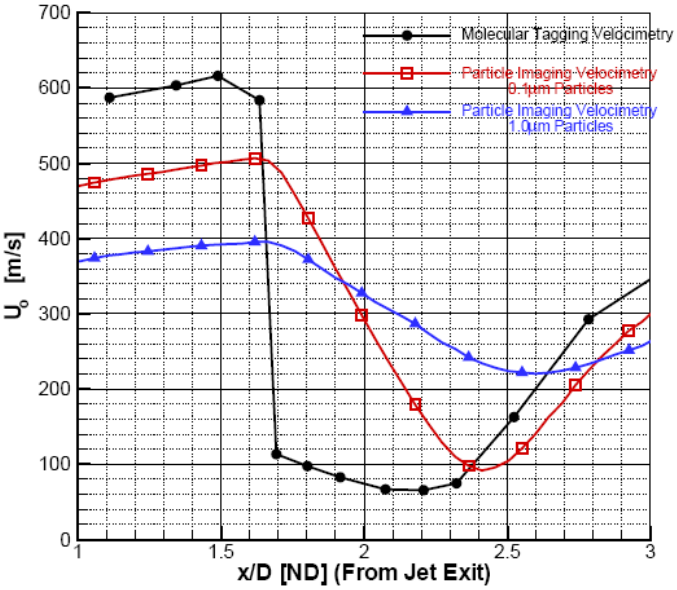


Fig. 1.1 Comparison of MTV and PIV showing the degrading effect of increasing particle size in the vicinity of a strong shock [adapted from Huffman (2009)]

The data shows PIV and MTV data in the presence of a strong normal shock. Before the flow encounters the shock, the flow is accelerated to about 600 m/s, as shown. The MTV captures this acceleration accurately, while the PIV data experiences lag which increases with particle size. Even small particles 100 nm in diameter experience axial velocity uncertainties of about 20% when compared to MTV data. The shock is also not recovered by even the small 100 nm PIV particles. PIV also requires particle seeding, which is not preferred in some facilities such as quiet facilities, where particles can damage the highly polished walls. In addition, uniform seeding can be difficult. This research is interested in flowfield vibrational relaxation, and these effects are difficult to characterize with PIV. Because particles are much larger than the molecules, they essentially act as a surface and vibrationally quench molecules upon collision. Therefore, the size distribution of the particles and the seeding density must be known accurately to model the vibrational kinetics accurately, a difficult task. Last, scattering near walls or surfaces can be problematic in PIV. Since the signal is Mie scattering, ways to decrease wall scattering include painting the surfaces either with a black paint or fluorescent dyes such as Rhodamine 6G which absorb 532 nm light and emit in the red, and then using a filter in front of the camera.

In MTV, the molecules of interest are “tagged”, usually by a “write” laser and probed at a known time later by a “read” laser. This process also has disadvantages and advantages. It relies on tracking molecules through molecular fluorescence (or phosphorescence, in some cases). The main advantage of MTV is that the molecules accurately track the flow, and there are no problems near strong shocks. Also, it is easy to uniformly seed the flow, since the molecules will not settle out. Vibrational relaxation can be modeled since there is no worry of size distribution, and only requires an accurate measurement of seeding density, which can be easily recorded. Scattering problems can be temporally eliminated; since MTV uses fluorescence, which has a much longer lifetime than scattering, the imaging cameras can be gated to begin the acquisition after the scattering has passed. In terms of disadvantages, the main disadvantage of MTV is that it is in general signal-poor, since, as mentioned before, fluorescence signal is much weaker than Mie scattering. Also, because MTV is molecule-specific, a particular probe wavelength is required, and laser intensity at these wavelengths is typically much lower than the 532 nm light used in PIV. For example, NO fluorescence is used in these studies, and excitation of NO (A-X) transitions requires a tunable UV laser at around 226 nm. Typical laser energies at this wavelength are about 1-2 mJ/pulse, whereas 532 nm light out of an Nd:YAG can range from 100 to 1000 mJ/pulse. As a result of lower signal to noise ratio (SNR) the raw images must be smoothed, and smoothing reduces the spatial resolution. Experimental setup can be extensive and expensive; however, this is not always a requirement. MTV setups are highly varied and price for setup depends on the probed molecule of choice. For example, in addition to requiring a tunable laser source, UV intensified cameras and optics are required for probing NO. However, if a molecule which fluoresces (or phosphoresces) in the visible spectrum is chosen (e.g. acetone) the issues with requiring more expensive UV imaging equipment could be eliminated. The following paragraphs will focus on MTV applications found in the literature.

MTV encompasses a wide range of techniques that can be applied in both gaseous and liquid flowfields. The liquid flowfield techniques generally use phosphorescent molecules which are dissolved into the flow and illuminated via laser excitation as in Bohl (2004), Hu (2006), and Hu (2006). Gaseous MTV techniques are divided into seeded and unseeded techniques. Unseeded techniques generally fall into 4 main areas: ozone tagging velocimetry (OTV), hydroxyl tagging velocimetry (HTV), Raman excitation plus laser-induced electronic fluorescence (RELIEF), and nitric oxide (NO) tagging velocimetry. OTV involves the formation of ozone via photodissociation of molecular oxygen naturally occurring in the flow, where the ozone is then photodissociated to form vibrationally hot O_2 and simultaneously probed via Schumann-Runge fluorescence as in Pitz (1996). HTV involves the photodissociation of water and probing of OH by LIF [e.g., see Ribarov (2004), Pitz (2005), Wehrmeyer (1999)]. RELIEF involves LIF probing of tagged vibrationally excited O_2 molecules as in Miles (1993). NO tagging velocimetry are conducted in shock tubes where NO is naturally formed as in Danehy (2003) or by photodissociation of air [e.g., see van der Laan (2003), Nandula (2004)]. In the case of Danehy (2003), the NO fluorescence is imaged in hypersonic flowfield, so that the displacement is tracked through its fluorescence lifetime.

In both cases, reported studies have been limited to probing of the ground vibrational state of NO ($NO_{v=0}$) at 226 nm. Seeded techniques mainly consist of tert-butyl nitrite photodissociation, acetone tagging, biacetyl tagging, and NO_2 photodissociation. Both acetone [e.g., see Lempert 2002 and 2003] and biacetyl tagging as in Ismailov (2006) involve laser excitation and fluorescence probing (or phosphorescence in the case of biacetyl). NO_2 [e.g., see Nakaya (2005) and Orlemann (1999)] and tert-butyl nitrite as in Kruger (1999) photodissociation techniques involve the photodissociation of these molecules and probing of the product NO by LIF.

MTV has been applied using both single-line and grid techniques. The single-line techniques provide only one component of velocity, while the grid techniques give two components of velocity in the laser plane by observing the warping of the grid, i.e. the movement of the intersection points of the grid. Currently, no studies that address the simultaneous mapping of vibrational temperature and velocity using a single technique were found. Also, this research will address both the single-line and grid techniques. First, the single-line technique will be used to demonstrate capabilities of a novel application of the NO_2 photodissociation MTV technique which will emphasize the utility of probing $NO_{v=1}$ in the NTE flowfield created through photodissociation of NO_2. This technique would be especially useful in a flowfield where background NO would mask $NO_{v=0}$ fluorescence by photodissociation. This would apply in, for example, a combustion flowfield (T = 1000 – 2000 K), where $NO_{v=0}$ is naturally occurring in larger quantities compared to $NO_{v=1}$. The effect of temperatures up to 20000 K on the advantage of probing $NO_{v=1}$ over $NO_{v=0}$ will be explored. The study will add an extra twist and discuss the simultaneous extraction of vibrational temperature and velocity maps using the same photodissociation technique. Second, the photodissociation technique will be extended from the single-line velocimetry (yielding one component of velocity) to two-component velocity mapping using a two dimensional photodissociation grid. Both studies will be conducted in a highly underexpanded axisymmetric free jet flowfield.

C. Theoretical Background

C.1 LIF Background: Fluorescence Signal

The simplest method of representing the LIF process is via a 2-level model, where the transitions are displayed in the schematic shown in Fig. 1.2 [e.g. see Eckbreth (1988), McMillin (1993)]:

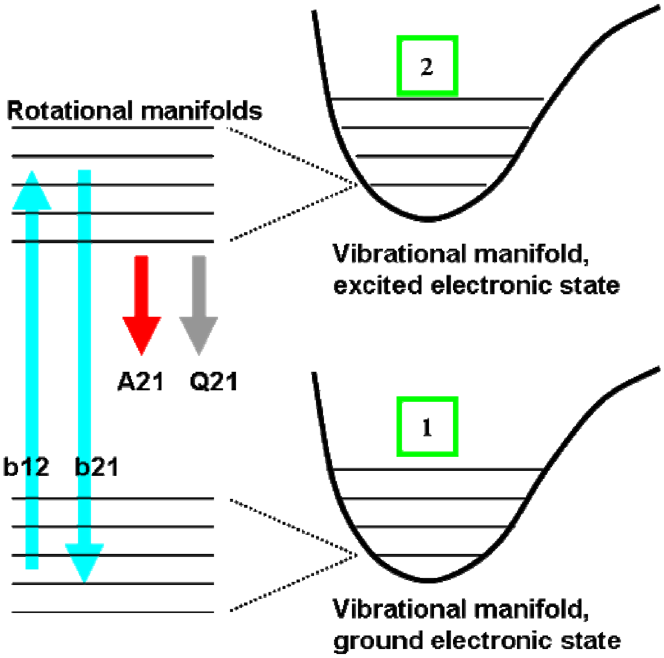

Rotational manifolds
2
1
A21
Q21
b12
b21
Vibrational manifold, excited electronic state
Vibrational manifold, ground electronic state

Fig. 1.2 Two-level LIF energy diagram

Starting with the rate equations for the populations of level 1 (ground) and level 2 (excited), the time dependent population of level 1 can be written as follows, where the nomenclature and symbols are defined in the Appendix.

$$\frac{dN_1}{dt} = -b_{12}N_1 + N_2\left(A_{21} + Q_{21} + b_{21}\right), \text{ where } b_{12} = B_{12}\frac{I}{c} \qquad \text{(Eq. 1-1)}$$

$$\frac{dN_2}{dt} = -\frac{dN_1}{dt} \text{ and } N_1^0 = N_1 + N_2 \qquad \text{(Eq. 1-2)}$$

and using the assumption of steady state,

$$\frac{dN_1}{dt} = 0 = N_2(b_{21} + A_{21} + Q_{21}) - b_{12}(N_1^0 - N_2) \qquad \text{(Eq. 1-3)}$$

we can solve for the population of level 2,

$$N_2 = \frac{N_1^0 b_{12}}{b_{21} + A_{21} + Q_{21} + b_{12}} = \left(\frac{N_1^0 B_{12}}{B_{12} + B_{21}}\right)\left(\frac{1}{1 + \frac{I_{sat}}{I}}\right) \qquad \text{(Eq. 1-4)}$$

$$\text{where } I_{sat} = \frac{c(A_{21} + Q_{21})}{(B_{21} + B_{12})} \qquad \text{(Eq. 1-5)}$$

The fluorescence signal intensity, S_f, in a PLIF experiment is proportional to population of level 2:

$$S_f \propto N_2 A_{21}, \text{ so } S_f \propto \left[\left(\frac{N_1^0 B_{12}}{B_{12} + B_{21}}\right)\left(\frac{1}{1 + \frac{I_{sat}}{I}}\right)\right] A_{21} \qquad \text{(Eq. 1-6)}$$

In reality, the fluorescence signal depends on other factors, which will be denoted as:

$$S_f = N_2 A_{21}\eta \text{ , where } \eta = (hv_{emitted})V\frac{\Omega}{4\pi}\left[g(v_{abs}, \Delta v_{abs}, v_{las}, \Delta v_{las})\right] \qquad \text{(Eq. 1-7)}$$

The final expression for fluorescence signal is written:

$$S_f = \left[\left(\frac{N_1^0 B_{12}}{B_{12} + B_{21}}\right)\left(\frac{1}{1 + \frac{I_{sat}}{I}}\right)\right] A_{21}\left[(hv_{emitted})V\frac{\Omega}{4\pi}\left[g(v_{abs}, \Delta v_{abs}, v_{las}, \Delta v_{las})\right]\right] \qquad \text{(Eq. 1-8)}$$

Examination of Eq. 1-8 yields two limits of the fluorescence signal with laser intensity: saturated and unsaturated limits. In the saturated limit, $I >> I_{sat}$ and the following formula for fluorescence signal results:

$$S_{f\_saturated} = \left(\frac{N_1^0 B_{12}}{B_{12} + B_{21}}\right) A_{21}\left[(hv_{emitted})V\frac{\Omega}{4\pi}\left[g(v_{abs}, \Delta v_{abs}, v_{las}, \Delta v_{las})\right]\right] \qquad \text{(Eq. 1-9)}$$

where the fluorescence signal is independent of laser intensity and quenching. Ideally, it would be best to operate in this region, however, because of the loss of laser intensity through use of a laser sheet rather than a line, and because of laser beam inhomogeneities, it is difficult to operate in the fully saturated regime throughout the field of view. Operation in the partially saturated regime is undesirable as the relationship between fluorescence signal and laser intensity is non-linear, so successful temperature measurements in this regime require that both probed states in the two-line method be saturated to exactly the same degree. Alternatively, the unsaturated limit is obtained by taking the limit, $I << I_{sat}$,

$$S_{f\_unsaturated} = \left(\frac{N_1^0 B_{12} I}{c} \right) \left(\frac{A_{21}}{A_{21} + Q_{21}} \right) \left[(h\nu_{emitted}) V \frac{\Omega}{4\pi} \left[g(\nu_{abs}, \Delta\nu_{abs}, \nu_{las}, \Delta\nu_{las}) \right] \right] \quad \text{(Eq. 1-10)}$$

where the term $\left(\frac{A_{21}}{A_{21} + Q_{21}} \right)$ is commonly called the fluorescence efficiency, because it represents the fraction of number of excited molecules which relax through fluorescence divided by the total number of excited molecules; the higher the fluorescence efficiency, the lower the effect of quenching on the total fluorescence signal. The unsaturated fluorescence signal is linearly dependent on laser intensity, and therefore operation in this regime is referred to as the linear fluorescence regime. In this research, all PLIF measurements for temperature mapping will operate in the linear regime.

C.2 LIF Background - Fluorescence Signal Ratio for Temperature Measurements

As seen in the section above, the resulting expression for unsaturated fluorescence signal is a complicated function of many flow parameters (pressure, temperature, and velocity). By taking the ratio of two simultaneously obtained fluorescence signals, the dependencies on all parameters except temperature can be eliminated; the ratio then is reduced to a function of the Boltzmann population fraction of the probed states and an experimentally determined calibration constant as follows:

$$R = \frac{S_{f2}}{S_{f1}} = C_{12} f_B \text{, where } f_B = \left[e^{\frac{-\Delta E_{21vib}}{kT_{vib}}} \frac{\left(2J_2''+1\right)}{\left(2J_1''+1\right)} e^{\frac{-\Delta E_{21rot}}{kT_{rot}}} \right] \qquad \text{(Eq. 1-11)}$$

As shown in Palmer (1993), probing two different rotational states which originate from the same vibrational state will yield the rotational temperature, and similarly, probing two identical rotational states which originate from different vibrational states will yield the vibrational temperature. The selection of lines is an important issue [see McMillin (1993)]. Briefly, by taking the derivative of Eq. 1-11, the following equation results which gives the temperature uncertainty as a function of the choice of rotational lines, (i.e. ΔE):

$$\frac{\partial T}{T} = \left(\frac{kT}{\Delta E} \right) \frac{\partial R}{R} \qquad \text{(Eq. 1-12)}$$

The propagation of error in the relative uncertainty of the temperature measurement is a function of the uncertainty in the fluorescence ratio, and the energy spacing between the selected transitions. Therefore, to minimize the temperature uncertainty, it is beneficial to choose two states which are energetically far apart (large ΔE); however, in practice, there is an experimental limit since higher states are very scarcely populated at temperatures near 300K, and as a result, the noise in the fluorescence ratio increases, leading to larger $\frac{\partial R}{R}$. Therefore, selection of transition states is a tradeoff between sensitivity and signal to noise.

C.3 Molecular Tagging Velocimetry Background

As stated previously, molecular tagging velocimetry techniques are based on tracking molecules as they displace through the flowfield. For one component of velocity, the initial image is tracked in one direction, and the velocity is calculated in a simple manner for all points along the tracked line, as shown in Fig. 1.3. The concept of MTV is simple; much of the intricacies of the technique lay in the image processing and the method of tracking the molecules.

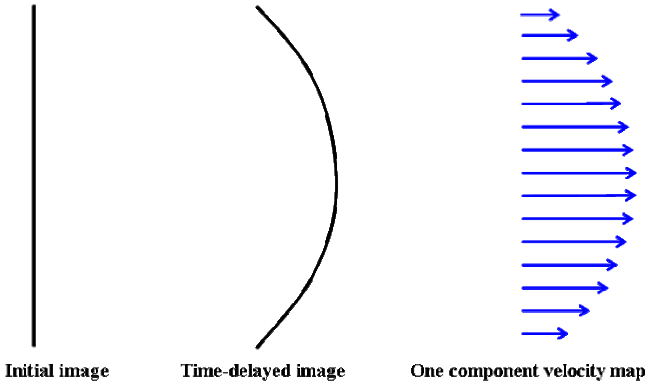


Fig. 1.3 One component molecular tagging velocimetry

For two components of velocity, a grid is introduced into the flow, and the intersection points of the grid are located and tracked through time, as shown in Fig. 1.4:

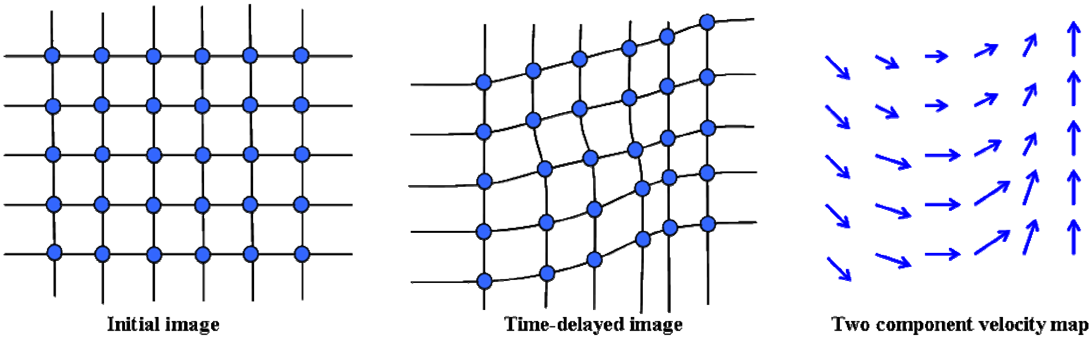


Fig. 1.4 Two component molecular tagging velocimetry

Velocimetry requires two temporally separated images, an initial image and a time delayed image. The velocity is calculated by dividing the spatial displacement of the tracked molecules by the temporal separation between the images. Spatially averaged and instantaneous velocities are defined simply as,

$$u_{\text{avg}} = \frac{\Delta x}{\Delta t} \qquad u_{\text{instant}} = \lim_{\Delta t \to 0}\left[\frac{\Delta x}{\Delta t}\right] \qquad \text{(Eq. 1.13)}$$

where Δx is the spatial displacement and Δt is the temporal delay.

CHAPTER II

EXPERIMENTAL SETUP AND HARDWARE

A. THE LASER AND OPTICAL DIAGNOSTICS SYSTEM

The experimental laser setup for the research documented in this dissertation consisted of a combination of a Spectraphysics PRO 290-10 Nd:YAG laser, two Spectraphysics LAB 150-10 Nd:YAG lasers, and two Sirah Cobrastretch tunable dye lasers. The Sirah dye lasers included calibration programs to auto tune the angle of the BBO doubling crystal to laser wavelength. Additional hardware included a high-resolution 0.66 m, triple grating, SPEX spectrometer, an Oriel broadband spectrometer (MS125 1/8m with LineSpec CCD detector). The SPEX spectrometer was fitted with options for PMT scanning, as well as an electron-multiplying (EM) CCD (Andor, DU970N-BV). Both spectrometers were fitted with fiber-optic couplings. Two Andor iStar ICCD cameras (model DH734, 16-bit) fitted with Nikon 105mm F/4.0 UV lenses and extension rings for close-up viewing were used for imaging NO PLIF. Photodiodes (Thorlabs, DET10A) and PMT were used for scanning the dye laser wavelength to the appropriate NO transitions. The overall timing for the entire experiment was controlled by a digital delay generator (Berkeley Nucleonics Corporation, model DG565). Timing details for each of the experiments will be discussed in further detail in their respective chapters.

A simplified layout of the diagnostics systems in relation to the hypersonic and subsonic facilities within the National Aerothermochemistry Laboratory at Texas A&M University is shown (Fig. 2.1), along with a digital photo of the laboratory (Fig. 2.2):

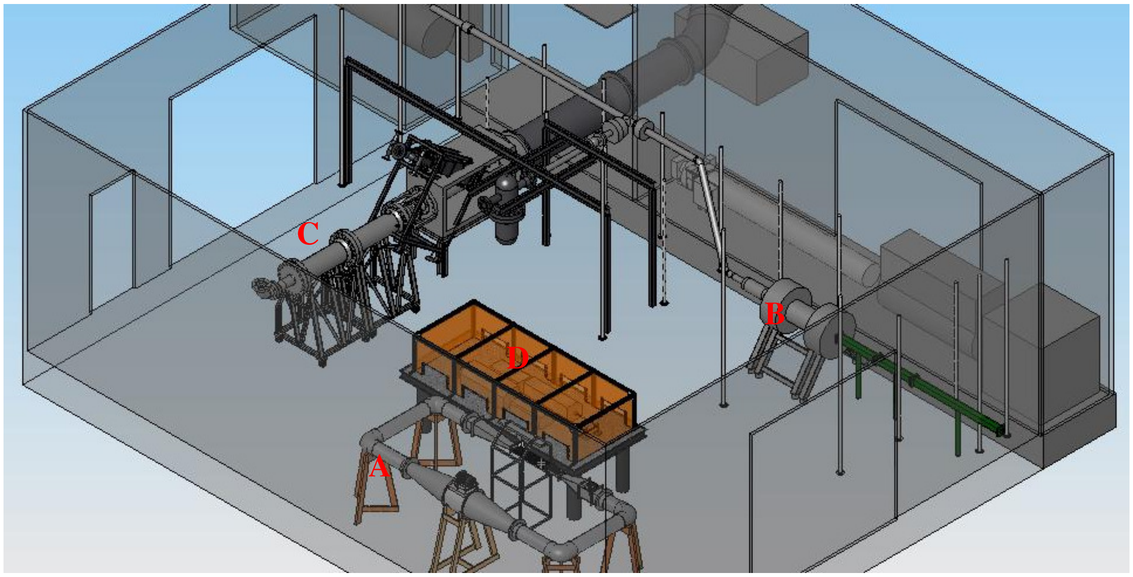


Fig. 2.1 National Aerothermochemistry Laboratory at Texas A&M University layout (schematic)

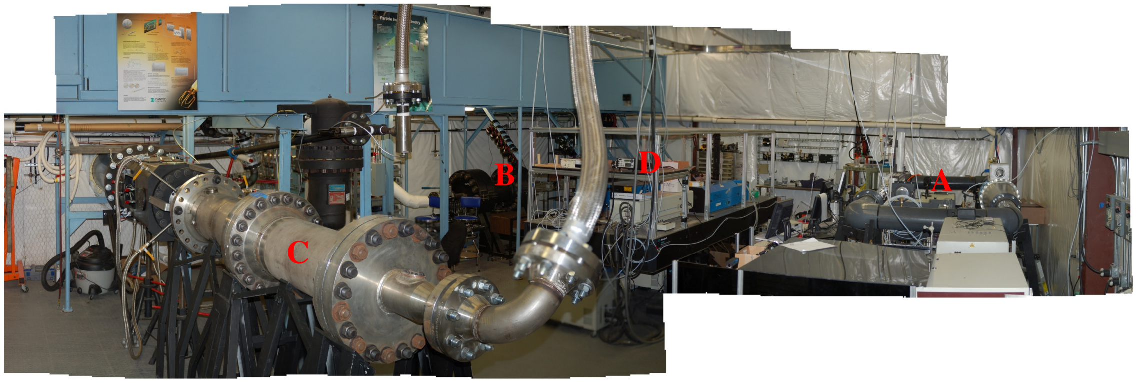


Fig. 2.2 National Aerothermochemistry Laboratory at Texas A&M University (digital photo)

The diagnostic system is enclosed within the orange box situated in the center of the laboratory. The subsonic, supersonic, hypersonic facilities, and diagnostics center are labeled A, B, C, and D respectively in both figures. In this way, the laser beams can be directed towards any of the three facilities within the lab. For further information involving the design, construction, and operation of these facilities, the reader is directed towards dissertations written by Semper (2009), Tichenor (2009), and Fuller (2009) with the exception of the design and construction of the subsonic test section, which will be described in the following section.

B. Chambers and Associated Hardware

B.1 Portable Vacuum and Calibration Cell

This cell was versatile and was used in a number of experiments. It was used either as a flow cell or a static cell. This cell was an improvement to an earlier version, shown in Fig. 2.3:

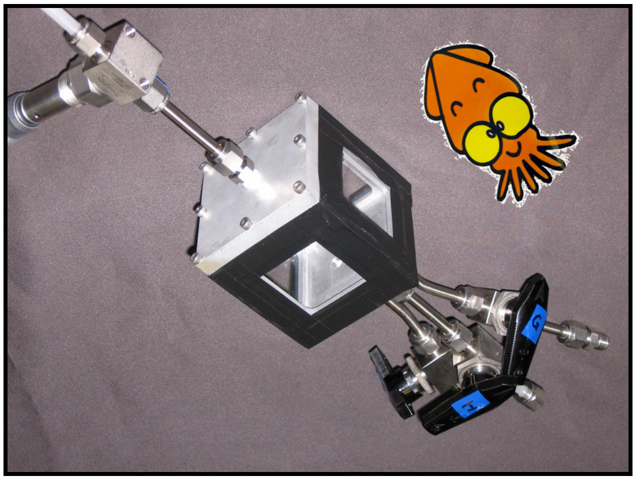


Fig. 2.3 SQUID chamber (first version)

The cell was sentimentally named SQUID, machined out of a single block of aluminum and served as a calibration cell for many preliminary experiments. The second version was modeled after the first but was much easier to machine and convenient to use. As a slow-flow or static cell, it served as the calibration cell for fundamental studies such as measurement of fluorescence linearity with laser intensity and location of probed NO rotational lines. As a fast-flow cell, it was attached to a Roots blower pumping system (~250 cfm pumping speed) to create an underexpanded jet flowfield for demonstration of hypersonic velocity laser diagnostics. The chamber is shown in Fig. 2.4:

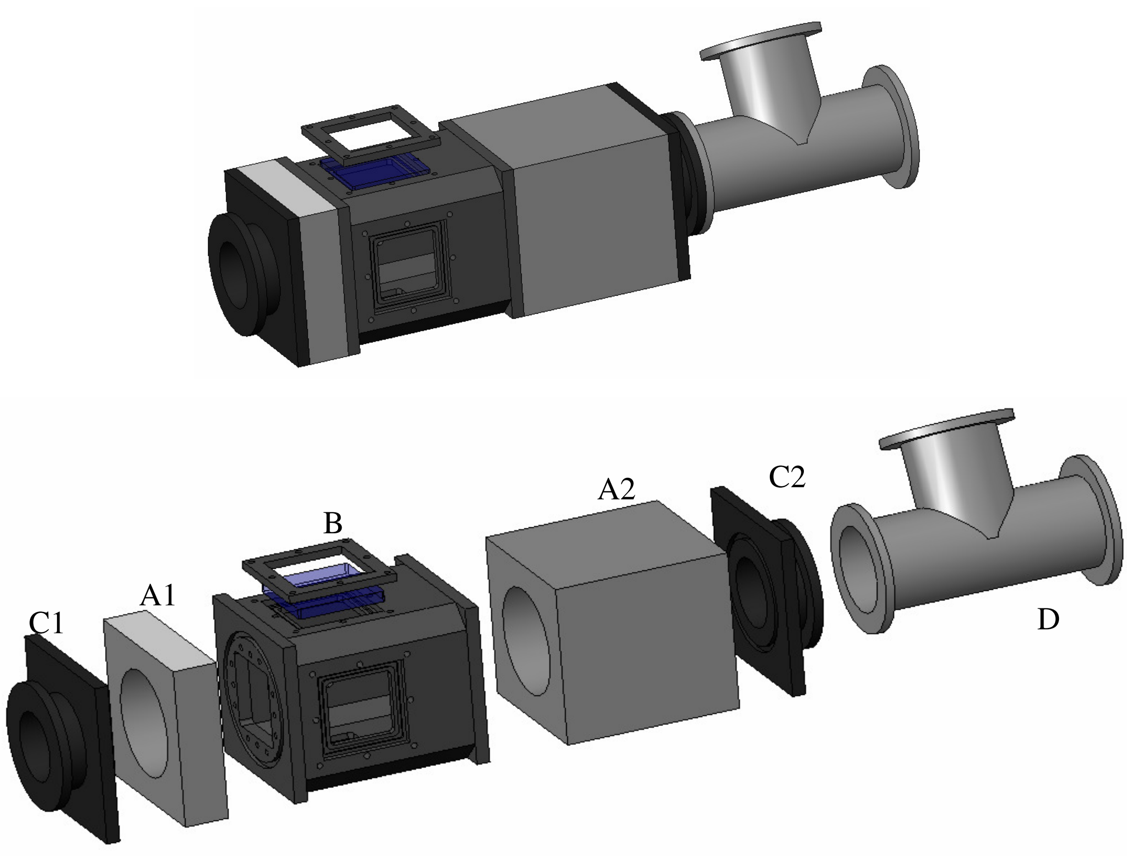


Fig. 2.4 SQUID Chamber assembled and exploded view (second version)

Parts A and B were ordered from Thorlabs (Aluminum XT95 mm railing, XT95B100 sealed instrumentation box) and modified with four O-ring sealed insets for 2”x2”x1/4” UV fused silica windows (Edmund Optics, NT47-231) and mounting holes inside of the chamber for holding a dot-card mount. The fused silica windows were held in place by 3/16” brass window covers. This allowed windows to be removed and cleaned without disassembling the chamber. Parts C and D were ordered from McMaster-Carr. No alterations were needed.

The versatility of the cell stemmed from the fact that many of its parts were interchangeable as well as commercially available and inexpensive. Vacuum tube caps were ordered from McMaster-Carr (4518K61) to attach to points C1, C2, and D and modified for various experiments. For example, for wavelength calibration, a cap was modified with two 1/8” NPT taps, one for slow flow into the chamber (controlled by MKS 1179 flowmeters), and the other for a pressure transducer (MKS 902 piezo transducer). A second blank would be modified with only one 1/8” NPT tap for pump out to a Welch 1224 pump and attached at C2. A third blank was modified with an inset 1” round fused silica window and attached at the right leg of D. A PMT was placed outside the window to detect fluorescence and the chamber would be pumped out using the upper tee leg at D. For the underexpanded jet configuration, an adapter was ordered (McMaster-Carr, 4518K47) and modified with small set screws to hold a ¼” stainless steel tube orthogonal to the face of the cap, and the nozzle was inserted through C1 and A1 until barely visible at the edge of the window. A Roots blower pump system was used at D. In the molecular tagging studies, the cap with the 1” round fused silica window was again used at the right leg of D to allow a fifth optical access window. The entire chamber could also be sealed off (static cell) for examination of diffusion rates. In addition, the XT95 railing allowed the entire chamber to be mounted rigidly at A2 in any direction using a number of attachments available through Thorlabs (XT95P2, XT95P1, etc.). A digital photo of the chamber is shown in Fig. 2.5.

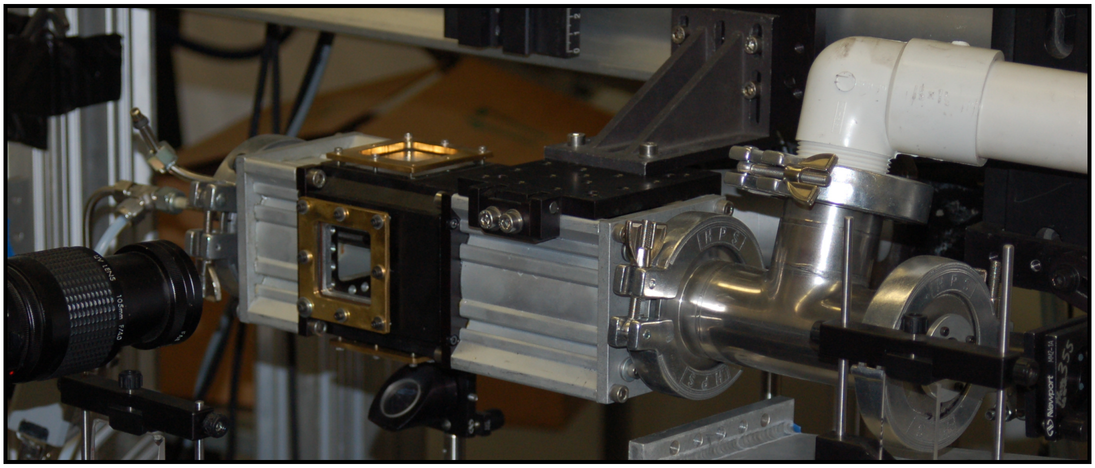


Fig. 2.5 Portable vacuum chamber (digital photo)

B.2 Subsonic Test Section

The subsonic test section was designed in Solidworks 2008. The following considerations were taken into account: 1) the area around the plasma section must be made of non-metallic materials; 2) optical access was required on 4 sides with flush mounted O-ring sealed fused silica windows (so as not to disturb the flow); 3) the test section must be able to be completely sealed for leak-testing; 4) the windows must be interchangeable incase of accidental breakage; 5) the entire top and bottom walls (including the area surrounding the electrodes) must be angled for a slight expansion down the length of the test section (~0.18°) to account for boundary layer growth (room for adjustment was preferable); and 6) the entire structure must be structurally rigid. The Solidworks drawing of the entire test section is shown on the following page. In this design, all of the pieces were machined of aluminum except those in the vicinity of the plasma discharge region. The Solidworks design of the test section is shown in Fig. 2.6. A closeup of the plasma discharge section is shown in Fig. 2.7. This region was especially important and has several additional special design requirements: 1) the region around where the plasma is discharged must be made of non-metallic, dielectric, high-temperature resistant materials to prevent arcing and/or melting; 2) optical access must be available as close to the copper electrodes as possible on all four sides for plasma diagnostics; 3) the electrode tilt (pivot) must be adjustable to obtain an even plasma discharge across the width of the test section; 4) the Macor section must also expand out to relieve the growth of the boundary layer (~0.18°); and 5) the region must be leak-proof.

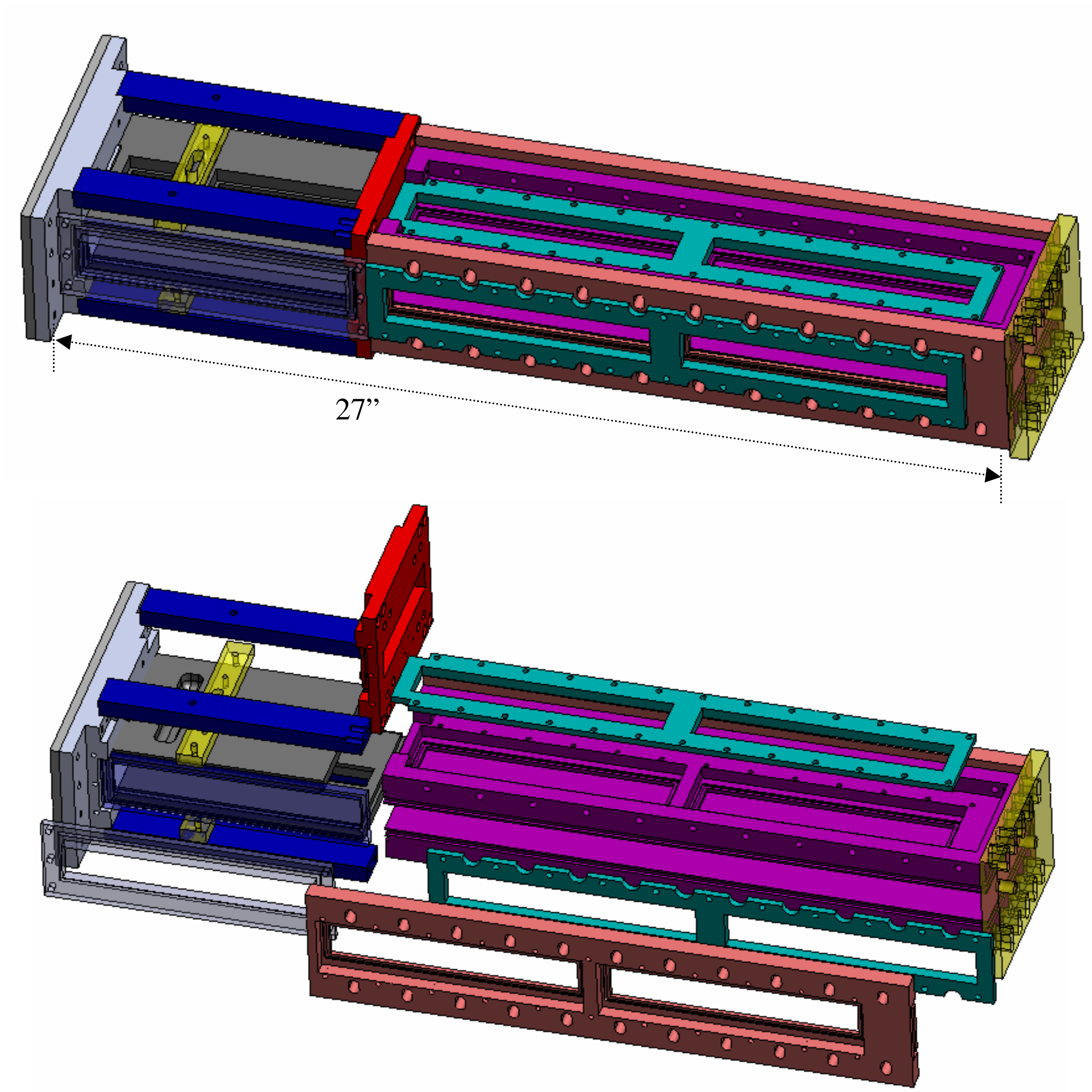


Fig. 2.6 Subsonic test section assembled and exploded view

The special parts of this section are labeled in Fig. 2.7 from A to H. The top and bottom (A) were made of solid ½” thick Macor machinable ceramic slabs. Macor has excellent dielectric properties, and can withstand extremely high temperatures, but it is comparable to fused silica in terms of brittleness. D is the placement of the copper electrode. The electrode was made of solid copper and was hand polished to remove any sharp edges. The bar protruding from the top of the electrode was a length of all-thread which had been soldered to the electrode and served as the connection to ground on the top Macor slab, i.e. “ground electrode”. The bottom electrode was mounted similarly and the all-thread connects to the RF source, i.e. “hot electrode”. The groove in which the electrode sits was filleted to approximately match the curvature of the electrode. C is a crossbar which was made of plastic (Delrin), and tapped through to allow nylon screws to contact and pivot the electrode from side to side to adjust the plasma discharge for uniformity. F and G are the front and back flanges of the plasma discharge section and were machined from aluminum. The aluminum structural bars (B) locked the entire section together, and were important for preventing the brittle Macor slabs from torque induced cracking. The side fused silica window (E) was sealed with O-ring grooves cut into the Macor and flanges F and G. The top and bottom windows (H) were sealed to the Macor with high-temperature putty (Cotronics, Duraseal). Therefore, in this design, no metallic or low-temperature materials were used in the vicinity of the electrodes and plasma discharge.

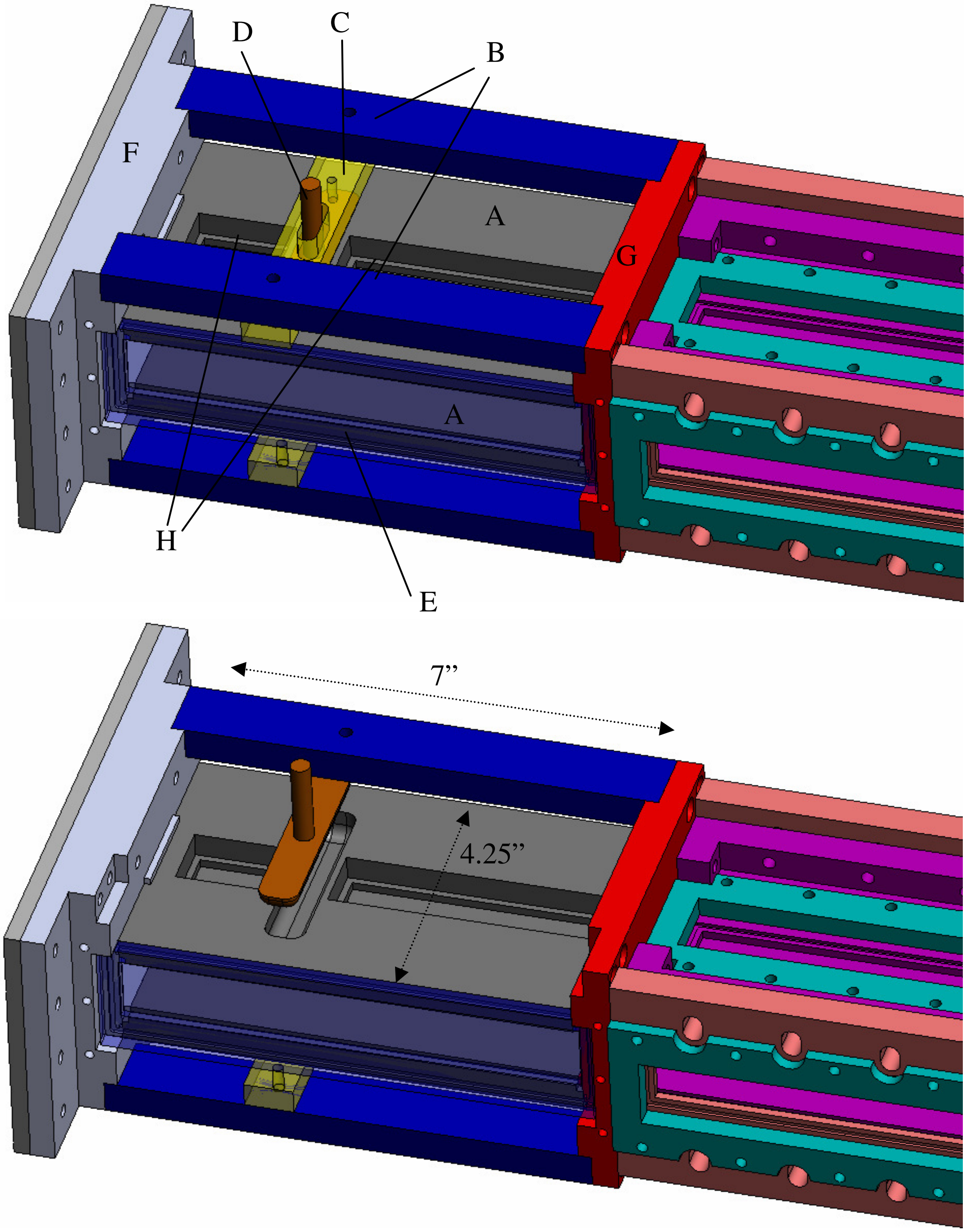


Fig. 2.7 Subsonic test section plasma discharge section assembled views

Fig. 2.8 shows a side view close-up of the plasma discharge vicinity with the window and window cover removed. The electrodes were machined with the same radius of curvature as the Macor seat for a close fit. Note that the electrode did not physically touch the flowfield, but was instead separated by approximately 1 mm of the dielectric Macor; this separation helped ensure a more uniform glow discharge, and also prevented secondary electron emission and ablation from the copper surface. The 3/8” windows had a 1/8” lip for increased structural strength, and were separated from the electrode by < 0.1” of Macor. The windows in this area were not sealed with o-rings for two reasons. First, o-rings would likely have melted being so close to the electrodes, and second, including o-rings would have increased the minimum distance between the electrode and clear viewing area, restricting optical access. This entire section diverged out at 0.18° matching the remainder of the test section; however, the angle was set by the openings in flanges F and G. Adjustment of the angle would require re-machining of the inner cavities of these flanges but could be done rather easily. To avoid machining the large flat surfaces of the Macor at this small angle (which would likely lead to flaking of the material), the o-ring grooves (marked by arrows in Fig. 2.8) were machined at the slight angle.

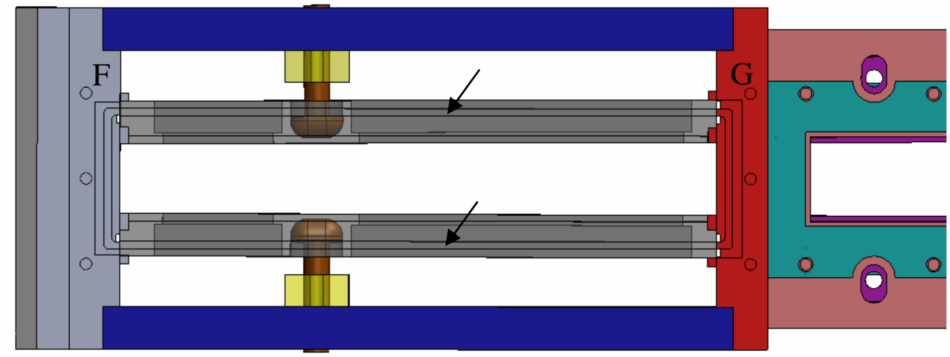


Fig. 2.8 Discharge section closeup

Stress analyses were performed on the Macor slabs, the fused silica windows, and the aluminum wall pieces. The aluminum properties used were the properties given by

Solidworks for the standard 6061 Alloy. The properties of Macor and of fused silica are shown in Table 2.1 , obtained from manufacturer's specifications:

Table 2.1 Properties of Macor and UV Fused Silica

	Macor	Fused Silica	6061 Aluminum Alloy
Elastic Modulus	$9.7\text{x}10^6$ lb/in^2	$1.07\text{x}10^7$ lb/in^2	$1.00\text{x}10^7$ lb/in^2
Poisson's Ratio	0.29	0.17	0.33
Shear Modulus	$3.7\text{x}10^6$ lb/in^2	$4.5\text{x}10^6$ lb/in^2	$3.8\text{x}10^6$ lb/in^2
Thermal Expansion Coefficient	$7\text{x}10^{-6}$	$5.5\text{x}10^{-7}$	$2.4\text{x}10^{-5}$
Density	0.0909 lb/in^3	0.0799 lb/in^3	0.0975 lb/in^3
Thermal Conductivity	1.5 W/m*K	1.4 W/m*K	170 W/m*K
Specific Heat	790 J/kg*K	670 J/kg*K	1300 J/kg*K
Flexural Strength	13600 lb/in^2	7251.9 lb/in^2	-----
Fracture toughness	1.53 MPa*m$^{1/2}$	0.7-0.8 MPa*m$^{1/2}$ (glass, typical)	29.00 MPa*m$^{1/2}$ [www.matweb.com]

The stress analysis for the Macor slabs is shown in Fig. 2.9. The highest stress was shown to be in the region between the electrode seat and the long window. The thickness of the Macor wall in this region was <0.1", as mentioned before, to allow optical access close to the plasma. The maximum displacement in this region was calculated to be almost 10 um, occurring in the region of highest stress, and also along the edge of the long window. However, the lowest calculated factor of safety was 9.7, well above operating range.

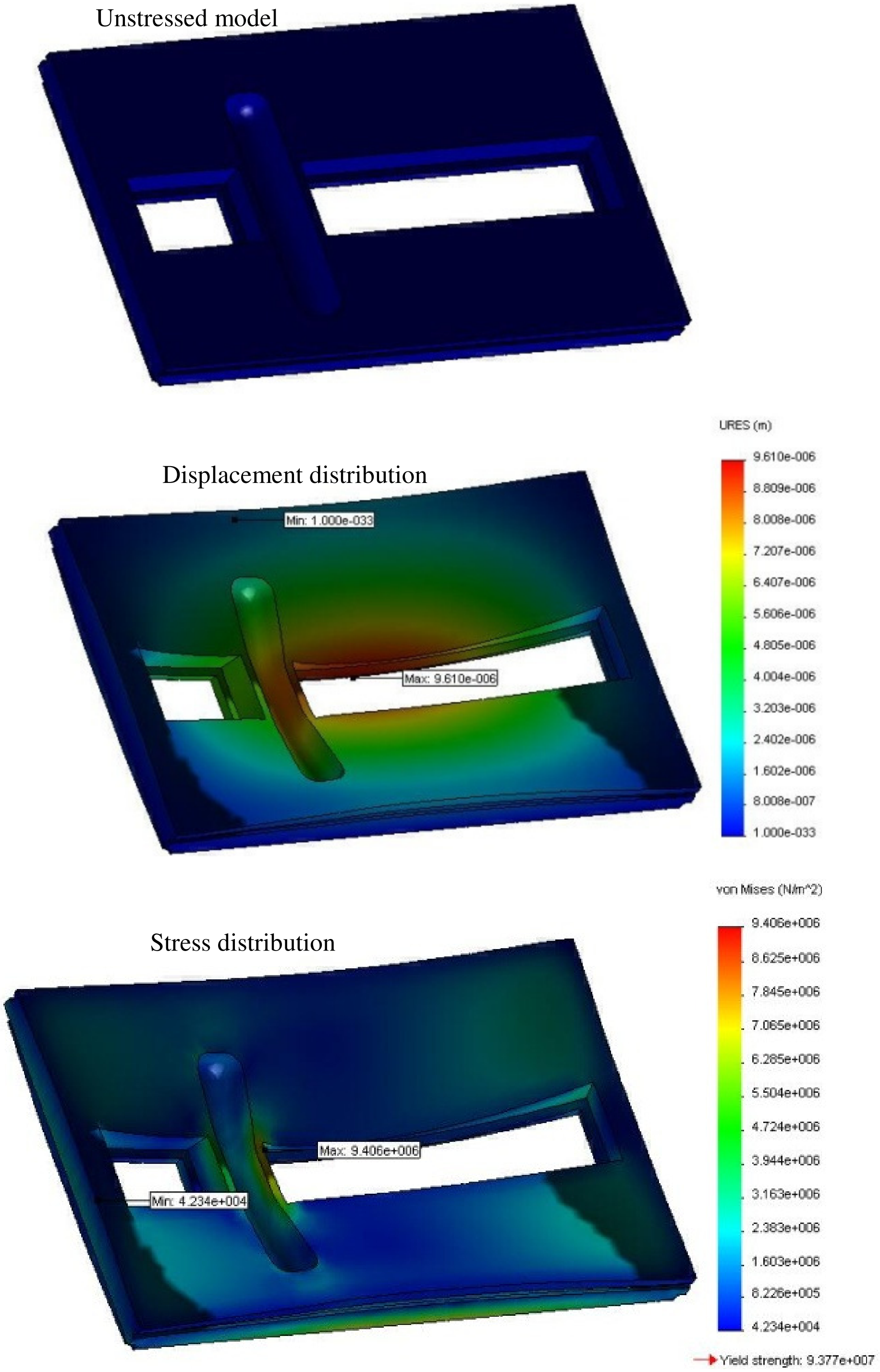


Fig. 2.9 Macor slabs stress analysis

The top wall was used to test the stress in the post-plasma region. The sidewalls had a similar design and would withstand similar stresses. The walls were 0.5" thick with the exception of the window settings. The results are shown in Fig. 2.10:

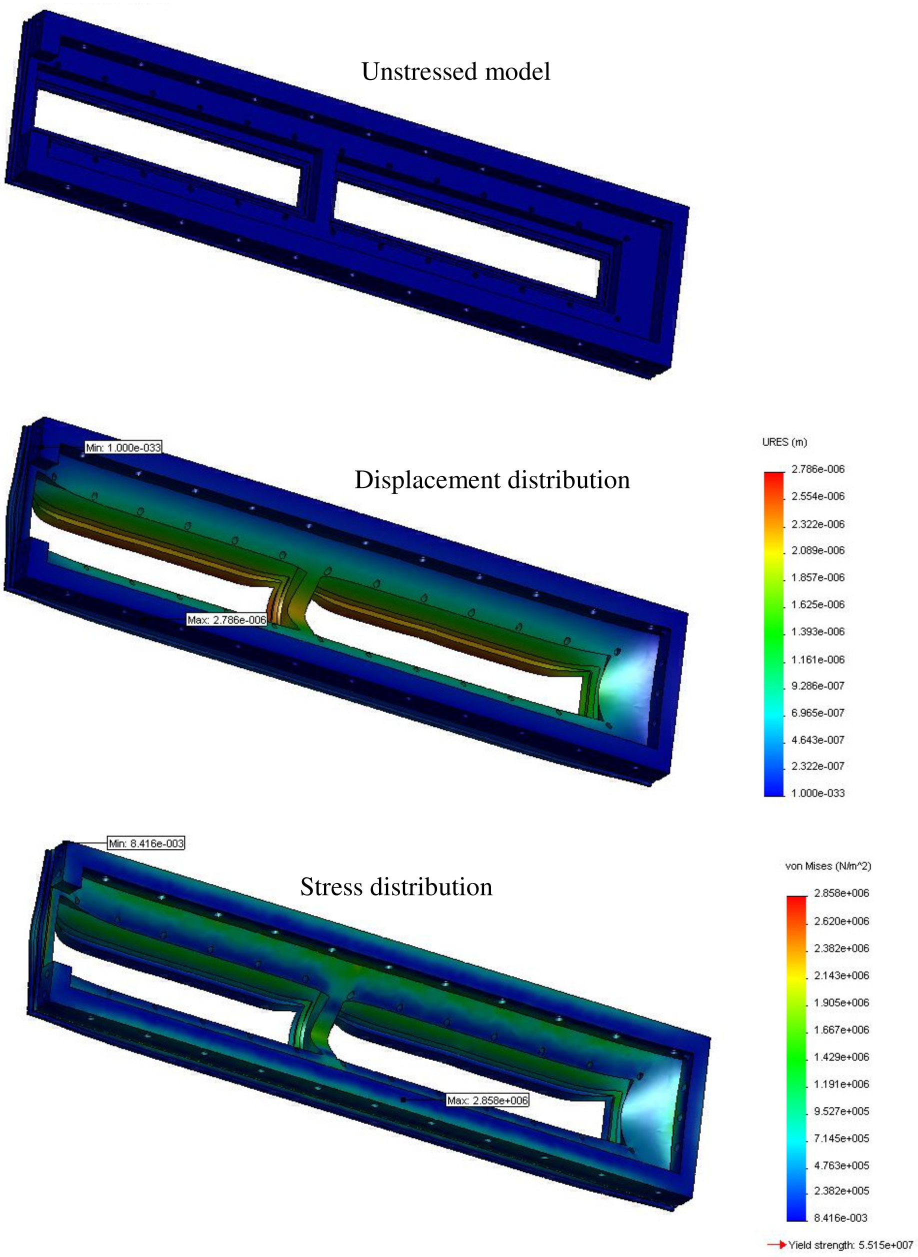


Fig. 2.10 Aluminum wall stress analysis

The weakest (i.e. highest stress) area of the aluminum walls was in the vicinity of the o-ring grooves which sealed the fused silica windows to the test section and the lateral division between the two windows. The thickness of the material in o-ring region was about 0.08”. The piece had a maximum displacement of only 3 um. The lowest factor of safety calculated was 19.3.

Lastly, the windows were tested. Because of their long, thin, and narrow shape, (7.5”x1.8”x3/8”), there was some concern that the windows would shatter when subjected to vacuum pressures. The windows had a 1/8” lip to allow them to lie flush against the test section walls and were held with gentle pressure by aluminum window covers padded with soft felt material. The stress analyses results are shown in Fig. 2.11:

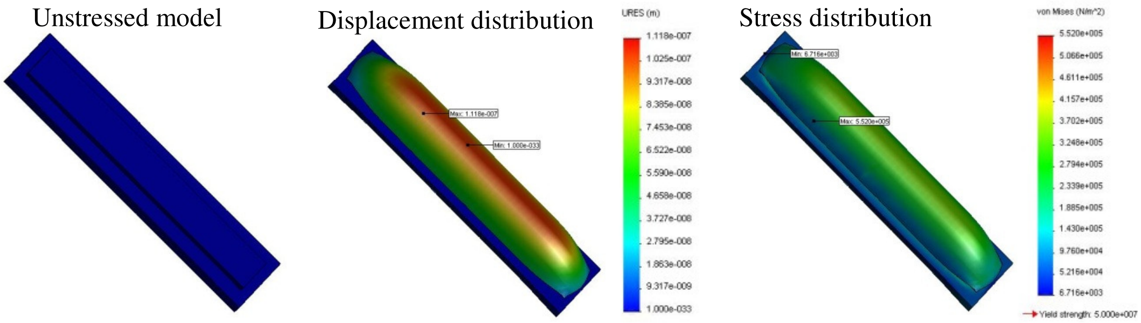


Fig. 2.11 UV fused silica window stress analysis

The windows were calculated to have a lowest factor of safety of 90.6 with minimal displacement (0.11 μm). The weakest point of the windows occurred in the center, as expected. Therefore, no weakening of the windows was expected. From the stress analysis results, it seemed that the area to be most concerned with was the Macor slabs. Although the results were promising in that no regions of the Macor showed factors of safety of <1, this material possesses a low fracture toughness (1.53 MPa*m$^{1/2}$) similar to that of glass (0.7-0.8 MPa*m$^{1/2}$) owing to its brittleness and tendencies to fracture rather than deform. Fracture toughness is the ability of a material to resist failure when a crack is already present. Compared to aluminum (29 MPa*m$^{1/2}$), both these materials are much more likely to fail in this situation. Because the Macor will be subjected to high temperatures from electrode heating, it is anticipated that small cracks may form over time from repeated thermal stresses which may cause the Macor to be more likely to fail.

Following design, the actual test section was machined, leak tested, and installed. Several photos of the test section (with opaque blanks in the windows for leak testing) are shown in Fig. 2.12, Fig. 2.13, Fig. 2.14, and Fig. 2.15.

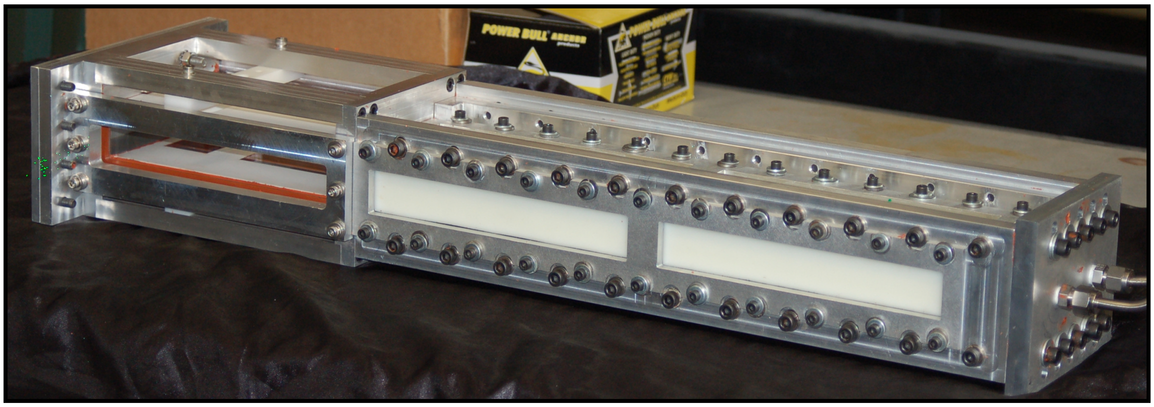


Fig. 2.12 Subsonic test section leak testing assembly (digital photo)

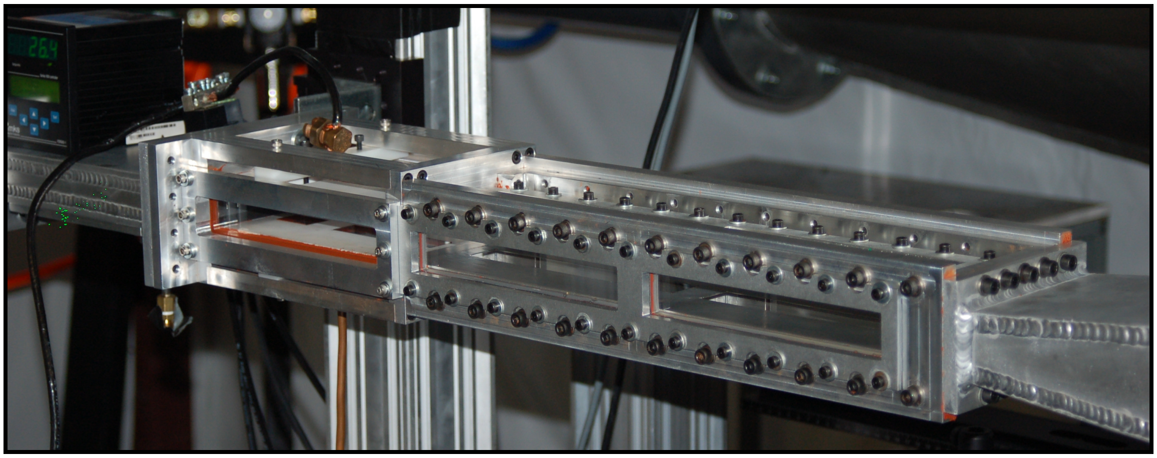


Fig. 2.13 Installed test section in subsonic facility (digital photo)

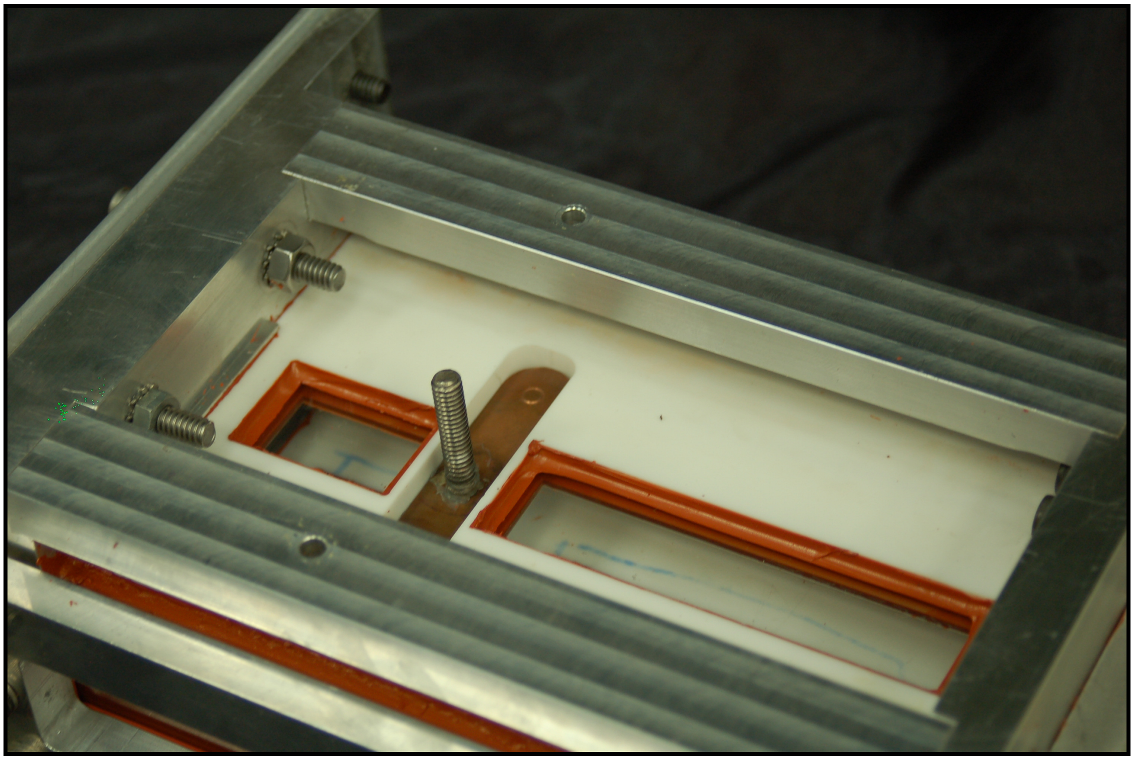


Fig. 2.14 Closeup of Macor plasma discharge section of subsonic test section (digital photo)

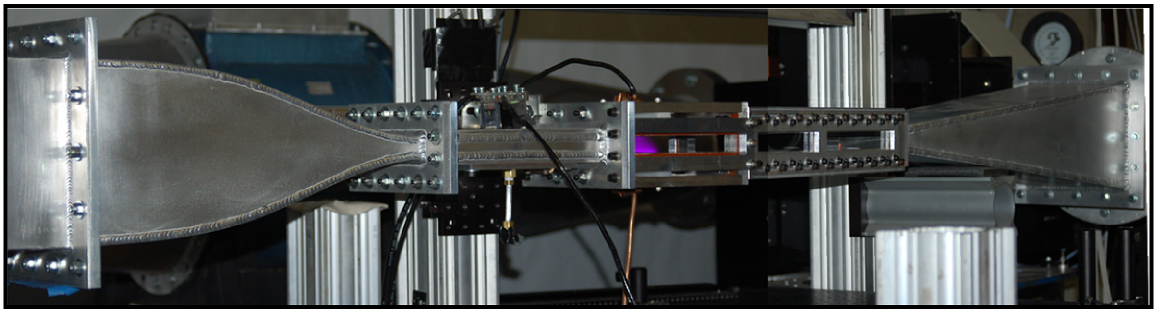


Fig. 2.15 Subsonic test section with nozzle and diffuser attached (digital photo)

C. Radio Frequency Plasma Hardware and Plasma Generation

The RF plasmas was powered by a water-cooled Dressler RF generator and matching network (Cesar Model 1325 200V RF Generator and Variomatch auto-matching network, 2-27 MHz, 700-1500W), and an in-house constructed impedance-matching coil. The generator provided RF power to the matching network through a heavily shielded coaxial cable. The matching network terminated in another heavily shielded coaxial cable which was modified with an alligator clip. The alligator clip was attached to the impedance-matching coil and manually tuned (relocating the clip to different coils) to achieve the lowest reflected power to forward power ratio. The impedance-matching oil consisted of a length of ¼" hollow copper tubing (~10'), coiled into a diameter of about 2". The plasma matching was highly sensitive to both the tuning of the alligator clip and to the spread of the coil. The end of the coil was attached to the hot electrode. Advice for setting up the RF discharge, including advice on construction of the impedance-matching coil was kindly provided by Dr. Roger Kimmel (Wright-Patterson AFB, OH) and Dr. Igor Adamovich (Ohio State University, OH). Photographs of the plasma discharge in 26.4 Torr air at 120 W forward power (12 W reflected power) are shown in Fig. 2.16 and Fig. 2.17.

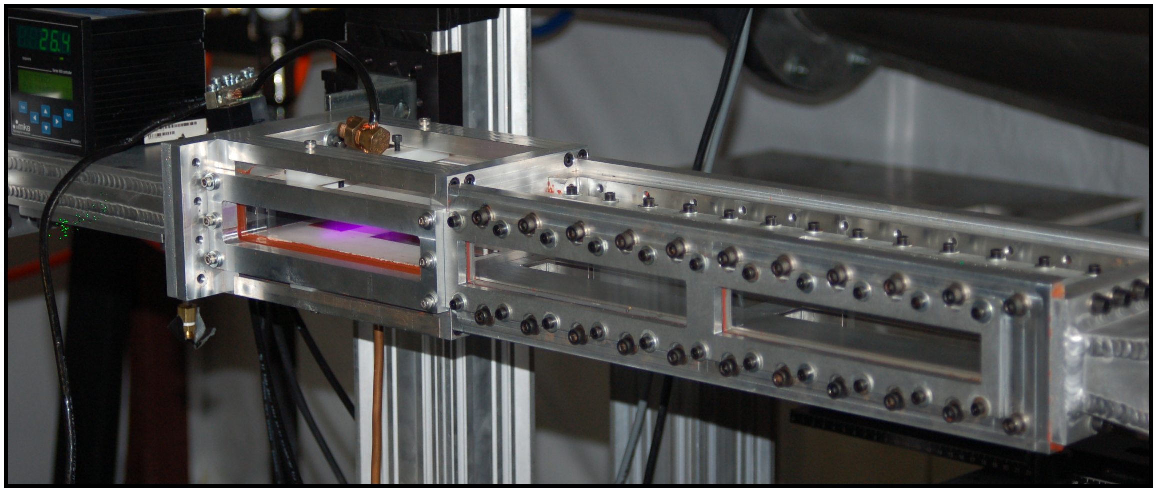


Fig. 2.16 Plasma discharge in full test section (digital photo)

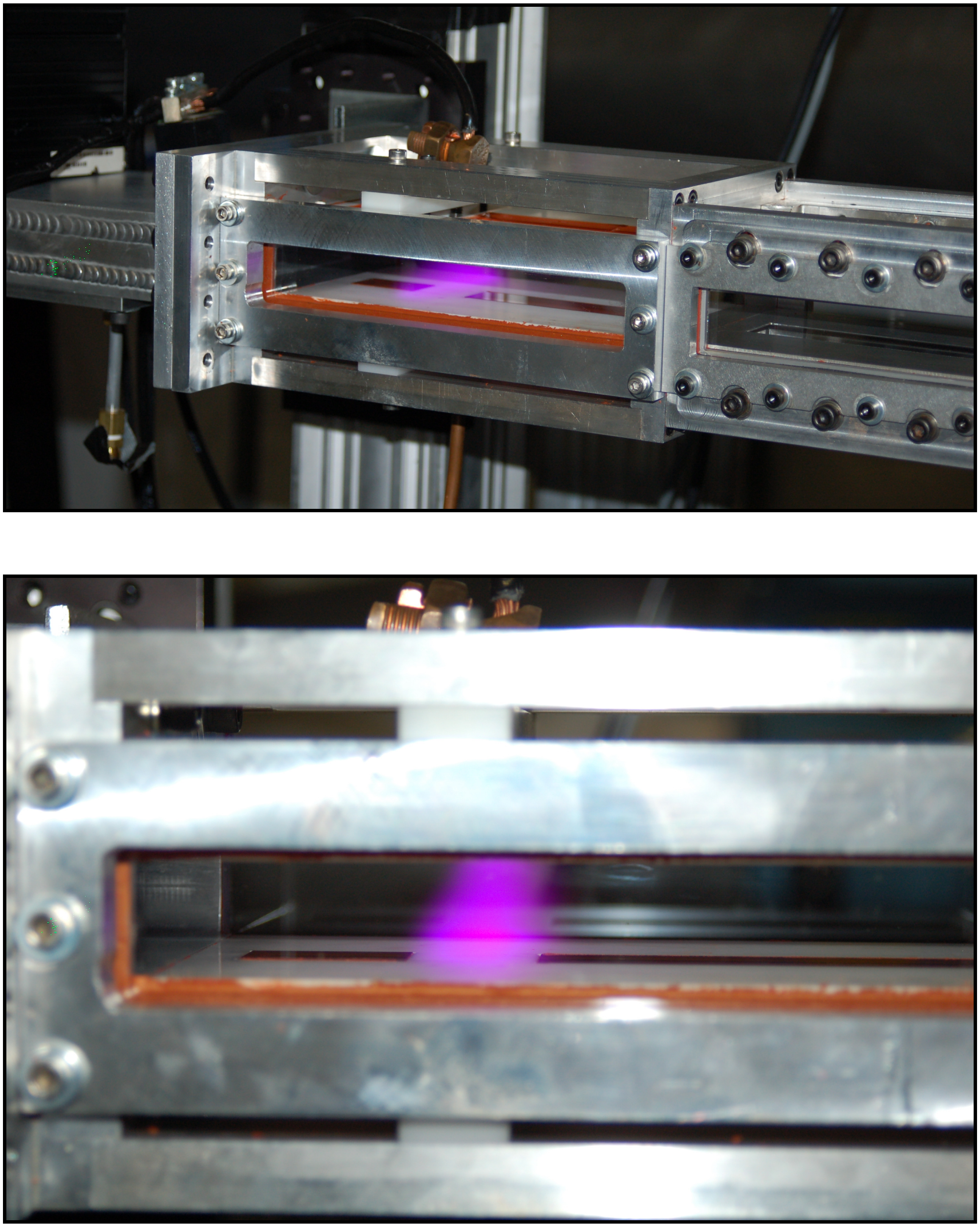


Fig. 2.17 Plasma discharge in full test section (digital photo, closeup)

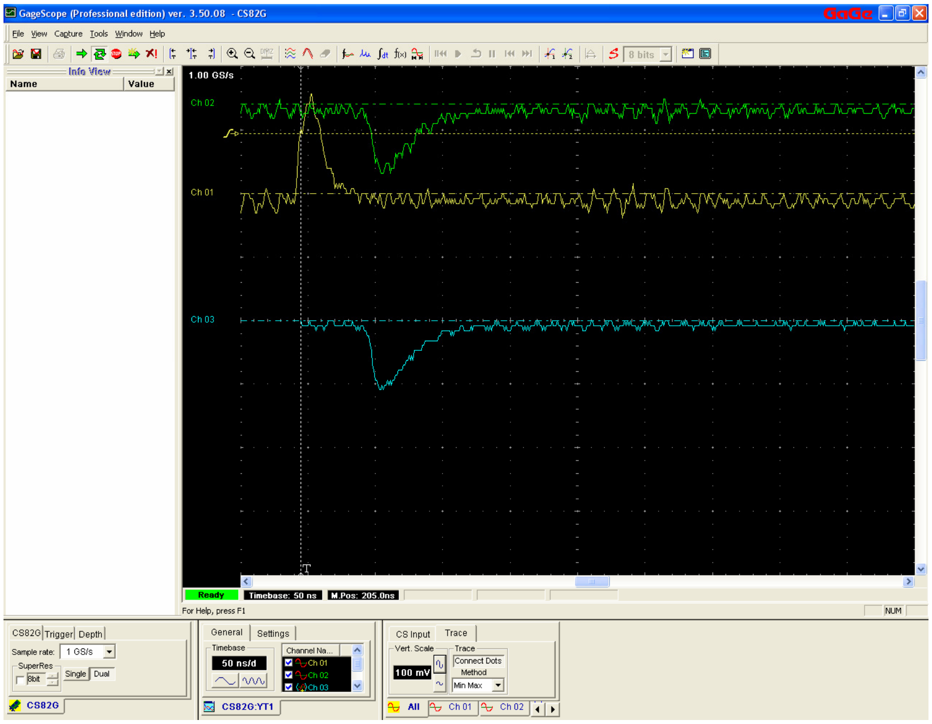
The grounding of the RF system was extremely important to minimize radiative RF noise. The generator, matching network, and ground electrode were grounded to a nearby rod driven into the concrete laboratory floor about three feet from the plasma discharge, while all other electronic equipment was routed to a separate grounding rod located roughly 20 feet away. However, even with the RF equipment grounded, there was still significant RF noise which spread to the lasers and ICCD cameras, and would cause spurious laser firing, drastic decrease (up to about 80%) in laser power, and noise on the camera CCD. It was determined that most of the noise originated from the digital delay generator used to control the overall timing of the experiments, and was used to externally trigger the lasers and cameras. Many attempts were made to eliminate the transmission of this noise, including frequency filters, extra grounding on the electronic equipment (including the use of flashing and braided wire to increase grounding surface area), Faraday cages made of solid copper sheeting surrounding the RF coil and matching network and/or around the digital delay generator, braided wire and/or foil shielding on the trigger cables, and dielectric barriers to physically isolate the cameras from their aluminum mounts. However, it was suspected that the plasma itself radiated a large portion of the RF noise which could not be completely enclosed. Furthermore, there was no way to fully shield the cameras from the radiated RF noise since they needed to be close (~1 foot) to the test section need to maintain optical transparency. The noise was seen on the CCD readout as a sinusoidal wave and increased the noise seen in images taken with the lens cap on by several fold. To alleviate the RF noise transmitted through the digital delay generator, the best solution found was to isolate the digital delay generator as far away from the discharge and RF generator as possible, and to lift the trigger cables off of the floor. This result was possibly due to the conducting properties of cement, although the cement floors were covered with an industrial plastic tile. Placing the digital delay generator in a metal containment such as a file cabinet also helped, although the degree to which it helped varied from day to day which only caused much graduate student frustration. Sadly, a permanent solution was never found.

CHAPTER III

FUNDAMENTAL STUDIES

A. LIF SCANNING AND SCANNING SOFTWARE

Custom software was written in LABVIEW for LIF excitation wavelength scanning. The program communicated with the Sirah dye laser via RS232 serial ports and differentiated between the two dye lasers via configuration files. The program began the LIF excitation scan at a user-selected wavelength and moved the dye laser resonator and BBO stepper motors to the calibrated position. Fluorescence was captured by a PMT fitted with a UG5 Schott glass filter. Wavelength scanning was typically conducted in the vacuum cell at slow-flow conditions using ~1% NO in air or N_2. An oscilloscope (PC Gagescope, CompuScope 82G) received the trace from the PMT and calculated the integrated fluorescence signal (subtracting out background) using a user-specified temporal window. The oscilloscope was triggered using either the digital delay generator (if the lasers were triggered externally as well), or using a photodiode placed near the output of the Nd:YAG laser (if the lasers were triggered internally). The user specified the number of traces to collect and average before moving onto the next laser wavelength; averaging more traces gave a smoother LIF scan. The process then repeated until the end wavelength was reached, generating a LIF excitation scan. An example printscreen of the Gagescope time trace is shown in Fig. 3.1. In the printscreen, Channel 1 (yellow trace) is the photodiode trigger signal near the output of the Nd:YAG laser to view passing 355 nm light. Channel 2 (green trace) is a single NO LIF time trace, and Channel 3 (light blue trace) is the averaged (10 shot) NO LIF time trace. An example printscreen of the LABVIEW program is shown in Fig. 3.2. The top left window labeled A captures each raw time trace. The window labeled B displays an optional input for photodiode signal to keep track of laser power throughout the scan. The window labeled C displays the LIF scan which is updated as the program moves through the desired wavelength range.



Sample rate: 1 GS/s

SuperRes: 8bit | Single | Dual

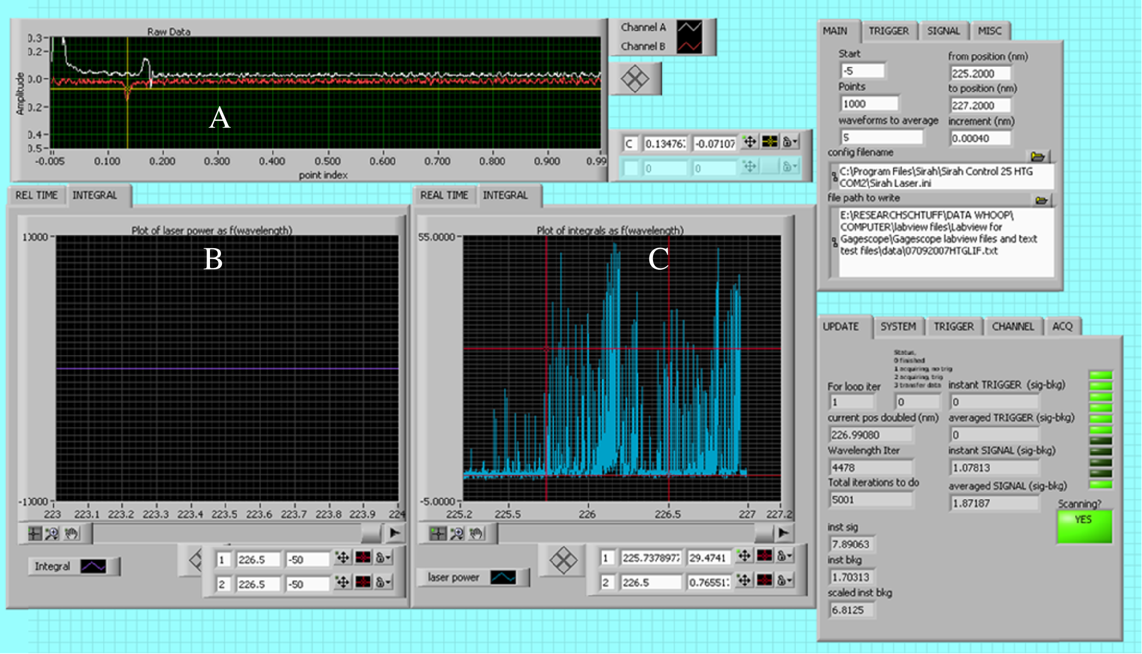
Timebase: 50 ns/d

Channel Na...: ☑ Ch 01 ☑ Ch 02 ☑ Ch 03

Vert. Scale: 100 mV

Trace: Connect Dots Method: Min Max

Fig. 3.1 Example NO LIF time trace from Gagescope



C 0.13476 -0.07107

1 226.5 -50

2 226.5 -50

1 225.7378977 29.4741

2 226.5 0.76551

Start: -5

Points: 1000

waveforms to average: 5

from position (nm): 225.2000

to position (nm): 227.2000

increment (nm): 0.00040

config filename: C:\Program Files\Sirah\Sirah Control 25 HTG COM2\Sirah Laser.ini

file path to write: E:\RESEARCHSCHTUFF\DATA WHOOP\COMPUTER\labview files\Labview for Gagescope\Gagescope labview files and text test files\data\07092007HTGLIF.txt

For loop iter: 1

Status, 0 finished, 1 acquiring, no trig, 2 acquiring, trig, 3 transfer data: 0

current pos doubled (nm): 226.99080

Wavelength Iter: 4478

Total iterations to do: 5001

instant TRIGGER (sig-bkg): 0

averaged TRIGGER (sig-bkg): 0

instant SIGNAL (sig-bkg): 1.07813

averaged SIGNAL (sig-bkg): 1.87187

Scanning?: YES

inst sig: 7.89063

inst bkg: 1.70313

scaled inst bkg: 6.8125

Fig. 3.2 Example printscreen of LABVIEW LIF scanning software

The scan was conducted at experimental pressures so that any pressure shifting was accounted for in calibrating for the wavelength shift of the dye laser relative to LIFbase. Typically, long scans which could last 1 to 2 hours were conducted every time the probe wavelength was changed significantly (> about 0.5 nm). Typically, 20-50 waveforms were averaged for each wavelength step. But on a day to day basis, the laser shifted perhaps only 0.005 nm, and the appropriate line was found simply by stepping one or two steps in either direction through the Sirah software. Verifying that the laser was at the maximum of the absorption line was usually done visually by observing the intensify of the imaged fluorescence via the ICCD cameras. Finally, the scans were imported into LIFbase (Luque, 1999) and shifted to match theoretical spectrum. The shift was noted and then used to calibrate the Sirah wavelength to the correct probe line. Example long experimental NO (1,1) and NO (0,0) excitation LIF scans are shown with the LIFbase simulated spectrum in Fig. 3.3 and Fig 3.4:

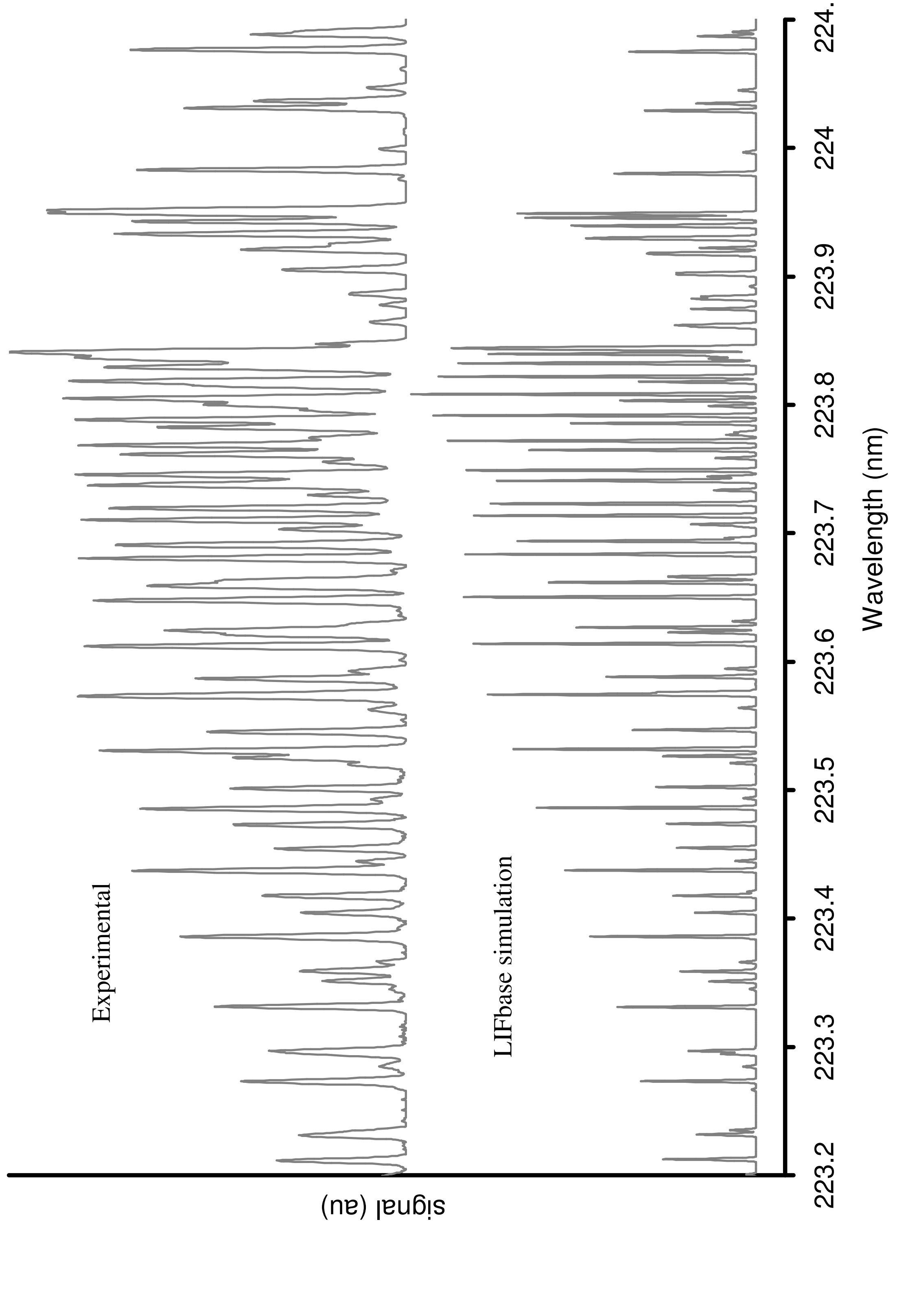


Fig. 3.3 NO(1,1) LIF excitation spectrum

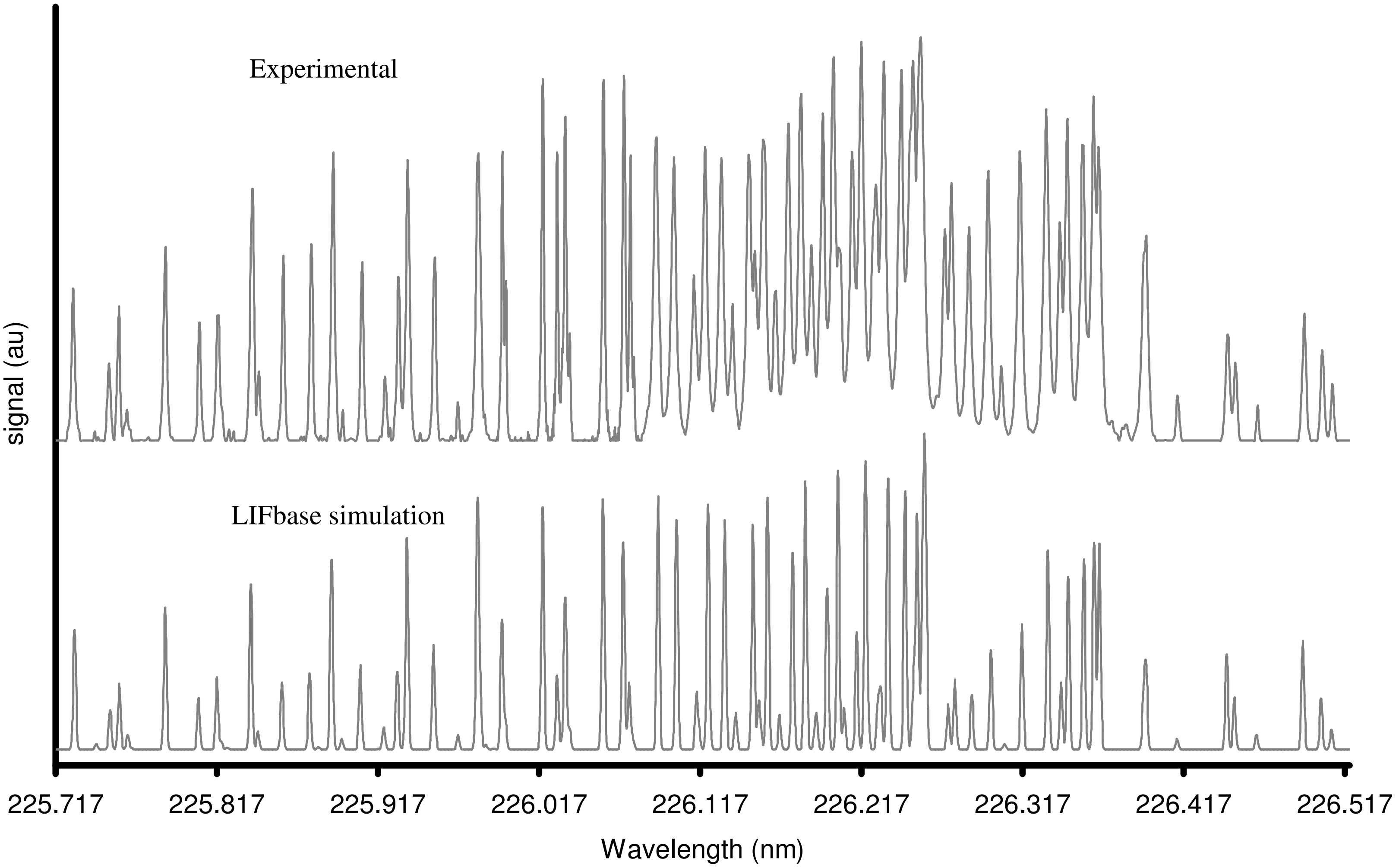


Fig. 3.4 NO(0,0) LIF excitation spectrum

As seen, the match is excellent. A threshold was applied to the experimental data to filter out base-line noise. Overall, the spectrum is clean, and the rotational lines can be located easily. One very distinct feature of the NO LIF excitation spectrum is the bandhead near 223.85 nm and 226.27 nm for NO(1,1) and NO(0,0), respectively, that make it easy to match up the experimental and simulated spectra. Shown in Fig. 3.5 is a simulated spectrum of the bandhead region of NO(0,0) run at 300 K and 100 Torr.

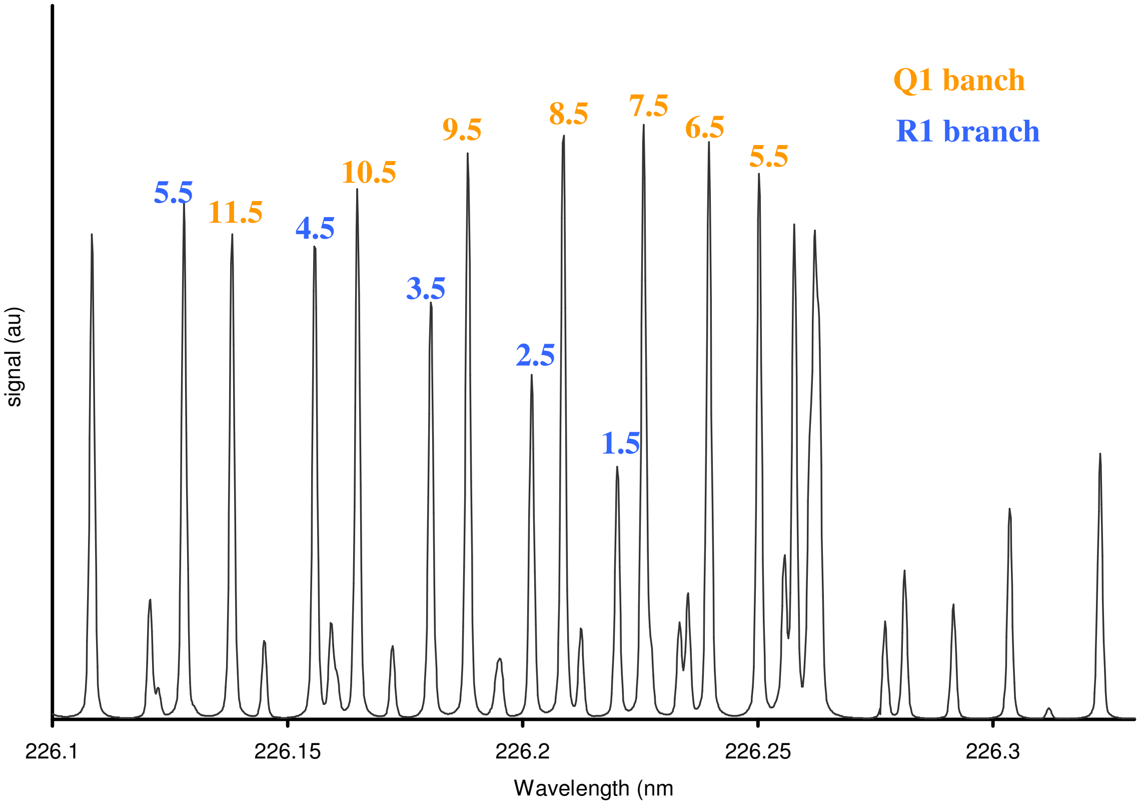


Fig. 3.5 Near-bandhead NO(0,0) showing distinctive alternating low and high J”

Another distinctive feature of the NO LIF scans is the alternating pattern of low and high rotational states near the bandhead. A few examples of these states are labeled in the figure above for NO(0,0). NO(1,1) shows a similar pattern. This pattern was especially useful in the underexpanded jet studies where in the expansion region, the temperatures drop to 20-30K, and the peak of the Boltzmann rotational distribution shifts to low rotational states (J”= 1.5, 2.5, 3.5). Probing these lines then gives then a rather uniform signal distribution across the flowfield. Whereas by stepping the laser a few steps, a higher rotational state is found (Ex. J”=6.5 to 10.5) and the cold expansion region becomes “dark”, clearly delineating the Mach disk. This made it convenient to use a high rotational line for focusing by using the Mach disk as a sharp edge, and then carrying out the experimental imaging using the neighboring low rotational line. This alternating pattern of well separated lines and the fact that they are are all in the vicinity of the bandhead (and therefore easy to locate) made these rotational lines especially favorable for probing.

B. Fluorescence Modeling Using a Multi-Level Model

The NO fluorescence signal can be modeled by considering a number of molecular states. The simplest method of modeling the fluorescence signal is via a 2 state system, where only one ground and one electronically excited state are considered. In this model, electronically excited NO decays through both fluorescence and quenching from the excited state to the ground state. However, a more thorough examination involves the extension of the 2-level model to a multi-level model. The 5-level model is commonly used, and comparisons between the 2-level and 5-level models have been discussed thoroughly in the literature [e.g. Lee(1993)]. 5-level models more accurately describe the behavior of NO fluorescence for several reasons. First, as iterated in Lee (1993), the electronically excited NO does not decay mainly to the pumped vibrational level in the ground state, but instead, decayes to mainly alternate vibrational states other than the pumped state. These must be considered as separate energy levels since vibrational energy transfer is slow compared to the laser pulse, and so the alternate vibrational states may be considered uncoupled to the pumped state within the fluorescence timescale. Second, rotational energy transfer (RET) occurs in the excited electronic state to a bath state, and NO may also fluoresce or quench from this state to the alternate vibrational state. In this model, all other vibrational states other than the pumped state are lumped into a single state. In the ground state, RET also occurs and all other rotational states are lumped into a bath state. A diagram of the 5-level model is shown in Fig. 3.6. The notation is the same as was described in Chapter I.

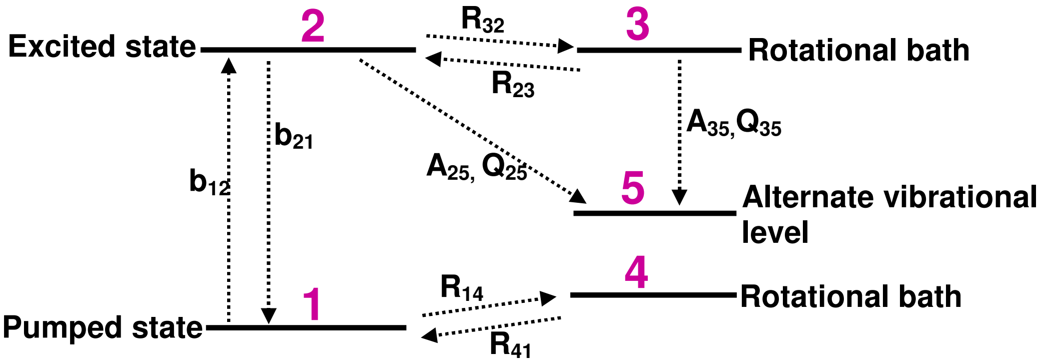


Fig. 3.6 NO fluorescence energy diagram (5-level model)

The time-dependent behavior of the fluorescence was modeled using Kintecus (Ianni, 1998). In the simulations, the laser power was altered by varying the rate constant, $b_{12} = B_{12}\frac{I}{c}$. The population is assumed to initially consist of only states 1 and 4, where the population of 1 is calculated via the Boltzmann distribution. The mixture is assumed to be 30 Torr of 1% NO in dry air at 300K. The laser pulse was assumed to be 20 ns in length. After the laser pulse, the rate constant term b_{12} is switched off, and the fluorescence signal decays. One sample simulation at rotational state J”=7 is shown in Fig. 3.7 (top) for 3 mJ/pulse laser power. The total fluorescence signal is the integral (area under the curve). This process was repeated at different laser intensities to obtain a plot of total fluorescence signal as a function of laser intensity, shown in Fig. 3.7 (bottom) for the 5-level and 2-level models.

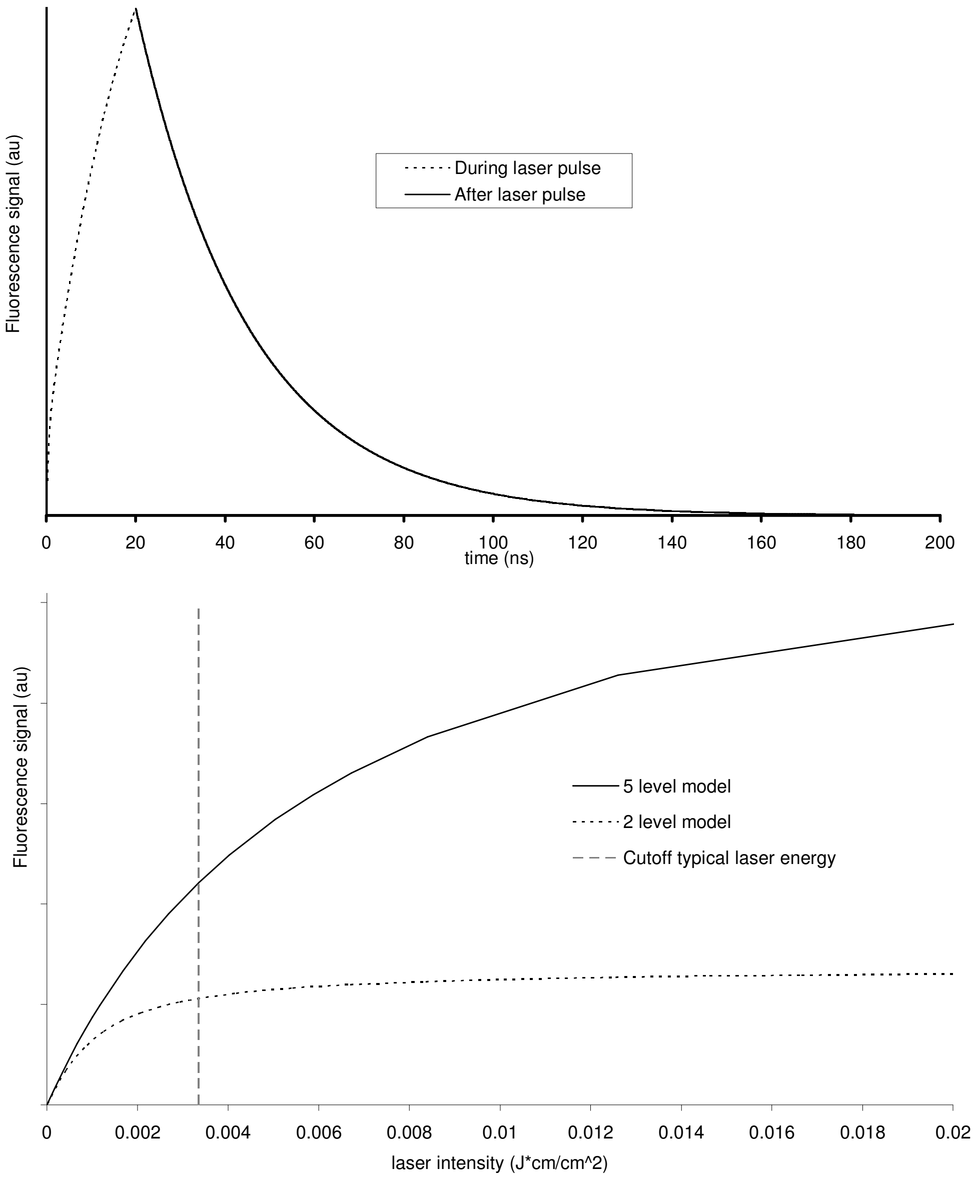


Fig. 3.7 NO fluorescence time-dependent 5-level modeling at J"=7 and 3 mJ/pulse laser power (top) and comparison of 2-level and 5-level modeling saturation behavior (bottom)

The plot on the left shows that the population of the electronically excited pumped state increases during the laser pulse and thus fluorescence increases. After the laser pulse, the fluorescence signal decays with a lifetime of 44 ns. The plot on the right shows the expected effect; that is, use of the 5-level model as opposed to the 2-level delays the onset of saturation. This is due to the quicker depletion of the excited electronic state through the increased number of sinks. The dotted line represents the typical amount of doubled laser energy (~ 2 mJ/pulse at 226 nm formed into a 1" x 300 um sheet) produced by the Sirah dye lasers. It is evident then, that with the available laser energy, it would not be possible to operate within the saturated fluorescence regime. Even with twice the power, the fluorescence signal would still be partially saturated according to the 5-level fluorescence model. Furthermore, even with enough overall power for saturated measurements, spatial variations in the laser beam intensity would prevent a typical PLIF image from operating in the saturated regime throughout the entire image. Therefore for the temperature mapping PLIF experiments, the power of the dye lasers must be maintained in the linear regime.

C. Saturation and Fluorescence Linearity

As discussed in the introduction, using the ratio of two fluorescence signal to yield a rotational temperature measurements is only valid in the linear or fully saturated fluorescence regimes. However, fully saturated measurements are difficult due to the required the laser intensity. For valid linear regime measurements, both the fluorescence signal and the camera signal corresponding to the fluorescence signal must vary linearly with laser power.

For verifying that the fluorescence signal was linear with laser power, an excitation scan was performed to locate the R1(3.5) line. A collimated laser sheet was formed (about 1" wide) and the resulting fluorescence was detected via a photomultiplier tube (PMT). The PMT voltage was set at 1000V. Glass slides and an iris were placed over the detector to lower and maintain linear output voltage signal. Scattered signal was virtually non-existent. Ten to fifteen clean 1" quartz plates were placed in the laser path (added one at a time) to systematically reduce the dye laser power in a uniform fashion between each PMT reading. A flow of 0.008 volume fraction NO in room air was fed into a calibration cell (with optical access on four sides) at 14 standard cubic centimeters per minute (sccm) and was maintained at 25 Torr, close to the experimental pressure and seeding fraction. The total integrated fluorescence signal (mV*ns) and dye laser power were then recorded. Relative uncertainties in the measurement of dye laser power and fluorescence signal were estimated at 0.02 mJ/pulse and 0.03 mV*ns, respectively. **Fig. 3.8** shows the raw fluorescence data collected as a function of laser intensity.

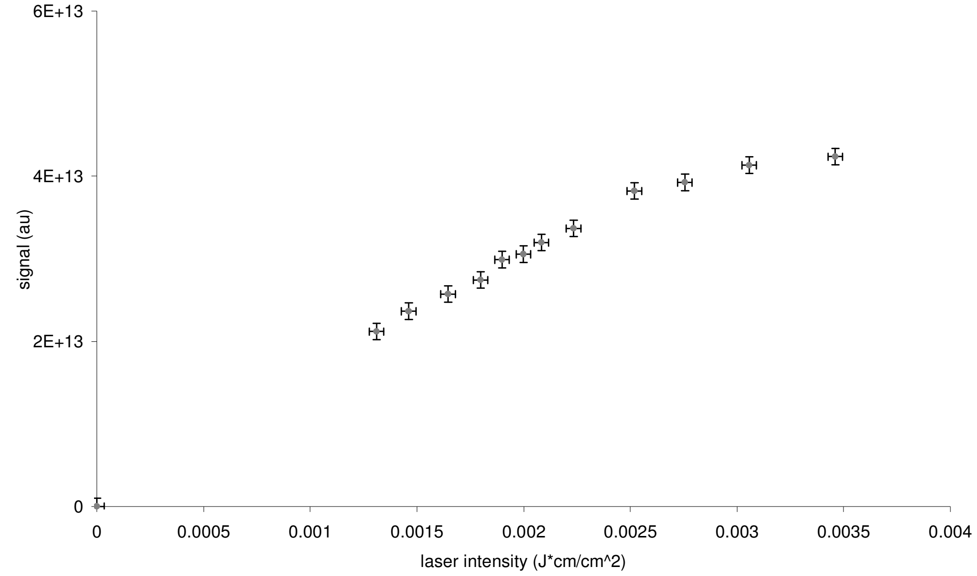


Fig. 3.8 PMT fluorescence vs. laser intensity

For the second measurement, full resolution ICCD images were taken as the laser power was varied. The camera was integrated on-chip for 20 seconds to minimize noise, and the gain was set at 50 (of 255). A selected bright area within the fluorescence image was examined, labeled in the sample image (Fig. 3.9). This area would represent a worse-case scenario, since it is where the laser intensity is maximum and thus would be most subject to saturation effects.

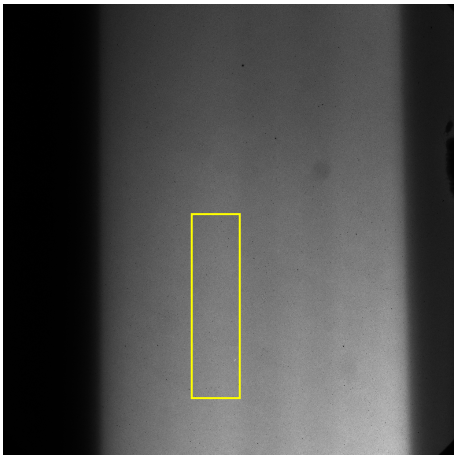


Fig. 3.9 Typical ICCD fluorescence image with examination area

The results for both the PMT and ICCD measurements and comparison with the 5-level and 2-level fluorescence models are shown in Fig. 3.10. The dotted line represents laser power of 0.4 mJ/pulse at the test section (about 80% losses due to optics between the laser output and test section with a direct laser output of about 2 mJ/pulse). At this laser power, the signal experienced about 20% saturation effects. The modeling results assume an overlap integral of 0.63 (using a calculated dye laser output full-width half max (FWHM) of 0.21 cm^{-1}, and absorption lineshape FWHM of 0.097 cm^{-1}, taking into account collisional and Doppler widths). Therefore, it was recommended that fluorescence measurements for temperature mapping be taken at laser powers of less than 0.4 mJ/pulse at the test section. The figure also shows that the fluorescence results best match the 5-level model, rather than the 2-level model, indicating that rotational energy transfer is important in modeling NO fluorescence. Modeling the fluorescence with a 2-level model reduces the threshold of the onset of saturation, leading to an overly conservative estimate of onset of saturation. The results from the examined area of the laser sheet showed that the assumption of linear florescence is valid over the entire image.

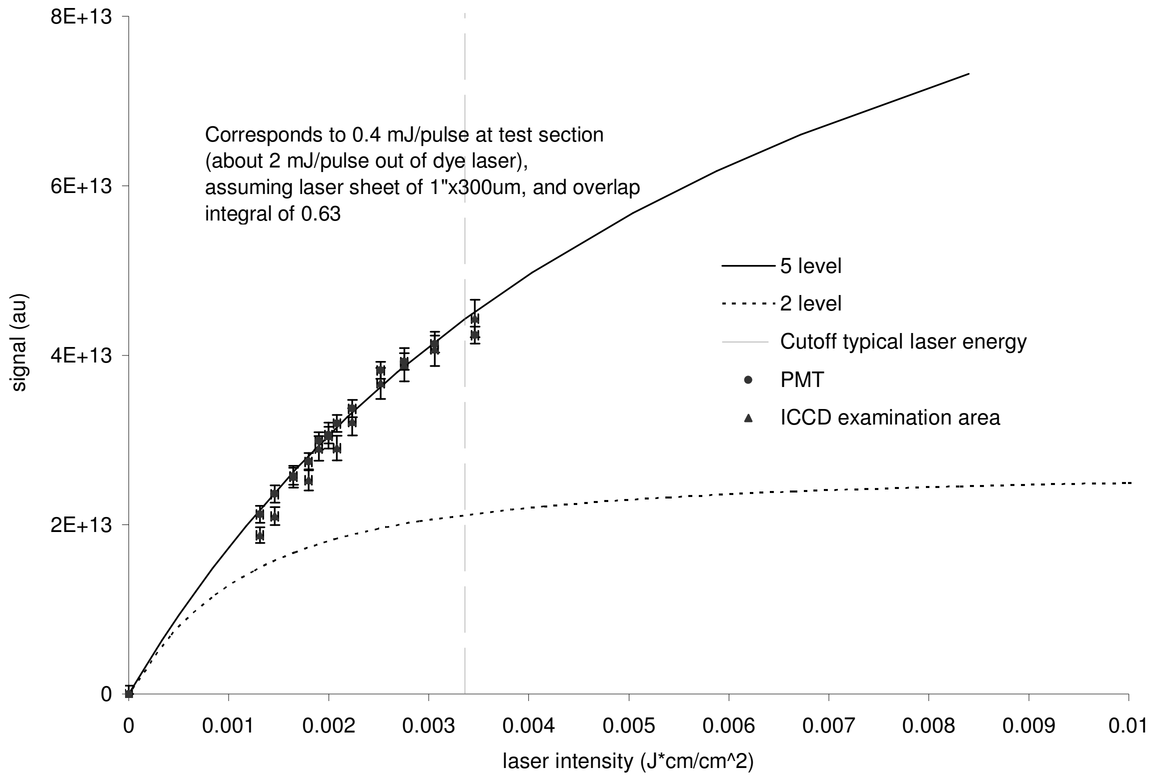


Fig. 3.10 PMT Fluorescence linearity comparison with modeling

D. Vibrational Kinetics Modeling

To explore the exchange and decay of vibrational energy in the flow species after the plasma discharge, an appropriate vibrational kinetics model was sought. The species in the flow considered were: N_2, O_2, H_2O, CO_2, NO, and O. For N_2, O_2, and NO, only the ground state and first excited vibrational states were considered. The vibrational energy spacings for N_2, O_2, and NO are taken as 2358.6, 1580.2, and 1904.2 cm^{-1}, respectively [Herzberg (1939)]. Vibrational temperatures were assumed to follow Boltzmann statistics. For H_2O, only the bending mode (ν2), with an energy of 1595 cm^{-1}, was considered since it was close in energy to the vibrational spacings of other species (especially O_2), and because other states of H_2O were far higher in energy (symmetric stretch, ν1=3657 cm^{-1}; asymmetric stretch, ν3=3755 cm^{-1}) and unlikely to exchange energy with other flow species. The denotation for this state of H_2O is $H_2O_{(010)}$. For CO_2, the lowest energetic bending mode (010) at 667 cm^{-1} and the asymmetrical stretch (2349 cm-1) were considered because the higher state is close in energy to the N_2 vibrational energy spacing, and rate constants for the exchange of the lowest bending mode (010) are known. Other states of CO_2 (symmetric stretch mode (100) at 1388 cm^{-1}, bending modes (020) and 030) at 1286 and 1933 cm^{-1}, respectively) were not considered mainly due to lack of literature rate constants. For some energy exchanges, other state of CO_2 were already accounted for and lumped into the total rate constant (such as reaction 46). Rate constants for all energy exchanges were collected from the literature for such energy exchanges, and reverse rate constants were calculated via detailed balance. The model amounted to 104 reactions (52 forward and reverse pairs) of which 9 rate constants were unaccounted for in the literature. Many of the reactions had multiple sources, so the most valid was chosen and is listed here. Experimental data were used where available. The vibrational decay model is shown in Table 3.1, Table 3.2, and Table 3.3 with associated references.

Table 3.1 Vibrational Translational (VT) energy exchanges considered for vibrational kinetics modeling

VT reaction	Forward rate constant	Ref.	Reverse rate constant	Ref.	Rxn #
$N_{2,v=1}$ + N_2 ==> N_2 + N_2	4.13262 x 10^{-24}	Candler (2002)	5.2152 x 10^{-29}	DB	1
$N_{2,v=1}$ + O_2 ==> N_2 + O_2	4.13262 x 10^{-24}	Candler (2002)	5.2152 x 10^{-29}	DB	2
$N_{2,v=1}$ + NO ==> N_2 + NO	4.13262 x 10^{-24}	Candler (2002)	5.2152 x 10^{-29}	DB	3
$N_{2,v=1}$ + CO_2 ==> N_2 + CO_2	5.2877 x 10^{-15}	Bass (1980)	6.67286 x 10^{-20}	DB	4
$N_{2,v=1}$ + H_2O ==> N_2 + H_2O	5.2877 x 10^{-15}	Bass (1980)	6.67286 x 10^{-20}	DB	5
$N_{2,v=1}$ + O ==> N_2 + O	3.5 x 10^{-15}	McNeal (1972)	4.41685 x 10^{-20}	DB	6
$O_{2,v=1}$ + O_2 ==> O_2 + O_2	5.77103 x 10^{-19}	Bass (1980)	3.01357 x 10^{-22}	DB	7
$O_{2,v=1}$ + N_2 ==> O_2 + N_2	5.58402 x 10^{-19}	Bass (1980)	2.91592 x 10^{-22}	DB	8
$O_{2,v=1}$ + NO ==> O_2 + NO	6.54768 x 10^{-19}	Candler (2002)	3.41914 x 10^{-22}	DB	9
$O_{2,v=1}$ + CO_2 ==> O_2 + CO_2	1.76896 x 10^{-20}	Bass (1980)	9.23734 x 10^{-24}	DB	10
$O_{2,v=1}$ + H_2O ==> O_2 + H_2O	4.08765 x 10^{-15}	Bass (1980)	2.13453 x 10^{-18}	DB	11
$O_{2,v=1}$ + O ==> O_2 + O	8.3 x 10^{-12}	Breen (1973)	4.33418 x 10^{-15}	DB	12
$NO_{v=1}$ + NO ==> NO + NO	7.52 x 10^{-14}	Kosanetzky (1980)	8.33612 x 10^{-18}	DB	13
$NO_{v=1}$ + O_2 ==> NO + O_2	2.58 x 10^{-14}	Kosanetzky (1980)	2.85908 x 10^{-18}	DB	14
$NO_{v=1}$ + N_2 ==> NO + N_2	1.30 x 10^{-16}	Kosanetzky (1980)	1.44677 x 10^{-20}	DB	15
$NO_{v=1}$ + CO_2 ==> NO + CO_2	7.14522 x 10^{-16}	Tsang (1986)	7.92276 x 10^{-20}	DB	16
$NO_{v=1}$ + H_2O ==> NO + H_2O	1.8329 x 10^{-12}	Stephenson (1974)	2.03236 x 10^{-16}	DB	17
$NO_{v=1}$ + O ==> NO + O	2.4 x 10^{-11}	Dodd (1999)	2.64847 x 10^{-15}	DB	18
$H_2O_{(010)}$ + H_2O ==> H_2O + H_2O	5.50 x 10^{-11}	Finzi (1977)	2.67578 x 10^{-14}	DB	19
$H_2O_{(010)}$ + N_2 ==> H_2O + N_2	4.99009 x 10^{-14}	Bass (1980)	2.42771 x 10^{-17}	DB	20
$H_2O_{(010)}$ + O_2 ==> H_2O + O_2	4.99009 x 10^{-14}	Bass (1980)	2.42771 x 10^{-17}	DB	21
$H_2O_{(010)}$ + NO ==> H_2O + NO	5.92126 x 10^{-14}	Palmer (1993)	2.88073 x 10^{-17}	DB	22
$H_2O_{(010)}$ + CO_2 ==> H_2O + CO_2	no data		no data		23
$H_2O_{(010)}$ + O ==> H_2O + O	no data		no data		24
$CO_{2,(001)}$ + CO_2 ==> CO_2 + CO_2	6.81 x 10^{-15}	Huetz-Aubert (1971)	9.00134 x 10^{-20}	DB	25
$CO_{2,(010)}$ + CO_2 ==> CO_2 + CO_2	5.24 x 10^{-15}	Bass (1980)	2.15724 x 10^{-16}	DB	26
$CO_{2,(001)}$ + N_2 ==> CO_2 + N_2	1.85802 x 10^{-15}	Huetz-Aubert (1971)	2.45491 x 10^{-20}	DB	27
$CO_{2,(001)}$ + O_2 ==> CO_2 + O_2	3.19348 x 10^{-15}	Cannemeyer (1973)	4.21938 x 10^{-20}	DB	28
$CO_{2,(010)}$ + N_2 ==> CO_2 + N_2	3.16276 x 10^{-15}	Bass (1980)	1.30217 x 10^{-16}	DB	29
$CO_{2,(010)}$ + O_2 ==> CO_2 + O_2	4.20879 x 10^{-15}	Bass (1980)	1.73284 x 10^{-16}	DB	30
$CO_{2,(001)}$ + NO ==> CO_2 + NO	2.46976 x 10^{-14}	Bauer (1986)	3.38726 x 10^{-19}	DB	31
$CO_{2,(001)}$ + O ==> CO_2 + O	1.9465 x 10^{-13}	Cramp (1973)	2.57181 x 10^{-18}	DB	32
$CO_{2,(001)}$ + H_2O ==> CO_2 + H_2O	1.01586 x 10^{-12}	Bauer (1986)	1.39325 x 10^{-17}	DB	33
$CO_{2,(010)}$ + O ==> CO_2 + O	3.00 x 10^{-12}	Castle (2004)	1.23516 x 10^{-13}	DB	34
$CO_{2,(010)}$ + H_2O ==> CO_2 + H_2O	2.03 x 10^{-11}	Bass (1980)	8.35671 x 10^{-13}	DB	35

Note: DB refers to rate constants calculated via detailed balance, all rate constants are bimolecular rates

Table 3.2 Vibrational Vibrational (V-V) energy exchanges considered for vibrational kinetics modeling

VT reaction	Forward rate constant	Ref.	Reverse rate constant	Ref.	Rxn#
$N_{2,v=1} + O_2$ ==> $O_{2,v=1} + N_2$	7.58×10^{-18}	Bass (1980)	1.83067×10^{-19}	DB	36
$N_{2,v=1} + NO$ ==> $NO_{v=1} + N_2$	1.04×10^{-15}	Candler (2002)	1.18432×10^{-16}	DB	37
$NO_{v=1} + O_2$ ==> $O_{2,v=1} + NO$	1 x 10-14	Ree (1993)	2.1234×10^{-15}	DB	38
$N_2 + CO_{2,(001)}$ ==> $N_{2,v=1} + CO_2$	5.11×10^{-13}	Bass (1980)	5.35359×10^{-13}	DB	39
$N_2 + H_2O_{(010)}$ ==> $N_{2,v=1} + H_2O$	1.31×10^{-16}	Zuckerwar (1987)	5.04275×10^{-15}	DB	40
$O_{2,v=1} + CO_2$ ==> $O_2 + CO_{2,(010)}$	6.66×10^{-15}	Bass (1980)	8.4456×10^{-17}	DB	41
$O_{2,v=1} + H_2O$ ==> $O_2 + H_2O_{(010)}$	2×10^{-12}	Joly (1977)	2.1467×10^{-12}	DB	42
$NO_{v=1} + CO_2$ ==> $NO + O_{2,(010)}$	no data		no data		43
$NO_{v=1} + H_2O$ ==> $NO + H_2O_{(010)}$	1.1×10^{-12}	Palmer (1993)	2.50218×10^{-13}	DB	44
$CO_2 + H_2O_{(010)}$ ==> $CO_{2,(010)} + H_2O$	no data		no data		45
$N_{2,v=1} + CO_2$ ==> $N_2 + CO_{2,(010)}$	3.27×10^{-15}	Zuckerwar(1987)	1.00232×10^{-18}	DB	46

Note: DB refers to rate constants calculated via detailed balance, all rate constants are bimolecular rates

Table 3.3 Chemical reactions considered for vibrational kinetics modeling

Chemical Reaction	Forward Rate constant	Ref.	Reverse rate constant	Ref.	Rxn#
$NO_{v=1} + NO_2$ ==> $NO + NO_2$	2.0000×10^{-12b}	Dodd (1999)	2.2000×10^{-16b}	DB	47
$O_2 + O$ ==> O_3	5.9200×10^{-34c}	DeMore (1997)	4.3800×10^{-26b}	Heimerl (1979)	48
$NO + O$ ==> NO_2	9.9925×10^{-32c}	Atkinson (2004)	1.0000×10^{-23a}	(estimated)	49
$O + O$ ==> O_2	1.0500×10^{-33c}	Tsang (1986)	1.0000×10^{-94b}	Tsang (1986)	50
$O + NO_2$ ==> $NO_{v=1} + O_2$	9.5000×10^{-12b}	Dodd (1999)	no data		51
$2NO + O_2$ ==> $2NO_2$	2.0000×10^{-38c}	Atkinson (2004)	2.9400×10^{-31b}	Tsang (1991)	52

Note: [a] unimolecular rate constant, [b] bimolecular rate constant, [c] termolecular rate constant

D.1 Modeling the Vibrational Energy Decays of N_2, O_2, and NO With and Without the Effects of CO_2 and Humidity

The effect of humidity and CO_2 on the vibrational decay of N_2, O_2, and NO was examined for two main reasons: 1) Humidity, especially in Texas where these experiments were conducted, can be extremely high, and in the laboratory, the humidity was uncontrolled and has reached 40-50%, corresponding to about 1% mole fraction of H_2O in the ambient air. This air would be inside of the recirculating subsonic tunnel.

Also, ambient air contains a small fraction, 0.04% CO_2. Although these fractions may be small, they are significant because the vibrational relaxation of O_2, N_2, and NO with these molecules are rapid and their presence must be taken into account in the vibrational kinetics decay modeling. 2) In the hypersonic tunnel, pump oil used in the compression of the supply air may produce CO_2 at levels higher than ambient air and these higher levels of CO_2 may affect the vibrational decay of the flow. There may also be a small amount of water in the flow, although at a much smaller fraction than in the ambient air, since the compressed air flow will be heated. For these reasons, it is important to model the effects of CO_2 and H_2O on the vibrational decay kinetics of the different flow species. The vibrational model described in the previous section was used in this examination. Simulations were run at 30 Torr, with 1% mole fraction seeded NO into air. The main species (N_2, O_2, and NO) were assumed to decay from an initial vibrational temperature of 2000 K with populations in the v=0 and v=1 states following Boltzmann statistics. Initially vibrationally excited H_2O and CO_2 were not considered. In reality, the different species would not exit the plasma at the same vibrational temperature. In fact, calculations from electron cross sections at plasma conditions similar to those tested in the experiment suggested that we may expect initial temperatures for N_2, O_2, and NO may be closer to 1700K, 800K, and 2000K, respectively. The humidity and concentration of CO_2 in the air flow were varied during these simulations, and the results are presented in the following paragraphs. All of the simulations presented were run with Kintecus software written by Ianni (2008).

The effect of water on the vibrational decay rates of N_2, O_2, and NO is shown in Fig. 3.11, run with no CO_2 in all cases:

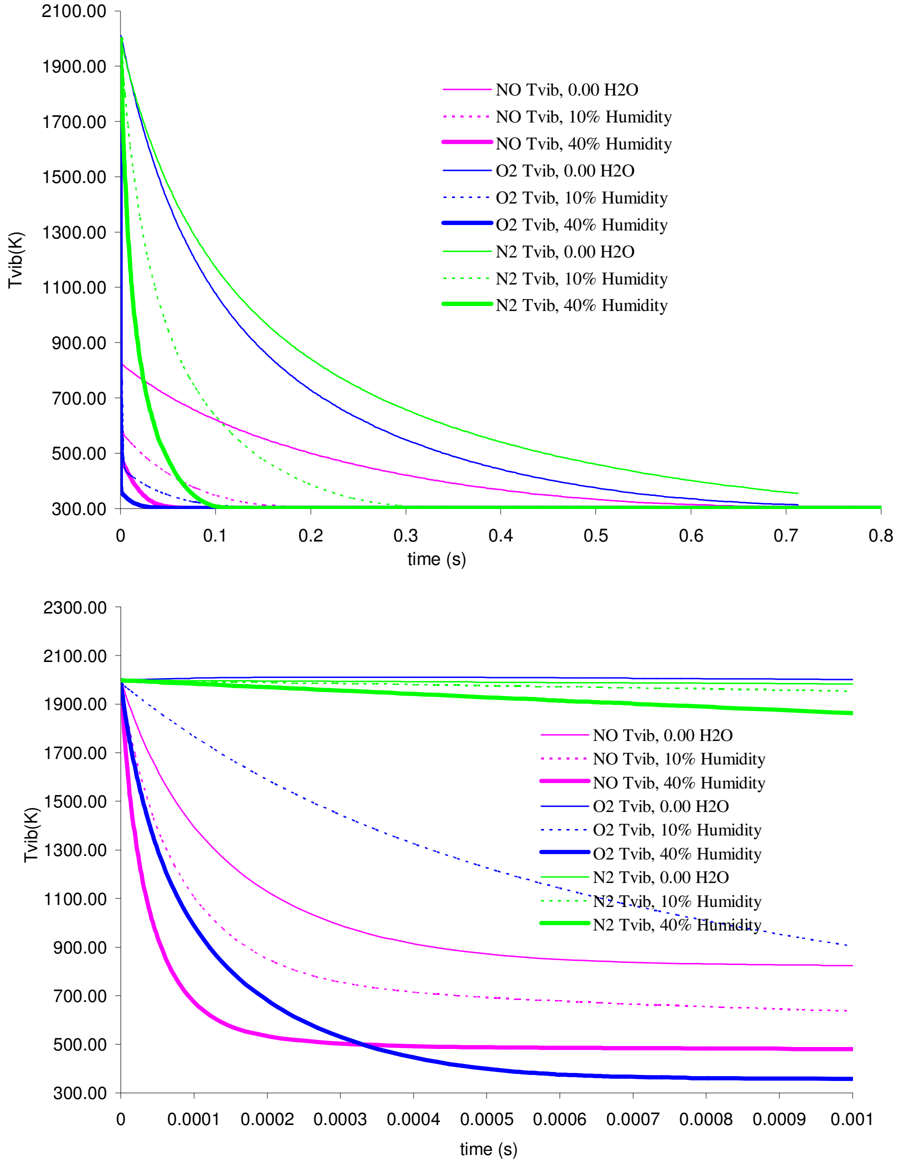


Fig. 3.11 Long-time and short-time effect of water on vibrational decay kinetics

The plot on the left verifies that in the long term, the vibrational temperatures of N_2, O_2, and NO reach 300K, as expected. In completely dry air with ambient levels of CO_2, O_2 and N_2 have similar decays, reaching 300K in about 0.5 to 0.7 s. NO has a much faster decay which is presented in greater detail in the plot on the right. In this plot, which shows the time decay of the different species to 1 ms, the vibrational temperatures of O_2 and N_2 in dry air appear frozen over the short timescale. The overall effect of humidity is to decrease the lifetimes of all three vibrationally excited species. Similarly, the plots for the effect of CO_2 are shown in Fig. 3.12, run at 0% humidity:

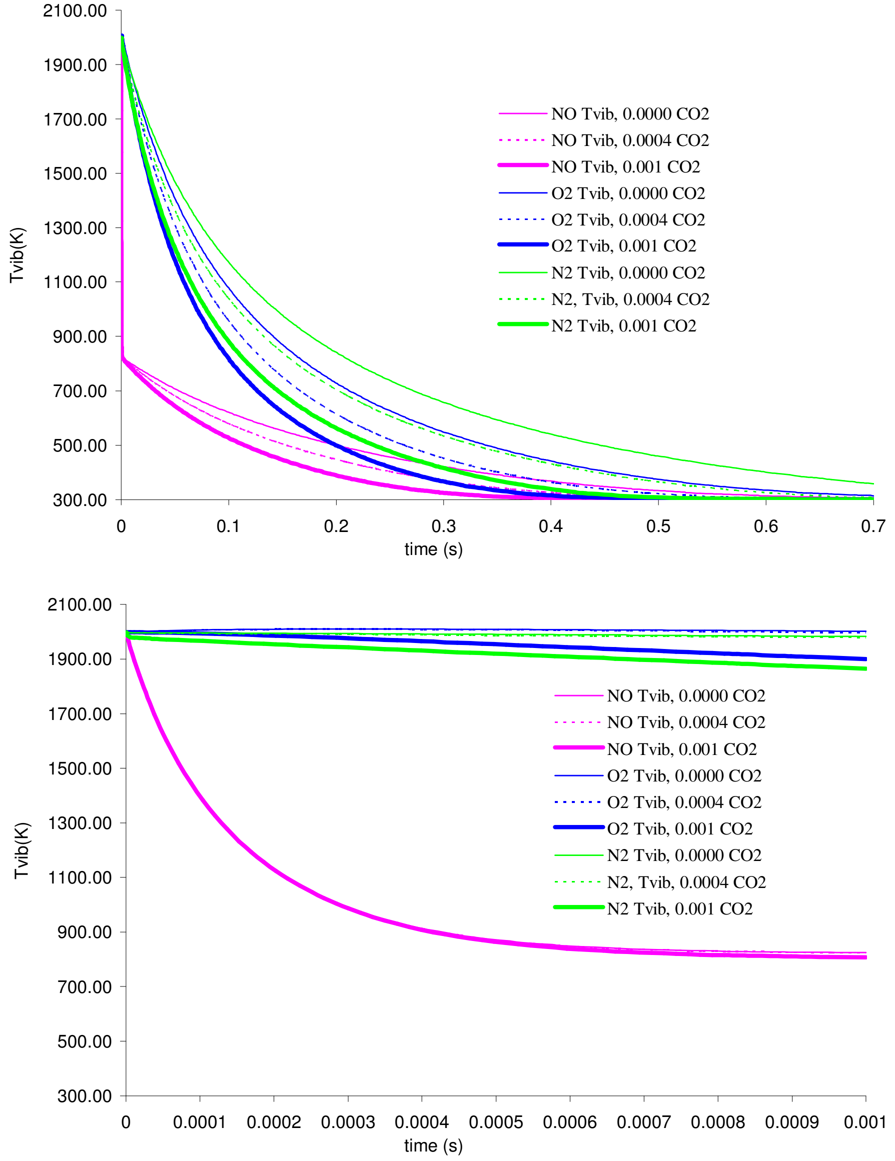


Fig. 3.12 Long-time and short-time effect of CO_2 on vibrational decay kinetics

Without H_2O and CO_2, the lifetimes of the vibrational temperature decay were 0.17 s, 0.13 s, and 350 μs for N_2, O_2, and NO, respectively. The very short lifetime of NO was mainly driven by the fast kinetics for V-V and V-T energy exchanges. For example, the V-T energy exchanges for NO with other species were 5 orders of magnitude faster than the V-T rates for O_2, and 10 orders of magnitude faster than those for N_2. Within the V-T rates for NO, the exchange was much faster for NO with O_2 (2.58 x 10^{-14} cm^3/molecule/s) than with N_2 (1.30 x 10^{-16} cm^3/molecule/s). NO V-V rates showed a similar trend, with energy exchanges with O_2 about 2 orders of magnitude faster for O_2 than N_2. The lifetimes of O_2, and N_2 were both long, with O_2 being slightly shorter (0.13 s) than N_2 (0.17 s). This trend was due a combined effect of faster V-T and V-V rates for O_2 as compared to N_2 with other species. V-T energy exchanges for O_2 varied slightly about 10 x 10^{-19} cm^3/molecule/s, while the same exchanges for N_2 were extremely slow and constant in value, with the main species of the flow (N_2, O_2, and NO). The V-V energy exchanges between O_2 and N_2 were close in magnitude and were shown by the rate constant for reaction #36. The forward rate of this reaction (7.58 x 10^{-18} cm^3/molecule/s was only slightly faster than its reverse counterpart (1.83067 x 10^{-19} cm^3/molecule/s), and was due to the vibrational energy spacing differences. The V-V exchange for O_2 with NO was about 4 orders of magnitude faster, so in general, O_2 will dump available vibrational energy into NO, which then very rapidly undergoes V-T energy exchanges and decays to the ground state. N_2 energy exchange (both V-T and V-V) was much slower, and this resulted in a much longer lifetime of N_2 when compared to O_2 and NO. O_2 vibrational energy decay was also much more sensitive to the amount of NO seeded into the flow than N_2.

The chemical reactions, in this case, played an insignificant role in the vibrational energy decay since there was assumed to be no dissociation (formation of O atom) and three body recombination of NO with O_2 to form NO_2 was extremely slow at 30 Torr. However, at higher pressures where this reaction could become important, reaction #47 could prove an emerging contender since its rate is two orders of magnitude faster than comparable rates for V-T exchanges of NO with either O_2 or NO, and four orders of magnitude faster than those with N_2. In addition, its reverse reaction, referred to as a chemical feed reaction in Dodd (1999), regenerates vibrationally excited NO, and would directly affect the vibrational decay of NO, as well as indirectly affecting O_2 (and N_2 to a lesser degree).

With 10% humidity (0.0023 fraction H_2O), the lifetimes for N_2, O_2, and NO were shortened to 0.05 s, 930 μs and 150 μs, caused by the extremely fast V-T energy exchanges. The V-T exchange between N_2, O_2, and NO with H_2O were 9, 4, and 4 orders of magnitude faster than with the other diatomic species. V-V exchanges were also extremely fast, whereby any vibrationally excited H_2O very quickly (rates in the range of 10^{-11} cm^3/molecule/s to 10^{-14} cm^3/molecule/s) relaxed through V-T exchanges to the ground state. Increasing the humidity to 40% further shortened the lifetimes to 0.015 s, 106 μs, and 60 μs for N_2, O_2, and NO, respectively. These results showed that the concentration of water should be known precisely for long-time vibrational temperature measurements, since an increase in a fraction of a percent of H_2O in the air has a drastic effect on the vibrational temperature lifetimes of N_2, O_2, and NO. However, on a short timescale only the vibrational lifetimes of NO and O_2 are affected to a large degree, while the vibrational temperature of N_2 remains largely frozen and unaffected by the humidity level. This rapid decrease in lifetime of NO and O_2 is due to the fact that the V-T exchanges with H_2O for these two species are very fast compared to other diatomic molecules, as mentioned previously.

The affect of CO_2 was similar to that of H_2O. The presence of CO_2 shortened the lifetimes of the vibrationally excited N_2, O_2, and NO, although the mechanism through which this occurred was different than that of H_2O. This is again due to the fact that V-V and V-T energy exchange rates were fast with CO_2. CO_2 kinetics modeling is more complicated than H_2O in that multiple states must be considered and the rate constants involving the exchange between all of the states of CO_2 with the major species are not known. CO_2 had the largest effect on the vibrational decay of N_2, given that the V-T rate constant (5.2877 x 10^{-15} cm^3/molecule/s) is identical to that of H_2O and extremely fast compared to V-T exchanges with other species. For O_2 and NO, the effect is less pronounced since the rate constants are not very different from V-T exchanges with other molecules. Again, similarly to H_2O, the effect of CO_2 is seen on the long timescale, but unlike H_2O, the effect is nearly nonexistent on the short timescale. It was thought that the effect of CO_2 was most felt through the vibrational decay of N_2; the V-T and V-V rate between N2 and CO_2 are fast (reactions #4 and #39, with rate constants of 5.2877 x 10^{-15} cm^3/molecule/s and 5.11 x 10^{-13} cm^3/molecule/s, respectively). The V-V rate is very fast because the energy difference between the two vibrational states (2349 cm-1 for CO_2 and 2358.6 cm^{-1} for N_2) is small. This fast rate quickly reduces the population of vibrationally excited N_2, forming CO_2 (001). The vibrationally excited CO_2 (001) then slowly decays through V-T with other flow species mainly through direct deactivations since V-V exchanges between CO_2 (001) and other species besides N_2 were thought to be slow because the energy difference is large, although no literature sources providing the rates of these energy exchanges were found.

Therefore, vibrationally excited NO and O_2 decay preferentially through V-T collisions with CO_2 rather than V-V collisions. Collisions of CO_2 with vibrationally excited N_2, O_2, and NO can also form CO_2 in the energetically lower vibrational state (bending mode (010) at 667 cm^{-1}). The CO_2 which is formed in the (010) mode undergoes collisions to either relax to the ground state or produce vibrationally excited molecules through V-V exchanges. However, these “vibrational feed” reactions seem to play a small role since the overall effect of CO_2 was to decrease (not increase) the lifetimes of the vibrationally excited diatomics. All of these mechanisms point to the production of no real effect of CO_2 in the immediate short term (to 0.001 s, as seen inFig. 3.12, top), but affecting the long-term slow decay of all of the diatomic species to 300K. The difference between 0 and 0.0004 fraction CO_2 was negligible in the short term. Even mole fractions as high as 0.001 produced only minor effects in O_2 and N_2, and was negligible in NO because NO was more quickly relaxed through collisions with O_2 than with CO_2 (and CO_2 was present in only small quantities compared to O_2).

These results were very different than those considering the presence of H_2O. In the case of H_2O, all species were affected, with NO and O_2 being drasitcally affected drastically in the short-term. This is due to the overall increase in the rate of depletion of vibrationally excited diatomics mainly through increases in V-T rates. In the case of CO_2, again all species were affected, but the effect was not seen until longer times. In the short term, NO was virtually unaffected by the concentration of CO_2 while the bulk gases (O_2 and N_2) were affected to a mild degree. The mechanisms through which these "impurities" impart their effects on the flow differ. The effect of H_2O is felt immediately by all species, mainly through its very extremely fast V-T exchanges with the vibrationally excited diatomic species. The effect of CO_2, on the other hand, is felt through the fast V-V and V-T exchange with vibrationally excited N_2. This is due to the fact that V-T exchange rates for CO_2 with NO and O_2 are not very different from exchanges with other species. In addition, V-V exchange rates for CO_2 with NO and O_2 were largely unavailable in the literature but thought to be slow compared to the energy exchange with N_2 since vibrationally excited N_2 and CO_2 (001) are very close energetically.

D.2 Availability of Kinetic Rates

On the whole, reliable vibrational kinetics data was sparse and difficult to find. Many of the V-T rates available in the literature are calculated via the Landau-Teller formula and many times these results differed greatly from experimental data (often by several orders of magnitude). Also, much of the literature does not differentiate between V-V and V-T rates so that reported rates were the sum of the two, making it difficult to find rates specifically for V-V energy exchanges. The forward and backward rates in the literature often did not agree through detailed balance calculations. Therefore a collection of the forward rates was gathered from the various scattered resources, and the most representative of these was chosen.The reverse rate constants were then calculated through detailed balance to ensure that the long-time decays of all vibrationally excited species reach 300 K. Even through these efforts, some rate constants were not found in the literature, such as the vibrational relaxation of H_2O with CO_2 and with O atom. In practice, these rates would probably not make much difference in the overall decay since they are energy exchanges between minor flow species.

Because the availability of V-V energy exchange rates was low, the reliability of these rates was open to questioning. Further, some rates were not found at all, such as the V-V rates between NO and CO_2, and CO_2 and H_2O. Although these, again, are V-V exchanges between minor species, it would be nice to have these rates available. In addition, the energy exchange kinetics for a species with multiple vibrational levels such as CO_2, with N_2, NO, and O_2, are complex and largely unknown. Exchanges between these species and H_2O are even less well known. Also, this model only considers the ground and first excited states of the diatomics; directly measured rates for higher vibrational levels are even rarer in the literature. Initial populations of vibrationally excited H_2O and CO_2 were set at zero. Because of the shortcomings discussed above, there is much room for improvement and many other reactions could be included in the model. It is also important that the model be validated with experimental vibrational temperature data using a measurement technique such as CARS. However, the model was valuable in shedding light on the mechanisms of vibrational energy exchanges between the various species and for the expected effects of impurity (humidity and CO_2) levels on vibrational decay rates.

E. NO FLUORESCENCE MODELING

Compared to vibrational kinetics modeling, the fluorescence modeling rates was much simpler as the rates for electronic quenching of NO are well known and documented. The kinetics simulations were again run at 30 Torr and 300 K using Kintecus [Ianni (2008)] to 300 ns. The rates used in the modeling and their respective references are shown in Table 3.4:

Table 3.4 Electronic quenching rates for NO ($A^2\Sigma^+$, v'=0)

Chemical Reaction	Rate constant	Reference
NO ($A^2\Sigma^+$) + H_2O ==> NO ($X^2\Pi$) + H_2O	8.97 x 10^{-10b}	Paul (1996)
NO ($A^2\Sigma^+$) + O_2 ==> NO ($X^2\Pi$) + O_2	1.62 x $10^{-10\,b}$	Paul (1996)
NO ($A^2\Sigma^+$) + NO ==> NO ($X^2\Pi$) + NO	2.74 x $10^{-10\,b}$	Paul (1996)
NO ($A^2\Sigma^+$) + N_2 ==> NO ($X^2\Pi$) + N_2	3.70 x $10^{-13\,b}$	Nee (2004)
NO ($A^2\Sigma^+$) + CO_2 ==> NO ($X^2\Pi$) + CO_2	4.29 x $10^{-10\,b}$	Paul (1996)
NO ($A^2\Sigma^+$) ==> NO	4545454.545[a]	Luque (1999)

Note: [a] unimolecular rate constant (natural lifetime), [b] bimolecular rate constant

The results showing the effect of humidity and CO_2 on the fluorescence lifetime are shown in Fig. 3.13.

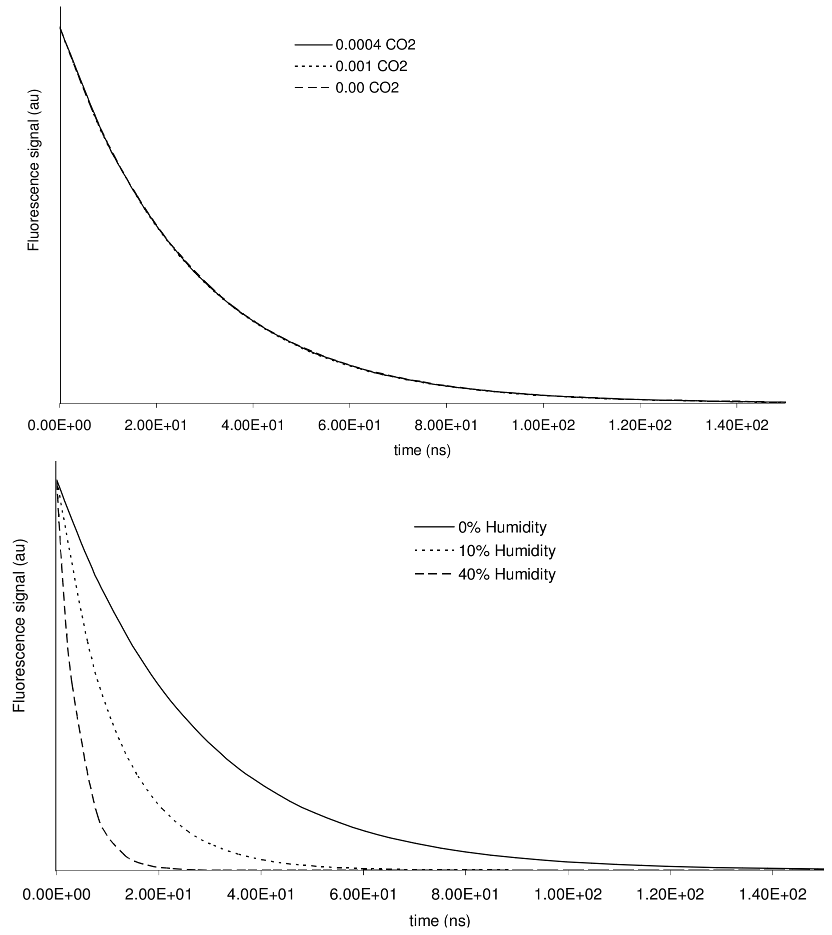


Fig. 3.13 Effect of CO_2 and humidity on fluorescence decay of NO ($A^2\Sigma^+$, v'=0)

The results showed that water had a larger effect than CO_2 on the NO ($A^2\Sigma^+$, v'=0) fluorescence decay due to the fact that the quenching rate constant for H_2O exceeds that of CO_2 by about a factor of two. Also, at the low concentrations of CO_2 expected in the flow, the effect of CO_2 was minimized. However, humidity had a significant effect on the overall fluorescence decay, and thus the intensity of the collected broadband fluorescence signal. Increasing the humidity from 0% to 10% and to 40% decreased the fluorescence lifetime from 27 ns to 11 and 4.6 ns, respectively. The CO_2 had a negligible effect on the fluorescence decay. Sensitivity analysis showed the expected trends; in the 10% humidity case, the fluorescence signal was most sensitive to the rate constant for the quenching through H_2O, and the sensitivity increased as humidity increased (NSC at 300 ns = -17.2 and -45.4 for 10% and 40% humidity, respectively). In the cases without humidity, the normalized sensitivity coefficient was largest for quenching with O_2 (NSC = -9.96 at 300 ns) with a slight sensitivity to the natural lifetime (NSC = -1.37 at 300 ns). Cases run with CO_2 showed very small sensitivities to quenching rate with CO_2 (NSC = -0.125 at 300 ns). It should be reiterated that these simulations were run with v'=0, although the quenching rates for v'=1 have been found in the literature to differ by only a few percent. For example, the rate for quenching of NO ($A^2\Sigma^+$, v'=0) with O_2 was listed as 1.51 ± 0.12 and the rate for quenching of NO ($A^2\Sigma^+$, v'=1) with O_2 was found to be 1.48 ± 0.35 in Nee (2004).

F. Resolution and Modulation Transfer Function of the Imaging System

It is important to have an understanding of the limiting resolution of the imaging system since the PLIF images will be smoothed. Ideally, the images would be smoothed to retain the maximum resolution, while minimizing the effects of shot and readout noise. The resolution of a system can be defined in many ways. One way is a simple spatial resolution. However, more commonly the resolution is defined in terms of line pairs; that is, how well an imaging system can resolve a single white line placed next to a black line. The term is adopted from terminology commonly used in the photography community.

An example of a trade tool for visually assessing the resolution of a system in terms of line pairs is shown in Fig. 3.14.

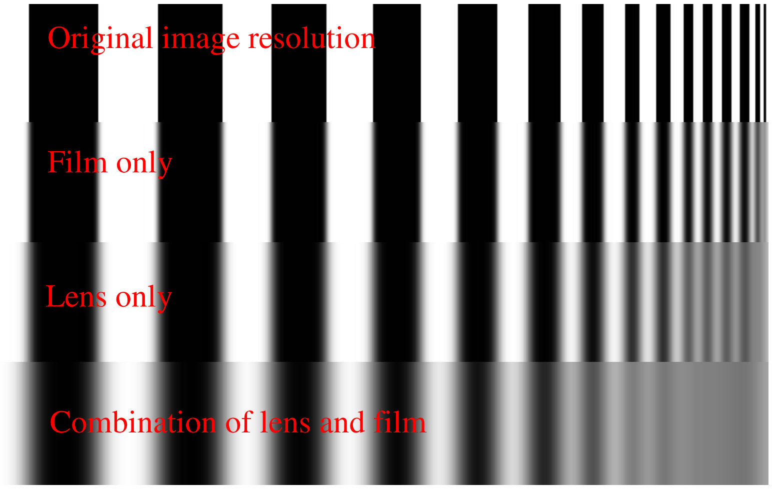


Fig. 3.14 Demonstration of resolution through use of line pair tool

Qualitatively, the effect of the film and lens is to blur the resulting recorded image. Therefore, not all of the original features are recorded; the thinnest lines which are paired the closest together are degraded the most, while the thicker lines are still resolved on the resulting image. The imaging system is then assigned a limiting resolution, and typically, this is subjective because it relies on the user determining where the lines are no longer distinct, and this value is the limiting line pair resolution of the imaging system, expressed in line pairs per mm, or lp/mm. The highest resolution achievable, where the sharpness of the original image is absolutely preserved, is the Nyquist spatial frequency, and is calculated by the equation below:

$$NyquistSpatialFrequency = \frac{1}{2w_{pixel}} \qquad \text{(Eq. 3-1)}$$

where w_{pixel} is the pixel width. For our imaging system with 23 μm pixels, this limiting resolution is 21.71 lp/mm. The details of this calculation can be found in McMillin (1993). Mapping each pixel onto a 1”x1” field of view, and assuming that the laser sheet is about 200 μm thick, the calculated field of view spatial resolution is 25 μm x 25 μm x 200 μm.

However, the original image is never completely preserved, and each component of the imaging system (lens and film in the above figure) serves to decrease the resolution, blurring the final image. The PLIF imaging system is more complex than a typical digital camera and consists of 4 main components: lens, intensifier, fiber optic coupling, and finally the CCD array. Quantitatively, the combinative effects can be expressed by the equation below:

$$\frac{1}{f_{MTFsystem}} = \sqrt{\frac{1}{f_{\text{lens}}^2} + \frac{1}{f_{\text{intensifier}}^2} + \frac{1}{f_{\text{fiber}}^2} + \frac{1}{f_{\text{array}}^2}} \qquad \text{(Eq. 3-2)}$$

By far the resolution limiting component is the intensifier, at an estimated 16 lp/mm from the literature [eg. See McMillin (1993)]. By estimating the cutoff frequencies of the components, the resulting spatial resolution in the image plane has been calculated to be about 200 um x 200 um x 200 um. However, since the cutoff frequencies for each component independently are difficult to find, and since the lens and the camera array were purchased from different manufacturers, a more accurate way to asses the resolution of the imaging system is direct measurement of the modulation transfer function, or MTF. The MTF is most commonly measured using the razor-blade method ("knife edge response"). In this technique, a sharp edge is used to cut the image into a dark and light band, and the edge response is measured by averaging or integrating the image down each column. The edge response of the system would look something like Fig. 3.15:

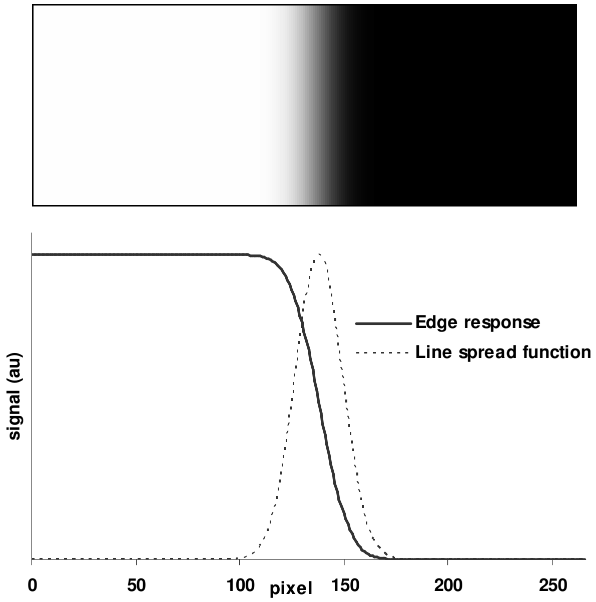


Fig. 3.15 Raw image and plotted edge response

The line spread function (LSF) is defined as the derivative of the edge response and is plotted over the edge response above. The MTF can be calculated by taking the FFT of the LSF and plotting the signal amplitude versus the spatial frequency of the imaging system. The line pair resolution is determined by locating the spatial frequency at a certain cutoff amplitude, typically 3% to 10%. This means that the original signal amplitude is damped to only 3%-10% of its original intensity and corresponds roughly to a human eye's ability to distinguish a pair of white and black lines. In the following experimental studies, the cutoff frequency has been set at 3%. The reader may find a more in-depth discussion of MTF in Smith (1990).

F.1 Experimental Determination of MTF

As described above, an edge response was obtained experimentally by seeding a vacuum cell with 1% NO in N_2 and placing a razor blade in the path of the laser beam. The resulting fluorescence was captured. Also, to investigate whether the gain on the camera (which controls the voltage across the MCP and therefore the amplification of signal) has an effect on the MTF, full resolution fluorescence images (1024 x 1024 pixels) were collected at 3 different gain levels: 0, 100, and 200 (maximum gain =255). Also, to investigate the effects of binning, the MTF was also calculated for images which were software binned 2x2 pixels. Binning is the process of regrouping pixels to create “superpixels”; for example, binning 2x2 groups every two pixels in the horizontal and vertical directions and outputs a single value for the four grouped pixels. The results showed that increasing the gain resulted in a noisier MTF, making the location of the 3% cutoff frequency harder to detect. However, the trend was still the same and smoothing the edge response had a negligible effect on the spatial frequency corresponding to the cutoff frequency. Example raw experimental signals, edge function, and calculated MTF are shown in Fig. 3.16. The 3% cutoff is labeled in the plot for MTF (bottom window).

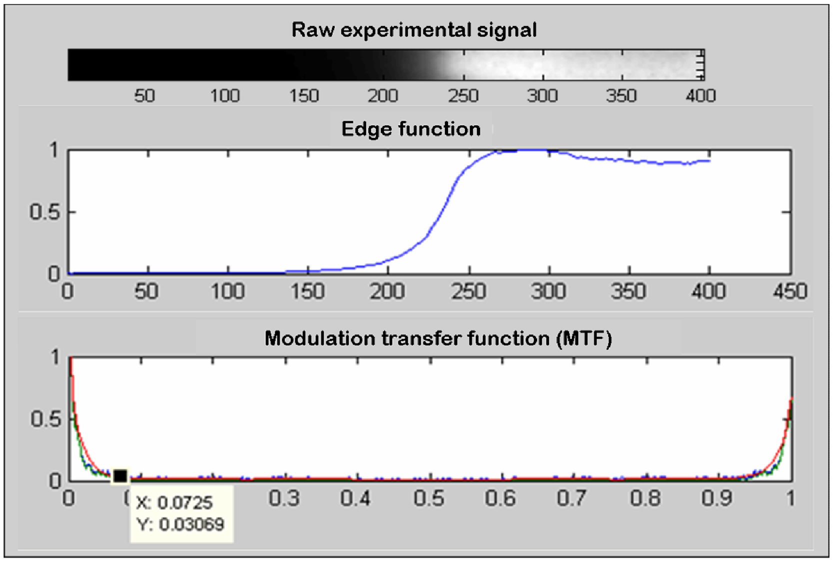


Fig. 3.16 Example experimental MTF calculation

The result yielded a line pair resolution of 0.07 line pairs/pixel, or at an image plane spatial conversion of 75.36 pixels/mm, 1 line pair per 190 µm for all levels of gain. Software binning the images also had a negligible effect, and the resolution remained near 190 to 200 um. These results agree with the estimated values using equation 4-5. A quick and rough estimate of the resolution can be obtained by observing the 10-90% distance (in pixels) of an edge response. For example, if a system has a 10-90% edge response of 5 pixels, then the 10% MTF amplitude will correspond to 1/5 line pairs/pixel, or 0.2 line pairs/pixel.

G. Computer-based Image Analysis Study of PLIF Fluorescence Signal Processing

As mentioned before, the plasma is expected to induce rotational and translational temperature fluctuations of 300 +/- 10 K. The goal is to detect these small fluctuations using NO PLIF, a challenging task; literature using NO PLIF for these purposes has not been found to date. In order to detect these extremely small fluctuations, the inherent noise on the fluorescence images must be suppressed. The most straight-forward way to accomplish noise suppression is through the use of image Gaussian smoothing. However, since the temperature will be non-uniform throughout the image, i.e. it is expected that the fluctuations will occur in “pocket-like” structures, Gaussian smoothing must be used sparingly in order to avoid damping out true temperature fluctuations in the flow itself. Therefore, it becomes challenging task to preserve the true and very small temperature fluctuations that occur in the flow while damping the noise in the images.

The noise in the images appear in two main forms. The first is high frequency noise originating from the camera and include shot, dark, and readout noise. Shot noise (also called photon noise) comes from the Poisson distribution of photons which impinge onto the CCD array, and scales with the square root of the number of photons which hit the array. Shot noise increases with intensifier gain. Dark noise is inherent to the CCD array and is due to thermally generated electrons. Read noise is a product of the CCD read-out process and is a uniform amount of noise applied to each pixel as it is read out. One equation which shows the signal to noise ratio as a function of these sources of noise is shown below:

$$SNR = \frac{FQt}{\sqrt{FQt + BQt + Dt + R^2}} \qquad \text{(Eq. 3-3)}$$

As seen from the equation above, there are methods to decrease the effects of these sources of noise. Overall SNR can be increased by increasing quantum efficiency and/or increasing integration time. Shot noise can be decreased by increasing the signal (signal photon flux) (and therefore decreasing the gain and increasing signal to noise ratio). Readout noise cannot be decreased since it is inherent to the camera itself, but effect on SNR can be reduced by increasing integration time or by on-chip binning, where each superpixel would have only one value of read-out noise applied. Dark noise is usually not a large issue as most scientific CCD arrays are cooled to reduce dark noise to negligible levels. With the use of UV intensified ICCD cameras and the addition of filters which block background visible light, background photon flux is usually not an issue. Therefore, most PLIF applications are shot-noise limited. However, under very low signal conditions, such as single-shot imaging, read-noise can become a significant factor in SNR. Therefore it is preferable to on-chip bin under these low-signal read-noise limited applications.

The second source of noise is a low-frequency "banding" structure which originates from laser beam inhomogeneity. The laser profile exiting the Sirah dye lasers at 226 nm is very nonuniform and has bright and dark regions, which, when the laser is formed into a sheet, translates to bright and dark bands in the fluorescence images. Therefore, to develop an image analysis scheme to understand how to minimize the effects of these sources of noise while preserving the temperature fluctuation structures of a only few percent about the mean temperature using NO PLIF, a computer-based study has been conducted using in-house codes written in MATLAB. The computer based study is followed by experimental validation.

G.1 Generation of NO PLIF Data

The first goal of the software-based experiment is to create realistic computer-generated NO PLIF data on which to apply the image processing schemes. Shown in Fig.

3.17 is a typical NO PLIF image of the $A^2\Sigma^+_{1/2}(v'=0) \leftarrow X^2\Pi_{1/2}(v''=0)$ transition of NO at R_1+Q_{21}(3.5). This image was taken using the portable vacuum chamber. However, the nozzle was replaced with a simple stainless steel tube for the slow-flow conditions needed for this experiment. The flow conditions were controlled using two MKS flowmeters (Model 1179). The flowmeters mixed 10 sccm room air with 1 sccm 10%NO in N_2 balance (supplied by Matheson Tri-Gas). The total pressure in the evacuation chamber was held at 21.3 Torr using a Welch 1402 pump. The laser and imaging systems were the same as described previously. The dye laser power was about 1.5 mJ/pulse. The laser beam was collimated by a combination of cylindrical lenses to a width of about 0.5", and manually cut at the edges to a width of 0.25". The dye laser power should not exceed this amount at the risk of approaching the fluorescence roll-off regime. Both intensified cameras were fitted with UG5 filters to block elastic scatter, but it should be noted that these filters also significantly decreased fluorescence signal (leading to noisier data). This particular image was taken with an on-chip binning of 4x4. Note that this is a raw image; no image corrections have been applied.

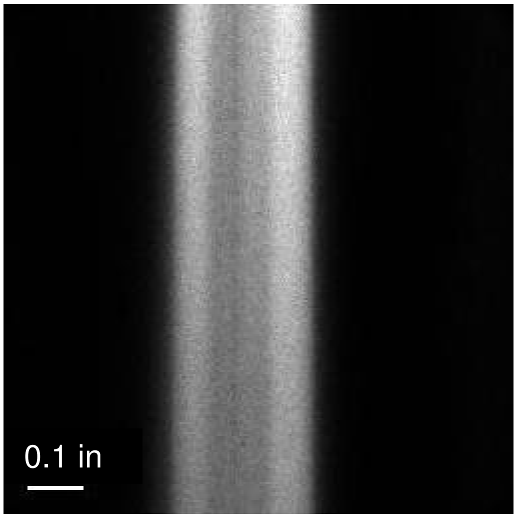


Fig. 3.17 Sample NO PLIF image for image generation

In the image above, the laser beam inhomogeneity is clear; the beam shows significant “banding” in the horizontal direction. Absorption effects are also observed by the slight loss of signal in the vertical direction (along the path of the laser sheet). By simply plotting slices along the horizontal and vertical directions, this image showed low-frequency banding variations of approximately 30-50%, and high frequency noise (mainly shot noise) variations of approximately 15-20% about the mean signal intensity. Both ICCD cameras showed similar variations. These variations served as the basis for the computer-generated NO PLIF data. The second step is to create a pair of computer-generated data which will represent the NO PLIF images from the $A^2\Sigma^+_{1/2}(v'=0) \leftarrow X^2\prod_{1/2}(v''=0)$ transitions of NO. In this analysis, I2 refers to the “lower” rotational line at J=3.5 and I1 refers to the “higher” rotational line which will be varied in the analysis. All of the structure in the images is confined to I2, although this assignment is arbitrary and does not affect the outcome of the analysis.

For I2, a 150 x 150 pixel image was generated with structures on the order of 5 pixels. These structures would be very small compared to the structures seen. For comparison, the turbulent structures will be on the order of 0.1 inches, which corresponds to roughly 30 pixels on the 4x4 image shown above, and 17 pixels on a 7x7 image. This image is shown in Fig. 3.18 (left top). These structures were randomly distributed throughout the image. The intensities of the structures ranged randomly within values which gave a temperature range of 300 +/- 10 K. A matrix of random noise was generated in the range of 0.9 to 1.1 and applied to the image. The resulting image is seen in Fig. 3.18 (top right). A banding function was then applied to the matrix. The banding function is overlaid on the results of its application to the noisy image (bottom left). Finally, a linear correction of 15% was applied in the vertical direction, and the result is shown (bottom right). As can be seen from the image, these corrections nearly masked the original structure of the flow. The goal of the image analysis software was to recover the true temperature map as accurately as possible.

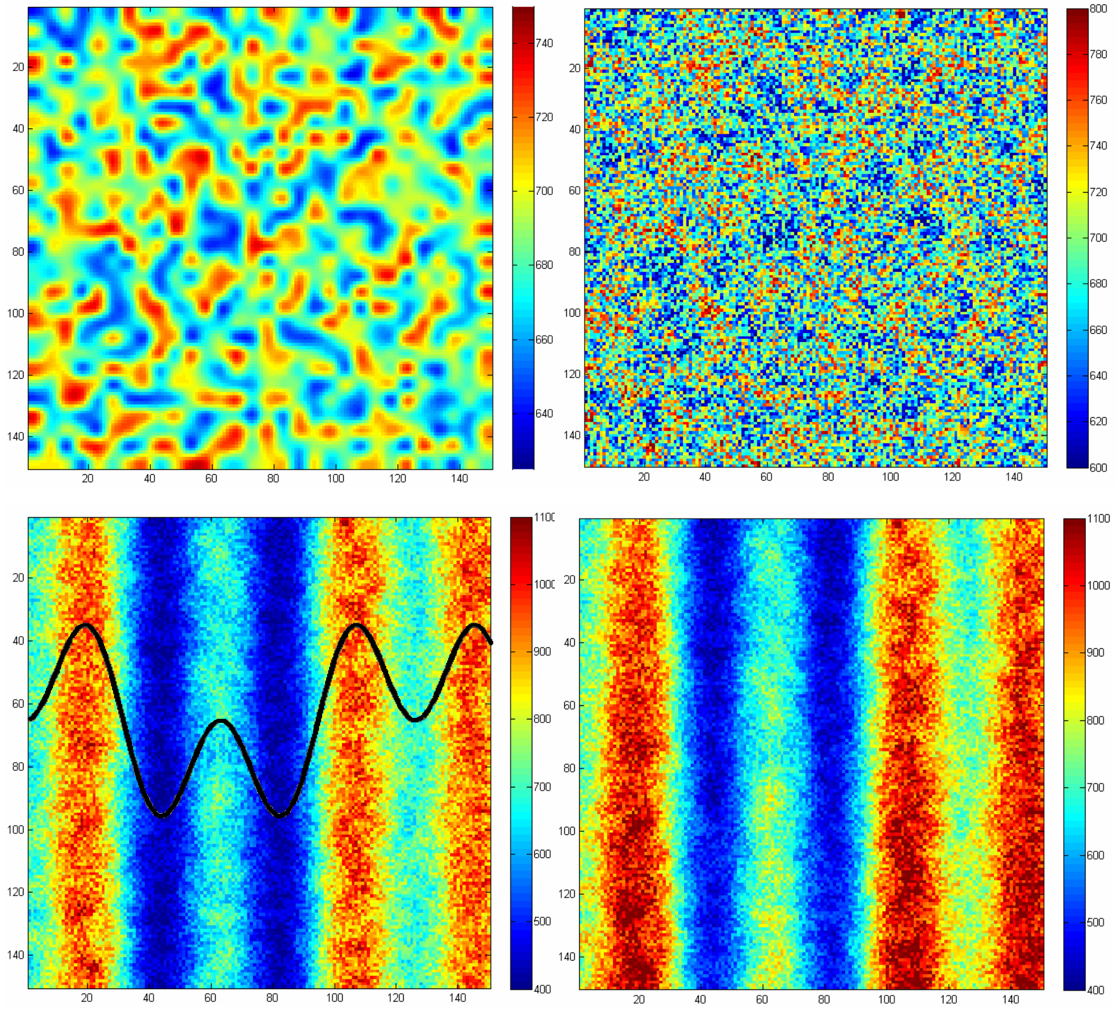


Fig. 3.18 NO fluorescence signal generation, I2

The corresponding images for I1 are shown in Fig. 3.19. The corrections were applied in the same way to I1, except that the image began with a uniform image with all values set to 100. The banding correction was altered slightly to give higher frequency banding; both low frequency and high frequency banding structures were seen in the dye laser beam outputs.

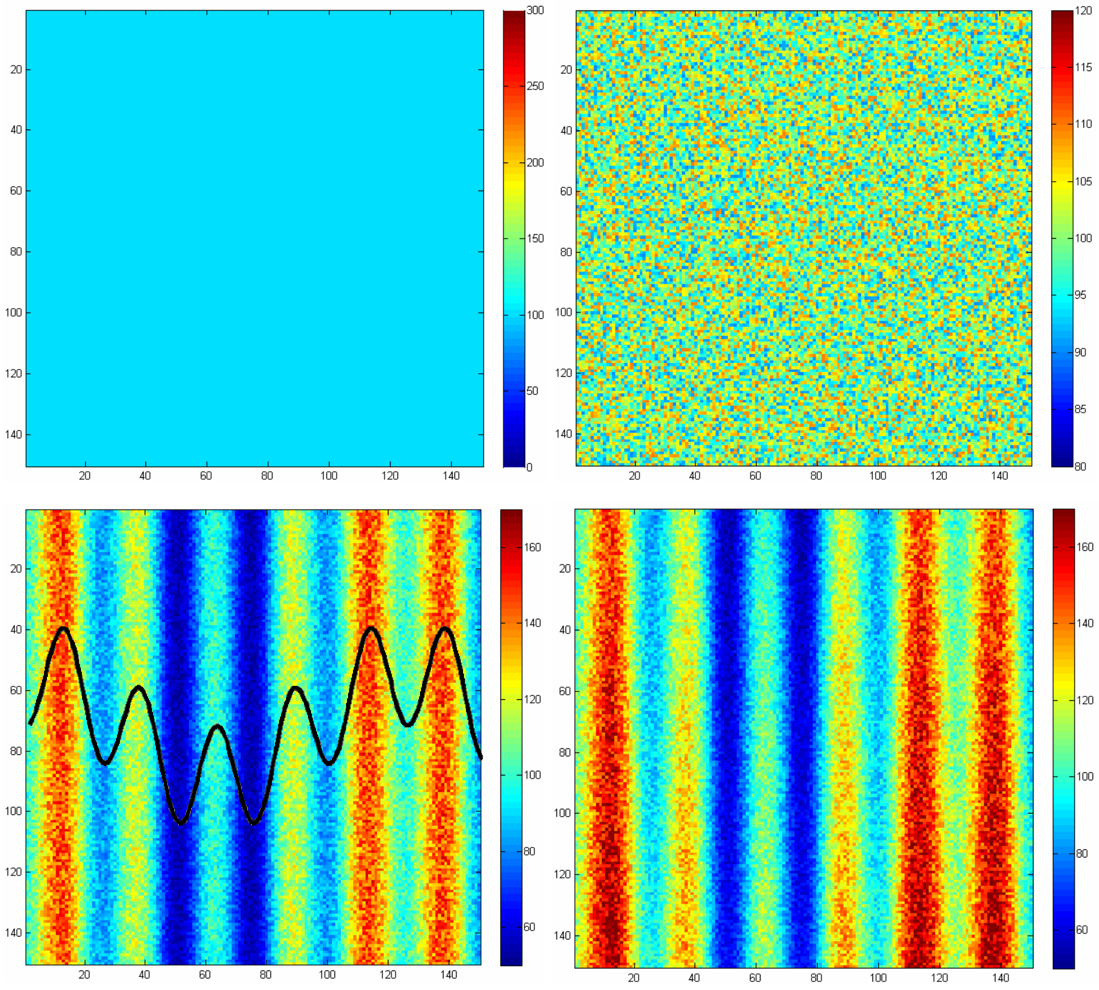


Fig. 3.19 NO fluorescence signal generation, I1

Next, the image pair was processed using a variety of commonly used corrective procedures. Exploration of how the following image processing procedures affected uncertainties of the resulting NO PLIF temperature map was conducted:

1. "Tightness" of the banding correction. In order to correct for the horizontal banding in the images, the image was integrated in the vertical direction to obtain one banding function. This function was then smoothed by a running average of a number of pixels; this number of pixels will be referred to as the "tightness" of the banding correction and was varied in this experiment. The image was then normalized in the horizontal direction by dividing each row by the function.

2. Banding correction of each row independently rather than integrating in the vertical direction, meaning that each row would have its own banding profile.
3. Gaussian filtering of the images. This was the equivalent of applying a low-pass filter to the images. We explored Gaussian filters with a number of kernel sizes, as well as not filtering the images at all.
4. Choice of rotational lines. This choice is an important variable since choosing a pair of rotational lines with a larger energy difference decreases temperature uncertainties, but gives inherently noisier images due to lower population of the higher rotational line. This tradeoff was explained in detail in McMillin (1993). For this analysis, J=3.5 was maintained as the lower rotational lines, and the higher rotational lines was varied from J=5.5, 10.5, 14.5, 15.5, 16.5, and 19.5.

G.2 Image Processing Results

The image processing test matrix is listed inTable 3.5:

Table 3.5 Image processing conditions

Test #	Banding correction tightness (pixels)	Vertical integration (Y/N)	Gaussian smoothing
1	2	No	None
2	2	No	2x2
3	2	No	3x3
4	2	Yes	2x2
5	10	No	2x2
6	5	No	2x2
7	3	No	2x2
8	2	No	4x4
9	2	No	5x5
10	2	No	6x6
11	2	No	7x7
12	2	No	8x8
13	2	No	9x9
14	2	No	10x10
15	2	No	11x11
16	2	No	12x12
17	2	No	13x13
18	2	No	14x14
19	2	No	15x15

The test matrix above was repeated for each pair of rotational lines. Also, to explore the effect of Gaussian smoothing as a function of average structure size, Gaussian kernels of 1 to 30 were applied towards an analogous temperature map with temperature structures on the order of 10x10 pixels. The true temperature maps (with structures on the order of 5x5 and 10x10 pixels), are shown in Fig. 3.20.

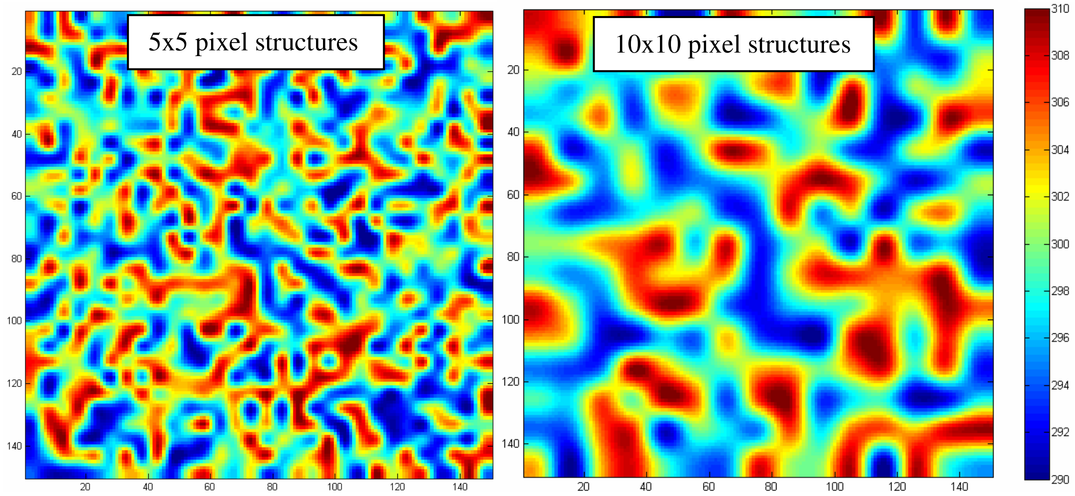


Fig. 3.20 True temperature map for 5x5 and 10x10 structures

The banding tightness was varied from 10 to 2 pixels, and the resulting derived temperature maps are presented in Fig. 3.21. As an example, the top left image is overlaid with the true banding function (black), and the derived banding function (white).

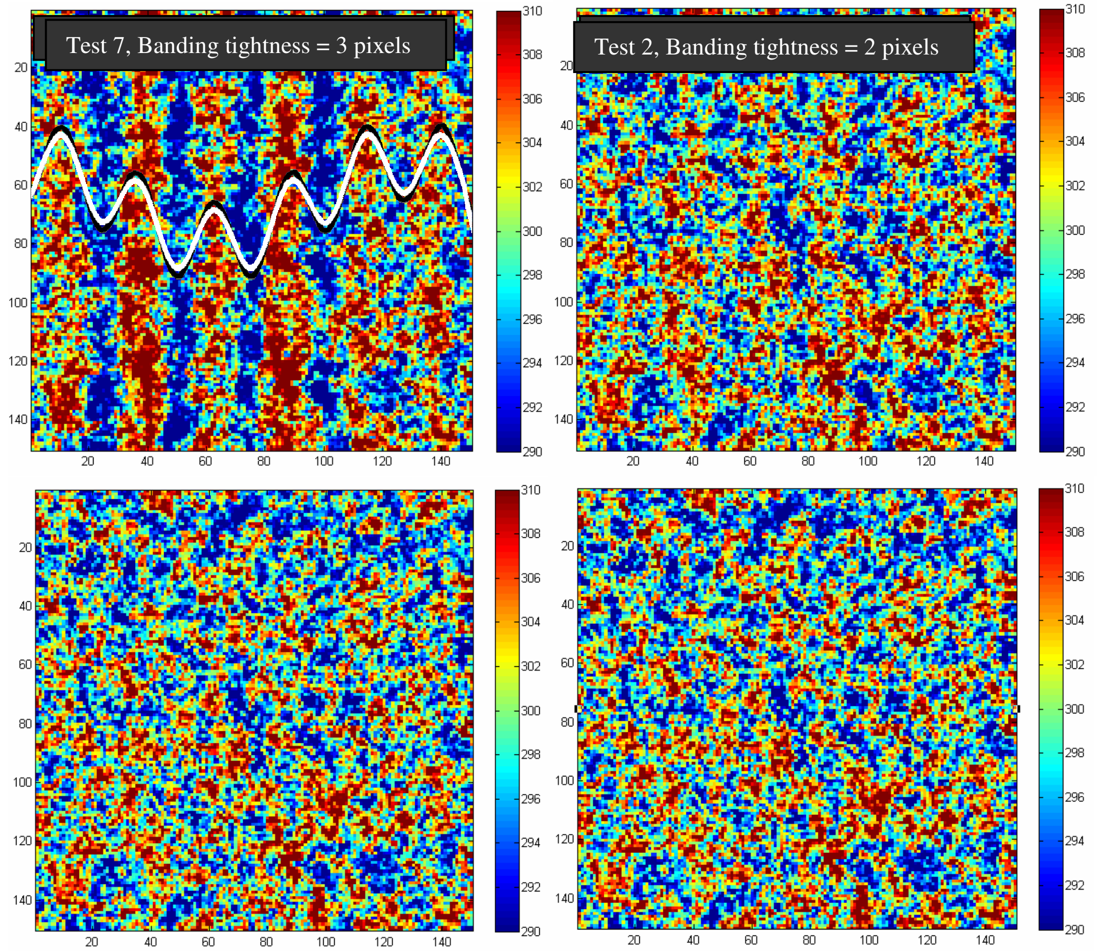


Fig. 3.21 Effects of banding tightness

The effects of banding tightness were intuitive; as the number of pixels smoothed was increased, the image software was unable to capture the full amplitude of the banding function, and the image was under corrected for the banding function. This under correction caused inherent errors which were manifested in hot and cold bands (from 315 to 285K) in the temperature map. In fact, the temperature map is extremely sensitive to the banding function; a few percent errors (3-4%) in banding function correction cause the temperature to swing in the range of +/- 20K. These errors ranged from mild to severe as the banding tightness was increased from 2 to 10 pixels. Also, the banding errors were more severe for the image with a higher frequency banding structure

(I1 in this case), since these structures would be most affected due to banding function smoothing. The study on banding tightness showed that the least number of pixels as possible should be used in this analysis in order to fully capture the banding function of the laser beam profile. A banding tightness of 2 pixels (lower right of Fig. 3.21), removed all hints of hot and cold temperature bands and thus demonstrated that the full banding structure was captured. However, by comparing this image to the true temperature map, even the most robust banding function correction alone does not adequately recover the true temperature structure due to insufficient noise suppression.

Next, image processing by correcting each row with its own banding function was investigated. The results are shown in Fig. 3.22. This conclusion could have been reached intuitively, since operating under this correction would smear out all structures in the x direction. Also, this leads to large errors in the y direction due to the fact that each row is uncorrected with adjacent rows.

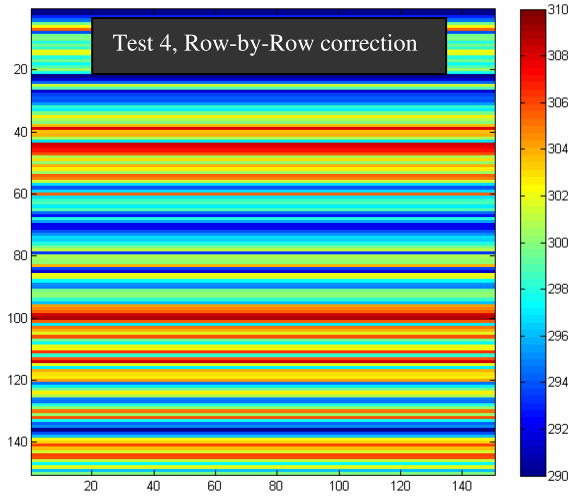


Fig. 3.22 Effects of row-by-row banding correction

Another technique which is commonly used to decrease noise is Gaussian smoothing, which is equivalent to applying a low-pass filter to the data in an attempt to decrease the high-frequency shot noise. Selected temperature maps from the line pair 3.5 and 16.5 for varying Gaussian kernel sizes (in pixels) are shown in Fig. 3.23.

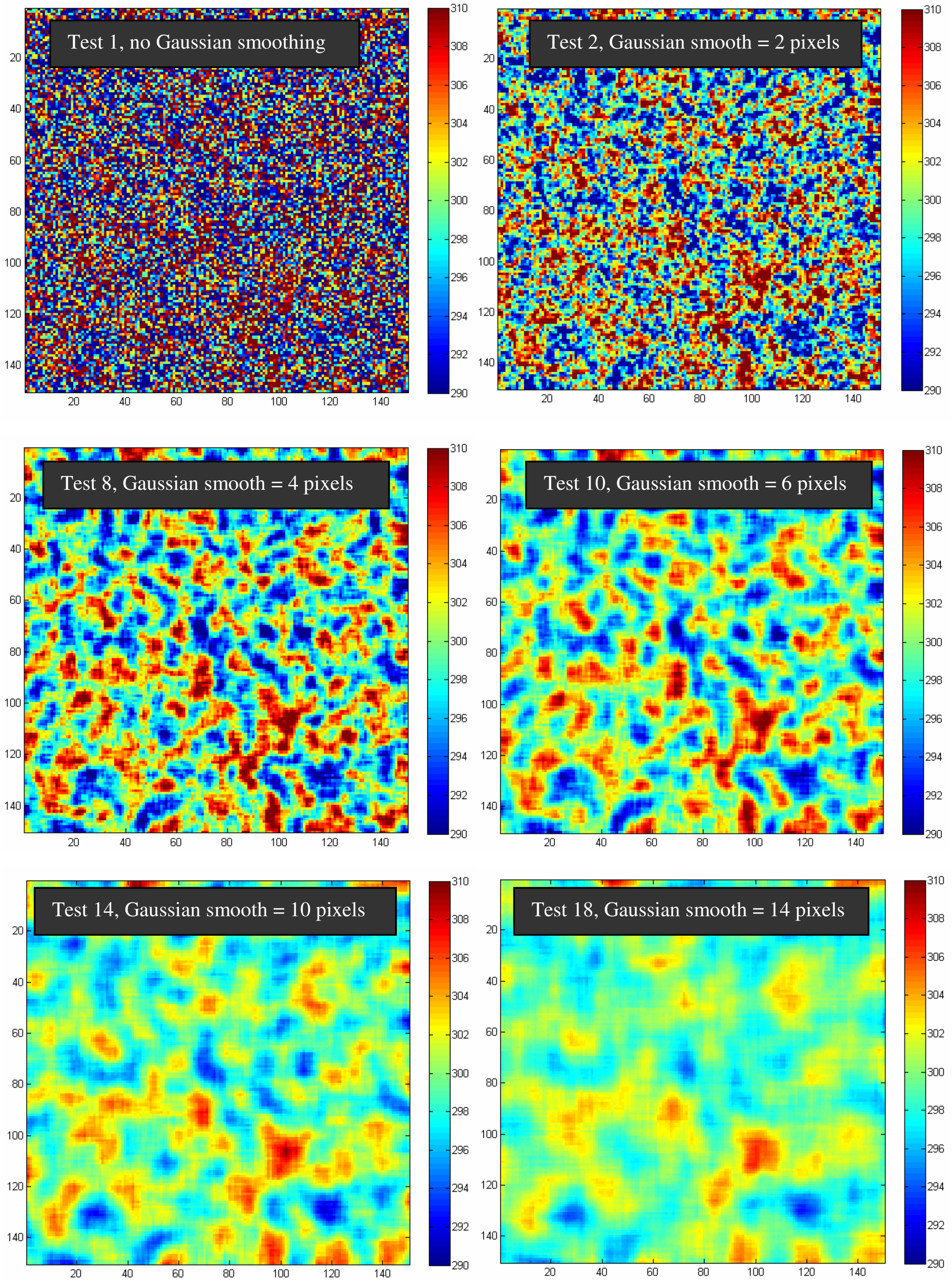


Fig. 3.23 Effects of Gaussian smoothing

The application of Gaussian smoothing was interesting because it demonstrated a trade-off between increasing signal to noise and resolution. For example, in Test 1 (no Gaussian smoothing), the structure of the temperature was virtually indistinguishable from noise. However, as the Gaussian kernel was increased, the structure began to emerge, and looks qualitatively similar to the true temperature when a Gaussian kernel of 5-7 pixels was used. Further increases in kernel size served to blur the structure, as seen in Tests 14 and 18. Quantitatively, we can measure the effect of Gaussian smoothing on the image by investigating T$_{\mathrm{rms}}$, defined as: $T_{rms} = \left[\overline{(T')^2}\right]^{1/2}$ taken over the central 100 x 100 pixels of the image, thereby negating any edge effects of the Gaussian smoothing. A plot of the effect of Gaussian smoothing on T$_{\mathrm{rms}}$ is shown in Fig. 3.24, with an inset showing a single row.

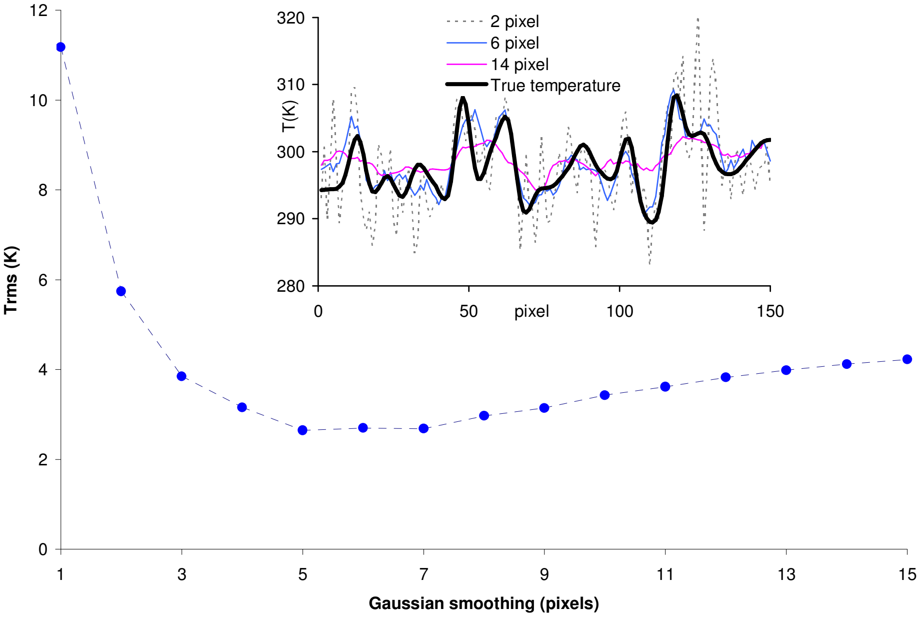


Fig. 3.24 Effect of Gaussian smoothing on T$_{\mathrm{rms}}$(Structure size = 5 pixels)

From the above plot, it was seen that Gaussian smoothing was definitely beneficial in decreasing noise. Initially, Gaussian smoothing offers a dramatic decrease in uncertainty. Even minimal 2x2 and 3x3 smoothing decreased the uncertainties by factors of 2 and 3, respectively. The Gaussian smoothing was most beneficial where T_{rms} was lowest, about 5-7 pixels. This point represented the optimal trade-off between uncertainties and resolution. At this point T_{rms} reached a very low <1%. At these measurement uncertainty levels, real temperature fluctuations on the order of 3% (300+/-10 K) were easily distinguished. Below this size, the image was dominated by noise, whereas above this point, decreases in resolution lead to larger uncertainties. Generally, it was found that Gaussian smoothing with a kernel on the order of the structure size led to the lowest temperature uncertainties, known as the theory of matched filters. In order to verify that this theorem follows in this case, differing kernels of Gaussian smoothing were applied towards the larger structure sizes of 10 pixels. The analogous plot of Trms versus Gaussian smoothing kernel is shown in Fig. 3.25, verifying that indeed, the theory of matched filters applies to differing structure sizes, giving a minimum Trms around Gaussian smoothing kernel sizes of 9-11 pixels:

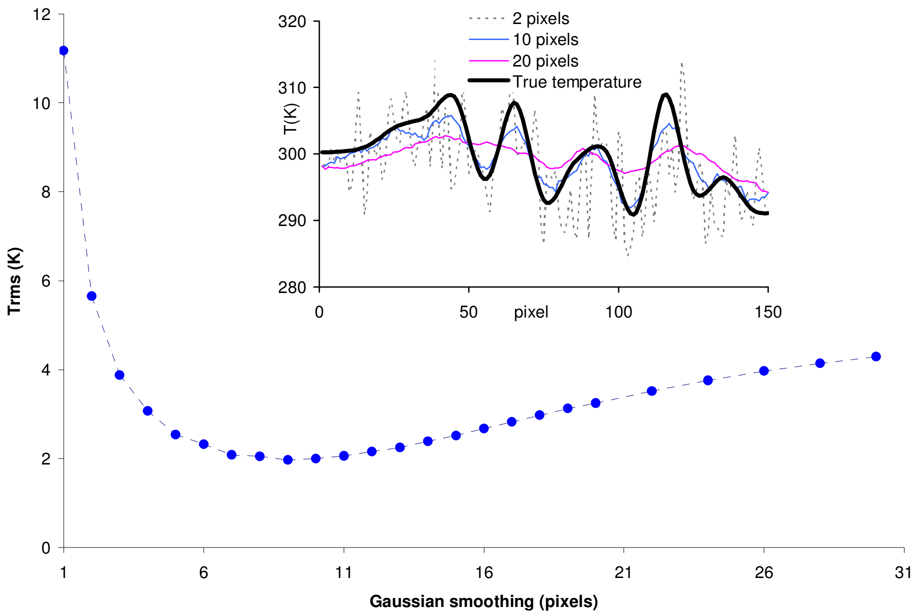


Fig. 3.25 Effect of Gaussian smoothing on T_{rms}(Structure size = 10 pixels)

Lastly, the effect of rotational lines was investigated. The lower line was maintained at J=3.5 since this line was experimentally easily accessible (well separated and easily distinguished from adjacent lines). The higher rotational lines were varied (J=5.5, 10.5, 14.5, 15.5, 16.5, and 19.5) to investigate the tradeoff between greater energy separation (lower uncertainty) and lower population of the higher rotational line (lower image signal to noise). A selective set of data showing the effect of choice of rotational lines on T_{rms} for a Gaussian smoothing kernel of 1,2, and 3 pixels is shown in Fig. 3.26.

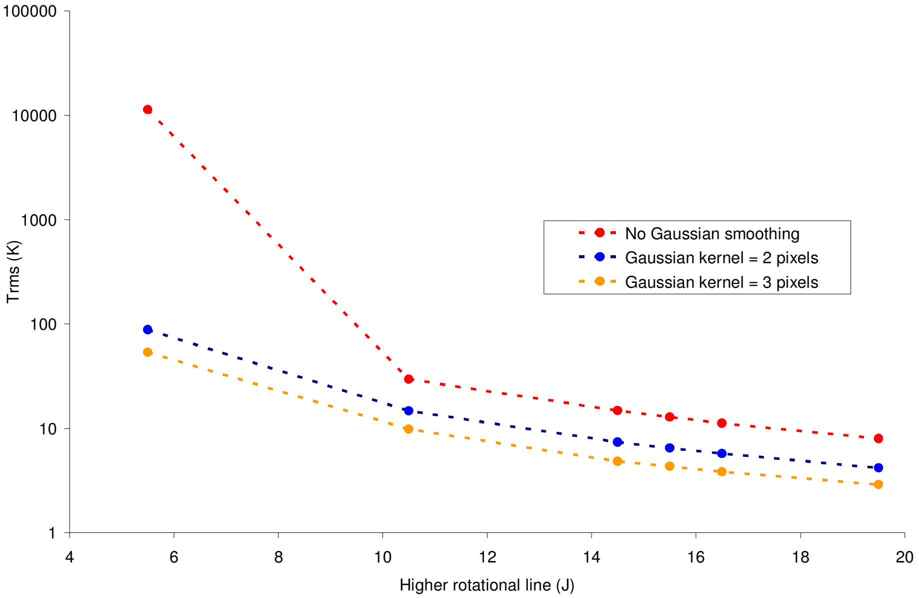


Fig. 3.26 Effect of rotational lines on T_{rms}

In practice, the lower line would be R_1+Q_{21}(3.5) and the higher line would originate in the Q_1+P_{21} branch. This choice was made on the basis that maximum signal would be available in these branches for the rotational states chosen while avoiding interference from other branches. From the plot above, it is seen that T_{rms} decreases as the rotational energy spacing is increased, as expected. The rotational states which give an uncertainty or <2% included J=14.5, 15.5, 16.5, and 19.5. The NO LIF spectra simulated using LIFBASE showed that J=14.5 was somewhat close to the R_{21} branch and was discarded as an option. Between J=15.5 and 16.5, 16.5 was chosen because although J=16.5 was slightly lower in signal, an additional energy spacing of 54 cm^{-1} was gained over J=15.5, a non-trivial amount when compared to kT of 208 cm^{-1}. The energy spacing between J=3.5 and J=16.5 was 440 cm^{-1}, while the energy spacing between J=3.5 and 19.5 was 621 cm^{-1}, however, the 19.5 transition was significantly lower in signal (roughly half) compared to the 16.5 line. Taking into thought the advice of McMillin (1993), a good starting point is to begin with a pair of lines where the spacing is approximately equal to kT. However, past experiments using J=3.5 and 10.5 (ΔE=165 cm^{-1}) gave much too large temperature uncertainties of ~10%. Therefore, it was decided that J=3.5 and J=16.5, which gave T_{rms} of <2% and <1% at 2 and 5 pixel Gaussian smoothing, was an appropriate pair of lines to begin the real NO PLIF temperature experiments.

G.3 Experimental Validation of Image Analysis Procedures

In order to test the validity of the image processing software in producing acceptable levels of temperature uncertainty, experiments were conducted in a slow uniform, 300K, air flow with seeded NO. The air was simple room air (~40% humidity) and the 10% NO in N_2 mixture was supplied by Matheson Tri-Gas and were mixed in a 9:1 ratio (18 sccm and 2 sccm, respectively) using MKS flow controllers to produce NO seeding conditions of about 1% in air. Pressure inside of the cell was monitored and maintained at 28 Torr. The probe laser powers were held at approximately 2 mJ/pulse which translated to approximately 0.3-0.4 mJ/pulse at the test section. UG5 Schott filters were used to block elastically scattered light, but also decreased signal levels by about 50%. The data were binned (on board) 2x2 and 4x4 to produce images which were 512 x 512 and 256 x 256 pixels, respectively. These two different binning conditions provided insight into the effect of binning on Trms.

The fluorescence was imaged onto two Andor 16-bit ICCD cameras fitted with F4.5 UV lenses, each gated for 100 ns. Gain levels on the cameras were adjusted to keep signal levels high (40000-50000 counts out of 65536). In addition, the cameras were angled relative to the laser sheet plane to collect fluorescence from the same side of the flow cell. These conditions represent the worst-case signal to noise ratio scenario for the experiment, since the on-board binning was kept low, filters were used, and the angling of the camera necessitated the use of anti-warping software which introduces a small amount of error due to software interpolation. A schematic of the experiment layout is shown in Fig. 3.27.

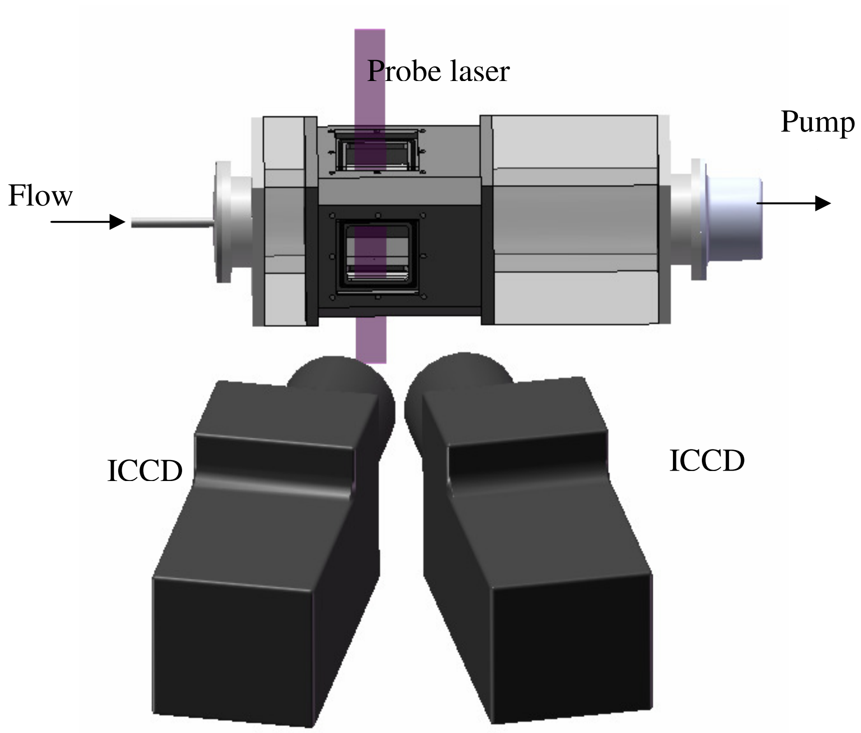
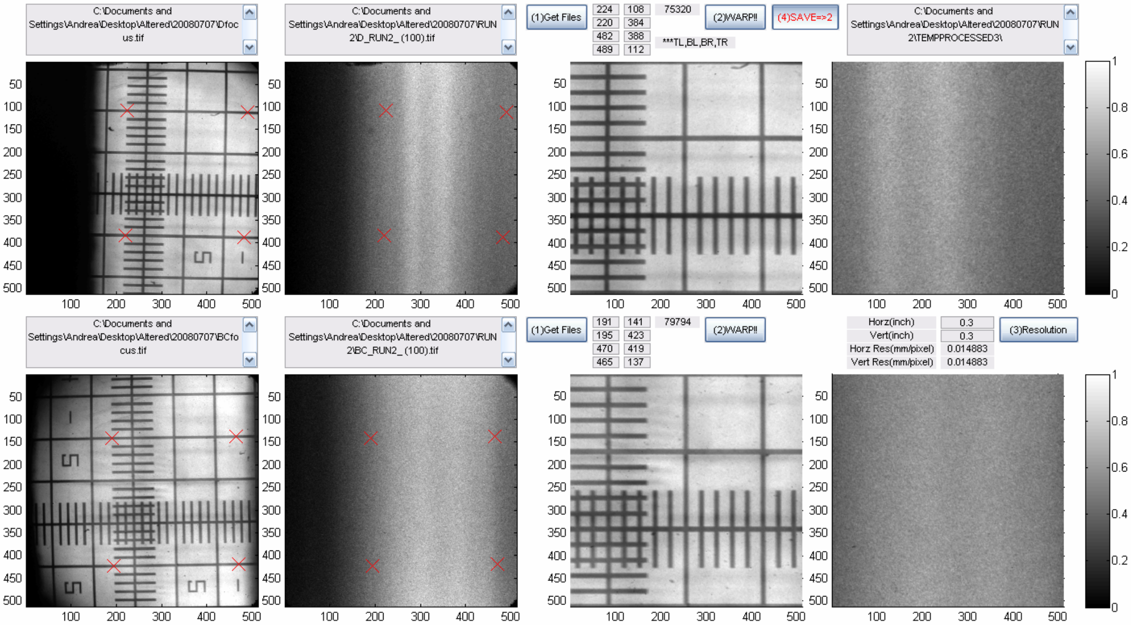


Fig. 3.27 NO PLIF computer image analysis study experimental schematic

A dot card was used to ensure that the cameras imaged equivalent fields of view. The dot card image was collected using a mercury pen lamp (emission at around 250 nm) for minimal camera lens chromatic aberration. The timing of the experiment was controlled using a BNC digital delay generator. Three hundred single shot image pairs were collected, and the first 20 image pairs were discarded due to laser warm-up. The remaining image pairs were processed in MATLAB using the image processing schemes discussed previously. The image processing software and its use will be discussed briefly in the following sections.

G.3.1 Image processing software

The image processing software was written in-house using MATLAB and included a graphical user interface (GUI) to make it more user-friendly. The features of its use will be discussed, and the full code is given in the Appendix. Printscreens showing its use at each stage of the image processing are given in this section. First, the software warps a selected image pair to equivalent fields of view via a dot card image to a user selected field of view. In this analysis, an area of 0.3"x 0.3" was chosen. In Fig. 3.28, the top and bottom row images corresponded to the lower (J"=3.5) and higher (J"=16.5) rotational lines, respectively.



C:\Documents and Settings\Andrea\Desktop\Altered\20080707\Dfocus.tif

C:\Documents and Settings\Andrea\Desktop\Altered\20080707\RUN 2\D_RUN2_(100).tif

224	108	75320
220	384	
482	388	***TL,BL,BR,TR
489	112	

C:\Documents and Settings\Andrea\Desktop\Altered\20080707\RUN 2\TEMPPROCESSED3\

C:\Documents and Settings\Andrea\Desktop\Altered\20080707\BCfocus.tif

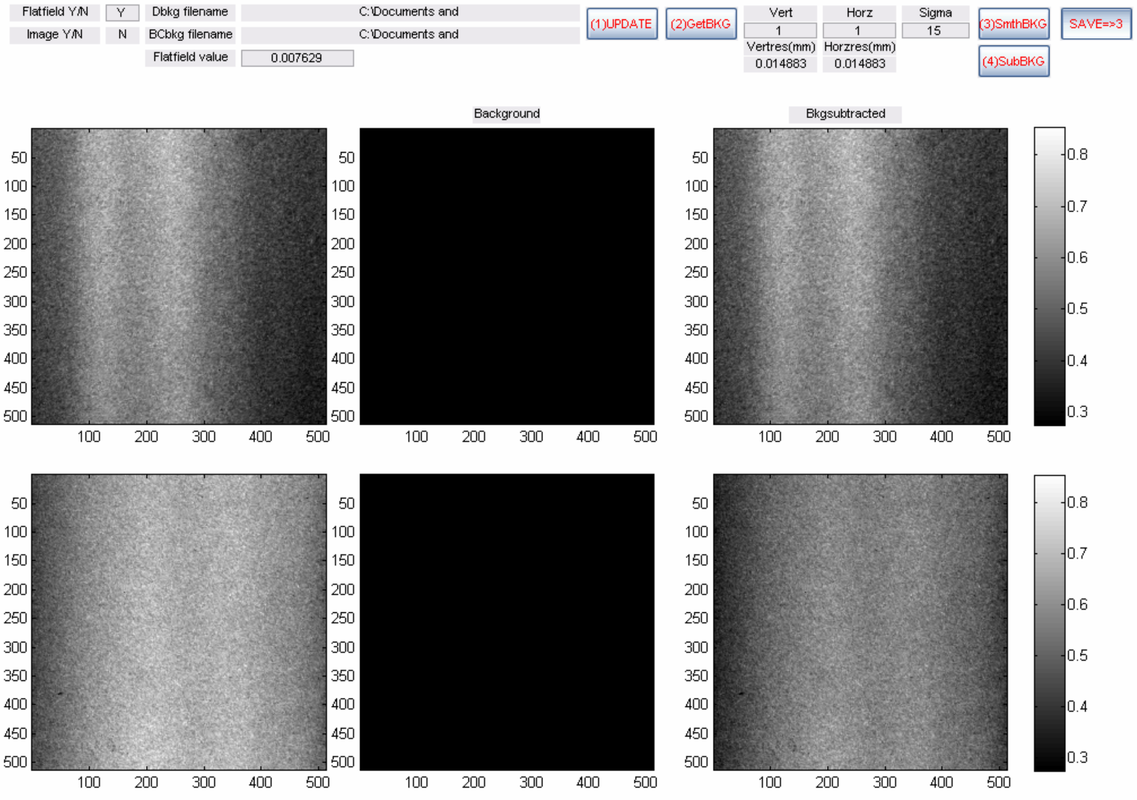
C:\Documents and Settings\Andrea\Desktop\Altered\20080707\RUN 2\BC_RUN2_(100).tif

191	141	79794
195	423	
470	419	
465	137	

Horz(inch)	0.3
Vert(inch)	0.3
Horz Res(mm/pixel)	0.014883
Vert Res(mm/pixel)	0.014883

Fig. 3.28 Image processing stage 1

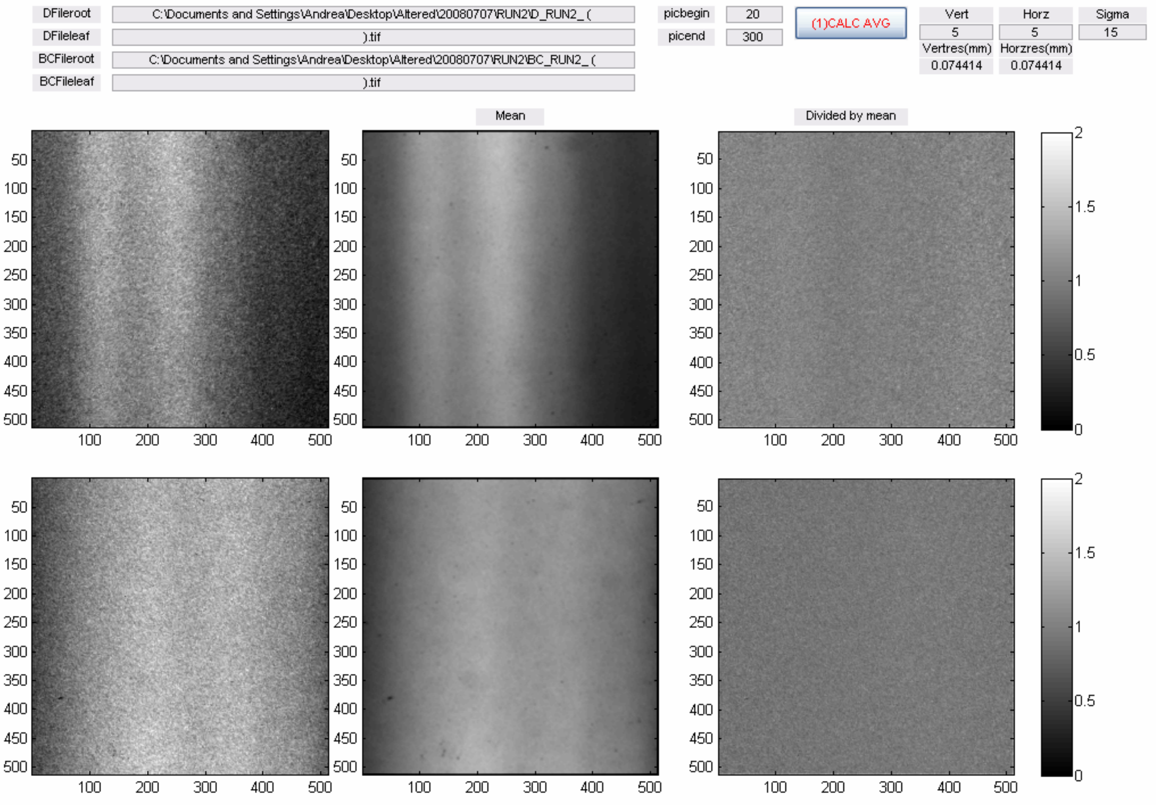
In the second step of image processing (Fig. 3.29), the background was subtracted. The software gives the user the option of using a background image, or a flat field background correction. This should be given some consideration since subtraction of the background is linear correction on the signal level, but a nonlinear correction to the temperature map. Therefore, incorrect subtraction of the background can lead to systematic errors in temperature mapping. Fortunately, typical background levels (dark counts and background scatter) totaled to about 500 counts out of a full scale 65535 counts on the 16-bit ICCD detector, or <1% full scale. Therefore, maximizing signal counts decreased the effect of any slight errors in background subtracted. In addition, in the case of a background image, the background should be smoothed to minimize propagation of uncertainties to the image. A flat field correction was applied in this case for demonstration purposes. This step is shown in Fig. 3.29



Flatfield Y/N Y
Image Y/N N
Dbkg filename C:\Documents and
BCbkg filename C:\Documents and
Flatfield value 0.007629
Vert 1
Horz 1
Sigma 15
Vertres(mm) 0.014883
Horzres(mm) 0.014883

Fig. 3.29 Image processing stage 2

In the third step of image processing (Fig. 3.30), the software averaged all the images and divided the instantaneous fluorescence image by the averaged image. This technique effectively eliminated any false temperature features incurred by features which are present in every image, such as spots on the windows or camera lenses. However, dividing by the mean image should be used cautiously because it will damp out any large-scale, static, temperature gradients which are present in every image. For example, if one side of the image is slightly warmer than the other, such as would be the case near the electrodes, this technique would remove this feature from the calculated temperature map. Therefore, dividing by the mean should be used in flows where the average temperature of the entire image is constant, or in flows where large scale static temperature gradients are to be ignored. In Fig. 3.30, the instantaneous fluorescence images are shown in the left column, the mean image in the middle column, and the divided image in the right column. As in the background subtraction, the average image was smoothed to avoid propagation of uncertainties. The images in the right column showed that dividing by the mean removes nearly all laser beam inhomogeneities present in the images. This technique is also excellent in the case that the laser beams are not exactly vertical making a column-by-column integrated banding correction difficult. The remaining faint banding structure in the divided images resulted from instantaneous fluctuations in laser beam inhomogeneity.



DFileroot C:\Documents and Settings\Andrea\Desktop\Altered\20080707\RUN2\D_RUN2_(

DFileleaf).tif

BCFileroot C:\Documents and Settings\Andrea\Desktop\Altered\20080707\RUN2\BC_RUN2_(

BCFileleaf).tif

picbegin 20

picend 300

Vert 5

Horz 5

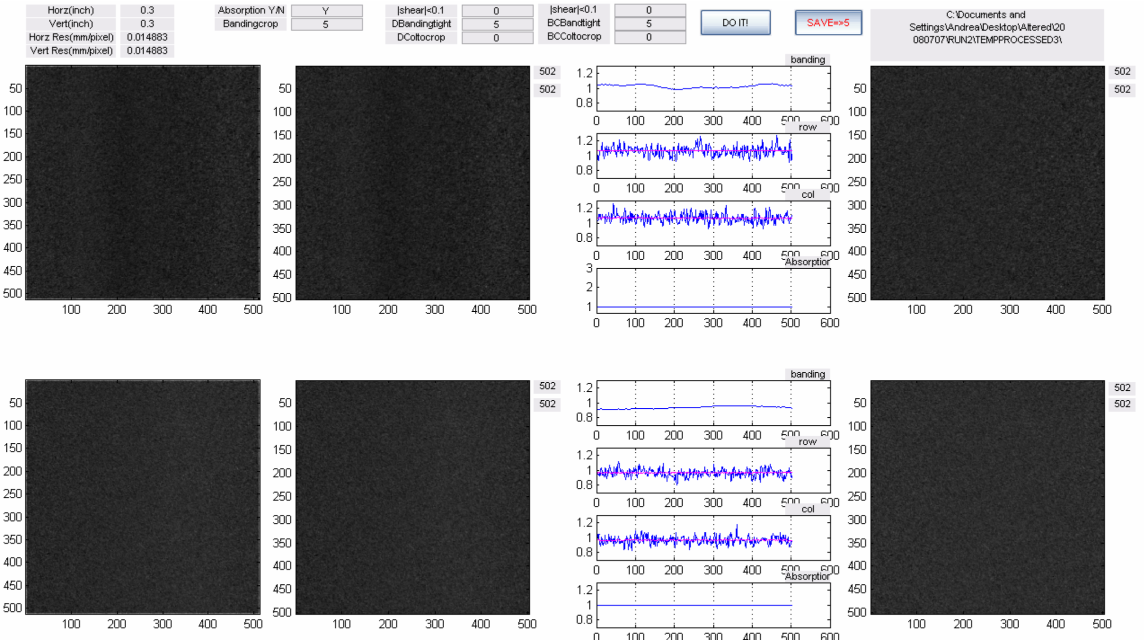
Sigma 15

Vertres(mm) 0.074414

Horzres(mm) 0.074414

Fig. 3.30 Image processing stage 3

In the fourth step of image processing (Fig. 3.31), any remaining instantaneous fluctuations in laser beam inhomogeneity were corrected. This stage provided the user with image shearing options to make the banding structures vertical in preparation for the banding correction. The banding correction was performed as described in previous sections, by integrating the signal down each column and normalizing the image across the columns by a single, smoothed correction function. The banding tightness was controlled in this panel, as well as the option to apply the absorption correction. The resulting image is shown in the right column. As seen in Fig. 3.31, the correction resulted in a pair of very uniform fluorescence image.

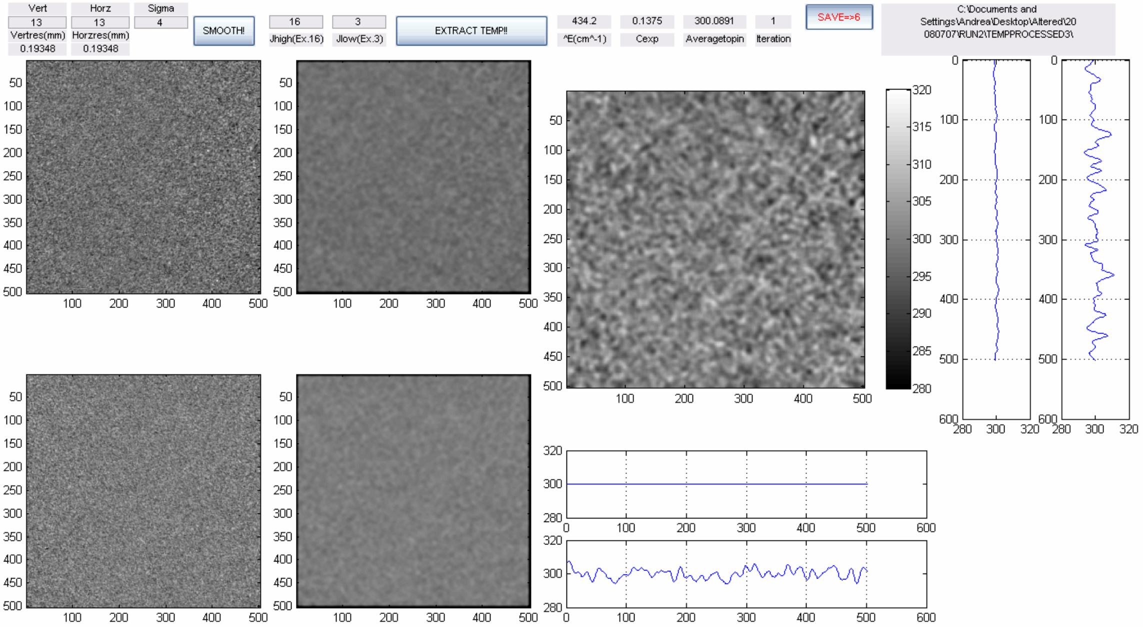


Horz(inch)	0.3
Vert(inch)	0.3
Horz Res(mm/pixel)	0.014883
Vert Res(mm/pixel)	0.014883
Absorption Y/N	Y
Bandingcrop	5
|shear|<0.1	0
DBandingtight	5
DColtocrop	0
|shear|<0.1	0
BCBandtight	5
BCColtocrop	0

C:\Documents and Settings\Andrea\Desktop\Altered\20080707\RUN2\TEMPPROCESSED3\

Fig. 3.31 Image processing stage 4

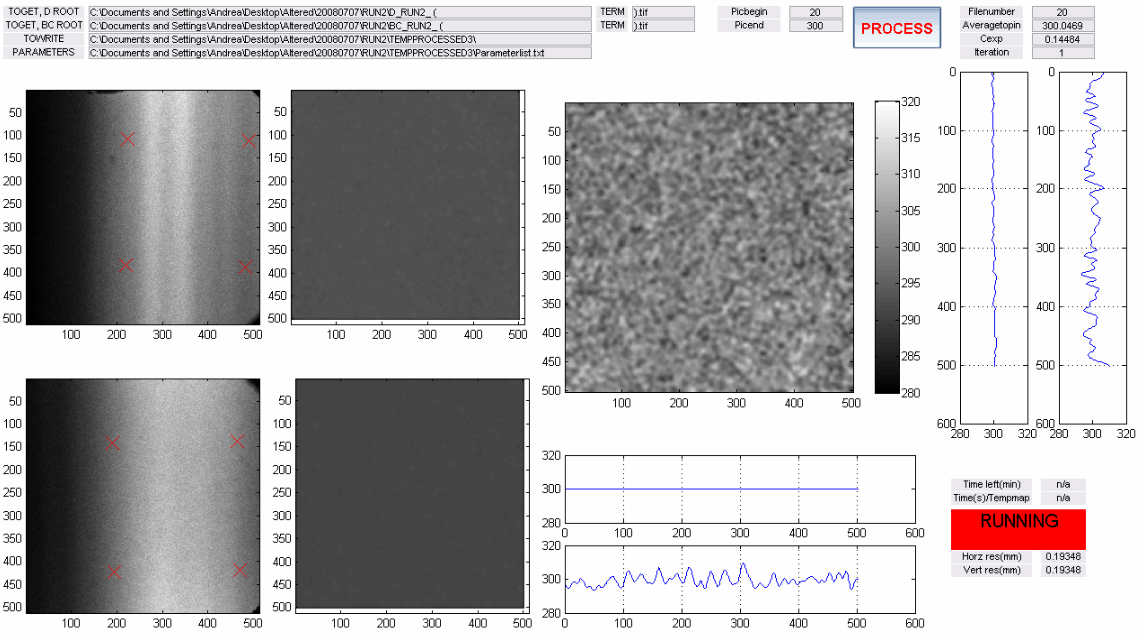
In the fifth step of image processing (Fig. 3.32), the images were smoothed, and the temperature was calculated by iterating on the calibration constant, Cexp, until the average in the center portion of the temperature map was within the bounds of 300 +/- 0.1 K. The images resulting from stage 4 are shown in the left column, and the smoothed images are shown in the second column. The images here were smoothed to produce resolutions on the order of the laser sheet thickness (~200 um). In practice, the images may be smoothed less or more depending on the size of turbulent structures (i.e. Temperature fluctuations) to be resolved. The large image in Fig. 3.32 is the calculated temperature map. The dark stripe on the right of the image is a false element caused by smoothing of the mean and instantaneous images and should be disregarded. The first plot in Fig. 3.32 underneath the temperature map is the temperature profile in the horizontal direction (averaging down the columns), and the second plot are the temperature values of a single row. The two plots on the right of the temperature map are the corresponding plots in the vertical direction. By examining the temperature map and the plots of a single row and column, it was seen that the image processing schemes were successful in producing measurement temperature uncertainties on the order of 1-2%, making true temperature fluctuations of 3% or greater easily visible.



Vert: 13
Horz: 13
Sigma: 4
Vertres(mm): 0.19348
Horzres(mm): 0.19348
Jhigh(Ex.16): 16
Jlow(Ex.3): 3
^E(cm^-1): 434.2
Cexp: 0.1375
Averagetopin: 300.0891
Iteration: 1
C:\Documents and Settings\Andrea\Desktop\Altered\20080707\RUN2\TEMPPROCESSED3\

Fig. 3.32 Image processing stage 5

In the sixth and last step of image processing (Fig. 3.33), a mass number of images were processed. All of the parameters used in stages 1-5 were stored and written to a configuration file, for use in every subsequent image pair. The layout of this stage is similar to stage 5, and the program indicates status and processing time remaining. This step is shown in Fig. 3.33.



TOGET, D ROOT C:\Documents and Settings\Andrea\Desktop\Altered\20080707\RUN2\D_RUN2_(TERM).tif

TOGET, BC ROOT C:\Documents and Settings\Andrea\Desktop\Altered\20080707\RUN2\BC_RUN2_(TERM).tif

TOWRITE C:\Documents and Settings\Andrea\Desktop\Altered\20080707\RUN2\TEMPPROCESSED3\

PARAMETERS C:\Documents and Settings\Andrea\Desktop\Altered\20080707\RUN2\TEMPPROCESSED3\Parameterlist.txt

Picbegin 20

Picend 300

Filenumber 20

Averagetopin 300.0469

Cexp 0.14484

Iteration 1

Time left(min) n/a

Time(s)/Tempmap n/a

RUNNING

Horz res(mm) 0.19348

Vert res(mm) 0.19348

Fig. 3.33 Image processing stage 6

G.3.2 Experimental results and discussion

The resulting maps of $\overline{T}$ and T_{rms} for the 2x2 binned 280 fluorescence image pairs are shown in Fig. 3.34. The process was repeated for images from a 4x4 run (on-board binning, not software binned) at identical conditions and yielded the maps in Fig. 3.35. The 4x4 images were smoothed at kernels which equaled ½ of the 2x2 run to give equal resolutions for both cases.

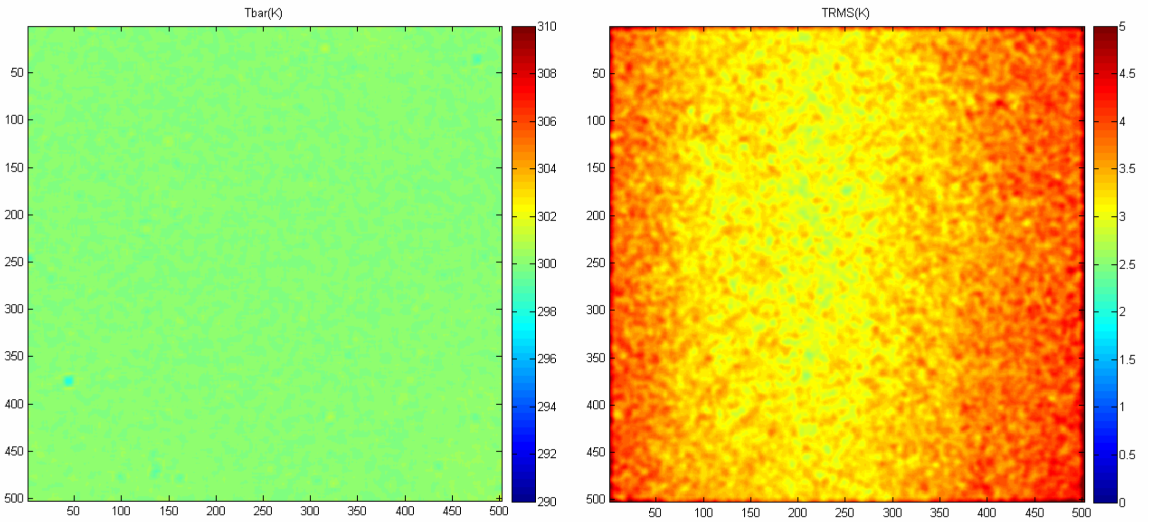


Fig. 3.34 Experimental $\overline{T}$ and T_{rms} maps (On board binning = 2x2 pixels)

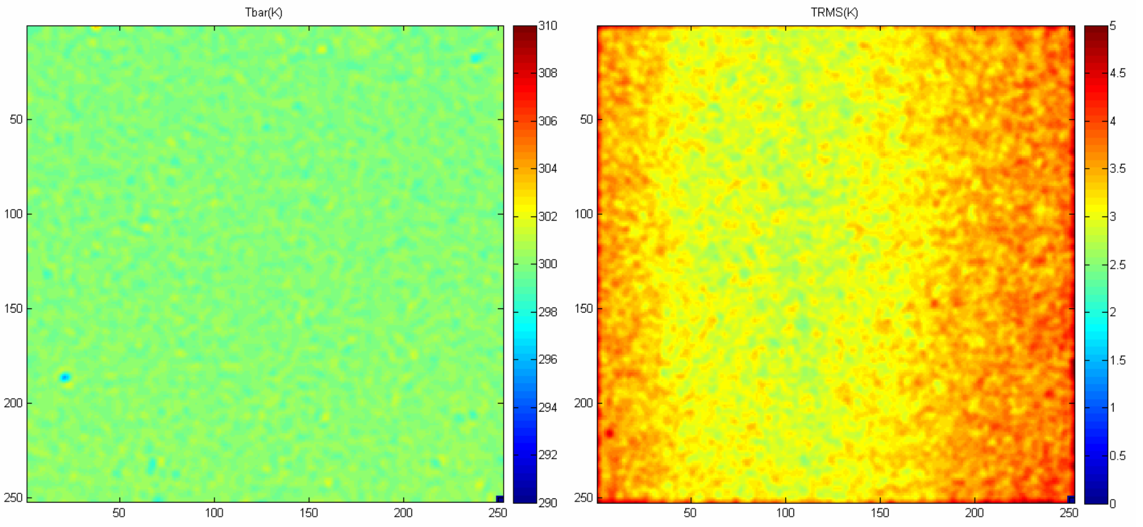


Fig. 3.35 Experimental $\overline{T}$ and T_{rms} maps (On board binning = 4x4 pixels)

The plot of $\overline{T}$ showed a very uniform, 300K map, as expected. The T_{rms} map showed that the image processing was successful in producing measurement uncertainties of <5K, or <2%. In addition, the uncertainties reflected the laser power distribution; uncertainties were lowest (<1%) in areas of maximum laser power, and greatest (~1.5%) near the edges of the field of view (where the laser power decreased to about 2/3 of the maximum power). As can be seen from the comparison between the 4x4 and 2x2 cases, binning 4x4 gave a slightly lower T_{rms} although the difference is very small, less than 1%. In terms of computing time, the 2x2 case took about 22 hours to run on a 1 GB RAM, 1.66 GHz Intel CPU Processor, Windows XP system, and only approximately 3 hours to run the 4x4 case. The limiting step during image processing was time spent warping the images. For the 2x2 case, the size of the warping matrix was approximately 80,000 points, while for the 4x4 case, the size was approximately 20,000 points for a field of view of 0.3"x 0.3". Therefore, the user may opt for the 4x4 case to reduce processing time.

In summary, the application of PLIF towards flowfields with temperature uncertainties of a few percent (300 +/- 10K) was explored. The effect of commonly used image processing procedures such as banding tightness, Gaussian smoothing, and rotational line pair selection were explored via a computer generated temperature map. It was concluded that a small banding tightness value, combined with Gaussian smoothing with a kernel on the order of the size of temperature fluctuation structures to be resolved, and a rotational line pair selection of J"=3.5 and 16.5 were appropriate starting conditions. These conditions were applied towards an experimental, uniform air flow seeded with 1% NO. Resulting maps of $\overline{T}$ and T_{rms} supported the image processing conclusions and gave T_{rms}<1.5%, or T_{rms}<5K making resolution of true temperature of 300 +/- 10K (or greater) possible. Comparison of on-board binning showed that binning 4x4 gave only slightly better T_{rms} than the 2x2 case, although binning 4x4 greatly reduced image processing time.

CHAPTER IV

THE EXPERIMENTS: CHARACTERIZATION OF THE RF PLASMA

The RF plasma in the subsonic facility was characterized for the following properties: 1) species identification using broadband emission spectroscopy, 2) rotational and vibrational temperature (of the electronically excited states) of N_2 using narrowband emission spectroscopy, 3) vibrational temperature of the electronic ground state of N_2 using CARS with comparison vibrational kinetics modeling. It should be noted that all three of these techniques were not temporally resolved. That is, they represent temporally averaged values. However, the plasma was spatially and temporally stable during all of these measurements, as determined from single-shot images from the ICCD cameras and visual assessment of the uniformity of the plasma through a variety of filters. Also, the emission spectra were rich in signal, and usually only required a few seconds of integration time and so the effect any small temporal variation of the plasma uniformity were minimized. A photograph of the RF plasma at about 100 W forward power (FP) in air with emission collection is shown in Fig. 4.1:

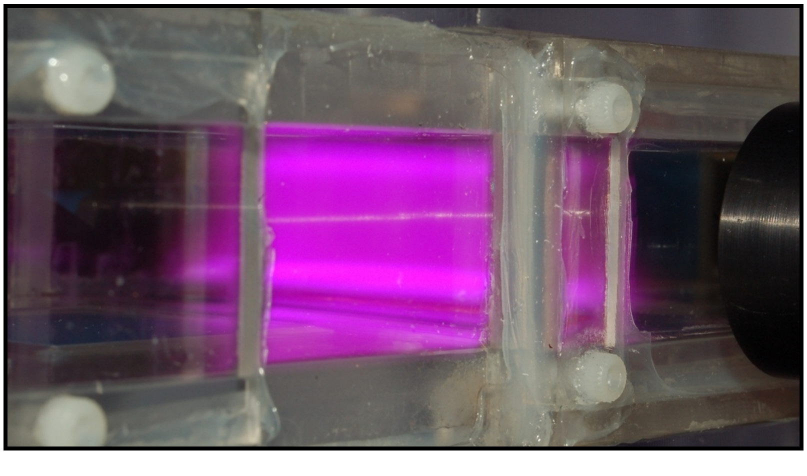


Fig. 4.1 Digital photograph of RF plasma discharge in air, FP = 100 W

The broadband emission spectra were taken via a fiber-optically coupled Oriel spectrometer (MS 125 1/8m with 2048 pixel linear CCD, 600 line/mm grating, 10 um slit, spectral range 200-1100 nm, resolution ~0.5 nm) adjusted to probe the central region of the plasma. A broadband emission spectrum is shown in Fig. 4.2 with room air (red), and room air seeded with NO (blue). These spectra were taken at about 100 W of RF FP at static pressures of 50 Torr within the test section.

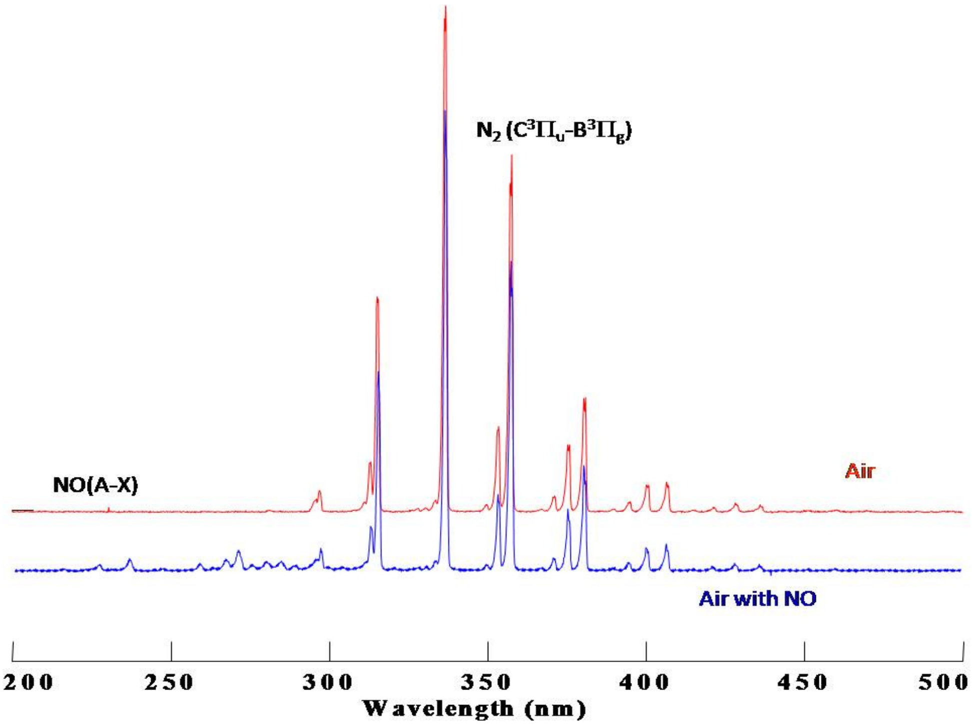


Fig. 4.2 Broadband emission spectra for species identification

The broadband emission spectra show several distinct features. First, the absence of emission from NO and OH in the unseeded air case, and of OH in the seeded case verify that the plasma was not dissociating and forming significant fractions of NO and OH. Second, the largest features of the spectra are the $N_2(C^3\prod_u\text{-}B^3\prod_g)$ emission bands from about 320 nm to 400 nm. These bands cause the plasma to appear purplish. To examine these features more carefully, high resolution spectra were taken of these regions using a fiber optically coupled (Ocean Optics) SPEX 1877 0.6 Triplemate (triple grating with 1800 line/mm final stage grating, resolution ~0.025 nm) adjusted to probe the central region of the plasma. An Andor back-illuminated EMCCD (Newton, DU-970-BV, water-cooled to -90 C) detected the emission. A typical spectrum is shown in Fig. 4.3, with the fit to theoretical spectrum calculated using custom software graciously provided by DeJoseph (2008):

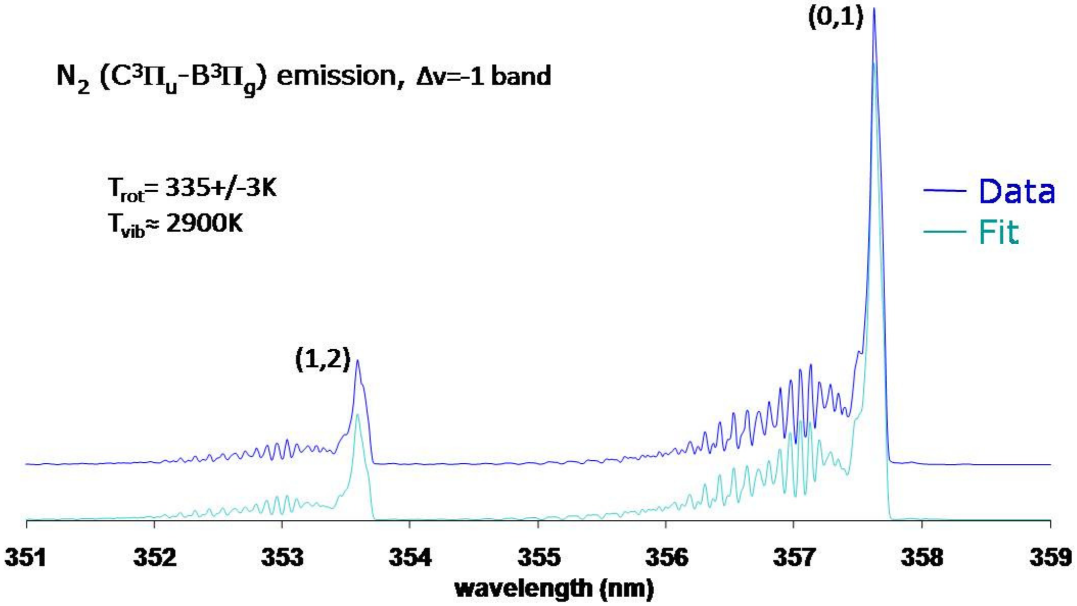


Fig. 4.3 Narrowband emission spectra of the $N_2(C^3\prod_u$-$B^3\prod_g)$, Δv= -1 band

The experimental data were fitted to the theoretical spectrum through minimization of residuals, to obtain very accurate (335 +/- 3 K) rotational temperatures. This value was obtained very accurately because of the high spectral resolution and signal to noise ratios, and could then be used in the NO PLIF image analysis to calibrate the pair of images to a bulk average temperature, since this value is known more accurately (<1% uncertainty) than the expected flow temperature fluctuations (3-10%). The vibrational temperature (2900 K) is interesting, but in this case, it is the vibrational temperature of the N_2 C state, and cannot be linked to the ground state vibrational temperature. Therefore, uncertainty analyses were not conducted on this value.

Since the vibrational distribution of the ground electronic state of N_2 cannot be obtained through emission spectroscopy, Coherent anti-Stokes Raman spectroscopy (CARS) in a planar boxcars configuration [e.g. see Eckbreth (1988)] was used to obtain these data directly. The CARS spectroscopy system consisted of a Nd:YAG laser (Spectraphysics PRO 290-10), which provided about 0.8 J/pulse at 532 nm. This beam was split using a beam splitter, and about 200 mJ/pulse was used to pump a PDL2 dye laser. The dye laser was modified by replacing the grating with a visible mirror to produce the Stokes broadband radiation. The output power was about 30 mJ/pulse at a center wavelength of 606 nm. The remaining 532 nm energy was split again and used as the pump beam. The 532 nm radiation was guided to the test section using high-power laser mirrors (Lattice Electro-Optics). The broadband red radiation was guided using standard high-reflectivity visible mirrors (Thorlabs). A delay line was installed using 45 degree and 0 degree incidence 532 nm mirrors to account for the increased path length through the dye laser, and was adjusted by using a fast photodiode (1 ns rise time) to verify that all three beams passed through the test section simultaneously. In practice, the 532 nm beam intensity was decreased (by about half) to avoid breakdown and window damage to the tunnel. The three beams were combined using a 532 nm mirror which served as a dichroic (through which the red broadband beams was passed, and the 532 nm beam was reflected). A 100 mm EFL spherical lens was used to focus the three beams into the test section. An additional 100 mm EFL spherical lens was used to collimate the CARS signal.

A schematic showing the layout of the laser system is given in Fig. 4.4 (top). A closeup of the CARS system in the vicinity of the test section is given in Fig. 4.4 (bottom). After the CARS signal was generated and collimated, it was filtered using either a 532 nm notch filter (Semrock), or a 500 nm shortpass filter (Thorlabs). The filtered signal was fed into a fiber optic collection assembly, which was routed to the spectrometer. The output of the fiber optic was f# matched to the spectrometer (f/4) using spherical optics. The transmission of the fiber optic was about 90%. For further information on the boxcars configuration, the reader is directed towards Eckbreth (1988), an excellent source for CARS spectroscopy.

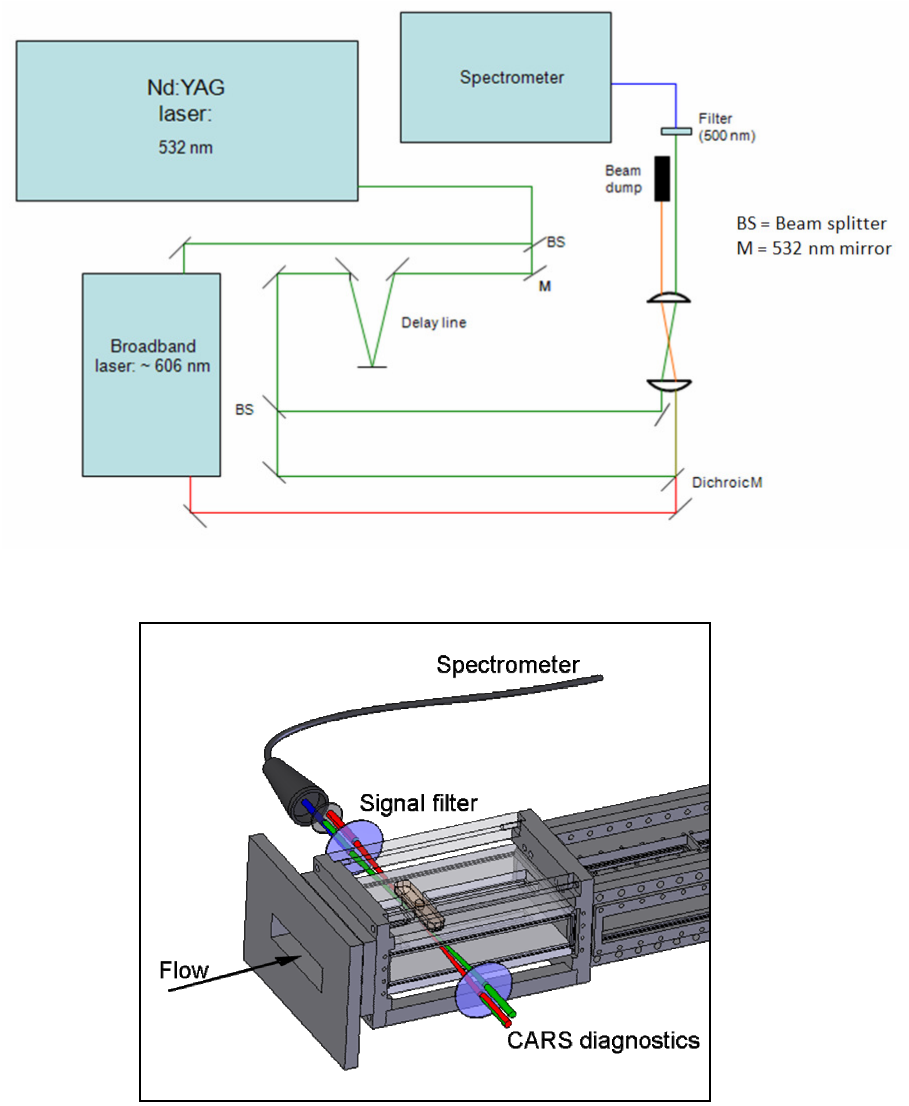


Fig. 4.4 Experimental N_2 CARS layout

The configuration for the locations of each vibrational temperature measurement relative to the plasma is shown in the schematic in Fig. 4.5, along with experimental values of vibrational temperature in the streamwise direction and fits using the chemical kinetics model described previously:

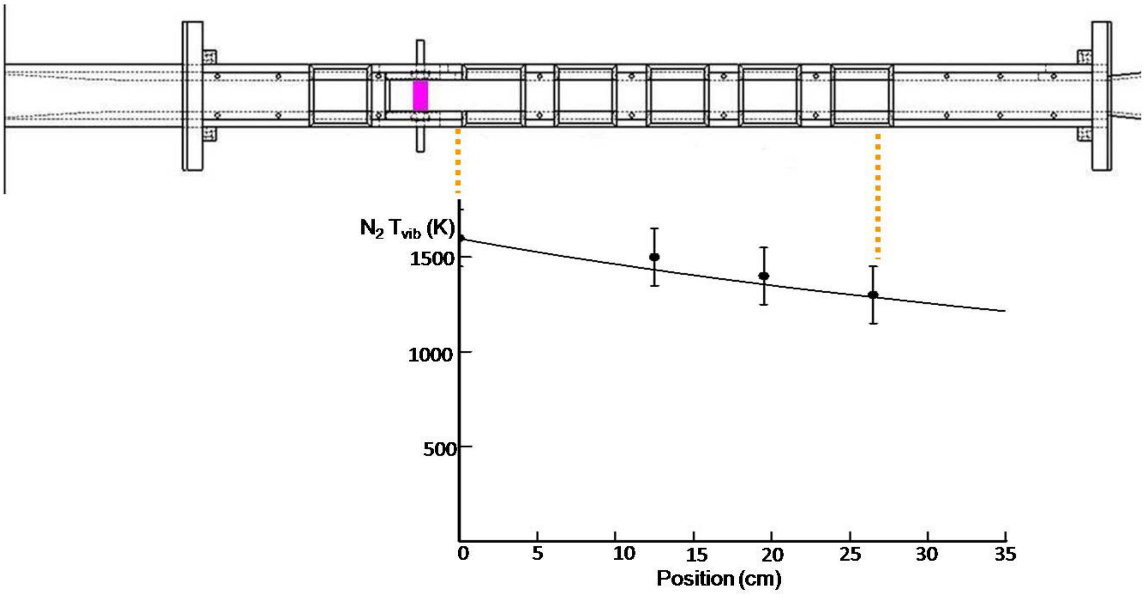


Fig. 4.5 CARS schematic for direct measurement of N_2 vibrational temperature

The values of vibrational temperature were obtained by fitting the experimental spectra using CARSFIT, developed at Sandia National Laboratories [Palmer (1989)], modified for non-equilibrium flows where different rotational and vibrational temperatures could be specified. The rotational temperature was held at 300K for these simulations, while the vibrational temperature was adjusted to fit experimental data. The CARS data were taken at several locations directly after the plasma and up to about 25 cm downstream of the plasma. The first location was about 1 inch downstream of the plasma. The comparison between the vibrational kinetics model (solid line) and the experimental data (points) is good, and serves to validate the kinetics model. The initial starting vibrational temperature of N_2 for the kinetics modeling was set at 1600 K, which was consistent with both the experimental CARS results and calculations estimated from electron collisions cross sections [e.g. Trevisan (2005), Mojarrabi (1995), Higgins (1995), Brennan (1992), Itikawa (1989 and 1986)] and energies estimated from similar experimental conditions using multiple sources of which several examples include Raizer (1995), Chintala (2004), and Palm (2003)]. From these sources, the average electron energy is about 1 eV. Also, according to Palm (2003) and Raizer (1995), for air plasmas at similar experimental conditions, about 90% of the electron energy is transferred to the vibrational mode of N_2, whereas O_2 is only mildly vibrationally excited. As mentioned before, this leads to a thermal nonequilibrium between the different species, where the vibrational temperatures for N_2, O_2 , and NO were expected to have plasma-induced vibrational temperatures of 1700, 800, and 2000 K..

Since the agreement between experiment and theory was good, the model was used to examine the mechanisms dominating the vibrational decay. The model predicts that the vibrational temperature decay for both N_2 and O_2 are driven mainly by V-V and V-T transitions with H_2O room humidity (set at 40% in these simulations). Experimentally, no vibrationally excited O_2 was detected at the points plotted for N_2. O_2 has a much more rapid rate of relaxation through H_2O and suggests that CARS measurements should be taken closer to the plasma discharge to detect vibrationally excited O_2.

In summary, from the broadband and narrowband emission data, it was seen that the plasma imparts only slight gas heating (35 +/- 3 K) with negligible molecular dissociation. The experimental vibrational temperature of N_2 directly after the plasma was in good agreement with calculations using electron cross sections and experiments in the literature run at similar experimental conditions. This value was used as a starting value for the kinetics modeling. Pointwise experimental streamwise measurements of vibrational temperature of N_2 yielded good agreement with theoretical kinetics simulation results, and suggest that the driving relaxation mechanisms for vibrational relaxation of N_2 and O_2 are V-V and V-T energy exchanges with ambient H_2O. Current efforts are focused towards installation of a dual-pump CARS system to probe both N_2 and O_2 vibrational temperatures concurrently [Lucht (1987) and (2003)].

CHAPTER V

THE EXPERIMENTS: PHOTODISSOCIATION STUDIES 1, SINGLE COMPONENT MOLECULAR TAGGING VELOCIMETRY AND VIBRATIONAL TEMPERATURE IMAGING IN AN UNDEREXPANDED JET FLOWFIELD UTILIZING NO_2 PHOTODISSOCIATION AND NO PLIF

A. METHOD

A.1 Diagnostics Technique and Background

The present MTV technique makes use of the fact that the photodissociation of NO_2 produces both ground and excited vibrational states of NO,

$$\begin{aligned} NO_2 + hv_{355nm} &\longrightarrow NO_{v=0} + O(3P) \\ &\longrightarrow NO_{v=1} + O(3P) \end{aligned} \qquad \text{(Eq. 5-1)}$$

with an experimentally determined branching ratio of $NO_{v=1}$:$NO_{v=0}$ of 41.2 ± 6.2:58.8 ± 8.8 [e.g., see Hunter (1993), Brookes (2007), Harrison (1994)]. This ratio corresponds to a 2-level Boltzmann vibrational temperature of approximately T_{vib} = 7000K. Therefore, photodissociation of NO_2 at 355 nm produces a spatially localized non-thermal equilibrium population distribution of NO. If the populations of the two vibrational levels of NO can be probed simultaneously using NO planar laser-induced fluorescence (PLIF) the vibrational temperature decay of NO can be temporally tracked. The fluorescence signal intensity is a function of the initial population of the probed state, the ratio of the stimulated emission Einstein coefficients, the saturation and laser intensity, the spontaneous emission Einstein coefficient, and an overall efficiency term which can be obtained via a calibration point. The ratio of two simultaneous fluorescence signals in which identical rotational states are probed in two different vibrational states yields the vibrational temperature. Then, by imaging the NO fluorescence onto ICCD sensors, a vibrational temperature map can be determined by performing pixel-by-pixel calculation of the vibrational temperature. On the other hand, velocimetry can be performed by tracking the movement of the NO molecules in time, and is simpler than the extraction of the vibrational temperature since only a single vibrational state (either $NO_{v=0}$ or $NO_{v=1}$) is required. Advantages and disadvantages of probing $NO_{v=1}$ or $NO_{v=0}$ in temperatures ranging from below 300K to 20000K will be discussed. The experiments were carried out at the National Aerothermochemistry Laboratory at Texas A&M University.

B. EXPERIMENTAL FLOWFIELD

The flowfield that was examined for demonstration of the combined vibrational temperature and velocity mapping was the axially symmetric highly underexpanded jet. The terminology that will be used in this study was referenced from Woodmansee (2004). Briefly, underexpanded jets are broken down into two categories, moderately underexpanded and highly underexpanded jets. Moderately underexpanded jets possess a shock diamond structure caused by oblique shock waves, while highly underexpanded jets possess a Mach disk. The transition from subsonic to moderately underexpanded to highly underexpanded is a function of the jet pressure ratio (JPR), which is defined as the ratio of the exit pressure to ambient pressure, P_e/P_a. Generally, jets with JPRs>2 are defined as highly underexpanded. The structure of the highly underexpanded jet is well known [e.g., see Donaldson (1971)] and relevant structures to this study are labeled in Fig. 5.1.

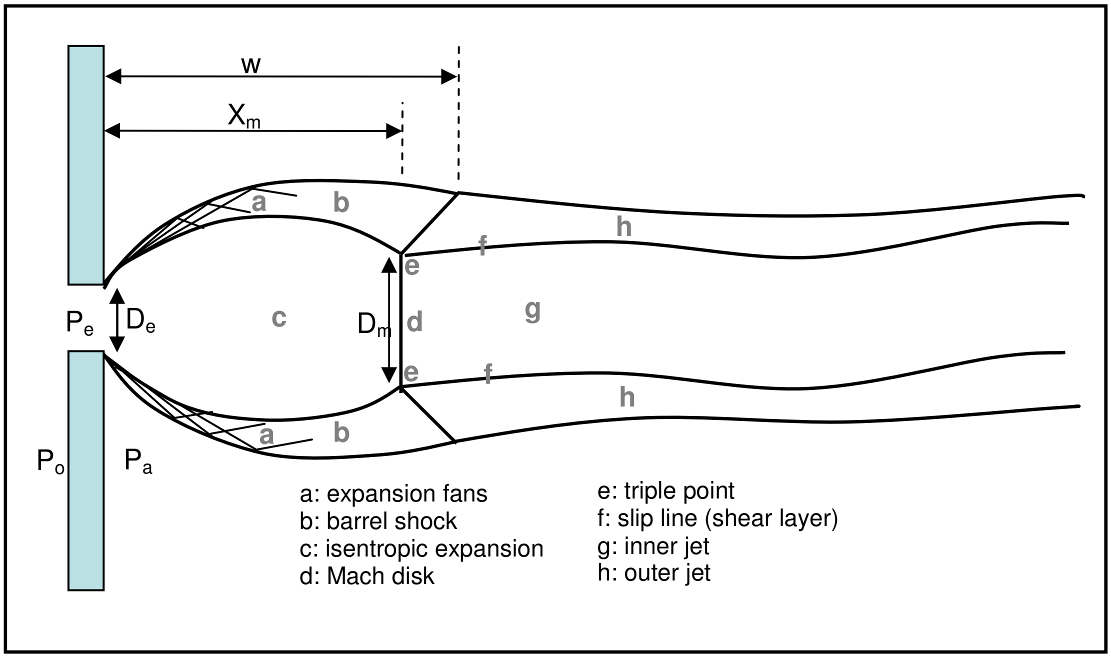


Fig. 5.1 Relevant structures associated with a highly underexpanded jet flowfield

The symbols in the figure are defined in the Appendix. At the nozzle exit, Prandtl-Meyer expansion fans give rise to the barrel shock structure that surrounds the inner isentropic expansion region. The expansion in this region is terminated by a Mach disk which recompresses the flow. The intersection of the barrel shock and Mach disk is known as the Mach disk triple point and this point gives rise to a slip line between the subsonic core (inner jet) and supersonic region (outer jet) which develops into a shear layer. There also exists a shear layer between the outer jet and ambient, nearly stagnant fluid. As the fluid travels past the plane of the Mach disk, a series of oblique shocks recompress the flow, but there are no further normal shocks for very highly underexpanded jets [e.g., see Donaldson (1971)]. These oblique shocks cause weak oscillations in the streamwise velocity and temperature.

The structure of underexpanded jets are usually classified by the distance to the Mach disk (X_m), diameter of the Mach disk (D_m), and primary wavelength (w) which have all been found to be a function of the JPR in Wilkes (2006). Underexpanded jets can be further classified as laminar, transitional, and turbulent underexpanded jets by defining a Reynolds number which takes into account the JPR in Wilkes (2005). Transition is caused by instabilities within the shear layer that develop between the inner and outer regions, giving rise to fluid mixing between these two regions as the flow becomes more and more turbulent. By using this Reynolds number, the distance from the nozzle exit to transition and also to turbulence can be calculated. Recent studies of the underexpanded jet have focused on measurements of simultaneous temperature and velocity using NO PLIF [e.g., see Palmer (1991)], simultaneous pressure and temperature measurements using N_2 CARS [see Woodmansee (2004)] and simultaneous pressure, velocity, and temperature measurements using high spectral resolution NO PLIF [see Kulatilaka (2008)]. Numerical studies have focused primarily on a comparison of the measured properties of underexpanded jet structure with CFD [e.g., see Wilkes (2006)].

C. Experimental Setup and CFD Simulation

C.1 Experimental Setup

The goals of the experiments were two-fold and were both conducted in the vacuum chamber with similar setups. The vacuum chamber was fitted with a stainless steel nozzle to create the highly underexpanded jet flowfield, quartz windows on four sides to allow optical access, and a vacuum line. The schematic layout of the setup with a closeup of the field of view is shown in Fig. 5.2.

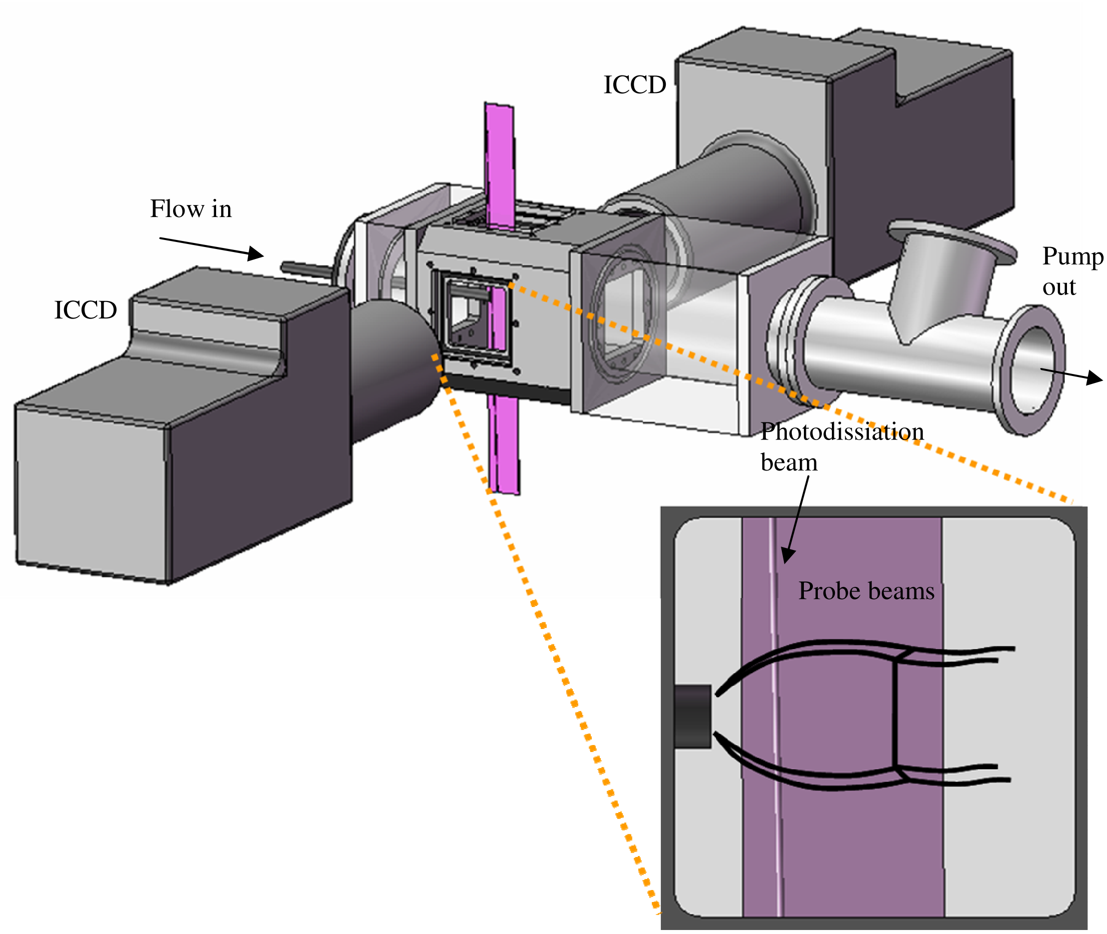


Fig. 5.2 Schematic of experiment and closeup of field of view

The two experiments conducted in the setup above are listed here:

C.2 Experiment 1: Single-Component Velocimetry

The first experiment is a velocimetry study utilizing NO_2 photodissociation. First, the photodissociation “write” laser (Spectraphysics PRO 290-10, 355 nm at 10 Hz) was focused into a beam and generated a column of local non-thermal equilibrium distribution of $NO_{v=1}$ and $NO_{v=0}$. Molecular tagging velocimetry requires a pair of images: an initial image, and a time delayed image. At 100 ns after photodissociation, the first probe laser sheet (355 nm- pumped using a Spectraphysics LAB 150-10 Nd:YAG at 10 Hz) Sirah Cobrastretch dye laser read the initial location of $NO_{v=1}$

molecules by fluorescence imaging onto an ICCD camera (Andor iStar ICCD, model DH734, 16-bit, fitted with a Nikon 105mm F/4.5 UV lens and extension rings for close-up viewing). No external filters were used since scatter was virtually nonexistent (through time delaying the ICCD cameras by about 10-20 ns after the arrival of the probe laser pulse). The second image was taken 500 ns after photodissociation by a second probe laser and ICCD camera, also imaging $NO_{v=1}$. The $NO_{v=1}$ was then tracked and its displacement determined and divided by the known time delay to give the streamwise velocity. The rotational states probed for the first and second probe lasers were $R_1+Q_{21}(1.5)$ and the group of lines which form the bandhead surrounding Q_1 (1.5), respectively in the $A^2\Sigma^+_{1/2}(v'=1) \leftarrow X^2\Pi_{1/2}(v''=1)$ transition. The experiment was then repeated, except $NO_{v=0}$ was probed instead of $NO_{v=1}$. The rotational states of $NO_{v=0}$ probed were $R_1+Q_{21}(2.5)$ for both probe lasers in the $A^2\Sigma^+_{1/2}(v'=0) \leftarrow X^2\Pi_{1/2}(v''=0)$ transition. The probed rotational states were located using the spectral simulation software LIFbase [Luque (1999)]. The photodissociation beam was focused using a 1m effective focal length (EFL) cylindrical lens into a thin sheet which was then cut on either side using a pair of facing razor blades to produce a sharp bar. The razor blades were less than 1 mm apart. The bar passed just adjacent to the probe beam sheeting optics, so that a slight angle in the photodissociation beam is visible. The power of the photodissociation beam was about 10 mJ/pulse at the test section, and the calculated NO_2 photodissociation fraction was greater than 99.9%. All three beams were carefully aligned in the test chamber and fine-tuned for maximum photodissociation (i.e. $NO_{v=1}$) signal. The short time delay (400 ns) was used in this case for increased spatial resolution. For this setup, the camera exposure was set at 50 ns and set at maximum aperture for both cameras. The two cameras were aligned at equivalent magnifications and focused to ensure that the images overlaid one another as closely as possible. Minor differences in magnifications were later corrected via image processing. The experiment was carried out at the conditions listed in Table 5.1:

Table 5.1 Experiment 1 run conditions

						Mole Fraction			
Run	Probed state	P_a (Torr)	P_0 (Torr)	JPR	T_o (K)	NO	N_2	NO_2	O_2
1	$NO_{v=0}$	4.8	450	49.53	300	0.009	0.921	0.07	0.00
2	$NO_{v=1}$	4.6	450	51.68	300	0.02	0.94	0.04	0.00

The nozzle construction was simple, and consisted of ¼” stainless steel sealed tubing with a 1 mm circular nozzle throat. The nozzle was carefully smoothed to remove burs and was slightly countersunk on the back face in order to minimize the effect of a building viscous boundary layer within the nozzle orifice. The chamber was continuously evacuated to maintain steady ambient pressures using a roots blower and backing pump with a combined pumping speed of ~250 cfm. The stagnation and ambient pressures were monitored using calibrated pressure transducers (MKS Series 902). The offset of the pressure transducers were <800 mTorr. The N_2 was high purity gas from Brazos Valley Welding, and the NO_2 was supplied by Sigma Aldrich (>99.5% pure). The NO was diluted using N_2 from a 10% NO/N_2 mix, supplied by Matheson Tri-Gas. The gases were delivered via needle valves into the chamber. The chamber was first stabilized using the dry air and NO_2 was introduced into the nozzle as close as possible to the testing chamber in order to prevent backflow and liquification of NO_2 into the dry air feed line. The NO_2 bottle was also gently heated to encourage flow into the system. The overall timing for the entire experiment was controlled by a digital delay generator (Berkeley Nucleonics Corporation, model DG565).

The goal of this experiment is to explore the advantages of probing $NO_{v=1}$ over $NO_{v=0}$ for velocimetry since the fluorescence signal collected from $NO_{v=1}$ would be immune to background NO (which would exist only in the ground state, v=0). This technique was carried out using both integrated (30 s) and single shot 1024 x 1024, full resolution images. The integrated images were taken by translating the photodissociation beam from the nozzle across the flowfield to past the Mach disk and taking initial and time delayed images at each location. 16 total locations were used. A statistical set of single shot images (630 images) at location 7 of 16 were taken for uncertainty analyses.

Sample raw images of Run 1 are shown in Fig. 5.3 and Fig. 5.4. The field of view is about 14 mm x 14 mm.

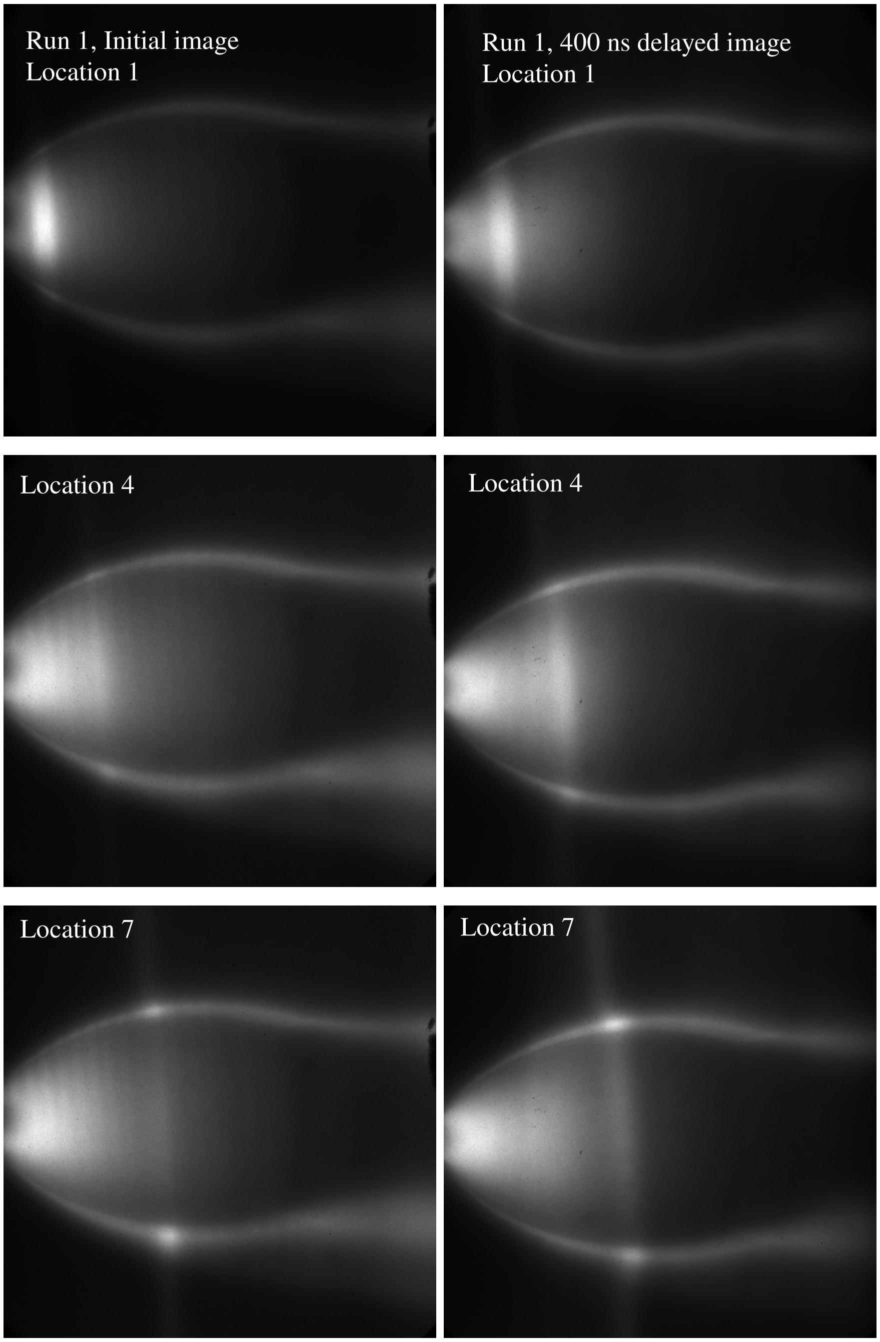


Fig. 5.3 Initial and time delayed integrated images (Run 1, probing NO$_{v=0}$), 1 of 2

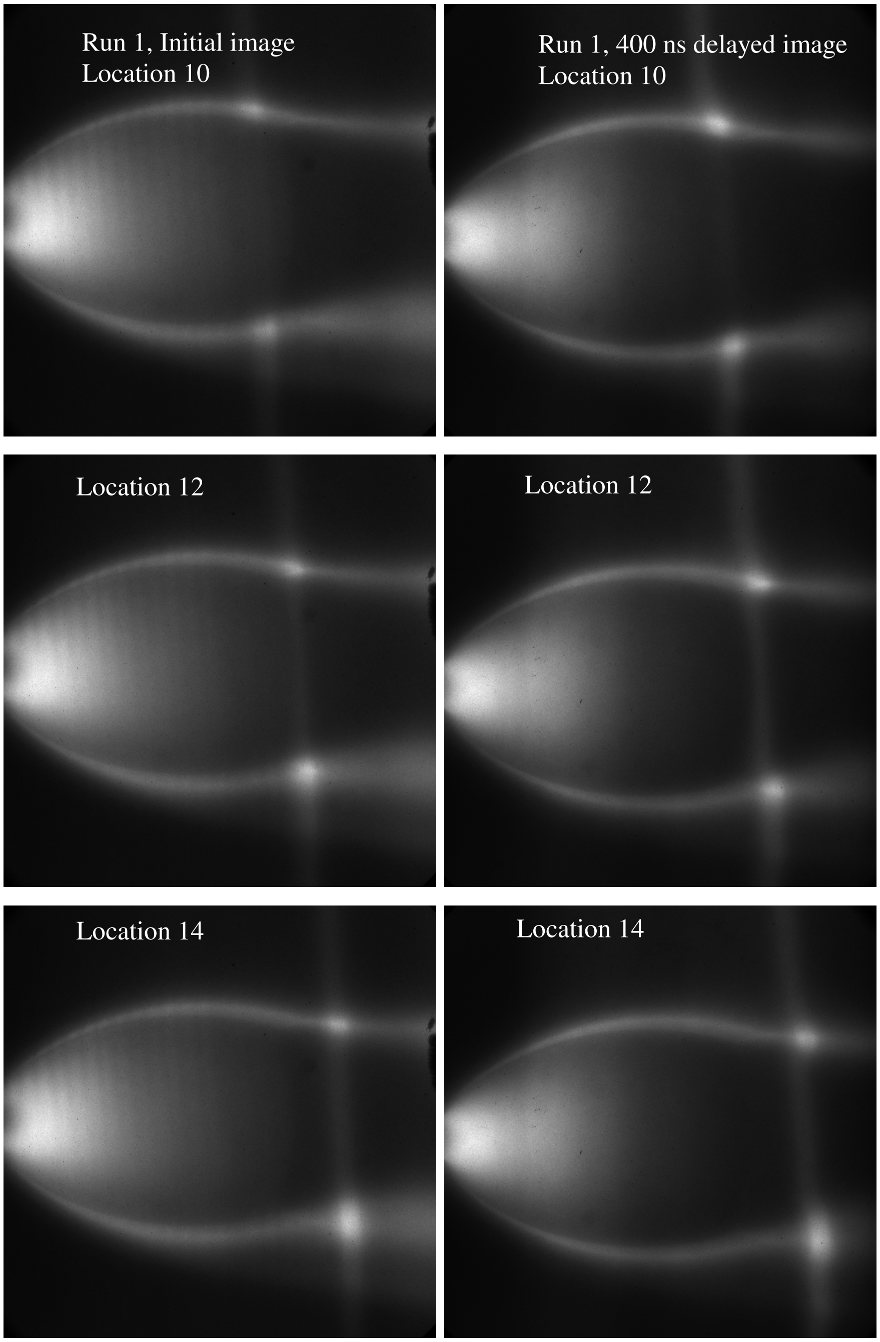


Fig. 5.4 Initial and time delayed integrated images (Run 1, probing $NO_{v=0}$), 2 of 2

A sample single shot image pair from Run 1 is shown in Fig. 5.5:

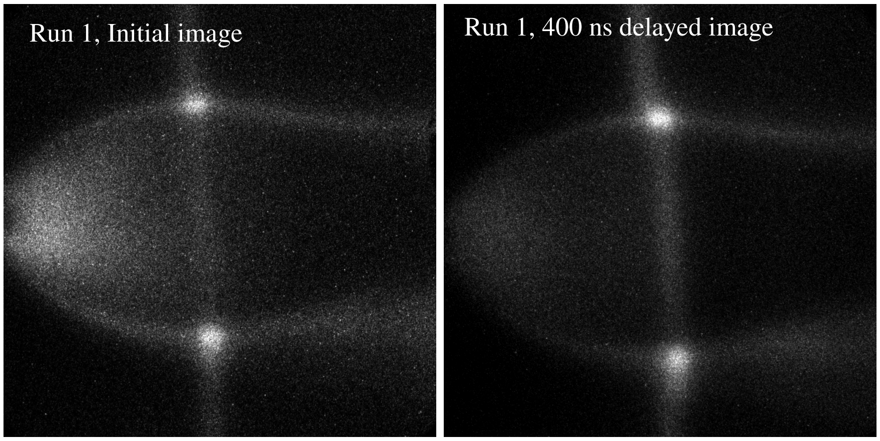


Fig. 5.5 Initial and time delayed single shot images (Run 1, probing $NO_{v=0}$)

Sample raw integrated images of Run 2 shown in Fig. 5.6 and Fig. 5.7. The field of view is the same as in Run 1.

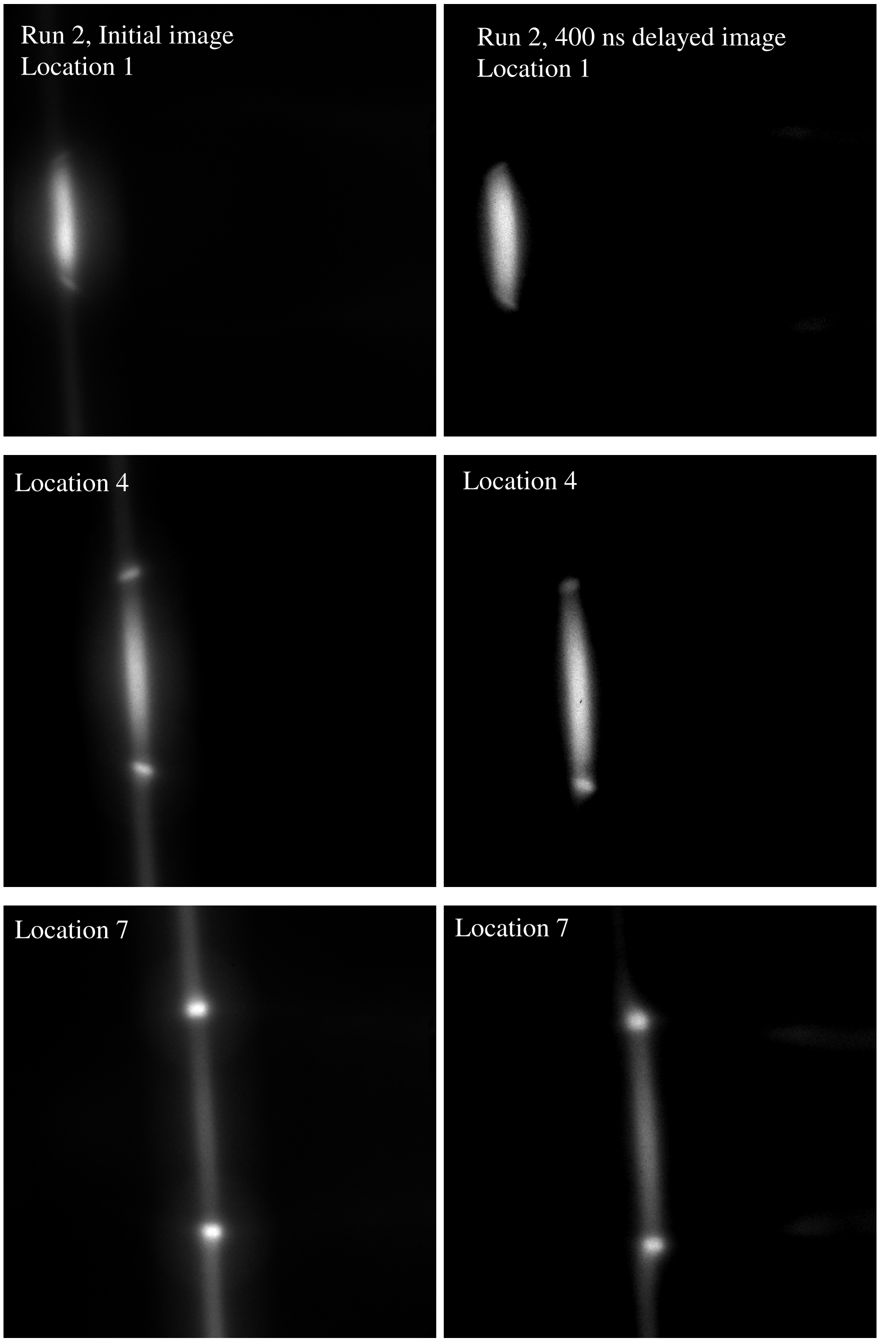


Fig. 5.6 Initial and time delayed integrated images (Run 2, probing NO$_{v=1}$), 1 of 2

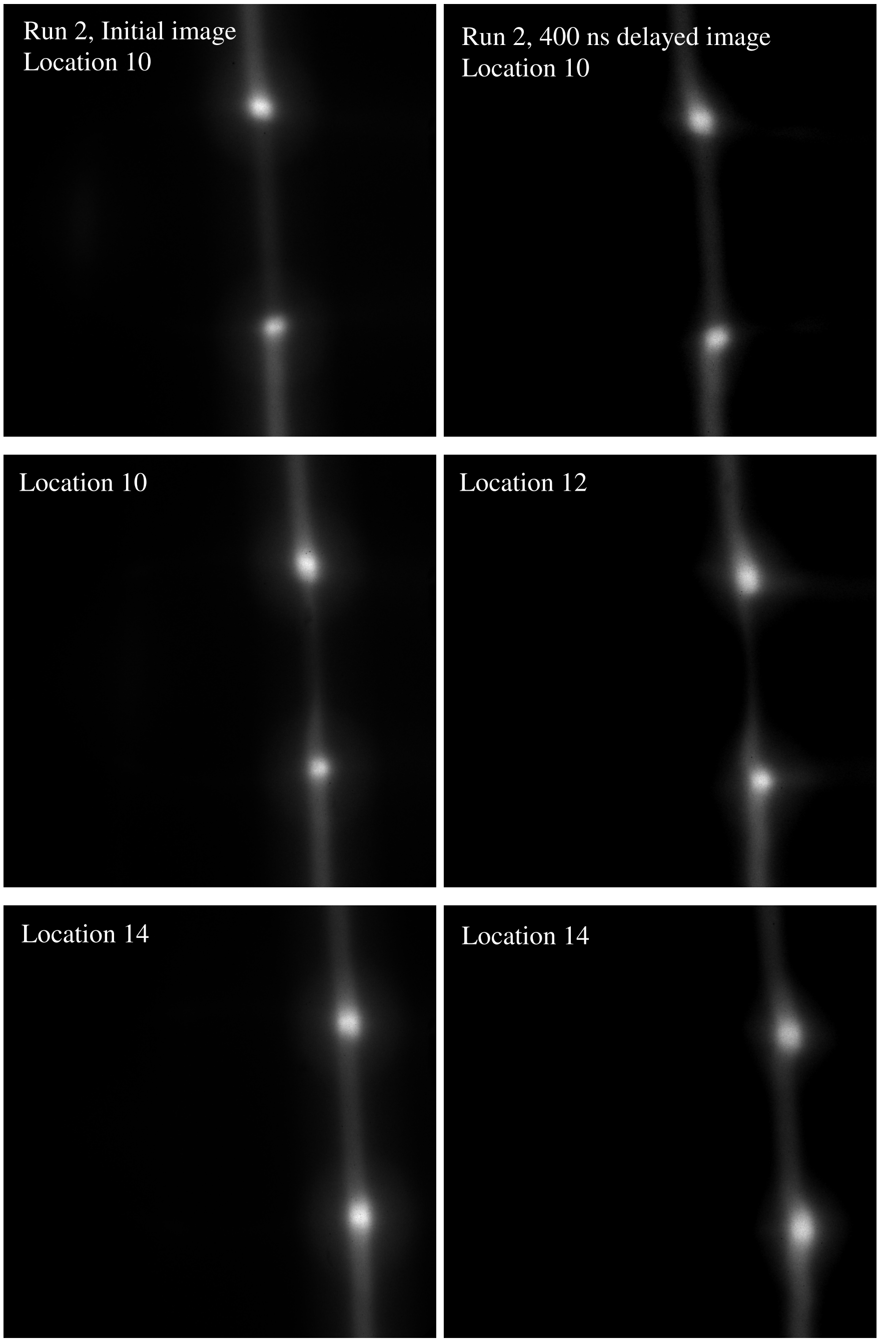


Fig. 5.7 Initial and time delayed integrated images (Run 2, probing NO$_{v=1}$), 2 of 2

A sample single shot image pair from Run 2 is shown in Fig. 5.8:

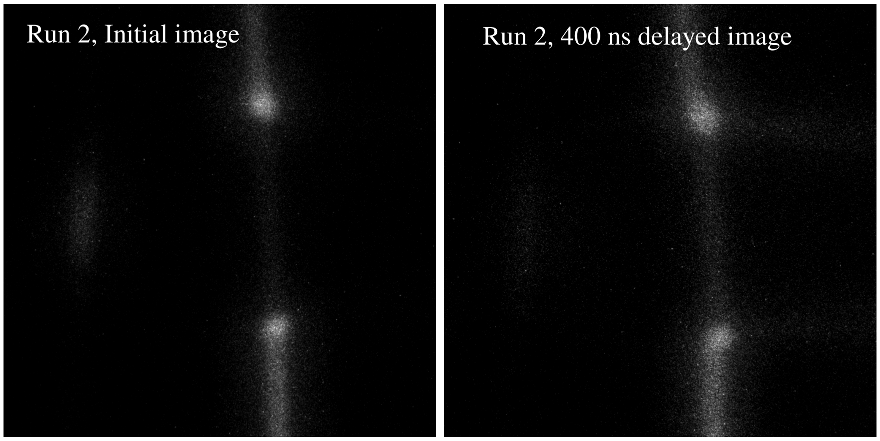


Fig. 5.8 Initial and time delayed single shot images (Run 2, probing $NO_{v=1}$)

Simply from visual assessment of the raw images, it is clear that the images probing $NO_{v=0}$ (Run 1), suffer from background NO fluorescence. The photodissociated NO is clear in both cases, although the $NO_{v=1}$ images in Run 2 provide much larger contrast and hence better signal to noise.

C.3 Experiment 2: Vibrational Temperature Decay

The second experiment involves exploration of the vibrational temperature decay by using long time delays. The goal was to map out the vibrational energy decay of the created local non-thermoequilibrium distribution of NO as a function of space and time in the flowfield. For this experiment, the time delay was varied in steps of μs (much longer than in Experiment 1) and the movement of the tagged NO molecules were tracked until the flow had moved out of the field of view. Instead of imaging only $NO_{v=1}$ with both cameras, one camera imaged $NO_{v=1}$ while the other imaged $NO_{v=0}$. Then, the vibrational temperature of NO could be calculated through the Boltzmann equation. integrated (30 s) and single shot image were both obtained for this case. The experimental setup was similar to that of the velocimetry setup. The air was a high-purity compressed dry air (0% hydrocarbons). The other gases used (NO, NO_2) were from the same suppliers as mentioned in Experiment 1. The experimental run conditions are listed in Table 5.2.

Table 5.2 Experiment 2 run conditions

P_a (Torr)	P_0 (Torr)	JPR	T_o (K)	Bulk gas flow (Torr)	NO_2 flow (Torr)	Re (m^{-1})
12.3	551.8	23.70	300	Air, 538.8	13	$1.67x10^6$

The experiment was repeated at a variety of JPR and NO_2 concentration conditions. However, the conditions listed above were chosen as the best demonstration of the technique because significant vibrational decay was observed within the field of view of the cameras while achieving excellent signal to noise levels (approximately 600:1 at initial time). The photodissociation laser was focused to a beam of <500 μm diameter inside the cell, corresponding to a laser fluence of ~ 5000 mJ/cm^2 giving a calculated NO_2 photodissociation fraction of ~99.9%.

The photodissociated $NO_{v=1}$ and $NO_{v=0}$ were then imaged at subsequent times after photodissociation in order to map out the vibrationally decaying flowfield. To probe the vibrationally decaying NO, two probe lasers were fired at various time delays after the photodissociation laser. The first and second probe laser systems were tuned to probe the $A^2\Sigma^+_{1/2}(v'=1) \leftarrow X^2\Pi_{1/2}(v''=1)$ transition of NO at R_1+Q_{21}(3.5) and the $A^2\Sigma^+_{1/2}(v'=0) \leftarrow X^2\Pi_{1/2}(v''=0)$ transition of NO at R_1+Q_{21}(3.5), respectively. The time delay between the photodissociation laser and the first probe laser will be referred to as the nominal time delay in subsequent sections. In order to stay within the linear fluorescence regime for NO, the output powers of the dye lasers were maintained at <2 mJ/pulse, which corresponded to 0.4-0.5 mJ/pulse at the chamber. Also, the method of calculating the NO vibrational temperature by taking the ratio of these temporally correlated fluorescence intensities is only valid if the rates of electronic quenching for $A^2\Sigma^+_{1/2}(v'=1)$ are the same as those for $A^2\Sigma^+_{1/2}(v'=0)$. According to Nee (2004), the rates for electronic quenching of NO $A^2\Sigma^+_{1/2}(v'=0)$ with O_2, the most significant collisional quenching partner to NO, has a rate of 1.51 x 10^{-10} cm^3/molecule/s. The rate for quenching of NO $A^2\Sigma^+_{1/2}(v'=1)$ with O_2 is 1.48 x 10^{-10} cm^3/molecule/s, a difference of a few percent.

It should be noted that the diameter of the photodissociation beam was comparable to that of the probe laser sheets in the field of view. The photodissociation laser beam was positioned in the post-shock region of the highly underexpanded jet, just beyond the Mach disk. The main reason for this choice lay in the fact that in the cold expansion region before the Mach disk, the vibrational temperature would freeze out and so would not be adequate for demonstrating vibrational relaxation. For this experiment, the ICCD cameras were externally triggered and gated for 100 ns. The resulting images were binned 2x2 for a final image size of 512x512 pixels and mapped to field of view of 18.4 x 18.4 mm^2.

Fig. 5.9 shows the location of the nascent NO in relation to the highly underexpanded jet flowfield at nominal delays of 0 and 12 μs. The images were constructed by overlaying images of the raw molecular tagging data with 1%NO in N_2 run at nearly the same stagnation and ambient pressures (554.8 Torr and 12.1 Torr) as the molecular tagging data. The 1%NO in N_2 images were corrected for laser sheet inhomogeneity (banding) across the image but no other corrections were applied. The nozzle is faintly visible on the left hand side of the images. We focused our analysis on specific regions of the flow because they had differing pressures. Listed in order of increasing pressure, the examined areas were the ambient region, centerline region, and the turbulent outer jet regions. These areas are labeled. The scaled Reynolds number was calculated using expressions found in Hunter (1993) and it has been assumed that the small amount of added NO_2 was negligible in affecting bulk gas properties.

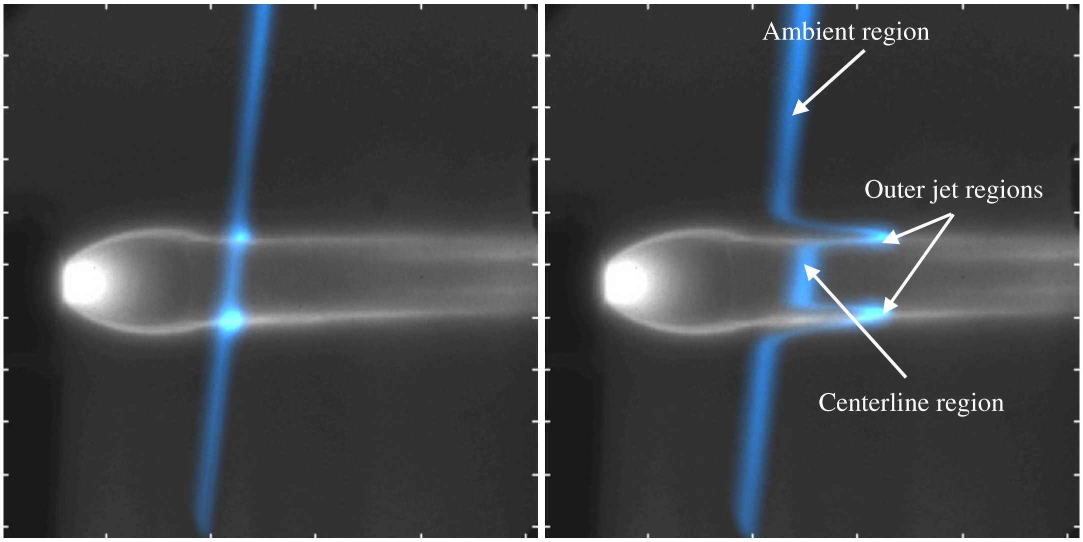


Fig. 5.9 Highly underexpanded jet flowfield: JPR= 23.7, Re=1.67x10^{6} m^{-1}, 2.4%NO$_2$ in dry air

The overall timing for the entire experiment was controlled by a digital delay generator (Berkeley Nucleonics Corporation, model DG565). Since the exact timing is important especially for the velocimetry analysis, it will be briefly described here. The photodissociation laser was triggered at T_0 at 10 Hz. The NO$_{v=1}$ probe laser was triggered at a later time, which was adjusted after each image set acquisition in order to create an entire set of sequentially delayed images. The first NO$_{v=1}$ image was taken at a delay of 50 ns to protect the ICCD's from the large scatter of the photodissociation pulse. Thus, the total delay on the first image taken (at nominal delay of 0 μs) was defined as 50 ns (initial delay) + 170 ns (programmed ICCD delay accounting for light path to chamber) + 35 ns (intrinsic ICCD insertion delay) + 100 ns (camera exposure)/2 + 10 ns (laser pulse duration)/2 for a total of 310 ns at a nominal delay of 0 μs. The first NO$_{v=0}$ image was taken at an additional 150 ns after the NO$_{v=1}$ image to allow the fluorescence from the first read laser pulse to decay before the second probe pulse. Images acquired of the second probe laser with its respective dye laser blocked showed that <1% of the signal from the first probe laser remained and was captured by the second ICCD camera. The movement of the tagged molecules during the 150 ns time delay was insignificant so that errors caused by blurring between the $NO_{v=1}$ and $NO_{v=0}$ image acquisition times were assumed negligible. This assumption was based on the fact that the average movement of the centerline region was ~6 pixels/μs, or about 200 μm; therefore, a 150 ns blurring effect would amount to less than one pixel of blurring. However, the faster moving outer jet regions of the flow experienced more significant uncertainties from the blurring effect.

Data were acquired at various nominal time delays from 0 μs to 200 μs and consisted of 62 image pairs. In addition, 40 single shot images at 0, 5, 10, 15, 20, 25, and 30 μs were taken for statistical error analyses. Sample raw (uncorrected) images of $NO_{v=0}$ and $NO_{v=1}$ are shown in Fig. 5.10.

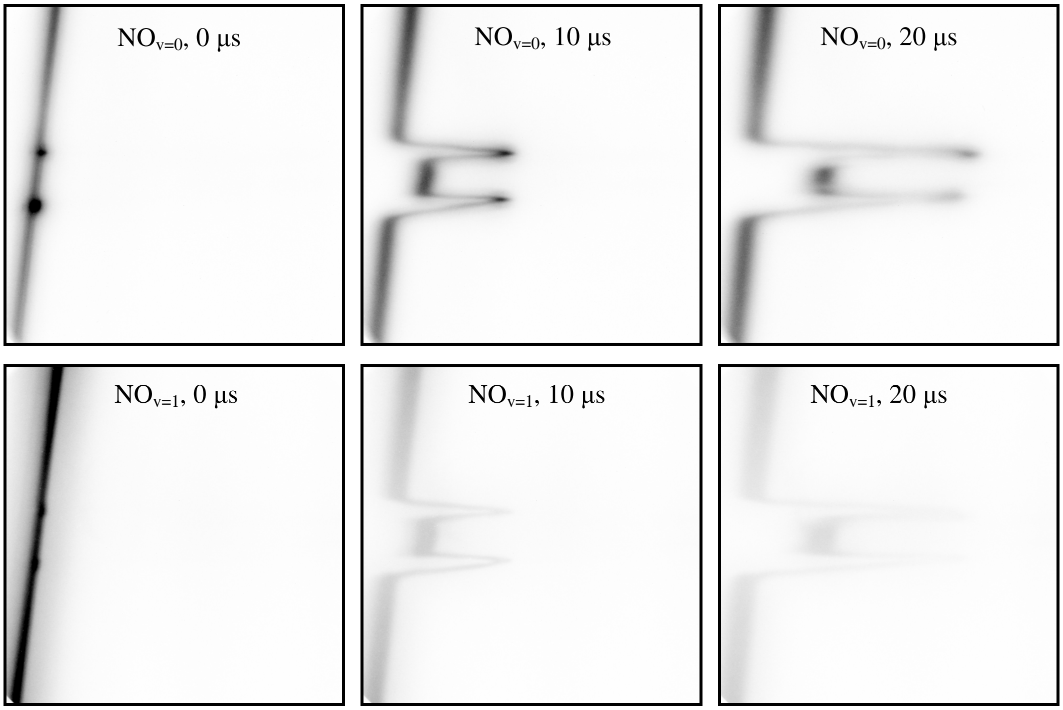


Fig. 5.10 Raw images of 0 μs, 10 μs, and 20 μs for the $NO_{v=0}$ (top) and $NO_{v=1}$ (bottom) probe lasers

In the images above, the vibrational decay is apparent through comparison of the $NO_{v=0}$ and $NO_{v=1}$ images. Sample single-shot raw images are shown in Fig. 5.11 at 25 μs for $NO_{v=0}$ probe laser. The images show significant turbulence, especially in the outer jet region of the highly underexpanded jet.

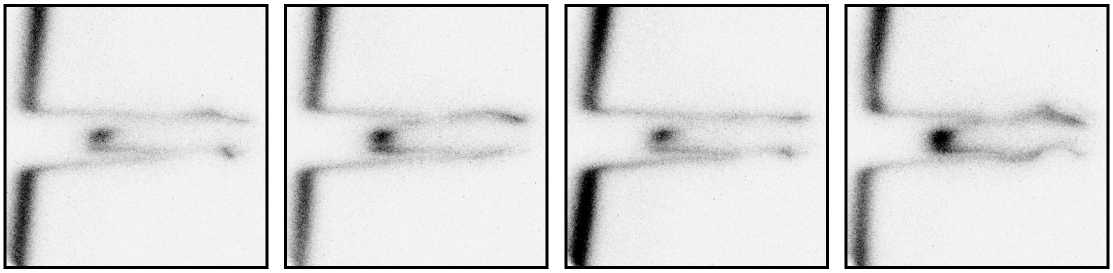


Fig. 5.11 Raw single-shot images of 25 μs for $NO_{v=0}$

C.3.1 Summary of experimental goals

The important difference between the two experiments was that Experiment 1 was focused on time-correlated velocimetry using a short time delay (few hundred ns) between the image pairs for better spatial resolution. Therefore, each pair of single shot images represents an instantaneous velocity measurement. Vibrational temperature is not addressed in this experiment because the short time delay is inadequate for vibrational energy decay which occurs on the μs timescale. The main objectives of Experiment 1 are: 1) to show the advantages of probing $NO_{v=1}$ over $NO_{v=0}$ for single-component velocimetry for suppressing the effects of background NO fluorescence, and 2) to demonstrate the use of photodissociation of NO_2 and probing of $NO_{v=1}$ for low uncertainty single component velocimetry through single shot image analyses. The effect of elevated temperature on the advantages of probing $NO_{v=1}$ or $NO_{v=0}$ beyond those available in this flowfield will be explored.

Experiment 2 focused on time-correlated vibrational temperature analysis using long time delays (tens of μs). In this case, since the images at long time delays suffered from low signal to noise, velocimetry accuracy was reduced and was not the focus of the experiment. Rather, the main objective of Experiment 2 was to 1) map the vibrational temperature decay of NO as a function of time and space, and 2) to explore the underlying chemical kinetics mechanisms which control the vibrational decay of NO.

C.3.2 CFD simulations

Two-dimensional axisymmetric simulations of the under expanded jet were performed using the Cobalt [Strang (1999)] flow solver. Cobalt is an unstructured flow solver designed to integrate the Euler and Navier-Stokes equations on two and three-dimensional grids. The grid used for this simulation consisted of unstructured hexahedral cells and is shown in Fig. 5.12. The subsonic part of the nozzle was included in the simulation in order to capture the effects of the boundary layer. The effects of turbulence were modeled by Menter's two-equation model [Menter (1993)].

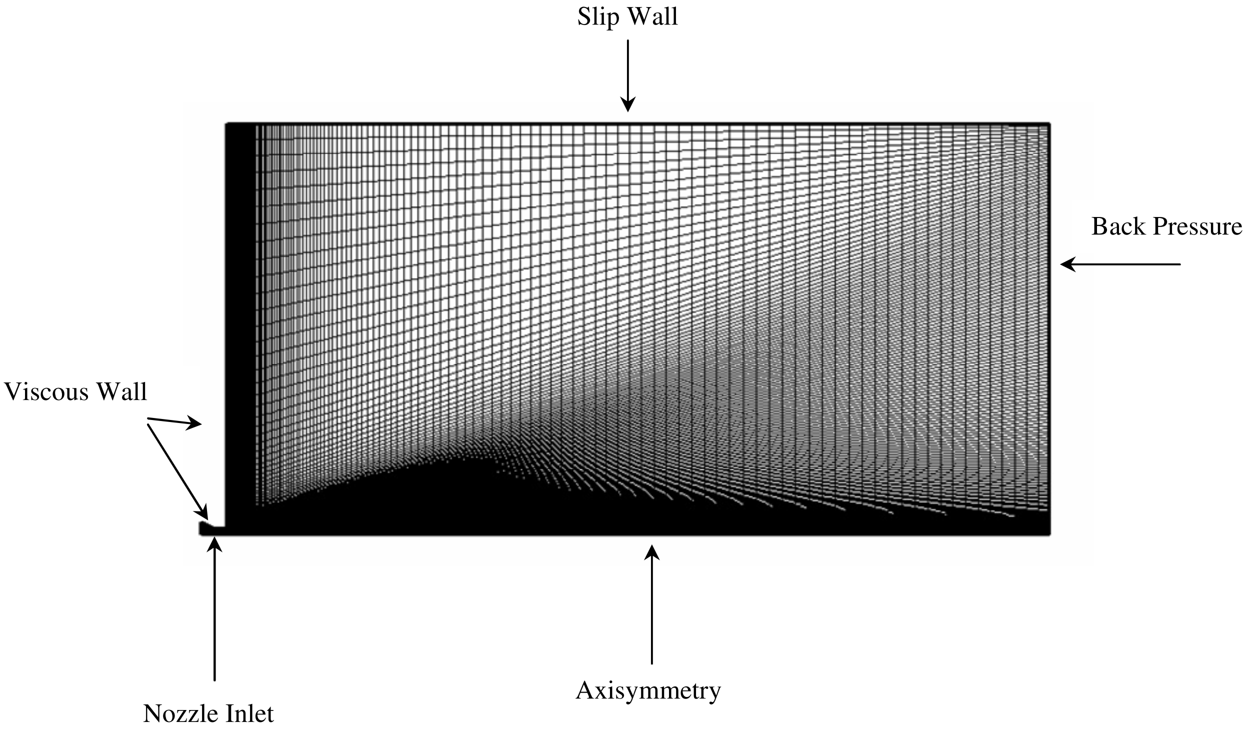


Fig. 5.12 CFD simulation grid

The initial condition within the test chamber was set to the experimentally measured pressure. Adiabatic viscous wall boundary conditions were specified for the nozzle wall and the chamber surface with the nozzle exit. Static conditions, based on the stagnation values and area ratio, were specified at the nozzle inlet. Slip conditions were used for the chamber sidewall and backpressure was specified at the chamber exit surface. The results obtained from these simulations were compared against the experimental data.

D. RESULTS AND DISCUSSION: EXPERIMENT 1

D.1 Integrated Images for Single-Component Velocimetry

Each of the 16 30 s integrated image pairs were analyzed to give a streamwise velocity map. To do this, each image was examined separately. After the images were set at equivalent fields of view, they were analyzed with a similar edge-finding procedure to locate the rising and falling edges of each photodissociation line, and the spatial displacement of the edges were tracked and divided by the 400 ns to calculate the streamwise velocity. One sample picture of two locations (location 6 on the left and location 13 on the right) are shown in Fig. 5.13. Location 6 corresponds to about 2/3 of the way from the nozzle to the Mach disk, and Location 13 corresponds to nearly immediately after the Mach disk. The gray line is the initial position, while the white line is the 400 ns delayed location. As discussed, only the streamwise velocity component was calculated, and was simply the displacement divided by the elapsed time.

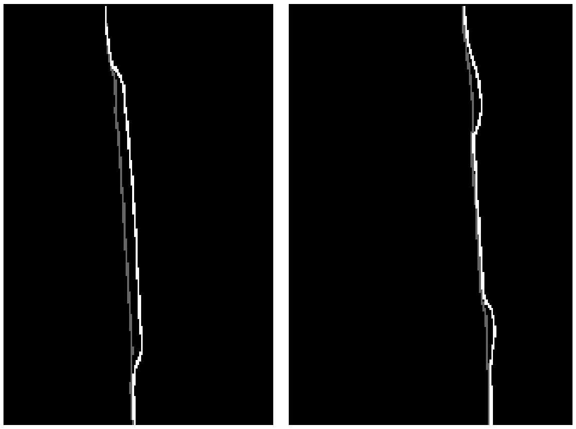


Fig. 5.13 Examples of edge finding procedures for streamwise velocity calculation at locations before and after the Mach disk (1)

Since the intensity of raw signal was the only factor for velocimetry analysis, no background subtraction or correction for laser sheet inhomogeneity was performed. Nozzle scatter (which would lead to erroneous "edges", from the probe lasers was minimized by temporally delaying the cameras by about 10 ns after the arrival of the laser. Since only movement in the streamwise direction was considered, there was a slight systematic error in the measurements. The areas with the highest radial velocity, such as directly outside of the nozzle, were most subject to this error. To calculate the velocity, the software located each point in the initial image and then searched for a corresponding point in the delayed image using a user-defined pixel roaming value. For example, the user may choose to specify the roaming pixel maximum as -10 to 20 pixels, meaning that for each point in the initial image, the corresponding point in the delayed image must be less than 10 pixels to the left (negative streamwise velocity) to 20 pixels (positive streamwise velocity) from the initial point, or else no velocity was calculated. Since the flowfield structure and velocity map was well studied in the literature, an estimate of the roaming pixel values could be calculated easily. The program also included routines to filter for multiple vectors originating from the same initial point. In this case, the nearest column (smallest streamwise velocity) was accepted while the other(s) were discarded. This served as a safety net in the case that too large a pixel roaming window was selected.

The velocity associated with the movement of each edge was assigned to a spatial position by locating the pixel which was equidistant between the location of the initial and the displaced pixels for each row. For example, if the rising edge of one of the photodissociation lines moved from column 20 to 30 in a particular row, the velocity associated with this movement would be assigned to the column 25 of the same row. Similarly, if the falling edge moved from 25 to 35 in the same row, the calculated velocity was assigned to column 30 of that row. Therefore, for each photodissociation line, two streamwise velocity values were calculated. Examples of these for location 6 and 13 are shown in Fig. 5.14:

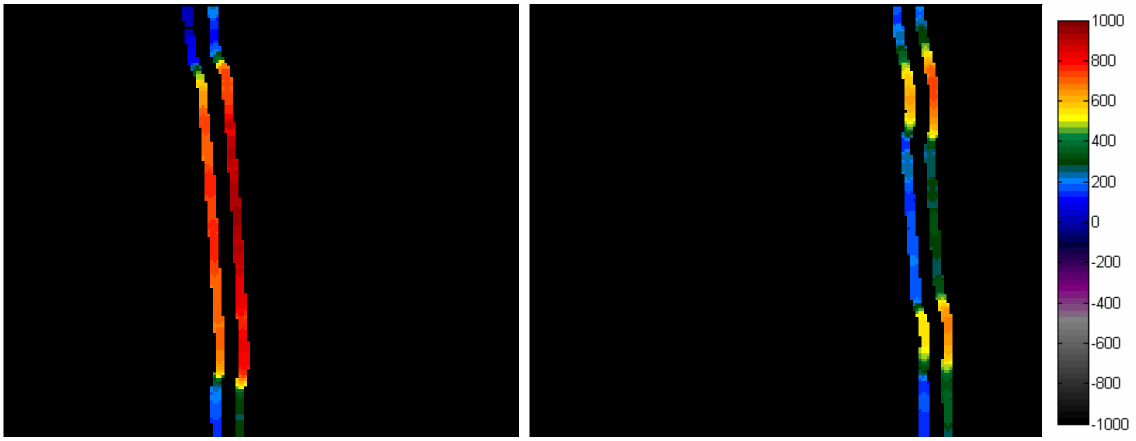


Fig. 5.14 Examples of edge finding procedures for streamwise velocity calculation at locations before and after the Mach disk (2)

This process was repeated for each image pair (16 total). The resulting streamwise velocity lines were added together to create a velocity map, as shown in Fig. 5.15:

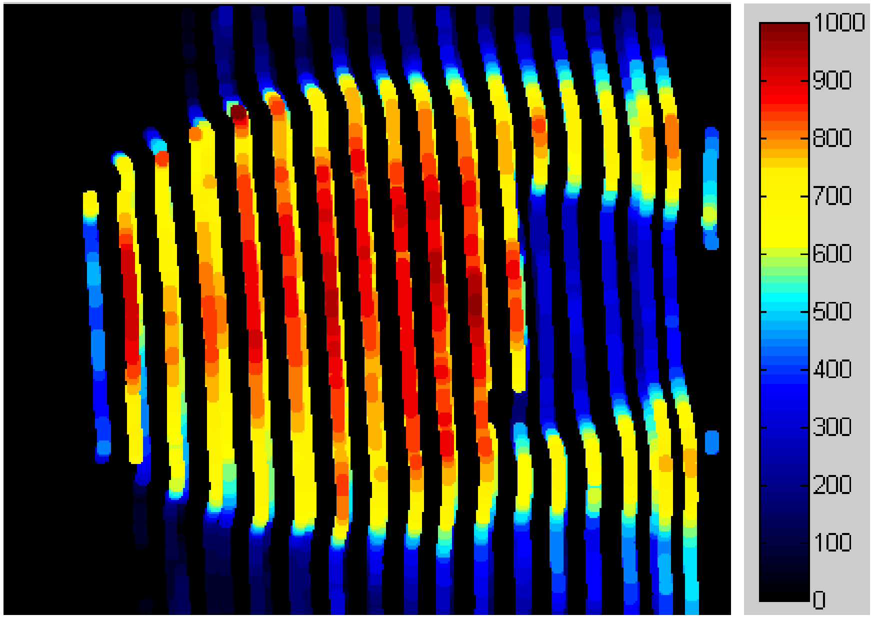


Fig. 5.15 Resulting streamwise velocity map from single line photodissociation

For this analysis, only very minimal smoothing was applied to the images, and the final resolution stands at about 500 µm x 500 µm in the field of view. The velocity map shows the expected trends of the underexpanded jet. That is, in the streamwise direction, the flow accelerates to hypersonic velocities until it reaches the Mach disk, wherein the flow undergoes a normal shock and immediately slows to subsonic speeds. The supersonic outer jet regions are seen in the velocity map as well. The corresponding images for Run 1 could not be analyzed due to the high NO background fluorescence which masked the photodissociation lines; the analysis software could not locate the rising and falling edges. Additionally, Run 1 was run at nearly twice the NO_2 concentration (doubling the signal from photodissociation, and at less than half of the NO concentration compared to Run 2, and even with the very small amount of background NO compared to NO_2, the photodissociation signal was not great enough for use in the velocimetry analysis, clearly showing the advantages of probing $NO_{v=1}$ over $NO_{v=0}$ in this situation.

D.2 Single Shot Image Analysis

630 single shot images were taken at location 7. This location is roughly 2/3 of the distance from the nozzle to the Mach disk and was chosen because it gave moderately good signal (better than directly before the Mach disk, but not as good as near the nozzle), and because the high velocities at this location provided an appreciable amount of spatial movement in the 400 ns, the time delay used for increased velocity resolution. The image pairs were analyzed using the same edge-finding procedure as the integrated images (previous section). A typical image pair and calculated velocity map for Run 2 is shown Fig. 5.16, cropped to the region of interest to reduce processing time.

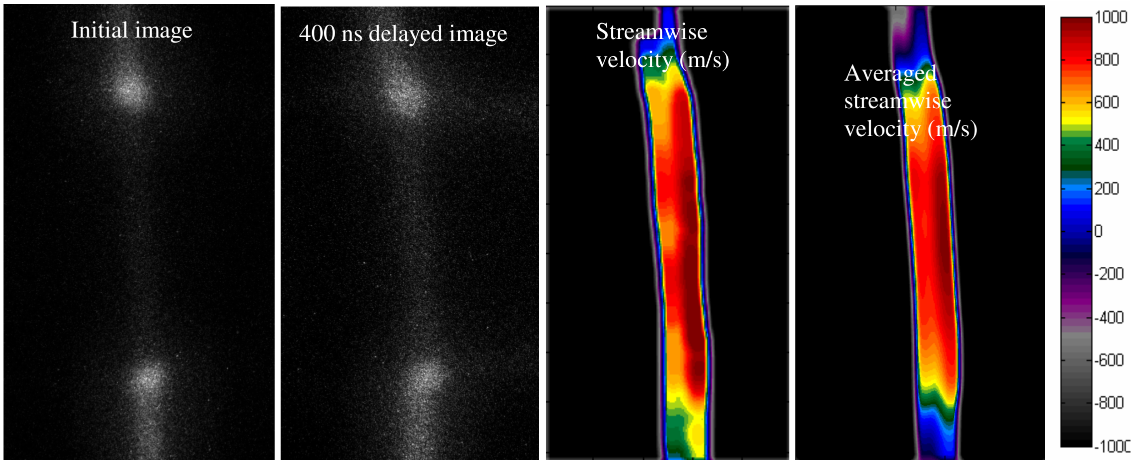


Fig. 5.16 Initial and time delayed single shot images (Run 2) with interpolated streamwise velocity map (m/s) and averaged streamwise velocity map, $\overline{U}$ (m/s)

All of the single shot interpolated velocity maps were averaged to produce an average streamwise, $\overline{U}$ velocity map. The spatial resolution was slightly greater than that in the integrated images, about 1 mm x 1 mm in the field of view.

The $\overline{U}$ map showed a slight acceleration which is expected at this location, near the Mach disk. Also, the outline of the jet was seen, with surrounding areas on top and bottom of the jet at nearly zero velocities. The U_{rms} map showed 1σ uncertainties of 40 to 60 m/s throughout the flowfield, about 5% given that the average streamwise velocity in the expansion was about 800 m/s. The uncertainties were lowest surrounding the bright knobs (see fig above), which corresponded to the higher number density in the barrel shock region, and higher in the middle of the expansion region due to lower signal to noise, i.e. lower number density, in this region. The low uncertainties for the single shot analysis are promising for application of this technique towards single-shot facilities such as shock tubes, or for statistical analysis in dynamic flowfields.

The blurring induced by a finite camera exposure time was also considered. To this effect, the camera exposure time (50 ns) was kept short compared to the time delay (400 ns), causing only minimal blurring of the signal in the images. In practice, the exposure time could be shortened to 20 or 30 ns to further reduce any blurring. However, since edge-finding procedures were used instead of simply finding the maximum of the lines, any slight blurring would spread out the intensity distribution and flatten the intensity profile, but likely not change the location of the largest gradient. The best way to reduce the uncertainty would be to increase signal to noise. This could be accomplished with a higher seeding NO_2 density followed by higher photodissociation laser intensity since the fraction of photodissociated NO_2 was >99.9%. However, it should be noted NO_2 absorbs at 355 nm so increasing the concentration of NO_2 may cause a decrease in the fraction of initial laser intensity that reaches the field of view. In addition, in a different optical setup, the razor blades could be placed closer to the flowfield, which would provide sharper lines and more precise location of the edges. In this setup, the razor blades could only be placed approximately 3 inches from the jet. A longer time delay could also be used, although this would produce more accurate velocity maps at the cost of spatial resolution.

D.3 Effects of Diffusion on Velocimetry Measurements

Diffusion causes the photodissociation line to blur as it translates in time and its effect on the velocimetry image analysis was investigated. The gaseous diffusion coefficient (m^2/s) can be calculated using the following equation:

$$D_{Diffusion} = \frac{1}{3} < v > \lambda \qquad \text{(Eq. 5-2)}$$

where $< v >= \sqrt{8kT / \pi\mu}$ and $\lambda = \frac{< v >}{Z_1} = \frac{< v >}{\pi d^2 < v > n_1}$. The terms are defined in the Appendix. The calculated diffusion coefficient for the areas along the centerline directly outside of the nozzle, directly before the Mach disk, and directly after the Mach disk were 0.00011, 0.01044, and 0.01184 m^2/s, respectively. Using a one-dimensional random walk, the rms distance (m) that a molecule travels in 400 ns was calculated by $z_{rms} = \sqrt{2D_{Diffusion}t_{randomwalk}}$, where $t_{randomwalk}$ is the time (s). z_{rms} was calculated to be 1.2, 11.7, and 12.5 pixels, respectively. Therefore, near the nozzle, diffusion was negligible, since the flow was moving only about 15 pixels during the measurement time. However, for the area directly before the Mach disk and for the region directly after the Mach, these values were significant. The fluid directly before the Mach disk, assuming movement of about 800 m/s and using a spatial conversion of 74 pixels/mm, moved approximately 24 pixels in 400 ns and the fluid directly after the Mach disk (about 300 m/s), moved about 9 or 10 pixels.

To investigate these effects, a synthetic binary image with a perfectly crisp photodissociation bar was created, where the initial line was not diffused. Two displaced images were generated, each with 20 pixel streamwise displacements, corresponding to a streamwise velocity of 670 m/s. The first was a crisp, undiffused image, and the second was diffused by convoluting the displaced image with a Gaussian kernel of 12 x 12 pixels, representing the effect of diffusion calculated from z_{rms} . These images are presented in Fig. 5.17.

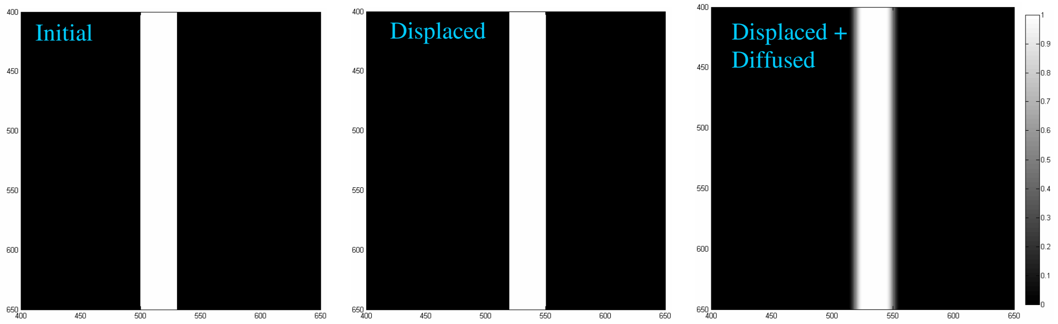


Fig. 5.17 Synthetic images for investigation of diffusion effects on velocimetry

These images were then run through the same edge-finding software used in the image analysis with no additional smoothing. The X gradient was taken and the peaks were located for velocimetry. A slice of this analysis is shown in Fig. 5.18 for the undiffused and diffused cases with the peaks and valleys used for velocimetry marked in green and red stars.

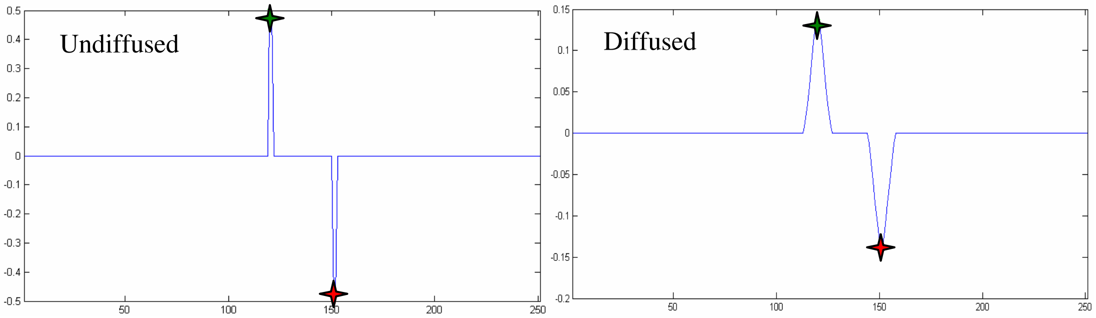


Fig. 5.18 Located edges with and without the effect of diffusion

In fact, the analysis shows that there was no difference between the undiffused and diffused peak location. Both pinpointed the same pixels for the maximum and minimum peaks, corresponding to the left and right edges of the photodissociation bar. Therefore, exactly the same velocity values were calculated either with or without diffusion. However, one obvious effect of diffusion was the decrease of the intensity of the gradient. The thresholding for which peaks and valleys are identified as edges was user specified, and if the overall intensity of the gradient was decreased, some of these would be missed if they drop below the threshold. This could be an issue in the case where the photodissociation bars are spaced very closely together where diffusion would cause them to blur into one another making the identification of peaks and valleys in the gradient difficult. In noisy images, also, the decrease in gradient intensity due to diffusion would cause mis-identification or non-identification of true edges. It should be noted that these conclusions are in the limit of isotropic diffusion. In the presence of strong density gradients, such as those encountered across the Mach disk, diffusion may not be isotropic, and thus may lead to uncertainties in the calculated velocities due to edge-finding inaccuracies.

D.4 Advantages and Disadvantages of Probing $NO_{v=1}$ vs. $NO_{v=0}$ as a Function of Temperature

The advantage of probing $NO_{v=1}$ was demonstrated in the underexpanded jet flowfield. However, this flowfield possesses relatively low temperatures (320 K and below), so it would be useful to examine the application of such a technique in higher temperature flowfields, such as combustion flowfields, which would contain naturally produced NO; this NO would then cause NO background fluorescence which would decrease the contrast of MTV images, and cause higher uncertainties in the derived velocity maps. The fraction of NO molecules in $NO_{v=0}$ and $NO_{v=1}$ as a function of temperature is shown in Fig. 5.19, plotted from 300K to 20,000K:

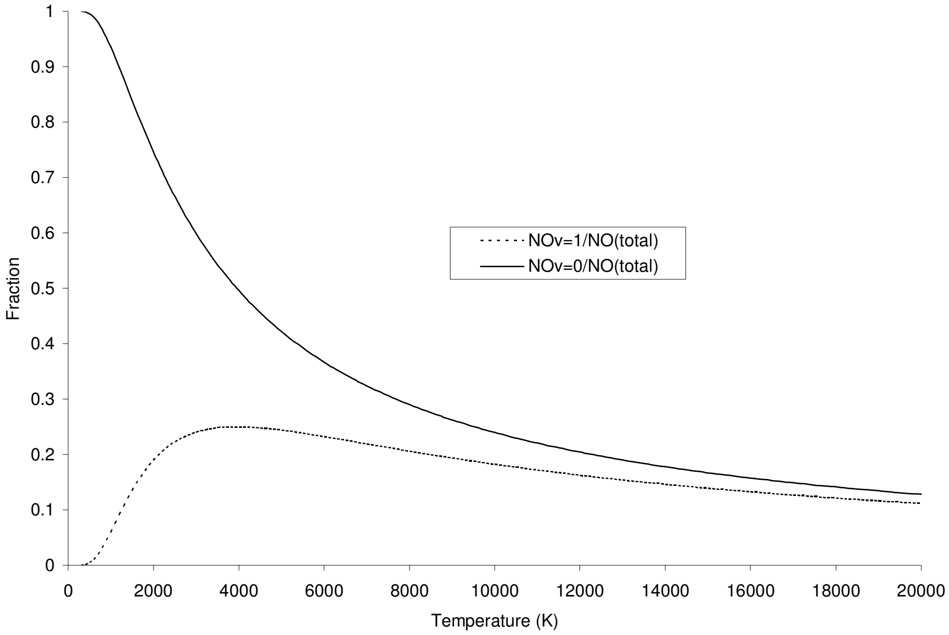


Fig. 5.19 Population fractions of $NO_{v=1}$ and $NO_{v=0}$ as a function of temperature

The population of $NO_{v=0}$ was highest at low temperatures, and decreased as temperature was increased. However, the population of $NO_{v=1}$ peaked around 4000K and then decreased slowly. This was indicative of the fact that at very high temperatures, the Boltzmann distributed shifted towards vibrational levels higher than v=1. The signal enhancement, defined as the photodissociation signal (provided that all of the seeded NO_2 is photodissociated along the laser path), divided by signal from background $NO_{v=0}$ or $NO_{v=1}$ is shown in Fig. 5.20. It was assumed that the fraction of seeded NO_2 was the same as the fraction of naturally occurring NO.

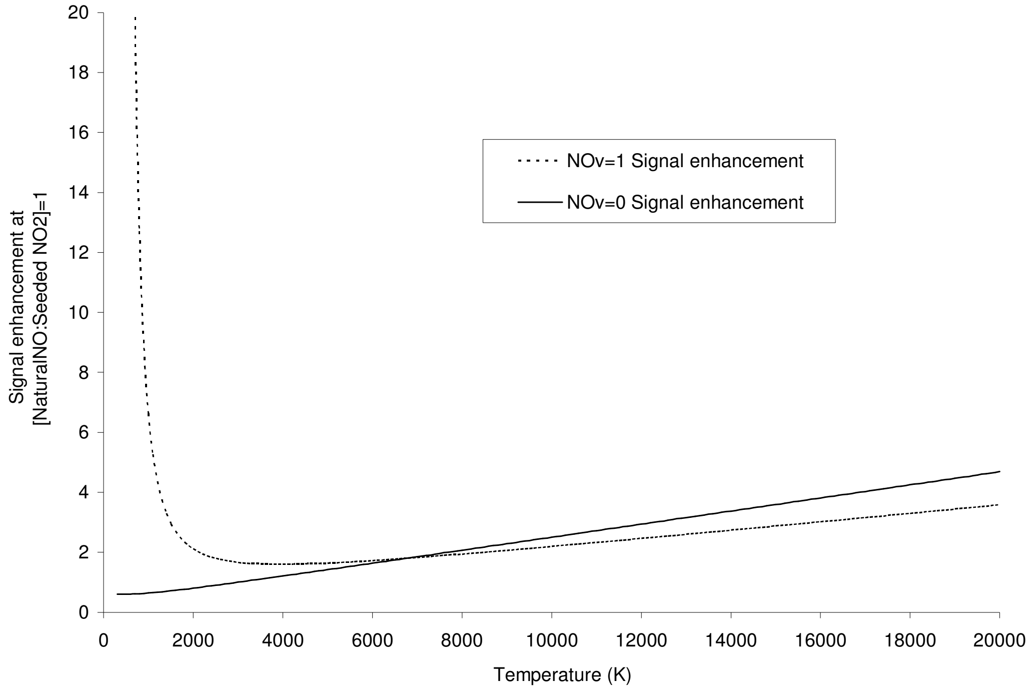


Fig. 5.20 Signal enhancement at elevated temperatures for probing $NO_{v=1}$ vs. $NO_{v=0}$

The plot showed the expected trends. First, at low temperatures, probing $NO_{v=1}$ clearly offered superior signal levels compared to $NO_{v=0}$ since the natural population of $NO_{v=1}$ is low. Second, since the fraction of $NO_{v=1}$ peaked around 4000K, this is the lowest point in the signal enhancement. The signal enhancement for both $NO_{v=0}$ and $NO_{v=1}$ increases monotonically as their natural populations decreased with temperature. However, as the temperature rose, the advantage of probing $NO_{v=1}$ decreased, and was overtaken by $NO_{v=0}$ around 6000K. Past this point, probing $NO_{v=0}$ incurred only a slight advantage over $NO_{v=1}$. Also, it was interesting to note that the signal enhancement of $NO_{v=1}$ never dropped below 1.5. This was due to the fact that the peak population fraction of $NO_{v=1}$ was around 25%, and never reached the 40% population produced through photodissociation. Therefore, probing $NO_{v=1}$ in all situations will always give a signal enhancement of 1.5 or above.

The assumption that the fraction of seeded NO_2 was the same as the fraction of naturally occurring NO was not unreasonable, as the gas composition in high enthalpy facilities such as shock tubes can be tailored to produce a few percent of NO. The effect of altering the ratio of [Naturally Occurring NO: Seeded NO_2] on the signal enhancement of $NO_{v=0}$ and $NO_{v=1}$ are shown in Fig. 5.21:

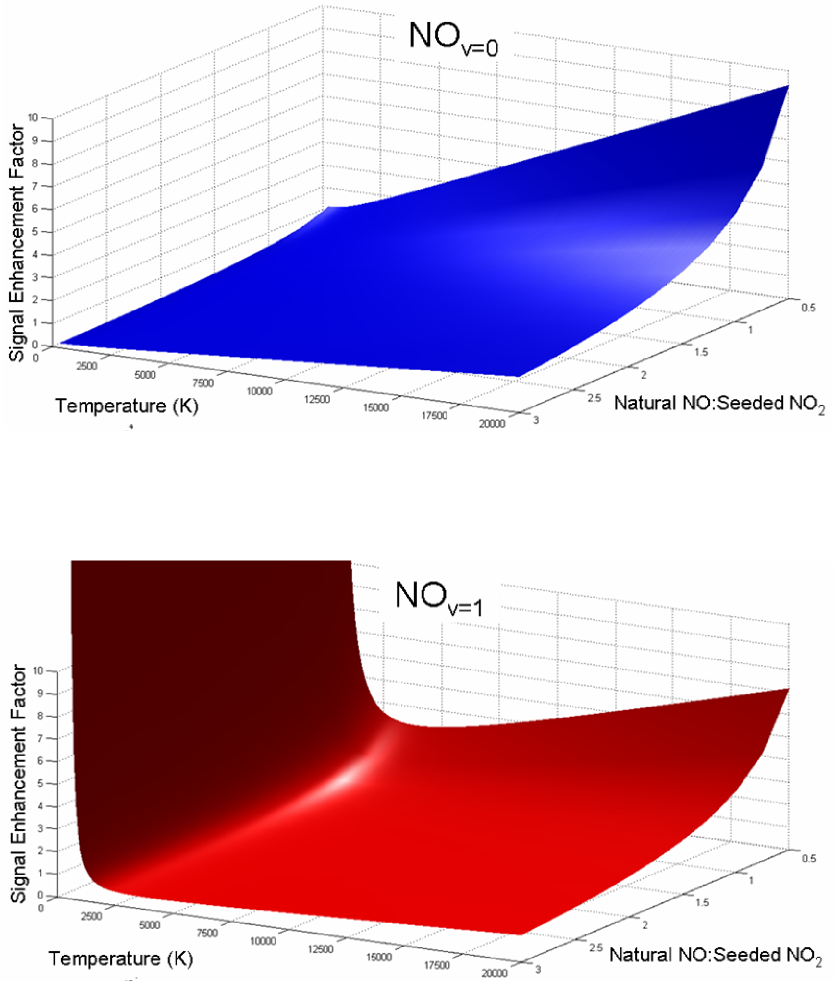


Fig. 5.21 Signal enhancement at elevated temperatures for probing $NO_{v=1}$ vs. $NO_{v=0}$ as a function of background NO concentrations

As expected, as the concentration of background NO compared to NO_2 increased, the signal enhancement afforded through NO_2 photodissociation decreased. At ratios of 2.5

and above, the photodissociation signal gives enhancements of 2 or below. Especially around typical flame temperature (1000K to 2000K), the advantage of probing $NO_{v=1}$ drops below 1, meaning that the photodissociation signal is lower than fluorescence from naturally occurring $NO_{v=1}$.

An important caviat is that these calculations assume that all of the seeded NO_2 is available for photodissociation. In fact, the thermal decomposition of NO_2 at elevated temperatures must be considered, as this will decrease the amount of NO_2 surviving for photodissociation, and would lower the signal enhancement for probing either $NO_{v=0}$ or $NO_{v=1}$. For example, Menon (2007) reported an 80% loss of NO_2 seeded at 5% due to thermal decomposition in a laminar premixed ethylene-air flame. At these conditions, the signal enhancement for probing $NO_{v=1}$ would drop below unity but the signal enhancement for $NO_{v=1}$ would still be greater than that of $NO_{v=0}$. In the images, the photodissociation signal would show up as a slightly brighter area compared to background fluorescence, but would still be distinguishable. Therefore, velocimetry at these conditions could still be accomplished. The most straight-forward method to correct for background fluorescence in such a flowfield would be to take an integrated image of the background fluorescence and then subtract this image from each of the velocimetry images. This would help to enhance the contrast of the images and minimize the effects of background fluorescence.

E. Results and Discussion: Experiment 2

E.1 Integrated Images for Vibrational Temperature Mapping

In order to calculate the vibrational temperature from each pair of images ($NO_{v=1}$ and $NO_{v=0}$ probe lasers) at each nominal delay, an image processing code was developed in-house based on extraction of vibrational temperature by assuming a Boltzmann distribution of the two states. Therefore, correct calculation of the vibrational temperature requires accurate ratios of signal intensity, so the image processing included background image subtraction, exact overlaying of each pair of images, and laser sheet inhomogeneity corrections. Also, an initial correction which accounted for the difference in camera gain was applied such that the 0 μs nominal delay vibrational temperature in the ambient region was set to the known ratio of $NO_{v=1}$:$NO_{v=0}$ formed during photodissociation of NO_2 [e.g., see Hunter (1993), Brookes (2007), Harrison (1994)]. In addition, the final temperature map was filtered to produce a vibrational temperature map only where there was signal in both the $NO_{v=0}$ and $NO_{v=1}$ images. This prevented a cold biased temperature map from being created, since a small amount of $NO_{v=0}$ was always present as a contaminant in the NO_2 bottle. Sample vibrational temperature contour maps at several different time delays are shown in Fig. 5.22.

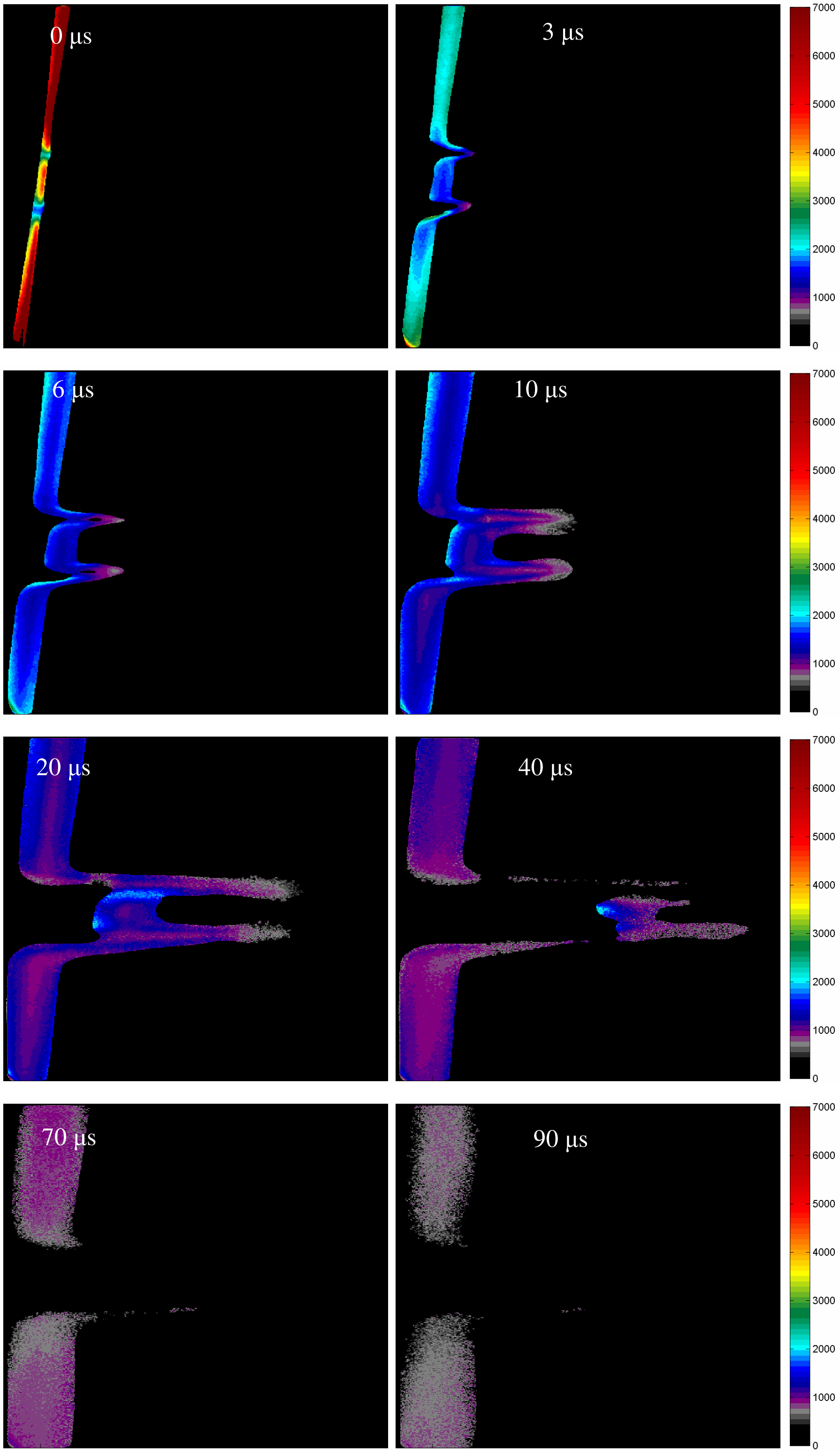


Fig. 5.22 Sample contour maps of vibrational temperature for nominal delays

In the images shown above, the vibrational decay is apparent in all sections of the flow. It was interesting to note that even at a nominal delay of 0 μs, the flow already showed significant vibrational decay, especially in the higher pressure areas of the flow (outer jet regions). This is mainly due to the fact that even the nominal 0 μs image was taken at a true delay time of 310 ns after the photodissociation pulse. The widening of the photodissociation beam due to diffusion was seen. Also apparent in the image sequence was the difference in vibrational decay rates between the ambient region, the centerline, and the outer jet regions. Thus, these regions of the flow were examined further by averaging the signal intensities of 15 rows along the centerline region, 3 rows for the outer jet regions, and 61 rows for the ambient region. These averaged profiles were smoothed to minimize the effects of hot pixels. The signal intensities for $NO_{v=0}$ and $NO_{v=1}$ were then plotted as a function of nominal delay time and smoothed. The extracted vibrational temperatures plotted as a function of nominal delay time is shown in Fig. 5.23 for each of the interrogation regions with an inset showing a closeup view of the initial decay.

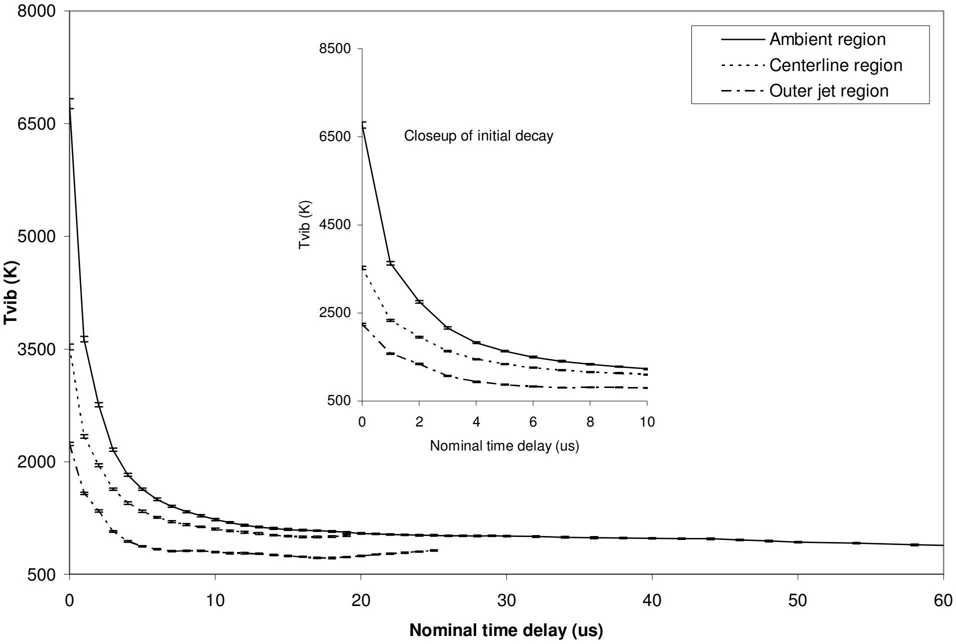


Fig. 5.23 Vibrational temperature decay for interrogated regions

It is apparent from Fig. 5.23 that the different regions of the flow exhibited very different decay rates; as expected, the decay rates were fastest for the high pressure region (outer jet regions), and slowest for the low pressure region (ambient region). This trend was also reflected in the vibrational temperature contour images where the outer jet region showed the lowest vibrational temperatures, the centerline showed moderate vibrational temperatures, and the ambient region showed the highest vibrational temperatures. The lowest vibrational temperature observed was 485.0 K which corresponded to a population of $NO_{v=1}$ to $NO_{v=0}$ of 0.00349:1 and reflected the limits imposed by both low signal levels and the dynamic range of the cameras. The comparison of experimental data to kinetic modeling of the vibrational temperature decay will be discussed in the next section.

As a brief aside, the velocity of the molecules was analyzed in a way similar to in Experiment 1 to extract the streamwise velocity map in this post-Mach disk region of the flow. In order to determine the velocity of the tagged molecules, both the spatial conversion of the image (pixels to mm) and the time delay between successive images used for the velocimetry were critical. The spatial conversion of the image was determined by placing an electroformed mesh (19 lines/inch) in the path of the laser sheet and taking an integrated image of the flowfield, which then showed distinct bright and dark lines. Using this image, and accounting for laser sheet divergence, yielded a spatial conversion of 28 pixels/mm. This value was then used in the velocity image processing to convert pixel displacement to length displacement. For this experiment, since the tagged molecules formed a line, only the x-component of the velocity was calculated. The velocity imaging only required one set of images (probing either $NO_{v=0}$ or $NO_{v=1}$) although velocities were calculated for both the $NO_{v=0}$ and $NO_{v=1}$ probe lasers in order to verify their equivalence. Since the velocity map was compared to CFD simulations, a spatially local, not average, velocity map was desired. To approximate the instantaneous velocity, the image processing always used a pair of successive images in time. The first of the pair is referred to as the initial image, and the second as the displaced image.

The images were background corrected, and then Gaussian smoothed using a 3x3 filter matrix, and each row was smoothed individually by 10 pixels to minimize the effect of hot pixels. No additional smoothing was done in the y direction. The maximum intensity of each smoothed row was determined and marked; this maximum intensity will be referred to as the marked position. The velocity was extracted by calculating the distance traveled by the marked pixel in each row and dividing by the time elapsed. By repeating this process and assigning each pair of images used for the velocimetry to their respective spatial positions, a velocity profile at each displacement could be generated. The velocity profiles were filtered for negative or abnormally high values. The resulting velocity map was interpolated in the same way as in Experiment 1 and is displayed in Fig. 5.24, overlaid with the original molecular tagging velocity profiles (black lines).

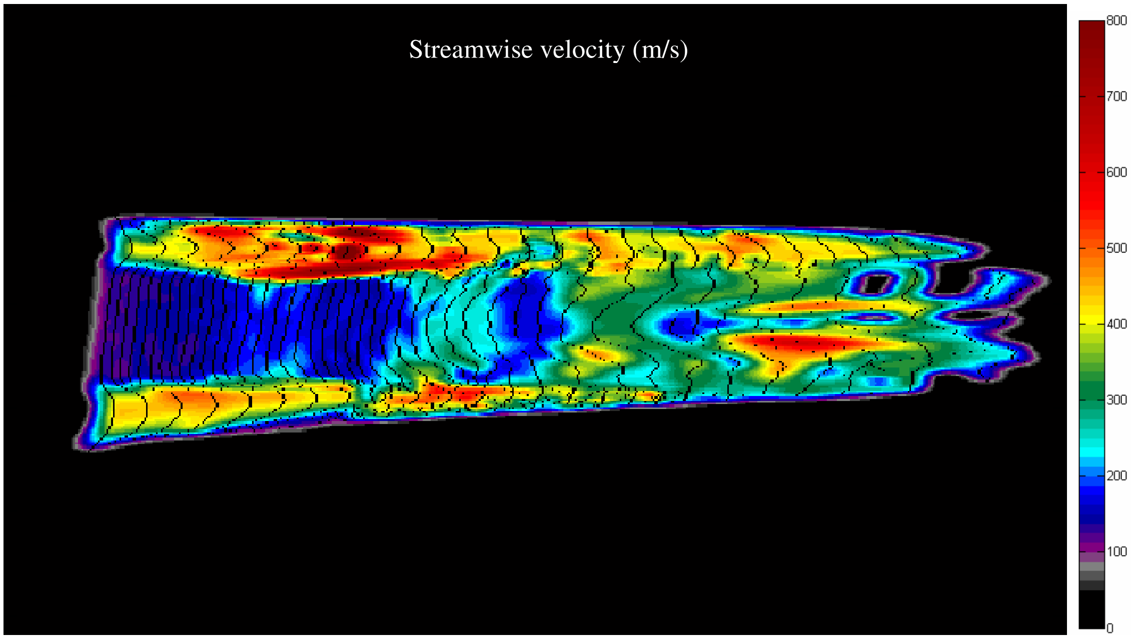


Fig. 5.24 Interpolated velocity map for NO$_{v=0}$

The velocity map was then compared with CFD simulations of the flowfield run at P_o = 554 Torr and $P_{ambient}$ = 12.2 Torr. The resulting plots are shown in Fig. 5.25 for $NO_{v=0}$, the CFD result is shown as the dashed line, and the dots represent raw experimental (not interpolated) data points. Representative errors bars are drawn. The streamwise displacement of the experimental data to the CFD results were determined by alignment of the Mach disc using the 1% NO in N_2 balance at equivalent pressures with the resulting CFD structure. The uncertainties were determined by using the single-shot images and appropriate scaling by the square root of the number of laser shots collected for each averaged image. The experimental data reflect the range of values obtained by evaluating each row in the centerline region, and are evenly spaced across 30 rows.

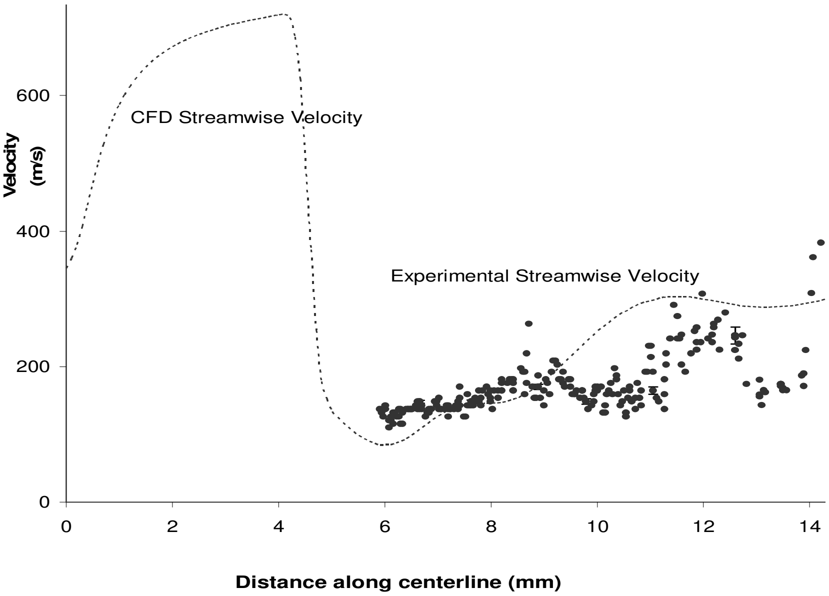


Fig. 5.25 Comparison of CFD and experimental streamwise velocites in the centerline region for $NO_{v=0}$

The comparison of experimental velocity to CFD results will be discussed in the following section. The velocimetry process was repeated for $NO_{v=1}$ and produced a similar velocity map, although lack of signal in later delays due to vibrational relaxation produced “holes” in the data which can be seen by the abnormally high velocities in these regions. The comparison to CFD is also shown for reference (Fig. 5.26).

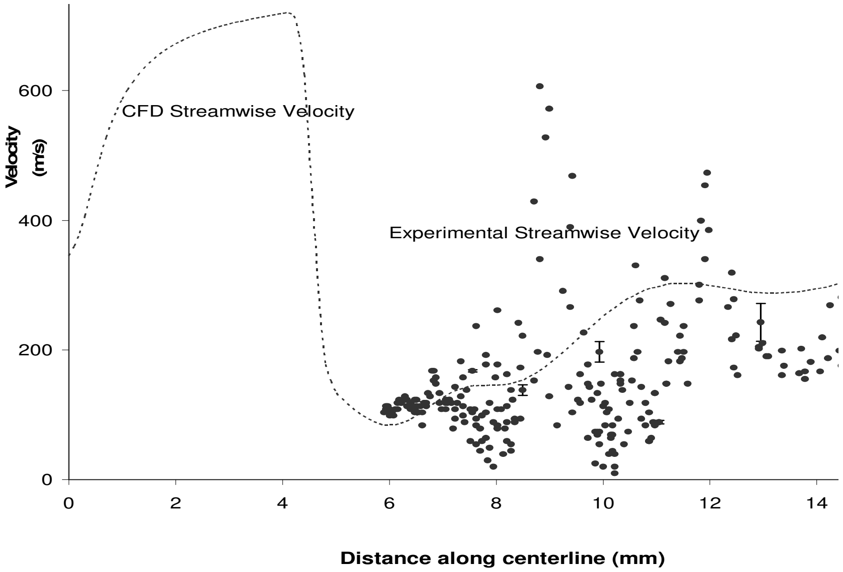


Fig. 5.26 Comparison of CFD and experimental streamwise velocites in the centerline region for $NO_{v=1}$

From the figure above, it is clear that the $NO_{v=1}$ suffers from higher uncertainties due to loss of signal. This was seen both in the scatter of data around the CFD velocity, and the larger errors bars. From the $NO_{v=1}$ data, the oscillations in velocity due to the recompressions following the Mach disk were more difficult to discern. However, they followed the same trend and occurred in the same streamwise locations as the $NO_{v=0}$ oscillations, thus validating that either set of probe laser data could be used in the velocimetry analysis.

Lastly, it should be emphasized that since the purpose of Experiment 2 was to use long time delays for examination of vibrational temperature, velocimetry accuracy suffered since at long times, the natural diffusion and mixing of fluid causes the photodissociated line to broaden significantly leading to loss of signal. This can be seen by the scatter in the data which increased with distance, and therefore time from photodissociation. In addition, depletion of the population of $NO_{v=1}$ through vibrational quenching hinders the use of velocimetry using time delays greater than around 30 μs. Therefore, for purely velocimetry study of flowfields, an approach such as Experiment 1 utilizing short time delays and movement of the photodissociation line across the flowfield and probing of $NO_{v=1}$ should be applied. In this way, the photodissociation line remains crisp, and the short time delay provides superior spatial resolution.

E.2 Flowfield Structure Comparison to CFD

Comparisons were made between the experimental PLIF data to CFD modeling results. A critical parameter in comparing the flowfield structures from experiment and theory was the distance from the nozzle to the Mach disk, X_m. PLIF images of Q_1+P_{21}(9.5) transition were used for locating the Mach disk since the Mach disk was clearly delineated in these images. The Q_1+P_{21}(9.5) fluorescence image is shown in Fig. 5.27. It is also interesting to note that the weak recompressions following the Mach disk and the nozzle itself are visible in this image.

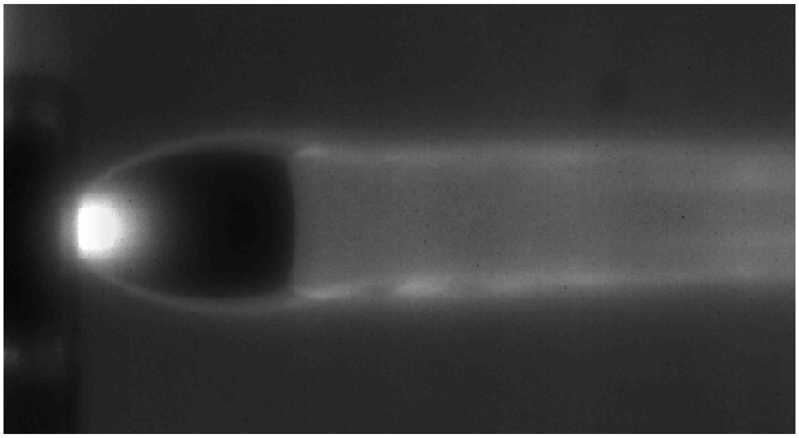


Fig. 5.27 NO PLIF image of Q_1+P_{21}(9.5) transition

The CFD modeling yielded a Mach disk location at X_m/D_e of 4.77 compared to the experimental value of 4.40, giving a difference of only 330 μm on the experimental scale. Also, the primary wavelength, w/D_e, and the Mach disk diameter, D_m/D_e, were found to be 5.10 and 2.48, respectively. These values agree well with the literature [Wilkes (2006)]. In addition, the Reynolds numbers fall within the transitional regime, and the jet should transition at ~10 mm (from the nozzle) and become fully turbulent at ~25 mm according to data reported in Wilkes (2005). The experimental data seems to agree well with this prediction through visual examination of the raw single shot images.

E.3 Vibrational Temperature Kinetics and Comparison to Modeling

Oscillations in streamwise velocity were seen in both the CFD and experimental results which originated from the weak recompressions following the Mach disk. The vibrational temperature decay along the centerline was modeled in Kintecus [Ianni (2008)] using a variable pressure profile which was obtained through the CFD modeling. The kinetics model consisted of 33 reactions and included both vibration-vibration (V-V) and vibration-translation (V-T) exchanges between all flow species as well as chemical reactions. The flow species considered included ground and first excited vibrational states of NO, O_2, and N_2, as well as chemical species NO_2, O_3, and O. Rate constants for the relevant reactions were taken from the literature [Candler (2002), Kosanetzky (1980), Bohm (1999), Green (1982), Dodd (1999), Ahn (2004), Jonbaik (1993), Demore (1997), Atkinson (2004), Tsang (1986)] and reverse rate constants which were not found in the literature were calculate via detailed balance.

The initial concentrations were determined by using the photodissociation energy flux at the field of view. The cross section for photodissociation of NO_2 was taken as 4.7 x 10^{-19} cm^2 [Demore (1997)], and the fraction of NO_2 photodissociated at the waist of the laser beam was calculated to be > 99.99%. This photodissociated NO_2 was assumed to follow the branching ratio to form $NO_{v=1}$ and $NO_{v=0}$ previously discussed. The rotational temperature was assumed to be 300K, so that initial fractions of vibrationally excited O_2 and N_2 were essentially zero. There was no initial concentration of O_3. Initial concentrations of O atom were set to equal the concentration of nascent NO. To observe the effect of the oscillations seen in the post Mach disk flow due to the weak recompressions, a volume profile (i.e. pressure profile) interpolated at 0.1 μs intervals was applied to the kinetics simulation. The profile was generated from the CFD results. Although the CFD results could not reproduce the experimental data exactly, there was no other method of extracting the pressure from the experimental data. This profile is shown in Fig. 5.28, along with the results of the simulations (Fig. 5.29, Fig. 5.30, and Fig. 5.31) to 70 μs (top) and to 200 μs (bottom) .

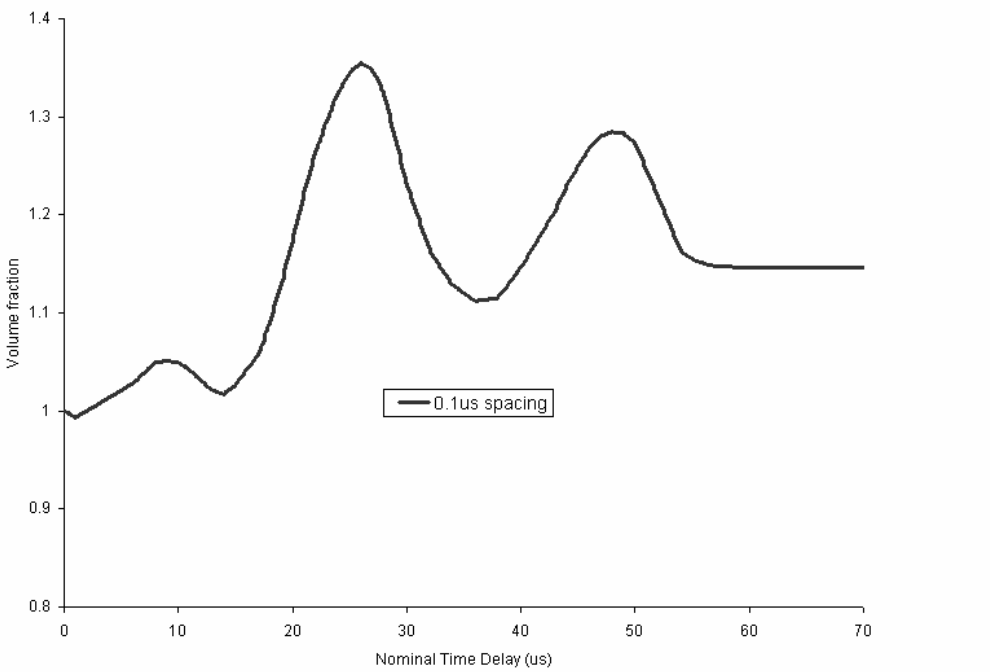


Fig. 5.28 Pressure profile used for kinetics simulations

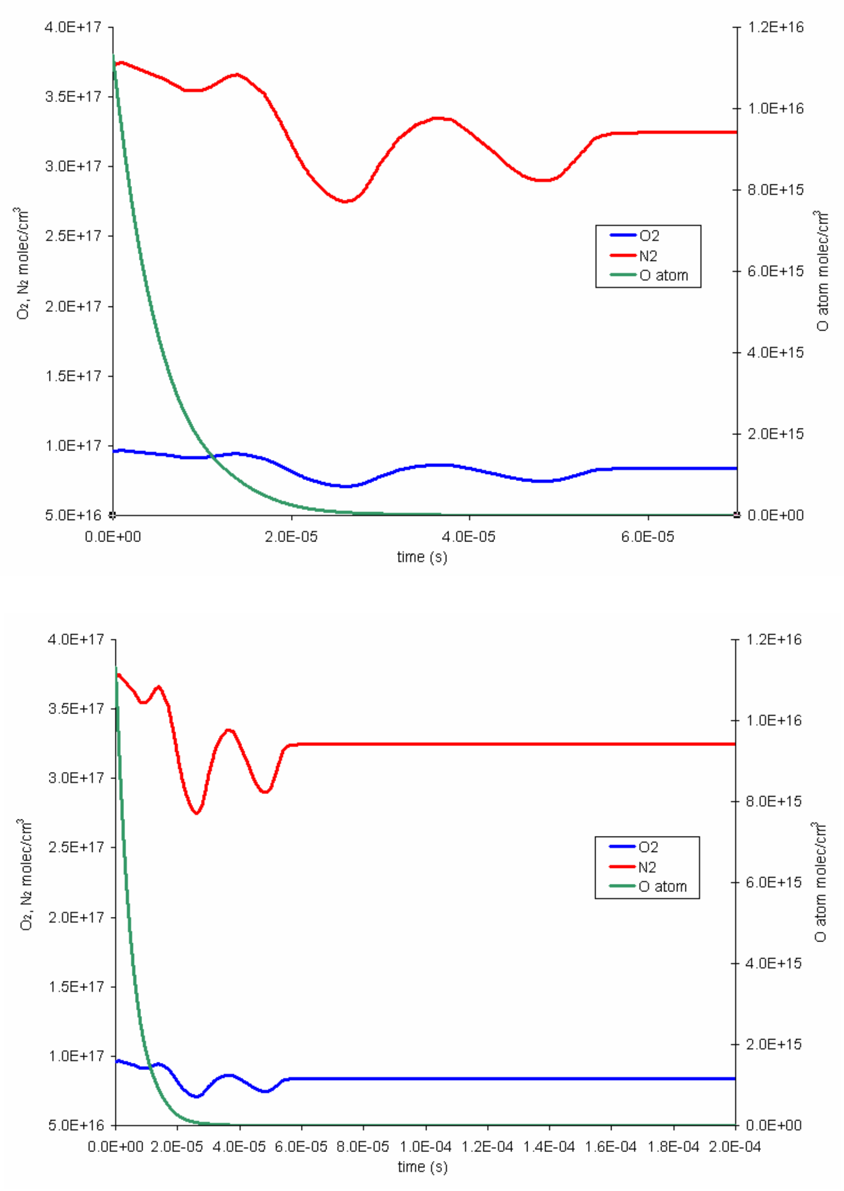
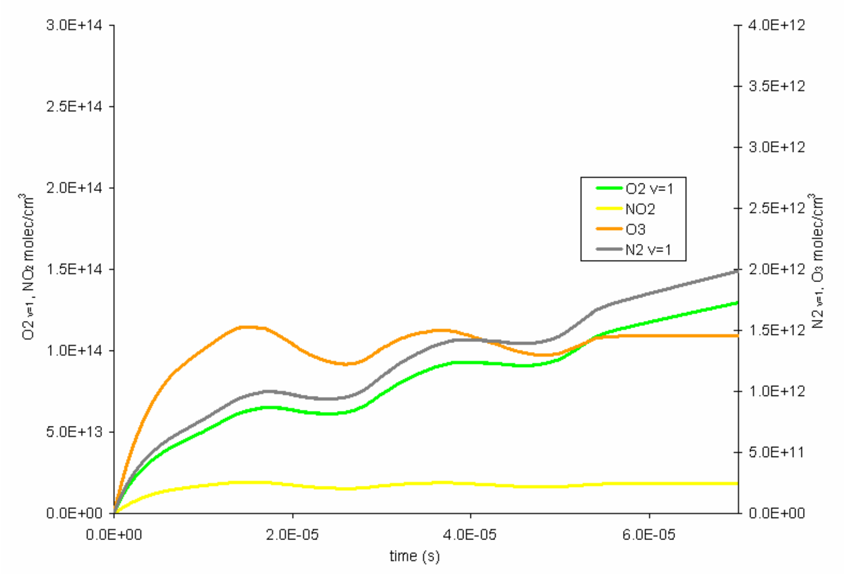


Fig. 5.29 Kinetics simulations concentrations of O_2, N_2, and O atom as a function of time results for short term (top) and long term (bottom)



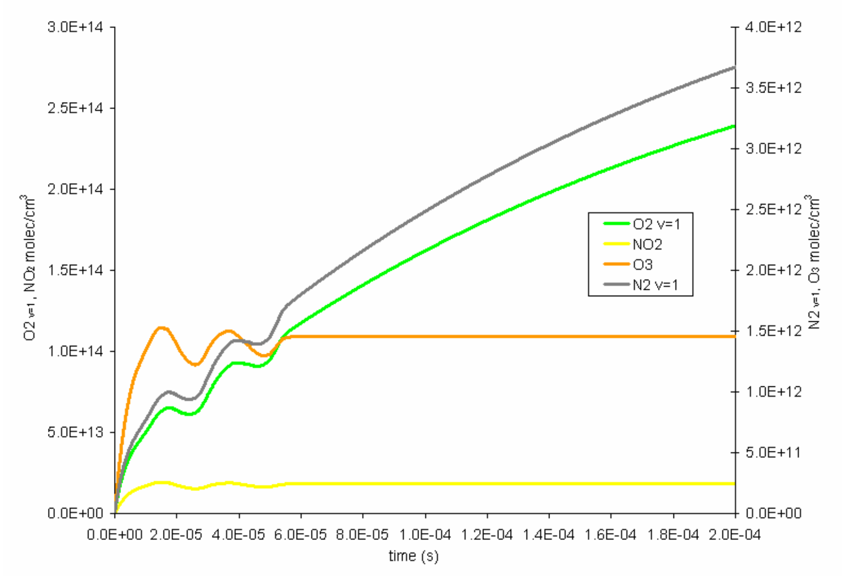
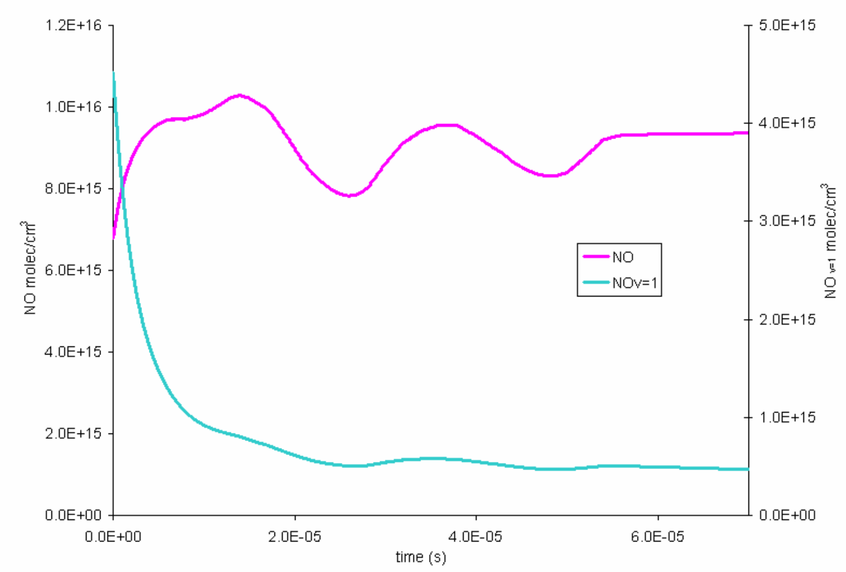


Fig. 5.30 Kinetics simulations concentrations of $O_{2,v=1}$, NO_2, O_3, and $N_{2,v=1}$ as a function of time results for short term (top) and long term (bottom)



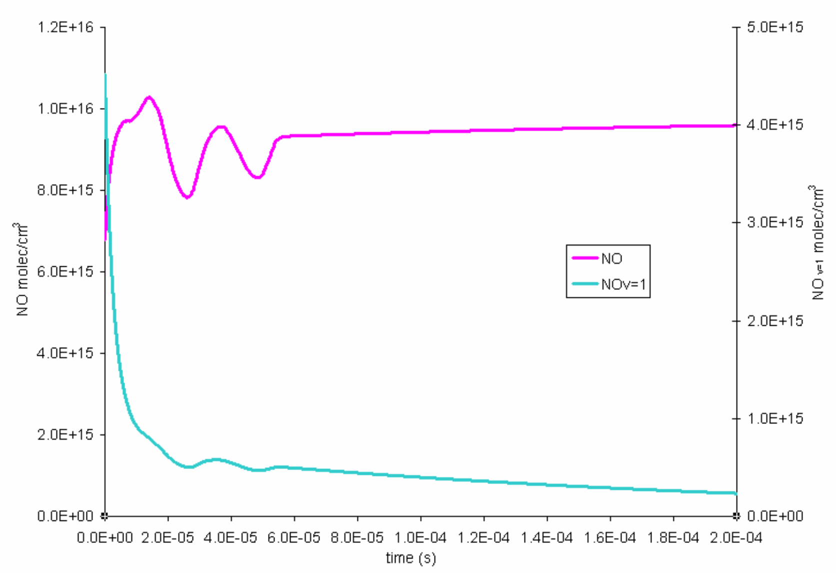


Fig. 5.31 Kinetics simulations concentrations of NO and NO$_{v=1}$ as a function of time results for short term (top) and long term (bottom)

From the results (Fig. 5.29, Fig. 5.30, and Fig. 5.31), it was clear that the volume profile has a large effect on the concentrations of the species throughout the flow as a function of time. It followed that a decrease in volume, which corresponded to the location of the weak recompression due to pressure increase, caused an increase in the concentrations of all species except for $NO_{v=1}$ and O atom The major species (O_2, N_2) experienced an overall drop in concentration (~12.8% at 70 μs) due to the increase in volume fraction with increases in concentration corresponding to a decrease in volume fraction. To model the diffusion of O atom an extra reaction was added with a rate constant which was adjusted to give a good fit of NO vibrational temperature with time. Therefore, an exponential drop in O atom concentration was seen. Other reactions served to consume O atom, including recombination reactions to reform NO_2, O_2, and O_3, as well as the chemical feed reaction to form $NO_{v=1}$. The concentration of $N_{2,v=1}$ and $O_{2,v=1}$ continued to rise after the initial recompressions, demonstrating small contributions to V-V and V-T energy exchanges, although the rise of $N_{2,v=1}$ was proportionally larger than that of $O_{2v=1}$ due to the smaller contribution of $N_{2,v=1}$ compared to $O_{2,v=1}$. O_3 demonstrated the short term fluctuations from the volume profile but leveled off due to elimination of O atom (and subsequent drop in rate of formation of O_3 through collisions of O with O_2. The concentration of NO_2 showed a leveling off effect due to drop in concentration of O atom (which could recombine with NO to reform NO_2), and the very small reaction rate of the three-body recombination reaction of NO with O_2 to form NO_2, along with no reactions which consume NO_2.

Finally, the concentrations of $NO_{v=0}$ and $NO_{v=1}$ are shown in the bottom row of Fig. 5.31. The kinetics model predicted a series of oscillations in the $NO_{v=0}$ population with time delay, as a result of local pressure oscillations, but only weak oscillations in $NO_{v=1}$ population due to the strong sink terms. The presence of some (although weak) oscillations is due to the depletion of O atom, since vibrational quenching of $NO_{v=1}$ was especially fast in collisions with O atom. As the concentration of O atom was increased, the oscillations were further dampened and eventually disappeared. The oscillations in the population of $NO_{v=0}$ peaked when the flow recompressed at each weak shock. The fact that oscillations in $NO_{v=1}$ were much weaker than those in $NO_{v=0}$ was due to the fact that the timescale of the oscillations following the Mach disk was much slower than the timescale of $NO_{v=1}$ loss. This trend was reflected in the experimental data: the locations of the recompressions, seen by a decrease in the streamwise velocity along the centerline, corresponded to the locations of the peaks in $NO_{v=0}$ oscillations. Therefore, the experimental data was self consistent. In comparing the experimental data to the CFD results and to the kinetic model results, perfect agreement was not expected since the pressure profile used to generate the kinetic model results was derived from CFD results, which, as seen before, failed to reproduce the experimental streamwise velocities. However, the magnitude and period of the oscillations seemed to correspond and the overall acceleration of the centerline region was captured.

Sensitivity analyses of $NO_{v=1}$ were conducted to explore the importance of the reactions on the decay rate of $NO_{v=1}$ (and consequently, vibrational temperature decay). The results from sensitivity analyses of $NO_{v=1}$ are shown in Fig. 5.32. The top and bottom panels show the sensitivity analysis zoomed in to $-0.1<NSC<0.1$ and $-1<NSC<1$, respectively. The important reactions are labeled. Sensitivity analyses revealed that the vibrational temperature decay was extremely sensitive throughout the simulation runtime to vibrational quenching of $NO_{v=1}$ by O atoms produced via photodissociation. Within the first 20 μs, the decay was only sensitive to the above mentioned reaction with minor contributions from other reactions such as V-V and V-T energy exchange with O_2, recombination of NO and O to form NO_2, and chemical feed reactions forming $NO_{v=1}$ [Dodd (1999)]. Vibrational quenching of $NO_{v=1}$ by O atoms was responsible for the extremely fast initial decays in the interrogated regions; at a nominal time delay of 0 μs, the higher pressure regions (centerline and outer jet regions) show significant vibrational decay from the initial T_{vib} of ~7000K to 3526K and 2236K, respectively. Therefore, the effect of this reaction was greatest in the outer jet regions since they were the highest pressure region, and lowest in the ambient region. At later times, the simulation became more sensitive to the recombination and chemical feed reactions. However, the sensitivity to vibrational quenching by O atom also increased and maintained a NSC several times greater than those of the nearest competitors, V-V and V-T energy exchanges with O_2.

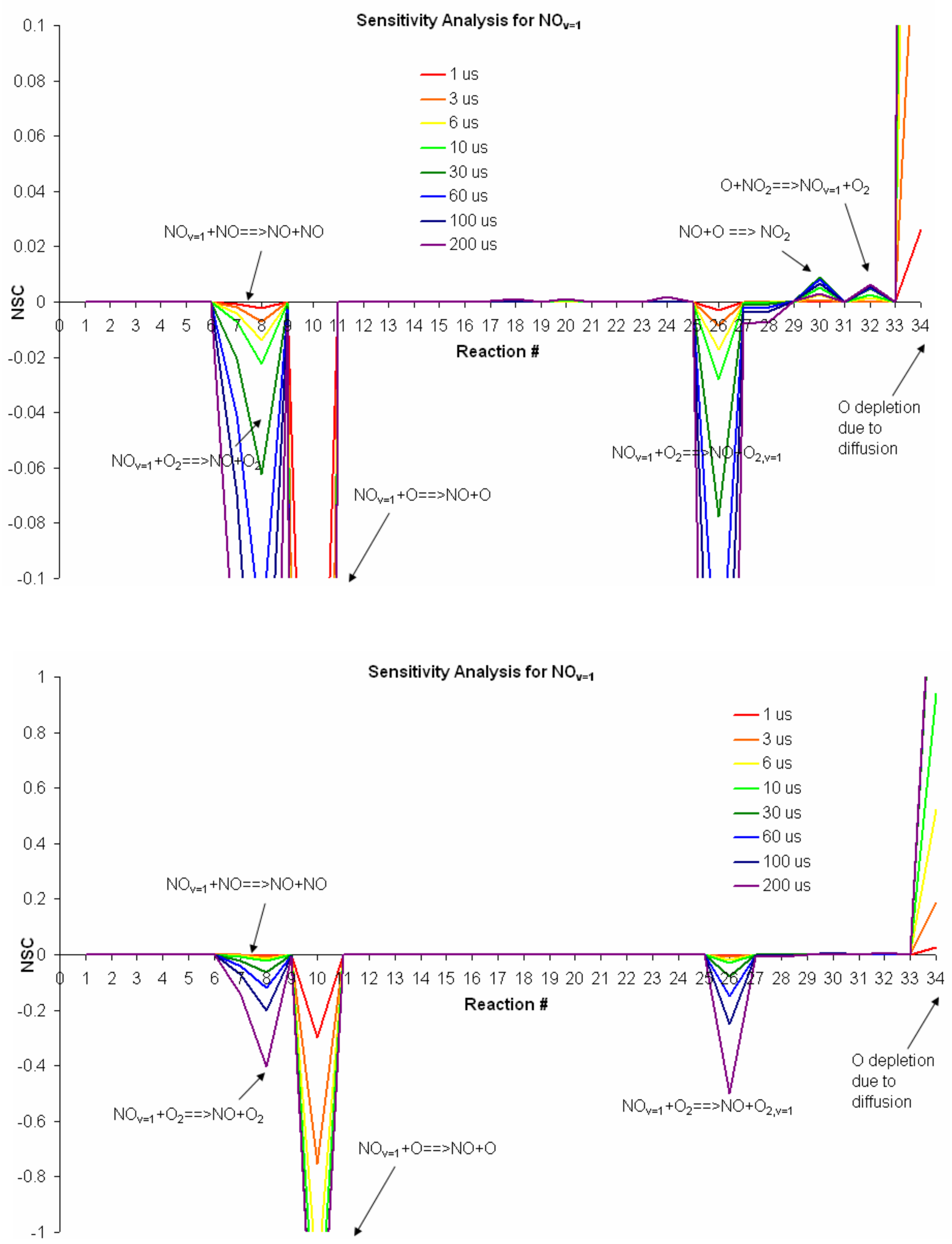


Fig. 5.32 Sensitivity analyses for $NO_{v=1}$

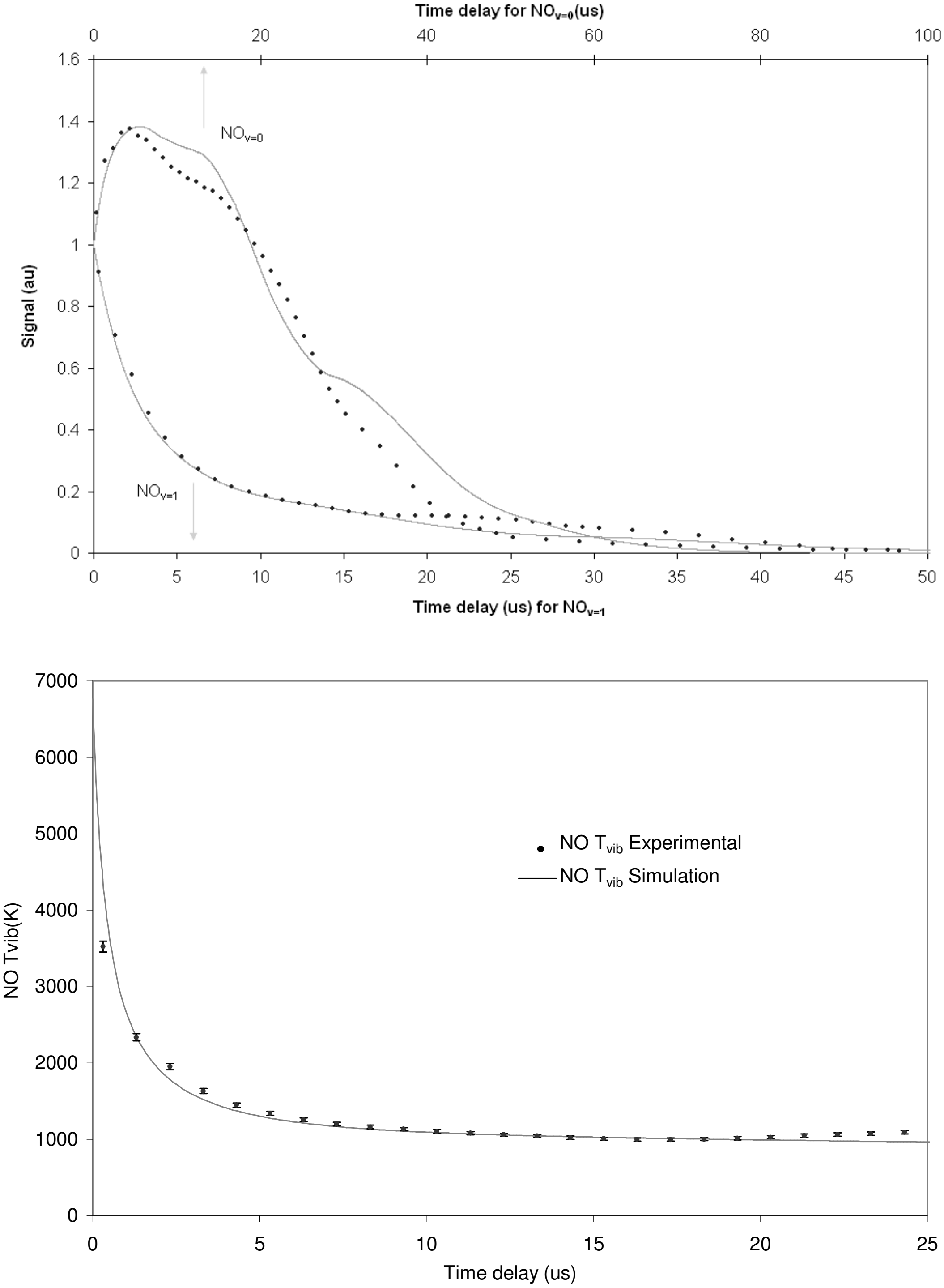


Fig. 5.33 Vibrational temperature decay modeling results

We further explored the possibility of modeling the raw $NO_{v=0}$ and $NO_{v=1}$ fluorescence signals. In the actual flowfield, diffusion and mixing effects cause an overall drop in signal which could not be accounted for in the initial kinetics simulations. In order to model these effects, the results from the kinetics simulations were multiplied by a diffusion-like term which took the form of $e^{-c(\Delta t^2)}$, where the constant, c, was adjusted to give best fit. These results were normalized to an initial value of 1, since the raw fluorescence counts from the ICCD camera do not correspond to the kinetics simulation output (molecules/cm^3). The experimental (points) and simulated signals (solid line) are presented in Fig. 5.33 (top), along with the corresponding T_{vib} (bottom).

This figure shows some interesting aspects of the flow. First, it should be pointed out that the kinetics modeling required pressure, or number density, profiles which were unavailable from the experimental data, and so the pressure profile from the CFD simulations was used, as mentioned before. Therefore, the locations of the compressions, as seen above, were not expected to match with the experimental results.

There are distinct oscillations in $NO_{v=0}$, but only weak oscillations in $NO_{v=1}$. This is due to the fact that the strong sink terms which deplete the population of $NO_{v=1}$ occur on a much faster time scale than the oscillations, whereas the population of $NO_{v=0}$ is mainly depleted by diffusion and mixing of fluid out of the laser sheet. Also, the locations of the oscillations for $NO_{v=0}$ match the location of oscillations in streamwise velocity, as we would expect, so that the experiment is self-consistent, but the locations of these experimental oscillations are slightly offset from those predicted through CFD simulations for both $NO_{v=0}$ and $NO_{v=1}$. Overall the fit was quite good.

The experimental data flattened near 1000K, which was reflected in the kinetics modeling. This limit stems from a combination of two effects. Computationally, the diffusion of O atom removes a major sink for vibrational relaxation of $NO_{v=1}$ through collisional quenching, causing $NO_{v=1}$ to persist for longer and maintain the NO vibrational temperature. The second reason was that experimentally, 1000K corresponded to a population ratio of $NO_{v=0}$:$NO_{v=1}$ of 0.06:1 and approached the signal to noise limit of the ICCD cameras. In fact, a slight rise in vibrational temperature caused by the signal levels of $NO_{v=1}$ reaching the noise limit of the detectors, and the continued slow diffusion and loss of $NO_{v=0}$. Therefore, the $NO_{v=0}$ signal continued to decrease while $NO_{v=1}$ was maintained at noise levels leading to an artificial temperature rise. Loss of overall signal in both $NO_{v=0}$ and $NO_{v=1}$ with time was caused by the combined effect of mixing of fluid (from the outer jet regions into the inner jet, causing fluid flow out of the laser sheet plane) and diffusion.

Last, the experimental profiles of NO T_{vib} did not show oscillations. This was due to the fact that the experimental data was self-consistent; the oscillations in $NO_{v=0}$ and weak oscillations in $NO_{v=1}$ matched in location and cancelled out in the calculation of NO T_{vib}. This conclusion was also reflected in the simulated T_{vib}, which also did not show oscillations, as expected.

E.4 Single Shot Image Analysis

The error analyses involved examination of multiple (40) single-shot pairs of images taken at nominal delays of 0, 5, 10, 15, 20, 25, and 30 μs. These images were analyzed for vibrational temperature uncertainties. To calculate statistics, the average of the single-shot images at 0 μs in the ambient region was set at the initial two-level Boltzmann temperature by applying a single scaling factor to all 40 images. This same scaling factor was used in all following nominal delays so that fluctuations about the means could be calculated. The image acquisition time for the single-shot images was short compared to the lifetime of Coumarin 450 dye so that minimal laser power drifting occurred during the experimental time. As expected, in general the vibrational temperature uncertainties were small when the nominal time delay was small, but as the nominal time delay was increased, the uncertainties increased. This increase in uncertainty was due to a decrease in signal to noise again due to diffusion and fluid movement out of the laser probe volume. The initial vibrational temperatures are much more sensitive to small fluctuations in laser power and beam profile than subsequent nominal time delays and so showed larger uncertainties at all regions of the flow. The averages and standard deviation for each set of data taken at each nominal delay are plotted in Fig. 5.34, where the plots show the statistics. The plots show the ambient region (top), the centerline region (middle), and the outer jet region (bottom). The bars in each plot are the average values ±1σ.

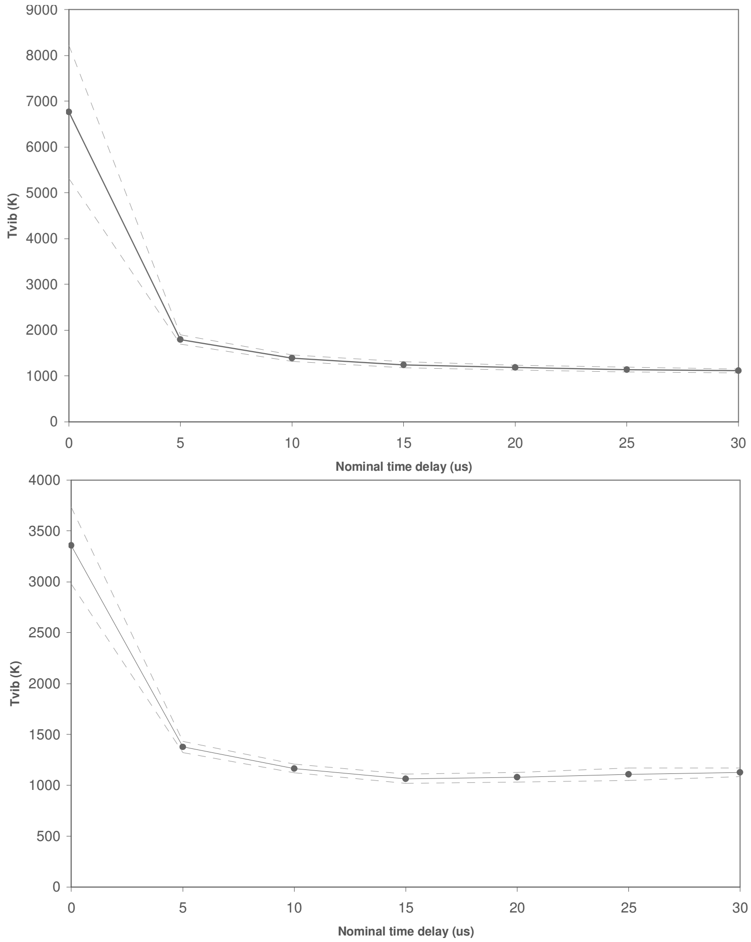


Fig. 5.34 Vibrational temperature uncertainties for single shot images for ambient (top) and centerline (bottom) regions

As seen from Fig. 5.34, besides the initial large uncertainties due to laser power fluctuation, the ambient region had the smallest uncertainties, and these uncertainties remained minimal as the nominal time delay was increased, as expected since this region of the flow was virtually undisturbed. The outer jet region displayed similar low uncertainties at roughly 5-10 μs, but gradually incurred larger uncertainties due to the natural transition of the flow to from laminar to turbulent. From Wilkes (2005), the flow should start to transition at 10 mm, or about a nominal delay of 23 μs for the centerline. However, for the outer jet regions, transition would occur prior to this point. Therefore, the fraction of this total uncertainty which was attributed to actual measurement uncertainties could not be determined. We assumed the centerline region was completely laminar and used this region for quantifying the true measurement uncertainty, as this assumption would give a conservative estimate for the vibrational temperature measurement uncertainty. The uncertainties for the centerline region decreased from 1σ of <12% at 0 μs to <5% at 10 μs and maintained this error until a slight rise to 6% at 25 μs due to decreased signal to noise. The small uncertainties associated with the single shot images proved single shot images very valuable for statistical analyses. Following this analysis, the error for the averaged images would be significantly smaller than the single shot uncertainties and would scale by the square root of the number of averaged images (in this case, 300 images during 30 seconds), giving averaged uncertainties of well less than 1%.

F. Summary and Recommendations

We have demonstrated the feasibility of photodissociation of NO_2 for two experiments demonstrated in an underexpanded jet. The first experiment demonstrated the use of probing photodissociated $NO_{v=1}$ in flowfields with naturally occurring NO, which would lead to background $NO_{v=0}$ fluorescence. These efforts represented the first known study to address the use of probing $NO_{v=1}$ for molecular tagging velocimetry to obtain one-component velocity maps. By probing $NO_{v=1}$, only the photodissociated NO is imaged, and the resulting images had much higher contrast than their $NO_{v=0}$ counterparts. This was true for all temperatures within the underexpanded jet (320K and below). At elevated temperatures (320K to 20,000K), the population of $NO_{v=1}$ would increase, and the advantage of probing $NO_{v=1}$ over $NO_{v=0}$ would lessen. At temperatures above 6000K, probing either $NO_{v=0}$ or $NO_{v=1}$ would produce MTV images with similar contrast. Even in low contrast conditions, background fluorescence could be minimized by subtracting out an integrated image of the flowfield in the absence of photodissociation. For the velocimetry calculations, a single photodissociation line was used and translated across the flowfield. At each location, $NO_{v=1}$ images at two time delays after photodissociation were collected for high spatial resolution, single-component (streamwise only) velocimetry analysis. The results recovered the expected streamwise velocity map of the underexpanded jet flowfield, and single shot 1σ uncertainties were about 5%.

The second experiment explored the mechanism of $NO_{v=1}$ vibrational decay using kinetics modeling. The main driving mechanism was found to be vibrational quenching by photodissociated O atom. Comparisons with CFD showed moderate agreement, although the locations of the oscillations following the Mach disk were slightly offset. However, the experimental results were self-consistent and were ultimately limited by loss of fluorescence signal and detector dynamic range. Error analysis using single-shot PLIF images produced single-shot vibrational temperature uncertainties of only a few percent. The present diagnostic technique could be extended to photodissociate a grid rather than a single line of NO_2 and then probing the photodissociated $NO_{v=1}$ to give two components of velocity and could be applied towards a variety of flowfields. This experiment is described in the next chapter.

CHAPTER VI

THE EXPERIMENTS: PHOTODISSOCIATION STUDIES 2, EXTENSION TO TWO COMPONENT MOLECULAR TAGGING VELOCIMETRY UTILIZING NO_2 PHOTODISSOCIATION AND NO PLIF

A. DIAGNOSTIC TECHNIQUES AND BACKGROUND

As discussed in the Chapter I, MTV has been applied using both single-line and grid techniques. The single-line techniques provide only one component of velocity, while the grid techniques give two components of velocity in the laser plane by observing the warping of the grid, i.e. the movement of the intersection points of the grid. Gaseous grid studies for two-component velocimetry are exceedingly rare in the literature and limited to only a few studies using biacetyl [Ismailov (2006), Stier (1999)] and OH [Ribarov (2004). The present experiment examines the application of MTV towards two-component velocimetry in a variety of quenching environments and focuses on the advantages and disadvantages of MTV techniques in each. The technique relies on the formation of two co-planar, perpendicular laser sheets into a grid, and the tracking of the grid (more specifically, the intersection points of the grid) via imaging onto ICCD detectors. The velocity is calculated simply by calculating the displacement (warping) of the grid and dividing this value by the known time delay between two images. Two types of MTV will be explored. The first is the acquisition of both the initial and time delayed image during a single fluorescence lifetime of seeded NO using a two-laser, two camera setup ("fluorescence lifetime" technique). The second utilizes photodissociation of seeded NO_2 to create NO, which is then probed using two temporally separated lasers for collection of the initial and time delayed image: a three-laser, two camera setup ("photodissociation" technique). The advantages of the different setups in different quenching environments will be examined. The study represents the first known application of molecular tagging velocimetry utilizing either the fluorescence lifetime or photodissociation technique towards two-component velocity mapping.

Resolution issues will also be addressed, since it is one of the main shortcomings of MTV as compared to PIV. Whereas PIV resolution is limited by the seeding density of the particles, molecular tagging resolution is primarily limited by SNR (low SNR decreases the sharpness of the gridding lines). The signal to noise directly limits the resolution in that the noisier the images, the higher the gain on the ICCD must be used for imaging, and subsequently, the fewer line pairs that can be distinguished in the images. In addition, noisier images require increased smoothing so that the software can detect intersection points, and the increased smoothing inherently decreases spatial resolution. Techniques for increasing spatial resolution through image processing and through increasing signal to noise to levels comparable to PIV are discussed. Lastly, single shot images for application towards short or single-shot facilities such as blow-down wind tunnels or shock tubes will be obtained for uncertainty analyses. The experiments were carried out at the National Aerothermochemistry Laboratory at Texas A&M University.

B. Experimental Flowfield and Setup

The flowfield that was examined for demonstration of this technique was again the underexpanded jet flowfield. However, instead of imaging the flowfield after the Mach disk, as was done in the previous chapter, the imaged region consisted of the region from the nozzle to just after the Mach disk. This region was chosen for several reasons. First, the flowfield is well-studied, and the streamwise and radial velocity maps can be calculated and compared to experimental results. Second, this region has a significant radial velocity component, whereas the region after the Mach disk is primarily composed of a streamwise component (which was a primary reason why it was chosen for demonstration of the single-line, streamwise velocimetry technique). Third, this flowfield is challenging in that it presents wide variety of temperatures, pressures, and velocities, and is ideal in demonstrating the robustness of the velocimetry technique in a variety of conditions. The main features of the flow have already been discussed in the previous chapter.

The experimental setup was similar to that shown in the previous chapter and consisted of a vacuum chamber which was fitted with a stainless steel nozzle to create the highly underexpanded jet flowfield, quartz windows on four sides to allow optical access, and a vacuum line. An additional quartz window was added opposite the nozzle to pass the horizontal laser sheet. The specific experimental setups for each sub-experiment will be discussed in the following sections.

The nozzle construction and pumping configuration were identical to that used in the previous chapter. The gas mixture used was a 1%NO in N_2 gas mixture and was supplied by Matheson Tri-Gas. The NO_2 was supplied by Sigma Aldrich (>99.5% pure). The NO_2 contained a small amount of NO impurity. The air used is a 0% hydrocarbon dry compressed air supplied by Brazos Valley Welding. The gases were delivered via needle valves into the chamber. The reported experiments were carried out at the following conditions listed in Table 6.1:

Table 6.1 Two-component velocimetry experimental run conditions

Run	P_a (Torr)	P_0 (Torr)	JPR (calc)	T_o (K)	N_2	O_2	NO	NO_2	Images	Purpose
1	2.5	455	96.15	300	0.99	0	0.01	0	Integrated, Single shot	Low quenching
2	2.5	450.2	95.13	300	0.894	0.106	0.05	0	Integrated, Single shot	High quenching
3	3.3	523.8	83.85	300	0.759	0.196	0	0.045	Integrated	Photodissociation
4	2.7	463	90.59	300	0.742	0.191	0	0.067	Single shot	Photodissociation

The purpose of the experiments is to illustrate the versatility of the 2-component velocimetry technique in a variety of quenching conditions for use in a variety of flowfields. The experimental setup differs slightly for each run and will be delineated here:

B.1 Run 1: Low Quenching Flow Conditions

The schematic layout of the setup with a closeup of the field of view is shown in Fig. 6.1.

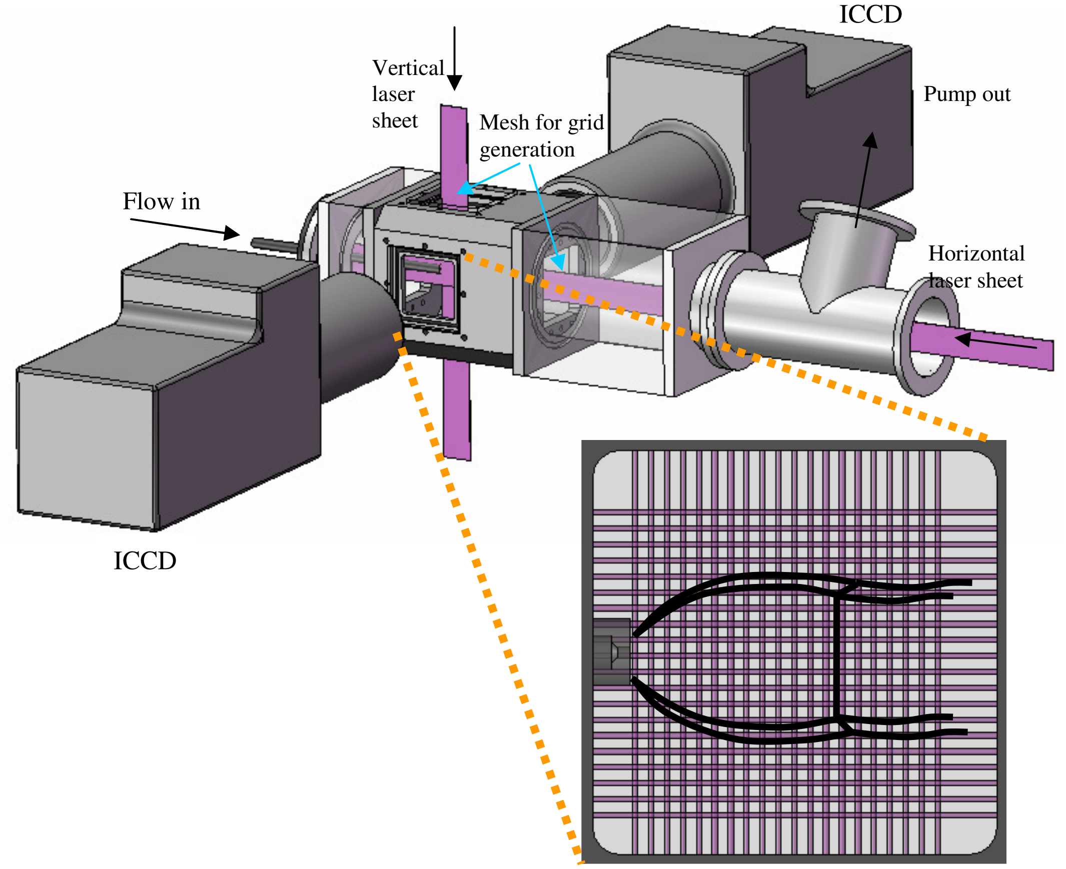


Fig. 6.1 Schematic of experiment for 2-component velocimetry, Runs 1 and 2, (low and high quenching, MTV during a single fluorescence lifetime)

The laser system used consisted of two 355 nm- pumped (Spectraphysics LAB 150-10 Nd:YAG at 10Hz) dye lasers (Sirah Cobrastretch) tuned to probe the $A^2\Sigma^+_{1/2}(v'=0) \leftarrow X^2\Pi_{1/2}(v''=0)$ transition of NO at R_1+Q_{21}(2.5). On average, the doubled UV laser light exiting the dye lasers was about 1 to 2 mJ/pulse. Both laser systems were triggered to fire simultaneously and the UV laser sheets were combined at the chamber for maximum laser power. The probed rotational states were located using spectral simulation software LIFbase [Luque (1999)]. The collimated laser sheets were formed by the appropriate combination of long-focal length diverging cylindrical, converging cylindrical, and spherical UV fused silica, uncoated lenses from Thorlabs. The combination of lenses differed for both lasers as the divergence of the dye lasers differed slightly. The waist of the laser sheet was located within the field of view of the cameras to minimize the width of the "write" line. Finally, the grid was created by placing a small square of aluminum mesh (which the author is very grateful to have received from Mike Semper and his magic stock of miscellaneous McMaster-Carr items) in both laser sheet paths as close to the underexpanded jet as possible to reduce the effect of diffraction patterns causing the blurring of the grid lines. For the vertical laser sheet, the mesh was placed on top of the top window, while for the horizontal laser sheet the mesh was actually mounted inside of the chamber close to the flowfield, roughly 4 inches from the nozzle exit. For this study, since the goal was to recover the velocity field, there was no concern to remain within the linear fluorescence regime. Therefore, more laser power would have been desired.

The imaging system consisted of a single water-cooled Andor iStar ICCD camera (model DH734, 16-bit) fitted with a Nikon 105mm F/4.0 UV lens and extension rings for close-up viewing. The ICCD camera was externally triggered and gated for 20 ns and set at maximum aperture. The resulting images were not binned and were read at full resolution of 1024 x 1024 pixels for a full image field-of-view of 13.6 mm x 13.6 mm. The full resolution, non-binned image was chosen to maximize velocity (i.e. spatial) resolution at the cost of lower signal to noise. The gain on the camera was adjusted to give maximum signal. The overall timing for the entire experiment was controlled by a digital delay generator (Berkeley Nucleonics Corporation, model DG565). The ICCD camera was triggered with an internal delay (about 10 ns after the arrival of the laser sheet at the flowfield) to avoid elastic scatter.

The purpose of Run 1 was to show the use of 2-component velocimetry in low-quenching environments, where fluorescence signal was long-lived (a lifetime of about 200 ns). Therefore, the goal involved a pair of images: one initial image (slightly time-delayed to avoid elastic scatter) which captured the fluorescence directly after the probe pulse arrives at the chamber and one time-delayed image which imaged the remaining fluorescence. The timing for this setup is shown in Fig. 6.2:

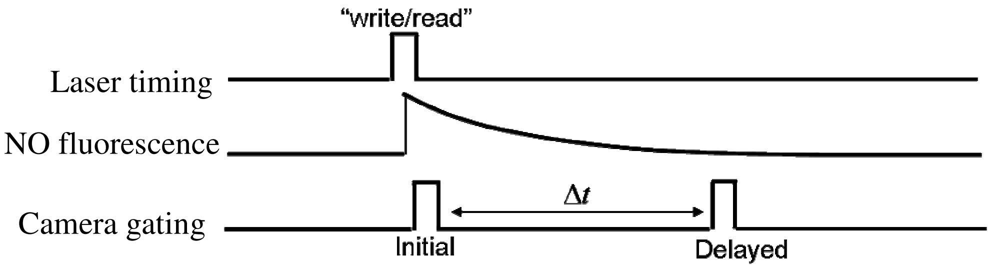


Fig. 6.2 Timing diagram for MTV images collected for a single fluorescence lifetime (Runs 1 and 2)

The noisier image is the time-delayed image, which suffers from fluorescence quenching. To evaluate the optimum time delays, fluorescence images were captured at various time delays, with a short exposure time (20 ns) from 0 ns to 1 μs and run through the data analysis software to find the optimal time delay between the two images. This process involves finding an optimum time delay where the flowfield has moved an appreciable amount to increase velocity resolution, but not too far where loss of fluorescence signal leads to significant loss of signal to noise and the lines become blurred (preventing the software from locating the intersection points). Through repeated data sets, 0 and 400 ns, and 0 and 300 ns, were chosen as the optimal temporal pairs. This temporal separation optimized the velocity resolution in the hypersonic (isentropic expansion) region of the flow shortly before the Mach disk. The 300 ns delayed image offered slightly more signal and was used for the single shot images while the 400 ns delayed image offered more velocity resolution and was used for the integrated images. 60 s integrated and single shot images were taken at these conditions and analyzed.

Since the underexpanded jet flowfield contains a wide range of temperatures, the choice of the probed rotational line was important. According to LIFbase [Luque (1999)], at temperatures of 20K which may be encountered in the isentropic expansion region, the Boltzmann population largely populated at rotational states of J”=3.5 and below. However, at 300K, the Boltzmann population peaks at the J”=7.5 rotational state. Therefore, to maximize the signal to noise in the flowfield in all temperature regions, a probed rotational state of J”=2.5 was chosen. The effect of Doppler shifting was also considered. For the vertical laser sheet, slight detuning of the laser wavelength resulted in an asymmetrical fluorescence image; therefore, the laser wavelength was carefully tuned to the line center by observing the fluorescence image symmetry. For the horizontal laser sheet, Doppler shifting posed a more difficult problem, since the streamwise velocity in the isentropic expansion region (700-800 m/s) and immediately after the Mach disk (<300 m/s) were drastically different. Tuning the laser to line center in either portion resulted in loss of signal in other regions. To maximize signal in the overall image, the laser was slightly detuned so that signal in both the isentropic region and subsonic region after the Mach disk were roughly balanced. Unfortunately, this resulted in an overall lower intensity (about 50%) of the horizontal laser sheet as compared to the vertical laser sheet. Raw 60 s and single shot images for Run 1 are shown in Fig. 6.3 and Fig. 6.4:

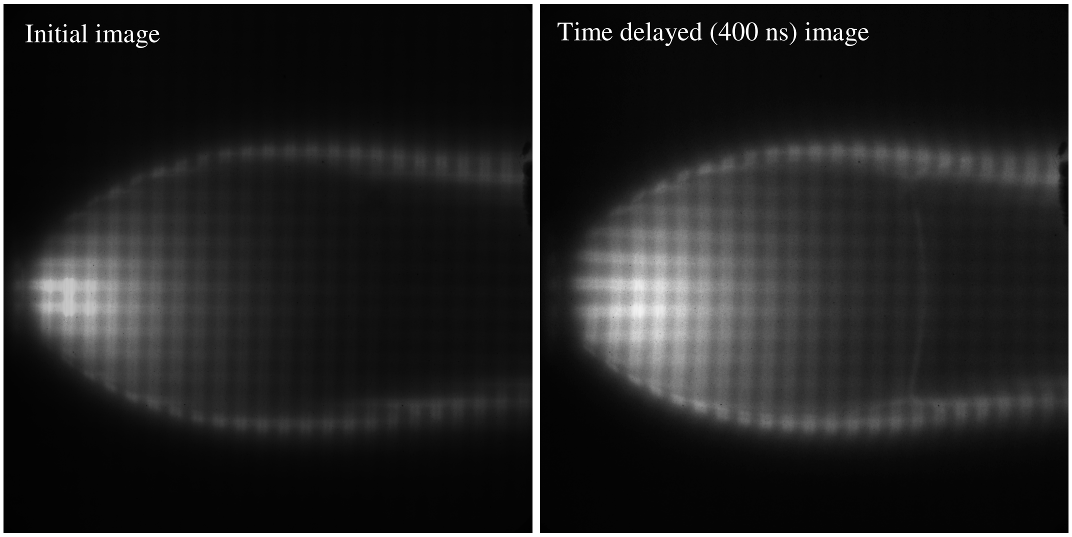


Fig. 6.3 Run 1, 60 s integrated images (low quenching)

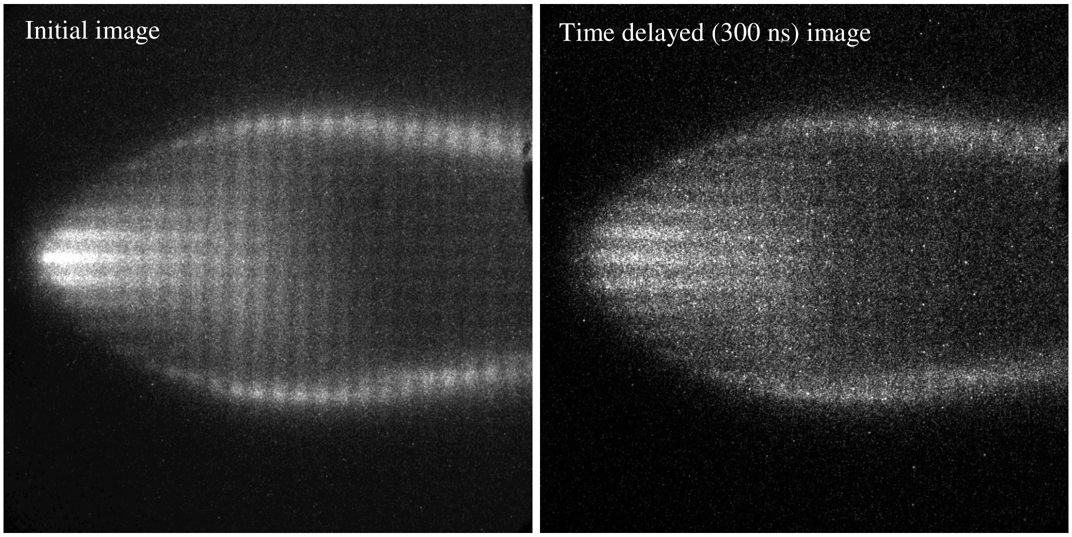


Fig. 6.4 Run 1, single shot images (low quenching)

B.2 Run 2: High Quenching Flow Conditions

The purpose of Run 2 was to showcase differences encountered using the velocimetry technique discussed for Run 1 in high-quenching environments (i.e. environments at high pressure, or in the presence of a species which rapidly quenches NO fluorescence such as O_2, H_2O, or CO_2) which would shorten the fluorescence lifetime significantly. In these conditions, the time-delayed image SNR suffers. The experimental setup for Run 2 was the same as that of Run 1, except high quenching species were introduced: O_2 partially replaced the N_2 (O_2 quenches NO fluorescence with a rate constant of 1.62 x 10^{-10} cm^3/molecule/s and a higher concentration of NO was used (NO-NO quenching rate is 2.74 x 10^{-10} cm^3/molecule/s) as listed in Paul (1996). Two ICCD cameras were used in this run, and integrated and single shot data were collected at 0 and 300 ns. The choice of rotational lines for this Run was the same as for Run 1. Raw 60 s and single shot images for Run 2 are shown in Fig. 6.5 and Fig. 6.6:

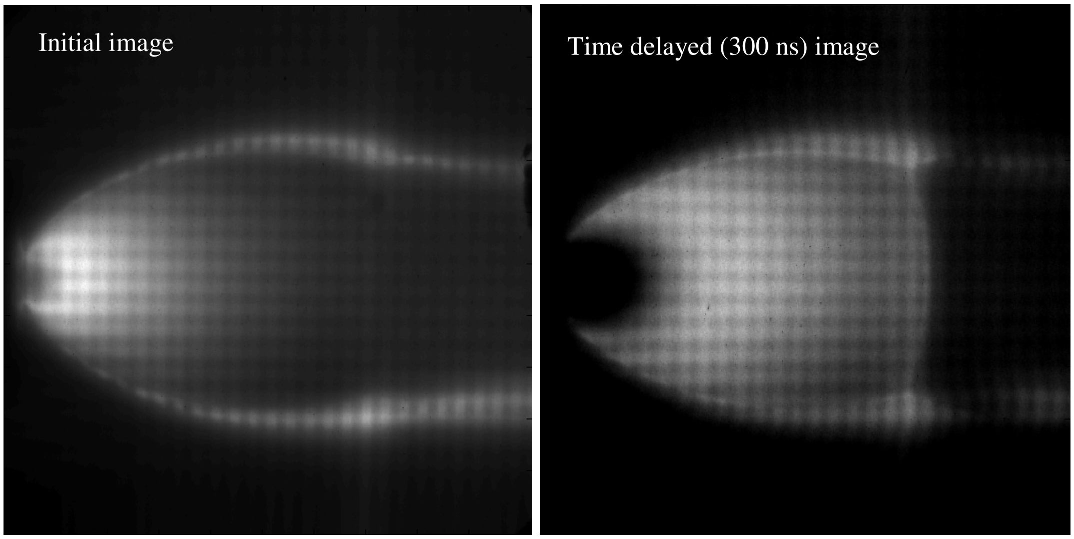


Fig. 6.5 Run 2, 60 s integrated images (high quenching)

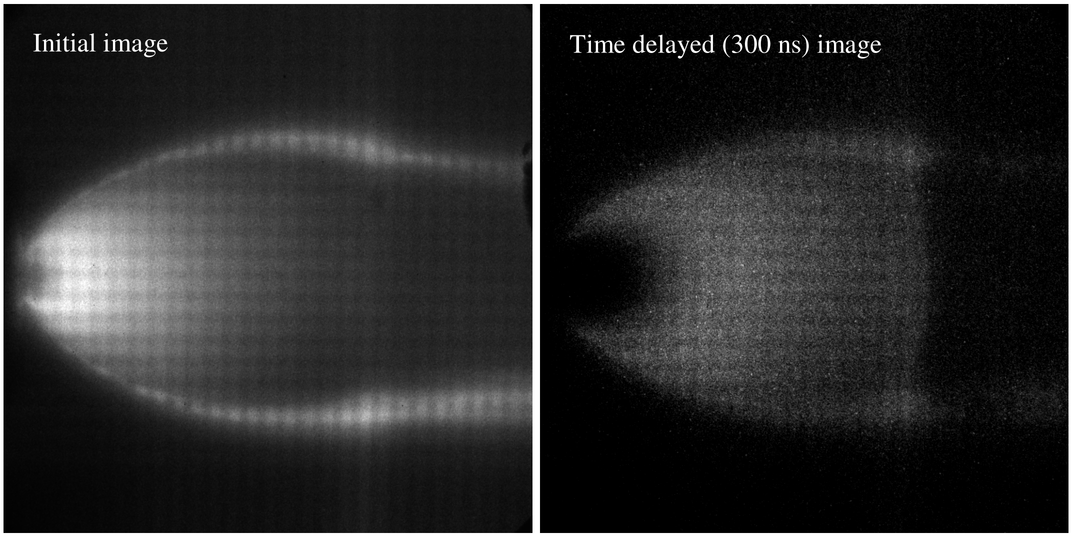


Fig. 6.6 Run 2, single shot images (high quenching)

B.3 Runs 3 and 4: Photodissociation at High Quenching Conditions

The purpose of Runs 3 and 4 were to show that in high-quenching environments, photodissociation of seeded NO_2 can be used to overcome the difficulties encountered in

Run 2 due to loss of signal in the time-delayed image. The experimental setup is more demanding in this case and requires three lasers. The schematic is shown in Fig. 6.7:

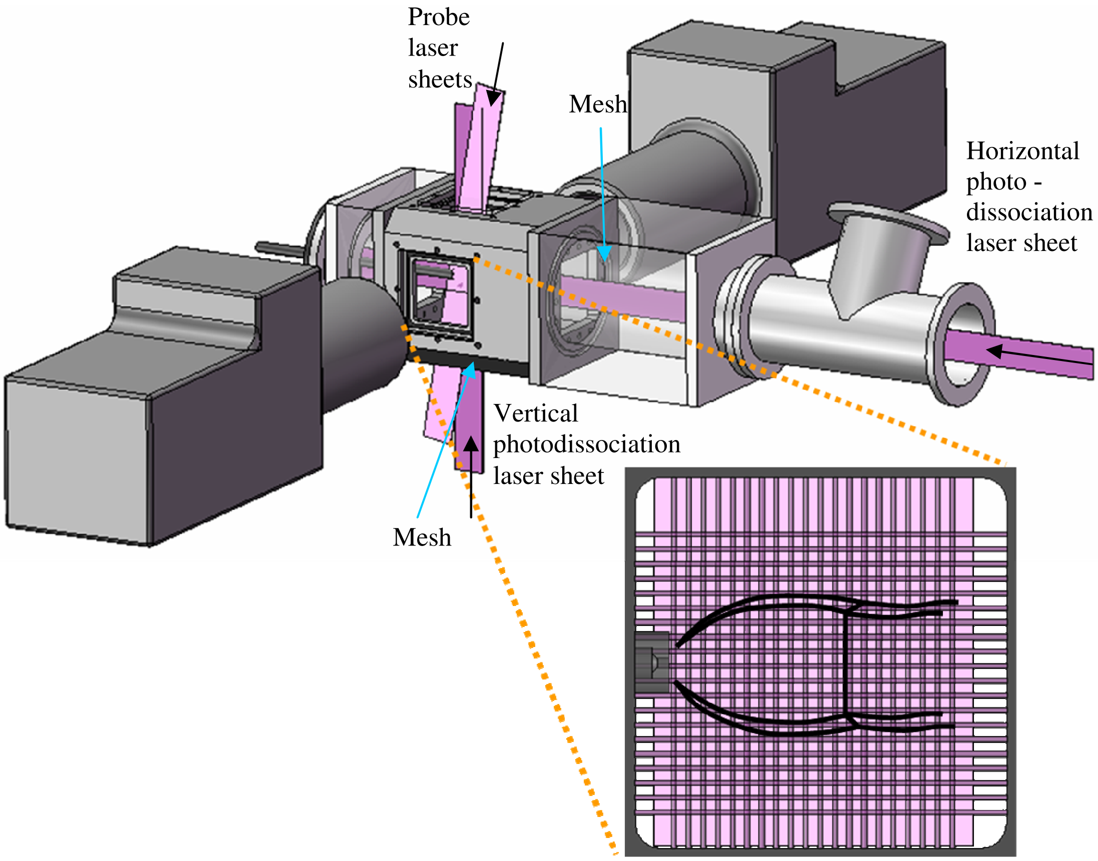

Probe laser sheets
Mesh
Horizontal photo - dissociation laser sheet
Vertical photodissociation laser sheet
Mesh

Fig. 6.7 Schematic of experiment for 2-component velocimetry, Runs 3 and 4, (high quenching, MTV using photodissociation of NO_2)

First, the seeded NO_2 was photodissociated at 355 nm (“write” laser) using a Spectraphysics LAB 290-10 Nd:YAG (10 Hz). The 355 nm light was split using a 50:50 beamsplitter and one arm entered the chamber vertically from bottom to top, while the other entered the chamber horizontally. Both arms passed through the aluminum mesh and entered the chamber at the same time, writing a grid of photodissociated NO into the flowfield. The 355 nm laser power used for Run 3 was about 80 mJ/pulse, and about 400 mJ/pulse for Run 4. The calculated NO_2 photodissociation fraction was greater than

99.9% for both cases. The first probe "read" laser (355 nm pumped (Spectraphysics LAB 150-10 Nd:YAG at 10Hz) dye lasers (Sirah Cobrastretch) tuned to probe the $A^2\Sigma^+_{1/2}(v'=1) \leftarrow X^2\Pi_{1/2}(v"=1)$ transition of NO at R_1+Q_{21}(1.5) was formed into a sheet and entered the chamber vertically from top to bottom (and therefore remained un-gridded) 100 ns after photodissociation. One of the ICCD cameras captured this fluorescence, with a 20 ns gate for the integrated images (Run 3), and a 100 ns gate for the single shot images (Run 4). At 400 ns after photodissociation, the second probe laser (355 nm pumped (Spectraphysics LAB 150-10 Nd:YAG at 10Hz) dye lasers (Sirah Cobrastretch) tuned to probe the $A^2\Sigma^+_{1/2}(v'=1) \leftarrow X^2\Pi_{1/2}(v"=1)$ transitions of NO at Q1(1.5, 2.5, and 3.5) overlapping lines entered the chamber in a similar way as the first probe laser. The timing diagram for this setup is shown in Fig. 6.8:

The resulting fluorescence was captured with again 20 ns and 100 ns gates for the integrated and single shot images, respectively. In this way, because the time delayed image is not dependent on the fluorescence lifetime (since two separate probe lasers are used), this technique utilizing photodissociation of NO_2 is immune to fluorescence quenching effects. Since both probe beams enter the chamber vertically, the fluorescence image is largely unaffected by Doppler shifting issues described above in Runs 1 and 2. However, the required wavelength shifted for probing the $NO_{v=1}$ transition, at about 224 nm rather than 226 for $NO_{v=0}$, results in a decrease in laser power to about 1 mJ/pulse for each laser because of the narrow dye curve, and, as opposed to Runs 1 and 2, the laser power is not combined. This results in a slightly lower SNR; more laser power would have been desired. But, probing $NO_{v=1}$ instead of $NO_{v=0}$ ensures that any background NO in the flowfield does not appear on the fluorescence images as discussed in the previous chapter. This would be important for flowfields where background NO may be a problem, such as combustion flowfields. In this case, the NO_2 provided contained a small impurity of NO which was visible in the collected images as background fluorescence. Therefore, in these studies, $NO_{v=1}$ was probed rather than $NO_{v=0}$. Raw 60 s and single shot images for Runs 3 and 4 are shown in Fig. 6.9 and Fig. 6.10:

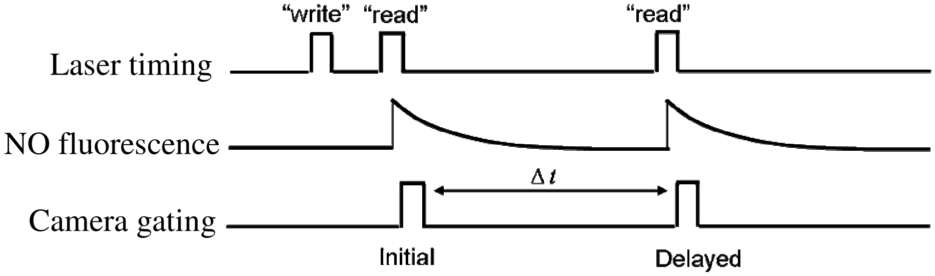


Fig. 6.8 Timing diagram for MTV imaging collected for photodissociation setup (Runs 3 and 4)

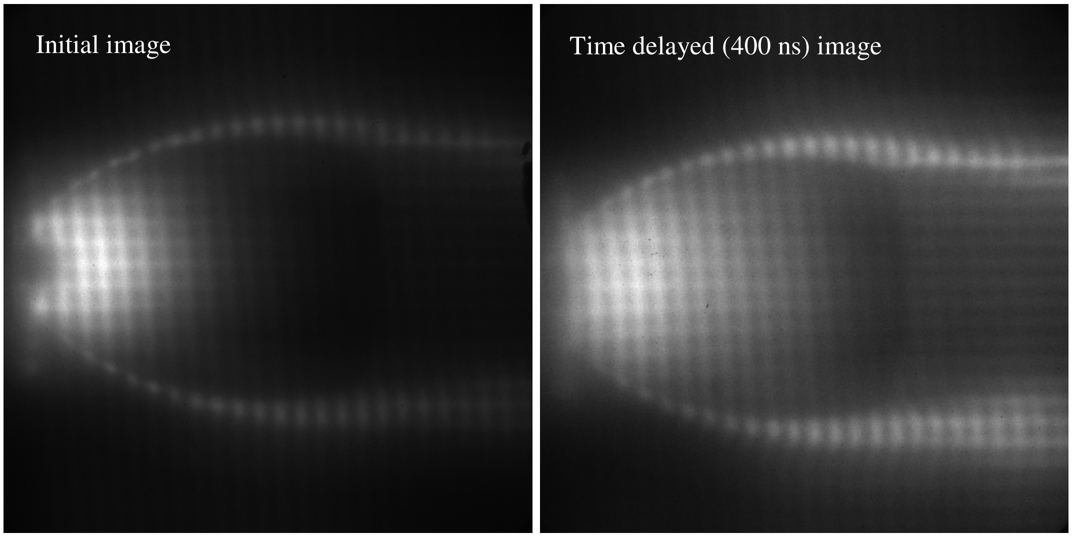


Fig. 6.9 Run 3, 60 s integrated images (high quenching, photodissociation)

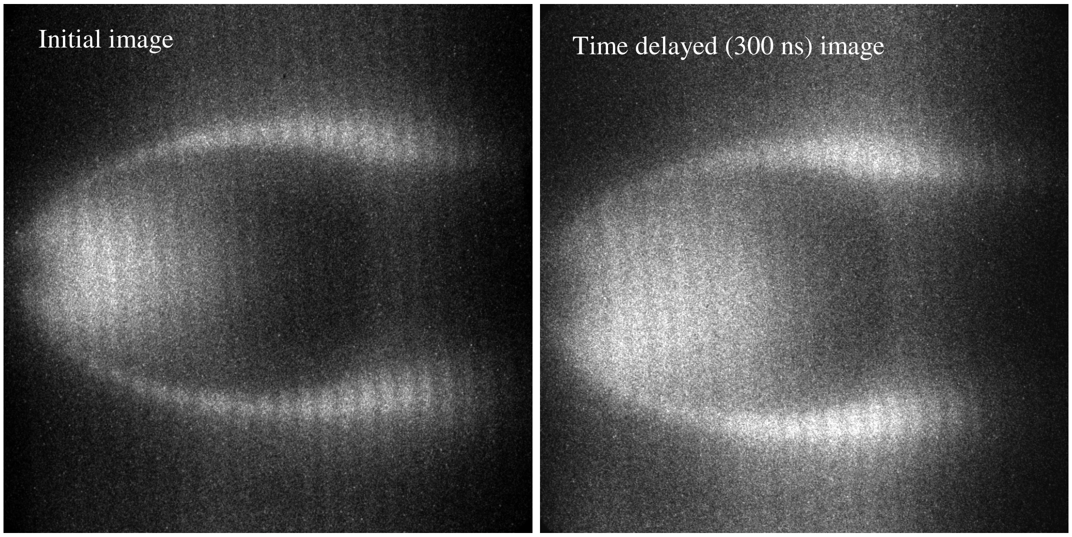


Fig. 6.10 Run 4, single shot images (high quenching, photodissociation)

Because the 355 nm power was increased in the single shot images (Run 4) for better SNR, a large amount of ablation of the nozzle surface was encountered from the horizontal grid impacting the face of the nozzle which could not be eliminated. The ablation of the stainless steel nozzle caused a visible emission which could be seen with the eye. It was thought that the ablation caused the formation of a localized plasma, where the excited species were long lived and entrained into the flow. A digital photo of this visible emission is shown in Fig. 6.11.

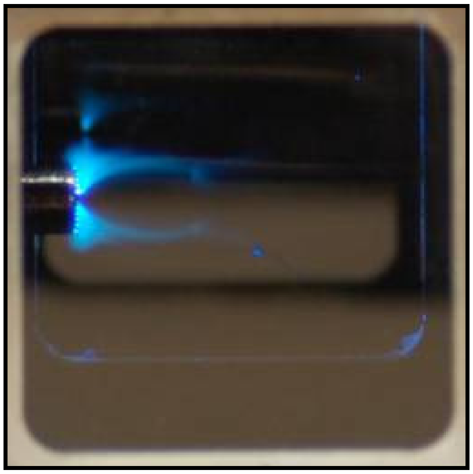


Fig. 6.11 Visible emission from laser ablation of nozzle surface (digital photo)

Although the ablation made for beautiful visible images and outlined the shape of the underexpanded jet, the plasma showed up as saturated pixels on the ICCD images, especially near the nozzle face, corrupted data in these regions. Therefore, only the vertical photodissociation laser sheet was used in this run, and so only streamwise velocity maps were calculated.

B.4 Examination of Short Exposure Camera Gating

To verify that the microchannel plate (MCP) gate widths on the camera were true to specifications, a quartz plate was placed close to the nozzle and elastic laser scatter off of the quartz plate was imaged while the camera delay was temporally swept at 500 ps steps across the scattered signal. The resulting experimental signal was a convolution of the camera gate and the laser pulse temporal profile. Assuming a Gaussian 8 ns laser pulse temporal profile and a square camera gate, the square gate width was altered until the convolution of the laser pulse and camera gate equaled the experimental width. For this study, the width was defined as the FWHM, or 50% open. At a nominal 20 ns gate width, the square gate width needed was 19 ns, very close to specifications. At a nominal 15 ns gate width, the square gate decreased to 13.5 ns, slightly less than specification. At 10 ns, the square width was 8 ns, and there was a slight double-shuttering phenomena. In other words, the experimental signal showed two peaks as the gate was swept across the scattered light, indicating that the MCP voltage could not be accurately controlled during the gating period, or that the MCP was not charged uniformly across the image. The 5 ns gate width displayed the same behavior. Therefore, it was advised that the gate be maintained at 15 ns or above, to avoid the double-shuttering behavior of the ICCD cameras at shorter gates.

C. Method and Image Analysis

The image analysis software was written in-house using MATLAB. This section describes in detail how the velocity map was calculated using this software. A series of printscreens at each stage of the image processing are provided. Before the velocity map could be calculated, the spatial conversion was first determined by placing an electroformed mesh (19 lines/inch) in the path of the laser sheet and taking an integrated image of the flowfield, which then showed distinct bright and dark lines. This calculation yielded a spatial conversion in pixels/mm and was used to convert displacement in pixels to millimeters. Then, the velocity map was calculated.

One unique aspect of the software was that instead of simply locating the intersection points as is traditionally done in MTV image analysis, each intersection point was found and then broken down into four individual points via edge-finding routines, quadrupling the spatial resolution. In this way, spatial resolution was increased to levels which were comparable to particle-based tracking techniques such as PIV. This procedure is illustrated in Fig. 6.12:

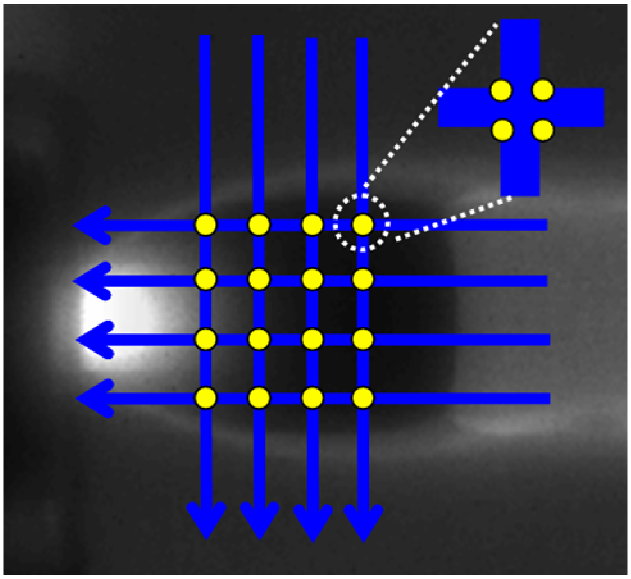


Fig. 6.12 Increased resolution routine via edge finding routines

In the following paragraphs, each step of the image analysis software will be discussed in detail. As examples, the images presented were taken directly from the data processing of Run #1 (integrated images).

In the first step, edge finding routines were used the find the edges of the horizontal and vertical laser grids. Shown in Fig. 6.13 and Fig. 6.14 are screen shots of the procedures performed on the 0 ns and 400 ns raw images. The raw image for each is shown in top left of their respective figures. First, the gradients in the streamwise and radial directions were taken and added together, and the region of interest was specified by the user (top row, middle and top right columns). Decreasing the region of interest increased processing speed significantly. The streamwise and radial gradients (middle and bottom rows, left column) were smoothed (middle and bottom rows, middle column) and then the edges were located by a peak-finding procedure which located both the peaks (rising edges) and valleys (falling edges) shown in the middle and bottom rows, right column, as green and red stars. The amount of smoothing was user-specified and kept to a minimum so that each calculated velocity vector maintained its measurement individuality.

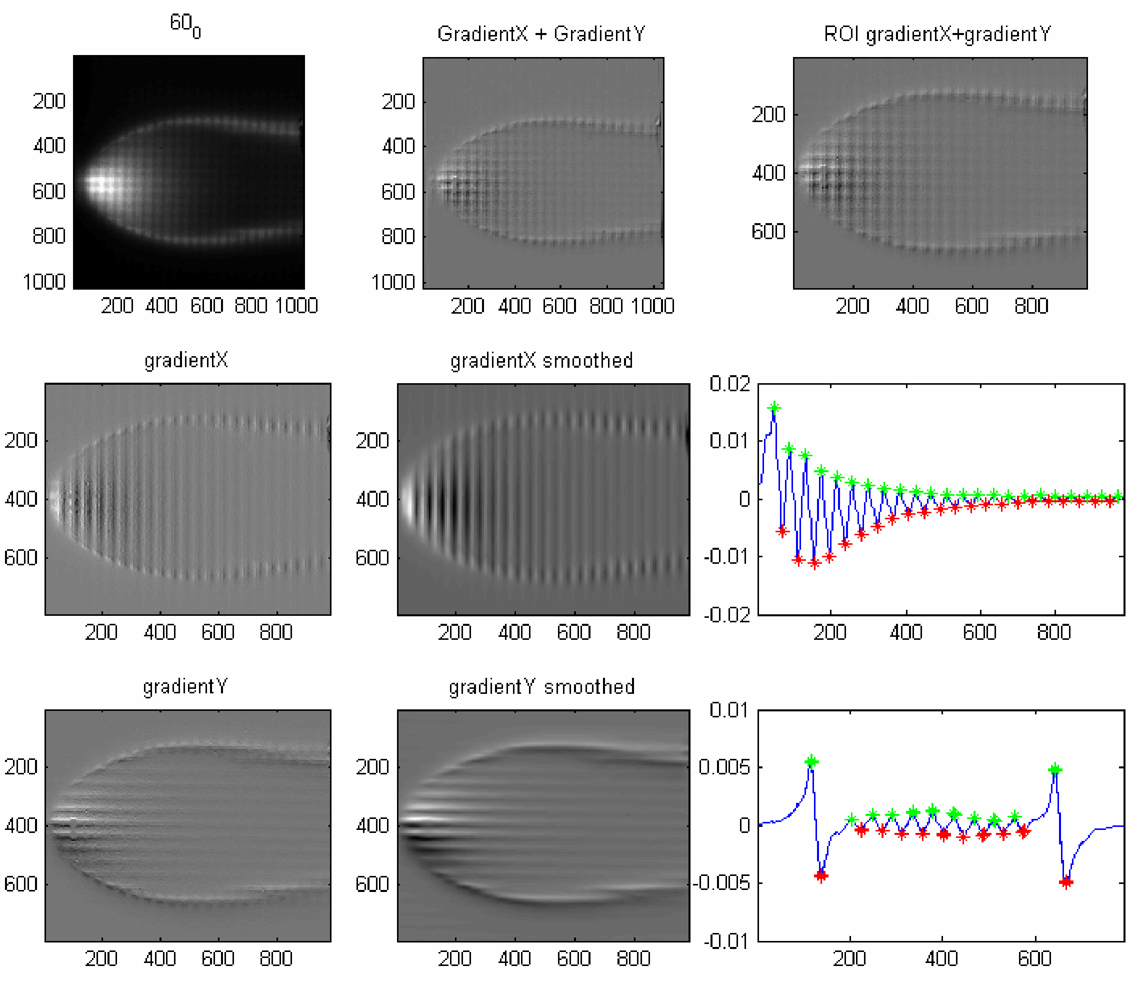


Fig. 6.13 Screenshot of edge finding procedures for 0 ns (initial image)

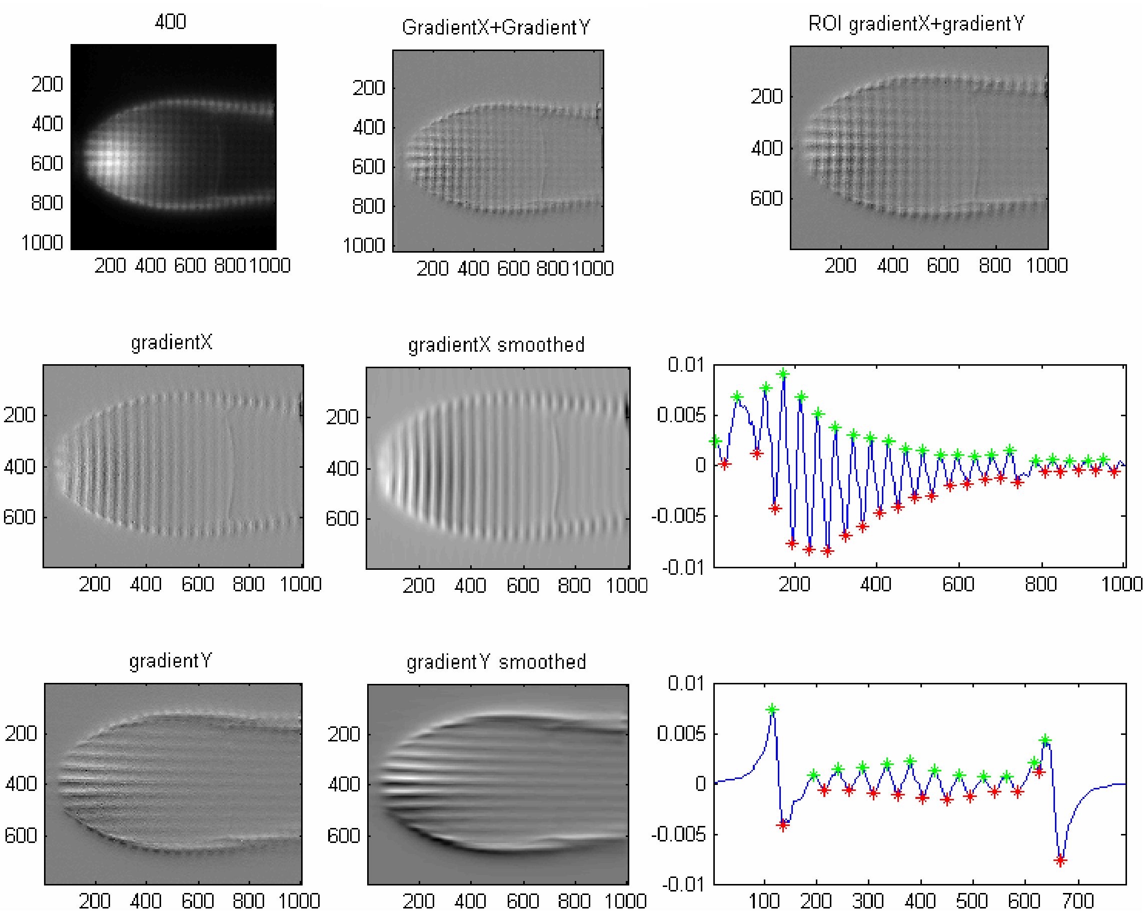
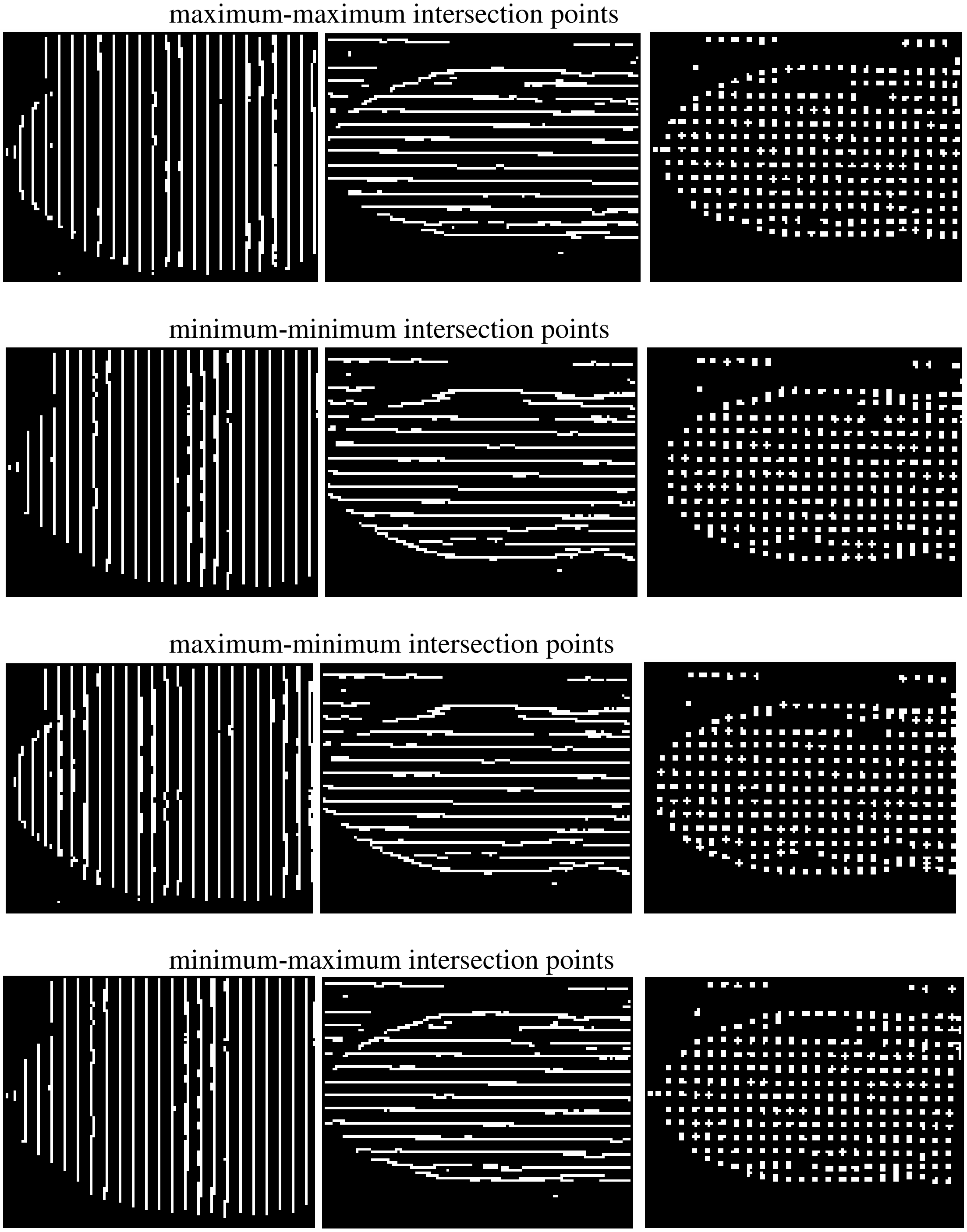


Fig. 6.14 Screenshot of edge finding procedures for 400 ns (time delayed image)

Fig. 6.15 and Fig. 6.16 show the resulting map after the edge-finding routine was employed for the 0 ns image and for the 400 ns image. The left column are the maximums and minimums in the streamwise gradients. The middle column are the maximums and minimums in the radial gradient. The software added these two images together and located the intersection points for each the four combinations, which represented the four combinations for each intersection point. These are shown in the right panels.



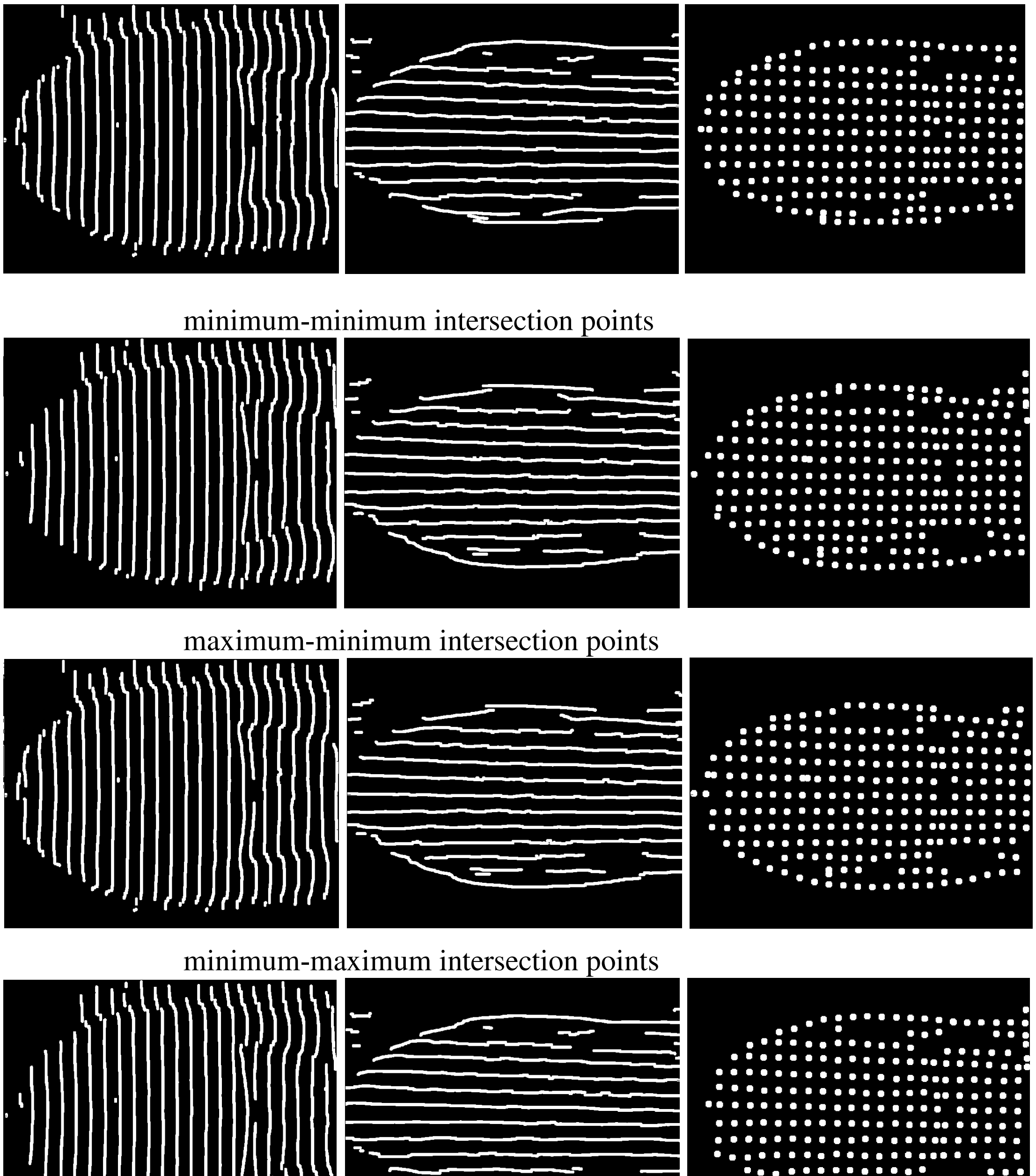
maximum-maximum intersection points

minimum-minimum intersection points

maximum-minimum intersection points

minimum-maximum intersection points

Fig. 6.15 Locations of intersections for 0 ns (initial image)



maximum-maximum intersection points

minimum-minimum intersection points

maximum-minimum intersection points

minimum-maximum intersection points

Fig. 6.16 Locations of intersections for 400 ns (time delayed image)

Then, the intersection points for each of the four combinations (maximum-maximum, minimum-minimum, maximum-minimum, minimum-maximum) for 0 ns and 400 ns were plotted overlaid. The four plots are shown in Fig. 6.17. The locations of the intersection points for 0 ns are shown in blue, and the corresponding points at 400 ns are shown in red. Visual inspection showed that the velocity increased as the gas expands, and then slowed directly after the Mach disk.

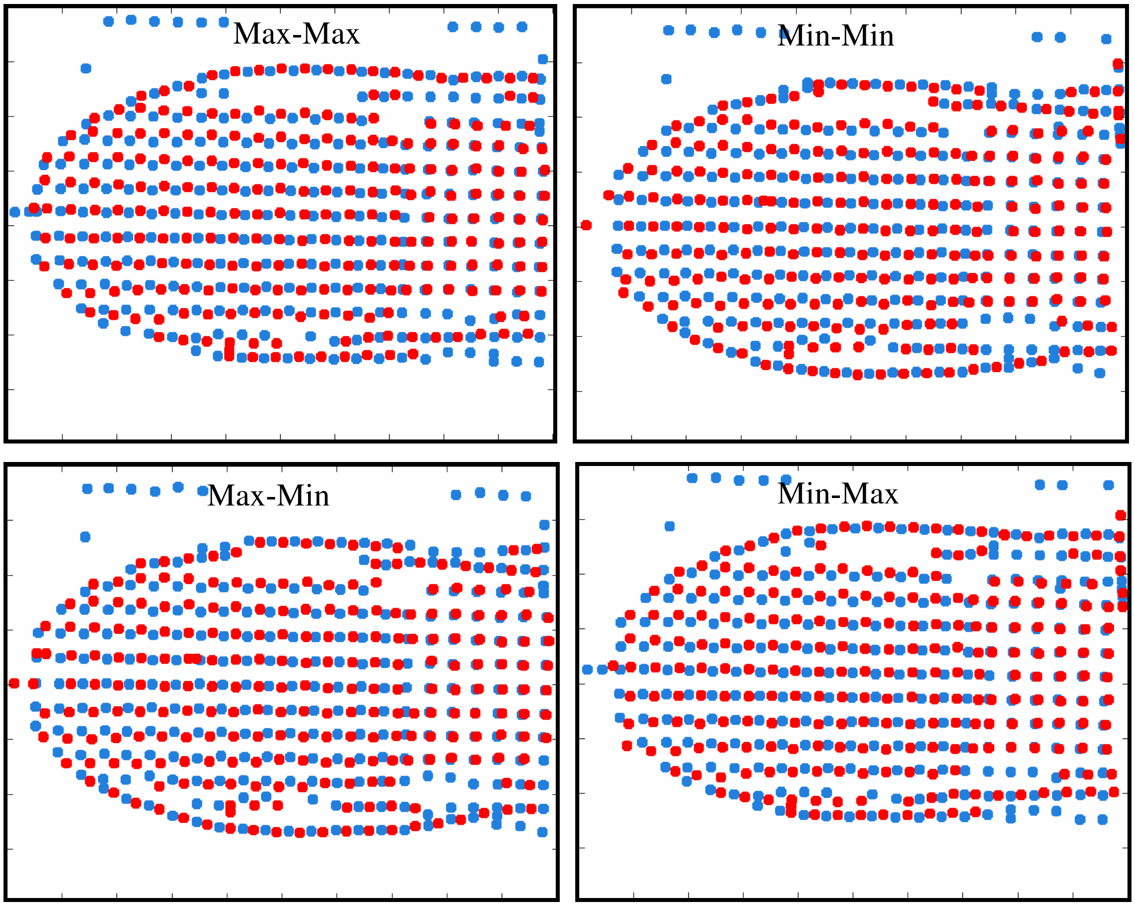


Fig. 6.17 Displacements of intersection points

Finally, the streamlines were drawn between the intersection points found above by a roaming procedure where the program first located points in the initial (0 ns) image, and then searched for the displaced point in the delayed (400 ns) image through a user-defined roaming window. Once a corresponding point was found, the displacement in pixels was calculated and converted into velocity (m/s) through the spatial conversion value. In this case, the value was 75.34 pixels/mm. The program also filtered for multiple velocity vectors which originated from the same initial point and for negative streamwise velocities. If no corresponding point was found in the roam window, no velocity vector was calculated. The results for only the maximum-maximum points are shown in Fig. 6.18, where the overall velocity magnitude and streamlines are overlaid in the image on top. Each velocity vector can be deconstructed into its streamwise and radial components, and these velocity maps are shown in the bottom left and right panels, respectively. The other three combinations (minimum-minimum, maximum-minimum, minimum-maximum) look similar. By adding together all four velocity maps, the full resolution velocity maps (streamlines and 2-component) were produced. These maps are shown in Fig. 6.19 and Fig. 6.20. In Fig. 6.19 and Fig. 6.20, the raw value of the radial velocities was retained, so that velocities in the "up" direction are denoted as negative, and velocities in the "down" direction are denoted positive. Finally, the maps were linearly interpolated in the horizontal and vertical directions. These results are presented in the next section. The method of quadrupling the resolution by edge-finding routines increased the resolution from 1 vector for every 50 pixels x 50 pixels to approximately 1 vector for every 20 pixels x 20 pixels (~270 um x ~270 um). Also, by calculating the vectors for each combination separately and then adding the final velocity maps instead of calculating the vectors for all combinations simultaneously, a larger roaming window and thus better velocity resolution was afforded.

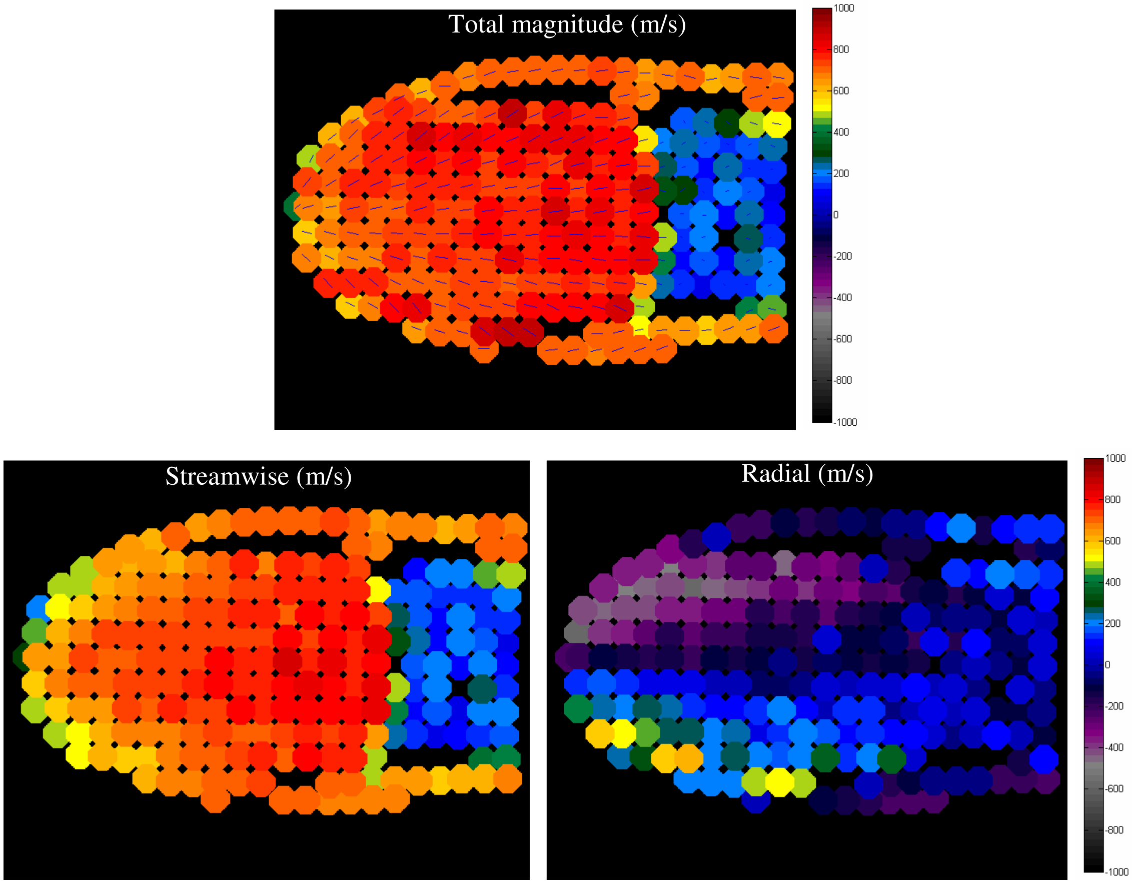


Fig. 6.18 Overall velocity, streamwise, and radial components (m/s) for max-max intersection points

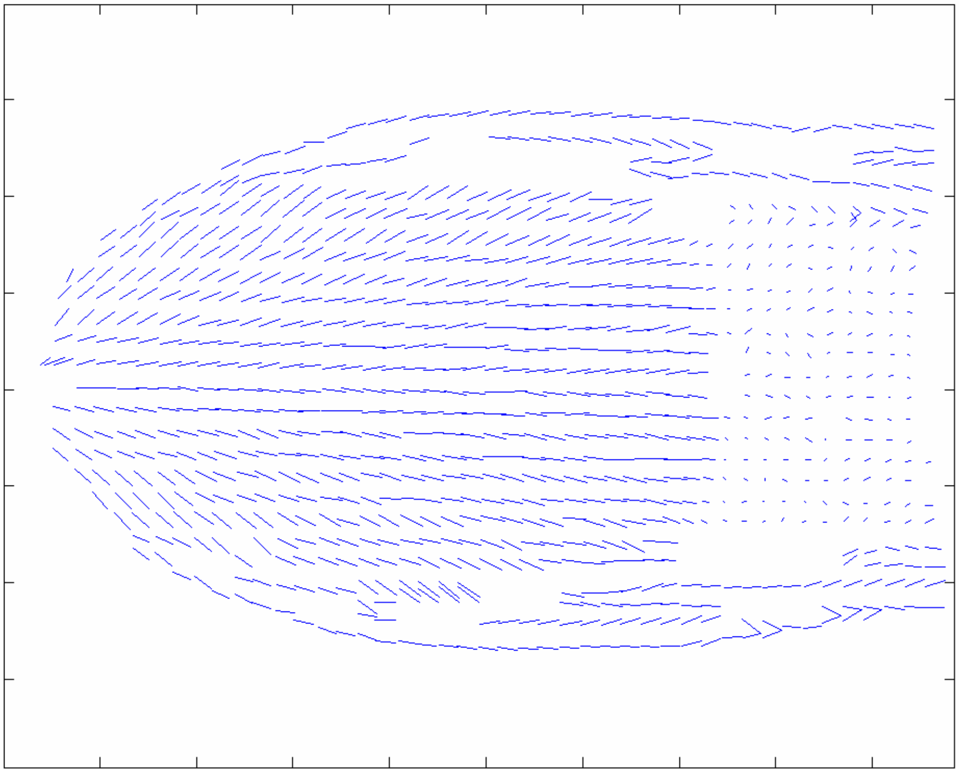


Fig. 6.19 Full resolution streamlines

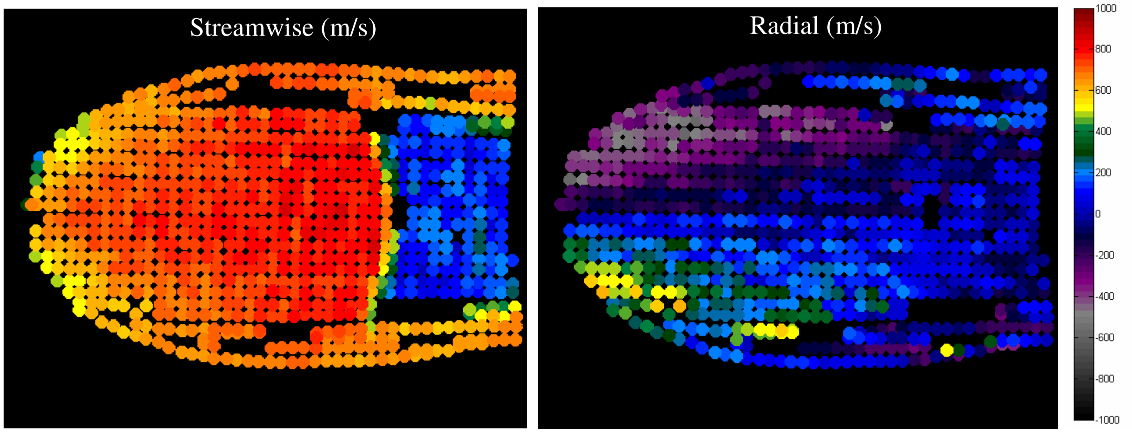


Fig. 6.20 Full resolution streamwise and radial velocity maps (m/s)

It should also be noted that in Runs 2, 3, and 4, an additional step was required to first shift the images obtained from both cameras to the same field of view. To accomplish this, a long-time (typically 1 or 2 minutes) integrated image was taken at identical time delays, and the edge-finding software was modified and used to locate the uprising edges on each image. The images were then manually aligned for slight differences in magnification, translation, and rotation of the images until the uprising edges matched. One example of this was taken from Run 4 and shown in Fig. 6.21, where the blue and white lines correspond to uprising edges in the two cameras.

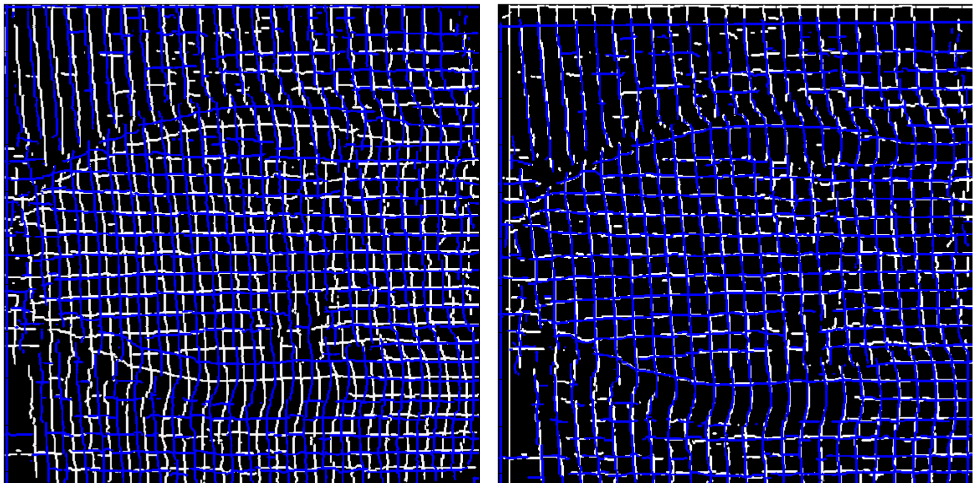


Fig. 6.21 Unwarped images (left) and warped images (right) from the two imaging cameras to equivalent fields of view

This step was very important because the calculated velocity maps were extremely sensitive to the location of both the initial and delayed points. For example, in the expansion region where the velocity was ~800 m/s, and using a typical spatial conversion of 74 pixels/mm, the flow moved 18 pixels in 300 ns. If the images were not carefully matched to the same field of view and there was a slight offset of 2 pixels, a velocity of 720 m/s would be calculated, resulting in a velocity error of 10%. In slower moving regions, the error would be proportionately larger. Therefore, warping the images to the same field of view was very important and extreme care was taken in ensuring the images were as closely overlaid as possible.

D. Results and Discussion: Integrated Images

The final data were the calculated vector streamlines, and raw and interpolated two-component velocity maps. These are presented in the following pages for each of the runs (integrated images only) in Fig. 6.22, Fig. 6.23, and Fig. 6.24. As a side note, the Run 2 interpolated velocity map images have been thresholded as compared to the raw velocity maps.

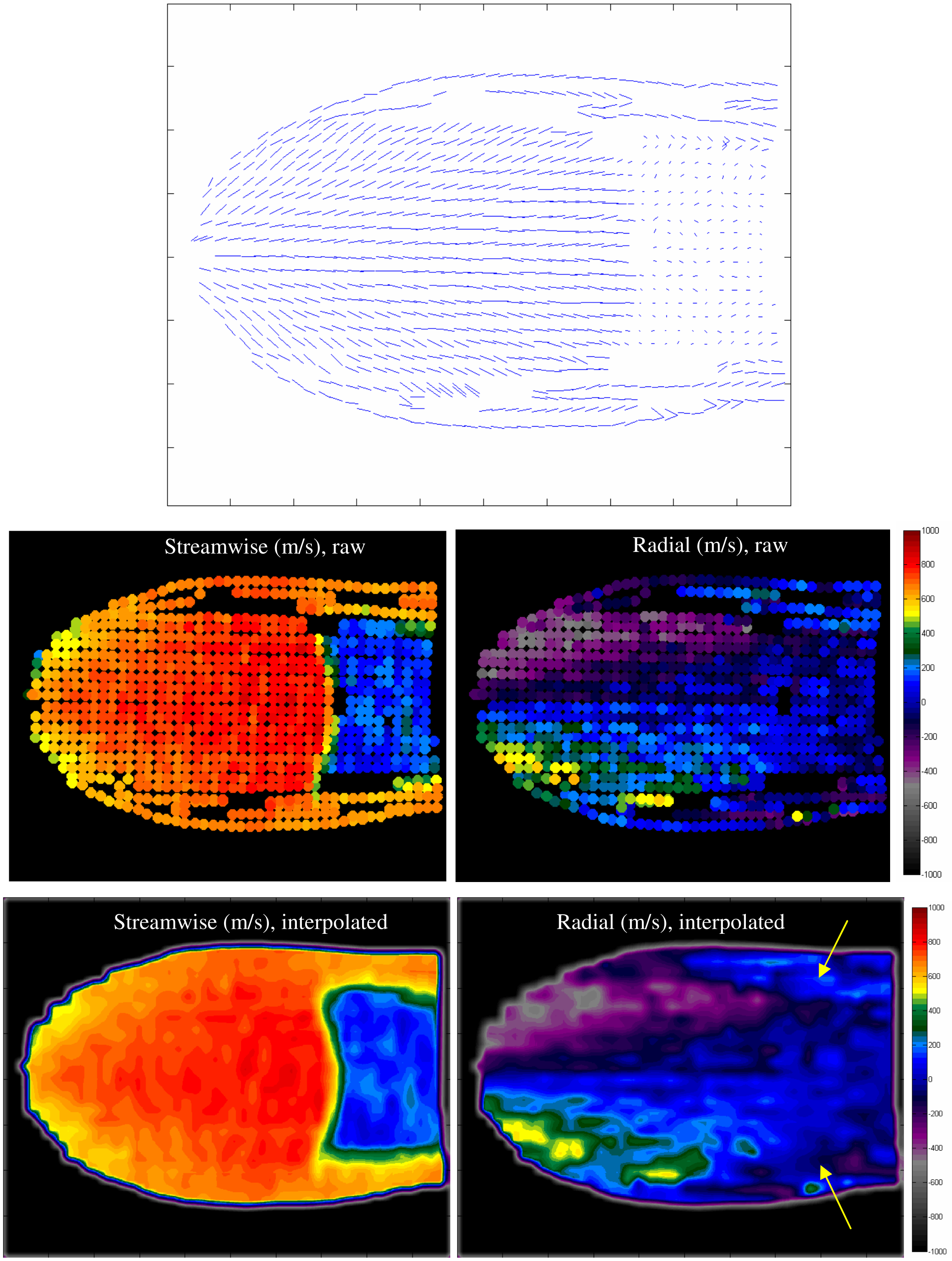


Fig. 6.22 Streamlines and raw and interpolated velocity maps for Run 1, integrated images (m/s) (low quenching, single fluorescence lifetime)

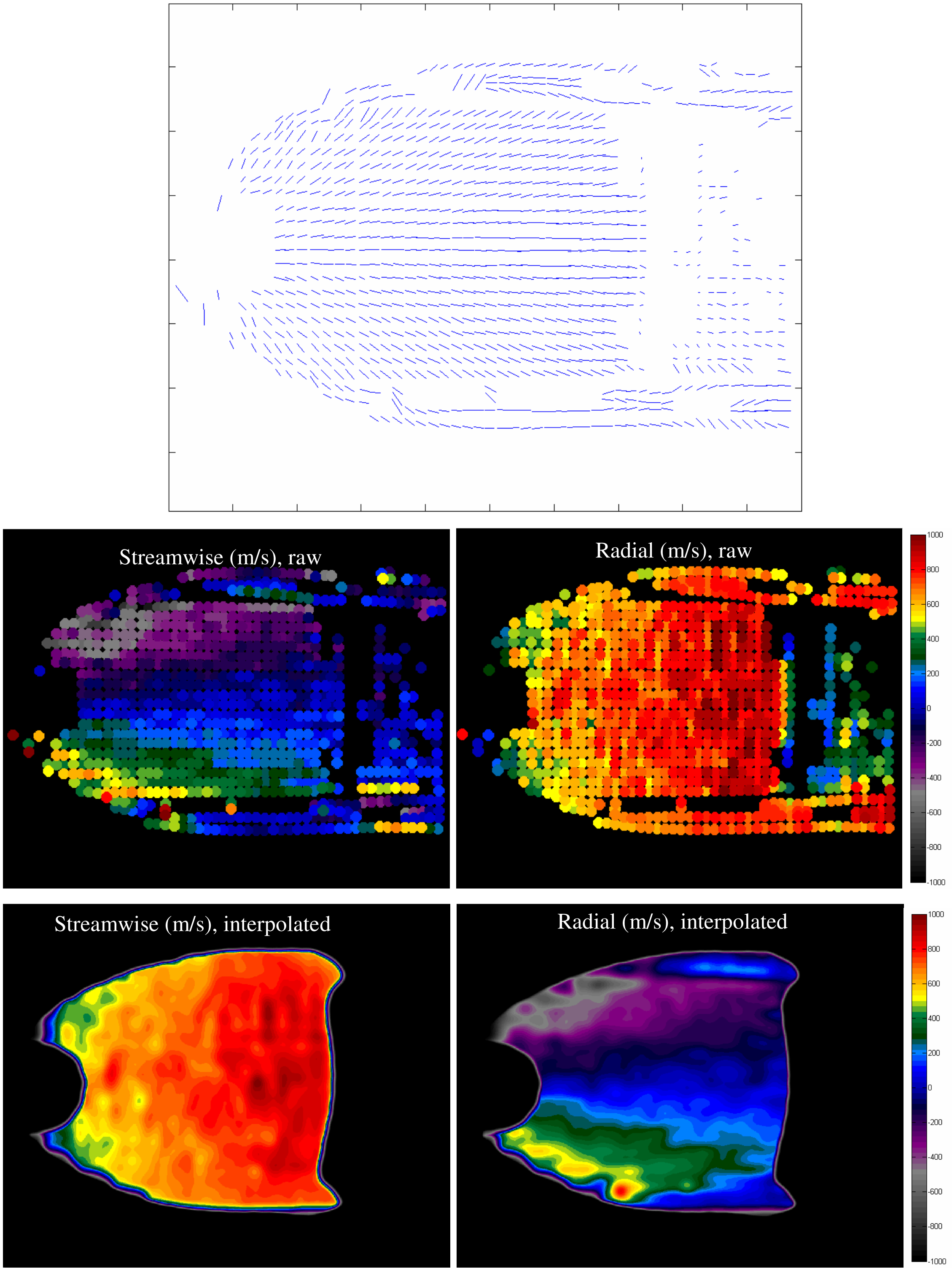


Fig. 6.23 Streamlines and raw and interpolated velocity maps for Run 2, integrated images (m/s) (high quenching, single fluorescence lifetime)

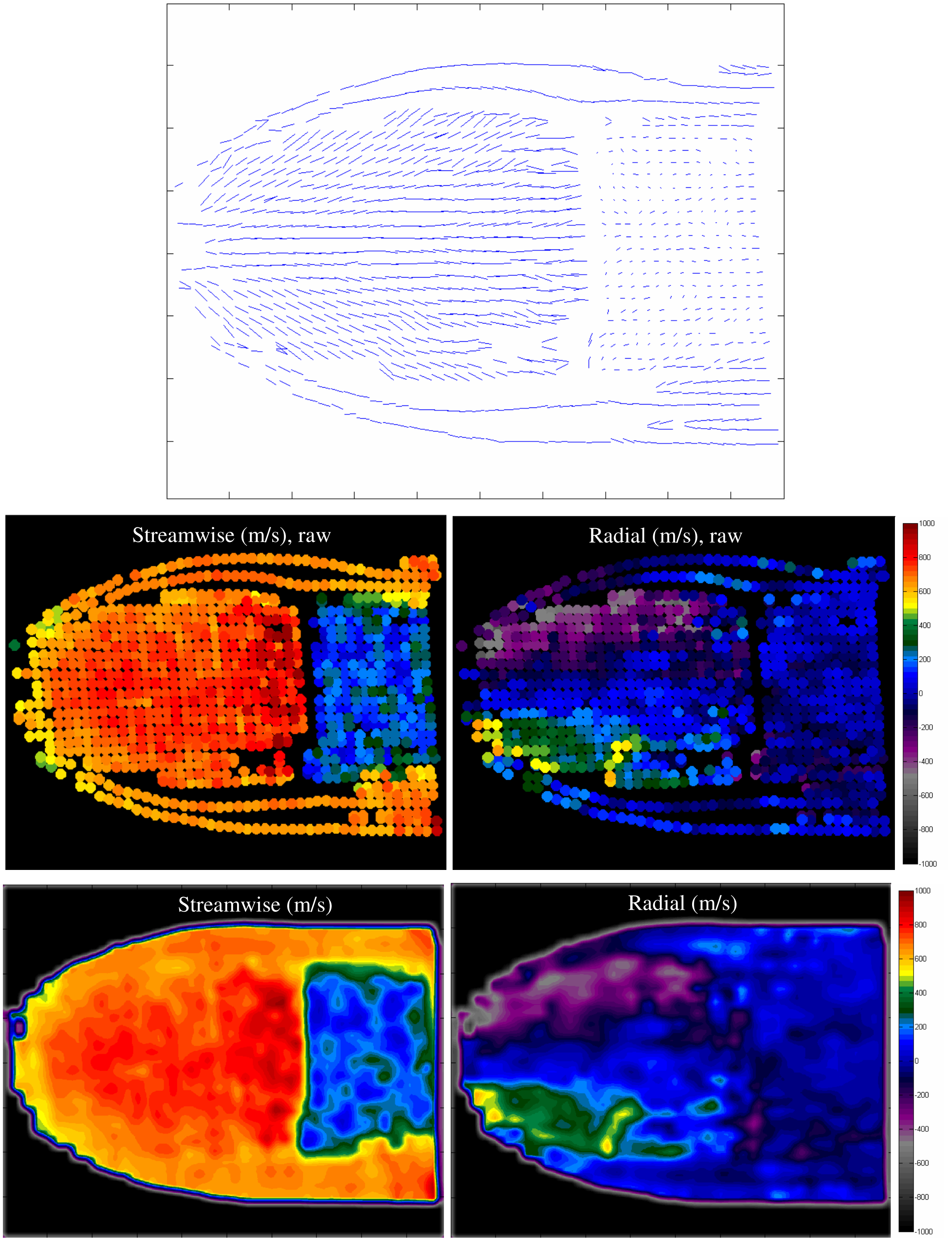


Fig. 6.24 Streamlines and raw and interpolated velocity maps for Run 3, integrated images (m/s) (high quenching, photodissociation)

Since all four of the runs were run at approximately equal JPR’s, the general structure of the velocity maps were similar. For the streamwise velocity map, the gas quickly expanded and accelerated. The largest velocity magnitude was directly before the Mach disk, and was about 850 m/s in all cases. A small region of mild velocity (about 500 m/s) was seen on either side of the centerline directly exiting the nozzle. This region corresponded to the largest magnitude of the radial velocity. Also, the streamwise velocity quickly dropped to subsonic velocities directly after the Mach disk as expected. The outer jet regions remained supersonic and surrounded the subsonic inner core. The majority of the overall velocity magnitude of the flowfield was contained in the streamwise component with a smaller radial component. The centerline of the radial velocity maps remained at 0 velocity, meaning that the gas in this region only experienced streamwise acceleration. On either side of the centerline, the gas expanded mildly (with the exception of the near-nozzle region). The Mach disk was not well delineated in the radial direction, as expected, since the shock acted as a normal shock in the streamwise direction. After the Mach disk, the flow began to transition from laminar to turbulent and lead to pockets of larger radial velocities and small variations in streamwise velocity. Another interesting aspect of the flow captured in the radial velocity maps was the presence of a slight turning in of the flow after the triple point causing a change in radial velocity from about 200 m/s to -200 m/s. An example of this region is marked in the results for integrated images for Run 1 (Fig. 6.22) using small arrows.

It was also useful to compare the results to CFD simulations to verify the overall structure of the flow and for overall comparison of the validity of the experimental velocity maps. The CFD simulations were run at the experimental JPR of Run 1. Comparisons between experimental results from Run 1 and the CFD results are presented in Fig. 6.25 and Fig. 6.26.

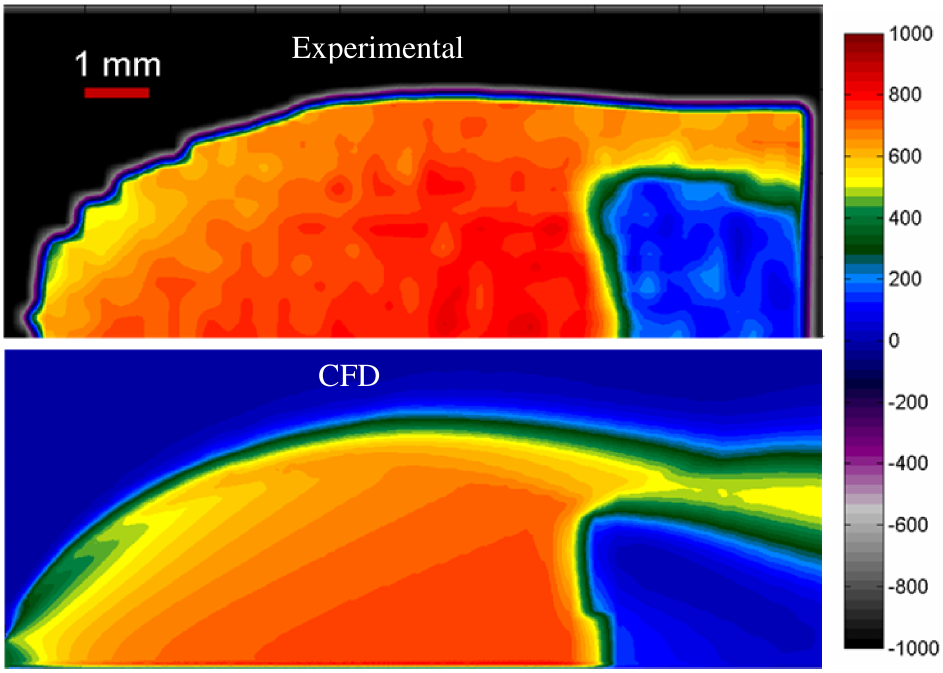


Fig. 6.25 Comparison of experimental streamwise velocity maps with CFD (m/s)

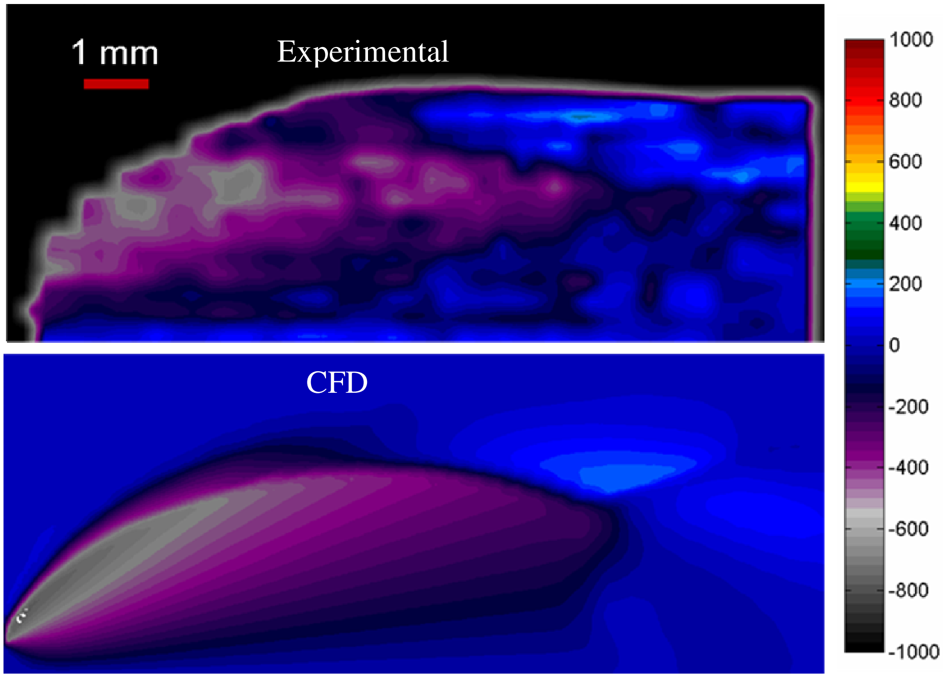


Fig. 6.26 Comparison of experimental radial velocity maps with CFD (m/s)

The comparison of the experimental results was in excellent agreement with CFD. Only qualitative comparisons were sought simply for sanity checking the experimental results. The overall structure was similar, and the magnitudes of the velocity in both the streamwise and radial directions matched. The CFD provided a "smoother" look than the experimental data, and this difference was due to the fact that the experimental velocity maps were interpolated from distinct velocity data points. Overall, the validity of the experimental results was confirmed. Because of this good agreement, the CFD was also useful for assessing the effects of fluorescence quenching. Quenching is a function of the local pressure and temperature, and since these are not available using the experimental data, the CFD provided these data. An image of the CFD results with several areas of interest pointed out using numbers from 1 to 8 are shown in Fig. 6.27.

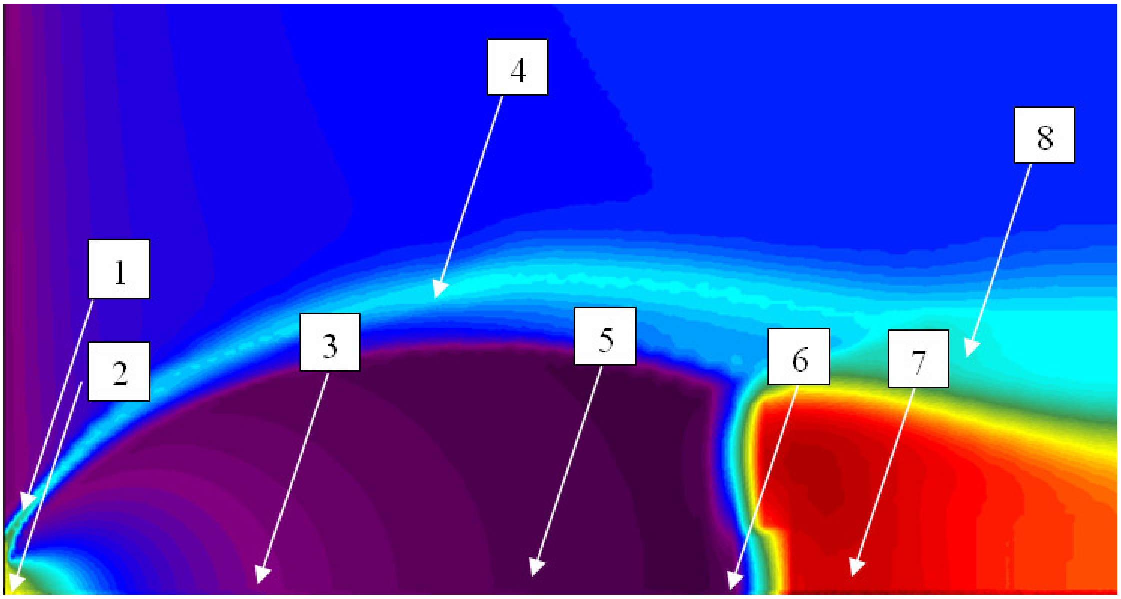


Fig. 6.27 Areas of interest for examination of quenching effects

The areas were pinpointed as regions of different temperature and pressures where quenching would have the largest effect. For example, near the nozzle, the area of highest radial expansion was chosen (Area 1). Also, the near nozzle area, which was the highest pressure area in the flowfield, was also chosen (Area 2). The expansion areas which corresponded to a rapid decrease in both pressure and temperature were chosen (Areas 3,5, and 6). Also, the barrel shock structure of medium pressure, Area 4, was examined. Finally, the post-shock regions in the subsonic inner jet and supersonic outer jet, Areas 7 and 8, respectively, were chosen. In order to look at the effect of quenching at each of these areas, Table 6.2 was compiled, using the model for fluorescence quenching discussed in Chapter III.

Table 6.2 Calculated fluorescence lifetime comparisons using CFD flowfield

	Flow conditions		Fluorescence lifetime (ns)	Fraction fluorescence signal remaining at		Fluorescence lifetime (ns)	Fraction fluorescence signal remaining at	
Area	P (Torr)	T (K)	Run 1	300 ns	400 ns	Run 2	300 ns	400 ns
1	2	190	218	0.2525	0.1596	171	0.1730	0.0964
2	200	225	119	0.0803	0.0346	84	0.0281	0.0085
3	0.5	50	218	0.2525	0.1596	173	0.1765	0.0990
4	1.5	135	218	0.2525	0.1596	169	0.1694	0.0937
5	0.1	30	221	0.2573	0.1636	202	0.2264	0.1380
6	0.03	15	221	0.2573	0.1636	209	0.2380	0.1475
7	3	315	219	0.2541	0.1609	175	0.1800	0.1017
8	5	150	211	0.2412	0.1502	123	0.0872	0.0386

The effects of quenching were highlighted at the following areas. In Area 2 (the high pressure region very near to the nozzle), the lifetime was the shortest in both the low (Run 1) and high (Run 2) quenching cases. Even in the low quenching case, the lifetime was only 119 ns, leaving only 3.5% of the original signal after 400 ns. In the high quenching case, the effect of quenching was exacerbated. After 300 ns, only 2.8% of the original signal remained, and after 400 ns, less than 1% remained, making tagging velocimetry very difficult at this time delay. Area 7 was another area which was affected heavily by quenching. The lifetimes for the low and high quenching cases after the Mach disk at a pressure of 3 Torr and temperature of 315K were 219 ns and 175 ns, respectively. This area was also very prone to loss of signal in the high-quenching case, leaving only 18% and 10.2% of the initial fluorescence signal at 300 ns and 400 ns delays, respectively. Although this was not much different from the low-quenching case, in which 25.4% and 16.1% of the initial fluorescence remained at 300 and 400 ns, respectively, the effect was exaggerated by fact that the initial fluorescence signal was inherently low (due to the low pressure). Finally, in Area 8, the second highest pressure area (after the area immediately outside the nozzle) the flow conditions were about 5 Torr and 150K. The lifetime of the low quenching case was 211 ns in this case, but only 123 ns in the high-quenching case, the second shortest lifetimes after the area immediately after the nozzle. Therefore, while the low-quenching case only suffered a decrease of signal of 75.9% and 85% at 300 and 400 ns, respectively, the high quenching case encountered losses of 92.8% and 96.4 % at the same time delays. These “problem” areas were demonstrated in Run 2, where quenching was high, and these regions experienced very fast signal decay due to a dramatically reduced fluorescence lifetimes, making the identification of any points in the time-delayed image virtually impossible, as seen experimentally in Fig. 6.23.

While the areas discussed above experienced large effects due to quenching, other areas were only mildly affected by quenching and maintained good signal even in high quenching conditions (Run 2). These were areas with low number densities where the collision frequency was reduced such as Areas 5 and 6. The pressure and temperature in these regions were very low at 30-100 mTorr and 15-30K, respectively. These conditions produced very long fluorescence lifetimes both in the low quenching and high quenching case, of greater than 200 ns. At these conditions, more than 20% of the initial signal remained at 300 ns, and about 15% remained at 400 ns in both cases. Then, even though the signal experienced a decrease in intensity of about an order of magnitude, the software was still able to locate the intersection points of the grid, and good velocity vectors were recovered in these regions.

Finally, the important conclusion was that the use of photodissociation (Run 3) yielded the velocimetry analysis immune to fluorescence quenching. In fact, Run 3 was run in high quenching conditions, with nearly double the O_2 content compared to Run 2, and yet maintained good velocity vectors in all of the "problem" areas of Run 2, namely, near the nozzle (Area 2), and after the Mach disk (Areas 7 and 8). This was due to the fact that, as mentioned before, imaging throughout the fluorescence lifetime was not used. Rather, a grid of NO was written into the flow, which was then probed at an initial and time-delayed image with two separate lasers, wherein the fluorescence was imaged immediately after each probe laser. In addition, as mentioned previously, probing the $NO_{v=1}$ state rather than $NO_{v=0}$ prevented any contaminant NO from appearing in the images and disrupting the velocimetry analysis. Also, the choice of rotational lines was important. In the expansion region of the flow, the low rotational temperatures meant that only low rotational states were well populated. Therefore, probing a low rotational line such as J"= 1.5, 2.5, or 3.5 helped maintain good signal intensity throughout the flowfield. The probe laser configuration (both probe lasers entered the chamber from above) reduced the effect of Doppler shifting in the horizontal laser sheet from the large streamwise velocity gradient encountered in Runs 1 and 2.

Another important advantage was that photodissociation of NO_2 produces NO which is a stable species and therefore, NO population depletion would be only governed by very slow three-body recombination reactions with O_2 (reforming NO_2), vibrational relaxation (if the user probes $NO_{v=1}$), and diffusion and/or fluid mixing out of the laser probe volume. Therefore, the photodissociation technique would be valuable in slow-moving flowfields where imaging though the fluorescence lifetime would yield inadequate spatial displacements for velocimetry analysis. If the user choses to probe $NO_{v=1}$, the vibrational relaxation of NO (and thus the depletion of $NO_{v=1}$) was mainly driven by vibrational quenching through photodissociated O atoms, as seen in the previous chapter. However, even with O atom quenching, $NO_{v=1}$ persisted for several tens of microseconds and thus may be probed with a very long delay time. For comparison, the fluorescence lifetime of NO was on the order of 10-200 ns depending on quenching conditions, much shorter than that of vibrational relaxation. If background $NO_{v=0}$ is not an issue, then the user may opt to probe $NO_{v=0}$ rather than $NO_{v=1}$. The population of $NO_{v=0}$ actually increases with time (due to vibrational relaxation from $NO_{v=1}$) nearly doubling within a few μs, and then very slowly decreases due to diffusion and fluid mixing. Then, the time delayed image is likely to have even more signal than the initial image, a clear advantage for the location of intersection points.

E. Results and Discussion: Single Shot Images

Single shot data would be very useful in short run time facilities; a prime example of such a facility is a shock tube, where runtimes of only a few milliseconds limit data collection to a single laser shot. Therefore, single shot data with good signal to noise are required. A statistical set of single shot data were taken at each of the quenching conditions (Runs 1, 2, and 4) for uncertainty analysis. The raw images are shown in Fig. 6.4, Fig. 6.6, and Fig. 6.10. The image processing procedure was the same as for the integrated images. The resulting interpolated velocity maps for the single shot data are shown in Fig. 6.28, Fig. 6.29, and Fig. 6.30): Simply by comparing the raw images, the signal to noise of the 60 s integrated images (Fig. 6.3, Fig. 6.5, and Fig. 6.9) were far superior overall compared to those of the single shot images. In these shot-noise limited images, the increase in integration times from single shot to 60 s at a laser repetition rate of 10 Hz represented a increase in signal to noise of the square root of 600, or about 25. The effect of signal to noise was most apparent in the streamline plots, where the 60 s integrated images produced very smooth, radially symmetric velocity vectors, while the single shot images produced more noisy, sporadic vectors due to the reduced ability of the software to locate the edges of the gridding lines, and thus the intersection points. Also noted was a tradeoff between signal to noise and velocity resolution. In the 60 s integrated images, where signal was plentiful, the 0 and 400 ns image pair produced velocity vectors with higher resolution than a 0 and 300 ns image pair (not shown) due to increased spatial displacement. The total number of vectors recovered was also greater than that of the 0 and 300 ns image pair. This was especially true in the subsonic (post- Mach disk) region of the flow where the spatial displacement between the two time delays was small and could be mistaken in the software as a zero or negative velocity and filtered out. It should be pointed out that the temporal separation between the initial and time delayed image of 300 ns and 400 ns were chosen to optimize spatial displacement in the hypersonic region (before the Mach disk) and not for the subsonic region after the Mach disk. Therefore, it was expected that the velocity resolution in these regions was lower than that of the hypersonic region.

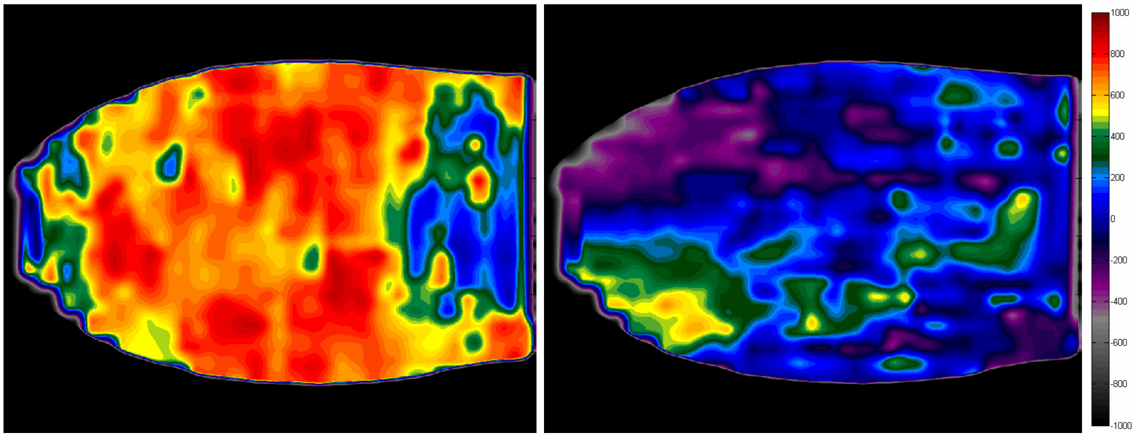


Fig. 6.28 Interpolated streamwise and radial velocity maps for Run 1, single shot images (m/s)

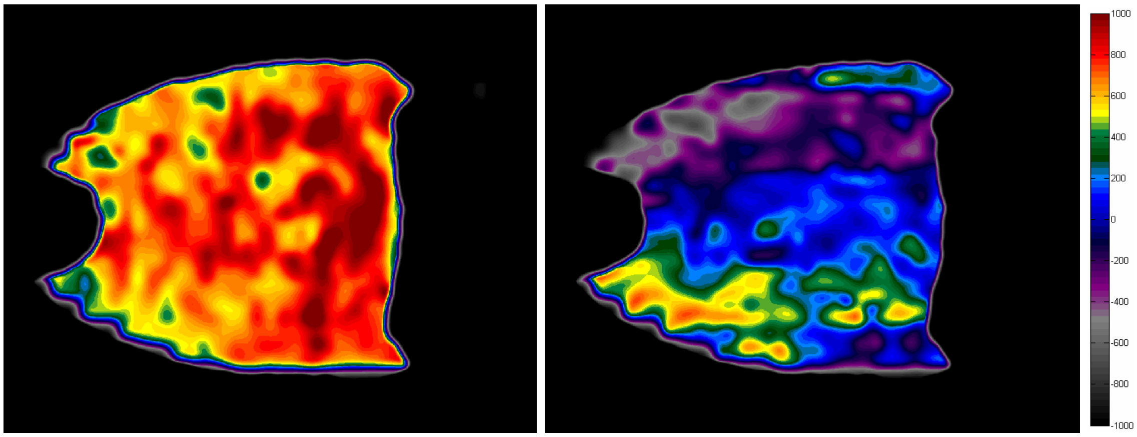


Fig. 6.29 Interpolated streamwise and radial velocity maps for Run 2, single shot images (m/s)

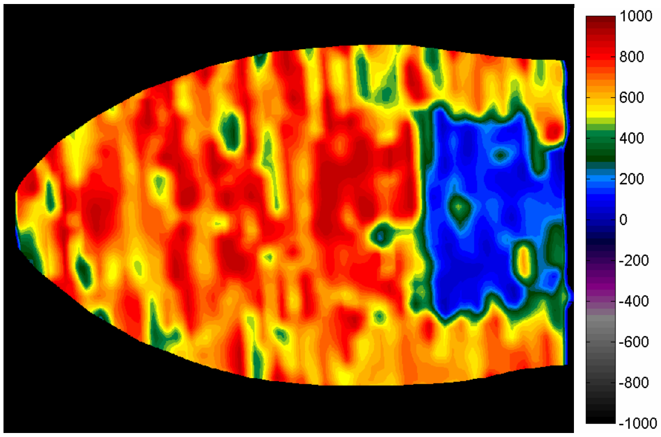


Fig. 6.30 Interpolated streamwise velocity map for Run 4, single shot images (m/s)

Statistical analyses were run on the 480 single shot images collected in Run 4. All of the images were run at the same field of view warping parameters as the integrated images. The average streamwise velocity, $\overline{U}$, and rms streamwise velocity, U_{rms} maps are shown in Fig. 6.31. It should be noted that the rms map included both uncertainties inherent to the flow (shot to shot fluctuations in flow structure) as well as uncertainties originating from the image processing due to lack of SNR.

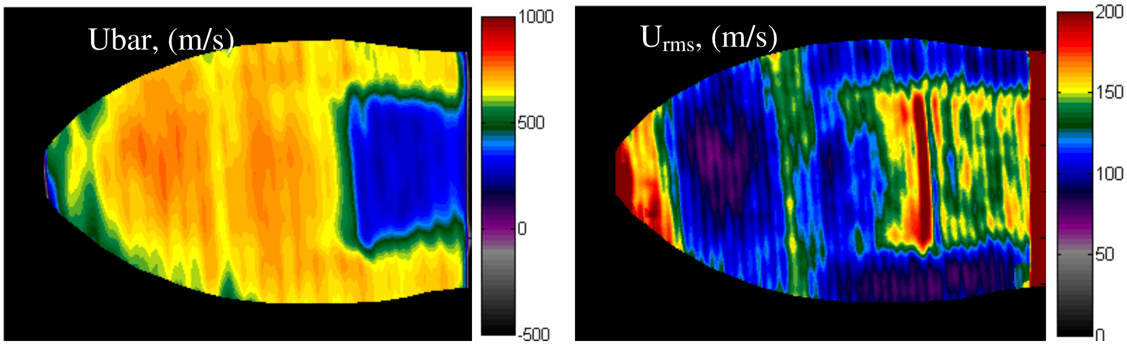


Fig. 6.31 $\overline{U}$ and U_{rms} maps for Run 4

The $\overline{U}$ map showed the expected trends corresponding to the flowfield characteristics already discussed. The U_{rms} map showed that the lowest uncertainty was obtained about at the middle of the expansion region, where the signal to noise was still good. At this point, the relative uncertainty was about 5%. After this point, the uncertainty increased from lack of signal. The Mach disk region experienced the largest uncertainty and is seen in the U_{rms} map by the dark red band in this region. However, this uncertainty was likely to be due to shot to shot fluctuations in the location of the Mach disk inherent to the flow rather than uncertainties originating from lack of SNR. After the Mach disk, the large uncertainties originated from lack of signal and from the choice of time delay (300 ns) as discussed in the previous paragraph. Since the flow only moved about 6 pixels between the acquisition of the initial and time delayed image, lower velocity resolution and higher relative uncertainties resulted. One way to increase velocity resolution and therefore decrease relative uncertainty which would have a large effect in the subsonic region would be the adaptation of the velocimetry code to give sub-pixel resolution.

E.1 Exploration of Doppler Shifted Fluorescence for Velocimetry

As a side note, a brief study into the utility of Doppler-shifted fluorescence was performed. In this study, only the radial velocity was considered since it was about 2-3 times smaller than the streamwise velocity and thus would be the limiting velocity measurement. Only the vertical laser grid was then used, and the laser wavelength was swept across the rotational line. The images clearly showed Doppler shifting in the radial direction, as seen in Fig. 6.32:

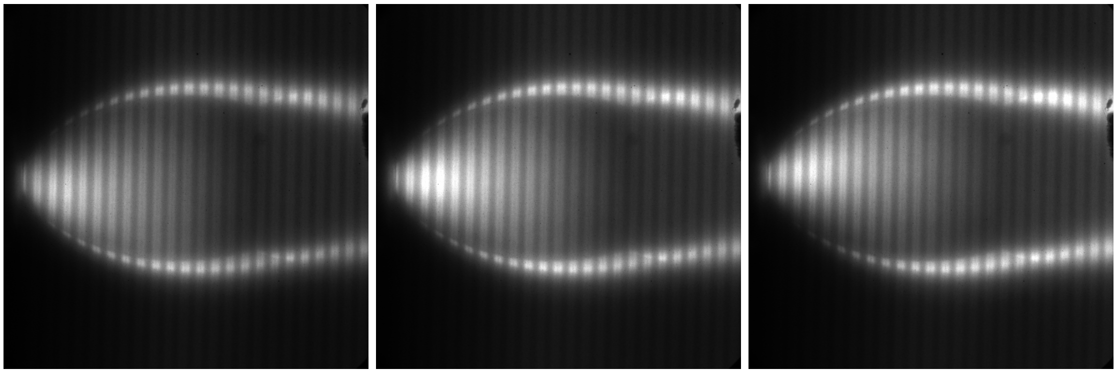


Fig. 6.32 Doppler shifted fluorescence using vertical laser grid

Brief calculations showed that at the smallest laser tuning step (0.00025 nm for the fundamental), and rough-fitting the images so that the velocity was determined at the laser wavelength where the signal was maximum, the highest velocity resolution possible was 165 m/s. In the fluorescence lifetime technique, the difference of a single pixel produces a resolution of 33 m/s at a time delay of 400 ns and 44 m/s at a time delay of 300 ns. Using the photodissociation technique, a far higher resolution can be accomplished since the time delay used is not limited by fluorescence lifetime, and increased spatial displacement can be afforded using longer time delays. It should be noted, however, that much more accurate Doppler shifted velocimetry could be obtained by fitting the spectra. However, this option was not pursued because data analysis software was not available. Furthermore, this technique has already been applied to the underexpanded jet flowfield through both averaged and instantaneous measurements in Paul (1989) and Palmer (1993).

F. SUMMARY AND RECOMMENDATIONS

We have demonstrated the use of two-component molecular tagging velocimetry towards an underexpanded jet flowfield. The study represented the first known application of molecular tagging velocimetry utilizing either the fluorescence lifetime or photodissociation technique towards two-component velocity mapping. Different techniques were presented which had advantages in different flow conditions. All techniques made use of taking a pair of images: an initial image and a time-delayed image. The intersection points were located for each image, and the velocity was calculated by tracking the spatial displacement of the points divided by the known temporal separation. The resolution was improved by splicing each intersection point of the grid into four points using edge-finding procedures, and rendered the technique's resolution competitive with PIV at 1 vector/(270 um x 270 um) in the field of view. For low quenching, high speed flows, the flow was seeded with NO, and the two images were taken during the NO fluorescence lifetime yielding excellent recovery of velocity vectors. In high quenching conditions, fluorescence lifetime was decreased and it became difficult to retain signal in the time delayed image, leading to loss of vectors in particular "problem" areas. In the underexpanded jet flowfield, these areas were identified as mainly those near the nozzle, and in the subsonic core following the Mach disk.

To circumvent the problem of quenching, a technique utilizing the photodissociation of seeded NO_2 into these high-quenching environments to form a grid of NO, and the subsequent tracking of the photodissociated NO using NO PLIF was demonstrated. The technique recovered vectors which were lost in the high quenching "problem" areas using the fluorescence lifetime technique and was virtually immune to quenching effects. The two-component technique produced velocity maps which matched CFD simulation results. Lastly, single shot analyses of the photodissociation technique yielded relative uncertainties of ~5% and were primarily limited by signal to noise. Large fluctuations in the vicinity of the Mach disk were due to shot-to-shot fluctuations in the location of the Mach disk and were inherent to the nature of the flow. Since the calculated NO_2 photodissociation fraction was greater than 99.9%, the uncertainties could be improved by either increasing NO_2 number density (followed by a subsequent increase in photodissociation laser intensity), or by increasing probe laser power. One way to increase velocity resolution (and also decrease uncertainties) which would have an especially large effect in the subsonic region would be the adaptation of the velocimetry code to give sub-pixel resolution.

CHAPTER VII

CONCLUSIONS AND RECOMMENDATIONS

In conclusion, a set of tools has been developed to study non-thermochemical equilibrium (NTE) flows. A number of fundamental studies have been completed which lay the groundwork for future diagnostics on subsonic and hypersonic NTE flowfields. In addition, new laser diagnostic techniques for studying these types of flowfields have been demonstrated. However, these techniques are not limited to NTE flowfields and could be applied towards a variety of flowfields. The specific conclusions and recommendations from each of the major experiments are briefly summarized here.

A radio-frequency (RF) discharge was generated within a subsonic flowfield and characterized using broadband and narrowband emission spectroscopy, and CARS spectroscopy. The emission spectra concluded that the plasma imparted only slight gas heating (35 +/- 3 K) with negligible molecular dissociation. N_2 CARS pointwise measurements at several streamwise locations verified that the plasma induced vibrational excitation, with a vibrational temperature of N_2 ~1600K measured about 1" downstream from the plasma,. The vibrational temperature gradually decayed to about 1300K at ten inches downstream from the first location. The experimentally observed decay was compared to vibrational kinetics modeling which included all relevant species in the flow with good agreement. The modeling suggested that the driving relaxation mechanisms for vibrational relaxation of N_2, O_2 were V-V and V-T energy exchanges with ambient H_2O with weak contributions from ambient CO_2. NO vibrational relaxation was also heavily dependent on V-V and V-T exchanges with O_2. Future work in this area could include further validation of the kinetics model by measuring the vibrational decay of O_2 and of NO using dual-pump CARS and vibrational NO PLIF, respectively. The model could be extended to include higher vibrational states and initially vibrationally excited H_2O and CO_2 for completeness. Also, further characterization of the RF plasma could include measurement of the electron density as a function of RF power, and the vibrational temperature decay rates of the different molecular species could be measured as the power was varied.

Under development of new diagnostic techniques, variations on molecular tagging velocimetry (MTV) were conducted. One of the main advantages of MTV techniques as opposed to particle-based techniques such as PIV lies in the ability of molecules to accurately track the flow, especially in the vicinity of strong shocks. First, single line MTV and vibrational temperature studies were conducted utilizing photodissociation of NO_2. This study was demonstrated in the well-studied underexpanded jet flowfield, but could be extended towards a variety of other flowfields. Photodissociation of NO_2 at 355 nm produced a non-thermal equilibrium distribution of $NO_{v=1}$: $NO_{v=0}$ which corresponded to a vibrational temperature of about 7000K. Velocimetry focused on tracking the movement of $NO_{v=1}$ and $NO_{v=0}$ with a pair of images: one initial, and one time-delayed by 400 ns. Streamwise velocity maps were calculated by noting the spatial displacement between the two images through edge-finding routines and recovered the expected velocity map for the underexpanded jet. In the low temperatures (320K and below) encountered in the underexpanded jet, the $NO_{v=1}$ images gave much better contrast than the $NO_{v=0}$ images because the ambient population of $NO_{v=1}$ was very low. At combustion temperatures of 1000 – 2000K, NO would be naturally formed, and the relative population of $NO_{v=1}$ to $NO_{v=0}$ occuring in the flow would be greater than at low temperatures. However, this ratio is still less than the non-thermal distribution of $NO_{v=1}$:$NO_{v=0}$ formed through photodissociation of NO_2, and thus probing $NO_{v=1}$ would continue to be advantageous in minimizing background $NO_{v=0}$ fluorescence. At even higher temperatures, the thermal decomposition of NO_2 must be considered as the amount of NO_2 that remains for photodissociation may be reduced. But, the signal enhancement for probing either vibrational state increased steadily with temperature as the Boltzmann distribution shifted towards higher vibrational states. Even in the presence of background fluorescence, the effects could be minimized through software subtraction of an integrated fluorescence image in the absence of photodissociation. Single shot 1σ streamwise velocity relative uncertainties for single-line photodissociation were about 5%. Diffusion served to decrease the magnitude of the gradients used for edge-finding but caused no systematic errors in the velocimetry analysis.

The vibrational temperature study explored the mechanism of $NO_{v=1}$ vibrational quenching by tracking $NO_{v=1}$ and $NO_{v=0}$ as they moved down the streamwise direction over a much longer time delay than the velocimetry study, several tens of microseconds. From the relative intensities of the images of $NO_{v=1}$ and $NO_{v=0}$, a NO vibrational temperature map was produced at each time delay and plotted as a function of time and space. The NO vibrational temperature decays were compared to vibrational kinetics modeling and the main driving mechanism was found to be vibrational quenching of $NO_{v=1}$ by O atoms formed through NO_2 photodissociation. Comparisons with CFD showed moderate agreement, although the locations of the oscillations following the Mach disk were slightly offset. However, the experimental results were self-consistent and were ultimately limited by loss of fluorescence signal (through fluid mixing and diffusion) and detector dynamic range. Error analysis using single-shot PLIF images produced single-shot vibrational temperature uncertainties of 5-6%. This technique was highly dependent on signal to noise, and more probe laser power would have been desired. However, since the ratio of the two images was used, careful verification that the fluorescence remained in the linear regime would be required. Better agreement between the vibrational modeling and experimental results could be obtained by extending the model to include higher vibrational levels.

The single component velocimetry technique was extended to yield two-components of velocity (streamwise and radial) in an underexpanded free jet through the use of a grid rather than a single line. The study compared low quenching conditions to high quenching conditions for velocimetry in NO seeded flows by forming a probe laser sheet into a two-dimensional, coplanar grid and acquiring two temporally separated images during the NO fluorescence lifetime (300 ns or 400 ns apart). The low quenching conditions gave excellent signal to noise throughout the flowfield, leading to excellent recovery of velocity vectors. The high quenching environment shortened the fluorescence lifetime and led to loss of signal in the time delayed image in some areas, namely those near the nozzle and after the Mach disk. Velocity vectors could not be calculated in these areas. Using the photodissociation technique, NO_2 was seeded into a high quenching environment and photodissociated into a grid of $NO_{v=1}$ and $NO_{v=0}$. The $NO_{v=1}$ was probed (following the recommendation from the single line study) using two temporally (400 ns) separated probe lasers. The fluorescence from each of these lasers was collected. Since two separate probe lasers are used, there was no concern of fluorescence lifetime and thus this technique was virtually immune to quenching effects. Using this technique, signal was recovered in areas which were lost in the high quenching flow. The streamwise and radial velocity maps produced had a maximum spatial resolution of 1 velocity vector per 270 um x 270 um in the field of view. Typical resolutions were approximately 1 velocity vector per 400 um x 400 um in the field of view. Resolution was limited by the amount of smoothing applied to the raw images, which was directly limited by signal to noise. The two-component technique was shown to produce velocity maps which agreed with CFD simulation results. Single shot analyses of the photodissociation technique yielded relative velocity uncertainties of ~5% and were limited by signal to noise. Also, since the photodissociation technique was not affected by quenching, a much longer time delay could be used, since NO is a stable molecule and the signal would essentially only be limited by fluid mixing and diffusion out of the probe laser sheet.

Both the single and two-component velocimetry methods gave low single-shot relative uncertainties, rendering them useful for short runtime facilities such as blow-down facilities and shock tubes, where runtimes may range anywhere from a few milliseconds (i.e. single laser shot allotted) to a few minutes, and/or for statistical measurements in dynamic flowfields. Both were dependent on good signal to noise, and resolution and uncertainties could be improved with increases in signal. This could be accomplished by either increasing NO_2 number density (and then subsequently increasing photodissociation laser intensity), or by increasing probe laser intensity. A longer time delay, resulting in larger spatial displacements, could also be used in both cases at the cost of spatial resolution. One recommendation which would increase velocity resolution and decrease relative uncertainties would be the adaptation of the velocimetry code to give sub-pixel resolution.

The compilation of all of the above topics represents a collection of diagnostic tools that may be used to characterize and understand not only NTE systems, but can be extended for use in a variety of flowfields.

REFERENCES

Adamovich, I., *Personal Communication*, 2005.

Ahn, T., Adamovich, I., Lempert, W., “Determination of Nitrogen V-V Transfer Rates by Stimulated Raman Pumping,” *Chemical Physics*, Vol. 298, 2004, pp. 233-240.

Atkinson, R., Baulch, D., Cox, R., Crowley, J., Hampson, R., Hynes, R., Jenkin, M., Rossi, M., Troe, J., “Evaluated Kinetic and Photochemical Data for Atmospheric Chemistry: Volume I - Gas Phase Reactions of Ox, HOx, NOx and SOx Species,” *Atmospheric Chemistry and Physics*, Vol. 4, 2004, pp. 1461-1738.

Bass, H., “Absorption of Sound by Air: High Temperature Predictions*,” Journal of Acoustic Society of America*, Vol. 69, No. 1, 1981, pp. 124-138.

Bauer, S., Caballero, J., Curtis, R., Wiesenfeld, J., “Vibrational Relaxation Rates of CO_2(001) with Various Collision Partners for T<300K,” *Journal of Physical Chemistry*, Vol. 91, No. 7, 1987, pp. 1778-1785.

Bohl, D. G., Koochesfahani, M. M., “Molecular Tagging Velocimetry Measurements of Axial Flow in a Concentrated Vortex Core,” *Physics of Fluids*, Vol. 16, No. 11, 2004, pp. 4185-4191.

Bohm, B., Doughty, A., Hancock, G., Moore, E., Morrell, C., “Vibrational Relaxation of NO(v=1-3) and NO_2(0,0,1) with Atmospheric Gases,” *Physical Chemistry Chemical Physics*, No. 1, 1999, pp. 1833-1842.

Bowersox, R., North, S., Srinivasan, R., “High-Speed Free Shear Layers with Molecular Non-Equilibrium: An Example of the Fluids Information Triad (Invited),” *46th AIAA Aerospace Sciences Meeting and Exhibit*, AIAA-2008-0685, Reno, Nevada, 2008.

Breen, J., Quy, R., Glass, G., “Vibrational Relaxation of O_2 in the Presence of Atomic Oxygen,” *Journal of Chemical Physics*, Vol. 59, No. 1, 1973, pp. 556-557.

Brennan, M., Alle, D., Euripides, P., Buckman, S., Brunger, M., “Elastic Electron Scattering and Rovibrational Excitation of N_2 at Low Incident Energies*,” J. Phys. B: At. Mol. Opt. Phys*. Vol. 25, 1992, pp. 2669-2682.

Brooks, C., Hancock, G., Saunders, M., “Dependence of the Nascent Vibrational Distribution of NO(v) on the Photolysis Wavelength of NO_2 in the Range λ=266-327 nm Measured by Time-resolved Fourier Transform Infrared Emission,” *Physical Chemistry Chemical Physics*, Vol. 9, 2007, pp. 5232-5240.

Byrne, S., Danehy, P., Houwing, A., "Investigation of Hypersonic Nozzle Flow Uniformity Using NO Fluorescence," *Shock Waves*, Vol. 15, No. 2, 2006, pp. 81-87.

Castle, K., Hwang, E., Dodd, J., "Laboratory Studies of CO_2(ν2)-O Vibrational Energy Transfer," *Proceedings of SPIE*, Vol. 5235, Bellingham, WA, 2004

Candler, G., "Translation-Vibration-Dissociation Coupling in Nonequilibrium Hypersonic Flows," *AIAA 24th Thermophysics Conference*, AIAA-89-1739, Buffalo, NY, 1989.

Candler, G., Kelley, J., Macheret, S., Shneider, M., Adamovich, R., "Vibrational Excitation, Thermal Nonuniformities, and Unsteady Effects on Supersonic Blunt Bodies," *AIAA Journal*, Vol 40, No. 9, 2002, pp. 1803-1810.

Cannemeyer, F., Vries, A., "Vibrational Relaxation of CO_2 in CO_2-N_2 and CO_2-O_2 Mixtures," *Physica*, Vol. 74, 1974, pp. 196-204.

Cattolica,R., "OH Rotational Temperature from Two-Line Laser-Excited Fluorescence," *Applied Optics,* Vol. 20, No. 7, 1981, pp. 1156-1166.

Cheng, H., Emanuel, G., "Perspective on Hypersonic Nonequilibrium Flow," *AIAA Journal*, Vol. 33, No. 3, 1995, pp. 385-400.

Chintala, N., Meyer, R., Hicks, A., Bystricky, B., Rich, J., Lempert, W., Adamovich, I., "Non-Thermal Ignition of Premixed Hydrocarbon-Air and CO-Air Flows by Nonequilibrium RF Plasma," *42nd AIAA Aerospace Sciences Meeting and Exhibit*, AIAA-2004-0835, Reno, Nevada, 2004.

Cramp, J., Lambert, J., "Vibrational Relaxation of $CO_2(\nu_3)$ by O Atoms," *Chemical Physics Letters*, Vol. 22, No. 1, 1973, pp. 146-149.

Danehy, P., Bathel, B., Inman, J., Alderfer, D., Jones, S., "Stereoscopic Imaging in Hypersonic Boundary Layers Using Planar Laser-Induced Fluorescence," *26th AIAA Aerodynamic Measurements Technology and Ground Testing Conference*, AIAA-2008-4267, Seattle, Washington, 2008.

Danehy, P., O'Byrne, S., Houwing, F., Fox, J., Smith, D., "Flow-Tagging Velocimetry for Hypersonic Flows Using Fluorescence of Nitric Oxide," *AIAA Journal*, Vol. 41, No. 2, 2003, pp. 263-271.

Danehy, P., Inman, J., Alderfer, D., Buck, G., Bathel, B., "Visualization of Flowfield Modification by RCS Jets on a Capsule Entry Vehicle," *46th AIAA Aerospace Sciences Meeting and Exhibit*, AIAA-2008-1231, Reno, Nevada, 2008.

Danehy, P., Wilkes, J., Alderfer, C., Jones, S., Robbins, A., Patry, D., Schwartz, R., “Planar Laser-Induced Fluorescence (PLIF) Investigation of Hypersonic Flowfields in a Mach 10 Wind Tunnel (Invited),” *25th AIAA Aerodynamics Measurement Technology and Ground Testing Conference*, AIAA-2006-3442, San Francisco, California, 2006.

DeJoseph, C. N_2(C-B) Emission Code, Air Force Research Laboratory, Wright-Patterson, A.F.B, OH. (May 2005)

Demore, W., Sander, S., Golden, D., Hampson, R., Kurylo, J., Howard, C., Ravishankara, A., Kolb, C., Molina, J., “Chemical Kinetics and Photochemical Data for Use in Stratospheric Modeling*,” JPL Publication 97-4*, 1997, pp. 1-266.

Dodd, J., Lockwood, R., Hwang, E., Miller, S., Lipson, S., “Vibrational Relaxation of NO(v=1) by Oxygen Atoms,” *Journal of Chemical Physics*, Vol. 11, No. 8, 1999, pp. 3496-3507.

Donaldson, C. D., Snedeker R. S., “A Study of Free Jet Impingement. Part 1. Mean Properties of Free and Impinging Jets,” *Journal of Fluid Mechanics*, Vol. 45, Part 2, 1971, pp. 281-319.

Eckbreth, A., *Laser Diagnostics for Combustion Temperature and Species*, Abacus Press, Kentucky and Massachusetts, 1988.

Finzi, J., Hovis, F., Panfilov, V., Hess, P., Moore, C., “Vibrational Relaxation of Water Vapor,” *Journal of Chemical Physics*, Vol. 67, No. 9, 1977, pp. 4053-4961.

Frank, J., Kalt, P., Bilger, R., “Measurements of Conditional Velocities in Turbulent Premixed Flames by Simultaneous OH PLIF and PIV,” *Combustion and Flame*, Vol. 116, 1999, pp. 220-232.

Fuller, T.J., "The Effect of Vibrational Non-Equilibrium on the Decay of Grid Generated Turbulence,” (PhD dissertation, Texas A&M University, 2009).

Green, B., Caledonia, G., Murphy, R., Robert, F., “The Vibrational Relaxation of NO(v=1-7) by O_2,” *Journal of Chemical Physics*, Vol. 76, No. 5, 1982, pp 2441-2448.

Hanson, R.K., “Planar Laser-Induced Fluorescence Imaging,” *Journal of Quantitative Spectroscopy and Radiative Transfer*, Vol. 40, No. 3, 1988, pp. 343- 362.

Harrison, J., Yang, X., Rosslein, M., Felder, P., Huber, J., “Photodissociation of NO_2 at 355 and 351 nm Investigated by Photofragment Translational Spectroscopy,” *Journal of Physical Chemistry*, Vol. 98, 1994, pp. 12260-12269.

Heimerl, J., Coffee, T., “The Unimolecular Ozone Decomposition Reaction,” *Combustion and Flame*, Vol. 35, 1979.

Herzberg, G., *Molecular Spectra and Molecular Structure I. Spectra of Diatomic Molecules*, Prentice-Hall, Inc, Canada, 1939.

Higgins, K., Gillan, C., Burke, P., Noble, C., “Low-Energy Electron Scattering by Oxygen Molecules: II. Vibrational Excitation,” *J. Phys. B: At. Mol. Opt. Phys.*, Vol. 28, 1995, pp. 3391-3402.

Hishida, K., Sakakibara, J., “Combined Planar Laser-Induced Fluorescence-Particle Image Velocimetry Technique for Velocity and Temperature Fields,” *Experiments in Fluids*, Vol. 29, 2000, pp. 129-140.

Hu, H., Koochesfahani, M. M., “Molecular Tagging Velocimetry and Thermometry and Its Application to the Wake of a Heated Circular Cylinder,” *Measurement Science and Technology*, Vol. 17, 2006, pp. 1269-1281.

Hu, H., Koochesfahani, M. M., “Molecular Tagging Velocimetry for the Simultaneous Measurements of Flow Velocity and Temperature Fields,” *44th AIAA Aerospace Sciences Meeting and Exhibit*, AIAA 2006-41, Reno, NV, 2006.

Huffman, R., Elliott, G., “An Experimental Investigation of Accurate Particle Tracking in Supersonic, Rarefied Axisymmetric Jets,” *47th AIAA Aerospace Sciences Meeting and Exhibit*, AIAA 2009-1265, Orlando, FL, 2009.

Hunter, M., Reid, S. A., Robie, D. C., Reisler, H., “The Monoenergetic Unimolecular Reaction of Expansion-Cooled NO_2: NO Product State Distributions at Excess Energies 0-3000cm^{-1},” *Journal of Chemical Physics*, Vol. 99, No. 2, 1993, pp. 1093-1108.

Huetz-Aubert, M., Tripodi, R., “Rate Equations for the Vibrational Relaxation of CO_2-N_2 or CO_2-Noble Gas Mixtures- Application to Comparison of Spectrophone Data with Results from Other Experimental Techniques,” *Journal of Chemical Physics*, Vol. 55, No. 12, 1971, pp. 5724-5734.

Ianni, J., KINTECUS, Windows XP Version, www. Kintecus.com, 2008.

Ismailov, M. M., Schock H. J., Fedewa, A. M., “Gaseous Flow Measurements in an Internal Combustion Engine Assembly Using Molecular Tagging Velocimetry,” *Experiments in Fluids*, Vol. 41, 2006, pp. 57-66.

Itikawa, Y., Ichimura, A., Onda, K., Sakimoto, K., Takayanagi, K., Hatano, Y., Hayashi, M., Nishimura, H., Tsurubuchi, S., "Cross Sections for Collisions of Electrons and Photons with Oxygen Molecules," *Journal of Physical Chemistry Reference Data,* Vol. 18, No. 1, 1989, pp. 23-42.

Itikawa, Y., Hayashi, M., Ichimura, A., Onda, K., Sakimoto, K., Takayanagi, K., Nakamura, M., Nishimura, H., Takayanagi, T., "Cross Sections for Collisions of Electrons and Photons with Nitrogen Molecules," *Journal of Physical Chemistry Reference Data,* Vol. 15, No. 3, 1986, pp. 985-1010.

Jacobsen, L., Carter, C., Jackson, T., Williams, S., Barnett, J., Tam, C., Baurle, R., Bivolaru, C., Kuo, S., "Plasma-Assisted Ignition in Scramjets," *Journal of Propulsion and Power*, Vol. 24, No. 4, 2008, pp. 641-654.

Jiang, N., Lempert, W., "Ultra-High Frame Rate Nitric Oxide Planar Laser Induced Fluorescence Imaging," *Optics Letters*, Vol. 33, No. 19, 2008, pp. 2236-2238.

Johnbaik, R., "Theoretical Investigation of the Vibrational Relaxation of Nitric Oxide (n=1-7) in Collisions with O_2 and N_2," *Bulletin of the Korean Chemical Society*, Vol. 14, No. 1, 1993, pp. 47-52.

Joly, V., Marmignon, C., "Effect of H_2O Vapor on the Vibrational Relaxation in Hypersonic Wind Tunnels," *Journal of Thermophysics and Heat Transfer*, Vol. 11, No. 2, 1977, pp. 266-272.

Kimmel, R., *Personal Communication*, 2005.

Kosanetzky, J., List, U., Urban, W., Vormann, H., "Vibrational Relaxation of NO($X_2\Pi$,v=1) Studied by an IR-UV-Double-Resonance Technique", *Chemical Physics*, Vol. 50, 1980, pp 361-371.

Kronemayer, H., Omerbegovic, K., Schulz, C., "Quantification of the Evaporative Cooling in an Ethanol Spray Created by a Gasoline Direct-Injection System Measured by Multiline NO-LIF Gas-Temperature Imaging," *Applied Optics*, Vol. 46, No. 34, 2007, pp 8322-8327.

Kruger, S., Grunefeld, G., "Stereoscopic Flow-Tagging Velocimetry," *Applied Physics B*, Vol. 69, 1999, pp. 509-512.

Kulatilaka, W. D., Naik, S. V., Lucht, R. P., "Development of High-Spectral-Resolution Planar Laser-Induced Fluorescence Imaging Diagnostics for High-Speed Gas Flows," *AIAA Journal*, Vol. 46, No. 1, 2008, pp. 17-20.

Kychakoff, G., Howe, R., Hanson, R., “Quantitative Flow Visualization Technique for Measurements in Combustion Gases,” *Applied Optics*, Vol. 23, 1984, pp. 704-712.

Kychakoff, G., Howe, R., Hanson, R., McDaniel, J., “Quantitative Visualization of Combustion Species in a Plane,” *Applied Optics*, Vol. 21, No. 18, 1982, pp. 3225-3227.

Kychakoff, G., Knapp, K., Howe, R., Hanson, R., “Flow Visualization in Combustion Gases Using Nitric Oxide Fluorescence,” *AIAA Journal*, Vol. 22, No. 1, 1984, pp. 153-154.

Lachney, E., Clemens, N., “PLIF Imaging of Mean Temperature and Pressure in a Supersonic Bluff Wake,” *Experiments in Fluids*, Vol. 24, 1998, pp 354-363.

Lee, T., Bessler, W., Kronemayer, H., Schulz, C., Jeffries, J., “Quantitative Temperature Measurements in High-Pressure Flames with Multiline NO-LIF Thermometry,” *Applied Optics*, Vol. 44, No. 31, 2005, pp 6718-6728.

Lee, M., McMillin, B., Hanson, R., “Temperature Measurements in Gases by Use of Planar Laser-Induced Fluorescence Imaging of NO,” *Applied Optics*, Vol. 32, No. 27, 1993, pp 5379-5396.

Lee, T., “Strategies for Nitric Oxide Laser-Induced Fluorescence in High-Pressure Combustion Systems,” (PhD Dissertation, Stanford University, 2006).

Lemaire, A., Meyer, T., Zahringer, K., Gord, J., Rolon, J., “PIV/PLIF Investigation of Two-Phase Vortex-Flame Interactions: Effects of Vortex Size and Strength,” *Experiments in Fluids*, Vol. 36, 2004, p. 36-42.

Lempert, W. R., Jiang, N., Sethuram, S., Samimy, M., “Molecular Tagging Velocimetry Measurements in Supersonic Microjets,” *AIAA Journal*, Vol. 40, No. 6, 2002, pp. 1065-1070.

Lempert, W. R., Boehm, M., Jiang, N., Gimelshein, S., Levin, D., “Comparison of Molecular Tagging Velocimetry Data and Direct Simulation Monte Carlo Simulations in Supersonic Micro Jet Flows,” *Experiments in Fluids*, Vol. 34, 2003, pp. 403-411

Lucht, R., Velur-Natarajan, V., Carter, C., Grinstead, K., Gord, J., Danehy, P., Fiechtner, G., Farrow, R., “Dual-Pump Coherent Anti-Stokes Raman Scattering Temperature and CO_2 Concentration Measurements,” *AIAA Journal*, Vol. 41, No. 4, 2003, pp. 679-686.

Lucht, R., “Three-Laser Coherent Anti-Stokes Raman Scattering Measurements of Two Species,” *Optics Letters*, Vol. 12, 1987, pp. 78-80.

Luque, J., Crosley, D. R., “LIFBASE: Database and Spectral Simulation Program (Version 1.5),” SRI International Report MP 99-009, 1999.

http://www.matweb.com/search/datasheet.aspx?matguid=1b8c06d0ca7c456694c7777d9e10be5b&ckck=1 [fracture toughness of Aluminum 6061 Alloy]. Accessed April 2009.

McMillin, B., "Instantaneous Two-Line PLIF Temperature Imaging of Nitric Oxide in Supersonic Mixing and Combustion Flowfields,” (PhD dissertation, Stanford University, 1993).

McMillin, B., Lee, M., Hanson, R., “Planar Laser-Induced Fluorescence Imaging of Nitric Oxide in Shock Tube Flows with Vibrational Nonequilibrium,” *21st Fluids Dynamics, Plasma Dynamics and Laser Conference*, AIAA-90-1519, Seattle, Washington, 1990.

McMillin, B., Palmer, J., Hanson, R., “Two-Dimensional Temperature Measurements of Shock Tube Flows Using Planar Laser-Induced Fluorescence Imaging of Nitric Oxide,” *22nd AIAA Fluid Dynamics, Plasma Dynamics, and Lasers Conference*, AIAA-91-1670, Honolulu, Hawaii, 1991

McNeal, R., Whitson, M., Cook, G., “Quenching of Vibrationally Excited N2 by Atomic Oxygen,” *Chemical Physics Letters*, Vol. 16, No. 13, 1972, pp. 507-510.

Menon, A., Lee, S., Linevsky, M., Litzinger, T., Santoro, R., “Addition of NO_2 to a Laminar Premixed Ethylene-Air Flame: Effect on Soot Formation,” *Proceedings of the Combustion Institute*, Vol. 31, 2007, pp. 593-601.

Menter, F. R. “Zonal Two Equation Turbulence Model Predictions,” AIAA Paper No. 93-2906

Meyer, T., Feichtner, G., Gogineni, S., Rolon, J., Carter, C., Gord, J., “Simultaneous PLIF/PIV Investigation of Vortex-Induced Annular Extinction in H_2-Air Counterflow Diffusion Flames,” *Experiments in Fluids*, Vol. 36, 2004, pp. 259-267.

Miles, R., Zhou, D., Zhang, B., Lempert, W., “Fundamental Turbulence Measurements by Relief Flow Tagging,” *AIAA Journal*, Vol. 31, No. 3, 1993, pp. 447-452.

Mojarrabi, B., Gulley, R., Middleton, A., Cartwright, D., Teubner, P., Buckman, S., Brunger, M., “Electron Collisions with NO: Elastic Scattering and Rovibrational (0 -> 1,2,3,4) Excitation Cross Sections,” *J. Phys. B: At. Mol. Phys.*, Vol. 28, 1995, pp. 487-504.

Naik, S., Kulatilaka, W., Lucht, R., “Development of High-Spectral-Resolution Planar Laser-Induced Fluorescence Imaging Diagnostics for High-Speed Gas Flow,” *46th AIAA Aerospace Sciences Meeting and Exhibit*, AIAA-2008-246, Reno, Nevada, 2008.

Nakaya, S., Kasahara, M., Tsue, M., Kono, M., “Velocity Measurements of Reactive and Non-reactive Flows by NO-LIF Method Using NO_2 Photodissociation,” *Heat Transfer-Asian Research*, Vol. 34, No. 1, 2005, pp. 40-52.

Nandula, S. P., Pitz, R. W., Bominaar, J., Schoemaecker, C., Dam, N. J., ter Meulen, J. J., “Kinetics of NO Tag Formation in Air for Unseeded Molecular Tagging Velocimetry,” *42nd AIAA Aerospace Sciences Meeting and Exhibit*, AIAA 2004-390, Reno, NV, 2004.

Nee, J., Juan, C., Hsu, J., Yang, J., Chen, W., “The Electronic Quenching Rates of NO ($A^2\Sigma^+$,v’=0-2),” *Chemical Physics*, Vol. 300, 2004, pp. 85-92.

Orlemann, C., Schulz, C., Wolfrum, J., “NO-Flow Tagging by Photodissociation of NO_2. A New Approach for Measuring Small-Scale Flow Structures,” *Chemical Physics Letters*, Vol. 307, 1999, pp 15-20.

Osipov, A., Uvarov, A. and Vinnichenko, N., “Influence of the Initial Vibrationally Non-equilibrium State of a Medium on the Structure of Von Karman Vortex Sheet,” *Physics of Fluids,* Vol. 18, 2006, pp. 105106-1-7.

Palm, P., Meyer, R., Plonjes, E., Rich, J., Adamovich, I., “Nonequilibrium Radio Frequency Discharge Plasma Effect on Conical Shock Wave: M=2.5 Flow,” *AIAA Journal*, Vol. 41, No. 3, 2003, pp. 465-469.

Palmer, J., Hanson, R., “Planar Laser-Induced Fluorescence Imaging in Free jet Flows with Vibrational Nonequilibrium,” *31st AIAA Aerospace Sciences Meeting and Exhibit*, AIAA-93-0046, Reno, Nevada, 1993.

Palmer, J., Hanson, R., “Single-Shot Velocity Using Planar Laser-Induced Fluorescence Imaging of Nitric Oxide,” *29th Joint Propulsion Conference and Exhibit*, AIAA-93-2020, Monterey, California, 1993.

Palmer, J. L., McMillin, B. K., Hanson, R. K., “Planar Laser-Induced Fluorescence Imaging of Underexpanded Free Jet Flow in a Shock Tunnel Facility,” *22nd AIAA Fluid Dynamics, Plasma Dynamics and Lasers Conference, AIAA-91-1687*, Honolulu, Hawaii, 1991.

Palmer, J., McMillin, B., Hanson, R., “Planar Laser-Induced Fluorescence Imaging of Velocity and Temperature in Shock Tunnel Free Jet Flow,” *30th Aerospace Sciences Meeting and Exhibit*, AIAA-92-1762, Reno, Nevada, 1992.

Palmer, R., “The CARSFIT Computer Code for Calculating Coherent Anti-Stokes Raman Spectra: User and Programmer Information,” Sandia Report SAND89–8206: Sandia National Laboratories, Livermore, CA., 1989.

Paul, P., Gray, J., Durant, J., Thoman, J., “Collisional Electronic Quenching Rates for NO for $A^2\Sigma^+$(v’=0),” *Chemical Physics Letters*, Vol. 259, 1996, pp. 508-514.

Paul, P. H., Lee, M. P., Hanson, R. K., “Molecular Velocity Imaging of Supersonic Flows Using Pulsed Planar Laser-Induced Fluorescence of NO,” *Optics Letters*, Vol. 13, No. 9, 1989, pp. 417-419.

Pitz, R. W., Brown, T. M., Nandula, S. P., Skaggs, P. A., DeBarber, P. A., Brown, M. S., Segall, J., “Unseeded Velocity Measurement by Ozone Tagging Velocimetry,” *Optics Letters*, Vol. 21, No. 10, 1996, pp. 755-757.

Pitz, R. W., Lahr, M. D., Douglas, Z. W., Wehrmeyer, J. A., Hu, S., Carter, C. D., Hsu, K. Y., Lum, C., Koochesfahani, M. M., “Hydroxyl Tagging Velocimetry in a Mach 2 Flow with a Wall Cavity,” *43rd AIAA Aerospace Sciences Meeting and Exhibit, AIAA-2005-36*, Reno, NV, 2005.

Raizer, Y., *Radio-Frequency Capacitive Discharges*, CRC Press, Florida, 1955.

Ree, J., “Theoretical Investigation of the Vibrational Relaxation of Nitric Oxide (n=1-7) in Collisions with Oxygen and Nitrogen,” *Bulletin of the Korean Chemical Society,* Vol. 14, No. 1, 1993, pp. 47-52.

Rehm, J., Clemens, N., “A PIV/PLIF Investigation of Turbulent Planar Non-Premixed Flames,” *35th AIAA Aerospace Sciences Meeting and Exhibit*, AIAA-1997-250, Reno, Nevada, 1997.

Ribarov, L. A., Wehrmeyer, J. A., Hu, S., Pitz, R.W., “Multiline Hydroxyl Tagging Velocimetry Measurements in Reacting and Nonreacting Experimental Flows,” *Experiments in Fluids*, Vol. 37, 2004, pp. 65-74.

Roy, C., Blottner, F., “Review and Assessment of Turbulence Models for Hypersonic Flows: 2D/Axisymmetric Cases,” *44th AIAA Aerospace Sciences Meeting and Exhibit*, AIAA-2006-713, Reno, NV, 2006.

Ryan, M., Clemens, N., Varghese, P., “Spectroscopic Measurements During an Electrothermal Plasma-JA2 Solid Propellant Interaction,” *44th AIAA Aerospace Sciences Meeting and Exhibit*, AIAA-2006-1464, Reno, Nevada, 2006.

Scalabrin, L., Boyd, I., “Development of an Unstructured Navier-Stokes Solver for Hypersonic Nonequilibrium Aerothermodynamics,” *38th AIAA Thermophysics Conference*, AIAA 2005-5203, Toronto, Canada, 2005.

Semper, M., "Turbulence Measurements In A High-Speed Free Shear Layer With Thermal Nonequilibrium & Hypersonic Facility Design & Construction,” (PhD dissertation, Texas A&M University, 2009).

Simpson, D., Gait, P., Simmie, J., “The Vibrational Deactivation of the Bending Mode of CO_2 by O_2 and N_2,” *Chemical Physics Letters*, Vol. 47, No. 1, 1977, pp. 133-136.

Smith, Steven. *The Scientist and Engineer’s Guide to Digital Signal Processing.* California: California Technical Publishers, 1997.

Stephenson, J., “Vibrational Relaxation of NO($X_2\Pi$,v=1) in the Temperature Range 100-300K,” *Journal of Chemical Physics*, Vol. 60, No. 11, 1974, pp. 4289-4294.

Stier, B., Koochesfahani, M., “Molecular Tagging Velocimetry (MTV) Measurements in Gas Phase Flows,” *Experiments in Fluids*, Vol. 26, 1999, pp. 297-304.

Strang, W., Tomaro, R., Grismer, M., “The Defining Methods of Cobalt 60: A Parallel, Implicit, Unstructured Euler/Navier-Stokes Flow Solver,” *37th Aerospace Sciences Meeting and Exhibit*, Reno, NV, AIAA Paper 1999-786, 1999

Sutton, J., Driscoll, J., “A Method to Simultaneously Image Two-Dimensional Mixture Fraction, Scalar Dissipation Rate, Temperature and Fuel Consumption Rate Fields in a Turbulent Non-Premixed Jet Flame,” *Experiments in Fluids*, Vol. 41, 2006, pp 608-627.

Tichenor, N., "Spatial Spectra In A High-Speed Free Shear Layer With Thermal Nonequilibrium & Hypersonic Facility Design & Construction,” (PhD dissertation, Texas A&M University, 2009).

Trevisan, C., Houfek, K., Zhang, Z., Orel, A., McCurdy, C., Rescigno, T., “Nonlocal Model of Dissociative Electron Attachment and Vibrational Excitation of NO,” *Physical Review A.*, Vol. 71, 2005.

Tsang, W., Hampson, R., “Chemical Kinetic Data Base for Combustion Chemistry. Part I. Methane and Related Compounds,” *Journal of Physics Chemistry Reference Data*, Vol. 15, No.3, 1986, pp. 1087-1222.

Tsang, W., Herron, J., “Chemical Kinetic Data Base for Propellant Combustion. I. Reactions Involving NO, NO_2, HNO, HNO_2, HCN and N_2O,” *Journal of Physical Chemistry Reference Data*, Vol. 21, 1991, pp 609-663.

Tsurikov, M., Rehm, J., Clemens, N., “High-Resolution PIV/PLIF Measurements of a Gas-Phase Turbulent Jet,” *37th AIAA Aerospace Sciences Meeting and Exhibit*, AIAA-1999-0930, Reno, Nevada, 1999.

van der Laan, W. P. N., Tolboom, R. A. L., ter Meulen, J. J., “Molecular Tagging Velocimetry in the Wake of an Object in Supersonic Flow,” *Experiments in Fluids*, Vol. 34, 2003, pp. 531-533.

Kruger, S., Grunefeld, G., “Stereoscopic Flow-Tagging Velocimetry,” *Applied Physics B*, Vol. 69, 1999, pp. 509-512.

Wehrmeyer, J. A., Ribarov, L. A., Oguss, D. A., Pitz, R. W., “Flame Flow Tagging Velocimetry with 193-nm H_2O Photodissociation,” *Applied Optics*, Vol. 38, No. 33, 1999, pp. 6912-6917.

Wilkes, J. A., Danehy, P. M., Nowak, R. J., “Fluorescence Imaging Study of Transition in Underexpanded Free Jets,” NASA TR-20050232745, 2005.

Wilkes, J. A., Glass, C. E., Danehy, P. M., Nowak, R. J., “Fluorescence Imaging of Underexpanded Jets and Comparison with CFD,” *44th AIAA Aerospace Sciences Meeting and Exhibit*, AIAA-2006-910, Reno, NV, 2006

Woodmansee, M. A., Iyer, V., Dutton, J. C., Lucht, R. P., “Nonintrusive Pressure and Temperature Measurements in an Underexpanded Sonic Jet Flowfield,” *AIAA Journal*, Vol. 42, No. 6, 2004, pp. 1170-1180.

Zuckerwar, A., Miller, K., “Vibrational-Vibrational Coupling in Air at Low Humidities,” *Journal of Acoustic Society of America*, Vol. 84, No. 3, 1988, pp. 970-977.

APPENDIX A

NOMENCLATURE AND SYMBOLS

c = speed of light (3 x 10^8 m/s)

k = Boltzmann constant (1.38 x 10^{-23} J/K)

N_1^0 = initial population of the ground state (molecules/cm^3)

N_1 = population of the ground state (molecules/cm^3)

N_2 = population of the excited state (molecules/cm^3)

B_{12} = stimulated absorption Einstein coefficient for transition from state 1 to state 2 $\left(\frac{m^3}{J*s^2}\right)$

B_{21} = stimulated emission Einstein coefficient for transition from state 2 to state 1 $\left(\frac{m^3}{J*s^2}\right)$

R_{21} = rotational energy transfer (RET) from state 2 to 1 (s^{-1})

I_{sat} = saturation intensity $\left(\frac{J}{m^2}\right)$

I_L = laser intensity $\left(\frac{J}{m^2}\right)$

A_{21} = spontaneous emission Einstein coefficient (s^{-1})

Q_{21} = quenching coefficient (s^{-1})

$\frac{\Omega}{4\pi}$ = collection solid angle

η = fluorescence collection efficiency term

$h\nu_{emitted}$ = emitted photon energy (J/photon)

S_f = fluorescence signal intensity (J/s)

V = measurement volume (m^3)

$g(v_{abs}, \Delta v_{abs}, v_{las}, \Delta v_{las})$	=	overlap integral
f_B	=	Boltzmann population fraction
C_{12}	=	experimentally determined calibration constant
ΔE_{21vib}	=	energy difference between vibrational states (J)
ΔE_{21rot}	=	energy difference between rotational states (J)
T	=	translational temperature (K)
T_{vib}	=	vibrational temperature (K)
T_{rot}	=	rotational temperature (K)
T'	=	temperature fluctuation (K)
$\overline{T}$	=	mean temperature (K)
T_{rms}	=	root mean squared temperature (K)
D_e	=	nozzle diameter
D_m	=	Mach disk diameter
X_m	=	distance from nozzle to Mach disk
w	=	primary wavelength
P_e	=	exit pressure
P_a	=	ambient pressure
P_o	=	stagnation pressure
T_o	=	stagnation temperature
Re	=	Reynold’s number (m^{-1})
$\overline{U}$	=	average streamwise velocity (m/s)
U_{rms}	=	root mean squared streamwise velocity (m/s)
$\overline{V}$	=	average radial velocity (m/s)
V_{rms}	=	root mean squared radial velocity (m/s)

$J_i"$ or J	=	rotational quantum state
$v^{"}$ or v	=	vibrational quantum state
$D_{Diffusion}$	=	diffusion coefficient m^2/s
$t_{randomwalk}$	=	random walk time (s)
z_{rms}	=	root mean squared distance traveled (m or pixels)
$< v >$	=	molecular velocity (m/s)
λ	=	mean free path (m)
μ	=	reduced mass (kg/molecule)
Z_1	=	collision frequency (collisions/s)
d	=	molecular diameter (m)
n_1	=	number density (molecules/m^3)
u_{avg}	=	spatially averaged velocity (m/s)
u_{instant}	=	instantaneous velocity (m/s)
Δx	=	spatial displacement (pixels or m)
Δt	=	temporal separation (s)
w_{pixel}	=	pixel width (m)
$f_{MTFsystem}$	=	MTF of imaging system (lp/mm)
f_{lens}	=	MTF of lens (lp/mm)
$f_{\mathrm{intensifer}}$	=	MTF of intensifier (lp/mm)
f_{fiber}	=	MTF of fiber optic coupling (lp/mm)
f_{array}	=	MTF of CCD array (lp/mm)
Q	=	quantum efficiency of the CCD
F	=	signal photon flux (photons/pixel/s)

B = background photon flux (photons/pixel/s)

D = dark current noise (e-/pixel/s)

R = read out noise (e-/pixel/s)

APPENDIX B

LIST AND DEFINITION OF ACRONYMS APPEARING IN THE TEXT

CARS = “Coherent Anti-Stokes Raman Spectroscopy”

CCD = “Charge-Coupled Device”

CCRF = “Capacitively-Coupled Radio-Frequency”

CFD = “Computational Fluid Dynamics”

EFL = “Effective Focal Length”

FFT = “Fast Fourier Transform”

FP = “Forward Power”

FWHM = “Full Width Half Max”

GUI = “Graphical User Interface”

ICCD = “Intensified Charge-Coupled Device”

JPR = “Jet Pressure Ratio”

LDV = “Laser Doppler Velocimetry”

LSF = “Line Spread Function”

LIF = “Laser-Induced Fluorescence”

MATLAB = “Awesome”

MCP = “Microchannel Plate”

MTF = “Modulation Transfer Function”

MTV = “Molecular Tagging Velocimetry”

MURI = “Multidisciplinary University Research Initiative”

NSC = “Normalized Sensitivity Coefficient”

NTE = “Non-Thermochemical Equilibrium”

PIV = “Particle Image Velocimetry”

PLIF = “Planar Laser-Induced Fluorescence”

RET = “Rotational Energy Transfer”

RF = “Radio-Frequency

SCCM = “Standard Cubic Centimeters per Minute”

SCRAMJET = “Supersonic Combustion Ramjet”

SNR = “Signal to Noise Ratio”

V-T = “Vibrational-Translational”

V-V = “Vibrational-Vibrational”

APPENDIX C

MTV TWO-COMPONENT VELOCIMETRY CODE

```
close();
clear();
picbegin=1;
picend=630;

for piccount=picbegin:1:picend
piccount
%settings---------------------------------------
firstfile=strcat('D');
secondfile=strcat('BC_');
writefile1='C:\Documents and Settings\Andrea\Desktop\12032008\PROCESS\';
getfile1='C:\Documents and Settings\Andrea\Desktop\12032008\';
plotgraphs=1;
savefiles=1; %save figures
initialtime=0;
delaytime=300;
spatialres=75.34; %used for velocity calculation (pixel/mm)
filteryvel=0;
gaussian1_1=20;
gaussian2_1=10;
gaussiankernel_1=10;
gaussianfilterinitial1 = fspecial('gaussian', ...
[gaussian1_1, gaussian2_1],gaussiankernel_1); %smooth initial images

gaussian1_12=10;
gaussian2_12=20;
gaussiankernel_12=10;
gaussianfilterinitial2 = fspecial('gaussian', ...
[gaussian1_12, gaussian2_12],gaussiankernel_12); %smooth initial images

%ROI settings
rowbegin=120;
rowend=860;
colbegin=1;
colend=1024;

%shear settings
shearanglehorizontal=0.03;
shearanglevertical=0.000;
```

```
%peakdet settings
%smoothing parameters
gaussian1_2=40; %vertical stripes
gaussian2_2=10;
gaussiankernel_2=20;
gaussianfilterX_1 = fspecial('gaussian', ...
[gaussian1_2, gaussian2_2],gaussiankernel_2);
gaussian1_3=10; %horizontal stripes
gaussian2_3=40;
gaussiankernel_3=20;
gaussianfilterY_1 = fspecial('gaussian', ...
[gaussian1_3, gaussian2_3],gaussiankernel_3);
gaussian1_4=40;
gaussian2_4=10;
gaussiankernel_4=20;
gaussianfilterX_2 = fspecial('gaussian',...
[gaussian1_4, gaussian2_4],gaussiankernel_4);
gaussian1_5=10;
gaussian2_5=40;
gaussiankernel_5=20;
gaussianfilterY_2 = fspecial('gaussian', ...
[gaussian1_5, gaussian2_5],gaussiankernel_5);

gradientXthreshold_1=0.00001; %thresholding for peakdet.m
gradientYthreshold_1=0.00003;
gradientXthreshold_2=0.000025;
gradientYthreshold_2=0.00002;

%velocity vectoring settings
leftedgecaution=0;
rightedgecaution=20;
topedgecaution=15;
bottomedgecaution=15;

strelsize=27;
%end settings---------------------------------
deltime=delaytime-initialtime; %used for velocity calculation (ns)
configarray=[deltime spatialres ...
gaussian1_1 gaussian2_1 gaussiankernel_1 ...
gaussian1_12 gaussian2_12 gaussiankernel_12 ...
rowbegin rowend colbegin colend ...
shearanglehorizontal shearanglevertical gaussian1_2 gaussian2_2 ...
gaussiankernel_2 gaussian1_3 gaussian2_3 gaussiankernel_3 ...
gaussian1_4 gaussian2_4 gaussiankernel_4 gaussian1_5 ...
gaussian2_5 gaussiankernel_5 gradientXthreshold_1 ...
gradientYthreshold_1 gradientXthreshold_2 ...
gradientYthreshold_2 leftedgecaution rightedgecaution ...
topedgecaution bottomedgecaution strelsize];
configarray=configarray';
if savefiles==1
dlmwrite(strcat(writefile1,num2str(piccount),'_Config_',num2str(initialtime),'_',num2str(
delaytime),'.txt'),configarray,'newline','pc');
end

disp('first image')
filename1=strcat(getfile1,firstfile,'.tif'); %READ FIRST IMAGE
file1=imread(filename1);
file1=im2double(im2uint16(file1));
file1original=file1;

if plotgraphs==1
figure(1)
subplot(3,3,1)
imagesc(file1)
axis image, colormap 'gray';
title(firstfile)
set(1,'Name',filename1)
end

shearform=[1,0,0;shearanglehorizontal,1,0;0,0,1];  %shear image both dimensions
tform_shear=maketform('affine',shearform);
[file1_shear xdata ydata]= imtransform(file1, tform_shear);
file1=file1_shear;
file1=file1';
shearform=[1,0,0;shearanglevertical,1,0;0,0,1];
tform_shear=maketform('affine',shearform);
[file1_shear xdata ydata]= imtransform(file1, tform_shear);
file1=file1_shear;
file1=file1';

file1_1=filter2(gaussianfilterinitial1,file1); %smooth
file1_2=filter2(gaussianfilterinitial2,file1); %smooth
[gradientX,gradientY1]=gradient(file1_1); %take gradients
[gradientX1,gradientY]=gradient(file1_2);

file1=gradientX+gradientY;
if plotgraphs==1
```

```
figure(1)
subplot(3,3,2)
imagesc(file1)
axis image, colormap 'gray';
title('GradientX + GradientY')
end

file1=file1(rowbegin:rowend,colbegin:colend); %cull to ROI
file1size=size(file1)
gradientX=gradientX(rowbegin:rowend,colbegin:colend);
gradientY=gradientY(rowbegin:rowend,colbegin:colend);

if plotgraphs==1
figure(1)
subplot(3,3,3)
imagesc(file1)
axis image, colormap 'gray';
title('ROI gradientX+gradientY')
figure(1)
subplot(3,3,4)
imagesc(gradientX)
axis image, colormap 'gray';
title('gradientX')
figure(1)
subplot(3,3,7)
imagesc(gradientY)
axis image, colormap 'gray';
title('gradientY')
end

gradientX=filter2(gaussianfilterX_1,gradientX);%smooth gradients
gradientY=filter2(gaussianfilterY_1,gradientY);

if plotgraphs==1
figure(1)
subplot(3,3,5)
imagesc(gradientX)
axis image, colormap 'gray';
title('gradientX smoothed')
figure(1)
subplot(3,3,8)
imagesc(gradientY)
axis image, colormap 'gray';
title('gradientY smoothed')
```

```
end

positionsmatrixXmax=zeros(file1size); %find peaks using function peakdet.m (vertical
grid)
positionsmatrixXmin=zeros(file1size);
for rowcounter=1:1:file1size(1) %find peaks using function peakdet.m (vertical grid)
testslice=gradientX(rowcounter,:);
gradientXmax=zeros();
gradientXmin=zeros();
[gradientXmax, gradientXmin]=peakdet(testslice,gradientXthreshold_1);
if rowcounter==round(file1size(1)/2)
if plotgraphs==1
figure(1)
subplot(3,3,6)
plot(testslice)
xlim([1 max(size(testslice))]);           hold on ;
plot(gradientXmax(:,1),gradientXmax(:,2),'*g');       hold on;
plot(gradientXmin(:,1),gradientXmin(:,2),'*r');        hold off;
end
end %  if rowcounter==round(file1size(1)/2)
if max(size(gradientXmax)) >0 && max(size(gradientXmin))>0%if peaks are found
gradientXmax=gradientXmax(:,1);
gradientXmin=gradientXmin(:,1);
for count=1:1:max(size(gradientXmax))
positionsmatrixXmax(rowcounter,gradientXmax(count))=1;
end
for count=1:1:max(size(gradientXmin))
positionsmatrixXmin(rowcounter,gradientXmin(count))=1;
end
end %if max(size(gradientXmax)) >0
end %for rowcounter=1:1:file1size(1)

positionsmatrixYmax=zeros(file1size); %find peaks using function peakdet.m
(horizontal grid)
positionsmatrixYmin=zeros(file1size); %peaks matrix for horizontal stripes
for colcounter=1:1:file1size(2)
testslice=gradientY(:,colcounter);
gradientYmax=zeros();
gradientYmin=zeros();
[gradientYmax,gradientYmin]=peakdet(testslice,gradientYthreshold_1);
if colcounter==round(file1size(2)/2)
if plotgraphs==1
figure(1)
subplot(3,3,9)
```

```
plot(testslice)
xlim([1 max(size(testslice))])
hold on
plot(gradientYmax(:,1),gradientYmax(:,2),'*g'); hold on
plot(gradientYmin(:,1),gradientYmin(:,2),'*r'); hold on
end
end %if colcounter==round(file1size(2)/2)
if max(size(gradientYmax)) >0 && max(size(gradientYmin))>0
gradientYmax=gradientYmax(:,1);
gradientYmin=gradientYmin(:,1);
for count=1:1:max(size(gradientYmax))
positionsmatrixYmax(gradientYmax(count),colcounter)=1;
end
for count=1:1:max(size(gradientYmin))
positionsmatrixYmin(gradientYmin(count),colcounter)=1;
end
end %if max(size(gradientYmax)) >0
end %for rowcounter=1:1:file1size(1)

totalintersectionmax=positionsmatrixXmax+positionsmatrixYmax; %%locations of
intersecting points have values = 2
totalintersectionmin=positionsmatrixXmin+positionsmatrixYmin;
totalintersectionmax2=positionsmatrixXmax+positionsmatrixYmin;
totalintersectionmin2=positionsmatrixXmin+positionsmatrixYmax;
se = strel('disk',5); %for display only
if plotgraphs==1
figure(2)
positionsmatrixXdilatemax = imdilate(positionsmatrixXmax,se);
subplot(4,3,1)
imagesc(positionsmatrixXdilatemax)
axis image,colormap 'gray'
title('0max')
figure(2)
positionsmatrixXdilatemin = imdilate(positionsmatrixXmin,se);
subplot(4,3,4)
imagesc(positionsmatrixXdilatemin)
axis image, colormap 'gray'
title('0min')
figure(2)
subplot(4,3,7)
imagesc(positionsmatrixXdilatemax)
axis image, colormap 'gray'
title('0max2')
figure(2)
```

```
subplot(4,3,10)
imagesc(positionsmatrixXdilatemin)
axis image,  colormap 'gray'
title('0min2')
figure(2)
positionsmatrixYdilatemax = imdilate(positionsmatrixYmax,se);
subplot(4,3,2)
imagesc(positionsmatrixYdilatemax)
axis image, colormap 'gray'
figure(2)
positionsmatrixYdilatemin = imdilate(positionsmatrixYmin,se);
subplot(4,3,5)
imagesc(positionsmatrixYdilatemin)
axis image, colormap 'gray'
figure(2)
subplot(4,3,8)
imagesc(positionsmatrixYdilatemin)
axis image, colormap 'gray'
figure(2)
subplot(4,3,11)
imagesc(positionsmatrixYdilatemax)
axis image, colormap 'gray'
end

intersectionmapmax=zeros(file1size);
intersectionmapmin=zeros(file1size);
intersectionmapmax2=zeros(file1size);
intersectionmapmin2=zeros(file1size);
for rowcounter=1:1:file1size(1)
for colcounter=1:1:file1size(2)
if totalintersectionmax(rowcounter,colcounter)==2
intersectionmapmax(rowcounter,colcounter)=1;
end
if totalintersectionmin(rowcounter,colcounter)==2
intersectionmapmin(rowcounter,colcounter)=1;
end
if totalintersectionmax2(rowcounter,colcounter)==2
intersectionmapmax2(rowcounter,colcounter)=1;
end
if totalintersectionmin2(rowcounter,colcounter)==2
intersectionmapmin2(rowcounter,colcounter)=1;
end
end
end
```

```
shearform=[1,0,0;-shearanglehorizontal,1,0;0,0,1]; %reverse shear
tform_shear=maketform('affine',shearform);
[intersectionmap_shearmax xdata ydata]= imtransform(intersectionmapmax,
tform_shear,'nearest');
intersectionmapmax=intersectionmap_shearmax;
intersectionmapmax=intersectionmapmax';
shearform=[1,0,0;-shearanglevertical,1,0;0,0,1];
tform_shear=maketform('affine',shearform);
[intersectionmap_shearmax xdata ydata]= imtransform(intersectionmapmax,
tform_shear,'nearest');
intersectionmapmax=intersectionmap_shearmax;
intersectionmapmax=intersectionmapmax';

shearform=[1,0,0;-shearanglehorizontal,1,0;0,0,1];
tform_shear=maketform('affine',shearform);
[intersectionmap_shearmin xdata ydata]= imtransform(intersectionmapmin,
tform_shear,'nearest');
intersectionmapmin=intersectionmap_shearmin;
intersectionmapmin=intersectionmapmin';
shearform=[1,0,0;-shearanglevertical,1,0;0,0,1];
tform_shear=maketform('affine',shearform);
[intersectionmap_shearmin xdata ydata]= imtransform(intersectionmapmin,
tform_shear,'nearest');
intersectionmapmin=intersectionmap_shearmin;
intersectionmapmin=intersectionmapmin';

shearform=[1,0,0;-shearanglehorizontal,1,0;0,0,1]; %reverse shear
tform_shear=maketform('affine',shearform);
[intersectionmap_shearmax2 xdata ydata]= imtransform(intersectionmapmax2,
tform_shear,'nearest');
intersectionmapmax2=intersectionmap_shearmax2;
intersectionmapmax2=intersectionmapmax2';
shearform=[1,0,0;-shearanglevertical,1,0;0,0,1];
tform_shear=maketform('affine',shearform);
[intersectionmap_shearmax2 xdata ydata]= imtransform(intersectionmapmax2,
tform_shear,'nearest');
intersectionmapmax2=intersectionmap_shearmax2;
intersectionmapmax2=intersectionmapmax2';

shearform=[1,0,0;-shearanglehorizontal,1,0;0,0,1];
tform_shear=maketform('affine',shearform);
[intersectionmap_shearmin2 xdata ydata]= imtransform(intersectionmapmin2,
tform_shear,'nearest');
```

```
intersectionmapmin2=intersectionmap_shearmin2;
intersectionmapmin2=intersectionmapmin2';
shearform=[1,0,0;-shearanglevertical,1,0;0,0,1];
tform_shear=maketform('affine',shearform);
[intersectionmap_shearmin2 xdata ydata]= imtransform(intersectionmapmin2,
tform_shear,'nearest');
intersectionmapmin2=intersectionmap_shearmin2;
intersectionmapmin2=intersectionmapmin2';

se = strel('disk',10);
intersectionmapdilatemax=imdilate(intersectionmapmax,se);
intersectionmapdilatemin=imdilate(intersectionmapmin,se);
intersectionmapdilatemax2=imdilate(intersectionmapmax2,se);
intersectionmapdilatemin2=imdilate(intersectionmapmin2,se);

if plotgraphs==1
figure(2)
subplot(4,3,3)
imagesc(intersectionmapdilatemax)
axis image, colormap 'gray'
figure(2)
subplot(4,3,6)
imagesc(intersectionmapdilatemin)
axis image, colormap 'gray'
figure(2)
subplot(4,3,9)
imagesc(intersectionmapdilatemax2)
axis image, colormap 'gray'
figure(2)
subplot(4,3,12)
imagesc(intersectionmapdilatemin2)
axis image, colormap 'gray'
end
points1max=intersectionmapmax;
points1dilatemax=intersectionmapdilatemax;
points1min=intersectionmapmin;
points1dilatemin=intersectionmapdilatemin;
points1max2=intersectionmapmax2;
points1dilatemax2=intersectionmapdilatemax2;
points1min2=intersectionmapmin2;
points1dilatemin2=intersectionmapdilatemin2;
finalsize1=size(points1max)

disp('second image')
```

```
filename1=strcat(getfile1,secondfile,'.tif'); %READ SECOND IMAGE
file1=imread(filename1);
file1=im2double(im2uint16(file1));
if plotgraphs==1
figure(3)
subplot(3,3,1)
imagesc(file1)
axis image, colormap 'gray';
title(secondfile)
set(3,'Name',filename1)
end

shearform=[1,0,0;shearanglehorizontal,1,0;0,0,1]; %shear image
tform_shear=maketform('affine',shearform);
[file1_shear xdata ydata]= imtransform(file1, tform_shear);
file1=file1_shear;
file1=file1';
shearform=[1,0,0;shearanglevertical,1,0;0,0,1];
tform_shear=maketform('affine',shearform);
[file1_shear xdata ydata]= imtransform(file1, tform_shear);
file1=file1_shear;
file1=file1';

file1_1=filter2(gaussianfilterinitial1,file1); %smooth
file1_2=filter2(gaussianfilterinitial2,file1); %smooth
[gradientX,gradientY1]=gradient(file1_1); %take gradients
[gradientX1,gradientY]=gradient(file1_2);
file1=gradientX+gradientY;
if plotgraphs==1
figure(3)
subplot(3,3,2)
imagesc(file1)
axis image, colormap 'gray';
title('GradientX+GradientY')
end
file1=file1(rowbegin:rowend,colbegin:colend); %cull ROI
gradientX=gradientX(rowbegin:rowend,colbegin:colend);
gradientY=gradientY(rowbegin:rowend,colbegin:colend);
if plotgraphs==1
figure(3)
subplot(3,3,3)
imagesc(file1)
axis image, colormap 'gray';
title('ROI gradientX+gradientY')
```

```
figure(3)
subplot(3,3,4)
imagesc(gradientX)
axis image, colormap 'gray';
title('gradientX')
figure(3)
subplot(3,3,7)
imagesc(gradientY)
axis image, colormap 'gray';
title('gradientY')
end
gradientX=filter2(gaussianfilterX_2,gradientX); %smooth gradients
gradientY=filter2(gaussianfilterY_2,gradientY);
if plotgraphs==1
figure(3)
subplot(3,3,5)
imagesc(gradientX)
axis image, colormap 'gray';
title('gradientX smoothed')
figure(3)
subplot(3,3,8)
imagesc(gradientY)
axis image, colormap 'gray';
title('gradientY smoothed')
end

positionsmatrixXmax=zeros(file1size); %find peaks using function peakdet.m (vertical grid)
positionsmatrixXmin=zeros(file1size);
for rowcounter=1:1:file1size(1)
testslice=gradientX(rowcounter,:);
gradientXmax=zeros();
gradientXmin=zeros();

[gradientXmax,gradientXmin]=peakdet(testslice,gradientXthreshold_2);
if rowcounter==round(file1size(1)/2)

end %  if rowcounter==round(file1size(1)/2)
if max(size(gradientXmax)) >0 &&max(size(gradientXmin)) >0
gradientXmax=gradientXmax(:,1);
gradientXmin=gradientXmin(:,1);
for count=1:1:max(size(gradientXmax))
positionsmatrixXmax(rowcounter,gradientXmax(count))=1;
end
```

```
for count=1:1:max(size(gradientXmin))
positionsmatrixXmin(rowcounter,gradientXmin(count))=1;
end
end %if max(size(gradientXmax)) >0
end %for rowcounter=1:1:file1size(1)

positionsmatrixYmax=zeros(file1size);%find peaks using function peakdet.m (horizontal grid)
positionsmatrixYmin=zeros(file1size);
for colcounter=1:1:file1size(2)
testslice=gradientY(:,colcounter);
gradientYmax=zeros();
gradientYmin=zeros();
[gradientYmax,gradientYmin]=peakdet(testslice,gradientYthreshold_2);
if colcounter==round(file1size(2)/2)

end %if colcounter==round(file1size(2)/2)
if max(size(gradientYmax)) >0 &&max(size(gradientYmin)) >0
gradientYmax=gradientYmax(:,1);
gradientYmin=gradientYmin(:,1);

for count=1:1:max(size(gradientYmax))
positionsmatrixYmax(gradientYmax(count),colcounter)=1;
end
for count=1:1:max(size(gradientYmin))
positionsmatrixYmin(gradientYmin(count),colcounter)=1;
end
end %if max(size(gradientYmax)) >0
end %for rowcounter=1:1:file1size(1)

totalintersectionmax=positionsmatrixXmax+positionsmatrixYmax; %%locations of intersecting points have values = 2
totalintersectionmin=positionsmatrixXmin+positionsmatrixYmin;
totalintersectionmax2=positionsmatrixXmax+positionsmatrixYmin;
totalintersectionmin2=positionsmatrixXmin+positionsmatrixYmax;
se = strel('disk',5);
if plotgraphs==1
figure(4)
positionsmatrixXdilatemax = imdilate(positionsmatrixXmax,se);
subplot(4,3,1)
imagesc(positionsmatrixXdilatemax)
axis image, colormap 'gray'
title(strcat(secondfile,'max'))
figure(4)
```

```
positionsmatrixXdilatemin = imdilate(positionsmatrixXmin,se);
subplot(4,3,4)
imagesc(positionsmatrixXdilatemin)
axis image, colormap 'gray'
title(strcat(secondfile,'min'))
figure(4)
subplot(4,3,7)
imagesc(positionsmatrixXdilatemax)
axis image, colormap 'gray'
title(strcat(secondfile,'max2'))
figure(4)
subplot(4,3,10)
imagesc(positionsmatrixXdilatemin)
axis image, colormap 'gray'
title(strcat(secondfile,'min2'))
figure(4)
positionsmatrixYdilatemax = imdilate(positionsmatrixYmax,se);
subplot(4,3,2)
imagesc(positionsmatrixYdilatemax)
axis image, colormap 'gray'
figure(4)
positionsmatrixYdilatemin = imdilate(positionsmatrixYmin,se);
subplot(4,3,5)
imagesc(positionsmatrixYdilatemin)
axis image, colormap 'gray'
figure(4)
subplot(4,3,8)
imagesc(positionsmatrixYdilatemin)
axis image, colormap 'gray'
figure(4)
subplot(4,3,11)
imagesc(positionsmatrixYdilatemax)
axis image, colormap 'gray'
end
intersectionmapmax=zeros(file1size);
intersectionmapmin=zeros(file1size);
intersectionmapmax2=zeros(file1size);
intersectionmapmin2=zeros(file1size);
for rowcounter=1:1:file1size(1)
for colcounter=1:1:file1size(2)
if totalintersectionmax(rowcounter,colcounter)==2
intersectionmapmax(rowcounter,colcounter)=1;
end
if totalintersectionmin(rowcounter,colcounter)==2
```

```
intersectionmapmin(rowcounter,colcounter)=1;
end
if totalintersectionmax2(rowcounter,colcounter)==2
intersectionmapmax2(rowcounter,colcounter)=1;
end
if totalintersectionmin2(rowcounter,colcounter)==2
intersectionmapmin2(rowcounter,colcounter)=1;
end
end
end

shearform=[1,0,0;-shearanglehorizontal,1,0;0,0,1]; %reverse shear
tform_shear=maketform('affine',shearform);
[intersectionmap_shearmax xdata ydata]= imtransform(intersectionmapmax,
tform_shear,'nearest');
intersectionmapmax=intersectionmap_shearmax;
intersectionmapmax=intersectionmapmax';
shearform=[1,0,0;-shearanglevertical,1,0;0,0,1];
tform_shear=maketform('affine',shearform);
[intersectionmap_shearmax xdata ydata]= imtransform(intersectionmapmax,
tform_shear,'nearest');
intersectionmapmax=intersectionmap_shearmax;
intersectionmapmax=intersectionmapmax';
shearform=[1,0,0;-shearanglehorizontal,1,0;0,0,1];
tform_shear=maketform('affine',shearform);
[intersectionmap_shearmin xdata ydata]= imtransform(intersectionmapmin,
tform_shear,'nearest');
intersectionmapmin=intersectionmap_shearmin;
intersectionmapmin=intersectionmapmin';
shearform=[1,0,0;-shearanglevertical,1,0;0,0,1];
tform_shear=maketform('affine',shearform);
[intersectionmap_shearmin xdata ydata]= imtransform(intersectionmapmin,
tform_shear,'nearest');
intersectionmapmin=intersectionmap_shearmin;
intersectionmapmin=intersectionmapmin';
shearform=[1,0,0;-shearanglehorizontal,1,0;0,0,1]; %reverse shear
tform_shear=maketform('affine',shearform);
[intersectionmap_shearmax2 xdata ydata]= imtransform(intersectionmapmax2,
tform_shear,'nearest');
intersectionmapmax2=intersectionmap_shearmax2;
intersectionmapmax2=intersectionmapmax2';
shearform=[1,0,0;-shearanglevertical,1,0;0,0,1];
tform_shear=maketform('affine',shearform);
```

```
[intersectionmap_shearmax2 xdata ydata]= imtransform(intersectionmapmax2,
tform_shear,'nearest');
intersectionmapmax2=intersectionmap_shearmax2;
intersectionmapmax2=intersectionmapmax2';
shearform=[1,0,0;-shearanglehorizontal,1,0;0,0,1];
tform_shear=maketform('affine',shearform);
[intersectionmap_shearmin2 xdata ydata]= imtransform(intersectionmapmin2,
tform_shear,'nearest');
intersectionmapmin2=intersectionmap_shearmin2;
intersectionmapmin2=intersectionmapmin2';
shearform=[1,0,0;-shearanglevertical,1,0;0,0,1];
tform_shear=maketform('affine',shearform);
[intersectionmap_shearmin2 xdata ydata]= imtransform(intersectionmapmin2,
tform_shear,'nearest');
intersectionmapmin2=intersectionmap_shearmin2;
intersectionmapmin2=intersectionmapmin2';
se = strel('disk',10);

intersectionmapdilatemax=imdilate(intersectionmapmax,se);
intersectionmapdilatemin=imdilate(intersectionmapmin,se);
intersectionmapdilatemax2=imdilate(intersectionmapmax2,se);
intersectionmapdilatemin2=imdilate(intersectionmapmin2,se);

if plotgraphs==1
figure(4)
subplot(4,3,3)
imagesc(intersectionmapdilatemax)
axis image, colormap 'gray'
figure(4)
subplot(4,3,6)
imagesc(intersectionmapdilatemin)
axis image, colormap 'gray'
figure(4)
subplot(4,3,9)
imagesc(intersectionmapdilatemax2)
axis image, colormap 'gray'
figure(4)
subplot(4,3,12)
imagesc(intersectionmapdilatemin2)
axis image, colormap 'gray'
end

points2max=intersectionmapmax;
points2dilatemax=intersectionmapdilatemax;
```

```
points2min=intersectionmapmin;
points2dilatemin=intersectionmapdilatemin;
points2max2=intersectionmapmax2;
points2dilatemax2=intersectionmapdilatemax2;
points2min2=intersectionmapmin2;
points2dilatemin2=intersectionmapdilatemin2;
finalsize2=size(points2max)

disp('Calculating max-max velocities')% BEGIN CALCULATING VELOCITIES
velocitygridmax=zeros();
pointcountmax=0;
for rowcounter=(topedgecaution+1):1:file1size(1)-(bottomedgecaution+1) %roam ROI
for colcounter=(leftedgecaution+1):1:file1size(2)-(rightedgecaution+1)
rowcounter;
if points1max(rowcounter,colcounter)==1
if filteryvel==1
if rowcounter<round(file1size(1)/2)-10&&colcounter<round(file1size(2)*2/3)
bottomedgecautiontemp=round(bottomedgecaution*0.25);
topedgecautiontemp=topedgecaution;
end
if rowcounter>=round(file1size(1)/2)+10&&colcounter<round(file1size(2)*2/3)
bottomedgecautiontemp=bottomedgecaution;
topedgecautiontemp=round(topedgecaution*0.25);
end
if colcounter>=round(file1size(2)*2/3)
bottomedgecautiontemp=bottomedgecaution; topedgecautiontemp=topedgecaution;
end
end
if filteryvel==0
bottomedgecautiontemp=bottomedgecaution; topedgecautiontemp=topedgecaution;
end
for rowcounter2=rowcounter-
topedgecautiontemp:1:rowcounter+bottomedgecautiontemp
for colcounter2=colcounter-leftedgecaution:1:colcounter+rightedgecaution
if points2max(rowcounter2,colcounter2)==1
if (colcounter2-colcounter)>=-leftedgecaution
pointcountmax=pointcountmax+1; %count points found
velocitygridmax(pointcountmax,1)=rowcounter; %y initial
velocitygridmax(pointcountmax,2)=colcounter; %x initial
velocitygridmax(pointcountmax,3)=rowcounter2; %y final
velocitygridmax(pointcountmax,4)=colcounter2; %x final
velocitygridmax(pointcountmax,5)=round(colcounter+(colcounter2-colcounter)/2); % x
midpoint
```

```
velocitygridmax(pointcountmax,6)=round(rowcounter+(rowcounter2-
rowcounter)/2); %y midpoint
velocitygridmax(pointcountmax,7)=(rowcounter2-rowcounter); %y displacement,
^pixels
velocitygridmax(pointcountmax,8)=(colcounter2-colcounter);%x  displacement, ^pixels
velocitygridmax(pointcountmax,9)=sqrt(velocitygridmax(pointcountmax,7)^2+velocityg
ridmax(pointcountmax,8)^2); %magnitude of velocity (^pixels)
velocitygridmax(pointcountmax,10)=
velocitygridmax(pointcountmax,9)/deltime*10^9/1000/spatialres;%magnitude of
velocity (m/s)
velocitygridmax(pointcountmax,11)=velocitygridmax(pointcountmax,8)/deltime*10^9/1
000/spatialres;%x velocity (m/s)
velocitygridmax(pointcountmax,12)=velocitygridmax(pointcountmax,7)/deltime*10^9/1
000/spatialres;%y velocity (m/s)
end %if (colcounter2-colcounter)>0
end %if points2(rowcounter2,colcounter2)==1
end %for colcounter2=colcounter-leftedgecaution:1:colcounter+rightedgecaution
end %for rowcounter2=rowcounter-topedgecaution:1:rowcounter+bottomedgecaution
end %if points1(rowcounter,colcounter)==1
end %for colcounter=(leftedgecaution+1):1:file1size(2)-(rightedgecaution+1)
end %for rowcounter=(topedgecaution+1):1:file1size(1)-(bottomedgecaution+1)
disp('   Filtering for multiple vectors...');
for iteration=1:1:pointcountmax-1
if
velocitygridmax(iteration,1)==velocitygridmax(iteration+1,1)&&velocitygridmax(iterati
on,2)==velocitygridmax(iteration+1,2)
if abs(velocitygridmax(iteration,7))<=abs(velocitygridmax(iteration+1,7)) %take nearest
velocitygridmax(iteration+1,:)=velocitygridmax(iteration,:);
velocitygridmax(iteration,:)=[1 1 1 1 1 1 0 0 -1000 -1000 -1000 -1000];
end
if abs(velocitygridmax(iteration,7))>abs(velocitygridmax(iteration+1,7)) %take nearest
velocitygridmax(iteration,:)=[1 1 1 1 1 1 0 0 -1000 -1000 -1000 -1000];
end
end%if
velocitygridmax(iteration,1)==velocitygridmax(iteration+1,1)&&velocitygridmax(iterati
on,2)==velocitygridmax(iteration+1,2)
end %for iteration=1:1:pointcountmax-1
velocitymapmax =zeros(file1size); %overall magnitudes
velocitymapmax=velocitymapmax-1000;
for iterateplot=1:1:pointcountmax
velocitymapmax(velocitygridmax(iterateplot,6),
velocitygridmax(iterateplot,5))=velocitygridmax(iterateplot,10);
end
se = strel('octagon',strelsize);
```

```
velocitymapdilatemax=imdilate(velocitymapmax,se);
velocitymapsizemax=size(velocitymapdilatemax);
velocitymapXmax =zeros(file1size); %X velocities
velocitymapXmax=velocitymapXmax-1000;
for iterateplot=1:1:pointcountmax
velocitymapXmax(velocitygridmax(iterateplot,6),
velocitygridmax(iterateplot,5))=velocitygridmax(iterateplot,11);
end
se = strel('octagon',strelsize);
velocitymapdilateXmax=imdilate(velocitymapXmax,se);
velocitymapsizeXmax=size(velocitymapdilateXmax);
velocitymapYmax =zeros(file1size); %Y velocities
velocitymapYmax=velocitymapYmax-1000;
for iterateplot=1:1:pointcountmax
velocitymapYmax(velocitygridmax(iterateplot,6),
velocitygridmax(iterateplot,5))=abs(velocitygridmax(iterateplot,12));
end
se = strel('octagon',strelsize);
velocitymapdilateYmax=imdilate(velocitymapYmax,se);
velocitymapsizeYmax=size(velocitymapdilateYmax);
velocitymapYmaxraw =zeros(file1size); %Y velocities
velocitymapYmaxraw=velocitymapYmaxraw-1000;
for iterateplot=1:1:pointcountmax
velocitymapYmaxraw(velocitygridmax(iterateplot,6),
velocitygridmax(iterateplot,5))=(velocitygridmax(iterateplot,12));
end
se = strel('octagon',strelsize);
velocitymapdilateYmaxraw=imdilate(velocitymapYmaxraw,se);
velocitymapsizeYmaxraw=size(velocitymapdilateYmaxraw);
if savefiles==1
dlmwrite(strcat(writefile1,num2str(piccount),'_Velocitymapdilatemax_',num2str(initialti
me),'_',num2str(delaytime),'.txt'),velocitymapdilatemax,'\t');
dlmwrite(strcat(writefile1,num2str(piccount),'_Velocitygridmax_',num2str(initialtime),'_
',num2str(delaytime),'.txt'),velocitygridmax,'\t');
end

disp('Calculating max-min velocities')
velocitygridmax2=zeros();
pointcountmax2=0;
for rowcounter=(topedgecaution+1):1:file1size(1)-(bottomedgecaution+1) %roam ROI
for colcounter=(leftedgecaution+1):1:file1size(2)-(rightedgecaution+1)
rowcounter;
if points1max2(rowcounter,colcounter)==1
if filteryvel==1
if rowcounter<round(file1size(1)/2)-10&&colcounter<round(file1size(2)*2/3)
bottomedgecautiontemp=round(bottomedgecaution*0.25);
topedgecautiontemp=topedgecaution;
end
if rowcounter>=round(file1size(1)/2)+10&&colcounter<round(file1size(2)*2/3)
bottomedgecautiontemp=bottomedgecaution;
topedgecautiontemp=round(topedgecaution*0.25);
end
if colcounter>=round(file1size(2)*2/3)
bottomedgecautiontemp=bottomedgecaution; topedgecautiontemp=topedgecaution;
end
end
if filteryvel==0
bottomedgecautiontemp=bottomedgecaution; topedgecautiontemp=topedgecaution;
end
for rowcounter2=rowcounter-
topedgecautiontemp:1:rowcounter+bottomedgecautiontemp
for colcounter2=colcounter-leftedgecaution:1:colcounter+rightedgecaution
if points2max2(rowcounter2,colcounter2)==1
if (colcounter2-colcounter)>=-leftedgecaution

pointcountmax2=pointcountmax2+1;
velocitygridmax2(pointcountmax2,1)=rowcounter; %y initial
velocitygridmax2(pointcountmax2,2)=colcounter; %x initial
velocitygridmax2(pointcountmax2,3)=rowcounter2; %y final
velocitygridmax2(pointcountmax2,4)=colcounter2; %x final
velocitygridmax2(pointcountmax2,5)=round(colcounter+(colcounter2-colcounter)/2); %
x midpoint
velocitygridmax2(pointcountmax2,6)=round(rowcounter+(rowcounter2-
rowcounter)/2); %y midpoint
velocitygridmax2(pointcountmax2,7)=(rowcounter2-rowcounter); %y displacement,
^pixels
velocitygridmax2(pointcountmax2,8)=(colcounter2-colcounter);%x  displacement,
^pixels
velocitygridmax2(pointcountmax2,9)=sqrt(velocitygridmax2(pointcountmax2,7)^2+velo
citygridmax2(pointcountmax2,8)^2); %magnitude of velocity (^pixels)
velocitygridmax2(pointcountmax2,10)=
velocitygridmax2(pointcountmax2,9)/deltime*10^9/1000/spatialres;%magnitude of
velocity (m/s)
velocitygridmax2(pointcountmax2,11)=velocitygridmax2(pointcountmax2,8)/deltime*1
0^9/1000/spatialres;%x velocity (m/s)
velocitygridmax2(pointcountmax2,12)=velocitygridmax2(pointcountmax2,7)/deltime*1
0^9/1000/spatialres;%y velocity (m/s)
end %if (colcounter2-colcounter)>0
end %if points2(rowcounter2,colcounter2)==1
end %for colcounter2=colcounter-leftedgecaution:1:colcounter+rightedgecaution
end %for rowcounter2=rowcounter-topedgecaution:1:rowcounter+bottomedgecaution
end %if points1(rowcounter,colcounter)==1
end %for colcounter=(leftedgecaution+1):1:file1size(2)-(rightedgecaution+1)
end %for rowcounter=(topedgecaution+1):1:file1size(1)-(bottomedgecaution+1)

disp('   Filtering for multiple vectors...');
for iteration=1:1:pointcountmax2-1
if
velocitygridmax2(iteration,1)==velocitygridmax2(iteration+1,1)&&velocitygridmax2(it
eration,2)==velocitygridmax2(iteration+1,2)
if abs(velocitygridmax2(iteration,7))<=abs(velocitygridmax2(iteration+1,7)) %take
nearest
velocitygridmax2(iteration+1,:)=velocitygridmax2(iteration,:);
velocitygridmax2(iteration,:)=[1 1 1 1 1 1 0 0 -1000 -1000 -1000 -1000];
end
if abs(velocitygridmax2(iteration,7))>abs(velocitygridmax2(iteration+1,7)) %take
nearest
velocitygridmax2(iteration,:)=[1 1 1 1 1 1 0 0 -1000 -1000 -1000 -1000];
end
end%if
velocitygridmax(iteration,1)==velocitygridmax(iteration+1,1)&&velocitygridmax(iterati
on,2)==velocitygridmax(iteration+1,2)
end %for iteration=1:1:pointcountmax-1
velocitymapmax2 =zeros(file1size); %overall magnitudes
velocitymapmax2=velocitymapmax2-1000;
for iterateplot=1:1:pointcountmax2
velocitymapmax2(velocitygridmax2(iterateplot,6),
velocitygridmax2(iterateplot,5))=velocitygridmax2(iterateplot,10);
end
se = strel('octagon',strelsize);
velocitymapdilatemax2=imdilate(velocitymapmax2,se);
velocitymapsizemax2=size(velocitymapdilatemax2);
velocitymapXmax2 =zeros(file1size); %X velocities
velocitymapXmax2=velocitymapXmax2-1000;
for iterateplot=1:1:pointcountmax2
velocitymapXmax2(velocitygridmax2(iterateplot,6),
velocitygridmax2(iterateplot,5))=velocitygridmax2(iterateplot,11);
end
se = strel('octagon',strelsize);
velocitymapdilateXmax2=imdilate(velocitymapXmax2,se);
velocitymapsizeXmax2=size(velocitymapdilateXmax2);
velocitymapYmax2 =zeros(file1size); %Y velocities
```

```
velocitymapYmax2=velocitymapYmax2-1000;
for iterateplot=1:1:pointcountmax2
velocitymapYmax2(velocitygridmax2(iterateplot,6),
velocitygridmax2(iterateplot,5))=abs(velocitygridmax2(iterateplot,12));
end
se = strel('octagon',strelsize);
velocitymapdilateYmax2=imdilate(velocitymapYmax2,se);
velocitymapsizeYmax2=size(velocitymapdilateYmax2);
velocitymapYmax2raw =zeros(file1size); %Y velocities
velocitymapYmax2raw=velocitymapYmax2raw-1000;
for iterateplot=1:1:pointcountmax2
velocitymapYmax2raw(velocitygridmax2(iterateplot,6),
velocitygridmax2(iterateplot,5))=(velocitygridmax2(iterateplot,12));
end
se = strel('octagon',strelsize);
velocitymapdilateYmax2raw=imdilate(velocitymapYmax2raw,se);
velocitymapsizeYmax2raw=size(velocitymapdilateYmax2raw);
if savefiles==1
dlmwrite(strcat(writefile1,num2str(piccount),'_Velocitymapdilatemax2_',num2str(initialt
ime),'_',num2str(delaytime),'.txt'),velocitymapdilatemax2,'\t');
dlmwrite(strcat(writefile1,num2str(piccount),'_Velocitygridmax2_',num2str(initialtime),'
_',num2str(delaytime),'.txt'),velocitygridmax2,'\t');
end

disp('Calculating min-min velocities')
velocitygridmin=zeros(); %calculate velocities
pointcountmin=0;
for rowcounter=(topedgecaution+1):1:file1size(1)-(bottomedgecaution+1)
for colcounter=(leftedgecaution+1):1:file1size(2)-(rightedgecaution+1)
rowcounter;
if points1min(rowcounter,colcounter)==1
if filteryvel==1
if rowcounter<round(file1size(1)/2)-10&&colcounter<round(file1size(2)*2/3)
bottomedgecautiontemp=round(bottomedgecaution*0.25);
topedgecautiontemp=topedgecaution;
end
if rowcounter>=round(file1size(1)/2)+10&&colcounter<round(file1size(2)*2/3)
bottomedgecautiontemp=bottomedgecaution;
topedgecautiontemp=round(topedgecaution*0.25);
end
if colcounter>=round(file1size(2)*2/3)
bottomedgecautiontemp=bottomedgecaution; topedgecautiontemp=topedgecaution;
end
end
```

```
if filteryvel==0
bottomedgecautiontemp=bottomedgecaution; topedgecautiontemp=topedgecaution;
end
for rowcounter2=rowcounter-
topedgecautiontemp:1:rowcounter+bottomedgecautiontemp
for colcounter2=colcounter-leftedgecaution:1:colcounter+rightedgecaution
if points2min(rowcounter2,colcounter2)==1
if (colcounter2-colcounter)>=-leftedgecaution
pointcountmin=pointcountmin+1;
velocitygridmin(pointcountmin,1)=rowcounter; %y initial
velocitygridmin(pointcountmin,2)=colcounter; %x initial
velocitygridmin(pointcountmin,3)=rowcounter2; %y final
velocitygridmin(pointcountmin,4)=colcounter2; %x final
velocitygridmin(pointcountmin,5)=round(colcounter+(colcounter2-colcounter)/2); % x
midpoint
velocitygridmin(pointcountmin,6)=round(rowcounter+(rowcounter2-rowcounter)/2); %y
midpoint
velocitygridmin(pointcountmin,7)=(rowcounter2-rowcounter); %y displacement, ^pixels
velocitygridmin(pointcountmin,8)=(colcounter2-colcounter);%x  displacement, ^pixels
velocitygridmin(pointcountmin,9)=sqrt(velocitygridmin(pointcountmin,7)^2+velocitygri
dmin(pointcountmin,8)^2); %magnitude of velocity (^pixels)
velocitygridmin(pointcountmin,10)=
velocitygridmin(pointcountmin,9)/deltime*10^9/1000/spatialres;%magnitude of velocity
(m/s)
velocitygridmin(pointcountmin,11)=velocitygridmin(pointcountmin,8)/deltime*10^9/10
00/spatialres;%x velocity (m/s)
velocitygridmin(pointcountmin,12)=velocitygridmin(pointcountmin,7)/deltime*10^9/10
00/spatialres;%y velocity (m/s)
end %if (colcounter2-colcounter)>0
end %if points2(rowcounter2,colcounter2)==1
end %for colcounter2=colcounter-leftedgecaution:1:colcounter+rightedgecaution
end %for rowcounter2=rowcounter-topedgecaution:1:rowcounter+bottomedgecaution
end %if points1(rowcounter,colcounter)==1
end %for colcounter=(leftedgecaution+1):1:file1size(2)-(rightedgecaution+1)
end %for rowcounter=(topedgecaution+1):1:file1size(1)-(bottomedgecaution+1)
disp('   Filtering for multiple vectors...');
for iteration=1:1:pointcountmin-1
if
velocitygridmin(iteration,1)==velocitygridmin(iteration+1,1)&&velocitygridmin(iteratio
n,2)==velocitygridmin(iteration+1,2)
if abs(velocitygridmin(iteration,7))<=abs(velocitygridmin(iteration+1,7)) %take nearest
velocitygridmin(iteration+1,:)=velocitygridmin(iteration,:);
velocitygridmin(iteration,:)=[1 1 1 1 1 1 0 0 -1000 -1000 -1000 -1000];
end
```

```
if abs(velocitygridmin(iteration,7))>abs(velocitygridmin(iteration+1,7)) %take nearest
velocitygridmin(iteration,:)=[1 1 1 1 1 1 0 0 -1000 -1000 -1000 -1000];
end
end%if
velocitygridmax(iteration,1)==velocitygridmax(iteration+1,1)&&velocitygridmax(iterati
on,2)==velocitygridmax(iteration+1,2)
end %for iteration=1:1:pointcountmax-1
velocitymapmin =zeros(file1size); %overall magnitudes
velocitymapmin=velocitymapmin-1000;
for iterateplot=1:1:pointcountmin
velocitymapmin(velocitygridmin(iterateplot,6),
velocitygridmin(iterateplot,5))=velocitygridmin(iterateplot,10);
end
se = strel('octagon',strelsize);
velocitymapdilatemin=imdilate(velocitymapmin,se);
velocitymapsizemin=size(velocitymapdilatemin);
velocitymapXmin =zeros(file1size); %X velocities
velocitymapXmin=velocitymapXmin-1000;
for iterateplot=1:1:pointcountmin
velocitymapXmin(velocitygridmin(iterateplot,6),
velocitygridmin(iterateplot,5))=velocitygridmin(iterateplot,11);
end
se = strel('octagon',strelsize);
velocitymapdilateXmin=imdilate(velocitymapXmin,se);
velocitymapsizeXmin=size(velocitymapdilateXmin);
velocitymapYmin =zeros(file1size); %Y velocities
velocitymapYmin=velocitymapYmin-1000;
for iterateplot=1:1:pointcountmin
velocitymapYmin(velocitygridmin(iterateplot,6),
velocitygridmin(iterateplot,5))=abs(velocitygridmin(iterateplot,12));
end
se = strel('octagon',strelsize);
velocitymapdilateYmin=imdilate(velocitymapYmin,se);
velocitymapsizeYmin=size(velocitymapdilateYmin);
velocitymapYminraw =zeros(file1size); %Y velocities
velocitymapYminraw=velocitymapYminraw-1000;
for iterateplot=1:1:pointcountmin
velocitymapYminraw(velocitygridmin(iterateplot,6),
velocitygridmin(iterateplot,5))=(velocitygridmin(iterateplot,12));
end
se = strel('octagon',strelsize);
velocitymapdilateYminraw=imdilate(velocitymapYminraw,se);
velocitymapsizeYminraw=size(velocitymapdilateYminraw);
if savefiles==1
```

```
dlmwrite(strcat(writefile1,num2str(piccount),'_Velocitymapdilatemin_',num2str(initialti
me),'_',num2str(delaytime),'.txt'),velocitymapdilatemin,'\t');
dlmwrite(strcat(writefile1,num2str(piccount),'_Velocitygridmin_',num2str(initialtime),'_'
,num2str(delaytime),'.txt'),velocitygridmin,'\t');
end

disp('Calculating min-max velocities')
velocitygridmin2=zeros(); %calculate velocities
pointcountmin2=0;
for rowcounter=(topedgecaution+1):1:file1size(1)-(bottomedgecaution+1)
for colcounter=(leftedgecaution+1):1:file1size(2)-(rightedgecaution+1)
rowcounter;
if points1min2(rowcounter,colcounter)==1
if filteryvel==1
if rowcounter<round(file1size(1)/2)-10&&colcounter<round(file1size(2)*2/3)
bottomedgecautiontemp=round(bottomedgecaution*0.25);
topedgecautiontemp=topedgecaution;
end
if rowcounter>=round(file1size(1)/2)+10&&colcounter<round(file1size(2)*2/3)
bottomedgecautiontemp=bottomedgecaution;
topedgecautiontemp=round(topedgecaution*0.25);
end
if colcounter>=round(file1size(2)*2/3)
bottomedgecautiontemp=bottomedgecaution; topedgecautiontemp=topedgecaution;
end
end
if filteryvel==0
bottomedgecautiontemp=bottomedgecaution; topedgecautiontemp=topedgecaution;
end
for rowcounter2=rowcounter-
topedgecautiontemp:1:rowcounter+bottomedgecautiontemp
for colcounter2=colcounter-leftedgecaution:1:colcounter+rightedgecaution
if points2min2(rowcounter2,colcounter2)==1
if (colcounter2-colcounter)>=-leftedgecaution
pointcountmin2=pointcountmin2+1;
velocitygridmin2(pointcountmin2,1)=rowcounter; %y initial
velocitygridmin2(pointcountmin2,2)=colcounter; %x initial
velocitygridmin2(pointcountmin2,3)=rowcounter2; %y final
velocitygridmin2(pointcountmin2,4)=colcounter2; %x final
velocitygridmin2(pointcountmin2,5)=round(colcounter+(colcounter2-colcounter)/2); %
x midpoint
velocitygridmin2(pointcountmin2,6)=round(rowcounter+(rowcounter2-
rowcounter)/2); %y midpoint
```

```
velocitygridmin2(pointcountmin2,7)=(rowcounter2-rowcounter); %y displacement,
^pixels
velocitygridmin2(pointcountmin2,8)=(colcounter2-colcounter);%x  displacement,
^pixels
velocitygridmin2(pointcountmin2,9)=sqrt(velocitygridmin2(pointcountmin2,7)^2+veloci
tygridmin2(pointcountmin2,8)^2); %magnitude of velocity (^pixels)
velocitygridmin2(pointcountmin2,10)=
velocitygridmin2(pointcountmin2,9)/deltime*10^9/1000/spatialres;%magnitude of
velocity (m/s)
velocitygridmin2(pointcountmin2,11)=velocitygridmin2(pointcountmin2,8)/deltime*10^
9/1000/spatialres;%x velocity (m/s)
velocitygridmin2(pointcountmin2,12)=velocitygridmin2(pointcountmin2,7)/deltime*10^
9/1000/spatialres;%y velocity (m/s)
end %if (colcounter2-colcounter)>0
end %if points2(rowcounter2,colcounter2)==1
end %for colcounter2=colcounter-leftedgecaution:1:colcounter+rightedgecaution
end %for rowcounter2=rowcounter-topedgecaution:1:rowcounter+bottomedgecaution
end %if points1(rowcounter,colcounter)==1
end %for colcounter=(leftedgecaution+1):1:file1size(2)-(rightedgecaution+1)
end %for rowcounter=(topedgecaution+1):1:file1size(1)-(bottomedgecaution+1)
disp('   Filtering for multiple vectors...');
for iteration=1:1:pointcountmin2-1
if
velocitygridmin2(iteration,1)==velocitygridmin2(iteration+1,1)&&velocitygridmin2(iter
ation,2)==velocitygridmin2(iteration+1,2)
if abs(velocitygridmin2(iteration,7))<=abs(velocitygridmin2(iteration+1,7)) %take
nearest
velocitygridmin2(iteration+1,:)=velocitygridmin2(iteration,:);
velocitygridmin2(iteration,:)=[1  1 1 1 1 1 0 0 -1000 -1000 -1000 -1000];
end
if abs(velocitygridmin2(iteration,7))>abs(velocitygridmin2(iteration+1,7)) %take nearest
velocitygridmin2(iteration,:)=[1  1 1 1 1 1 0 0 -1000 -1000 -1000 -1000];
end
end%if
velocitygridmax(iteration,1)==velocitygridmax(iteration+1,1)&&velocitygridmax(iterati
on,2)==velocitygridmax(iteration+1,2)
end %for iteration=1:1:pointcountmax-1
velocitymapmin2 =zeros(file1size); %overall magnitudes
velocitymapmin2=velocitymapmin2-1000;
for iterateplot=1:1:pointcountmin2
velocitymapmin2(velocitygridmin2(iterateplot,6),
velocitygridmin2(iterateplot,5))=velocitygridmin2(iterateplot,10);
end
se = strel('octagon',strelsize);
```

```
velocitymapdilatemin2=imdilate(velocitymapmin2,se);
velocitymapsizemin2=size(velocitymapdilatemin2);
velocitymapXmin2 =zeros(file1size); %X velocities
velocitymapXmin2=velocitymapXmin2-1000;
for iterateplot=1:1:pointcountmin2
velocitymapXmin2(velocitygridmin2(iterateplot,6),
velocitygridmin2(iterateplot,5))=velocitygridmin2(iterateplot,11);
end
se = strel('octagon',strelsize);
velocitymapdilateXmin2=imdilate(velocitymapXmin2,se);
velocitymapsizeXmin2=size(velocitymapdilateXmin2);
velocitymapYmin2 =zeros(file1size); %Y velocities
velocitymapYmin2=velocitymapYmin2-1000;
for iterateplot=1:1:pointcountmin2
velocitymapYmin2(velocitygridmin2(iterateplot,6),
velocitygridmin2(iterateplot,5))=abs(velocitygridmin2(iterateplot,12));
end
se = strel('octagon',strelsize);
velocitymapdilateYmin2=imdilate(velocitymapYmin2,se);
velocitymapsizeYmin2=size(velocitymapdilateYmin2);
velocitymapYmin2raw =zeros(file1size); %Y velocities
velocitymapYmin2raw=velocitymapYmin2raw-1000;
for iterateplot=1:1:pointcountmin2
velocitymapYmin2raw(velocitygridmin2(iterateplot,6),
velocitygridmin2(iterateplot,5))=(velocitygridmin2(iterateplot,12));
end
se = strel('octagon',strelsize);
velocitymapdilateYmin2raw=imdilate(velocitymapYmin2raw,se);
velocitymapsizeYmin2raw=size(velocitymapdilateYmin2raw);
if savefiles==1
dlmwrite(strcat(writefile1,num2str(piccount),'_Velocitymapdilatemin2_',num2str(initialt
ime),'_',num2str(delaytime),'.txt'),velocitymapdilatemin2,'\t');
dlmwrite(strcat(writefile1,num2str(piccount),'_Velocitygridmin2_',num2str(initialtime),'
_',num2str(delaytime),'.txt'),velocitygridmin2,'\t');
end

%plot the corresponding points found
figure(6)
subplot(2,2,1)
imagesc(0.4*points1dilatemax+points2dilatemax)
axis image, colormap 'gray',colorbar, caxis([0 1])
% title('Point correlations max')
figure(5)
```

```
subplot(4,5,1)
load('MyColormaps','mycmap4')
imagesc(velocitymapdilatemax) %show block velocities
size(velocitymapdilatemax)
axis image, colormap 'jet',colorbar,caxis([-1000,1000])
hold on
caxis([-1000,1000])
for iterateplot=1:1:pointcountmax
plot([velocitygridmax(iterateplot,2)
velocitygridmax(iterateplot,4)],[velocitygridmax(iterateplot,1)
velocitygridmax(iterateplot,3)]])
hold on
end
hold off
title('Velocity magnitude (m/s) and vector streamlines')
subplot(4,5,2)
load('MyColormaps','mycmap4')
imagesc(velocitymapdilateXmax) %show block velocities
axis image, colormap 'jet',colorbar,caxis([-1000,1000])
title('X velocities (m/s) max')
subplot(4,5,3)
load('MyColormaps','mycmap4')
imagesc(velocitymapdilateYmax) %show block velocities
axis image, colormap 'jet',colorbar,caxis([-1000,1000])
title('Y velocities (m/s) max mag')
subplot(4,5,4)
load('MyColormaps','mycmap4')
imagesc(velocitymapdilateYmaxraw) %show block velocities
axis image, colormap 'jet',colorbar,caxis([-1000,1000])
title('Y velocities (m/s) max raw')
subplot(4,5,5)
for iterateplot=1:1:pointcountmax
plot([velocitygridmax(iterateplot,2)
velocitygridmax(iterateplot,4)],[velocitygridmax(iterateplot,1)
velocitygridmax(iterateplot,3)]])
xlim([1,velocitymapsizemax(2)])
ylim([1,velocitymapsizemax(1)])
set(gca,'YDir','reverse')
hold on
end
axis image
xlim([1 file1size(2)]);
ylim([1 file1size(1)]);
hold off
```

```
title('Streamlines max')
figure(6)
subplot(2,2,2)
imagesc(0.4*points1dilatemin+points2dilatemin)
axis image, colormap 'gray',colorbar, caxis([0 1])
title('Point correlations min')
figure(5)
subplot(4,5,6)
load('MyColormaps','mycmap4')
imagesc(velocitymapdilatemin) %show block velocities
axis image, colormap 'jet',colorbar,caxis([-1000,1000])
hold on
for iterateplot=1:1:pointcountmin %show velocity vectors over block velocities
plot([velocitygridmin(iterateplot,2)
velocitygridmin(iterateplot,4)],[velocitygridmin(iterateplot,1)
velocitygridmin(iterateplot,3)])
hold on
end
hold off
title('Velocity magnitude (m/s) and vector streamlines')
subplot(4,5,7)
load('MyColormaps','mycmap4')
imagesc(velocitymapdilateXmin) %show block velocities
axis image, colormap 'jet',colorbar,caxis([-1000,1000])
title('X velocities (m/s) min')
subplot(4,5,8)
load('MyColormaps','mycmap4')
imagesc(velocitymapdilateYmin) %show block velocities
axis image, colormap 'jet',colorbar,caxis([-1000,1000])
title('Y velocities (m/s) min mag')
subplot(4,5,9)
load('MyColormaps','mycmap4')
imagesc(velocitymapdilateYminraw) %show block velocities
axis image, colormap 'jet',colorbar,caxis([-1000,1000])
title('Y velocities (m/s) min raw')
subplot(4,5,10)
for iterateplot=1:1:pointcountmin
plot([velocitygridmin(iterateplot,2)
velocitygridmin(iterateplot,4)],[velocitygridmin(iterateplot,1)
velocitygridmin(iterateplot,3)])
xlim([1,velocitymapsizemin(2)])
ylim([1,velocitymapsizemin(1)])
set(gca,'YDir','reverse')
hold on
```

```
end
axis image
xlim([1 file1size(2)]);
ylim([1 file1size(1)]);
hold off
title('Streamlines min')
figure(6)
subplot(2,2,3)
imagesc(0.4*points1dilatemax2+points2dilatemax2)
axis image, colormap 'gray',colorbar, caxis([0 1])
title('Point correlations max2')
figure(5)

subplot(4,5,11)
load('MyColormaps','mycmap4')
imagesc(velocitymapdilatemax2) %show block velocities
axis image, colormap 'jet',colorbar,caxis([-1000,1000])
hold on
for iterateplot=1:1:pointcountmax2 %show velocity vectors over block velocities
plot([velocitygridmax2(iterateplot,2)
velocitygridmax2(iterateplot,4)],[velocitygridmax2(iterateplot,1)
velocitygridmax2(iterateplot,3)])
hold on
end
hold off
title('Velocity magnitude (m/s) and vector streamlines')
subplot(4,5,12)
load('MyColormaps','mycmap4')
imagesc(velocitymapdilateXmax2) %show block velocities
axis image, colormap 'jet',colorbar,caxis([-1000,1000])
title('X velocities (m/s) max2')
subplot(4,5,13)
load('MyColormaps','mycmap4')
imagesc(velocitymapdilateYmax2) %show block velocities
axis image, colormap 'jet',colorbar,caxis([-1000,1000])
title('Y velocities (m/s) max2 mag')
subplot(4,5,14)
load('MyColormaps','mycmap4')
imagesc(velocitymapdilateYmax2raw) %show block velocities
axis image, colormap 'jet',colorbar,caxis([-1000,1000])
title('Y velocities (m/s) max2 raw ')
subplot(4,5,15)
for iterateplot=1:1:pointcountmax2
```

```
plot([velocitygridmax2(iterateplot,2)
velocitygridmax2(iterateplot,4)],[velocitygridmax2(iterateplot,1)
velocitygridmax2(iterateplot,3)])
xlim([1,velocitymapsizemax2(2)])
ylim([1,velocitymapsizemax2(1)])
set(gca,'YDir','reverse')
hold on
end
axis image
xlim([1 file1size(2)]);
ylim([1 file1size(1)]);
hold off
title('Streamlines max2')
figure(6)
subplot(2,2,4)
imagesc(0.4*points1dilatemin2+points2dilatemin2)
axis image, colormap 'gray',colorbar, caxis([0 1])
title('Point correlations min2')
figure(5)
subplot(4,5,16)
load('MyColormaps','mycmap4')
imagesc(velocitymapdilatemin2) %show block velocities
axis image, colormap 'jet',colorbar,caxis([-1000,1000])
hold on
for iterateplot=1:1:pointcountmin2 %show velocity vectors over block velocities
plot([velocitygridmin2(iterateplot,2)
velocitygridmin2(iterateplot,4)],[velocitygridmin2(iterateplot,1)
velocitygridmin2(iterateplot,3)])
hold on
end
hold off
title('Velocity magnitude (m/s) and vector streamlines')
subplot(4,5,17)
load('MyColormaps','mycmap4')
imagesc(velocitymapdilateXmin2) %show block velocities
axis image, colormap 'jet',colorbar,caxis([-1000,1000])
title('X velocities (m/s) min2')
subplot(4,5,18)
load('MyColormaps','mycmap4')
imagesc(velocitymapdilateYmin2) %show block velocities
axis image, colormap 'jet',colorbar,caxis([-1000,1000])

title('Y velocities (m/s) min2 mag')
```

```
subplot(4,5,19)
load('MyColormaps','mycmap4')
imagesc(velocitymapdilateYmin2raw) %show block velocities
axis image, colormap 'jet',colorbar,caxis([-1000,1000])
title('Y velocities (m/s) min2 raw')
subplot(4,5,20)
for iterateplot=1:1:pointcountmin2
plot([velocitygridmin2(iterateplot,2)
velocitygridmin2(iterateplot,4)],[velocitygridmin2(iterateplot,1)
velocitygridmin2(iterateplot,3)])
xlim([1,velocitymapsizemin2(2)])
ylim([1,velocitymapsizemin2(1)])
set(gca,'YDir','reverse')
hold on
end
axis image
xlim([1 file1size(2)]);
ylim([1 file1size(1)]);
hold off
title('Streamlines min2')
set(5,'Colormap',mycmap4)
set(6,'Colormap',mycmap4)

disp('Adding velocities')        %add velocity maps from max, min, max2, and min2

velocitytotalmap=zeros(file1size);
for rowcounter=1:1:file1size(1)
for colcounter=1:1:file1size(2)
if velocitymapmax(rowcounter,colcounter)>-1000 &&
velocitymapmin(rowcounter,colcounter)==-1000
velocitytotalmap(rowcounter,colcounter)=velocitymapmax(rowcounter,colcounter);
end
if velocitymapmax(rowcounter,colcounter)==-1000 &&
velocitymapmin(rowcounter,colcounter)>-1000
velocitytotalmap(rowcounter,colcounter)=velocitymapmin(rowcounter,colcounter);
end
if velocitymapmax(rowcounter,colcounter)==-1000 &&
velocitymapmin(rowcounter,colcounter)==-1000
velocitytotalmap(rowcounter,colcounter)=-1000;
end
if velocitymapmax(rowcounter,colcounter)>-1000 &&
velocitymapmin(rowcounter,colcounter)>-1000
velocitytotalmap(rowcounter,colcounter)=(velocitymapmax(rowcounter,colcounter)+vel
ocitymapmin(rowcounter,colcounter))/2;
```

```
end
end %for colcounter=1:1:file1size(2)
end %for rowcounter=1:1:file1size(1)

velocitytotalmap2=zeros(file1size);
for rowcounter=1:1:file1size(1)
for colcounter=1:1:file1size(2)
if velocitymapmax2(rowcounter,colcounter)>-1000 &&
velocitymapmin2(rowcounter,colcounter)==-1000
velocitytotalmap2(rowcounter,colcounter)=velocitymapmax2(rowcounter,colcounter);
end
if velocitymapmax2(rowcounter,colcounter)==-1000 &&
velocitymapmin2(rowcounter,colcounter)>-1000
velocitytotalmap2(rowcounter,colcounter)=velocitymapmin2(rowcounter,colcounter);
end
if velocitymapmax2(rowcounter,colcounter)==-1000 &&
velocitymapmin2(rowcounter,colcounter)==-1000
velocitytotalmap2(rowcounter,colcounter)=-1000;
end
if velocitymapmax2(rowcounter,colcounter)>-1000 &&
velocitymapmin2(rowcounter,colcounter)>-1000
velocitytotalmap2(rowcounter,colcounter)=(velocitymapmax2(rowcounter,colcounter)+
velocitymapmin2(rowcounter,colcounter))/2;
end
end %for colcounter=1:1:file1size(2)
end %for rowcounter=1:1:file1size(1)

for rowcounter=1:1:file1size(1)
for colcounter=1:1:file1size(2)
if velocitytotalmap2(rowcounter,colcounter)>-1000 &&
velocitytotalmap(rowcounter,colcounter)==-1000
velocitytotalmap(rowcounter,colcounter)=velocitytotalmap2(rowcounter,colcounter);
end
if velocitytotalmap2(rowcounter,colcounter)==-1000 &&
velocitytotalmap(rowcounter,colcounter)>-1000
velocitytotalmap(rowcounter,colcounter)=velocitytotalmap(rowcounter,colcounter);
end
if velocitytotalmap2(rowcounter,colcounter)>-1000 &&
velocitytotalmap(rowcounter,colcounter)>-1000
velocitytotalmap(rowcounter,colcounter)=(velocitytotalmap2(rowcounter,colcounter)+v
elocitytotalmap(rowcounter,colcounter))/2;
end
end %for colcounter=1:1:file1size(2)
end %for rowcounter=1:1:file1size(1)
```

```
figure(23) %plot total velocity map
se = strel('octagon',strelsize*4/9);
velocitytotalmapdilate=imdilate(velocitytotalmap,se);
load('MyColormaps','mycmap4')
imagesc(velocitytotalmapdilate) %show block velocities
axis image, colormap 'jet',colorbar,caxis([-1000,1000])
set(23,'Colormap',mycmap4)
hold on
for iterateplot=1:1:pointcountmin %show velocity vectors over block velocities
plot([velocitygridmin(iterateplot,2)
velocitygridmin(iterateplot,4)],[velocitygridmin(iterateplot,1)
velocitygridmin(iterateplot,3)])
hold on
end
hold on
for iterateplot=1:1:pointcountmin2 %show velocity vectors over block velocities
plot([velocitygridmin2(iterateplot,2)
velocitygridmin2(iterateplot,4)],[velocitygridmin2(iterateplot,1)
velocitygridmin2(iterateplot,3)])
hold on
end
hold on
for iterateplot=1:1:pointcountmax2 %show velocity vectors over block velocities
plot([velocitygridmax2(iterateplot,2)
velocitygridmax2(iterateplot,4)],[velocitygridmax2(iterateplot,1)
velocitygridmax2(iterateplot,3)])
hold on
end
hold on
for iterateplot=1:1:pointcountmax %show velocity vectors over block velocities
plot([velocitygridmax(iterateplot,2)
velocitygridmax(iterateplot,4)],[velocitygridmax(iterateplot,1)
velocitygridmax(iterateplot,3)])
hold on
end
hold off

title('TOTAL Velocity magnitude (m/s) and vector streamlines')
if savefiles==1
dlmwrite(strcat(writefile1,num2str(piccount),'_Velocitymaptotal_',num2str(initialtime),'
_',num2str(delaytime),'.txt'),velocitytotalmap,'\t');
dlmwrite(strcat(writefile1,num2str(piccount),'_VelocitymaptotalDilated_',num2str(initial
time),'_',num2str(delaytime),'.txt'),velocitytotalmapdilate,'\t');
```

```
end
disp('Adding velocities')
velocitytotalmapY=zeros(file1size); %add velocity maps from max and min for X and Y
directions
for rowcounter=1:1:file1size(1)
for colcounter=1:1:file1size(2)
if velocitymapYmax(rowcounter,colcounter)>-1000 &&
velocitymapYmin(rowcounter,colcounter)==-1000
velocitytotalmapY(rowcounter,colcounter)=velocitymapYmax(rowcounter,colcounter);
end
if velocitymapYmax(rowcounter,colcounter)==-1000 &&
velocitymapYmin(rowcounter,colcounter)>-1000
velocitytotalmapY(rowcounter,colcounter)=velocitymapYmin(rowcounter,colcounter);
end
if velocitymapYmax(rowcounter,colcounter)==-1000 &&
velocitymapYmin(rowcounter,colcounter)==-1000
velocitytotalmapY(rowcounter,colcounter)=-1000;
end
if velocitymapYmax(rowcounter,colcounter)>-1000 &&
velocitymapYmin(rowcounter,colcounter)>-1000
velocitytotalmapY(rowcounter,colcounter)=(velocitymapYmax(rowcounter,colcounter)+
velocitymapYmin(rowcounter,colcounter))/2;
end
end %for colcounter=1:1:file1size(2)
end %for rowcounter=1:1:file1size(1)

velocitytotalmapY2=zeros(file1size); %add velocity maps from max and min for X and
Y directions
for rowcounter=1:1:file1size(1)
for colcounter=1:1:file1size(2)
if velocitymapYmax2(rowcounter,colcounter)>-1000 &&
velocitymapYmin2(rowcounter,colcounter)==-1000
velocitytotalmapY2(rowcounter,colcounter)=velocitymapYmax2(rowcounter,colcounter
);
end
if velocitymapYmax2(rowcounter,colcounter)==-1000 &&
velocitymapYmin2(rowcounter,colcounter)>-1000
velocitytotalmapY2(rowcounter,colcounter)=velocitymapYmin2(rowcounter,colcounter)
;
end
if velocitymapYmax2(rowcounter,colcounter)==-1000 &&
velocitymapYmin2(rowcounter,colcounter)==-1000
velocitytotalmapY2(rowcounter,colcounter)=-1000;
end
```

```
if velocitymapYmax2(rowcounter,colcounter)>-1000 &&
velocitymapYmin2(rowcounter,colcounter)>-1000
velocitytotalmapY2(rowcounter,colcounter)=(velocitymapYmax2(rowcounter,colcounte
r)+velocitymapYmin2(rowcounter,colcounter))/2;
end
end %for colcounter=1:1:file1size(2)
end %for rowcounter=1:1:file1size(1)

for rowcounter=1:1:file1size(1)
for colcounter=1:1:file1size(2)
if velocitytotalmapY2(rowcounter,colcounter)>-1000 &&
velocitytotalmapY(rowcounter,colcounter)==-1000
velocitytotalmapY(rowcounter,colcounter)=velocitytotalmapY2(rowcounter,colcounter);
end
if velocitytotalmapY2(rowcounter,colcounter)==-1000 &&
velocitytotalmapY(rowcounter,colcounter)>-1000
velocitytotalmapY(rowcounter,colcounter)=velocitytotalmapY(rowcounter,colcounter);
end
if velocitytotalmapY2(rowcounter,colcounter)>-1000 &&
velocitytotalmapY(rowcounter,colcounter)>-1000
velocitytotalmapY(rowcounter,colcounter)=(velocitytotalmapY2(rowcounter,colcounter)
+velocitytotalmapY(rowcounter,colcounter))/2;
end
end %for colcounter=1:1:file1size(2)
end %for rowcounter=1:1:file1size(1)

disp('Adding velocities')
velocitytotalmapYraw=zeros(file1size); %add velocity maps from max and min for X
and Y directions
for rowcounter=1:1:file1size(1)
for colcounter=1:1:file1size(2)
if velocitymapYmaxraw(rowcounter,colcounter)>-1000 &&
velocitymapYminraw(rowcounter,colcounter)==-1000
velocitytotalmapYraw(rowcounter,colcounter)=velocitymapYmaxraw(rowcounter,colco
unter);
end
if velocitymapYmaxraw(rowcounter,colcounter)==-1000 &&
velocitymapYminraw(rowcounter,colcounter)>-1000
velocitytotalmapYraw(rowcounter,colcounter)=velocitymapYminraw(rowcounter,colco
unter);
end
if velocitymapYmaxraw(rowcounter,colcounter)==-1000 &&
velocitymapYminraw(rowcounter,colcounter)==-1000
velocitytotalmapYraw(rowcounter,colcounter)=-1000;
```

```
end
if velocitymapYmaxraw(rowcounter,colcounter)>-1000 &&
velocitymapYminraw(rowcounter,colcounter)>-1000
velocitytotalmapYraw(rowcounter,colcounter)=(velocitymapYmaxraw(rowcounter,colc
ounter)+velocitymapYminraw(rowcounter,colcounter))/2;
end
end %for colcounter=1:1:file1size(2)
end %for rowcounter=1:1:file1size(1)

velocitytotalmapY2raw=zeros(file1size); %add velocity maps from max and min for X
and Y directions
for rowcounter=1:1:file1size(1)
for colcounter=1:1:file1size(2)
if velocitymapYmax2raw(rowcounter,colcounter)>-1000 &&
velocitymapYmin2raw(rowcounter,colcounter)==-1000
velocitytotalmapY2raw(rowcounter,colcounter)=velocitymapYmax2raw(rowcounter,col
counter);
end
if velocitymapYmax2raw(rowcounter,colcounter)==-1000 &&
velocitymapYmin2raw(rowcounter,colcounter)>-1000
velocitytotalmapY2raw(rowcounter,colcounter)=velocitymapYmin2raw(rowcounter,col
counter);
end
if velocitymapYmax2raw(rowcounter,colcounter)==-1000 &&
velocitymapYmin2raw(rowcounter,colcounter)==-1000
velocitytotalmapY2raw(rowcounter,colcounter)=-1000;
end
if velocitymapYmax2raw(rowcounter,colcounter)>-1000 &&
velocitymapYmin2raw(rowcounter,colcounter)>-1000
velocitytotalmapY2raw(rowcounter,colcounter)=(velocitymapYmax2raw(rowcounter,co
lcounter)+velocitymapYmin2raw(rowcounter,colcounter))/2;
end
end %for colcounter=1:1:file1size(2)
end %for rowcounter=1:1:file1size(1)

for rowcounter=1:1:file1size(1)
for colcounter=1:1:file1size(2)
if velocitytotalmapY2raw(rowcounter,colcounter)>-1000 &&
velocitytotalmapYraw(rowcounter,colcounter)==-1000
velocitytotalmapYraw(rowcounter,colcounter)=velocitytotalmapY2raw(rowcounter,colc
ounter);
end
if velocitytotalmapY2raw(rowcounter,colcounter)==-1000 &&
velocitytotalmapYraw(rowcounter,colcounter)>-1000
velocitytotalmapYraw(rowcounter,colcounter)=velocitytotalmapYraw(rowcounter,colco
unter);
end
if velocitytotalmapY2raw(rowcounter,colcounter)>-1000 &&
velocitytotalmapYraw(rowcounter,colcounter)>-1000
velocitytotalmapYraw(rowcounter,colcounter)=(velocitytotalmapY2raw(rowcounter,col
counter)+velocitytotalmapYraw(rowcounter,colcounter))/2;
end
end %for colcounter=1:1:file1size(2)
end %for rowcounter=1:1:file1size(1)

disp('Adding velocities')
velocitytotalmapX=zeros(file1size); %add velocity maps from max and min for X and Y
directions
for rowcounter=1:1:file1size(1)
for colcounter=1:1:file1size(2)
if velocitymapXmax(rowcounter,colcounter)>-1000 &&
velocitymapXmin(rowcounter,colcounter)==-1000
velocitytotalmapX(rowcounter,colcounter)=velocitymapXmax(rowcounter,colcounter);
end
if velocitymapXmax(rowcounter,colcounter)==-1000 &&
velocitymapXmin(rowcounter,colcounter)>-1000
velocitytotalmapX(rowcounter,colcounter)=velocitymapXmin(rowcounter,colcounter);
end
if velocitymapXmax(rowcounter,colcounter)==-1000 &&
velocitymapXmin(rowcounter,colcounter)==-1000
velocitytotalmapX(rowcounter,colcounter)=-1000;
end
if velocitymapXmax(rowcounter,colcounter)>-1000 &&
velocitymapXmin(rowcounter,colcounter)>-1000
velocitytotalmapX(rowcounter,colcounter)=(velocitymapXmax(rowcounter,colcounter)+
velocitymapXmin(rowcounter,colcounter))/2;
end
end %for colcounter=1:1:file1size(2)
end %for rowcounter=1:1:file1size(1)

velocitytotalmapX2=zeros(file1size); %add velocity maps from max and min for X and
Y directions
for rowcounter=1:1:file1size(1)
for colcounter=1:1:file1size(2)
if velocitymapXmax2(rowcounter,colcounter)>-1000 &&
velocitymapXmin2(rowcounter,colcounter)==-1000
velocitytotalmapX2(rowcounter,colcounter)=velocitymapXmax2(rowcounter,colcounter
);
```

```
end
if velocitymapXmax2(rowcounter,colcounter)==-1000 &&
velocitymapXmin2(rowcounter,colcounter)>-1000
velocitytotalmapX2(rowcounter,colcounter)=velocitymapXmin2(rowcounter,colcounter)
;
end
if velocitymapXmax2(rowcounter,colcounter)==-1000 &&
velocitymapXmin2(rowcounter,colcounter)==-1000
velocitytotalmapX2(rowcounter,colcounter)=-1000;
end
if velocitymapXmax2(rowcounter,colcounter)>-1000 &&
velocitymapXmin2(rowcounter,colcounter)>-1000
velocitytotalmapX2(rowcounter,colcounter)=(velocitymapXmax2(rowcounter,colcounte
r)+velocitymapXmin2(rowcounter,colcounter))/2;
end
end %for colcounter=1:1:file1size(2)
end %for rowcounter=1:1:file1size(1)

for rowcounter=1:1:file1size(1)
for colcounter=1:1:file1size(2)
if velocitytotalmapX2(rowcounter,colcounter)>-1000 &&
velocitytotalmapX(rowcounter,colcounter)==-1000
velocitytotalmapX(rowcounter,colcounter)=velocitytotalmapX2(rowcounter,colcounter);
end
if velocitytotalmapX2(rowcounter,colcounter)==-1000 &&
velocitytotalmapX(rowcounter,colcounter)>-1000
velocitytotalmapX(rowcounter,colcounter)=velocitytotalmapX(rowcounter,colcounter);
end
if velocitytotalmapX2(rowcounter,colcounter)>-1000 &&
velocitytotalmapX(rowcounter,colcounter)>-1000
velocitytotalmapX(rowcounter,colcounter)=(velocitytotalmapX2(rowcounter,colcounter)
+velocitytotalmapX(rowcounter,colcounter))/2;
end
end %for colcounter=1:1:file1size(2)
end %for rowcounter=1:1:file1size(1)

figure(24) %plot total X and Y velocities
subplot(3,1,1)
velocitytotalmapXdilate=imdilate(velocitytotalmapX,se);
load('MyColormaps','mycmap4')
imagesc(velocitytotalmapXdilate) %show block velocities
axis image, colormap 'jet',colorbar,caxis([-1000,1000])
set(24,'Colormap',mycmap4)
title('TOTAL X velocities (m/s) ')
```

```
subplot(3,1,2)
velocitytotalmapYdilate=imdilate(velocitytotalmapY,se);
load('MyColormaps','mycmap4')
imagesc(velocitytotalmapYdilate) %show block velocities
axis image, colormap 'jet',colorbar,caxis([-1000,1000])
set(24,'Colormap',mycmap4)
title('TOTAL Y velocities (m/s) mag ')

subplot(3,1,3)
velocitytotalmapYdilateraw=imdilate(velocitytotalmapYraw,se);
load('MyColormaps','mycmap4')
imagesc(velocitytotalmapYdilateraw) %show block velocities
axis image, colormap 'jet',colorbar,caxis([-1000,1000])
set(24,'Colormap',mycmap4)
title('TOTAL Y velocities (m/s) raw ')

if savefiles==1

dlmwrite(strcat(writefile1,num2str(piccount),'_VelocitymaptotalX_',num2str(initialtime)
,'_',num2str(delaytime),'.txt'),velocitytotalmapX,'\t');
dlmwrite(strcat(writefile1,num2str(piccount),'_VelocitymaptotalDilatedX_',num2str(initi
altime),'_',num2str(delaytime),'.txt'),velocitytotalmapXdilate,'\t');
dlmwrite(strcat(writefile1,num2str(piccount),'_VelocitymaptotalYmag_',num2str(initialt
ime),'_',num2str(delaytime),'.txt'),velocitytotalmapY,'\t');
dlmwrite(strcat(writefile1,num2str(piccount),'_VelocitymaptotalDilatedYmag_',num2str
(initialtime),'_',num2str(delaytime),'.txt'),velocitytotalmapYdilate,'\t');
dlmwrite(strcat(writefile1,num2str(piccount),'_VelocitymaptotalYraw_',num2str(initialti
me),'_',num2str(delaytime),'.txt'),velocitytotalmapYraw,'\t');
dlmwrite(strcat(writefile1,num2str(piccount),'_VelocitymaptotalDilatedYraw_',num2str(
initialtime),'_',num2str(delaytime),'.txt'),velocitytotalmapYdilateraw,'\t');
end
figure(25) %plot total streamlines
for iterateplot=1:1:pointcountmin
plot([velocitygridmin(iterateplot,2)
velocitygridmin(iterateplot,4)],[velocitygridmin(iterateplot,1)
velocitygridmin(iterateplot,3)])
xlim([1,velocitymapsizemin(2)])
ylim([1,velocitymapsizemin(1)])
set(gca,'YDir','reverse')
hold on
end
for iterateplot=1:1:pointcountmax
```

```
plot([velocitygridmax(iterateplot,2)
velocitygridmax(iterateplot,4)],[velocitygridmax(iterateplot,1)
velocitygridmax(iterateplot,3)])
xlim([1,velocitymapsizemax(2)])
ylim([1,velocitymapsizemax(1)])
set(gca,'YDir','reverse')
hold on
end
for iterateplot=1:1:pointcountmin2
plot([velocitygridmin2(iterateplot,2)
velocitygridmin2(iterateplot,4)],[velocitygridmin2(iterateplot,1)
velocitygridmin2(iterateplot,3)])
xlim([1,velocitymapsizemin(2)])
ylim([1,velocitymapsizemin(1)])
set(gca,'YDir','reverse')
hold on
end
for iterateplot=1:1:pointcountmax2
plot([velocitygridmax2(iterateplot,2)
velocitygridmax2(iterateplot,4)],[velocitygridmax2(iterateplot,1)
velocitygridmax2(iterateplot,3)])
xlim([1,velocitymapsizemax(2)])
ylim([1,velocitymapsizemax(1)])
set(gca,'YDir','reverse')
hold on
end
axis image
xlim([1 file1size(2)]);
ylim([1 file1size(1)]);
hold off
title('TOTAL STREAMLINES')

if savefiles==1 %save figures
hgsave(1,strcat(writefile1,num2str(piccount),'_1_',num2str(initialtime),'_',num2str(delay
time),'.fig'));
hgsave(2,strcat(writefile1,num2str(piccount),'_2_',num2str(initialtime),'_',num2str(delay
time),'.fig'));
hgsave(3,strcat(writefile1,num2str(piccount),'_3_',num2str(initialtime),'_',num2str(delay
time),'.fig'));
hgsave(4,strcat(writefile1,num2str(piccount),'_4_',num2str(initialtime),'_',num2str(delay
time),'.fig'));
hgsave(5,strcat(writefile1,num2str(piccount),'_5_',num2str(initialtime),'_',num2str(delay
time),'.fig'));
```

```
hgsave(6,strcat(writefile1,num2str(piccount),'_6_',num2str(initialtime),'_',num2str(delaytime),'.fig'));
hgsave(23,strcat(writefile1,num2str(piccount),'_23_',num2str(initialtime),'_',num2str(delaytime),'.fig'));
hgsave(24,strcat(writefile1,num2str(piccount),'_24_',num2str(initialtime),'_',num2str(delaytime),'.fig'));
hgsave(25,strcat(writefile1,num2str(piccount),'_25_',num2str(initialtime),'_',num2str(delaytime),'.fig'));
saveas(25,strcat(writefile1,num2str(piccount),'_25_',num2str(initialtime),'_',num2str(delaytime),'.tif'), 'tiffn');
end
end %for piccount=picbegin:1:picend
```

APPENDIX D

PEAK FINDING FUNCTION

```
function [maxtab, mintab]=peakdet(v, delta, x)
%PEAKDET Detect peaks in a vector
% Eli Billauer, 3.4.05 (Explicitly not copyrighted).
% This function is released to the public domain; Any use is allowed.

maxtab = [];
mintab = [];

v = v(:); % Just in case this wasn't a proper vector

if nargin < 3
  x = (1:length(v))';
else
  x = x(:);
  if length(v)~= length(x)
    error('Input vectors v and x must have same length');
  end
end

if (length(delta(:)))>1
  error('Input argument DELTA must be a scalar');
end

if delta <= 0
  error('Input argument DELTA must be positive');
end

mn = Inf; mx = -Inf;
mnpos = NaN; mxpos = NaN;
lookformax = 1;
for i=1:length(v)
  this = v(i);
  if this > mx, mx = this; mxpos = x(i); end
  if this < mn, mn = this; mnpos = x(i); end

  if lookformax
    if this < mx-delta
      maxtab = [maxtab ; mxpos mx];
      mn = this; mnpos = x(i);
      lookformax = 0;
    end
  else
    if this > mn+delta
      mintab = [mintab ; mnpos mn];
      mx = this; mxpos = x(i);
      lookformax = 1;
    end
  end
end
```

APPENDIX E

TWO-COMPONENT VELOCIMETRY IMAGE OVERLAYING CODE AND FUNCTIONS

```
clear();
close();
%settings--------------------------------------
firstfile='BCfocus';
secondfile='Dfocus';
TEST=0; %1 for yes, 0 for no
writefile1='C:\Documents and Settings\Andrea\Desktop\ALTERED\20090201\focus and
warp\';
getfile1='C:\Documents and Settings\Andrea\Desktop\ALTERED\20090201\focus and
warp\';
savefiles=0; %save figures
plotgraphs=1;
gaussian1_1=10;
gaussian2_1=10;
gaussiankernel_1=20;
gaussianfilterinitial = fspecial('gaussian', [gaussian1_1,
gaussian2_1],gaussiankernel_1); %smooth images right after reading
%peakdet settings
%smoothing parameters
gaussian1_2=40;
gaussian2_2=10;
gaussiankernel_2=50;
gaussianfilterX_1 = fspecial('gaussian', [gaussian1_2, gaussian2_2],gaussiankernel_2);
gaussian1_3=10;
gaussian2_3=40;
gaussiankernel_3=50;
gaussianfilterY_1 = fspecial('gaussian', [gaussian1_3, gaussian2_3],gaussiankernel_3);
gaussian1_4=40;
gaussian2_4=10;
gaussiankernel_4=50;
gaussianfilterX_2 = fspecial('gaussian', [gaussian1_4, gaussian2_4],gaussiankernel_4);

gaussian1_5=40;
gaussian2_5=10;
gaussiankernel_5=50;
gaussianfilterY_2 = fspecial('gaussian', [gaussian1_5, gaussian2_5],gaussiankernel_5);
```

```
gradientXthreshold_1=0.0001;
gradientYthreshold_1=0.0001;
gradientXthreshold_2=0.0001;
gradientYthreshold_2=0.0001;

%end settings---------------------------------
configarray=[gaussian1_1 gaussian2_1 gaussiankernel_1  ...
gaussian1_2 gaussian2_2 gaussiankernel_2 ...
gaussian1_3 gaussian2_3 gaussiankernel_3 gaussian1_4 gaussian2_4 gaussiankernel_4
gaussian1_5 ...
gaussian2_5 gaussiankernel_5 gradientXthreshold_1 gradientYthreshold_1
gradientXthreshold_2 ...
gradientYthreshold_2 ];
configarray=configarray';
dlmwrite(strcat(writefile1,'Config_',firstfile,'_',secondfile,'.txt'),configarray,'newline','pc')
;
filename1=strcat(getfile1,firstfile,'.tif'); %READ FIRST IMAGE
file1=imread(filename1);
file1=im2double(im2uint16(file1));
if plotgraphs==1
figure(1)
subplot(3,3,1)
imagesc(file1)
axis image, colormap 'gray';
title(firstfile)
set(1,'Name',filename1)
end
%-------begin warping
file1=WARPBC(file1); %point to WARPBC function
if plotgraphs==1
figure(8)
imagesc(file1)
title('warped BC')
axis image, colormap 'gray';
end
%-----end warping
file1=filter2(gaussianfilterinitial,file1); %smooth
[gradientX,gradientY]=gradient(file1); %take gradients
file1=gradientX+gradientY;
if plotgraphs==1
figure(1)
subplot(3,3,2)
imagesc(file1)
axis image, colormap 'gray';
```

```
title('GradientX + GradientY')
end
file1size=size(file1);
if plotgraphs==1
figure(1)
subplot(3,3,3)
imagesc(file1)
axis image, colormap 'gray';
title('gradientX+gradientY')
figure(1)
subplot(3,3,4)
imagesc(gradientX)
axis image, colormap 'gray';
title('gradientX')
figure(1)
subplot(3,3,7)
imagesc(gradientY)
axis image, colormap 'gray';
title('gradientY')
end
gradientX=filter2(gaussianfilterX_1,gradientX);                %smooth gradients
gradientY=filter2(gaussianfilterY_1,gradientY);
if plotgraphs==1
figure(1)
subplot(3,3,5)
imagesc(gradientX)
axis image, colormap 'gray';
title('gradientX smoothed')
figure(1)
subplot(3,3,8)
imagesc(gradientY)
axis image, colormap 'gray';
title('gradientY smoothed')
end
positionsmatrixXmax=zeros(file1size); %find peaks using function peakdet.m (vertical
grid)
positionsmatrixXmin=zeros(file1size);
for rowcounter=1:1:file1size(1) %find peaks using function peakdet.m (vertical grid)
testslice=gradientX(rowcounter,:);
gradientXmax=zeros();
gradientXmin=zeros();
[gradientXmax, gradientXmin]=peakdet(testslice,gradientXthreshold_1);
if rowcounter==10
if plotgraphs==1
```

```
figure(1)
subplot(3,3,6)
plot(testslice)
xlim([1 max(size(testslice))]);          hold on ;
plot(gradientXmax(:,1),gradientXmax(:,2),'*g');       hold on;
plot(gradientXmin(:,1),gradientXmin(:,2),'*r');        hold off;
end
end %  if rowcounter==round(file1size(1)/2)
if max(size(gradientXmax)) >0 && max(size(gradientXmin))>0%if peaks are found
gradientXmax=gradientXmax(:,1);
gradientXmin=gradientXmin(:,1);
for count=1:1:max(size(gradientXmax))
positionsmatrixXmax(rowcounter,gradientXmax(count))=1;
end
end %if max(size(gradientXmax)) >0
end %for rowcounter=1:1:file1size(1)
positionsmatrixYmax=zeros(file1size); %find peaks using function peakdet.m
(horizontal grid)
positionsmatrixYmin=zeros(file1size); %peaks matrix for horizontal stripes
for colcounter=1:1:file1size(2)
testslice=gradientY(:,colcounter);
gradientYmax=zeros();
gradientYmin=zeros();
[gradientYmax,gradientYmin]=peakdet(testslice,gradientYthreshold_1);
if colcounter==round(file1size(2)/2)
if plotgraphs==1
figure(1)
subplot(3,3,9)
plot(testslice)
xlim([1 max(size(testslice))])
hold on
plot(gradientYmax(:,1),gradientYmax(:,2),'*g'); hold on
plot(gradientYmin(:,1),gradientYmin(:,2),'*r'); hold on
end
end %if colcounter==round(file1size(2)/2)
if max(size(gradientYmax)) >0 && max(size(gradientYmin))>0
gradientYmax=gradientYmax(:,1);
gradientYmin=gradientYmin(:,1);
for count=1:1:max(size(gradientYmax))
positionsmatrixYmax(gradientYmax(count),colcounter)=1;
end
end %if max(size(gradientYmax)) >0
end %for rowcounter=1:1:file1size(1)
```

```
totalintersectionmax=positionsmatrixXmax+positionsmatrixYmax; %%locations of intersecting points have values = 2
se = strel('diamond',2);
if plotgraphs==1
figure(2)
positionsmatrixXdilatemax = imdilate(positionsmatrixXmax,se);
subplot (1,2,1)
imagesc(positionsmatrixXdilatemax)
axis image,colormap 'gray'
title('BCdotcard')
positionsmatrixYdilatemax = imdilate(positionsmatrixYmax,se);
subplot(1,2,2)
imagesc(positionsmatrixYdilatemax)
axis image, colormap 'gray'
end
intersectionmapmax=zeros(file1size);
for rowcounter=1:1:file1size(1)
for colcounter=1:1:file1size(2)
if totalintersectionmax(rowcounter,colcounter)>=1
intersectionmapmax(rowcounter,colcounter)=1;
end
end
end
if plotgraphs==1
figure(3)
intersectionmapdilatemax = imdilate(intersectionmapmax,se);
imagesc(intersectionmapdilatemax)
axis image, colormap 'gray', title('BC dotcard')
end
intersectionmapmaxBC=zeros();
intersectionmapmaxBC=intersectionmapmax; %%<=====BC dotcard

%---------------------------read second image
filename1=strcat(getfile1,secondfile,'.tif'); %READ FIRST IMAGE
file1=imread(filename1);
file1=im2double(im2uint16(file1));
if plotgraphs==1
figure(4)
subplot(3,3,1)
imagesc(file1)
axis image, colormap 'gray';
title(firstfile)
set(4,'Name',filename1)
end
```

```
%-------begin warping
file1=WARPD(file1);
if plotgraphs==1
figure(9)
imagesc(file1)
title('warped D')
axis image, colormap 'gray';
end
%-----end warping
file1=filter2(gaussianfilterinitial,file1); %smooth
[gradientX,gradientY]=gradient(file1); %take gradients
file1=gradientX+gradientY;
if plotgraphs==1
figure(4)
subplot(3,3,2)
imagesc(file1)
axis image, colormap 'gray';
title('GradientX + GradientY')
end
file1size=size(file1);
if plotgraphs==1
figure(4)
subplot(3,3,3)
imagesc(file1)
axis image, colormap 'gray';
title('gradientX+gradientY')
figure(4)
subplot(3,3,4)
imagesc(gradientX)
axis image, colormap 'gray';
title('gradientX')
figure(4)
subplot(3,3,7)
imagesc(gradientY)
axis image, colormap 'gray';
title('gradientY')
end
gradientX=filter2(gaussianfilterX_1,gradientX);                %smooth gradients
gradientY=filter2(gaussianfilterY_1,gradientY);
if plotgraphs==1
figure(4)
subplot(3,3,5)
imagesc(gradientX)
axis image, colormap 'gray';
```

```
title('gradientX smoothed')
figure(4)
subplot(3,3,8)
imagesc(gradientY)
axis image, colormap 'gray';
title('gradientY smoothed')
end
positionsmatrixXmax=zeros(file1size); %find peaks using function peakdet.m (vertical
grid)
positionsmatrixXmin=zeros(file1size);
for rowcounter=1:1:file1size(1) %find peaks using function peakdet.m (vertical grid)
testslice=gradientX(rowcounter,:);
gradientXmax=zeros();
gradientXmin=zeros();
[gradientXmax, gradientXmin]=peakdet(testslice,gradientXthreshold_1);
if rowcounter==round(file1size(1)/2)
if plotgraphs==1
figure(4)
subplot(3,3,6)
plot(testslice)
xlim([1 max(size(testslice))]);          hold on ;
plot(gradientXmax(:,1),gradientXmax(:,2),'*g');       hold on;
plot(gradientXmin(:,1),gradientXmin(:,2),'*r');        hold off;
end
end %  if rowcounter==round(file1size(1)/2)
if max(size(gradientXmax)) >0 && max(size(gradientXmin))>0%if peaks are found
gradientXmax=gradientXmax(:,1);
gradientXmin=gradientXmin(:,1);
for count=1:1:max(size(gradientXmax))
positionsmatrixXmax(rowcounter,gradientXmax(count))=1;
end
end %if max(size(gradientXmax)) >0
end %for rowcounter=1:1:file1size(1)
positionsmatrixYmax=zeros(file1size); %find peaks using function peakdet.m
(horizontal grid)
positionsmatrixYmin=zeros(file1size); %peaks matrix for horizontal stripes
for colcounter=1:1:file1size(2)
testslice=gradientY(:,colcounter);
gradientYmax=zeros();
gradientYmin=zeros();
[gradientYmax,gradientYmin]=peakdet(testslice,gradientYthreshold_1);
if colcounter==round(file1size(2)/2)
if plotgraphs==1
figure(4)
```

```
subplot(3,3,9)
plot(testslice)
xlim([1 max(size(testslice))])
hold on
plot(gradientYmax(:,1),gradientYmax(:,2),'*g'); hold on
plot(gradientYmin(:,1),gradientYmin(:,2),'*r'); hold on
end
end %if colcounter==round(file1size(2)/2)
if max(size(gradientYmax)) >0 && max(size(gradientYmin))>0
gradientYmax=gradientYmax(:,1);
gradientYmin=gradientYmin(:,1);
for count=1:1:max(size(gradientYmax))
positionsmatrixYmax(gradientYmax(count),colcounter)=1;
end
end %if max(size(gradientYmax)) >0
end %for rowcounter=1:1:file1size(1)
totalintersectionmax=positionsmatrixXmax+positionsmatrixYmax; %%locations of
intersecting points have values = 2
se = strel('diamond',2);
if plotgraphs==1
figure(5)
positionsmatrixXdilatemax = imdilate(positionsmatrixXmax,se);
subplot (1,2,1)
imagesc(positionsmatrixXdilatemax)
axis image,colormap 'gray'
title('Ddotcard')
positionsmatrixYdilatemax = imdilate(positionsmatrixYmax,se);
subplot(1,2,2)
imagesc(positionsmatrixYdilatemax)
axis image, colormap 'gray'
end
intersectionmapmax=zeros(file1size);
for rowcounter=1:1:file1size(1)
for colcounter=1:1:file1size(2)
if totalintersectionmax(rowcounter,colcounter)>=1
intersectionmapmax(rowcounter,colcounter)=1;
end
end
end
if plotgraphs==1
figure(6)
intersectionmapdilatemax = imdilate(intersectionmapmax,se);
imagesc(intersectionmapdilatemax)
axis image, colormap 'gray',title('Ddotcard')
```

```
end
intersectionmapmaxD=zeros();
intersectionmapmaxD=intersectionmapmax; %%<=====D dotcard
intersectionmapmaxtotal=zeros();
intersectionmapmaxtotal=2*intersectionmapmaxD+intersectionmapmaxBC;
figure(7)
intersectionmapmaxdilatetotal = imdilate(intersectionmapmaxtotal,se);
imagesc(intersectionmapmaxdilatetotal)
axis image, colormap 'jet',colorbar,caxis ([0 2])
title('overlap, D=red, BC=green');
if savefiles==1
hgsave(1,strcat(writefile1,'1.fig'));
hgsave(2,strcat(writefile1,'2.fig'));
hgsave(3,strcat(writefile1,'3.fig'));
hgsave(4,strcat(writefile1,'4.fig'));
hgsave(5,strcat(writefile1,'5.fig'));
hgsave(6,strcat(writefile1,'6.fig'));
hgsave(7,strcat(writefile1,'7.fig'));
hgsave(8,strcat(writefile1,'8.fig'));
hgsave(9,strcat(writefile1,'9.fig'));
saveas(3,strcat(writefile1,'3.tif'), 'tiffn');
saveas(6,strcat(writefile1,'6.tif'), 'tiffn');
dlmwrite(strcat(writefile1,'intersectionmapmaxdilatetotal.txt'),intersectionmapmaxdilatetotal,'newline','pc');
end
dlmwrite(strcat(writefile1,'intersectionmapmaxtotal.txt'),intersectionmapmaxtotal,'newline','pc');
=============================================================
function file1=WARPBC(file1)
stretchfactoroverall=1; %for superpixel resolution
A1=size(file1);
A1=A1*stretchfactoroverall;
rect=[1,1,round(stretchfactoroverall*1023),round(stretchfactoroverall*1023)]; %[1,1,254,254] will give a matrix which is 255 by 255 (includes the first pixel)
stretchfactor=stretchfactoroverall*1.005; %used only in the stretchimage code
file1=imresize(file1,stretchfactor);
%TRANSLATE
xform=[1,0,0;0,1,0;stretchfactoroverall*24,stretchfactoroverall*0,1];
%xform=[1,0,0;0,1,0;0,8,1](moves image 8 pixels down),xform=[1,0,0;0,1,0;6,0,1](moves image 6 pixels to right)
tform_translate=maketform('affine',xform);
[file1_trans xdata ydata]= imtransform(file1, tform_translate);
```

```
file1_trans = imtransform(file1,
tform_translate,'XData',[1,(size(file1,2)+xform(3,1))],'YData',
[1,(size(file1,1)+xform(3,2))]);
file1=file1_trans;
rectcrop=[0,0,A1(1),A1(2)];
file1=imcrop(file1,rectcrop);
%ROTATE
file1 = imrotate(file1,0,'bilinear','crop');
%SHEAR
shearangle=0; % <=======ADJUST
shearform=[1,0,0;shearangle,1,0;0,0,1]; %shear in x and y direction (19.5 line)
tform_shear=maketform('affine',shearform);
[file1_shear xdata ydata]= imtransform(file1, tform_shear);
file1=file1_shear;
%CROP
file1=imcrop(file1,rect);%crop images to region of interest
==================================================================
function file1=WARPD(file1)
stretchfactoroverall=1;
A1=size(file1);
A1=A1*stretchfactoroverall;
rect=[1,1,round(stretchfactoroverall*1023),round(stretchfactoroverall*1023)]; %[1,1,25
4,254] will give a matrix which is 255 by 255 (includes the first pixel)
stretchfactor=stretchfactoroverall*1.0; %used only in the stretchimage code
file1=imresize(file1,stretchfactor);
%TRANSLATE
xform=[1,0,0;0,1,0;stretchfactoroverall*0,stretchfactoroverall*42,1];
%xform=[1,0,0;0,1,0;0,8,1](moves image 8 pixels
down),xform=[1,0,0;0,1,0;6,0,1](moves image 6 pixels to right)
tform_translate=maketform('affine',xform);
[file1_trans xdata ydata]= imtransform(file1, tform_translate);
file1_trans = imtransform(file1,
tform_translate,'XData',[1,(size(file1,2)+xform(3,1))],'YData',
[1,(size(file1,1)+xform(3,2))]);
file1=file1_trans;
rectcrop=[0,0,A1(1),A1(2)];
file1=imcrop(file1,rectcrop);
%ROTATE
file1 = imrotate(file1,00.5,'bilinear','crop');
%SHEAR
shearangle=0.00; % <=======ADJUST
shearform=[1,0,0;shearangle,1,0;0,0,1]; %shear in x and y direction (19.5 line)
tform_shear=maketform('affine',shearform);
[file1_shear xdata ydata]= imtransform(file1, tform_shear);
```

```
file1=file1_shear;
%CROP
file1=imcrop(file1,rect);%crop images to region of interest
```

APPENDIX F

COMPUTER-BASED IMAGE ANALYSIS STUDY OF

PLIF FLUORESCENCE SIGNAL PROCESSING CODE

```
function PICTUREPROCESS()
close() %Begin warping pictures
clear()
global Z_D Z_BC Z_BCsmoothed Z_Dsmoothed Zdotcard_D Zdotcard_BC Z_Dmean
Z_BCmean Z_Dmeansmoothed Z_BCmeansmoothed Z_Dbackground
Z_BCbackground Z_Dbackgroundsmoothed Z_BCbackgroundsmoothed Z_Dsubtracted
Z_BCsubtracted
global Z_Ddivided Z_BCdivided
global parameterlist squaredimension horzres vertres hhorznum hvertnum Dbkgname
BCbkgname
global saveroot
parameterlist=zeros(34,1); % parameters to write tofile
f = figure('Visible','off','resize','off','Position',[360,1,1400,680]);
set(f,'Name','Stage1: Warp dot card')
movegui(f,'northwest')
warplabelorder=uicontrol('Style','text','String','***TL,BL,BR,TR',...
'Position',[735,650,90,13]);
hsavefigure = uicontrol('Style','pushbutton','String','(4)SAVE=>2','ForegroundColor','r',...
'Position',[865,670,77,30],'max',5,'min',0,'Callback',{@savefigurebutton_Callback});

hsavefiguretext= uicontrol('Style','edit','String','C:\Documents and
Settings\Andrea\Desktop\Altered\20080707\RUN1\TEMPPROCESSED3\',...
'Position',[950,642,260,58],'max',3,'min',1);
saveroot=get(hsavefiguretext,'String');
hmesh= uicontrol('Style','edit','String','C:\Documents and
Settings\Andrea\Desktop\Altered\20080707\Dfocus.tif','Position',[30,642,260,58],'max',
3,'min',1);
hWARP = uicontrol('Style','pushbutton','String','(2)WARP!!','Position',[790,670,70,30],...
'Callback',{@WARPbutton_Callback});
hplotpoints = uicontrol('Style','pushbutton','String','(1)Get
Files','Position',[590,670,70,30],'max',5,'min',0,'Callback',{@plotpointsbutton_Callback}
);
hcalciteration = uicontrol('Style','text','String','Iterations',...
'Position',[735,687,50,13]);
hxcoord1=uicontrol('Style','edit','String','266','Position',[665,687,30,13]);
hxcoord2=uicontrol('Style','edit','String','264','Position',[665,672,30,13]);
```

```
hxcoord3=uicontrol('Style','edit','String','350','Position',[665,657,30,13]);
hxcoord4=uicontrol('Style','edit','String','354','Position',[665,642,30,13]);
hycoord1=uicontrol('Style','edit','String','198','Position',[700,687,30,13]);
hycoord2=uicontrol('Style','edit','String','290','Position',[700,672,30,13]);
hycoord3=uicontrol('Style','edit','String','292','Position',[700,657,30,13]);
hycoord4=uicontrol('Style','edit','String','202','Position',[700,642,30,13]);
htestfile= uicontrol('Style','edit','String','C:\Documents and
Settings\Andrea\Desktop\Altered\20080707\RUN1\D_RUN1_
(100).tif','Position',[320,642,260,58],'max',3,'min',1);
plot1 = axes('Units','Pixels','Position',[30,375,260,260]);
plot2 = axes('Units','Pixels','Position',[320,375,260,260]);
plot3 = axes('Units','Pixels','Position',[640,375,260,260]);
plot4 = axes('Units','Pixels','Position',[930,375,335,260]);
hhorzlabel = uicontrol('Style','text','String','Horz(inch)','Position',[950,336,100,14]);
hvertlabel = uicontrol('Style','text','String','Vert(inch)','Position',[950,321,100,14]);
hhorzreslabel = uicontrol('Style','text','String','Horz
Res(mm/pixel)','Position',[950,306,100,14]);
hvertreslabel = uicontrol('Style','text','String','Vert
Res(mm/pixel)','Position',[950,291,100,14]);
hhorz = uicontrol('Style','edit','String','0.2','Position',[1055,336,60,14]);
hvert = uicontrol('Style','edit','String','0.4','Position',[1055,321,60,14]);
hhorzres = uicontrol('Style','text','String','n/a','Position',[1055,306,60,14]);
hvertres = uicontrol('Style','text','String','n/a','Position',[1055,291,60,14]);
hresbutton=
uicontrol('Style','pushbutton','String','(3)Resolution','Position',[1120,320,90,30],...
'Callback',{ @resbutton_Callback});
hmesh2= uicontrol('Style','edit','String','C:\Documents and
Settings\Andrea\Desktop\Altered\20080707\BCfocus.tif',...
'Position',[30,292,260,58],'max',3,'min',1);
hWARP2 =
uicontrol('Style','pushbutton','String','(2)WARP!!','Position',[790,320,70,30],...
'Callback',{ @WARPbutton_Callback2});
hplotpoints2 = uicontrol('Style','pushbutton','String','(1)Get
Files','Position',[590,320,70,30],'max',5,'min',0,'Callback',{ @plotpointsbutton_Callback2
});
hcalciteration2 = uicontrol('Style','text','String','Iterations',...
'Position',[735,337,50,13]);
hxcoord12=uicontrol('Style','edit','String','238','Position',[665,337,30,13]);
hxcoord22=uicontrol('Style','edit','String','240','Position',[665,322,30,13]);
hxcoord32=uicontrol('Style','edit','String','332','Position',[665,307,30,13]);
hxcoord42=uicontrol('Style','edit','String','328','Position',[665,292,30,13]);
hycoord12=uicontrol('Style','edit','String','234','Position',[700,337,30,13]);
hycoord22=uicontrol('Style','edit','String','326','Position',[700,322,30,13]);
hycoord32=uicontrol('Style','edit','String','324','Position',[700,307,30,13]);
```

```
hycoord42=uicontrol('Style','edit','String','232','Position',[700,292,30,13]);
htestfile2= uicontrol('Style','edit','String','C:\Documents and
Settings\Andrea\Desktop\Altered\20080707\RUN1\BC_RUN1_ (100).tif',...
'Position',[320,292,260,58],'max',3,'min',1);
plot5 = axes('Units','Pixels','Position',[30,25,260,260]);
plot6 = axes('Units','Pixels','Position',[320,25,260,260]);
plot7 = axes('Units','Pixels','Position',[640,25,260,260]);
plot8 = axes('Units','Pixels','Position',[930,25,335,260]);
set(f,'Visible','on');  %turn the figure on so it is visible -------------------------------Begin
program
function plotpointsbutton_Callback(source,eventdata)
for testiteration=1:1:2 %run twice, once for warp dot card, other for test image
if testiteration==1
Warpfilename=get(hmesh,'String');
else
Warpfilename=get(htestfile,'String');
end
TOWARP=imread(Warpfilename);
TOWARP=im2double(im2uint16(TOWARP));
squaredimension=size(TOWARP);
squaredimension=squaredimension(1);
if testiteration==1
axes(plot1) %send this to plot 1 axes
axis([1 squaredimension 1 squaredimension 0 1 0 1])
axis tight;
axis image;
else
axes(plot2)
end
imagesc(TOWARP),colormap 'gray'
axis image
hold on
xcoord1=str2double(get(hxcoord1,'String'));
xcoord2=str2double(get(hxcoord2,'String'));
xcoord3=str2double(get(hxcoord3,'String'));
xcoord4=str2double(get(hxcoord4,'String'));
ycoord1=str2double(get(hycoord1,'String'));
ycoord2=str2double(get(hycoord2,'String'));
ycoord3=str2double(get(hycoord3,'String'));
ycoord4=str2double(get(hycoord4,'String'));
plot(xcoord1,ycoord1,'x','MarkerSize',15,'MarkerEdgeColor','r')
plot(xcoord2,ycoord2,'x','MarkerSize',15,'MarkerEdgeColor','r')
plot(xcoord3,ycoord3,'x','MarkerSize',15,'MarkerEdgeColor','r')
plot(xcoord4,ycoord4,'x','MarkerSize',15,'MarkerEdgeColor','r')
```

```
hold off
xiterations=max(horzcat(xcoord1,xcoord2,xcoord3,xcoord4))-
min(horzcat(xcoord1,xcoord2,xcoord3,xcoord4));
yiterations=max(horzcat(ycoord1,ycoord2,ycoord3,ycoord4))-
min(horzcat(ycoord1,ycoord2,ycoord3,ycoord4));
iterations=xiterations*yiterations;
set(hcalciteration,'String',num2str(iterations));
end %run twice, once for warp dot card, other for test image
end %plotpointsbutton_Callback(source,eventdata)
function WARPbutton_Callback(source,eventdata)
for testiteration=1:1:2 %run twice, once for warp dot card, other for test image
if testiteration==1
Warpfilename=get(hmesh,'String');
else
Warpfilename=get(htestfile,'String');
end
TOWARP=imread(Warpfilename);
TOWARP=im2double(im2uint16(TOWARP));
xcoord1=str2double(get(hxcoord1,'String'));
xcoord2=str2double(get(hxcoord2,'String'));
xcoord3=str2double(get(hxcoord3,'String'));
xcoord4=str2double(get(hxcoord4,'String'));
ycoord1=str2double(get(hycoord1,'String'));
ycoord2=str2double(get(hycoord2,'String'));
ycoord3=str2double(get(hycoord3,'String'));
ycoord4=str2double(get(hycoord4,'String'));
X=[xcoord1;xcoord2;xcoord3;xcoord4];
Y=[ycoord1;ycoord2;ycoord3;ycoord4];
Xp=[0;0;squaredimension;squaredimension];
Yp=[0;squaredimension;squaredimension;0];
B = [ X Y ones(size(X)) zeros(4,3) -X.*Xp -Y.*Xp ...
zeros(4,3) X Y ones(size(X)) -X.*Yp -Y.*Yp ];
B = reshape(B',8,8)';
D = [Xp,Yp];
D = reshape(D',8,1);
l = inv(B'*B)*B'*D;
A = reshape([l(1:6)' 0 0 1 ],3,3)';
C = [l(7:8)' 1];
Secondsignalx=zeros();
Secondsignaly=zeros();
Secondsignalz=zeros();
counter=0;
for x=floor(min(X)):1:ceil(max(X))
for y=floor(min(Y)):1:ceil(max(Y))
```

```
counter=counter+1;
if rem(counter,5000)==0
counter
end
t=A*[x;y;1]/(C*[x;y;1]);
Secondsignalx(counter)=t(1);
Secondsignaly(counter)=t(2);
Secondsignalz(counter)=TOWARP(y,x);
end
end
xlin=linspace(1,squaredimension,squaredimension);
ylin=linspace(1,squaredimension,squaredimension);
[X,Y]=meshgrid(xlin,ylin);
if testiteration==1
Zdotcard_D=griddata(Secondsignalx,Secondsignaly,Secondsignalz,X,Y,'cubic');
axes(plot3)
mesh(X,Y,Zdotcard_D,'FaceColor','interp','EdgeColor','none')
axis([1 squaredimension 1 squaredimension 0 1 0 1])
axis tight;
view(2)
set(gca,'YDir','reverse')
colormap 'gray'
else
Z_D=griddata(Secondsignalx,Secondsignaly,Secondsignalz,X,Y,'cubic');
axes(plot4)
mesh(X,Y,Z_D,'FaceColor','interp','EdgeColor','none')
axis([1 squaredimension 1 squaredimension 0 1 0 1])
view(2)
set(gca,'YDir','reverse')
colormap 'gray'
colorbar('location','eastoutside');
axis tight;
axis image;
end
end %run twice, once for warp dot card, other for test image
end %WARPbutton_Callback(source,eventdata)
function plotpointsbutton_Callback2(source,eventdata)
for testiteration=1:1:2 %run twice, once for warp dot card, other for test image
if testiteration==1
Warpfilename=get(hmesh2,'String');
else
Warpfilename=get(htestfile2,'String');
end
TOWARP=imread(Warpfilename);
```

```
TOWARP=im2double(im2uint16(TOWARP));
squaredimension=size(TOWARP);
squaredimension=squaredimension(1);
if testiteration==1
axes(plot5) %send this to plot 1 axes
axis([1 squaredimension 1 squaredimension 0 1 0 1])
axis tight;
axis image;
else
axes(plot6)
end
imagesc(TOWARP),colormap 'gray'
axis image
hold on
xcoord12=str2double(get(hxcoord12,'String'));
xcoord22=str2double(get(hxcoord22,'String'));
xcoord32=str2double(get(hxcoord32,'String'));
xcoord42=str2double(get(hxcoord42,'String'));
ycoord12=str2double(get(hycoord12,'String'));
ycoord22=str2double(get(hycoord22,'String'));
ycoord32=str2double(get(hycoord32,'String'));
ycoord42=str2double(get(hycoord42,'String'));
plot(xcoord12,ycoord12,'x','MarkerSize',15,'MarkerEdgeColor','r')
plot(xcoord22,ycoord22,'x','MarkerSize',15,'MarkerEdgeColor','r')
plot(xcoord32,ycoord32,'x','MarkerSize',15,'MarkerEdgeColor','r')
plot(xcoord42,ycoord42,'x','MarkerSize',15,'MarkerEdgeColor','r')
hold off
xiterations2=max(horzcat(xcoord12,xcoord22,xcoord32,xcoord42))-
min(horzcat(xcoord12,xcoord22,xcoord32,xcoord42));
yiterations2=max(horzcat(ycoord12,ycoord22,ycoord32,ycoord42))-
min(horzcat(ycoord12,ycoord22,ycoord32,ycoord42));
iterations2=xiterations2*yiterations2;
set(hcalciteration2,'String',num2str(iterations2));
end %run twice, once for warp dot card, other for test image
end %plotpointsbutton_Callback(source,eventdata)
function WARPbutton_Callback2(source,eventdata)
for testiteration=1:1:2 %run twice, once for warp dot card, other for test image
if testiteration==1
Warpfilename=get(hmesh2,'String');
else
Warpfilename=get(htestfile2,'String');
end
TOWARP=imread(Warpfilename);
TOWARP=im2double(im2uint16(TOWARP));
```

```
xcoord12=str2double(get(hxcoord12,'String'));
xcoord22=str2double(get(hxcoord22,'String'));
xcoord32=str2double(get(hxcoord32,'String'));
xcoord42=str2double(get(hxcoord42,'String'));
ycoord12=str2double(get(hycoord12,'String'));
ycoord22=str2double(get(hycoord22,'String'));
ycoord32=str2double(get(hycoord32,'String'));
ycoord42=str2double(get(hycoord42,'String'));
X=[xcoord12;xcoord22;xcoord32;xcoord42];
Y=[ycoord12;ycoord22;ycoord32;ycoord42];
Xp=[0;0;squaredimension;squaredimension];
Yp=[0;squaredimension;squaredimension;0];
B = [ X Y ones(size(X)) zeros(4,3) -X.*Xp -Y.*Xp ...
zeros(4,3) X Y ones(size(X)) -X.*Yp -Y.*Yp ];
B = reshape(B',8,8)';
D = [Xp,Yp];
D = reshape(D',8,1);
l = inv(B'*B)*B'*D;
A = reshape([l(1:6)' 0 0 1 ],3,3)';
C = [l(7:8)' 1];
Secondsignalx=zeros();
Secondsignaly=zeros();
Secondsignalz=zeros();
counter=0;
for x=floor(min(X)):1:ceil(max(X))
for y=floor(min(Y)):1:ceil(max(Y))
counter=counter+1;
if rem(counter,5000)==0
counter
end
t=A*[x;y;1]/(C*[x;y;1]);
Secondsignalx(counter)=t(1);
Secondsignaly(counter)=t(2);
Secondsignalz(counter)=TOWARP(y,x);
end
end
xlin=linspace(1,squaredimension,squaredimension);
ylin=linspace(1,squaredimension,squaredimension);
[X,Y]=meshgrid(xlin,ylin);
if testiteration==1
Zdotcard_BC=griddata(Secondsignalx,Secondsignaly,Secondsignalz,X,Y,'cubic');
axes(plot7)
mesh(X,Y,Zdotcard_BC,'FaceColor','interp','EdgeColor','none')
axis([1 squaredimension 1 squaredimension 0 1 0 1])
```

```
axis tight;
view(2)
set(gca,'YDir','reverse')
colormap 'gray'
else
Z_BC=griddata(Secondsignalx,Secondsignaly,Secondsignalz,X,Y,'cubic');
axes(plot8)
mesh(X,Y,Z_BC,'FaceColor','interp','EdgeColor','none')
axis([1 squaredimension 1 squaredimension 0 1 0 1])
axis tight;
view(2)
set(gca,'YDir','reverse')
colormap 'gray'
colorbar('location','eastoutside');
axis tight;
axis image;
end
end %run twice, once for warp dot card, other for test image
end %WARPbutton_Callback2(source,eventdata)
function resbutton_Callback(source,eventdata) %calculate resolution
hhorznum=str2num(get(hhorz,'String'))
hvertnum=str2num(get(hvert,'String'))
horzres=((hhorznum)*25.4)/squaredimension(1);
vertres=((hvertnum)*25.4)/squaredimension(1);
set(hhorzres,'String',num2str(horzres));
set(hvertres,'String',num2str(vertres));
end %resbutton_Callback(source,eventdata)
function savefigurebutton_Callback(source,eventdata)
saveroot=get(hsavefiguretext,'String');
dlmwrite(strcat(saveroot,'Z_DStage1.txt'),Z_D,'newline','pc');
dlmwrite(strcat(saveroot,'Z_BCStage1.txt'),Z_BC,'newline','pc');
dlmwrite(strcat(saveroot,'ZDotcard_D.txt'),Zdotcard_D,'newline','pc');
dlmwrite(strcat(saveroot,'ZDotcard_BC.txt'),Zdotcard_BC,'newline','pc');
hgsave(1,strcat(saveroot,'Stage1.fig')); %save figure
parameterlist(1)=str2num(get(hxcoord1,'String'));
parameterlist(2)=str2num(get(hxcoord2,'String'));
parameterlist(3)=str2num(get(hxcoord3,'String'));
parameterlist(4)=str2num(get(hxcoord4,'String'));
parameterlist(5)=str2num(get(hycoord1,'String'));
parameterlist(6)=str2num(get(hycoord2,'String'));
parameterlist(7)=str2num(get(hycoord3,'String'));
parameterlist(8)=str2num(get(hycoord4,'String'));
parameterlist(9)=str2num(get(hxcoord12,'String'));
parameterlist(10)=str2num(get(hxcoord22,'String'));
parameterlist(11)=str2num(get(hxcoord32,'String'));
parameterlist(12)=str2num(get(hxcoord42,'String'));
parameterlist(13)=str2num(get(hycoord12,'String'));
parameterlist(14)=str2num(get(hycoord22,'String'));
parameterlist(15)=str2num(get(hycoord32,'String'));
parameterlist(16)=str2num(get(hycoord42,'String'));
dlmwrite(strcat(saveroot,'Parameterlist.txt'),parameterlist,'newline','pc');
stage2() %GOTO stage2
end %savefigurebutton_Callback(source,eventdata)
%==========================================================
=============function stage2()
close(1)
f = figure('Visible','off','resize','off','Position',[360,1,1400,680]);
set(f,'Name','Stage2: Background subtraction')
movegui(f,'northwest')
flatfieldlabel=uicontrol('Style','text','String','Flatfield Y/N','Position',[10,690,80,15]);
flatfieldyn=uicontrol('Style','edit','String','Y','Position',[95,690,30,15]);
imagesubtractynlabel=uicontrol('Style','text','String','Image
Y/N','Position',[10,670,80,15]);
imagesubtractyn=uicontrol('Style','text','String','N','Position',[95,670,30,15]);
Dfilenamelabel=uicontrol('Style','text','String','Dbkg
filename','Position',[130,690,80,15]);
Dfilename=uicontrol('Style','edit','String','C:\Documents and
Settings\Andrea\Desktop\ALTERED\20080707\Dfocus.tif','Position',[215,690,300,15]);

BCfilenamelabel=uicontrol('Style','text','String','BCbkg
filename','Position',[130,670,80,15]);
BCfilename=uicontrol('Style','edit','String','C:\Documents and
Settings\Andrea\Desktop\ALTERED\20080707\BCfocus.tif','Position',[215,670,300,15])
;
flatfieldvaluelabel=uicontrol('Style','text','String','Flatfield
value','Position',[130,650,80,15]);
flatfieldvalue=uicontrol('Style','edit','String',num2str(0.007629),'Position',[215,650,100,1
5]); %500 counts
hUPDATEbutton =
uicontrol('Style','pushbutton','String','(1)UPDATE','ForegroundColor','r',...
'Position',[520,673,65,30],'max',5,'min',0,'Callback',{@UPDATEimages_Callback});

hgetBKGbutton =
uicontrol('Style','pushbutton','String','(2)GetBKG','ForegroundColor','r',...
'Position',[590,673,65,30],'max',5,'min',0,'Callback',{@GetBKG_Callback});
hsmoothbutton =
uicontrol('Style','pushbutton','String','(3)SmthBKG','ForegroundColor','r',...
'Position',[867,673,65,30],'max',5,'min',0,'Callback',{@Smoothbkg_Callback});
```

```
hsubtractbutton =
uicontrol('Style','pushbutton','String','(4)SubBKG','ForegroundColor','r',...
'Position',[867,640,65,30],'max',5,'min',0,'Callback',{@Subtractbkg_Callback});
hsavefiguretext2=
uicontrol('Style','text','String',saveroot,'Position',[1010,645,260,58],'max',3,'min',1);
hsavefigure2 = uicontrol('Style','pushbutton','String','SAVE=>3','ForegroundColor','r',...
'Position',[940,673,65,30],'max',5,'min',0,'Callback',{@savefigurebutton2_Callback});

plot1 = axes('Units','Pixels','Position',[30,335,260,260]);
plot12 = axes('Units','Pixels','Position',[30,30,260,260]);
plot2 = axes('Units','Pixels','Position',[320,335,260,260]);
plot22 = axes('Units','Pixels','Position',[320,30,260,260]);
plot3 = axes('Units','Pixels','Position',[630,335,335,260]);
plot32 = axes('Units','Pixels','Position',[630,30,335,260]);
plotlabel1=uicontrol('Style','text','String','Background','Position',[420,600,60,15]);
plotlabel2=uicontrol('Style','text','String','Bkgsubtracted','Position',[700,600,100,15]);

hsmoothkernel1= uicontrol('Style','edit','String','2','Position',[660,675,65,14]);
hsmoothkernel2= uicontrol('Style','edit','String','2','Position',[730,675,65,14]);
hsmoothkernel3= uicontrol('Style','edit','String','15','Position',[800,675,60,14]);
hsmoothkernel1label= uicontrol('Style','text','String','Vert','Position',[660,690,65,14]);
hsmoothkernel2label= uicontrol('Style','text','String','Horz','Position',[730,690,65,14]);
hsmoothkernel3label= uicontrol('Style','text','String','Sigma','Position',[800,690,60,14]);
hsmoothkernel1reslabel=
uicontrol('Style','text','String','Vertres(mm)','Position',[660,660,65,14]);
hsmoothkernel2reslabel=
uicontrol('Style','text','String','Horzres(mm)','Position',[730,660,65,14]);
hsmoothkernel1res= uicontrol('Style','text','String','n/a','Position',[660,645,65,14]);
hsmoothkernel2res= uicontrol('Style','text','String','n/a','Position',[730,645,65,14]);
set(f,'Visible','on');
function UPDATEimages_Callback(source,eventdata)
if get(flatfieldyn,'String')=='Y'
set(imagesubtractyn,'String','N');
set(Dfilename,'Style','text');
set(BCfilename,'Style','text');
set(flatfieldvalue,'Style','edit');
end %if get(flatfieldyn,'String')='Y'
if get(flatfieldyn,'String')=='N'
set(imagesubtractyn,'String','Y');
set(Dfilename,'Style','edit');
set(BCfilename,'Style','edit');
set(flatfieldvalue,'Style','text');
end %if get(flatfieldyn,'String')='N'
end %function UPDATEimages_Callback(source,eventdata)
```

```
function GetBKG_Callback(source,eventdata)
if get(flatfieldyn,'String')=='Y'
Z_D=dlmread(strcat(saveroot,'Z_DStage1.txt'));
axes(plot1)
imagesc(Z_D), colormap 'gray'
axis image
Z_BC=dlmread(strcat(saveroot,'Z_BCStage1.txt'));
axes(plot12)
imagesc(Z_BC), colormap 'gray'
axis image
sizeTemp=size(Z_D);
Z_Dbackground=zeros(sizeTemp); %plot zeros
Z_Dbackground(:,:)=str2num(get(flatfieldvalue,'String'));
Z_BCbackground=zeros(sizeTemp); %plot zeros
Z_BCbackground(:,:)=str2num(get(flatfieldvalue,'String'));
end
if get(flatfieldyn,'String')=='N'
Z_D=dlmread(strcat(saveroot,'Z_DStage1.txt'));
axes(plot1)
imagesc(Z_D), colormap 'gray'
axis image
Z_BC=dlmread(strcat(saveroot,'Z_BCStage1.txt'));
axes(plot12)
imagesc(Z_BC), colormap 'gray'
axis image
Z_Dbackground=imread(get(Dfilename,'String'));
Z_Dbackground=im2double(im2uint16(Z_Dbackground));
Z_BCbackground=imread(get(BCfilename,'String'));
Z_BCbackground=im2double(im2uint16(Z_BCbackground));
%warp
for iteration=1:1:2 %run once for D, once for BC
if iteration==1
TOWARP=Z_Dbackground;
else
TOWARP=Z_BCbackground;
%set this to image to warp
end
xcoord1=parameterlist(1);
xcoord2=parameterlist(2);
xcoord3=parameterlist(3);
xcoord4=parameterlist(4);
ycoord1=parameterlist(5);
ycoord2=parameterlist(6);
ycoord3=parameterlist(7);
```

```
ycoord4=parameterlist(8);
xcoord12=parameterlist(9);
xcoord22=parameterlist(10);
xcoord32=parameterlist(11);
xcoord42=parameterlist(12);
ycoord12=parameterlist(13);
ycoord22=parameterlist(14);
ycoord32=parameterlist(15);
ycoord42=parameterlist(16);
xiterations=max(horzcat(xcoord1,xcoord2,xcoord3,xcoord4))-
min(horzcat(xcoord1,xcoord2,xcoord3,xcoord4));
yiterations=max(horzcat(ycoord1,ycoord2,ycoord3,ycoord4))-
min(horzcat(ycoord1,ycoord2,ycoord3,ycoord4));
iterations=xiterations*yiterations;
if iteration==1
X=[xcoord1;xcoord2;xcoord3;xcoord4];
Y=[ycoord1;ycoord2;ycoord3;ycoord4];
else
X=[xcoord12;xcoord22;xcoord32;xcoord42];
Y=[ycoord12;ycoord22;ycoord32;ycoord42];
end
Xp=[0;0;squaredimension;squaredimension];
Yp=[0;squaredimension;squaredimension;0];
B = [ X Y ones(size(X)) zeros(4,3) -X.*Xp -Y.*Xp ...
        zeros(4,3) X Y ones(size(X)) -X.*Yp -Y.*Yp ];
B = reshape(B',8,8)';
D = [Xp,Yp];
D = reshape(D',8,1);
l = inv(B'*B)*B'*D;
A = reshape([l(1:6)' 0 0 1 ],3,3)';
C = [l(7:8)' 1];
Secondsignalx=zeros();
Secondsignaly=zeros();
Secondsignalz=zeros();
counter=0;
for x=floor(min(X)):1:ceil(max(X))
for y=floor(min(Y)):1:ceil(max(Y))
counter=counter+1;
if rem(counter,5000)==0
counter
end
t=A*[x;y;1]/(C*[x;y;1]);
Secondsignalx(counter)=t(1);
Secondsignaly(counter)=t(2);
```

```
Secondsignalz(counter)=TOWARP(y,x);
end
end
xlin=linspace(1,squaredimension,squaredimension);
ylin=linspace(1,squaredimension,squaredimension);
[X,Y]=meshgrid(xlin,ylin);
if iteration==1
Z_Dbackground=griddata(Secondsignalx,Secondsignaly,Secondsignalz,X,Y,'cubic');
else
Z_BCbackground=griddata(Secondsignalx,Secondsignaly,Secondsignalz,X,Y,'cubic');
end
end %iteration=1:1:2 %run once for D, once for BC
end %if get(flatfieldyn,'String')=='N'
axes(plot2)
imagesc(Z_Dbackground), colormap 'gray'
axis image
caxis(get(plot1,'clim'))
axes(plot22)
imagesc(Z_BCbackground), colormap 'gray'
axis image
caxis(get(plot12,'clim'))
end %function GetBKG_Callback(source,eventdata)
function Smoothbkg_Callback(source,eventdata)
vertressmoothedkernel=(str2num(get(hsmoothkernel1,'String')))*vertres;
set(hsmoothkernel1res,'String',num2str(vertressmoothedkernel));
horzressmoothedkernel=(str2num(get(hsmoothkernel2,'String')))*horzres;
set(hsmoothkernel2res,'String',num2str(horzressmoothedkernel));
gaussianfilter = fspecial('gaussian', [str2num(get(hsmoothkernel1,'String')),
str2num(get(hsmoothkernel2,'String'))],
str2num(get(hsmoothkernel3,'String'))); %smooth images
Z_Dbackgroundsmoothed=filter2(gaussianfilter,Z_Dbackground);
axes(plot2)
imagesc(Z_Dbackgroundsmoothed), colormap 'gray'
axis image
caxis(get(plot1,'clim'))
Z_BCbackgroundsmoothed=filter2(gaussianfilter,Z_BCbackground);
axes(plot22)
imagesc(Z_BCbackgroundsmoothed), colormap 'gray'
axis image
caxis(get(plot1,'clim'))
end %function Smoothbkg_Callback(source,eventdata)
function Subtractbkg_Callback(source,eventdata)
Z_Dsubtracted=imsubtract(Z_D,Z_Dbackgroundsmoothed);
Z_BCsubtracted=imsubtract(Z_BC,Z_BCbackgroundsmoothed);
```

```
sizeTemp=size(Z_D);
for rowcounter=1:1:sizeTemp(1) %filter for values <0
for colcounter=1:1:sizeTemp(2)
if Z_D(rowcounter,colcounter)<0
Z_D(rowcounter,colcounter)=0;
end
if Z_BC(rowcounter,colcounter)<0
Z_BC(rowcounter,colcounter)=0;
end
end %for colcounter=1:1:sizeTemp(2)
end %for rowcounter=1:1:sizeTemp(1) %filter for values <0
axes(plot3)
imagesc(Z_Dsubtracted), colormap 'gray'
colorbar('location','eastoutside');
axis tight;
axis image;
caxis(get(plot1,'clim'))
axes(plot32)
imagesc(Z_BCsubtracted), colormap 'gray'
colorbar('location','eastoutside');
axis tight;
axis image;
caxis(get(plot1,'clim'))

end %function subtractimages_Callback(source,eventdata)
function savefigurebutton2_Callback(source,eventdata)
flatfieldyntowrite=get(flatfieldyn,'String');
Dbkgname=get(Dfilename,'String');
BCbkgname=get(BCfilename,'String');
set(Dfilename,'Style','text');
if flatfieldyntowrite=='Y'
parameterlist(32)=1;
parameterlist(33)=str2num(get(flatfieldvalue,'String'));
parameterlist(34)=str2num(get(flatfieldvalue,'String'));
end
if flatfieldyntowrite=='N'
parameterlist(32)=0;
parameterlist(33)=-1000;
parameterlist(34)=-1000;
end
dlmwrite(strcat(saveroot,'Z_DStage2.txt'),Z_Dsubtracted,'newline','pc');
dlmwrite(strcat(saveroot,'Z_BCStage2.txt'),Z_BCsubtracted,'newline','pc');
dlmwrite(strcat(saveroot,'Z_Dbackgroundsmoothed.txt'),Z_Dbackgroundsmoothed,'newl
ine','pc');
```

```
dlmwrite(strcat(saveroot,'Z_BCbackgroundsmoothed.txt'),Z_BCbackgroundsmoothed,'n
ewline','pc');
dlmwrite(strcat(saveroot,'Parameterlist.txt'),parameterlist,'newline','pc');
hgsave(1,strcat(saveroot,'Stage2.fig')); %save figure
stage3()
end %function savefigurebutton_Callback(source,eventdata)
end

%============================================================
=============
function stage3() %divide by the mean
close(1)
f = figure('Visible','off','resize','off','Position',[360,1,1400,680]);
set(f,'Name','Stage3: Divide by the mean')
movegui(f,'northwest')
stage1point5_Drootlabel=uicontrol('Style','text','String','DFileroot','Position',[30,690,60,1
5]);
stage1point5_Dleaflabel=uicontrol('Style','text','String','DFileleaf','Position',[30,670,60,1
5]);
stage1point5_BCrootlabel=uicontrol('Style','text','String','BCFileroot','Position',[30,650,6
0,15]);
stage1point5_BCleaflabel=uicontrol('Style','text','String','BCFileleaf','Position',[30,630,6
0,15]);
stage1point5_Droot=uicontrol('Style','edit','String','C:\Documents and
Settings\Andrea\Desktop\Altered\20080707\RUN2\D_RUN2_
(','Position',[100,690,460,15]);
stage1point5_Dleaf=uicontrol('Style','edit','String',').tif','Position',[100,670,460,15]);
stage1point5_BCroot=uicontrol('Style','edit','String','C:\Documents and
Settings\Andrea\Desktop\Altered\20080707\RUN2\BC_RUN2_
(','Position',[100,650,460,15]);
stage1point5_BCleaf=uicontrol('Style','edit','String',').tif','Position',[100,630,460,15]);
picbeginlabel=uicontrol('Style','text','String','picbegin','Position',[580,690,50,15]);
picendlabel=uicontrol('Style','text','String','picend','Position',[580,670,50,15]);
picbegin=uicontrol('Style','edit','String','50','Position',[640,690,50,15]);
picend=uicontrol('Style','edit','String','300','Position',[640,670,50,15]);
hFindaverage = uicontrol('Style','pushbutton','String','(1)CALC
AVG','ForegroundColor','r',...
'Position',[700,675,100,30],'max',5,'min',0,'Callback',{@Findaveragebutton_Callback});

plotlabel1=uicontrol('Style','text','String','Mean','Position',[420,600,60,15]);
plotlabel2=uicontrol('Style','text','String','Divided by mean','Position',[700,600,100,15]);

plot1 = axes('Units','Pixels','Position',[30,335,260,260]);
plot4 = axes('Units','Pixels','Position',[30,30,260,260]);
```

```
plot2 = axes('Units','Pixels','Position',[320,335,260,260]);
plot5 = axes('Units','Pixels','Position',[320,30,260,260]);
plot3 = axes('Units','Pixels','Position',[630,335,335,260]);
plot6 = axes('Units','Pixels','Position',[630,30,335,260]);
hsmoothkernel1= uicontrol('Style','edit','String','2','Position',[810,675,65,14]);
hsmoothkernel2= uicontrol('Style','edit','String','2','Position',[880,675,65,14]);
hsmoothkernel3= uicontrol('Style','edit','String','15','Position',[950,675,60,14]);
hsmoothkernel1label= uicontrol('Style','text','String','Vert','Position',[810,690,65,14]);
hsmoothkernel2label= uicontrol('Style','text','String','Horz','Position',[880,690,65,14]);
hsmoothkernel3label= uicontrol('Style','text','String','Sigma','Position',[950,690,60,14]);
hsmoothkernel1reslabel=
uicontrol('Style','text','String','Vertres(mm)','Position',[810,660,65,14]);
hsmoothkernel2reslabel=
uicontrol('Style','text','String','Horzres(mm)','Position',[880,660,65,14]);
hsmoothkernel1res= uicontrol('Style','text','String','n/a','Position',[810,645,65,14]);
hsmoothkernel2res= uicontrol('Style','text','String','n/a','Position',[880,645,65,14]);
hSmoothmean = uicontrol('Style','pushbutton','String','(2)SMOOTH
MEAN&DIVIDE','ForegroundColor','r',...
'Position',[1020,675,160,30],'max',5,'min',0,'Callback',{ @smoothimagebutton_Callback}
);
hsavefigure = uicontrol('Style','pushbutton','String','(3)SAVE=>4','ForegroundColor','r',...
'Position',[1187,675,80,30],'max',5,'min',0,'Callback',{ @savefigurebutton_Callback});
set(f,'Visible','on');
Z_D=dlmread(strcat(saveroot,'Z_DStage2.txt'));
axes(plot1)
imagesc(Z_D), colormap 'gray'
axis image
Z_BC=dlmread(strcat(saveroot,'Z_BCStage2.txt'));
axes(plot4)
imagesc(Z_BC), colormap 'gray'
axis image
function Findaveragebutton_Callback(source,eventdata)
TotalID=zeros(squaredimension,squaredimension)   ;
TotalIBC=zeros(squaredimension,squaredimension);
for piciteration=Str2num(get(picbegin,'String')):1:Str2num(get(picend,'String'))
if rem(piciteration,10)==0
piciteration
end
ID=imread(strcat(get(stage1point5_Droot,'String'),num2str(piciteration),get(stage1point
5_Dleaf,'String')));
ID=im2double(im2uint16(ID));
TotalID=imadd(TotalID,ID);
IBC=imread(strcat(get(stage1point5_BCroot,'String'),num2str(piciteration),get(stage1po
int5_BCleaf,'String')));
```

```
IBC=im2double(im2uint16(IBC));
TotalIBC=imadd(TotalIBC,IBC);
end
%calculate average
TotalID=imdivide(TotalID,(Str2num(get(picend,'String'))-
Str2num(get(picbegin,'String'))+1));
TotalIBC=imdivide(TotalIBC,((Str2num(get(picend,'String'))-
Str2num(get(picbegin,'String'))+1)+1));
%warp average to match
for iteration=1:1:2 %run once for D, once for BC
if iteration==1
TOWARP=TotalID;
else
TOWARP=TotalIBC;
%set this to image to warp
end
xcoord1=parameterlist(1);
xcoord2=parameterlist(2);
xcoord3=parameterlist(3);
xcoord4=parameterlist(4);
ycoord1=parameterlist(5);
ycoord2=parameterlist(6);
ycoord3=parameterlist(7);
ycoord4=parameterlist(8);
xcoord12=parameterlist(9);
xcoord22=parameterlist(10);
xcoord32=parameterlist(11);
xcoord42=parameterlist(12);
ycoord12=parameterlist(13);
ycoord22=parameterlist(14);
ycoord32=parameterlist(15);
ycoord42=parameterlist(16);
xiterations=max(horzcat(xcoord1,xcoord2,xcoord3,xcoord4))-
min(horzcat(xcoord1,xcoord2,xcoord3,xcoord4));
yiterations=max(horzcat(ycoord1,ycoord2,ycoord3,ycoord4))-
min(horzcat(ycoord1,ycoord2,ycoord3,ycoord4));
iterations=xiterations*yiterations;
if iteration==1
X=[xcoord1;xcoord2;xcoord3;xcoord4];
Y=[ycoord1;ycoord2;ycoord3;ycoord4];
else
X=[xcoord12;xcoord22;xcoord32;xcoord42];
Y=[ycoord12;ycoord22;ycoord32;ycoord42];
end
```

```
Xp=[0;0;squaredimension;squaredimension];
Yp=[0;squaredimension;squaredimension;0];
B = [ X Y ones(size(X)) zeros(4,3) -X.*Xp -Y.*Xp ...
zeros(4,3) X Y ones(size(X)) -X.*Yp -Y.*Yp ];
B = reshape(B',8,8)';
D = [Xp,Yp];
D = reshape(D',8,1);
l = inv(B'*B)*B'*D;
A = reshape([l(1:6)' 0 0 1 ],3,3)';
C = [l(7:8)' 1];
Secondsignalx=zeros();
Secondsignaly=zeros();
Secondsignalz=zeros();
counter=0;
for x=floor(min(X)):1:ceil(max(X))
for y=floor(min(Y)):1:ceil(max(Y))
counter=counter+1;
if rem(counter,5000)==0
counter
end
t=A*[x;y;1]/(C*[x;y;1]);
Secondsignalx(counter)=t(1);
Secondsignaly(counter)=t(2);
Secondsignalz(counter)=TOWARP(y,x);
end
end
xlin=linspace(1,squaredimension,squaredimension);
ylin=linspace(1,squaredimension,squaredimension);
[X,Y]=meshgrid(xlin,ylin);
if iteration==1
Z_Dmean=griddata(Secondsignalx,Secondsignaly,Secondsignalz,X,Y,'cubic');
%smoothimages to avoid adding noise
axes(plot2)
imagesc(Z_Dmean), colormap 'gray'
axis image
caxis(get(plot1,'clim'))
else
Z_BCmean=griddata(Secondsignalx,Secondsignaly,Secondsignalz,X,Y,'cubic');
axes(plot5)
imagesc(Z_BCmean), colormap 'gray'
axis image
caxis(get(plot4,'clim'))
end
end %iteration=1:1:2 %run once for D, once for BC
end %  function Findaveragebutton_Callback(source,eventdata)
function smoothimagebutton_Callback(source,eventdata)
vertressmoothedkernel=(str2num(get(hsmoothkernel1,'String')))*vertres;
set(hsmoothkernel1res,'String',num2str(vertressmoothedkernel));
horzressmoothedkernel=(str2num(get(hsmoothkernel2,'String')))*horzres;
set(hsmoothkernel2res,'String',num2str(horzressmoothedkernel));
gaussianfilter = fspecial('gaussian', [str2num(get(hsmoothkernel1,'String')),
str2num(get(hsmoothkernel2,'String'))],
str2num(get(hsmoothkernel3,'String'))); %smooth images
Z_Dmeansmoothed=filter2(gaussianfilter,Z_Dmean);
axes(plot2)
imagesc(Z_Dmeansmoothed), colormap 'gray'
axis image
caxis(get(plot1,'clim'))
Z_BCmeansmoothed=filter2(gaussianfilter,Z_BCmean);
axes(plot5)
imagesc(Z_BCmeansmoothed), colormap 'gray'
axis image
caxis(get(plot1,'clim'))
Z_Ddivided=imdivide(Z_D,Z_Dmeansmoothed);
axes(plot3)
imagesc(Z_Ddivided), colormap 'gray'
colorbar('location','eastoutside');
axis tight;
axis image;
axis([1 squaredimension 1 squaredimension 0 1 0 2])
Z_BCdivided=imdivide(Z_BC,Z_BCmeansmoothed);
axes(plot6)
imagesc(Z_BCdivided), colormap 'gray'
colorbar('location','eastoutside');
axis tight;
axis image;
axis([1 squaredimension 1 squaredimension 0 1 0 2])

end %function smoothimagebutton_Callback(source,eventdata)
function savefigurebutton_Callback(source,eventdata)
dlmwrite(strcat(saveroot,'Z_DStage3.txt'),Z_Ddivided,'newline','pc');
dlmwrite(strcat(saveroot,'Z_BCStage3.txt'),Z_BCdivided,'newline','pc');
dlmwrite(strcat(saveroot,'Z_Dmeansmoothed.txt'),Z_Dmeansmoothed,'newline','pc');
dlmwrite(strcat(saveroot,'Z_BCmeansmoothed.txt'),Z_BCmeansmoothed,'newline','pc');
dlmwrite(strcat(saveroot,'Parameterlist.txt'),parameterlist,'newline','pc');
hgsave(1,strcat(saveroot,'Stage3.fig')); %save figure
stage4()
end %function savefigurebutton_Callback(source,eventdata)
```

```
end

%=================================================================
=======
function stage4() %Begin shearing the warped images
close(1)
f = figure('Visible','off','resize','off','Position',[360,1,1400,680]);
set(f,'Name','Stage4: Shear images for processing')
movegui(f,'northwest')
hhorzlabel2 = uicontrol('Style','text','String','Horz(inch)','Position',[30,694,100,14]);
hvertlabel2 = uicontrol('Style','text','String','Vert(inch)','Position',[30,679,100,14]);
hhorzreslabel2 = uicontrol('Style','text','String','Horz
Res(mm/pixel)','Position',[30,664,100,14]);
hvertreslabel2 = uicontrol('Style','text','String','Vert
Res(mm/pixel)','Position',[30,649,100,14]);
hhorz2 = uicontrol('Style','text','String',num2str(hhorznum),'Position',[135,694,60,14]);
hvert2 = uicontrol('Style','text','String',num2str(hvertnum),'Position',[135,679,60,14]);
hhorzres2 = uicontrol('Style','text','String',num2str(horzres),'Position',[135,664,60,14]);
hvertres2 = uicontrol('Style','text','String',num2str(vertres),'Position',[135,649,60,14]);
habsorption=uicontrol('Style','edit','String','Y','Position',[325,694,80,14]);
habsorptionlabel= uicontrol('Style','text','String','Absorption
Y/N','Position',[240,694,80,14]);
hbandingcrop= uicontrol('Style','edit','String','0','Position',[325,679,80,14]);
hbandingcroplabel=
uicontrol('Style','text','String','Bandingcrop','Position',[240,679,80,14]);
hsavefiguretext2=
uicontrol('Style','text','String',saveroot,'Position',[970,645,260,58],'max',3,'min',1);
hsavefigure2 = uicontrol('Style','pushbutton','String','SAVE=>5','ForegroundColor','r',...
'Position',[885,673,77,30],'max',5,'min',0,'Callback',{@savefigurebutton2_Callback});

hshearangle= uicontrol('Style','edit','String','0.03','Position',[515,694,80,14]);
hshearanglecaution=
uicontrol('Style','text','String','|shear|<0.1','Position',[430,694,80,14]);
% > 0.1 generates an error
hbandingparameter= uicontrol('Style','edit','String','30','Position',[515,679,80,14]);
hbandinglabel=
uicontrol('Style','text','String','DBandingtight','Position',[430,679,80,14]);
hcropcolumns= uicontrol('Style','edit','String','0','Position',[515,664,80,14]);

hcroplabel= uicontrol('Style','text','String','DColtocrop','Position',[430,664,80,14]);

hshearangle2= uicontrol('Style','edit','String','0.03','Position',[685,694,80,15]);
hshearanglecaution2=
uicontrol('Style','text','String','|shear|<0.1','Position',[600,694,80,15]);
```

```
hbandingparameter2= uicontrol('Style','edit','String','30','Position',[685,679,80,15]);

hbandinglabel2= uicontrol('Style','text','String','BCBandtight','Position',[600,679,80,15]);

hcropcolumns2=uicontrol('Style','edit','String','0','Position',[685,664,80,15]);

hcroplabel2= uicontrol('Style','text','String','BCColtocrop','Position',[600,664,80,15]);

hshearanglelabel=uicontrol('Style','pushbutton','String','DO IT!',...
'Position',[780,673,80,30],'max',3,'min',1,'Callback',{@shearbutton_Callback});
plot1 = axes('Units','Pixels','Position',[30,380,260,260]);
plot4 = axes('Units','Pixels','Position',[330,380,260,260]);
plot5 = axes('Units','Pixels','Position',[665,590,260,50]);
plot6 = axes('Units','Pixels','Position',[665,440,260,50]);
plot7 = axes('Units','Pixels','Position',[665,515,260,50]);
plot9 = axes('Units','Pixels','Position',[665,365,260,50]);
plot8 = axes('Units','Pixels','Position',[970,380,260,260]);
hshearedsize1=uicontrol('Style','text','String','n/a','Position',[595,625,30,15],'max',3,'min',
1);
hshearedsize2=uicontrol('Style','text','String','n/a','Position',[595,605,30,15],'max',3,'min',
1);
hbandcorrection=uicontrol('Style','text','String','banding','Position',[875,640,50,13],'max',
3,'min',1);
hrow=uicontrol('Style','text','String','row','Position',[875,565,50,13],'max',3,'min',1);
hcol=uicontrol('Style','text','String','col','Position',[875,490,50,13],'max',3,'min',1);
habsorp=uicontrol('Style','text','String','Absorption','Position',[875,415,50,13],'max',3,'mi
n',1);
hcorrectedsize1=uicontrol('Style','text','String','n/a','Position',[1235,625,30,15],'max',3,'m
in',1);
hcorrectedsize2=uicontrol('Style','text','String','n/a','Position',[1235,605,30,15],'max',3,'m
in',1);
plot12 = axes('Units','Pixels','Position',[30,30,260,260]);
plot42 = axes('Units','Pixels','Position',[330,30,260,260]);
plot52 = axes('Units','Pixels','Position',[665,240,260,50]);
plot62 = axes('Units','Pixels','Position',[665,90,260,50]);
plot72 = axes('Units','Pixels','Position',[665,165,260,50]);
plot92 = axes('Units','Pixels','Position',[665,15,260,50]);
plot82 = axes('Units','Pixels','Position',[970,30,260,260]);
hshearedsize12=uicontrol('Style','text','String','n/a','Position',[595,275,30,15],'max',3,'min
',1);
hshearedsize22=uicontrol('Style','text','String','n/a','Position',[595,255,30,15],'max',3,'min
',1);
hbandcorrection2=uicontrol('Style','text','String','banding','Position',[875,290,50,13],'max
',3,'min',1);
```

```
hrow2=uicontrol('Style','text','String','row','Position',[875,215,50,13],'max',3,'min',1);
hcol2=uicontrol('Style','text','String','col','Position',[875,140,50,13],'max',3,'min',1);
hcolabsorp2=uicontrol('Style','text','String','Absorption','Position',[875,65,50,13],'max',3,'
min',1);
hcorrectedsize12=uicontrol('Style','text','String','n/a','Position',[1235,273,30,15],'max',3,'
min',1);
hcorrectedsize22=uicontrol('Style','text','String','n/a','Position',[1235,255,30,15],'max',3,'
min',1);
set(f,'Visible','on');
Z_D=dlmread(strcat(saveroot,'Z_DStage3.txt'));
axes(plot1)
imagesc(Z_D), colormap 'gray'
axis image
Dsize=size(Z_D);
Dsize=Dsize(1);
Z_BC=dlmread(strcat(saveroot,'Z_BCStage3.txt'));
axes(plot12)
imagesc(Z_BC), colormap 'gray'
axis image
BCsize=size(Z_BC);
BCsize=BCsize(1);
function shearbutton_Callback(source,eventdata)
shearangle=str2num(get(hshearangle,'String'));
if abs(shearangle)>0
shearform=[1,0,0;shearangle,1,0;0,0,1];
tform_shear=maketform('affine',shearform);
[D_shear xdata ydata]= imtransform(Z_D, tform_shear);
Z_Dsheared=D_shear;
else
Z_Dsheared=Z_D;
end
edgecrop=str2num(get(hbandingcrop,'String'));
parameterlist(31)=edgecrop;
sizeTemp=size(Z_Dsheared);
if edgecrop>0
Z_Dsheared(:,sizeTemp(2)-edgecrop+1:sizeTemp(2))=[];
Z_Dsheared(:,1:edgecrop)=[];
Z_Dsheared(sizeTemp(1)-edgecrop+1:sizeTemp(1),:)=[];
Z_Dsheared(1:edgecrop,:)=[];
end
axes(plot4)
imagesc(Z_Dsheared), colormap 'gray',caxis(get(plot1,'clim'))
axis image
caxis(get(plot1,'clim'))
```

```
Dsize=size(Z_Dsheared);
set(hshearedsize1,'String',num2str(Dsize(1)));
set(hshearedsize2,'String',num2str(Dsize(2)));
D_compress=mean(Z_Dsheared); %Banding correction, average all rows
smoothingfunction=smooth(D_compress,str2num(get(hbandingparameter,'String')));
axes(plot5)
plot(smoothingfunction)
set(plot5,'Xgrid','on');
ylim([0.7 1.3]);

sheetcorrectionfactor=zeros(1,Dsize(2));
for sheetcorrectioncol=1:1:Dsize(2)
sheetcorrectionfactor(sheetcorrectioncol)=max(smoothingfunction)./smoothingfunction(
sheetcorrectioncol);
end
for sheetcorrectionrow=1:1:Dsize(1)
for sheetcorrectioncol=1:1:Dsize(2)
Z_Dsheared(sheetcorrectionrow,sheetcorrectioncol)=Z_Dsheared(sheetcorrectionrow,sh
eetcorrectioncol)*sheetcorrectionfactor(sheetcorrectioncol);
end
end
if get(habsorption,'String')=='Y'
Dtopaverage=mean(mean(Z_Dsheared(2,50:Dsize(2)-50)));  %absorption correction
%give 50 pixels leeway on each side for shearing.
Dbottomaverage=mean(mean(Z_Dsheared(Dsize(1)-1,50:Dsize(2)-50)));
Dtopaverage=mean(mean(Z_Dsheared(2,50:Dsize(2)-50)));
Dbottomaverage=mean(mean(Z_Dsheared(Dsize(1)-1,50:Dsize(2)-50)));
Dabsorptionscalematrixpre=[Dtopaverage:(Dbottomaverage-Dtopaverage)/(Dsize(1)-
1):Dbottomaverage]; %create the linear fit
Dabsorptionscalematrixpre=Dabsorptionscalematrixpre/max(Dabsorptionscalematrixpre
);
size(Dabsorptionscalematrixpre);
for columncounter=1:1:Dsize(2)
for rowcounter=1:1:Dsize(1)
Z_Dsheared(rowcounter,columncounter)=Z_Dsheared(rowcounter,columncounter)/Dabs
orptionscalematrixpre(rowcounter);
end
end
axes(plot9)
plot(Dabsorptionscalematrixpre)
set(plot9,'Xgrid','on');
xlim(get(plot52,'xlim'));
ylim(get(plot1,'cLim'));
parameterlist(30)=1;
```

```
end %get(habsorption,'String')=='Y' || get(habsorption,'String')=='Yes'
if get(habsorption,'String')=='N'
Dabsorptionscalematrixpre=zeros(Dsize(1));
axes(plot9)
plot(Dabsorptionscalematrixpre)
set(plot9,'Xgrid','on');
xlim(get(plot52,'xlim'));
ylim([0.7 1.3]);
parameterlist(30)=0;
end %if get(habsorption,'String')=='N'
shearangle=(-str2num(get(hshearangle,'String'))); %reverse shear, and clip the bad edges (columns)
if abs(shearangle)>0
shearform=[1,0,0;shearangle,1,0;0,0,1];
tform_shear=maketform('affine',shearform);
[D_shear xdata ydata]= imtransform(Z_Dsheared, tform_shear);
Z_Dsheared=D_shear;
end
shearangle2=str2num(get(hshearangle2,'String'));
if abs(shearangle2)>0
shearform=[1,0,0;shearangle2,1,0;0,0,1];
tform_shear=maketform('affine',shearform);
[BC_shear xdata ydata]= imtransform(Z_BC, tform_shear);
Z_BCsheared=BC_shear;
else
Z_BCsheared=Z_BC;
end
edgecrop=str2num(get(hbandingcrop,'String'));
sizeTemp=size(Z_BCsheared);
if edgecrop>0
Z_BCsheared(:,sizeTemp(2)-edgecrop+1:sizeTemp(2))=[];
Z_BCsheared(:,1:edgecrop)=[];
Z_BCsheared(sizeTemp(1)-edgecrop+1:sizeTemp(1),:)=[];
Z_BCsheared(1:edgecrop,:)=[];
end
axes(plot42)
imagesc(Z_BCsheared), colormap 'gray',caxis(get(plot12,'clim'))
axis image
caxis(get(plot12,'clim'))
BCsize=size(Z_BCsheared);
set(hshearedsize12,'String',num2str(BCsize(1)));
set(hshearedsize22,'String',num2str(BCsize(2)));
BC_compress=mean(Z_BCsheared); %Banding correction, average all rows
smoothingfunction=smooth(BC_compress,str2num(get(hbandingparameter2,'String')));
```

```
axes(plot52)
plot(smoothingfunction)
set(plot52,'Xgrid','on');
ylim([0.7 1.3]);

sheetcorrectionfactor=zeros(1,BCsize(2));
for sheetcorrectioncol=1:1:BCsize(2)
sheetcorrectionfactor(sheetcorrectioncol)=max(smoothingfunction)./smoothingfunction(
sheetcorrectioncol);
end
for sheetcorrectionrow=1:1:BCsize(1)
for sheetcorrectioncol=1:1:BCsize(2)
Z_BCsheared(sheetcorrectionrow,sheetcorrectioncol)=Z_BCsheared(sheetcorrectionrow
,sheetcorrectioncol)*sheetcorrectionfactor(sheetcorrectioncol);
end
end
if get(habsorption,'String')=='Y'
BCtopaverage=mean(mean(Z_BCsheared(2,50:BCsize(2)-50)));   %absorption
correction     %give 50 pixels leeway on each side for shearing.
BCbottomaverage=mean(mean(Z_BCsheared(BCsize(1)-1,50:BCsize(2)-50)));
BCtopaverage=mean(mean(Z_BCsheared(2,50:BCsize(2)-50)));
BCbottomaverage=mean(mean(Z_BCsheared(BCsize(1)-1,50:BCsize(2)-50)));
BCabsorptionscalematrixpre=[BCtopaverage:(BCbottomaverage-
BCtopaverage)/(BCsize(1)-1):BCbottomaverage]; %create the linear fit
BCabsorptionscalematrixpre=BCabsorptionscalematrixpre/max(BCabsorptionscalematri
xpre);
size(BCabsorptionscalematrixpre);
for columncounter=1:1:BCsize(2)
for rowcounter=1:1:BCsize(1)
Z_BCsheared(rowcounter,columncounter)=Z_BCsheared(rowcounter,columncounter)/B
Cabsorptionscalematrixpre(rowcounter);
end
end
axes(plot92)
plot(BCabsorptionscalematrixpre)
set(plot92,'Xgrid','on');
xlim(get(plot52,'xlim'));
ylim([0.7 1.3]);
parameterlist(30)=1;
end %get(habsorption,'String')=='Y' || get(habsorption,'String')=='Yes'
if get(habsorption,'String')=='N'
BCabsorptionscalematrixpre=zeros(BCsize(1));
axes(plot92)
plot(BCabsorptionscalematrixpre)
```

```
set(plot92,'Xgrid','on');
xlim(get(plot52,'xlim'));
ylim([0.7 1.3]);
parameterlist(30)=0;
end %if get(habsorption,'String')=='N'
shearangle2=(-str2num(get(hshearangle2,'String'))); %reverse shear, and clip the bad
edges (columns)
if abs(shearangle2)>0
shearform=[1,0,0;shearangle2,1,0;0,0,1];
tform_shear=maketform('affine',shearform);
[BC_shear xdata ydata]= imtransform(Z_BCsheared, tform_shear);
Z_BCsheared=BC_shear;
end
Z_Dshearedsize=size(Z_Dsheared); %crop to smallest size
Z_BCshearedsize=size(Z_BCsheared);
Dremainingsize=str2num(get(hshearedsize2,'String'))-
2*(str2num(get(hcropcolumns,'String')));
BCremainingsize=str2num(get(hshearedsize22,'String'))-
2*(str2num(get(hcropcolumns2,'String')));
minimages=min(horzcat(Dremainingsize, BCremainingsize)); %smaller of the two
cropped images, make both the minimum size
Z_Dshearedcropped=Z_Dsheared(:,round(Z_Dshearedsize(2)/2)-
round(minimages/2)+1:round(Z_Dshearedsize(2)/2)+round(minimages/2));
Z_BCshearedcropped=Z_BCsheared(:,round(Z_BCshearedsize(2)/2)-
round(minimages/2)+1:round(Z_BCshearedsize(2)/2)+round(minimages/2));
Z_Dshearedcroppedsize=size(Z_Dshearedcropped);
Z_BCshearedcroppedsize=size(Z_BCshearedcropped);
set(hcorrectedsize12,'String',num2str(Z_BCshearedcroppedsize(1)));
set(hcorrectedsize22,'String',num2str(Z_BCshearedcroppedsize(2)));
axes(plot82)
imagesc(Z_BCshearedcropped), colormap 'gray',caxis(get(plot12,'clim'))
axis image
caxis(get(plot12,'clim'))
rowslice=Z_BCshearedcropped(round(Z_BCshearedcroppedsize(1)/2),:);
axes(plot72)
plot(rowslice)
set(plot72,'Xgrid','on');
xlim(get(plot52,'xlim'));
ylim([0.7 1.3]);
hold on;
meanplot7=zeros(Z_BCshearedcroppedsize(2)) ;
for iterationrow=1:1:Z_BCshearedcroppedsize(2)
meanplot7(iterationrow)=mean(Z_BCshearedcropped(:,iterationrow));
end
```

```
axes(plot72)
plot(meanplot7,'m')
set(plot72,'Xgrid','on');
ylim([0.7 1.3]);
hold off;
colslice=Z_BCshearedcropped(:,round(Z_BCshearedcroppedsize(2)/2));
axes(plot62)
plot(colslice)
set(plot62,'Xgrid','on');
xlim(get(plot52,'xlim'));
ylim([0.7 1.3]);
hold on;
meanplot6=zeros(Z_BCshearedcroppedsize(1)) ;
for iterationrow=1:1:Z_BCshearedcroppedsize(1)
meanplot6(iterationrow)=mean(Z_BCshearedcropped(iterationrow,:));
end
axes(plot62)
plot(meanplot6,'m')
set(plot62,'Xgrid','on');
ylim([0.7 1.3]);
hold off;
set(hcorrectedsize1,'String',num2str(Z_Dshearedcroppedsize(1)));
set(hcorrectedsize2,'String',num2str(Z_Dshearedcroppedsize(2)));
axes(plot8)
imagesc(Z_Dshearedcropped), colormap 'gray',caxis(get(plot1,'clim'))
axis image
caxis(get(plot1,'clim'))
rowslice=Z_Dshearedcropped(round(Z_Dshearedcroppedsize(1)/2),:);
Z_Dshearedcroppedsize(1)
axes(plot7)
plot(rowslice)
set(plot7,'Xgrid','on');
xlim(get(plot5,'xlim'));
ylim([0.7 1.3]);
hold on;
meanplot7=zeros(Z_Dshearedcroppedsize(2)) ;
for iterationrow=1:1:Z_Dshearedcroppedsize(2)
meanplot7(iterationrow)=mean(Z_Dshearedcropped(:,iterationrow));
end
axes(plot7)
plot(meanplot7,'m')
set(plot7,'Xgrid','on');
ylim([0.7 1.3]);
hold off;
```

```
colslice=Z_Dshearedcropped(:,round(Z_Dshearedcroppedsize(2)/2));
axes(plot6)
plot(colslice)
set(plot6,'Xgrid','on');
xlim(get(plot5,'xlim'));
ylim([0.7 1.3]);
hold on;
meanplot6=zeros(Z_Dshearedcroppedsize(1)) ;
for iterationrow=1:1:Z_Dshearedcroppedsize(1)
meanplot6(iterationrow)=mean(Z_Dshearedcropped(iterationrow,:));
end
axes(plot6)
plot(meanplot6,'m')
set(plot6,'Xgrid','on');
ylim([0.7 1.3]);
hold off;
dlmwrite(strcat(saveroot,'Z_DStage4.txt'),Z_Dshearedcropped,'newline','pc');
dlmwrite(strcat(saveroot,'Z_BCStage4.txt'),Z_BCshearedcropped,'newline','pc');
end %shearbutton_Callback(source,eventdata)
function savefigurebutton2_Callback(source,eventdata)
hgsave(1,strcat(saveroot,'Stage4.fig')); %save figure
parameterlist(17)=str2num(get(hshearangle,'String'));
parameterlist(18)=str2num(get(hbandingparameter ,'String'));
parameterlist(19)=str2num(get(hcropcolumns,'String'));
parameterlist(20)=str2num(get(hshearangle2,'String'));
parameterlist(21)=str2num(get(hbandingparameter2 ,'String'));
parameterlist(22)=str2num(get(hcropcolumns2,'String'));
dlmwrite(strcat(saveroot,'Parameterlist.txt'),parameterlist,'newline','pc');
stage5()
end %savefigurebutton2_Callback(source,eventdata)
end

%=============================================================
=============
function stage5()
close(1)
f = figure('Visible','off','resize','off','Position',[360,1,1400,680]);
set(f,'Name','Stage5: Smoothing and Temperature Extraction')
movegui(f,'northwest')
hsavefiguretext3=
uicontrol('Style','text','String',saveroot,'Position',[980,645,260,58],'max',3,'min',1);
hsavefigure3 = uicontrol('Style','pushbutton','String','SAVE=>6','ForegroundColor','r',...
'Position',[895,673,77,30],'max',5,'min',0,'Callback',{@savefigurebutton3_Callback});
```

```
hsmooth1= uicontrol('Style','edit','String','2','Position',[10,675,65,14]);
hsmooth2= uicontrol('Style','edit','String','2',.'Position',[80,675,65,14]);
hsmooth3= uicontrol('Style','edit','String','15','Position',[150,675,60,14]);
hsmooth1label= uicontrol('Style','text','String','Vert','Position',[10,690,65,14]);
hsmooth2label= uicontrol('Style','text','String','Horz','Position',[80,690,65,14]);
hsmooth3label= uicontrol('Style','text','String','Sigma','Position',[150,690,60,14]);
hsmooth1reslabel=
uicontrol('Style','text','String','Vertres(mm)','Position',[10,660,65,14]);
hsmooth2reslabel=
uicontrol('Style','text','String','Horzres(mm)','Position',[80,660,65,14]);
hsmooth1res= uicontrol('Style','text','String','n/a','Position',[10,645,65,14]);
hsmooth2res= uicontrol('Style','text','String','n/a','Position',[80,645,65,14]);
hsmoothimagebutton=uicontrol('Style','pushbutton','String','SMOOTH!',...
'Position',[215,655,70,35],'max',3,'min',1,'Callback',{ @smoothimagebutton_Callback});
hJhigh= uicontrol('Style','edit','String','16','Position',[300,675,60,15]);
hJlow= uicontrol('Style','edit','String','3','Position',[370,675,60,15]);
hJhighlabel= uicontrol('Style','text','String','Jhigh(Ex.16)','Position',[300,655,60,15]);

hJlowlabel= uicontrol('Style','text','String','Jlow(Ex.3)','Position',[370,655,60,15]);
hTEMPbutton=uicontrol('Style','pushbutton','String','EXTRACT TEMP!!',...
'Position',[440,655,170,35],'max',3,'min',1,'Callback',{ @EXTRACTTEMPbutton_Callba
ck});
hdelE= uicontrol('Style','text','String','n/a','Position',[620,675,60,15]);

hCexp= uicontrol('Style','text','String','n/a','Position',[690,675,60,15]);
hdelElabel= uicontrol('Style','text','String','^E(cm^-1)','Position',[620,655,60,15]);

hCexplabel= uicontrol('Style','text','String','Cexp','Position',[690,655,60,15]);
haveragetopin= uicontrol('Style','text','String','n/a','Position',[760,675,70,15]);
haveragetopinlabel=
uicontrol('Style','text','String','Averagetopin','Position',[760,655,70,15]);
hiteration= uicontrol('Style','text','String','n/a','Position',[840,675,40,15]);
haveragetopinlabel= uicontrol('Style','text','String','Iteration','Position',[840,655,40,15]);
plot1 = axes('Units','Pixels','Position',[30,380,260,260]);
plot4 = axes('Units','Pixels','Position',[330,380,260,260]);
plot12 = axes('Units','Pixels','Position',[30,30,260,260]);
plot42 = axes('Units','Pixels','Position',[330,30,260,260]);
plot5 = axes('Units','Pixels','Position',[630,240,400,400]);%Tempplot
plot6 = axes('Units','Pixels','Position',[630,30,400,75]);
plot7 = axes('Units','Pixels','Position',[630,130,400,75]);
plot8 = axes('Units','Pixels','Position',[1070,240,75,400]);
plot9 = axes('Units','Pixels','Position',[1180,240,75,400]);
set(f,'Visible','on');
Z_D=dlmread(strcat(saveroot,'Z_DStage4.txt'));
```

```
axes(plot1)
imagesc(Z_D), colormap 'gray'
axis image
Z_BC=dlmread(strcat(saveroot,'Z_BCStage4.txt'));
axes(plot12)
imagesc(Z_BC), colormap 'gray'
axis image
Dsize=size(Z_D);
BCsize=size(Z_BC);
function smoothimagebutton_Callback(source,eventdata)
gaussianfilter = fspecial('gaussian', [str2num(get(hsmooth1,'String')),
str2num(get(hsmooth2,'String'))], str2num(get(hsmooth3,'String'))); %smooth images
Z_Dsmoothed=filter2(gaussianfilter,Z_D);
axes(plot4)
imagesc(Z_Dsmoothed), colormap 'gray'
axis image
caxis(get(plot1,'clim'))
Z_BCsmoothed=filter2(gaussianfilter,Z_BC);
axes(plot42)
imagesc(Z_BCsmoothed), colormap 'gray'
axis image
caxis(get(plot12,'clim'))
vertressmoothed=(str2num(get(hsmooth1,'String')))*vertres;
set(hsmooth1res,'String',num2str(vertressmoothed));
horzressmoothed=(str2num(get(hsmooth2,'String')))*horzres;
set(hsmooth2res,'String',num2str(horzressmoothed));
end %smoothimagebutton_Callback(source,eventdata)
iterationarray=zeros(1,40);
function EXTRACTTEMPbutton_Callback(source,eventdata)
Bconst=1.67;
delE=(Bconst*str2num(get(hJhigh,'String'))*(str2num(get(hJhigh,'String'))+1))-
(Bconst*str2num(get(hJlow,'String'))*(str2num(get(hJlow,'String'))+1));
set(hdelE,'String',num2str(delE));
K=0.69473;
delEdivK=delE/K;
averagetopin=200; %only set at 200 for first iteration, the program calculates the
averagetopin
iteration=1;
Cideal=1;
Tcal=300;
while averagetopin>300.1||averagetopin<299.9
meanvalueBC=nanmean(nanmean(Z_BCsmoothed));%now calculating temps
meanvalueD=nanmean(nanmean(Z_Dsmoothed));
if iteration==1 %if first iteration, it uses the calculated Cexpcalibration as a first guess
```

```
Cexpcalibrationfirst=((meanvalueD/meanvalueBC)/Cideal)/exp(delEdivK/Tcal);
Temppic=delEdivK./(log((imdivide(Z_Dsmoothed,Z_BCsmoothed))/Cexpcalibrationfirst/Cideal));
iterationarray(iteration)=Cexpcalibrationfirst;
set(hCexp,'String',num2str(Cexpcalibrationfirst));
end
if iteration>1
Temppic=delEdivK./(log((imdivide(Z_Dsmoothed,Z_BCsmoothed))/Cexpcalibration/Cideal));
iterationarray(iteration)=Cexpcalibration;
set(hCexp,'String',num2str(Cexpcalibration));
end
averagetopin=mean(mean(Temppic(20:Dsize(1)-20,20:Dsize(2)-20)));
set(haveragetopin,'String',num2str(averagetopin));
set(hiteration,'String',num2str(iteration));
if iteration<2
if averagetopin>300
MaximumCexp=Cexpcalibrationfirst;
MinimumCexp=Cexpcalibrationfirst-.01;
Cexpcalibration=(MinimumCexp+MaximumCexp)/2;
end
if averagetopin<300
MinimumCexp=Cexpcalibrationfirst;
MaximumCexp=Cexpcalibrationfirst+.01;
Cexpcalibration=(MinimumCexp+MaximumCexp)/2;
end
end
if iteration>=2
if averagetopin>300
MaximumCexp=Cexpcalibration;
Cexpcalibration=(MinimumCexp+MaximumCexp)/2;
end
if averagetopin<300
MinimumCexp=Cexpcalibration;
Cexpcalibration=(MinimumCexp+MaximumCexp)/2;
end
end
iteration=iteration+1;
if iteration>=40 %safety to break out of while loop. It will not write any files for ditched sets
break
end
end %averagetopin>300.1||averagetopin<299.9
axes(plot5)
```

```
imagesc(Temppic),colormap 'gray',colorbar,caxis([280,320]) %averaged temp line
axis image
Temppicsize=size(Temppic);
dlmwrite(strcat(saveroot,'TempStage5.txt'),Temppic,'newline','pc');
dlmwrite(strcat(saveroot,'Z_DStage5.txt'),Z_Dsmoothed,'newline','pc');
dlmwrite(strcat(saveroot,'Z_BCStage5.txt'),Z_BCsmoothed,'newline','pc');
rowslice=Temppic(round(Temppicsize(1)/2),:);
axes(plot6)
plot(rowslice)
set(plot6,'Xgrid','on');
ylim(get(plot5,'cLim'));
rowaverage=mean(Temppic);
axes(plot7)
plot(rowaverage)
set(plot7,'Xgrid','on');
ylim(get(plot5,'cLim'));
for rowcounter=1:1:Temppicsize(1)
Temppicslicemean(rowcounter)=mean(Temppic(rowcounter,:));
end
axes(plot8)
plot(Temppicslicemean,1:1:Temppicsize(1))
set(plot8,'Ygrid','on');
xlim(get(plot5,'cLim'));
set(gca,'YDir','reverse')
axes(plot9)
colslice=Temppic(:,round(Temppicsize(2)/2));
plot(colslice,1:1:Temppicsize(1))
set(plot9,'Ygrid','on');
xlim(get(plot5,'cLim'));
set(gca,'YDir','reverse')
end %EXTRACTTEMPbutton_Callback(source,eventdata)
function savefigurebutton3_Callback(source,eventdata)
hgsave(1,strcat(saveroot,'Stage5.fig')); %save figure
parameterlist(23)=str2num(get(hsmooth1,'String'));
parameterlist(24)=str2num(get(hsmooth2,'String'));
parameterlist(25)=str2num(get(hsmooth3,'String'));
parameterlist(26)=str2num(get(hJhigh,'String'));
parameterlist(27)=str2num(get(hJlow,'String'));
parameterlist(28)=str2num(get(hsmooth1res,'String'));
parameterlist(29)=str2num(get(hsmooth2res,'String'));
dlmwrite(strcat(saveroot,'Parameterlist.txt'),parameterlist,'newline','pc');
stage6()
end %savefigurebutton2_Callback(source,eventdata)
end
```

```
%==============================================================
=========
function stage6()
close(1)
f = figure('Visible','off','resize','off','Position',[360,1,1400,680]);
set(f,'Name','Stage6: Mass Processing')
movegui(f,'northwest')
htowritefiles= uicontrol('Style','edit','String','C:\Documents and
Settings\Andrea\Desktop\Altered\20080707\RUN2\TEMPPROCESSED\','Position',[100
,659,560,14],'HorizontalAlignment','left');
htowritefileslabel= uicontrol('Style','text','String','TOWRITE',...
'Position',[5,659,90,14]);
hDtogetfilesroot= uicontrol('Style','edit','String','C:\Documents and
Settings\Andrea\Desktop\Altered\20080707\RUN2\D_RUN2_ (',...
'Position',[100,689,560,14],'HorizontalAlignment','left');
hDtogetfilesrootlabel= uicontrol('Style','text','String','TOGET, D ROOT',...
'Position',[5,689,90,14]);
hDtogetfilesleaf= uicontrol('Style','edit','String',').tif',...
'Position',[705,689,70,14],'HorizontalAlignment','left');
hDtogetfilesleaflabel= uicontrol('Style','text','String','TERM',...
'Position',[665,689,35,14]);
hBCtogetfilesleaf=
uicontrol('Style','edit','String',').tif','Position',[705,674,70,14],'HorizontalAlignment','left')
;
hBCtogetfilesleaflabel=
uicontrol('Style','text','String','TERM','Position',[665,674,35,14]);
hBCtogetfilesroot= uicontrol('Style','edit','String','C:\Documents and
Settings\Andrea\Desktop\Altered\20080707\RUN2\BC_RUN2_
(','Position',[100,674,560,14],'HorizontalAlignment','left');
hBCtogetfilesrootlabel= uicontrol('Style','text','String','TOGET, BC
ROOT','Position',[5,674,90,14]);
htogetparameterlist= uicontrol('Style','edit','String','C:\Documents and
Settings\Andrea\Desktop\Altered\20080707\RUN2\TEMPPROCESSED\Parameterlist.tx
t',...
'Position',[100,644,560,14],'HorizontalAlignment','left');
htogetparameterlistlabel=
uicontrol('Style','text','String','PARAMETERS','Position',[5,644,90,14]);
hpicbeginlabel=uicontrol('Style','text','String','Picbegin','Position',[800,689,70,14]);
hpicendlabel=uicontrol('Style','text','String','Picend','Position',[800,674,70,14]);
hpicbegin=uicontrol('Style','edit','String','100','Position',[880,689,60,14]);
hpicend=uicontrol('Style','edit','String','100','Position',[880,674,60,14]);
hfilenumberlabel=uicontrol('Style','text','String','Filenumber','Position',[1070,689,70,14])
;
```

```
haveragetopinlabel=uicontrol('Style','text','String','Averagetopin','Position',[1070,674,70,
14]);
hCexplabel=uicontrol('Style','text','String','Cexp','Position',[1070,659,70,14]);
hiterationlabel=uicontrol('Style','text','String','Iteration','Position',[1070,644,70,14]);
hfilenumber=uicontrol('Style','edit','String','Filenumber','Position',[1150,689,70,14]);
haveragetopin=uicontrol('Style','edit','String','Averagetopin','Position',[1150,674,70,14]);
hCexp=uicontrol('Style','edit','String','Cexp','Position',[1150,659,70,14]);
hiteration=uicontrol('Style','edit','String','Cexp','Position',[1150,644,70,14]);
hticlabel=uicontrol('Style','text','String','Time(s)/Tempmap','Position',[1060,150,90,14]);
htic=uicontrol('Style','text','String','n/a','Position',[1160,150,50,14]);
htictotallabel=uicontrol('Style','text','String','Time left(min)','Position',[1060,165,90,14]);
htictotal=uicontrol('Style','text','String','n/a','Position',[1160,165,50,14]);
hstatuslabel=uicontrol('Style','text','String','STATUS','Position',[1060,100,150,45],'FontS
ize',15,'BackgroundColor','blue');
hhorzrestotallabel=uicontrol('Style','text','String','Horz
res(mm)','Position',[1060,85,90,14]);
hvertrestotallabel=uicontrol('Style','text','String','Vert
res(mm)','Position',[1060,70,90,14]);
hhorzrestotal=uicontrol('Style','text','String','n/a','Position',[1160,85,50,14]);
hvertrestotal=uicontrol('Style','text','String','n/a','Position',[1160,70,50,14]);
hGObutton =
uicontrol('Style','pushbutton','String','PROCESS','ForegroundColor','r','FontSize',12,'Font
Weight','bold',...
'Position',[950,655,100,48],'max',5,'min',0,'Callback',{ @PROCESSBUTTON_Callback}
);
plot1 = axes('Units','Pixels','Position',[30,350,260,260]);
plot2 = axes('Units','Pixels','Position',[325,350,260,260]);
plot12 = axes('Units','Pixels','Position',[30,30,260,260]);
plot22 = axes('Units','Pixels','Position',[330,30,260,260]);
plot5 = axes('Units','Pixels','Position',[630,240,390,390]);%Tempplot
plot6 = axes('Units','Pixels','Position',[630,30,390,75]);
plot7 = axes('Units','Pixels','Position',[630,130,390,75]);
plot8 = axes('Units','Pixels','Position',[1070,240,75,390]);
plot9 = axes('Units','Pixels','Position',[1180,240,75,390]);
set(f,'Visible','on');

%---------------------------begin code--------------------------%
function PROCESSBUTTON_Callback(source,eventdata)

picbegin=str2num(get(hpicbegin,'String'));
picend=str2num(get(hpicend,'String'));
set(hstatuslabel,'String','RUNNING');
set(hstatuslabel,'BackgroundColor','r');
parameterlistfilename=strcat(get(htogetparameterlist,'String'));
```

```
parameterlist=dlmread(parameterlistfilename);
set(hhorzrestotal,'String',num2str(parameterlist(29)));
set(hvertrestotal,'String',num2str(parameterlist(28)));
%read in the mean
meanDimage=dlmread(strcat(saveroot,'Z_Dmeansmoothed.txt'));
meanBCimage=dlmread(strcat(saveroot,'Z_BCmeansmoothed.txt'));
Dbackgroundimage=dlmread(strcat(saveroot,'Z_Dbackgroundsmoothed.txt'));
BCbackgroundimage=dlmread(strcat(saveroot,'Z_BCbackgroundsmoothed.txt'));
for filenumber=picbegin:1:picend
tic %track time for processing
for iteration=1:1:2 %run once for D, once for BC
if iteration==1
Warpfilename=strcat(get(hDtogetfilesroot,'String'),num2str(filenumber),get(hDtogetfiles
leaf,'String')); %set this to image to warp
else
Warpfilename=strcat(get(hBCtogetfilesroot,'String'),num2str(filenumber),get(hBCtogetf
ilesleaf,'String'));
%set this to image to warp
end
TOWARP=imread(Warpfilename);
TOWARP=im2double(im2uint16(TOWARP));
squaredimension=size(TOWARP);
squaredimension=squaredimension(1);
if iteration==1
axes(plot1) %plot original image
else
axes(plot12)
end
imagesc(TOWARP),colormap 'gray'
axis image
hold on
xcoord1=parameterlist(1);
xcoord2=parameterlist(2);
xcoord3=parameterlist(3);
xcoord4=parameterlist(4);
ycoord1=parameterlist(5);
ycoord2=parameterlist(6);
ycoord3=parameterlist(7);
ycoord4=parameterlist(8);
xcoord12=parameterlist(9);
xcoord22=parameterlist(10);
xcoord32=parameterlist(11);
xcoord42=parameterlist(12);
ycoord12=parameterlist(13);
```

```
ycoord22=parameterlist(14);
ycoord32=parameterlist(15);
ycoord42=parameterlist(16);
if iteration==1
plot(xcoord1,ycoord1,'x','MarkerSize',15,'MarkerEdgeColor','r')
plot(xcoord2,ycoord2,'x','MarkerSize',15,'MarkerEdgeColor','r')
plot(xcoord3,ycoord3,'x','MarkerSize',15,'MarkerEdgeColor','r')
plot(xcoord4,ycoord4,'x','MarkerSize',15,'MarkerEdgeColor','r')
hold off
else
plot(xcoord12,ycoord12,'x','MarkerSize',15,'MarkerEdgeColor','r')
plot(xcoord22,ycoord22,'x','MarkerSize',15,'MarkerEdgeColor','r')
plot(xcoord32,ycoord32,'x','MarkerSize',15,'MarkerEdgeColor','r')
plot(xcoord42,ycoord42,'x','MarkerSize',15,'MarkerEdgeColor','r')
hold off
end
xiterations=max(horzcat(xcoord1,xcoord2,xcoord3,xcoord4))-
min(horzcat(xcoord1,xcoord2,xcoord3,xcoord4));
yiterations=max(horzcat(ycoord1,ycoord2,ycoord3,ycoord4))-
min(horzcat(ycoord1,ycoord2,ycoord3,ycoord4));
iterations=xiterations*yiterations;
if iteration==1
X=[xcoord1;xcoord2;xcoord3;xcoord4];
Y=[ycoord1;ycoord2;ycoord3;ycoord4];
else
X=[xcoord12;xcoord22;xcoord32;xcoord42];
Y=[ycoord12;ycoord22;ycoord32;ycoord42];
end
Xp=[0;0;squaredimension;squaredimension];
Yp=[0;squaredimension;squaredimension;0];
B = [ X Y ones(size(X)) zeros(4,3) -X.*Xp -Y.*Xp ...
zeros(4,3) X Y ones(size(X)) -X.*Yp -Y.*Yp ];
B = reshape(B',8,8)';
D = [Xp,Yp];
D = reshape(D',8,1);
l = inv(B'*B)*B'*D;
A = reshape([l(1:6)' 0 0 1 ],3,3)';
C = [l(7:8)' 1];
Secondsignalx=zeros();
Secondsignaly=zeros();
Secondsignalz=zeros();
counter=0;
for x=floor(min(X)):1:ceil(max(X))
for y=floor(min(Y)):1:ceil(max(Y))
```

```
counter=counter+1;
if rem(counter,5000)==0
counter
end
t=A*[x;y;1]/(C*[x;y;1]);
Secondsignalx(counter)=t(1);
Secondsignaly(counter)=t(2);
Secondsignalz(counter)=TOWARP(y,x);
end
end
xlin=linspace(1,squaredimension,squaredimension);
ylin=linspace(1,squaredimension,squaredimension);
[X,Y]=meshgrid(xlin,ylin);
if iteration==1
Z_D=griddata(Secondsignalx,Secondsignaly,Secondsignalz,X,Y,'cubic');
else
Z_BC=griddata(Secondsignalx,Secondsignaly,Secondsignalz,X,Y,'cubic');
end
end %iteration=1:1:2 %run once for D, once for BC
Z_D=imsubtract(Z_D,Dbackgroundimage); %subtract background (image or flatfield)
Z_BC=imsubtract(Z_BC,BCbackgroundimage);
Z_D=imdivide(Z_D,meanDimage); %divide by the mean
Z_BC=imdivide(Z_BC,meanBCimage);

%-------------------------------------------begin stage (2)
shearangle=parameterlist(17); %D corrections
shearform=[1,0,0;shearangle,1,0;0,0,1];
tform_shear=maketform('affine',shearform);
[D_shear xdata ydata]= imtransform(Z_D, tform_shear);
Z_Dsheared=D_shear;
edgecrop=parameterlist(31);
sizeTemp=size(Z_Dsheared);
if edgecrop>0
Z_Dsheared(:,sizeTemp(2)-edgecrop+1:sizeTemp(2))=[];
Z_Dsheared(:,1:edgecrop)=[];
Z_Dsheared(sizeTemp(1)-edgecrop+1:sizeTemp(1),:)=[];
Z_Dsheared(1:edgecrop,:)=[];
end
Dsize=size(Z_Dsheared);
hshearedsize1=(Dsize(1));
hshearedsize2=(Dsize(2));
D_compress=mean(Z_Dsheared); %Banding correction, average all rows
smoothingfunction=smooth(D_compress,parameterlist(18));
sheetcorrectionfactor=zeros(1,Dsize(2));
```

```
for sheetcorrectioncol=1:1:Dsize(2)
sheetcorrectionfactor(sheetcorrectioncol)=max(smoothingfunction)./smoothingfunction(
sheetcorrectioncol);
end
for sheetcorrectionrow=1:1:Dsize(1)
for sheetcorrectioncol=1:1:Dsize(2)
Z_Dsheared(sheetcorrectionrow,sheetcorrectioncol)=Z_Dsheared(sheetcorrectionrow,sh
eetcorrectioncol)*sheetcorrectionfactor(sheetcorrectioncol);
end
end
if parameterlist(30)==1
Dtopaverage=mean(mean(Z_Dsheared(2,50:Dsize(2)-50)));   %absorption
correction     %give 50 pixels leeway on each side for shearing.
Dbottomaverage=mean(mean(Z_Dsheared(Dsize(1)-1,50:Dsize(2)-50)));
Dtopaverage=mean(mean(Z_Dsheared(2,50:Dsize(2)-50)));
Dbottomaverage=mean(mean(Z_Dsheared(Dsize(1)-1,50:Dsize(2)-50)));
Dabsorptionscalematrixpre=[Dtopaverage:(Dbottomaverage-Dtopaverage)/(Dsize(1)-
1):Dbottomaverage]; %create the linear fit
Dabsorptionscalematrixpre=Dabsorptionscalematrixpre/max(Dabsorptionscalematrixpre
);
size(Dabsorptionscalematrixpre);
for columncounter=1:1:Dsize(2)
for rowcounter=1:1:Dsize(1)
Z_Dsheared(rowcounter,columncounter)=Z_Dsheared(rowcounter,columncounter)/Dabs
orptionscalematrixpre(rowcounter);
end
end
end %if parameterlist(28)==1
shearangle=(-shearangle); %reverse shear, and clip the bad edges (columns)
shearform=[1,0,0;shearangle,1,0;0,0,1];
tform_shear=maketform('affine',shearform);
[D_shear xdata ydata]= imtransform(Z_Dsheared, tform_shear);
Z_Dsheared=D_shear;
shearangle=parameterlist(20); %----------------------BC processing----------%
shearform=[1,0,0;shearangle,1,0;0,0,1];
tform_shear=maketform('affine',shearform);
[BC_shear xdata ydata]= imtransform(Z_BC, tform_shear);
Z_BCsheared=BC_shear;
edgecrop=parameterlist(31);
sizeTemp=size(Z_BCsheared);
if edgecrop>0
Z_BCsheared(:,sizeTemp(2)-edgecrop+1:sizeTemp(2))=[];
Z_BCsheared(:,1:edgecrop)=[];
Z_BCsheared(sizeTemp(1)-edgecrop+1:sizeTemp(1),:)=[];
Z_BCsheared(1:edgecrop,:)=[];
end
BCsize=size(Z_BCsheared);
hshearedsize12=(BCsize(1));
hshearedsize22=(BCsize(2));
BC_compress=mean(Z_BCsheared); %Banding correction, average all rows
smoothingfunction=smooth(BC_compress,parameterlist(21));
sheetcorrectionfactor=zeros(1,BCsize(2));
for sheetcorrectioncol=1:1:BCsize(2)
sheetcorrectionfactor(sheetcorrectioncol)=max(smoothingfunction)./smoothingfunction(
sheetcorrectioncol);
end
for sheetcorrectionrow=1:1:BCsize(1)
for sheetcorrectioncol=1:1:BCsize(2)
Z_BCsheared(sheetcorrectionrow,sheetcorrectioncol)=Z_BCsheared(sheetcorrectionrow
,sheetcorrectioncol)*sheetcorrectionfactor(sheetcorrectioncol);
end
end
if parameterlist(30)==1
BCtopaverage=mean(mean(Z_BCsheared(2,50:BCsize(2)-50)));   %absorption
correction     %give 50 pixels leeway on each side for shearing.
BCbottomaverage=mean(mean(Z_BCsheared(BCsize(1)-1,50:BCsize(2)-50)));
BCtopaverage=mean(mean(Z_BCsheared(2,50:BCsize(2)-50)));
BCbottomaverage=mean(mean(Z_BCsheared(BCsize(1)-1,50:BCsize(2)-50)));
BCabsorptionscalematrixpre=[BCtopaverage:(BCbottomaverage-
BCtopaverage)/(BCsize(1)-1):BCbottomaverage]; %create the linear fit
BCabsorptionscalematrixpre=BCabsorptionscalematrixpre/max(BCabsorptionscalematri
xpre);
size(BCabsorptionscalematrixpre);
for columncounter=1:1:BCsize(2)
for rowcounter=1:1:BCsize(1)
Z_BCsheared(rowcounter,columncounter)=Z_BCsheared(rowcounter,columncounter)/B
Cabsorptionscalematrixpre(rowcounter);
end
end
end %if parameterlist(28)==1
shearangle=(-shearangle); %reverse shear, and clip the bad edges (columns)
shearform=[1,0,0;shearangle,1,0;0,0,1];
tform_shear=maketform('affine',shearform);
[BC_shear xdata ydata]= imtransform(Z_BCsheared, tform_shear);
Z_BCsheared=BC_shear;
Z_Dshearedsize=size(Z_Dsheared); %crop to smallest size
Z_BCshearedsize=size(Z_BCsheared);
Dremainingsize=hshearedsize2-2*parameterlist(19);
```

```
BCremainingsize=hshearedsize22-2*parameterlist(22);
minimages=min(horzcat(Dremainingsize, BCremainingsize)); %smaller of the two
cropped images, make both the minimum size
Z_Dshearedcropped=Z_Dsheared(:,round(Z_Dshearedsize(2)/2)-
round(minimages/2)+1:round(Z_Dshearedsize(2)/2)+round(minimages/2));
Z_BCshearedcropped=Z_BCsheared(:,round(Z_BCshearedsize(2)/2)-
round(minimages/2)+1:round(Z_BCshearedsize(2)/2)+round(minimages/2));
Z_Dshearedcroppedsize=size(Z_Dshearedcropped);
Z_BCshearedcroppedsize=size(Z_BCshearedcropped);
hcorrectedsize12=(Z_BCshearedcroppedsize(1));
hcorrectedsize22=(Z_BCshearedcroppedsize(2));
hcorrectedsize1=(Z_Dshearedcroppedsize(1));
hcorrectedsize2=(Z_Dshearedcroppedsize(2));

%-------------------------------------------begin stage (3)
Dsize=size(Z_Dshearedcropped);
BCsize=size(Z_BCshearedcropped);
gaussianfilter = fspecial('gaussian', [parameterlist(23), parameterlist(24)],
parameterlist(25)); %smooth images
Z_Dsmoothed=filter2(gaussianfilter,Z_Dshearedcropped);
axes(plot2)
imagesc(Z_Dsmoothed), colormap 'gray'
axis image
axis([1 squaredimension 1 squaredimension 0 1 0 4])
Z_BCsmoothed=filter2(gaussianfilter,Z_BCshearedcropped);
axes(plot22)
imagesc(Z_BCsmoothed), colormap 'gray'
axis image
axis([1 squaredimension 1 squaredimension 0 1 0 4])
iterationarray=zeros(1,40);
Bconst=1.67;
delE=(Bconst*parameterlist(26)*(parameterlist(26)+1))-
(Bconst*parameterlist(27)*(parameterlist(27)+1));
K=0.69473;
delEdivK=delE/K;
averagetopin=200; %only set at 200 for first iteration, the program calculates the
averagetopin
iteration=1;
Cideal=1;
Tcal=300;
while averagetopin>300.1||averagetopin<299.9
meanvalueBC=nanmean(nanmean(Z_BCsmoothed));%now calculating temps
meanvalueD=nanmean(nanmean(Z_Dsmoothed));
if iteration==1 %if first iteration, it uses the calculated Cexpcalibration as a first guess
```

```
Cexpcalibrationfirst=((meanvalueD/meanvalueBC)/Cideal)/exp(delEdivK/Tcal);
Temppic=delEdivK./(log((imdivide(Z_Dsmoothed,Z_BCsmoothed))/Cexpcalibrationfirst/Cideal));
iterationarray(iteration)=Cexpcalibrationfirst;
set(hCexp,'String',num2str(Cexpcalibrationfirst));
end
if iteration>1
Temppic=delEdivK./(log((imdivide(Z_Dsmoothed,Z_BCsmoothed))/Cexpcalibration/Cideal));
iterationarray(iteration)=Cexpcalibration;
set(hCexp,'String',num2str(Cexpcalibration));
end
averagetopin=mean(mean(Temppic(20:Dsize(1)-20,20:Dsize(2)-20)));
set(haveragetopin,'String',num2str(averagetopin));
set(hiteration,'String',num2str(iteration));
set(hfilenumber,'String',num2str(filenumber));
if iteration<2
if averagetopin>300
MaximumCexp=Cexpcalibrationfirst;
MinimumCexp=Cexpcalibrationfirst-.01;
Cexpcalibration=(MinimumCexp+MaximumCexp)/2;
end
if averagetopin<300
MinimumCexp=Cexpcalibrationfirst;
MaximumCexp=Cexpcalibrationfirst+.01;
Cexpcalibration=(MinimumCexp+MaximumCexp)/2;
end
end
if iteration>=2
if averagetopin>300
MaximumCexp=Cexpcalibration;
Cexpcalibration=(MinimumCexp+MaximumCexp)/2;
end
if averagetopin<300
MinimumCexp=Cexpcalibration;
Cexpcalibration=(MinimumCexp+MaximumCexp)/2;
end
end
iteration=iteration+1;
if iteration>=40 %safety to break out of while loop. It will not write any files for ditched sets
break
end
end %averagetopin>300.1||averagetopin<299.9
```

```
axes(plot5)
imagesc(Temppic),colormap 'gray',colorbar,caxis([280,320]) %averaged temp line
axis image
Temppicsize=size(Temppic);
dlmwrite(strcat(get(htowritefiles,'String'),'TempStage6_',num2str(filenumber),'.txt'),Temppic,'newline','pc');
rowslice=Temppic(round(Temppicsize(1)/2),:);
axes(plot6)
plot(rowslice)
set(plot6,'Xgrid','on');
ylim(get(plot5,'cLim'));
rowaverage=mean(Temppic);
axes(plot7)
plot(rowaverage)
set(plot7,'Xgrid','on');
ylim(get(plot5,'cLim'));
for rowcounter=1:1:Temppicsize(1)
Temppicslicemean(rowcounter)=mean(Temppic(rowcounter,:));
end
axes(plot8)
plot(Temppicslicemean,1:1:Temppicsize(1))
set(plot8,'Ygrid','on');
xlim(get(plot5,'cLim'));
set(gca,'YDir','reverse')
axes(plot9)
colslice=Temppic(:,round(Temppicsize(2)/2));
plot(colslice,1:1:Temppicsize(1))
set(plot9,'Ygrid','on');
xlim(get(plot5,'cLim'));
set(gca,'YDir','reverse')
if filenumber==picbegin  %save the first stage5
hgsave(1,strcat(get(htowritefiles,'String'),'Stage6.fig')); %save figure
end
toc;
set(htic,'String',(num2str(toc)));
timeleft=((((picend-picbegin+1)*toc)-((filenumber-picbegin+1)*toc))/60); %total time in min
set(htictotal,'String',num2str(timeleft));
end %filenumber=picbegin:1:picend
set(hstatuslabel,'String','FINISHED');
set(hstatuslabel,'BackgroundColor','blue');
end %PROCESSBUTTON_Callback(source,eventdata)
end
end
```

VITA

Name: Andrea Grace Hsu

Address: Department of Chemistry
Texas A&M University
P.O. Box 30012
College Station, TX 77843-3012

Email Address: HsuAndreaG@yahoo.com

Education: B.S., Chemical Engineering, The University of Texas at Austin, 2003
Ph.D., Chemistry, Texas A&M University, 2009