Tanks for the Memories: Constructing a Particle Image Velocimetry (PIV) System for Low Reynolds Number Biological Hydrodynamics

Thesis by
Blair Subbaraman

Supervised By
Dwight Whitaker, Ph.D

In Partial Fulfillment of the Requirements for the
Degree of
Bachelor of Arts in Physics

POMONA COLLEGE
Claremont, California

ACKNOWLEDGEMENTS

I would like to thank Dwight Whitaker for his guidance, both for this thesis and over the last four years. I would also like to thank my family and friends for their constant love and support. And of course, I would like to thank the students, faculty, and staff here in the Physics and Astronomy department for creating such a wonderful, welcoming place to spend my undergraduate years.

ABSTRACT

This thesis documents work on the towing tank in the Pomona College Fluids Lab. Efforts are focused on the controls and imaging systems for particle image velocimetry (PIV) experiments. A custom GUI and associated hardware implemented to run the tank are presented, followed by PIV experiments used to inform final design choices for the imaging system. Results indicate that of the cameras tested, the Sony Alpha a7s is best suited for collecting PIV video, performing considerably better than other options as quantified by signal-to-noise ratio analyses. Methods for extraction of body forces from PIV data and other flow visualizations are also included, culminating in a lab manual for use by future student researchers.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

Number *Page*

LIST OF TABLES

Number *Page*

Chapter 1

INTRODUCTION

Almost all of the natural phenomena that we observe around us occur in fluids. Not only must scientists navigate the physics of fluids in most any engineering project, life itself has been tirelessly adapting to the problems posed by living in fluids for billions of years. It is at this interface between physics and biology that inspires much current research in fluid flow, and this thesis is no exception. By investigating the physics of a given biological system we are able to learn about the evolutionary morphology that enabled the life and longevity of the organism in question [7]. Moreover, nature has engineered incredibly creative solutions to singularly difficult problems. The analysis of biological systems can in turn guide the creation of future technologies through bio-inspired design [5].

Of course, our scientific investigations must remain grounded in the laboratory. The tool with which we hope to investigate fluid flow is the towing tank. The following sections introduce the tow tank as an effective experimental fluid mechanical facility. We then present the objectives of this thesis, which includes a design and test phase.

1.1 Towing Tanks

The Navier-Stokes equation is the equation of motion for incompressible, viscous, Newtonian fluids. Given this governing equation, we should theoretically be able to compute the flow patterns for any given problem. However, the equation yields highly nonlinear partial differential equations that have few analytic solutions. In fact, proving the existence and smoothness of solutions to the Navier-Stokes equation is one of the Clay Mathematics Institutes seven most important open problems in mathematics, and a successful proof or counterexample would be rewarded with $1,000,000 [11]. A more rigorous introduction to Navier-Stokes will be given in Chapter 2. For now, it suffices to say that this thesis avoids the Navier-Stokes equation. Instead, we turn to experimental techniques to understand the physics of fluid flow. There are various flow rigs that could be used to conduct

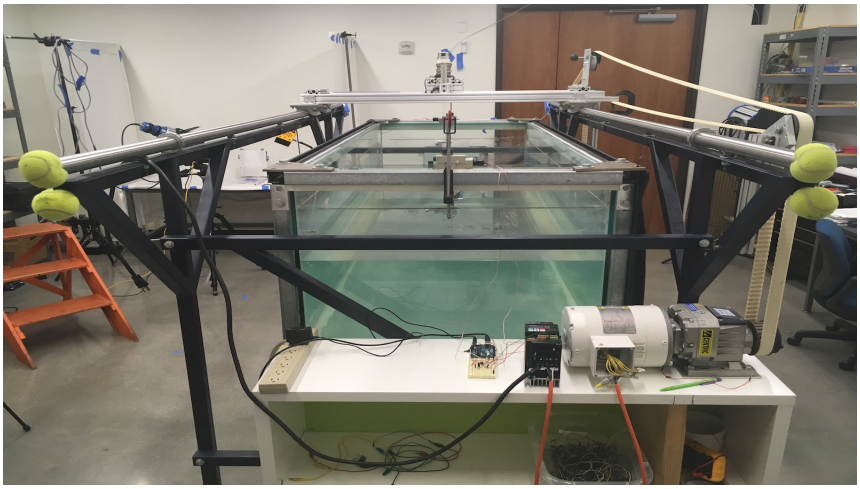


Figure 1.1: The Pomona College Tow Tank. Image taken on 11/16/17.

experiments, including the tow tank, wind tunnel and flume. The relative advantages of the tow tank are presented, as well as the motivation behind building one in the Pomona Fluids Lab.

1.1.1 What's a tow tank?

A tow tank consists of a water tank housed within a steel framework. Two rails above the tank carry the towing carriage, from which the name arises - objects are extended from the carriage into the water and are towed through the stationary water. They come in varying sizes: MIT's Large Towing Tank at the Center for Ocean Engineering is 100 ft x 8 ft x 4 ft, and the David W. Taylor Model Basin at the Naval Ship Research and Development Center is 0.9 kilometers, long enough to necessitate accounting for the curvature of the Earth [12, 6].

Figure 1 shows the towing tank in the Pomona College Fluids Lab. Ours is a 2 meter by 0.75 meter by 0.6 meter Fish Tank with a slightly larger framework. The water tank is encased by the support structure, along which the towing carriage rides. The carriage is attached to a belt system, which is attached to an AC motor. The motor is then driven by a variable frequency drive (VFD). Objects can be extended from the carriage into the water and towed at a given speed.

Tow tanks can be compared to other experimental fluid dynamics facilities such as the wind tunnel. Firstly, tow tanks allow for the use of whole-field flow visualization techniques such as particle

image velocimetry (PIV), which will be talked about in detail later [18]. In short, PIV allows for quantitative flow visualization in an entire plane of motion, as opposed to probes which only extract information from a point. Moreover, probes may disrupt flow as they have to be physically immersed in the fluid. PIV, on the other hand, is non-intrusive [18]. Using a tow tank enables the use of these robust techniques. Towing tanks also require less energy to power than wind tunnels [6]. It is more efficient to move an small object through water than it is to move large amounts of fluid past a stationary object, as in a wind tunnel. Finally, towing tanks can handle multi-dimensional object movement [6]. Some tow tanks, for example, include carriages that move like an X-Y plotter which allows for arbitrary tow trajectories.

There are, of course, drawbacks to using a tow tank. The main limitations are given by the physical parameters of the tank itself. The size of the tank and the speed of the motor put limits on the range of phenomenon that we can interrogate. Length limits speed as the carriage must accelerate and decelerate all within a single run. Width impacts the size of the object that we are able to analyze. Too large an object means that boundary layer effects near the wall become non-negligible. All of these limitations must be characterized so that effective experiments can be run; this will be addressed later in this thesis. The motor also results in vibration of the carriage, potentially posing imaging problems.

1.1.2 Previous Work

At this point, it is necessary to credit the work of past students from which this thesis builds. Jonah Grubb (PO '16) sourced many of the mechanical parts for our system. Many pieces were salvaged from a decommissioned flume in John Dabiri's lab at the California Institute of Technology. Jonah modified the fish tank to accommodate the translational carriage system that was to come and reinforced the walls to safely hold large volumes of water without reinforcements over the top of the tank. He also designed the optical system which forms the basis of the particle image velocimetry (PIV) system that is to be described later.

Tom Neumiller (PO' 17) picked up where Jonah left off, designing and constructing the linear translational carriage system described above. As a result of Tom's efforts, we have a working prototype of a tow tank that can safely and reliably operate at a given constant speed. It is on this sound foundation that this thesis rests.

1.1.3 Motivation

The difficulty posed by the Navier-Stokes equation is a central theoretical motivation for the construction of a tow tank at Pomona College. However, there are also pedagogical and biological motivations. Panah et al. write about the construction of a tow tank as a useful senior undergraduate capstone. Particularly, it combines concepts from electrical, mechanical and controls engineering through statics, strength of materials, fluids, programming, and instrumentation [15]. The resulting tank can also be used by future undergraduates for research and educational purposes. Fluids also provide an effective area for science outreach. Tow tanks, when outfitted with flow imaging systems like ours, can provide attractive and informational visualizations - winners of NSF scientific visualization challenge are often fluid dynamicists [13]. The tank can thus serve as an engaging facility for demostrations, photography, and outreach.

There is also a biological motivation for the tank. Many tow tanks were originally constructed for research in naval engineering and ship design testing. More recently, they have been repurposed for experimental hydrodynamics experiments and, in particular, investigations into biological flows [12]. A main area of research has been the movement of animals. The flapping of wings and the propulsion of jellyfish are two examples that have garnered extensive research attention [1]. A first motivation for the tank at Pomona College was to investigate the biomechanics of plants. Specifically, Professor Dwight Whitaker wished to analyze the flight of *Acanthaceae* seeds. Marchetto et al. writes about PIV as an effective method to investigate the seed dispersal of plants: while much PIV research focuses on animals, plants have often been overlooked as a subject of study [9]. Initial construction efforts on the tow tank began looking to conduct these biophysical experiments.

1.2 Objectives

The objectives of this thesis can be sectioned into two categories: **design** and **test**.

Design. As a result of Tom Neumiller's (PO '17) senior thesis, we have a mechanically sound and reliable tow tank in the Fluid Dynamics Lab. Tom provides initial direction for future construction on the tank. Principally, he articulates the need for a more robust controls system. The controls on the VFD that powers our drive motor are clunky and user-unfriendly. There is a need to design a system and interface that makes experimentation both sophisticated and easy. This can catalyze future discoveries made with the tank. Moreover, while there are mechanical stops on the rails to prevent the carriage from dismounting, we would like to avoid collisions. Electrical tripwires would alleviate safety concerns. An initial objective of this thesis is to extend the functionality of the controls system to enable easy, efficient, and safe experimentation. This involves introducing a microcontroller to the system and building a GUI to operate it from.

The imaging system also needs to be finalized. There are several different cameras that could be used, ranging in cost and specs. We must test these cameras under operating conditions to decide upon the one best for our needs. Camera, laser orientation, and mounting techniques must also be experimented with to make an informed decision for the final design.

Test. Upon completion of the controls and imaging systems, the tank needs to be experimentally tested. This involves towing an object whose flow is well known, such as a cylinder. We can attempt the sorts of analysis that are relevant to fluids research on this test data, and if the analysis agrees with the literature, we can be confident in using the tank for other experiments. Information concerning body forces (drag, lift) are of primary importance to us. However, deriving drag estimates from PIV data is not trivial. This thesis presents experimentation in data analysis, providing initial direction for future work. While characterizing the precise limitations of the tank will not be covered, this will also give us an idea of the range of speeds and sizes of phenomenon we can accurately interrogate in our tank. This will result in a lab manual for PIV experiments, from setting up a physical experiment through data analysis (Appendix B).

Chapter 2

THEORY

The experimental fluid dynamics facility is composed of the tow tank, a particle image velocimetry (PIV) flow visualization system, and the PIV post-processing software. This chapter presents the theory that undergirds each of these components. First, relevant topics in fluid mechanics will be introduced leading to a proof of dynamic similarity: two geometrically similar (i.e. scaled) objects with equal Reynolds numbers will produce identical flow fields. Having validated our use of a tow tank to investigate fluid phenomena, we then present the particle image velocimetry (PIV) technique for qualitative and quantitative flow visualization. This includes the experimental materials and methods necessary to outfit our tow tank with such a system. Finally, an overview of the open-source post-processing software PIVlab is given.

2.1 A Brief Overview of Relevant Fluid Mechanics

The following section introduces the basic properties of a fluid as well as the equations of motion for a Newtonian fluid: the Navier-Stokes equation. The definitions and equations will inform our subsequent exploration of particle image velocimetry.

2.1.1 Properties of a Fluid

A fluid is a material with no rigidity. This means that when a shear stress is applied and relieved, the material will not recover its original shape as a solid would. Instead, when subject to shear stress, a fluid will flow. Often times the line between a fluid and solid is blurred. Jellies and jams, for example, exhibit more complicated behavior than water. We will ignore such *non-Newtonian* fluids and instead restrict our scope to "normal", or Newtonian, fluids. Intuition is fairly reliable in deciding what fluids are Newtonian. Water, air, and honey are all Newtonian; ketchup, yogurts, and gelatins are not. A more rigorous classification will be given shortly.

When investigating the physical nature of a fluid, whether it be a liquid or a gas, we treat it as a continuous medium which can be characterized by macroscopic properties. We will briefly introduce the most significant of these properties to inform our discussion of fluid behavior.

Density, ρ. SI Unit: kg/m^3. If we assume the fluid is uniformly dense, then density is the mass of the fluid over a given volume: $\rho = \frac{m}{V}$. Density will change with temperature. For both water and air, a higher temperature generally means lower density, with water having a slight but important exception to this rule near its freezing point.

Dynamic Viscosity, μ, η. SI Unit: $Pa \cdot s$. When one refers to viscosity, they are generally talking about dynamic, or shear, viscosity. Informally, viscosity is a measure of a fluid's resistance to flow. Formally, it is the constant of proportionality that relates shear stress, τ, to the shear strain rate. Figure 2.1 shows two plates separated by a fluid. If we keep the lower plate fixed while moving the upper plate at a constant velocity U, then the fluid will in turn flow parallel to the plates. The velocity of the fluid will increase from no velocity at the bottom plate to U at the top plate - this follows from the no-slip condition, which is the assumption that the velocity of a fluid at a solid-fluid interface is the same as that of the solid. Because each layer is moving faster than the one below it, the friction between them yields an overall force resisting their motion. Most notably, the fluid will exert a force opposite to the direction of U on the top plate. This necessitates a force on the top plate to maintain a constant velocity, and this force is given by:

$$F = \frac{\mu U S}{z} \implies \tau = \mu \frac{dU}{dz} \tag{2.1}$$

where $\tau = \frac{F}{S}$ is the shear stress and $\frac{dU}{dz}$ is the shear strain rate. Viscosity will vary with temperature. At higher temperatures, air will become more viscous while the viscosity of water drops severely. For a sense of scale, the viscosity of water is 0.01 Pa · s at room temperature, while honey's is about 100 Pa · s.

Kinematic Viscosity, ν. SI Unit: $\frac{m^2}{s}$. The kinematic viscosity is the ratio of dynamic viscosity

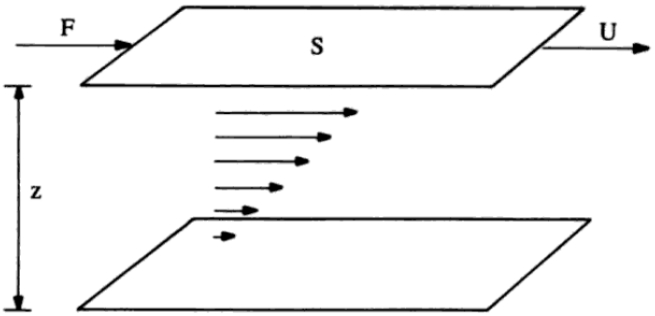


Figure 2.1: The force required to move the top plate at velocity U while keeping the bottom plate fixed is proportional to the dynamic viscosity, μ [22].

and density: $\nu = \frac{\mu}{\rho}$. Like dynamic viscosity, it represents a fluid's resistance to flow.

2.1.2 The Navier-Stokes Equation

The Navier-Stokes equation is the equation of motion of an incompressible viscous fluid. It holds for Newtonian fluids, or fluids with linear viscosity. For a formal derivation, the reader is directed to Faber [3]. While it can be written in various ways, one form of Navier-Stokes equation is given below:

$$\rho(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u}) = -\nabla p^* - \mu \, (\nabla \times \nabla \times \mathbf{u}) \tag{2.2}$$

where $\mathbf{u}$ is the velocity of the fluid parcel, μ is the dynamic viscosity, p^* is the excess pressure, and ρ is fluid density. This is essentially a restatement of Newton's second law, $\vec{F} = m\vec{a}$. As it turns out, this governing equation of Newtonian fluid flow results in a partial non-linear differential equation that has few exact analytic solutions [3]. The culprit is not the double cross product, which in fact simplifies considerably when we use our assumption of incompressible flow ($\nabla \cdot \mathbf{u} = 0$), but rather $(\mathbf{u} \cdot \nabla)\mathbf{u}$ which makes the equation non-linear. The difficulty of this equation motivates the need for alternative methods to evaluate complex flow problems, whether that be computationally or experimentally. In our case, we turn to the towing tank. We wish to validate the use of a tow tank as an experimental method. In the tank, we study the fluid flow of objects with different sizes and

speeds than the corresponding phenomena we are investigating, and claim that these flow patterns are related. This notion relies on the principle of dynamical similarity.

2.1.3 Dynamical Similarity

We wish to show that the flow field that results from towing a scaled object in the tank will be the same as that of the corresponding natural phenomena. In lieu of a formal proof, we will demonstrate dynamical similarity using a dimensional argument. This reasoning relies on the fact that any physically meaningful equation must have the same dimensions on both sides: if X = Y, then X must have the same units as Y.

First, note that the shape of the flow field is dimensionless. It is the path that a given parcel of fluid takes as it moves. By dimensional analysis, this means that any mathematical relationship that describes the flow field must be a dimensionless combination of variables. Consider the fluid dynamical variables that could relate to the flow field. The density of the fluid ρ, the velocity of the fluid with respect the object u, the characteristic length L, the viscosity of the fluid μ, and the shape of the object are all physically relevant variables. We could include compressibility, but we will assume that the pressures being generated will have a negligible effect on the density of the liquid; that is, we are assuming incompressible flow. The only dimensionless combination of these variables is known as the Reynolds number:

$$Re = \frac{\rho u L}{\mu} = \frac{u L}{\nu} \tag{2.3}$$

Therefore, the flow patterns of two differently scaled systems are scaled versions of one another so long as the Reynolds number and the shape of the object are the same. This is the useful result on which tow tanks rely. It indicates that to analyze the flow field of a given phenomenon, we must first recreate the geometry/shape of the object. We can then use a tow tank to move the object through water at a speed that matches the Reynolds number of the original phenomenon. This involves scaling the speed appropriately. What's more, it means that by scaling up objects, we can

decrease the velocity that the object needs to move. Small-scale, high-speed phenomena that would be impossible to otherwise investigate need only be scaled up and slowed down. Additionally, the ratio of kinematic viscosity ν of water to air at room temperature is 12.8, further decreasing the necessary velocity for events that generally occur in air. Similarly, very large objects can be scaled down, requiring a corresponding velocity increase. The limits on what we can investigate with a tow tank are set only by the size of the tank and the speed we can run the motor at, so long as our assumptions that the fluid is incompressible and that there is no heat flow hold under these conditions. The next section explores how we can extract information from the flow fields in the tank via particle image velocimetry (PIV).

2.2 Particle Image Velocimetry (PIV)

Now that we are confident that we can make use of a towing tank to analyze flow fields, we need a technique to extract information from our experimentally produced flow. To do so, we outfit our tank with a particle image velocimetry (PIV) system. The following section explicates the theory of PIV, from setup through post-processing.

2.2.1 Principles of PIV

For many years, investigations of fluid flow were restricted to qualitative statements. While this was and remains to be valuable, PIV provides a synthesis of optics, electronics, video, and computer techniques that extract quantitative measurements of complex velocity fields [18].

Figure 2.2 provides an example PIV arrangement for a wind tunnel. While a PIV system will vary from experiment to experiment, key elements remain the same. Tracer particles are added to the fluid flow. A laser is directed through an optical setup to construct a plane of light. This light-sheet illuminates a plane of tracer particles within the flow, and light scattered by the particles is captured by an imaging system. The particles must be illuminated at least twice in a short period of time which necessitates the use of either a pulsed laser or a camera with a short shutter speed. The tank in the Pomona Fluids lab opts for the latter. This results in short exposure time to avoid blurring

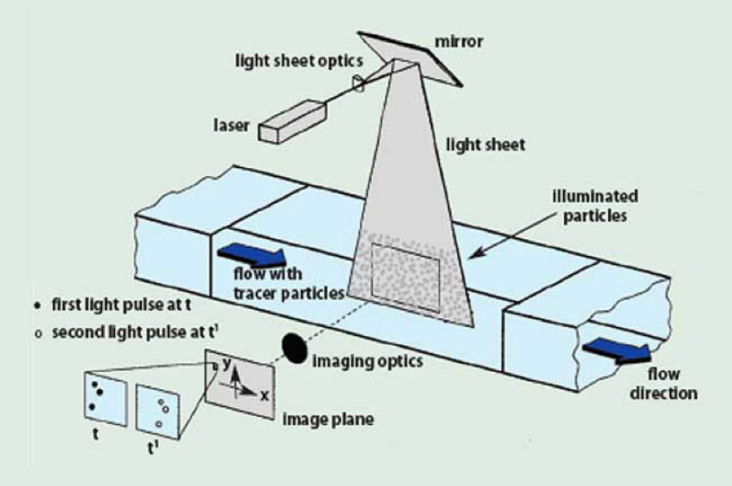


Figure 2.2: Experimental setup for PIV in a wind tunnel [18].

in images. Once a quality video has been recorded, post-processing begins. The frames in the resulting recording are divided into many subsections called interrogation areas. We assume that the particles within a given interrogation area move in a smooth and continuous fashion in the time between each laser pulse. Software is then used to determine the displacement vectors of the tracer particles within each of these areas by using cross-correlation statistical methods. Since the time between frames and any image magnification is known, fluid dynamical variables - such as the local flow velocity - can be calculated. This setup can become costly, as it requires a high intensity laser and a sophisticated imaging system. However, we get in return (1) a non-intrusive imaging technique, which unlike a probe will not disturb the flow field, and (2) large spatial resolution from which we can extract velocity information from the whole field, as opposed to other techniques in which velocity is measured only at a single point [18]. Moreover, significant improvements in camera quality have quickly decreased the cost of PIV systems.

2.2.2 Tracer Particles

The use of tracer particles to indirectly track fluid flow introduces a set of experimental problems to take into account. Selection of the correct type and quantity of tracer particles is critical to PIV. We will break these problems into that of **movement** and **light scattering**.

Movement. In a PIV anlaysis, we are actually making an indirect measurement of the fluid. All post-processing is based on the movement of the particles, not the fluid itself. We therefore need to ensure that the particle motion is representative of the fluid flow. A main error introduced is the difference in gravitational forces if the density of the particles are not the same as the density of the fluid. Raffel et al. solves the equations of particle motion assuming spherical particles at low Reynolds numbers in [18]. Under this assumption, Stokes' drag dominates, and the velocity of a particle induced by gravity $\mathbf{U}_g$ is given by

$$\mathbf{U}_g = d_p^2 \frac{\rho_p - \rho}{18\mu}\mathbf{g} \tag{2.4}$$

where d_p is the diameter of the particle, ρ_p is the density of the particle, $\mathbf{g}$ is the acceleration due to gravity, and μ is dynamic viscosity. Analogously, the velocity lag, or slip velocity, of the particles in a continuously accelerating fluid is

$$\mathbf{U}_{slip} = \mathbf{U}_p - \mathbf{U} = d_p^2 \frac{\rho_p - \rho}{18\mu}\mathbf{a} \tag{2.5}$$

These equations of motion become more complex at higher Reynolds numbers and non-constant accelerations, but equations 2.4 and 2.5 demonstrate that density matching is of critical importance. If a non-negligible slip velocity is present, then we can use equation 2.5 to define a characteristic relaxation time for the particle:

$$\tau = d_p^2 \frac{\rho_p}{18\mu} \tag{2.6}$$

We subsequently define another dimensionless quantity, the Stokes number S_k. The Stokes number is the ratio between τ and the time scale of the smallest motions in the flow, the Kolmogorov time scale τ_k. If $S_k < 0.1$, then the tracer particles will follow the fluid flow with an error below 1% [18]. For our purposes in a tow tank, tracer particles with a density similar to that of water can be sourced easily - density becomes a larger problem in gas flows. Larger particles can be used in liquid flows ($d_p \in [5\mu m, 50\mu m]$) while gas flows necessitate smaller particles ($d_p \in [.5\mu m, 5\mu m]$).

Fluid	Material	Diameter (μm)	Density (kg/m^3)
Air	DEHS	1–3	10^3
–	Glycol–water solution	1–3	10^3
–	Vegetable oil	1–3	10^3
–	TiO_2	0.2–0.5	$1–4\times10^3$
Water	Latex	5–50	10^3
–	Sphericell	10–100	$0.95–1.05\times10^3$
–	Silver-coated hollow glass spheres	30–100	$> 10^3$

Figure 2.3: Common tracer particles used in air and water [18]. We will use silver-coated hollow glass spheres, with and apparent density of 0.49 $\frac{\text{g}}{\text{cc}}$ and average diameter of 80 μm.

The benefit of larger particles is investigated in the following section on light scattering. Figure 2.3 shows materials, sizes, and densities for tracer particles commonly used in water and air.

We use silver coated hollow glass spheres in the Pomona tow tank. They have an apparent density of 0.49 $\frac{\text{g}}{\text{cc}}$ and an average diameter of 80 μm. Using equation 2.6 to find the characteristic relaxation time, we can estimate our Stokes number:

$$S_k = \frac{\tau u_o}{l_o}$$

where u_0 is the free stream velocity and l_0 is the characteristic length of our obstacle.

A generous estimate of the free stream velocity is 1 $\frac{\text{m}}{\text{s}}$. Given the exposure times of our cameras (see section 2.2.3) and the length of our tank (2 meters), we will never run the tank at this speed. The Stokes number will increase with a smaller l_0. Even at a length of one centimeter, the Stokes number is ≈ 0.017. For a sense of scale, the PVC rod that will be used in future experiments in this thesis has a diameter of 2 cm. Thus, we can proceed with confidence that the tracer particles will accurately follow fluid flow.

Light Scattering. A high image intensity - that is, the contrast of the PIV video - is achieved through high scattered light power. A high intensity laser is used to help increase power. However, a smart choice of tracer particles can be less costly than using a higher intensity laser. We want a large signal-to-noise ratio (SNR) of the scattered light signal. This is most easily achieved by

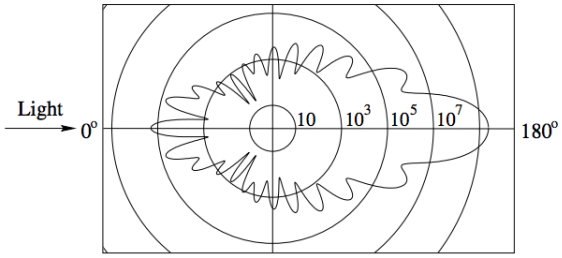


Figure 2.4: Intensity of scattered light is a function of scattering angle [18].

using large tracer particles. Unfortunately, this explicitly collides with our need for small diameter particles to track the fluid flow. Selection of tracer particles largely becomes a negotiation of these two parameters.

Melling cites the scattering cross-section as a good measure of light-scattering capabilities [10]:

$$C_s = \frac{P_s}{I_0} \tag{2.7}$$

This is the ratio of the total scattered power to laser intensity incident on the particle. For example, C_s for a molecule is about $10^{-33}m^2$, while for a 10 micrometer particle, $C_s \approx 10^{-9}m^2$ [10]. Evidently we need to move out of the Rayleigh scattering regime of molecules and into the Mie scattering regime, where the wavelength of light is smaller than the particle diameter to ensure a large enough SNR. Figure 2.4 shows the scattered light intensity of 532 nanometer light incident on a one micron oil particle in air as predicted by Mie's theory from [18]. We see that intensity of scattered light is a function of scattering angle. These plots can be characterized by the normalized diameter:

$$q = \frac{\pi d_p}{\lambda} \tag{2.8}$$

For $q > 1$, there will be approximately q maxima between 0°and 180°, with the ratio of forward to backwards scattering increasing with a larger q. However, PIV experiments are generally conducted with an imaging system that is set up orthogonal to the incident light sheet to resolve displacement

of particles within the plane of motion (see figure 2.2). Since I_{90} is orders of magnitude smaller than the forward scattered light, a powerful laser is imperative. These theoretical observations imply a straightforward result: changing the location, and thus the power per area, of our laser when conducting PIV experiments might have a significant impact on the quality of the resulting video. However, too close a laser means that the light sheet will not have time to spread out, limiting the length over which we can conduct experiments. The light source and associated optical system will be discussed more next.

2.2.3 Optical System

The optical system in a PIV experimental setup consists of the **light source**, a **camera** for imaging, and **light-sheet optics**. The standard setup is described here, along with the equipment and methods that are used in the Pomona Fluids lab.

Light Source. We know now that PIV systems need a high intensity of incident light in order to achieve quality imaging. Lasers become a natural choice of light source. Many PIV systems will use a pulsed laser with short illumination time to avoid streaking in the resulting images since we wish to resolve the tracer particles as distinct points. The system for the Pomona tow tank opts instead for a continuous wave laser using a camera with a short shutter speed to achieve the same effect. We use a 1W 532 nanometer green laser in our experiments.

Camera. In order to avoid streaking, the imaged particle displacement within a given frame must be much smaller than the size of the particle image. This puts a limit on the exposure time δt, and we must be using a camera that can accommodate such rapid shutter speeds. Mathematically, this means that

$$\vec{v}_p \delta t \ll \frac{d_\tau}{M} \implies \delta t \ll \frac{d_\tau}{\vec{v}_p M} \tag{2.9}$$

Equation 2.9 warrants some unpacking. $\vec{v}_p$ is the velocity of the particle and M is the image magnification. The diameter of the particle image is then given by $d_\tau = (M^2 d_p^2 + d_s^2)^{1/2}$, where d_p

is the diameter of the particle and d_s is the diffraction limited spot diameter, which is the maximum diameter that a given optical system can resolve due to the diffraction of light. This value is approximated by the diameter of the first null of the Airy disk, $d_s \approx 2.44(1 + M)f^{\#}\lambda$ where λ is the wavelength of light and $f^{\#}$ is the ratio of the optical system's focal length to the diameter of the entrance pupil [18]. A variety of cameras will be tested for this thesis. The Sony Alpha a7s, for example, has a maximum shutter speed of 0.125 ms and a minimum $f^{\#}$ equal of 3.5. Our tracer particles range in diameter- a conservatively small estimate is 50 μm. We can make an estimate for the magnification, $M \equiv \frac{i}{o}$, of our image using the relationship:

$$\frac{1}{f} = \frac{1}{i} + \frac{1}{o},$$

where i is the distance to from the lens the image, o is the distance from the lens to the object, and f is the focal length of the lens. For a focal length of 50 mm and a typical o of $\approx$ 1 m given the size of our tank, this results in a magnification of $\frac{1}{19}$. Returning to equation 2.9, this gives of an estimate of d_τ for the Sony of roughly 5.26 μm. Given this estimate and its shutter speed we can estimate the maximum velocity which we could image with the Sony. The velocity at which the left and right hand sides of equation 2.9 are equal is $v_p = 0.8\,\frac{\mathrm{m}}{\mathrm{s}}$, so we we wish to stay well below this theoretical maximum. Given the physical limitations of the length of the tank, this should not be an issue.

Light-Sheet Optics. To complete the optical system, the laser needs to shaped into a 2-dimensional light-sheet. The means to do this can vary depending on how wide a light-sheet is needed. Jonah Grubb (PO '16) determined that the use of a Powell lens is ideal given the limited lab space that we are working with. The Powell lens will form a linearly expanding sheet width with uniform illumination, as opposed to the Gaussian distribution given by the standard cylindrical lens (Figure 2.5). This makes the setup simple, since only one lens is needed.

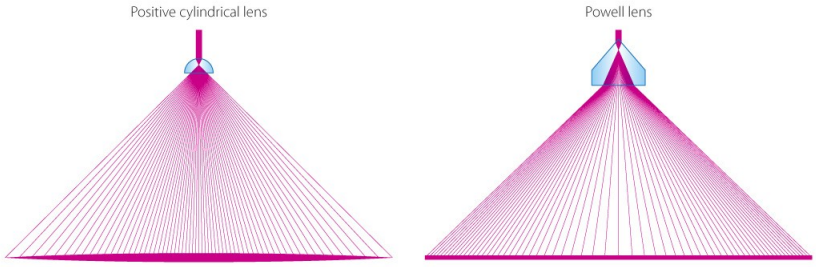


Figure 2.5: Effect of a Powell lens versus cylindrical lens [16]

2.2.4 Image Interrogation

After collecting quality PIV data, computer software is necessary to extract velocity information from the flow field. This process differs in complexity depending on the density of tracer particles in the fluid. If only a small amount of information needs to be extracted from the flow field (i.e. the flow is relatively simple), then low-image-density PIV, or particle tracking velocimetry (PTV) can be used. In PTV, it is possible to track an individual particle's movement because the distance between particles is larger than the displacement between frames. The tow tank, however, uses high-image-density PIV. Here, the displacement of particles between frames is larger than the average particle spacing. Statistical methods are used to identify the most probable particle match between frames. There are several packages and methods of doing this. We will be using PIVlab, an open source tool for PIV flow analysis in Matlab developed by Thielicke and Stamhuis [21]. To avoid using the software as a block box, the following section provides an overview of PIVlab's statistical methods for extracting flow information and evaluates its accuracy. The main image interrogation stages are **pre-processing**, **image evaluation**, and **post-processing**.

Pre-Processing. Even before any statistical analysis is applied to consecutive frames, the images can be enhanced to achieve the highest quality results. PIVlab can use three pre-processing techniques. The first is contrast limited adaptive histogram equalization (CLAHE). This increases the contrast in an image. The image is sectioned off into many tiles. The most frequent intensities in each tile are then distributed equally across the full range of data (0-255 for an 8-bit image)

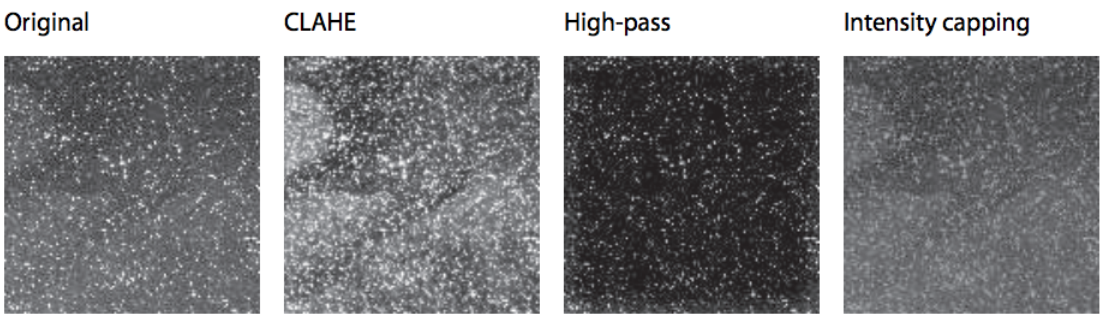


Figure 2.6: Effects of pre-processing techniques available in PIVlab [21]

so that the resulting distribution is a flat histogram. This process makes the resulting data more readable- information that was difficult to make out due to close contrast values are now spread out. The tiles are stitched back together through an interpolation process. This increases the probability of detecting valid displacement vectors by 4.7 ± 3.2% [21].

PIVlab can also apply an intensity highpass filter. Inhomogeneous lighting, possibly caused by inhomogeneous seeding, creates low frequency background information. Using a highpass filter suppresses this low frequency information, placing emphasis on the particle information.

Finally, intensity capping places an upper limit on intensity, and replaces all pixels that exceed this limit with the specified cap. This increases the probability of detecting valid displacement vectors by 5.2 ± 2.5% [21]. Figure 2.6 from Thielicke's paper on PIVlab visually shows the effect of each of these pre-processing technique.

Image Evaluation. In order to actually extract the velocity information, each image is split into smaller interrogation areas. Cross-correlation between a pair of interrogation areas from consecutive frames derives the most probable particle displacement in that time period. The discrete cross-correlation function is given by:

$$C(m,n) = \sum_{i} \sum_{j} A(i,j)B(i-m, j-n) \tag{2.10}$$

where A and B are the interrogation areas from the images. The function is measuring the agreement

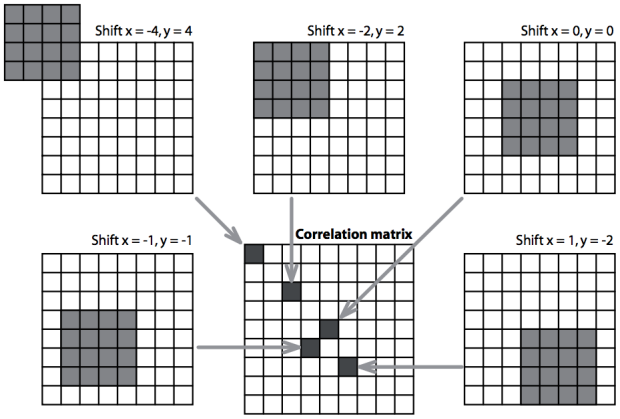


Figure 2.7: Example calculation of the correlation matrix as performed by PIVlab. The shaded interrogation window is correlated with the larger white window, producing the correlation matrix. Peaks in the correlation matrix correspond to statistically likely displacement vectors [21].

of the two interrogation areas for a given shift [18, 21]. The intensity peak in the correlation matrix C gives us the most probable particle displacement from A to B. Equation 2.9 can be solved in a couple of different ways.

The first way is to compute C directly in the spatial domain. Figure 2.7, taken from [21], shows a graphical representation of this computation. By choosing an interrogation area in B that is twice as large as the size of A, we reduce the chance that information is lost while still providing a reliable correlation matrix. Unfortunately, such a procedure is computationally expensive. Alternatively, we can compute C much quicker in the frequency domain using a discrete Fourier Transfrom (DFT). However, this approach is significantly less accurate: it requires using interrogation areas of equal size, meaning there is necessarily a loss of information for every particle displacement, among other inaccuracies. This requires various advanced techniques to reduce background noise and yield a reliable correlation matrix (see [21] for more). PIVlab uses both techniques, optimizing for both time and accuracy. Recent developments in GPU accelerated computing make direct cross-correlation increasingly fast, making it more and more possible to use.

After the correlation matrix is calculated, the integer pixel displacement is extracted from the intensity peak of C. PIVlab also makes use of various techniques to extract subpixel precision,

increasing the accuracy of the results.

Post-Processing. After velocity vectors have been calculated, it is standard to throw out outliers by manually setting maximum and minimum velocities. The missing vectors are then replaced by interpolated data and the data is smoothed through various statistical means. More centrally to the user, PIVlab can calculate integrals and derivatives (such as vorticity and divergence) easily. This allows for friendly data exploration.

Thankfully, PIVlab's image interrogation happens behind the scenes and we can use it to extract information from our experiments. Used in conjunction with our tow tank and PIV system, we have an end-to-end experimental fluid dynamics facility that allows us to quantitatively analyze fluid flow. Now it is necessary to test and finalize the tank to produce optimal PIV results.

Chapter 3

PIV SYSTEM

This chapter documents my work on two primary subsystems of the tow tank, the **controls system** and the **imaging system**. For a detailed overview of how to setup and use these systems, as well as how to conduct data analysis in PIVlab, consult the lab manual in Appendix B. The main components of the controls system are an Arduino and a variable frequency drive (VFD), which together drive the motor. By controlling the VFD with an Arduino, we are able to exchange its clunky interface for a custom GUI (graphical user interface) which better facilitates future scientific experimentation. The Arduino also allows for the addition of a range finder. This has several advantageous uses. In particular, we use this distance data to automate the stopping of the carriage making running the tank easier and safer, set up multi-speed tank runs from the GUI, and collect calibration data to correctly map a PWM signal from the Arduino to a frequency on the VFD to a velocity of the carriage. The controls system also includes limit switches as an emergency stop for the carriage, saving the mechanical stops for last-case scenarios.

The imaging system consists of a 532 nm 1 Watt diode pumped solid state laser and a camera. At the beginning of my thesis work, the specifics of this system were speculative. Several different cameras and setups were tested for in order to evaluate their potential applicability to our system. The cameras used were a GoPro Hero 4 Black, Nikon 1 J1, and Sony Alpha a7s. Video was captured from underneath the tank, the side of the tank, and in the case of the GoPro, submerged in the water. For bottom and side capture, an arm built by Tom Neumiller in his thesis was slightly modified and used. For submerged capture, the GoPro was attached to the PVC rod that was being imaged. A more sophisticated method of submerged capture is necessary to conduct further experiments. Based on data collected, the Sony Alpha a7s (or other camera with similar specifications) combined with submerged video capture is most suitable for the collection of PIV data.

3.1 Controls System: Hardware

The controls system consists of new hardware and software. An Arduino serves as the main hardware addition which feeds the VFD to drive the motor and handles the distance sensor and its associated functions. The following documents the circuitry and code used for this system and demonstrates their functionality. This should aid in future troubleshooting and debugging, and make additional features and changes easy.

3.1.1 Driving the VFD from an Arduino

At of the start of fall 2017, the tank was run exclusively from the VFD. The tow tank uses a Hitachi WJ200 Series Inverter. While it is more than capable of all of the functions necessary for the tow tank, it is limited by its 5-button interface and small LED display (figure 3.1). Labels for functions like 'forwards' and 'reverse' are hidden behind alphanumeric codes like 'F001' and 'F002', and changes in direction and speed involve multiple keystrokes. In order to move to an environment more conducive to experimentation, we control the VFD from an Arduino which will then be operated from a GUI. The Hitachi WJ200 manual is thorough in its documentation and can be consulted for most any questions. As a (very) quick start guide to using the VFD: toggle through the various settings using the ESC button, and press SET to view the available settings. Press SET again to submit any changes. Settings that were changed are documented in the following sections. Table B.1 in the lab manual consolidates all of this information, and the Hitachi manual can be consulted for further details.

As shown in figure 3.2, the front of the VFD has numerous 'intelligent terminals'. These can be programmed via the VFD to correspond to any of its various functions. In order to operate the drive externally, a 4-20 mA current needs to be inputted across OI (analog current input) and L (ground for analog signals). This current range will span the VFD's full range of 0-60 Hz. The 0-20 mA current loop used, adapted from industry standards, is shown in figure 3.4. A low pass filter serves as a digital to analog converter, smoothing the PWM signal sent from the Arduino (figure 3.3). The

Figure 3.1: The Hitachi WJ200 used to drive the tank motor

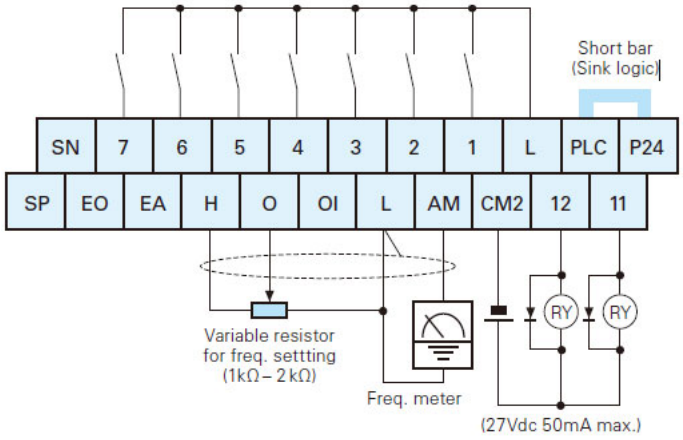
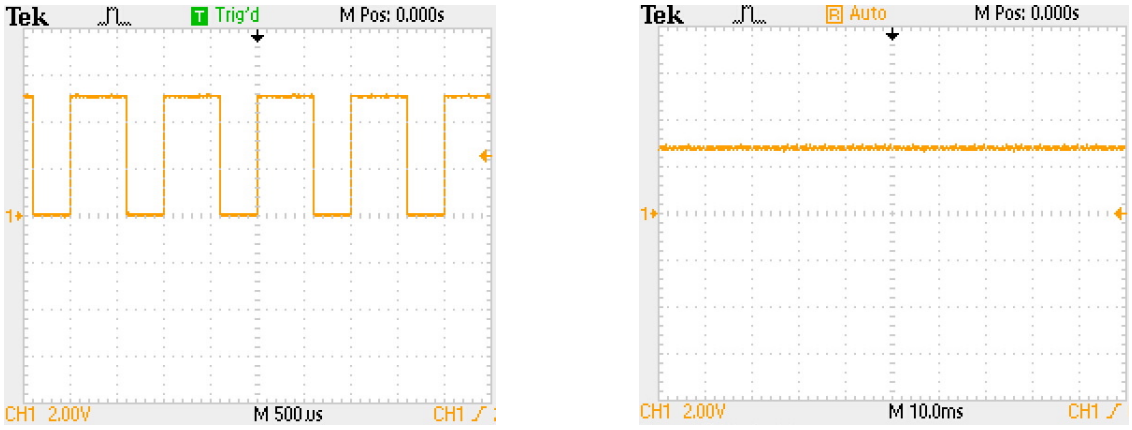


Figure 3.2: Diagram of terminals on face of VFD taken from the WJ200 manual. These are utilized to interface with an Arduino.

load current is equal to the input voltage divided by R2, plus the input impedance of the input on the VFD, which is 100Ω. Since the Arduino operates at 5 Volts, the output current is ensured to be from 0 to $\frac{5\,\mathrm{V}}{250\,\Omega} = 20\,\mathrm{mA}$. We thus have a voltage controlled current supply feeding the VFD . In order for the VFD to recognize the external current input, VFD parameter A001 was changed from 001 to 002.

A complication arises here because the Arduino's standard ATMega328 operates at 8-bits, which means that a PWM duty cycle can be anywhere from 0 (off) to 255 (on). However, the VFD can be set anywhere from 0-60 Hz with a step size of 0.1 Hz, resulting in 600 different frequency settings.



(a) Raw output from Arduino PWM pin at 50% duty cycle.

(b) A low pass filter smooths the PWM signal, yielding a stable voltage supply that can be digitally varied between 0 and 5 volts.

Figure 3.3: Effect of a low pass filter on PWM signal.

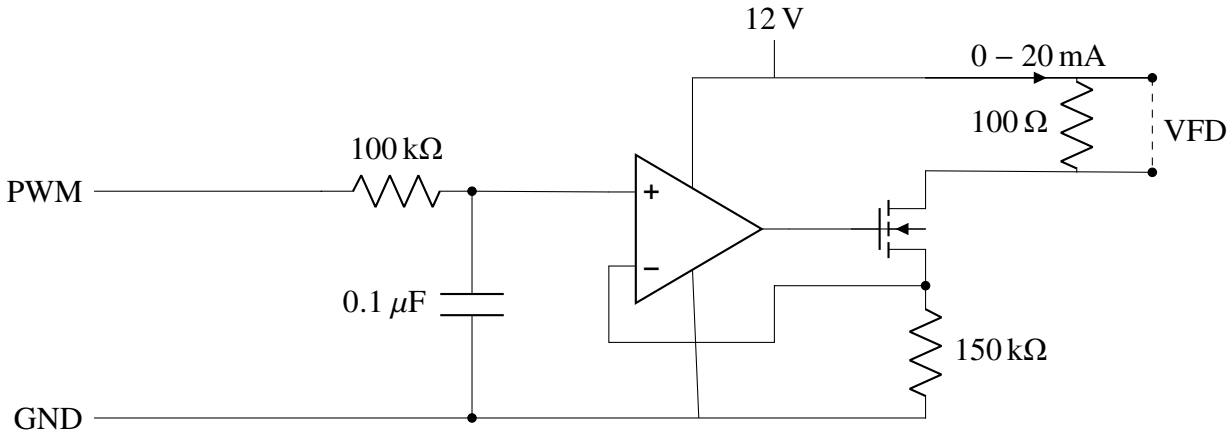


Figure 3.4: A 0-20 mA current loop inputted to the VFD and driven by a PWM pin of an Arduino.

This quantizes the possible frequencies and limits our choice of velocity. The problem can be avoided by using the 'Timer 1' library for the Arduino, which accesses the microcontroller's 16-bit timer [2]. This allows us to set a 10-bit PWM signal from 0 to 1023, which spans the VFD's entire range. The use of this library means that the standard 'analogWrite()' function cannot be used. This should not pose a problem for any additional work on the controls system, but should be noted to avoid future confusion.

3.1.2 Interfacing with Sonic Range Finder

As outlined earlier, information on the carriage's location at a given point in time would be useful to derive additional functionality. This is achieved by outfitting the Arduino with a simple ultrasonic proximity sensor. We use a cheap HC-SR04 which emits ultrasound at 40,000 Hz and computes distance by considering time of flight. It is reliable between 2-400 cm with an accuracy of about 3 mm, provided the temperature of the lab (and thus the speed of sound) is not changing significantly. Interfacing with the Arduino is straightforward and starter code is available online. Velocity data is calculated via the simple calculation, $\frac{\Delta d}{\Delta t}$. This assumes that the carriage is moving smoothly between measurements, which should be true so long as the belt is taut. For consistent measurements from the sensor, a slightly larger face than the carriage is necessary. A quick solution is to tape a sheet of cardboard to the carriage. As a final solution, it would be tidier to move the entire sensor system underneath the tank. By attaching the sensor to the tank frame using a bracket, the distance could be measured to the camera arm which extends underneath the tank. This would hide wires and sensors, as well as secure all components.

3.1.3 Calibrating Speed

Using the above sensor setup, velocities were recorded at various PWM signals. The frequency at which the motor is being driven can be displayed on the VFD by navigating to A001. The data collected is displayed in figure 3.5. The measurements agree with the raw data taken by Tom Neumiller, verifying the accuracy of the sensor data. The resulting linear fit is used to convert from user entered velocities to PWM signals. We see from the data, however, that there can be some error in this conversion which may become significant if future experiments wish to investigate phenomena at a very specific Reynolds number. In this case, a servo system would become relevant. This involves a feedback loop. We run the tank at a predicted velocity and compare the actual speed to the desired speed. If these numbers do not agree, we adjust the tank speed accordingly. While this thesis does not implement such a system, additional code could incorporate this functionality.

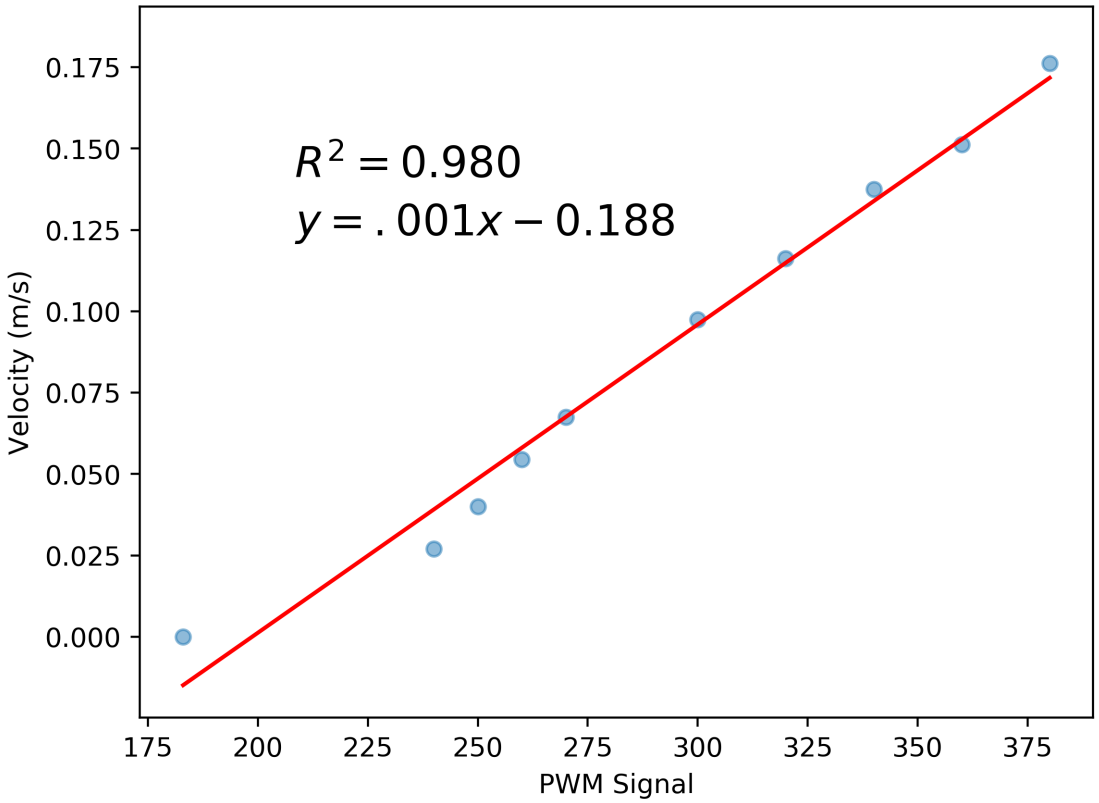


Figure 3.5: Tank Speed as a function of PWM signal. Data taken via Rangefinder matches hand-taken data.

3.1.4 Motor Direction

Without an Arduino, the direction of the motor is set by changing parameter F004 to either 00 for forwards or 01 for reverse. After setting this parameter, hitting run will start the motor. To set this instead from the Arduino, setting A002 was changed from 02 (the default) to 01. This disables the run button on the VFD and instead relies on terminal inputs to start and stop the motor. This was done by setting C001 to 00 and C002 to 01: this programs terminal input 1 to be forwards and 2 to be reverse. By connecting L to either of these inputs, the motor turns in the corresponding direction. Generally, mechanical switches are used to set the direction. In order to include direction in the GUI, a relay is used. The outputs of the relay terminals are connected to inputs 1, 2, and L. The relay is actuated by sending a 5V signal from the Arduino to the input pin of the relay. By

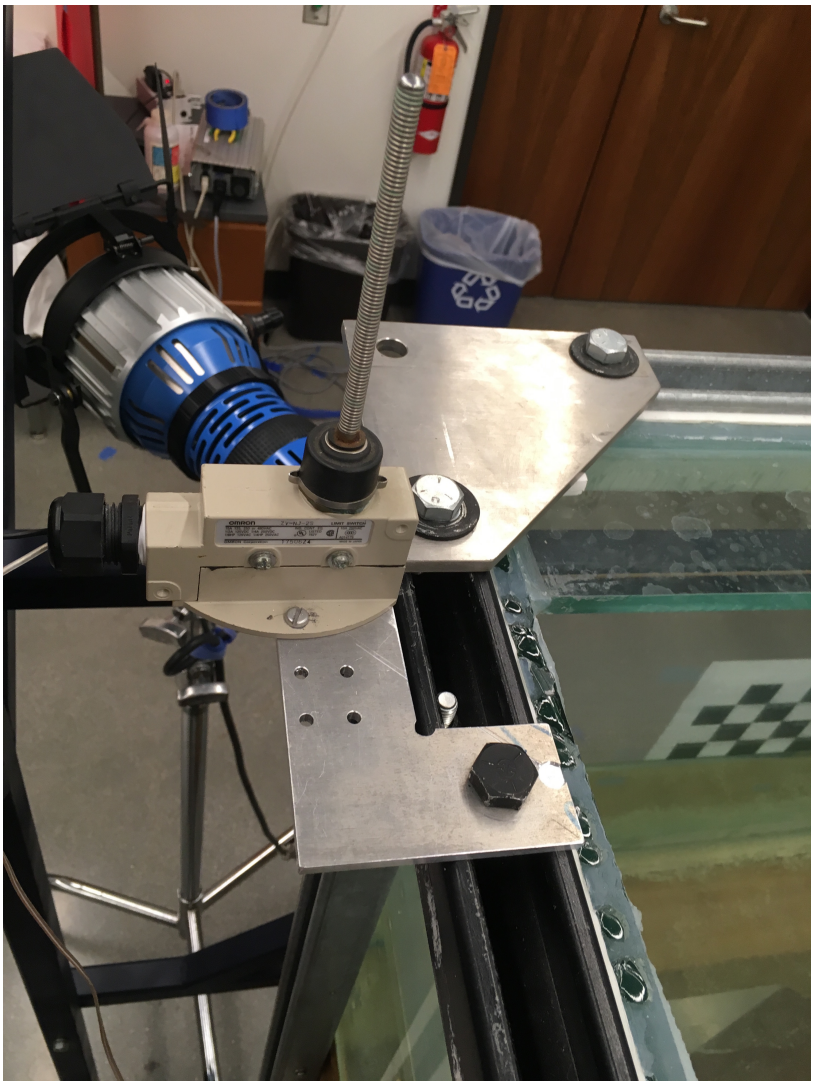


Figure 3.6: One of two limit switches mounted on either end of the tank. On contact, the motor will begin braking.

setting the pin high or low, the direction is set to forwards or reverse respectively.

3.1.5 Limit Switches

To ensure safety, two limit switches are positioned just before the mechanical stops on the tank, one at each end. Thanks to Hardy, the limit switches slide along the tank rails so that they can be moved to wherever they will function best (figure 3.6). The limit switches are each connected between the L terminal of the VFD and inputs 3 and 4. To configure inputs 3 and 4 to be brakes, we set C003 and C004 to 007. The switches are designed to be normally closed- by default there is

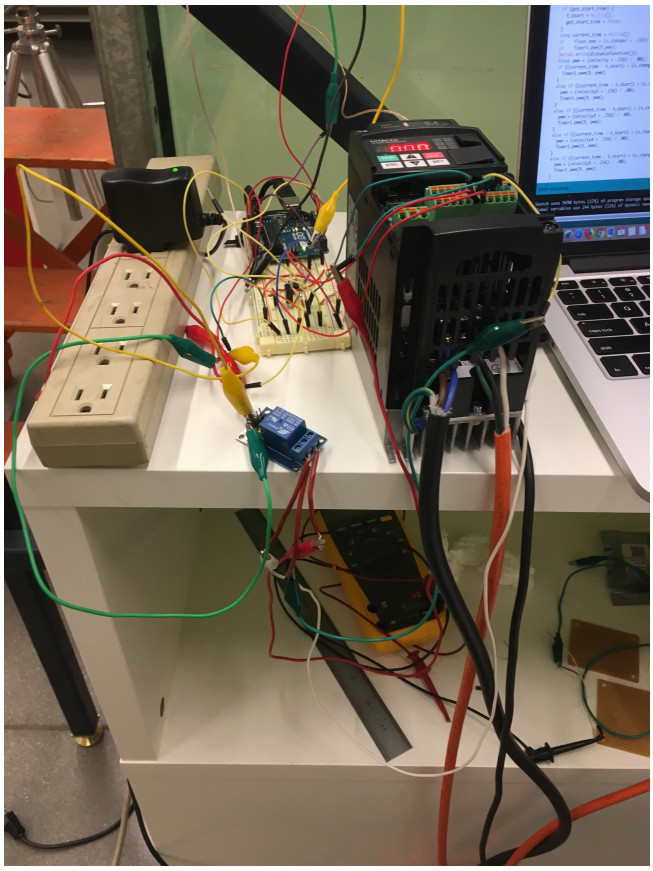


Figure 3.7: Prototyping the circuitry for the controls system.

current running through them. If the carriage hits the switch, the loop opens and the motor stops. This ensures a safe design. If any of the wires for the switches fall out of place, the tank will not run. It is noted that after an accidental collision with the tank running at full speed, the mechanical stops are proven to work well, although the carriage did need to be realigned.

3.2 Finalizing the Hardware

As of the final draft of this thesis, I am in conversation with Tony to make a box for the electronics and possibly a PCB for the circuit. The Uno used for prototyping will also be exchanged for a more space efficient Nano. This finalized version will likely be made and installed after my graduation, along with a permanent monitor to use rather than a personal computer. The circuitry as described above is completely functional and was used for all experiments in this thesis; figures 3.7 and 3.8 show the system in its current iteration.

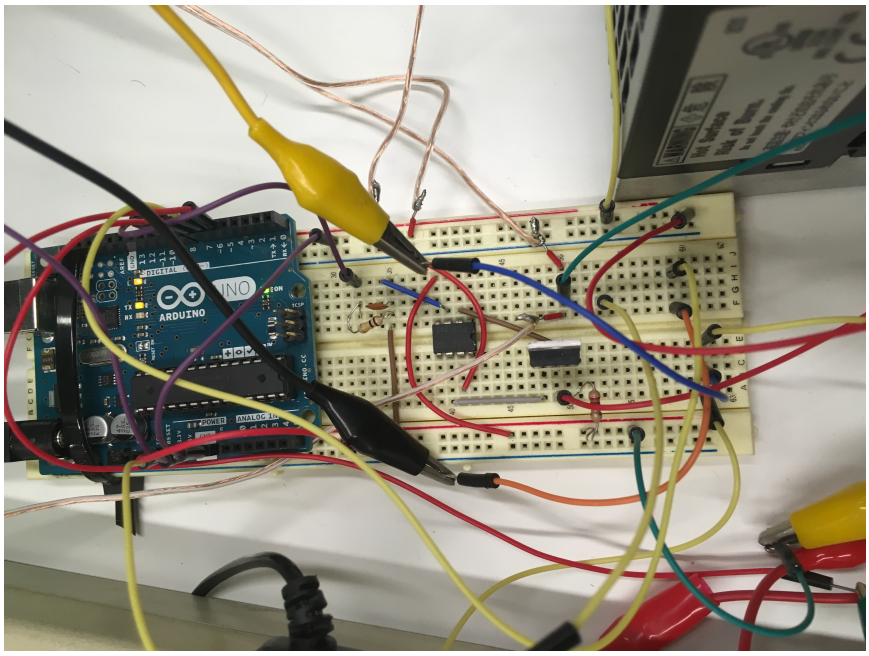
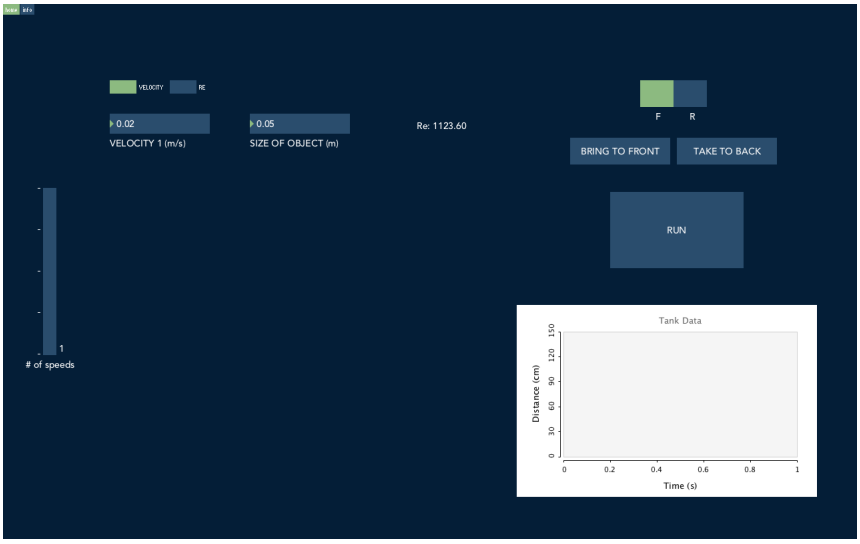


Figure 3.8: A zoomed in look at the controls system wiring.

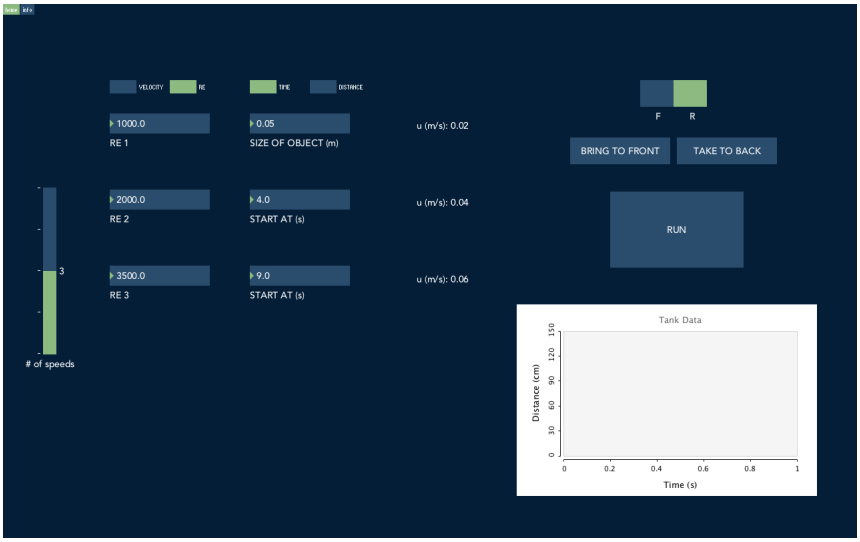
3.3 Controls System: Software

With the hardware in place, we wish to build a GUI to operate the tank from. This is done in Processing, a Java-based programming language for visual design. Processing is the parent project of Arduino and the two have similar development environments and syntax [17]. This makes communication between the two straightforward. Moreover, since Arduino is taught in Pomona College's Electronics class, using Processing with the tank should make it straightforward for future students to change and extend the current GUI.

The tank GUI relies heavily on a few different libraries. *ControlP5* is a visually appealing GUI library for processing with extensive online documentation. Graphs are presented using *graficas*, and information is sent and received via serial communication. The sections below present the design of the GUI and demonstrate its functionality. While some technical details will be discussed, the code is the best place to learn about how the GUI works. The commented Processing and Arduino code can be found on Blair Subbaraman's github account (https://github.com/bsubbaraman/pomona_towtank). This makes future changes, version control,



(a) Setting up a single-speed run by velocity from the GUI.



(b) Setting up a multi-speed run by Reynolds number.

Figure 3.9: Demonstrating GUI's adaptive behaviour, making setting up experiments easy.

and code sharing within the lab straightforward.

3.3.1 Design

Figures 3.9 and 3.10 display the home screen of the GUI. The functions of the numbered elements in figure 3.10 are described below. The layout corresponds to the chronological steps of setting up a run. It is built to accommodate up to five different speeds in the course of a single run- this number is easily scalable in the code, but suffices for now given the length of the tank. The user

can enter a velocity directly or, by providing the characteristic length of the object being towed, a Reynolds number. Given that dynamical similarity (see section 2.1.3) requires us to match a given phenomenon's Reynolds number, this is often an easier way to parameterize an experiment. Subsequent velocity changes can be cued by either a start time or start distance, again giving a choice depending on what is easier for an experiment. A series of buttons compose a control panel that allows the user to set the direction of the motor, run the tank given the entered parameters, and to move the carriage to either end of the tank for convenience. The position sensor is used to build in a few safety precautions when setting up a run: it will not allow the user to run the tank in reverse if it is already in the front, nor vice versa. It also automates the stopping of the tank. This is particularly helpful, since all experiments are conducted in the dark. The GUI is built to be dynamic, changing based on user inputs. Figure 3.9 demonstrates this behaviour. The GUI does not currently give any control over the acceleration of the motor. The acceleration time can be changed manually on the VFD, see the lab manual in appendix B for specifics. For now, we set the motor to accelerate and decelerate as quickly as possible. However, we do print a csv file of distance over time into the Processing data folder, located in the same directory as the sketch. This can be used to determine at what point velocities become constant, which can be used in later data analysis.

3.3.2 Overview

The following describes each numbered element in figure 3.8. Most components should be self-explanatory.

1. A slider sets the number of different velocities to set within a single run of the tank.
2. Choose to set up the run by velocity or Reynolds number.
3. Enter the n-th speed (entries for $n > 1$ are optional).
4. Choose to cue changes at a given time or distance.
5. Enter the characteristic length of the object being towed, used for Reynolds number calculation.

Figure 3.10: The main screen of the Tank GUI. The function of numbered components are explained in 3.2.2.

6. The time (or distance) at which the speed changes
7. The Reynolds number of the run if velocities are given. The velocity of the run if Reynolds numbers are given.
8. Set the direction of the motor. The side highlighted green is active. Forwards moves away from the controls center and towards the door of the lab.
9. Bring the carriage to the front (controls center) or to the back (door).
10. Begin the run given the entered parameters.
11. A plot of tank position over time.
12. Toggle between the home screen and the info screen.

3.3.3 Brief Comments on the GUI Code

The Processing sketch is relatively bulky at close to 700 lines, mostly due to the visual details and manipulation of the various components. The components from ControlP5 are documented well and the examples clearly demonstrate how to use them. The core functionality of the Arduino is consolidated into about 200 lines of code. This difficulty encountered in this code is communication.

Over serial communication, one byte of data can be sent or received at a time. The resulting problem is one of coordination- Processing must send all of it's information to Arduino, and Arduino must know how much information it is going to receive. To accommodate for this, data is only sent when components 9 or 10 in figure 3.8 are clicked. Each of these correspond to a different loop which the Arduino will enter until the task is complete.

3.4 Imaging System

The various cameras and experimental setups used are presented below, along with relative advantages and disadvantages of each. These qualitative comparisons will be made quantitative in Chapter 4, leading to a more informed discussion concerning final design decisions.

3.4.1 Orientation and Problems

As discussed in Chapter two, the laser light-sheet and optics must be set up orthogonal to one another. The tank was intended to have a light-sheet parallel to the ground with an arm underneath to which a camera can be mounted. However, it was discovered that many particles tend settle to the bottom of the tank over time. This particle collection can cloud the video taken looking up through the floor of the tank. Two methods were attempted to avoid this problem. Imaging from above the tank, looking down into the water, presents one alternative. With the current cameras, this is only possible with the GoPro, as it is able to be submerged underwater. For other cameras, a glass window is necessary to avoid imaging through the ripples on the surface of the water. Possible designs will be presented in Chapter 5. A second alternative is to rotate the light-sheet such that it is perpendicular to the floor, and capture video from the side of the tank. However, a problem with the laser is encountered: figure 3.11 shows the laser light before entering the light sheet optics. It is slightly multi-lobed. This results in the laser light attenuating more quickly than it should. Moreover, despite the Powell lens, the light is more intense on either side of the light sheet than in the middle (figure 3.12). For a light-sheet parallel to the ground this does not pose a large problem as the light is intense enough to illuminate the tracer particles in the proximal half

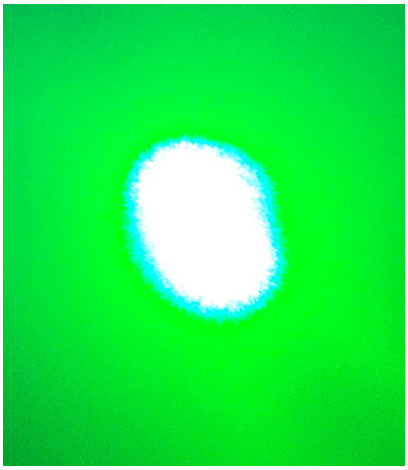


Figure 3.11: A zoomed in picture of the laser immediately prior to entering the Powell lens. Rather than a single beam of light, it appears to have two lobes.

of the tank. However when oriented perpendicular to the ground, not nearly enough of the tank is illuminated for PIV experiments. Upon fixing this problem with the laser, this rotated setup would be interesting to further explore. For now, we are restricted to a light-sheet parallel to the ground.

3.4.2 Laser

The Fluids lab has a 532nm 1 Watt diode pumped solid state laser. This laser was used for all experimentation. As discussed in 3.4.1, it was slightly malfunctional for the duration of this thesis. Figure 3.12 shows the basic setup of the laser, with the light-sheet oriented parallel to the ground. To maximize power, the laser was placed approximately 2 feet away from the side of the tank, with the light-sheet illuminating approximately 50% of the water. Results were best with a greater laser height: due to the particle collection on the bottom of the tank, a higher light-sheet enabled the camera to focus on just the illuminated particles. Trials used a laser height of 46 inches, which is 8.5 inches above the bottom of the water.

3.4.3 Camera

PIV experiments necessitate high frame rate video in a low-light environment. Generally when imaging in the dark, we decrease shutter speed to let more light in. As discussed in Chapter 2, we

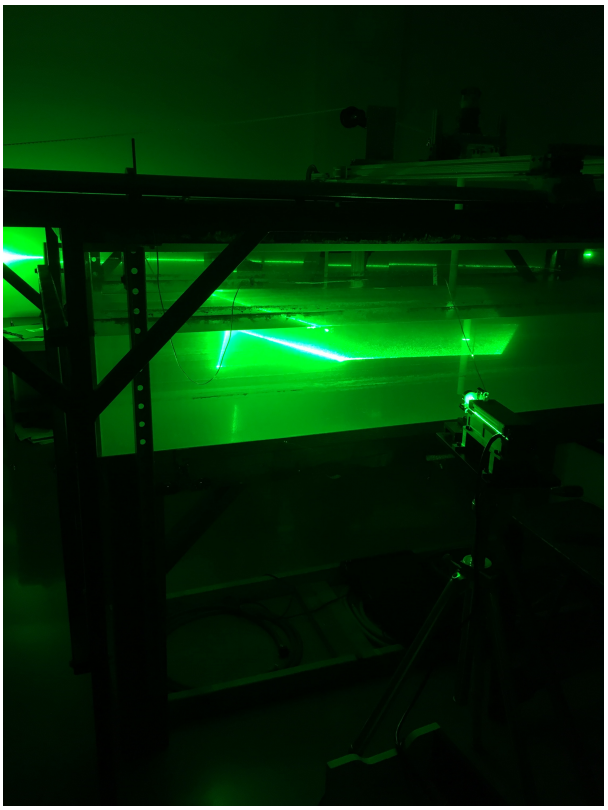


Figure 3.12: Basic setup for PIV runs. Note the non-uniformity in light despite the use of a Powell lens.

cannot do this due to the resulting streaking of our images. The solution to this problem is to find a camera with a high ISO. At a higher ISO the camera's sensor will absorb light faster, making filming in the dark possible. A high ISO also comes at the expense of higher image noise (or grain) and cost [8]. The three different cameras tested present various negotiations of these requirements. Qualitative image quality under PIV conditions of these cameras are presented and compared here.

GoPro Hero 4 Black. The GoPro is the most frugal option and is already owned by the Fluids Lab. It is able to film 1080p video at 120 frames per second, making it the camera with the highest frame rate tested for this thesis. Moreover, it is designed for underwater video, making it an ideal candidate to capture PIV video from above the light-sheet, submerged in the water. If filming in 'wide' mode, the GoPro's lens results in a barrel distortion that must be un-distorted in Matlab; this process is time-consuming, taking about 2 hours for 5 seconds of video. It also has no manual focus. It has a fixed autofocus distance of infinity, with experimentation suggesting that anything

greater than about 8 inches away is in focus. This poses problems for imaging since the illuminated particles are not the focus of the video capture. Its maximum ISO is 6400.

Nikon 1 J1. The Nikon - also already owned by the lab - presents a more sophisticated alternative to the GoPro. In particular, it is capable of manual focus which allows us to focus on the illuminated particles. This feature is not as sensitive as is desired, particularly when compared with the Sony below, but is certainly a plus. Images can only be captured at 60 frames per second in 720p. These numbers are too low for laboratory experiments, but the Nikon still serves as a test that can inform the purchase of other reasonably priced options with manual focus. It's maximum ISO of 6400 matches that of the GoPro.

Sony Alpha a7s. At $2,000, the a7s is the most sophisticated camera tested. Thanks to Professor Choi, we were able to experiment with this camera for free. Video was captured at 60 fps in 1080p. Its manual focus is more sensitive than that of the Nikon resulting in crisper video data. The highlight of the a7s, however, is its ISO of up to 409600 with ultra low noise [4]. This makes it ideal for PIV experiments if a higher frame rate could be achieved. In fact, some online resources show that the Sony is capable of shooting at 120 fps if configured with a large enough memory card. Unfortunately, this was not tested in this thesis.

3.4.4 Resulting Images

Snapshots from the video data of trials from each setup are shown in figure 3.13. Qualitatively, the Sony (b) performs the best. The submerged GoPro data (d) has the second best texture- a quantitive analysis in the next chapter will provide an interesting comparison of these two cameras. The Nikon has clear images (a), but it's lower ISO is readily apparent compared with the Sony. It will take a comparison within PIVlab to find out if these qualitative differences in video correspond to quantitative differences in PIV analysis quality.

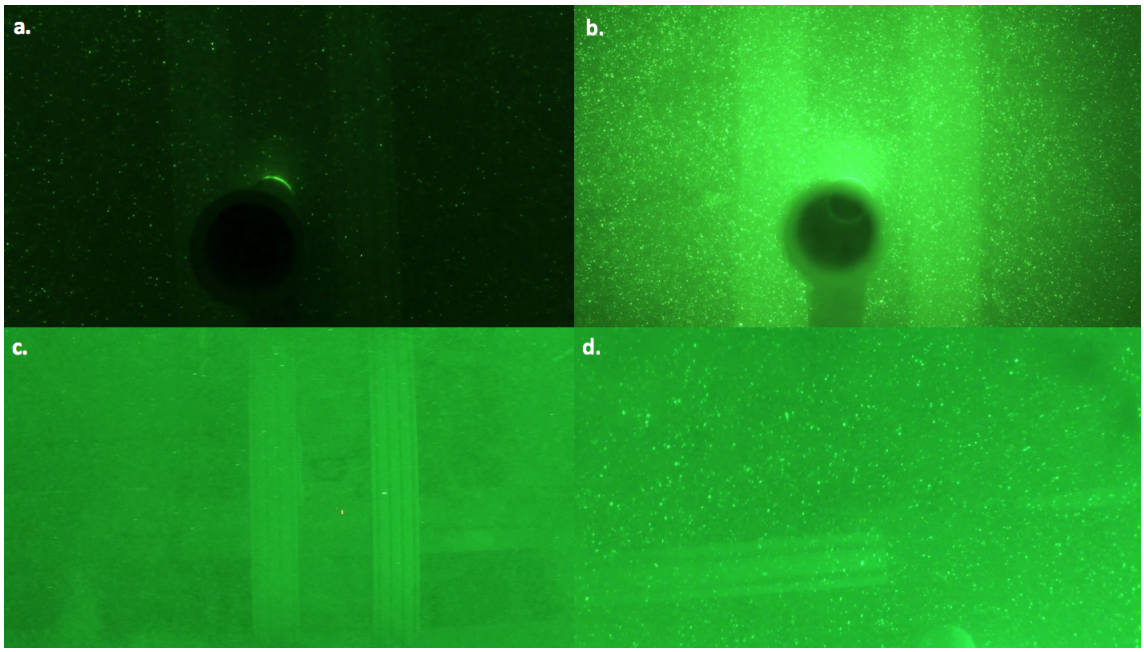


Figure 3.13: Snapshots of video images from various setups: a. Nikon 1 J1 imaged from underneath the tank at 60 fps. b. Sony Alpha a7s imaged from underneath the tank at 60 fps. c. GoPro Hero 4 Black imaged from underneath the tank at 120 fps. d. GoPro Hero 4 Black submerged in the tank, imaged from above at 120 fps.

Chapter 4

PIV ANALYSIS

Now that we have test videos for the different cameras, we can use PIVlab to test the relative improvements that various changes make. This will inform the final setup and camera that we use and make sure we optimize our PIV setup as much as possible within a budget. For a detailed walkthrough of how to collect PIV data and navigate the PIVlab interface, please see Appendix B. Section 4.1 provides a brief overview.

4.1 An Introduction to PIVlab

After taking a video, we have to extract its frames to use PIVlab. The Matlab script in Appendix A, adapted from Jonah Grubb's thesis, accomplishes this. When using the GoPro in wide mode, it is necessary to undistort the video as well; the steps for this are documented in Jonah Grubb's thesis. The GoPro was used mostly used in narrow mode for these tests, eliminating the need for this process. It is important to save the frames from the video as a PNG. Saving as a JPG introduces noise to the image. Note that running this script will create a new folder named 'frames' in the same directory as your video.

The frames can now be imported to PIVlab. The interface can be opened by typing 'PIVlab_GUI' in the command window. Clicking 'Load Images', adding your relevant frames, and then adding and importing them will bring these images into PIVlab. Note that the frame count in the bottom right hand side will be half the number of frames imported since PIVlab uses the second of every pair of frames for its cross-correlation and thus hides them. In the following sections, I added to the 'piv_FFTmulti.m' routine within PIVlab to extract and manipulate correlation matrices. Should this information be relevant, you must add the code found in Appendix A.3 to the appropriate area and run 'PIVlab_GUI.m' directly instead of using the command line prompt.

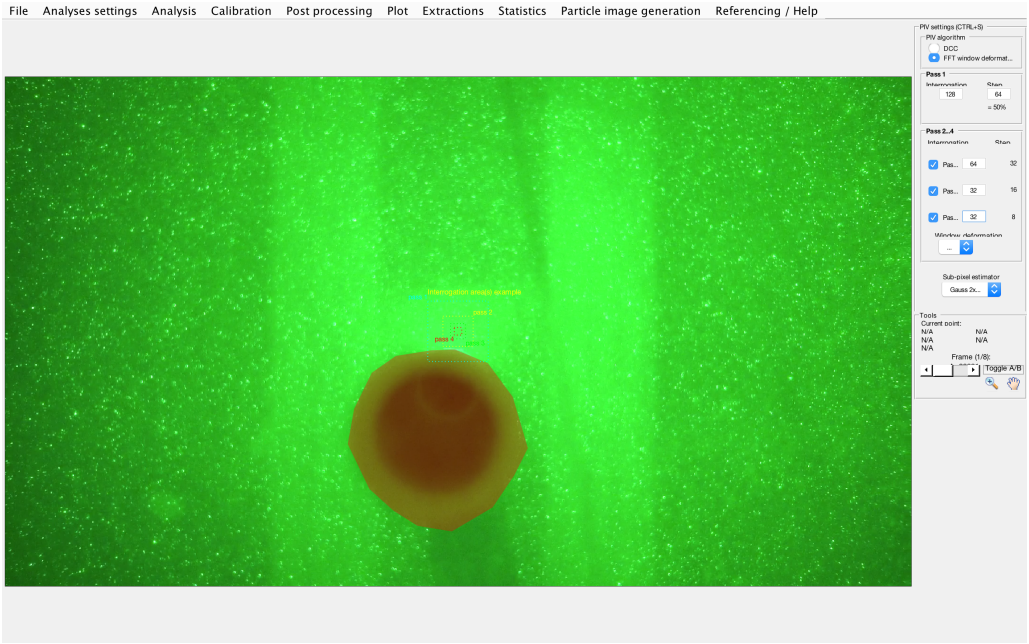


Figure 4.1: Applying a mask and setting interrogation window sizes. A mask should be applied to any object visible in video data to avoid erroneous vectors caused by PIVlab attempting to correlate these points. The dotted boxes show the size of interrogation window as specified on the right hand side. Four passes will result in the best quality analysis. Note that in a pass of 32 px, there will be $\approx 7 - 10$ particles visible

Once the frames are in PIVlab, each of the sections under 'Analysis settings' need to be set up. Under 'Exclusions', we can draw masks and set regions of interest. Masks are applied to parts of the image which should be ignored. For example, in figure 4.1, the cylinder itself should not be analyzed. The 'Draw mask(s) for current frame' function can be used for this. Since the cylinder will be in the same location in all frames due to our co-moving reference frame, select 'Apply current mask(s) to selected frames' to recreate this mask for all of our other images. In the Image Pre-processing section, the various image enhancements presented in 2.2.4 can be applied. Optimal settings will be discussed in 4.2.1. Finally under PIV settings, the number of passes as well as interrogation window size can be set. In general, more passes will produce a better analysis but also take longer. For 80 frame pairs (160 total frames, or about 3 seconds of footage at 60 fps), a four pass analysis takes about 20 minutes on the laptops available in the Pomona College Physics lounge. Using four passes is thus reasonable and makes for the best analysis. For frames of similar quality to figure 4.1, it is best to start with a large interrogation area of 128 pixels as it is quite noisy.

This will extract the large-scale structure of the flow. We gradually decrease the interrogation size keeping 50% overlap between windows as suggested by Thielicke [21]. For best results, there should be at least 7-10 particles in a given interrogation window, implying a minimum interrogation size of 32 pixels for the image above [14]. We thus have an interrogation size of 32 pixels for each of the last two passes. Figure 4.1 also shows the sizes of these windows for a sense of scale.

4.2 Signal-to-Noise Analysis

A first goal is to quantify the differences in video quality presented in Chapter 3. This can be done by calculating the signal-to-noise (SNR) of the cross-correlation. Recall from Chapter 2 that PIVlab computes a cross correlation matrix for each interrogation area in the frame, and the peak of this matrix corresponds to the most likely displacement vector of a particle. One measure of determining the SNR is the primary peak ratio (PPR). This is the ratio of the peak of the correlation matrix to the second highest peak. A measurement is considered valid if the PPR is above a user specified threshold: values above 1.2 are generally considered valid, and ratios around 2 can reliably avoid spurious vectors [24].

Unfortunately, PIVlab does not save the correlation matrices by default. Remember that we can specify the number of passes and the window size for each pass in our analysis. For each pass, PIVlab computes an $n \times n$ matrix, where n is the given interrogation size. It then stacks each correlation matrix, resulting in a three-dimensional matrix of size $n \times n \times m$ where m is the number of interrogation areas it takes to cover the region of interest. PIVlab repeats this p times for each pass as configured in the PIV settings, ultimately yielding p $n \times n \times m$ matrices for each pair of frames. These can all be accessed in the 'SUBPIXGAUSS' function of 'piv_FFTmulti.m'. To quantify the difference between our videos, we can find the mean and standard deviation of the PPRs of the correlation matrices in a given pass. This assumes that the quality of any two given frames is the same; we can confirm this by running the analysis on several sets of frames within the same video and compare the results. The modifications made to the PIVlab code to accomplish this analysis are in Appendix A.3

4.2.1 Optimizing PIV parameters

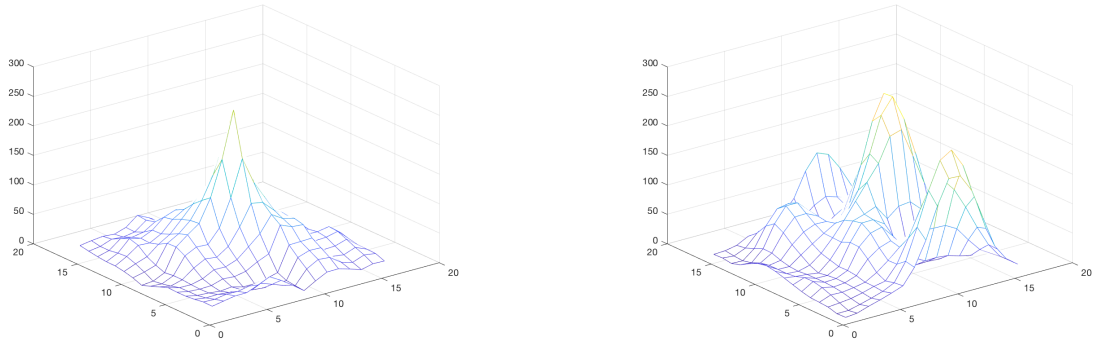
Using the SNR analysis described above, we can optimize our PIV pre-processing techniques and get an idea for the quality of our video data. Table 4.1 gives the average PPR, μ, and standard deviation, σ, for each pass on two consecutive frames taken with the Sony Alpha a7s at 60 fps. The video is of a rod travelling at 2 centimeters per second through the tank. The first pass, with an interrogation window of 128 pixels, unsurprisingly has the worst PPR, since the scale of the features of this flow field is much smaller than this and the particle density is relatively high (see figure 4.1). Using all of the pre-processing techniques results in the best quality per this metric; using only CLAHE and a highpass filter yields very slightly better mean PPRs but a worse standard deviation. Applying only a highpass filter produces the highest average PPRs, but has a similarly large standard deviation. Data is shown down to an interrogation window size of 16 pixels. We see that the standard deviation at this small a window is substantially higher.

	Pass 1 (128 px)		Pass 2 (64 px)		Pass 3 (32 px)		Pass 4 (16 px)	
Pre-Processing Technique	μ	σ	μ	σ	μ	σ	μ	σ
CLAHE (default)	1.08	.0590	1.15	0.0676	1.16	0.0761	1.20	0.126
Highpass	1.11	0.0904	1.21	0.101	1.25	0.144	1.32	0.321
Intensity Capping	1.06	0.0501	1.14	0.0759	1.17	0.0895	1.22	0.162
(1) + (2)	1.10	0.0757	1.18	0.0802	1.20	0.0943	1.25	0.187
(1) + (2) + (3)	1.11	0.0759	1.184	0.0787	1.20	0.0902	1.24	0.178

Table 4.1: Comparison of pre-processing techniques on frames 1 and 2 of a video taken with the Sony Alpha a7s at 60 fps. μ is the average PPR of the interrogation windows and σ is the standard deviation.

4.2.2 Comparing the Quality of Video Data

Unfortunately, this video is noisier than we would hope. This is likely due to the low frame rate of 60 fps. While we are operating under the maximum velocity limit of the Sony as calculated in section 2.2.3, meaning we will not have issues with streaking in images, the lower frame rate means that between two frames the particles have travelled a significant distance relative to their size. Since we have a high density of particles, this results in noise in the cross-correlation. We can



(a) Noise in Sony data at 60 fps. The peak is narrow, and there are few other relative maximums around it.

(b) Noise in Nikon at 60 fps. The peak is wide, with several other local peaks.

Figure 4.2: Difference in noise between cameras, visualized through surface plots of random correlation matrices from the fourth pass of PIV analysis (interrogation window = 16px).

extend this analysis by applying the PPR method to all of the cameras used. This data is presented below in table 4.2.

	Pass 1 (128 px)		Pass 2 (64 px)		Pass 3 (32 px)		Pass 4 (16 px)	
Camera	μ	σ	μ	σ	μ	σ	μ	σ
Sony (60 fps)	1.11	0.0759	1.18	0.0787	1.120	0.0902	1.24	0.178
Sony (24 fps)	1.05	0.0321	1.07	0.0366	1.08	0.0512	1.10	0.109
Nikon (60 fps)	1.05	0.0303	1.08	0.0359	1.09	0.0634	1.12	0.126
GoPro (120 fps, submerged)	1.04	0.0283	1.04	0.0317	1.05	0.0496	NaN	NaN

Table 4.2: SNR comparison of each camera tested. Analyzed in PIVlab with all pre-processing techniques applied, as was determined optimal in 4.2.1

We see that the Sony at 60 fps offers significant SNR improvements over the Nikon at the same frame rate, which matches our qualitative intuitions. Moreover, the quality of the Sony at 24 fps was significantly worse than at 60 fps. This leads us to believe that given a high enough frame rate, the average PPR should increase even more such that even one standard deviation below the mean is still above the validation threshold of 1.2. For the submerged GoPro, there were not enough particles visible to compute all values with an interrogation size of 16 px. The SNR is visualized in figure 4.2 for further qualitative comparison. While the PPR method compares the two highest peak, it could be that there are many peaks similar in magnitude, which the PPR would not illustrate. Thus, figure 4.2 shows surface plots of randomly chosen correlation matrices, all from the fourth pass

(interrogation window = 16 px) of analysis. Here we visually see the difference in noise between the Nikon and Sony.

4.3 Drag Analysis

The videos taken thus far have been for test purposes and are not of experimental quality. However, we can still use PIVlab to derive relevant data from them. In general, we are interested in the body forces on an object. Almost always we will be interested in drag. It is not trivial to derive drag information from PIV data. This section consolidates research on the *control volume method*, or wake survey method, for drag estimation. We then apply this approach to our video, resulting in an estimation of the drag coefficient of a cylinder.

4.3.1 Control Volume Method

The control volume method offers an approach to solve for drag indirectly. It relies on conservation of x-momentum over a designated volume as shown in figure 4.3.

The momentum difference is related to both the velocity difference and static pressure difference over the volume. However, we do not have any pressure data; in fact, the motivation for PIV was to avoid invasive pressure sensors that may disrupt flow. If we extend the control volume far enough downstream, we can assume that the flow is parallel and the pressure difference is zero. This becomes tricky experimentally. As we move farther downstream, the wake becomes more difficult to detect as it spreads in the y-direction. Moreover, contributions from walls of the tank become important as we look at a wider region. We wish to stay close enough to the wake such that boundary layer effects remain negligible. This problem requires analysis to identify a 'goldilocks control volume' which is a proper negotiation of these possible errors. Others have found that examining the flow 8-16 diameters downstream yields accurate results for an object moving at $\approx 1 \frac{\text{m}}{\text{s}}$. Again, the correct control volume will depend on the experimental setup in question.

We assume incompressible and steady two-dimensional flow. In the case of turbulent flows, time

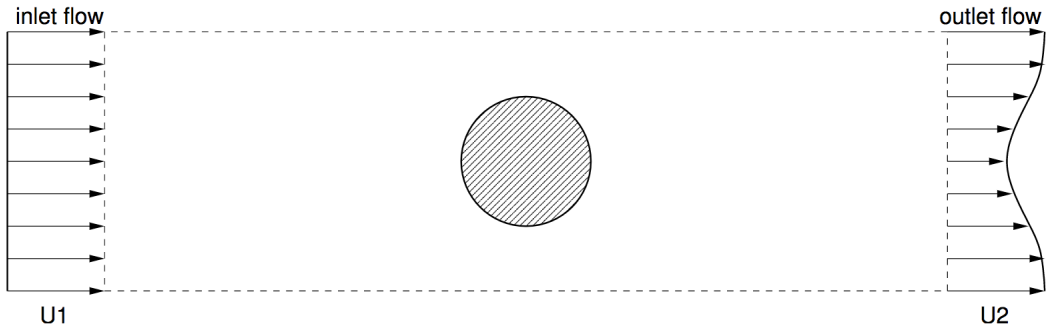


Figure 4.3: By applying conservation of momentum to a control volume, we can solve for drag using PIV velocity measurements [20].

average quantities can be used [19]. White derives a formula for the drag coefficient based on this geometry using conservation of mass and conservation of momentum [23]. It is presented here:

$$C_D = \frac{2\rho}{U^2 L} \int_2 u_2(y)(U - u_2(y))dy \tag{4.1}$$

where the subscript 2 corresponds the the outlet in figure 4.3, U is the inlet velocity, and L is the projected length of the object normal to the flow direction. Using equation 4.1, we can estimate the drag coefficient for our cylinder.

4.3.2 Rod Analysis

We can now make an estimate of the drag coefficient of a cylinder. Since this is a known result, we can use this to calibrate and refine our experimental use of the control volume method. We will use the Sony video for the analysis, running a PIV analysis of 4 passes as described in 4.1. We can find the average vectors for all frames and then derive the u-velocity along any poly-line. Several such lines of various lengths are chosen at varying distances downstream from the cylinder. Note that, as shown in figure 4.4, the current imaging setup extends ≈ 2 cylinder diameters downstream. Figure 4.4 also shows one example of a downstream poly-line over which the u-component of velocity is derived.

We can create a similar such line to confirm the free stream (inlet) velocity, which should equal the

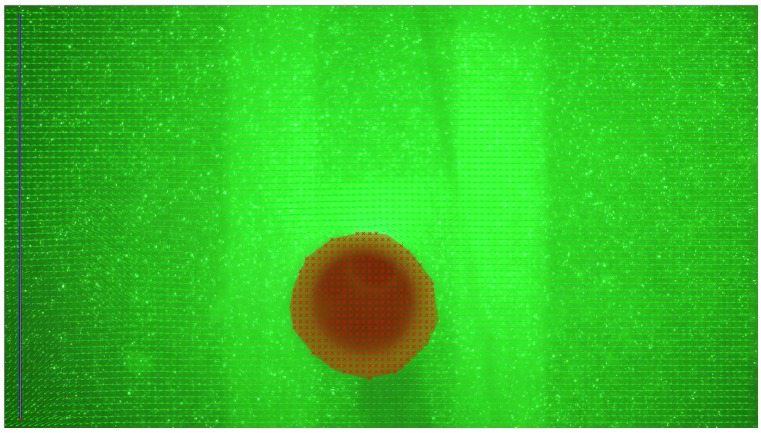


Figure 4.4: Example of a poly-line which defines the outlet of our control volume. Note also that the current imaging setup gives us approximately 2 cylinder diameters of downstream flow to work with.

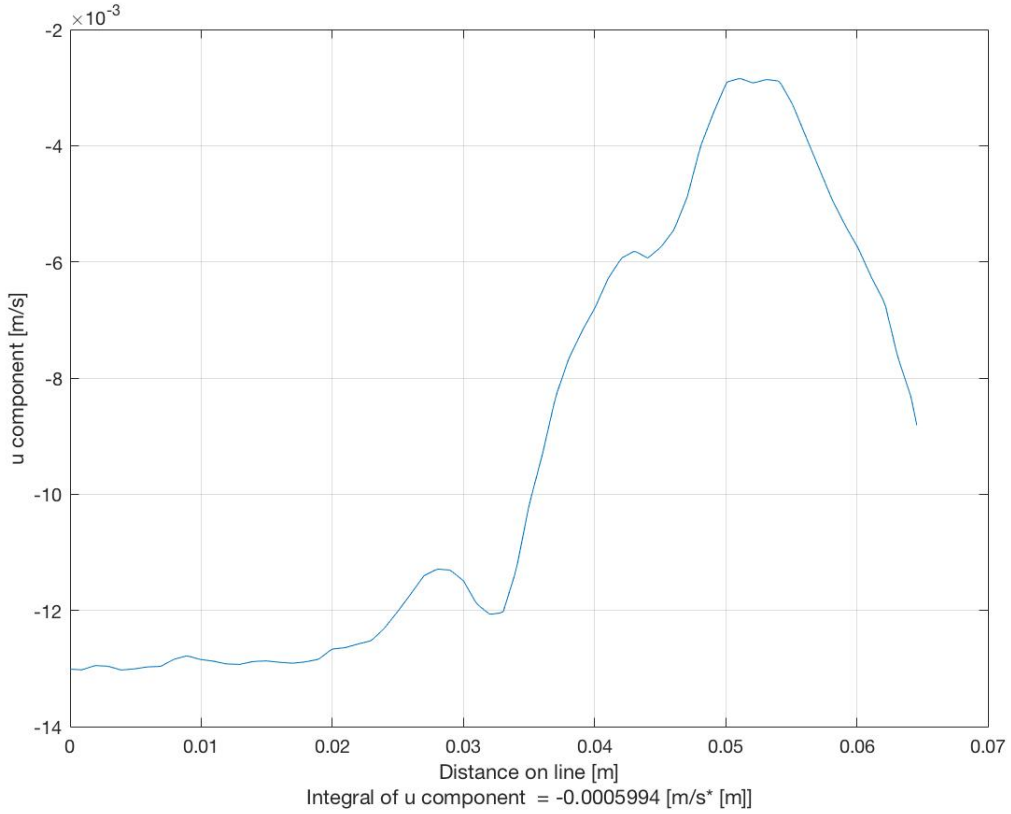


Figure 4.5: u component of velocity as a function of distance along a poly-line. We see velocity decrease immediately behind the rod as expected.

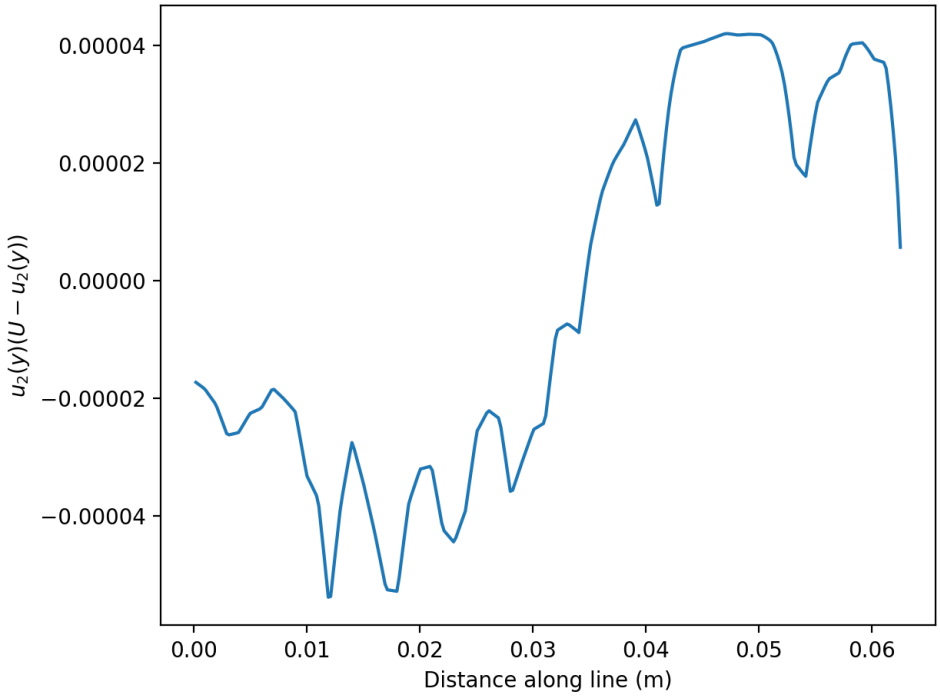


Figure 4.6: Plotting the integrand of equation 4.1 for this data. The most significant contributions are from within the wake of the cylinder. As the curve has not reached 0 once again, we can be confident that the image in figure 4.4 does not contain the entire wake. We also see a strong dependency on U due to the slow velocity. We would hope for $U - u_2(y)$ to be 0 while our line is not in the cylinder wake. Instead we see negative contributions.

velocity at which the tank was run. In this case, $U = -0.01\ \frac{\text{m}}{\text{s}}$. Figure 4.5 shows a PIVlab generated plot showing this data as a function of distance along the line. We see that the velocity decreases as it enters the projected area of the cylinder, and then increases back to the free stream velocity. We would like enough space above and below the cylinder such that we see the velocity reach the free stream velocity again- the video here does not allow for this. This introduces definite error to our drag coefficient calculation, as the wake almost certainly extends further into the y-direction. We consequently expect an underestimate of the drag coefficient. While PIVlab calculates the integral over this line for us, we wish to compute the integral expressed in equation 4.1. To do so, we can export the data as an ASCII chart. A short python script documented in Appendix A.4 reads in this file and calculates the relevant integral. Table 4.3 documents the calculated drag coefficients for using lines of various lengths and distances away from the cylinder, and figure 4.6 plots the

integrand of equation 4.1 for this data.

Distance from Cylinder (m)	Length of Line (m)	C_D
0.006	0.024	0.243
0.006	0.06	0.463
0.042	0.025	0.487
0.042	0.063	0.454

Table 4.3: Calculated drag coefficients using various control volumes

These numbers give us an indication that the control volume method is working. We are about a factor of two away from the correct value. However, the calculated drag coefficients are highly dependent on free stream velocity because the speed is so small ($0.01 \frac{\text{m}}{\text{s}}$). We see this in figure 4.6. In the wake of the cylinder, contributions to the integral are positive. Around the cylinder however, small fluctuations in free stream velocity make the quantity $U - u_2(y) < 0$, resulting in erroneous and significant negative contributions. This process should be repeated on faster videos to decrease the percent error of these small changes in speed.

4.4 Kármán Vortex Street

PIVlab also allows us to explore some information directly. This section demonstrates some of these features in an effort to show the success of these initial experiments. The choice of a PVC rod for our PIV tests was not arbitrary; a cylinder moving through water between Reynolds numbers of 40 to about $1,000$ will shed vortices in an alternating pattern known as a Kármán vortex street. Using PIVlab's derived flow information, we can visualize the shedding of these vortices. These frames were compiled into a movie. Snapshots are shown in the figure below.

4.4.1 Visualization

To illustrate the vortex street, vorticity is plotted. Figure 4.7 shows voriticity minus the free stream velocity for a cylinder with diameter of 0.75 in moving through water at a speed of $0.014 \frac{\text{m}}{\text{s}}$, or a Reynolds number of about 250. All data was taken with the Sony at 60 fps. We can clearly see two regions of equal and opposite vorticity forming on either side of the cylinder. We also see these

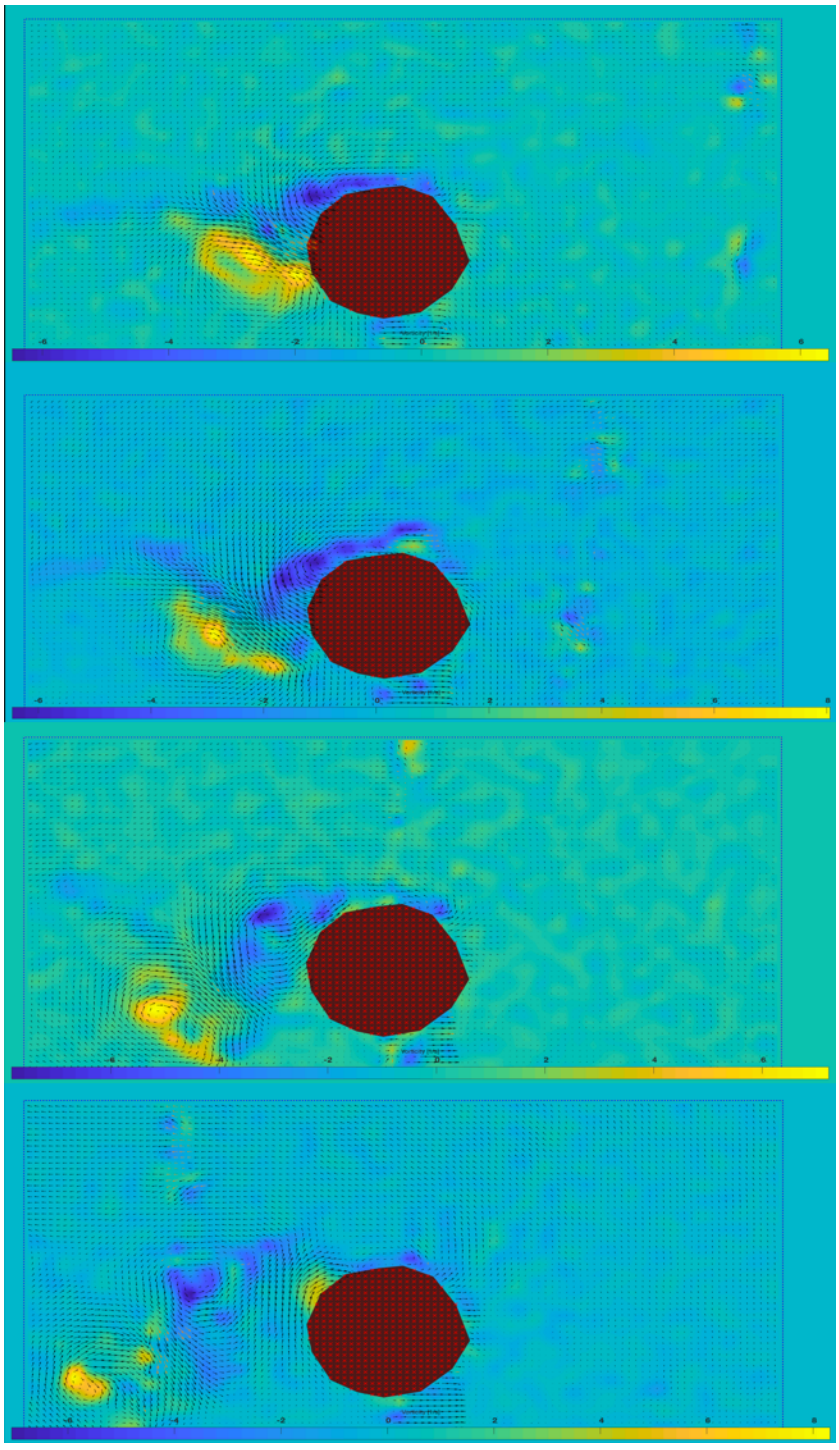


Figure 4.7: Vortices shed by a cylinder at a Reynolds number of 250. Time between first and last images is 5 seconds.

being shed in an alternating manner: in frame 2 of figure 4.7 we see a positive vortex shed, followed by a negative vortex in the fourth frame. The time between the first and last image in figure 4.7 is 5 seconds. For now, these images serve as confirmation that we are able to resolve these phenomena in the tank. In the future, such plots can be used to compare to literature data for a quantitative test of the validity of our measurements.

4.4.2 Shedding Frequency

The eddies in a vortex street are shed at a well-defined frequency. This frequency slightly depends on the Reynolds number, but is closely approximated by $0.2\frac{U}{a}$ [3]. We can compare our data to this number as a reality check. For now, determining the shedding frequency is a qualitative endeavor, decided by the point at which the streamlines found by PIVlab no longer begin on the surface of the cylinder (see figure 4.8).

The first vortex is shed on frame 40, and the second on frame 121. Since each frame in PIVlab is actually a pair of frames, these points are separated by 162 frames. Given the frame rate of the Sony (60 fps), these vortices are shed about 2.7 seconds after one another. This corresponds to half of the shedding period, as one full period would complete once another positive vortex is shed. We estimate the shedding frequency to be $\approx 0.19\,\text{Hz}$. This is compared to the expected frequency:

$$0.2\frac{U}{a} = 0.2\frac{0.014\,\text{ms}^{-1}}{0.019\,\text{m}} = 0.15\,\text{Hz}$$

Thus our vortex street agrees with the expected value. More quantitative future analyses can refine these results.

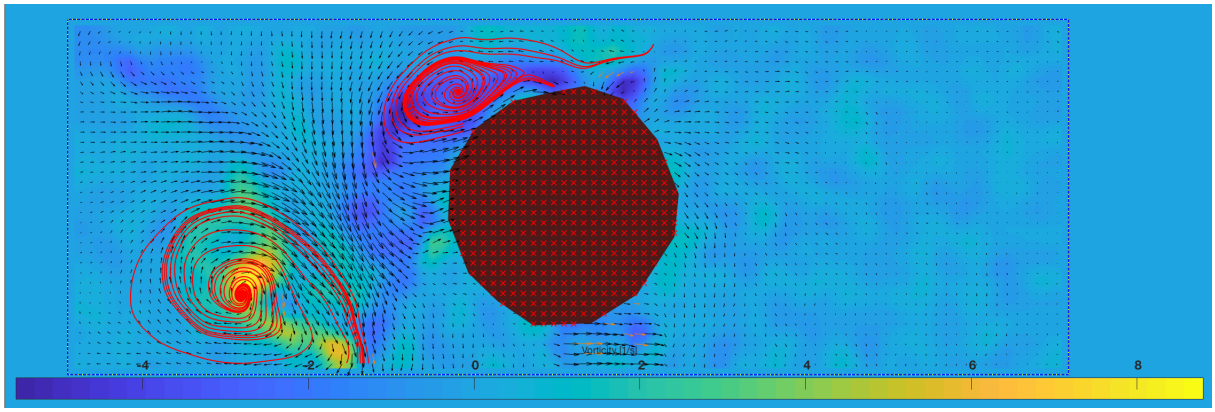


Figure 4.8: Streamlines overlaid on a heatmap of vorticity for a cylinder moving through water at $Re = 250$. We define a vortex to be 'shed' once the streamlines disconnect from the surface of the cylinder. Here we see the bottom vortex is clearly shed while the top vortex is not.

Chapter 5

CONCLUSION

This thesis sought to finalize the design of the Pomona College towing tank's controls & imaging subsystems and make initial efforts in validating its use as an experimental hydrodynamic facility. Design recommendations based on the results of this thesis are summarized below. Work remains before the tank can be used for novel fluids experiments. Recommended next steps are also presented here.

5.1 Summary of Work

The tow tank is now equipped with a controls system and GUI that should facilitate easy future experimentation. Built using accessible programming environments, it should be straightforward to change or extend its current functionality. Based on an SNR analysis of video data, the Sony Alpha a7s is the camera best equipped to handle the low-lighting environment of PIV experiments. Attempts at shooting at 120 fps could be promising. Otherwise, a camera with similar low-light ability and capable of shooting 240 fps would be ideal. Consideration should be given to what sorts of speeds will be used in the tank. If phenomena can be scaled up such that the velocity of the towing tank remains small, 120 fps may suffice. Tracer particle collection on the bottom of the tank clouds the camera's field of view too much for quality video. Trials with the submerged GoPro indicate that video taken from above, submerged in the water, would result in the best PIV video.

5.2 Future Work

5.2.1 Design for Submerged Video

To take submerged video data with a camera like the Sony, some sort of clear window must be used to avoid shooting through the ripples of the water. A possible solution is shown in figure 5.1. By

Figure 5.1: An underwater case found on Amazon for $20.00.

using a waterproof case, the camera can be mounted to the carriage and extended into the water. It must be confirmed that this case design does not restrict the camera's field of view.

An alternative approach is to repurpose a diver's box. A dome for the camera, as shown in figure 5.2, would provide a clear image as well as total field of view for the camera. In both cases, the 'window' needs to be mounted to the carriage. While there is room for this, care should be taken to ensure a design that does not get restrict functionality. For example, the the window should not enter too deep into the water to ensure that its wake does not interfere with the flow being imaged.

5.2.2 Characterization of Vibrations

No work was done in this thesis concerning the vibration of the carriage and camera arm. Specifically, both the camera and object being towed oscillate while the motor is running. Further efforts can be made to secure the rod, or whatever object is being towed, minimizing vibration. Vibration of the camera, however, should be analyzed to ensure that the amplitude of oscillation is small compared to the movement of our tracer particles.

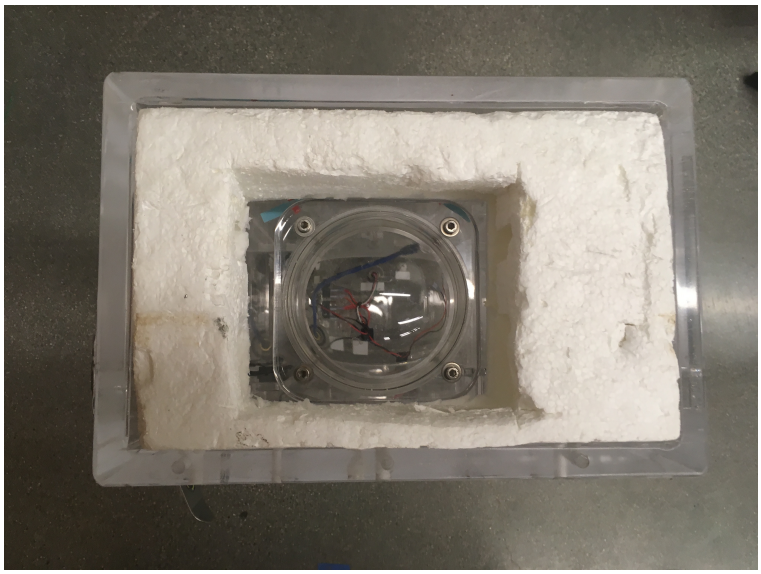


Figure 5.2: Diver's box in fluids lab. Repurposing a dome such as this one would ensure that the image is clear and the camera would have a full field of view.

5.2.3 Tank Characterization

Using the process outlined in 4.3.2, the tank needs to be further validated and characterized. In particular, we need to know how large an object can towed before boundary layer effects come into play. This information will identify what range of phenomena are analyzable in the tank. First, the methods of chapter 4 must be expanded to consistently derive accurate drag coefficients of known objects, such as a cylinder. By expanding the camera's field of view and corresponding control volume closer to the walls of the tank, we could see at what point the drag estimations deviate, implying that boundary layer effects have become significant.

BIBLIOGRAPHY

[1] R. J. Adrian. "Twenty years of particle image velocimetry". In: *Exp. Fluids. Online First* (2005).

[2] *Arduino Playground - Timer1*. URL: https://playground.arduino.cc/Code/Timer1 (visited on 03/29/2018).

[3] T. E. Faber. *Fluid dynamics for physicists*. English. Cambridge: New York : 1995. ISBN: 978-0-521-41943-7 978-0-521-42969-6.

[4] *Full Frame a7S Camera with 4K Video Recording | Alpha 7S | Sony US*. URL: https://www.sony.com/electronics/interchangeable-lens-cameras/ilce-7s (visited on 04/01/2018).

[5] Ashok K. Goel, Daniel A. McAdams, and Robert B. Stone. *Biologically inspired design: computational methods and tools*. English. London ; Springer-Verlag, 2014. ISBN: 978-1-4471-5248-4.

[6] M. Gad-el Hak. "The water towing tank as an experimental facility". en. In: *Experiments in Fluids* 5.5 (Sept. 1987), pp. 289–297. ISSN: 0723-4864, 1432-1114. DOI: 10.1007/BF00277707. URL: https://link.springer.com/article/10.1007/BF00277707 (visited on 11/06/2017).

[7] "Hydrodynamics of Insects. Part 1. Jetting of the Dragonfly Larvae. Part 2. Honeybee at the Air-water Interface: Surfing with the Capillary Wave". phd.

[8] *ISO and Image Noise*. URL: http://www.digital-slr-guide.com/iso-and-image-noise.html (visited on 04/01/2018).

[9] Katherine M. Marchetto et al. "Applications of particle image velocimetry for seed release studies". en. In: *Ecology* 91.8 (Aug. 2010), pp. 2485–2492. ISSN: 1939-9170. DOI: 10.1890/09-0853.1. URL: http://onlinelibrary.wiley.com/doi/10.1890/09-0853.1/abstract (visited on 11/12/2017).

[10] A. Melling. "Tracer particles and seeding for particle image velocimetry". en. In: *Measurement Science and Technology* 8.12 (1997), p. 1406. ISSN: 0957-0233. DOI: 10.1088/0957-0233/8/12/005. URL: http://stacks.iop.org/0957-0233/8/i=12/a=005 (visited on 10/25/2017).

[11] *Millennium Prize Problems*. en. Page Version ID: 811335128. Nov. 2017. URL: https://en.wikipedia.org/w/index.php?title=Millennium_Prize_Problems&oldid=811335128 (visited on 12/05/2017).

[12] *MIT Towing Tank |Home*. URL: http://web.mit.edu/towtank/www/ (visited on 12/05/2017).

[13] *National Science Foundation | Vizzies Visualization Challenge*. URL: https://www.nsf.gov/news/special_reports/scivis/vizzies-about.jsp (visited on 12/04/2017).

[14] *NPTEL Phase II :: Mechanical Engineering - Optical Measurement Techniques in Thermal Sciences*. URL: http://nptel.ac.in/syllabus/112104039/ (visited on 04/01/2018).

[15] Azar Eslam Panah and Amir Barakati. "Design and Build a Water Channel for a Fluid Dynamics Lab". In: June 2017. URL: https://peer.asee.org/design-and-build-a-water-channel-for-a-fluid-dynamics-lab (visited on 11/27/2017).

[16] *Powell Lenses - Lenses - General Optics - Optics*. URL: http://laserand.com/optics/general-optics/lenses/powell-lenses/ (visited on 11/20/2017).

[17] *Processing.org*. URL: https://processing.org/ (visited on 03/29/2018).

[18] Markus Raffel. *Particle image velocimetry: a practical guide*. English. 2nd ed. Experimental Fluid Mechanics; Experimental fluid mechanics. Heidelberg ; Springer, 2007. ISBN: 978-3-540-72308-0 978-3-540-72307-3.

[19] Luciano Amaury dos Santos et al. "Drag Estimation by Wake Survey Performed Measuring Velocities and Measuring Total and Static Pressures". In: (2006).

[20] Wouter Terra, Andrea Sciacchitano, and Fulvio Scarano. "Drag Analysis from PIV Data in Speed Sports". In: *Procedia Engineering* 147 (Dec. 2016), pp. 50–55. DOI: 10.1016/j.proeng.2016.06.188.

[21] William Thielicke and Eize Stamhuis. "PIVlab – Towards User-friendly, Affordable and Accurate Digital Particle Image Velocimetry in MATLAB". en. In: *Journal of Open Research Software* 2.1 (Oct. 2014). ISSN: 2049-9647. DOI: 10.5334/jors.bl. URL: http://openresearchsoftware.metajnl.com/articles/10.5334/jors.bl/ (visited on 10/28/2017).

[22] Steven Vogel. *Life in Moving Fluids: The Physical Biology of Flow*. en. Google-Books-ID: XBqncfXFsOIC. Princeton University Press, 1994. ISBN: 978-0-691-02616-9.

[23] Frank M. White. *Viscous fluid flow*. English. 3rd ed. McGraw-Hill series in mechanical engineering; McGraw-Hill series in mechanical engineering. New York, NY: McGraw-Hill Higher Education, 2006. ISBN: 978-0-07-240231-5 978-0-07-124493-0.

[24] Zhenyu Xue, John J. Charonko, and Pavlos P. Vlachos. "Particle image velocimetry correlation signal-to-noise ratio metrics and measurement uncertainty quantification". en. In: *Measurement Science and Technology* 25.11 (2014), p. 115301. ISSN: 0957-0233. DOI: 10.1088/0957-0233/25/11/115301. URL: http://stacks.iop.org/0957-0233/25/i=11/a=115301 (visited on 03/25/2018).

Appendix A

SCRIPTS

Commented scripts used in this thesis are documented below. They can also be pulled from github.

A.1 Get Frames

A script to extract the frames from a video in Matlab. Running this will create a new folder named 'frames' in the same directory as your video.

```
workingDir =  'frames';
mkdir(workingDir)
mkdir(workingDir, 'images')
PIVVideo = VideoReader('filename.m4v') %insert video name here
ii=00001;
while hasFrame(PIVVideo)
img = readFrame(PIVVideo)
filename = [sprintf('%05d',ii) '.png'];
fullname = fullfile(workingDir,'images',filename);
imwrite(img,fullname)
ii = ii+00001;
end
imageNames = dir(fullfile(workingDir,'images','*.png'));
imageNames = imageNames.name;
```

A.2 Undistortion Script

Matlab Ssript to undistort GoPro footage from Jonah Grubb's thesis.

```
workingDir = tempname;
mkdir(workingDir)
mkdir(workingDir, 'images')
PIVVideo = VideoReader('Filename.m4v') %insert filename here
ii=00001;
while hasFrame(PIVVideo)
img = readFrame(PIVVideo)
undistortedImage = undistortImage(img, cameraParams__);
filename = [sprintf('%05d',ii) '.png'];
fullname = fullfile(workingDir,'images',filename);
```

```
imwrite(undistortedImage,fullname)
ii = ii+00001;
end
imageNames = dir(fullfile(workingDir,'images','*.png'));
imageNames = imageNames.name;
```

A.3 Additions to PIVlab Code

Correlation matrices all pass through the SUBPIXGAUSS function of 'piv_FFTmulti.m'. Just before this function is defined (line 410), define these global variables:

```
global all_data;
global pass_number;
all_data = [];
pass_number = 1;
```

Then at the end of the function, add:

```
    global all_data;
    global pass_number;
    %get the x,y, and z sizes of result_conv
    [sz1,sz2,sz3] = size(result_conv);
    data = [];
    for k = 1:sz3 %iterate through every interrogation area
        MaxValue = 0;
        secondMax=0;
        correlation_matrix = result_conv,:,k)
        correlation_matrix = sort(correlation_matrix,1);
        correlation_matrix = sort(correlation_matrix,2);
        MaxValue = correlation_matrix(sz1,sz2);
        secondMax = correlation_matrix(sz1,sz2-1);

        if secondMax ~= 0 %avoid Nans
            data = [data,MaxValue/secondMax];
        end
    end
    data_mean = mean(data);
    data_std = std(data);
    all_data = [all_data, data_mean, data_std]
    %if you want figures
    figname = strcat("meshplot",num2str(pass_number),'.png');
    h=mesh(result_conv(:,:,2));
    saveas(gcf, figname)
    h=figure('visible','off')
    pass_number = pass_number + 1;
```

Be sure to run the 'PIVlab_GUI.m' file directly if you sush to use these changes- the command line prompt will not register them.

A.4 Drag Coefficient

The following is a Python script for reading in the csv outputted by PIVlab and calculating the integral for the drag coefficient. It is straightforward to translate this into Matlab and it would probably be better to stay in one environment; Python was used because I had an existing familiarity with reading and writing files. Run this script in the same folder that you save the csv in.

```
import csv
import numpy as np
import matplotlib.pyplot as plt
def readcsv( csv_file_name ):
    try:
        csvfile = open( csv_file_name, newline='' )
        csvrows = csv.reader( csvfile )

        all_rows = []
        for row in csvrows:
            all_rows.append( row )

        del csvrows
        csvfile.close()
        return all_rows

    except FileNotFoundError as e:
        print("File not found: ", e)
        return []

f = readcsv("PIVlab_data.txt") #insert your file name here
x_data = []
y_data = []
cntr = 0
for point in f:
    if cntr>1: #counter used to avoid the headers of the columns
        x_data.append(float(point[0]))
        y_data.append(float(point[1]))
    cntr += 1

U = -0.013 #enter your free stream velocity here
density = 1000 #density of water


### Now integrate for drag coefficient.  Calculation separated into parts:
```

```
#for subtraction term:
y_data2 = []
for point in y_data:
    y_data2.append(U-point)

integral_data = []
for i in range(len(y_data)): #this is what we'll actually integrate
    integral_data.append(y_data[i]*y_data2[i])


integral = np.trapz(integral_data,x_data)
integral = 2* integral / ( U*U * 0.02)
print(integral)
plt.plot(x_data,integral_data)
plt.xlabel("Distance along line (m)")
plt.ylabel("$u_2(y)(U-u_2(y))$")
plt.tight_layout()
plt.show()
# plt.savefig('integral_fig',dpi = 400) #Save the graph if you'd like

# Getting average free stream velocity along inlet
# freestream = readcsv("my_freestream_data.txt")
# cntr = 0
# x_data2 = []
# y_data2 = []
# for point in freestream:
#     if cntr>1:
#         x_data2.append(float(point[0]))
#         y_data2.append(float(point[1]))
#     cntr += 1
# print(np.mean(y_data2))
```

Appendix B

LAB MANUAL

B.1 Capturing Video

The particles in the tank first need to be swirled to mix in any that have settled to the bottom. This was done by using a ruler and scraping the entire bottom of the tank and continuously mixing. Some particle accumulation on the bottom is inevitable, but persistent stirring in this manner results in a relatively clear bottom of the tank. With time, the particles will begin to settle again. Waiting about 15 minutes after stirring seemed to be enough time to ensure the water was still again, but not too long such that the bottom becomes too cloudy to image. This is about the amount of time it takes to set up the rest of the experiment.

Next, open the GUI on the controls system and set up your run. It's best to have this ready to go so it doesn't need to be done with the lights off and laser on. The code is online on github at https://github.com/bsubbaraman/pomona_towtank, and once a permanent machine is installed in the lab, it can be run from as executable file.

Both the camera and the laser need to be setup up prior to running the tank. First, make sure the camera is configured with all of the correct settings. When using the Nikon or Sony, the standard settings used were manual focus with the lowest possible $f^{\#}$ at 60 fps with an autoset ISO. All cameras like the Nikon and Sony have a 1/4 inch socket for use with tripods. We make use of this to make a rig out of optical mounts. Screw the camera in as shown in B.1 and then attach this below the tank to the camera arm using a C-Clamp. When positioning the camera, keep in mind that it is necessary to have view downstream of the object being towed (see section 4.3.1). Try to position the camera such that the object is upstream and centered to give the best view of the wake. For the GoPro, a phone app can be used to change settings and start recording video. After downloading the app, go into Settings → WiFi on the GoPro to enable phone connectivity. Find it

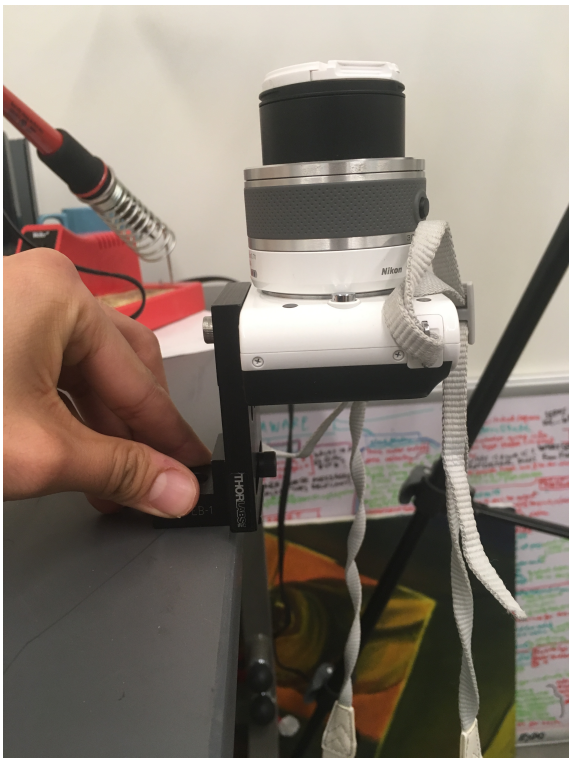


Figure B.1: Mount used to attach camera to tank. C-Clamp this mount under on to the camera arm.

on your phone's WiFi (named 'whitakerlabs') and connect to it with the password 'Sphagnum47!'. From the app, change the settings to 120 fps in narrow mode. If taking video from below the tank, secure the GoPro directly to the camera arm with a C-Clamp. From above the tank, the GoPro connector was ziptied to the object being towed (the PVC rod in this case).

Set up the laser $\approx$ 2 feet away from the tank. Next, adjust the optics. Before turning on the laser, be sure that you have laser safety glasses on! Make sure that the lens is oriented such that the light-sheet is parallel to the ground and positioned so that the light sheet is contained within the tank. Adjusting the tripod height so that the laser as high as possible yielded the best results as it makes focusing the camera easier. Set the intensity of the laser to 6.00. Be sure that the walkways between the laser, the light switch, and the tank controls system are clear, because you will have to navigate these in the dark! Once the laser is set up, turn off the lights. Next, if taking data with a digital camera from under the tank, access the camera to adjust the manual focus. This is done by rotating the lens of the Sony and by pressing the center button then rotating the dial on the Nikon.

Peeking around your laser goggles will probably be necessary here. Once everything is as you like it, press record (very important!). You will also want to have a calibration image. Keep a ruler underneath the tank and place it in the frame after you start recording. We can grab a snapshot of this in post-processing.

Go to the controls system to start the tank run. Ideally, this is already set up. All that needs to be done is press 'RUN', and the carriage will start and stop automatically. If running the GoPro from your phone, hit record just before running the tank. If you wish to run the tank multiple times, simply enter your new parameters and run again. Depending on the object and how fast you are running at, be careful to wait enough time between consecutive runs for the water to become still again.

After your run is done, turn the lights on. If using the GoPro, you can stop recording immediately from your phone. Turn the laser intensity down to zero before switching it off. Finally, stop recording on the camera and take it off of the arm and mount to access the SD card.

B.2 Analyzing Data in PIVlab

First, export your video to your computer, and save it in your working research directory. Trim the video to only what you wish to analyze. At the very least, this means trimming any time the carriage isn't moving. For my tests, I used a short 5 second clip so that I could run several different videos in a shorter amount of time. Name it something descriptive; my naming scheme included the camera used and speed I ran the tank at. You should also take a snapshot of a frame with the ruler in the picture for future use as a calibration image. Using VLC player (the media player that I use on my computer), you can take a snapshot of your video and save it directly into your working directory.

Next we need to grab the frames from the video. Open 'getFrames.m' (see Appendix A.1), and insert the name of your in line 4. This script will read all the frames and save them in a folder named 'frames' in the same directory as your video. The Nikon, Sony, and GoPro in narrow mode

Figure B.2: The PIVlab GUI's home screen. This will open after entering 'PIVlab_GUI' in the command window or running 'PIV_GUI.m' directly.

have no lens distortion to worry about. However, if using the GoPro in wide mode, you will have to un-distort the barrel distortion. See appendix A.2 for this code and Jonah Grubb's thesis for more information on this. It takes about 20 minutes to grab a few seconds worth of frames from a video at 60 fps. The undistortion script takes significantly longer, taking hours to get the same amount of frames.

Now we are ready to analyze these frames in PIVlab. After adding the PIVlab app to Matlab (go to the APPS tab and search for PIVlab), you can open the interface by typing 'PIVlab_GUI' in the command window. If you wish to do further analysis on the correlation matrices (see section 4.1), run the PIVlab script with the additions documented in Appendix A.3. The screen shown in figure B.1 will open. Note that on Mac, there is a bug that makes the text overlap. This has been noted by William Thielicke (PIVlab's creator) with no immediate plans to fix it. Select 'Load Images', then navigate to your images to import and add them to PIVlab (figure B.2). The standard sequencing style is '1-2, 3-4'; this means that the frames 1 and 2 will be compared to one another to determine displacement vectors, and then 3 and 4, and so on. This can be seen after images are added by

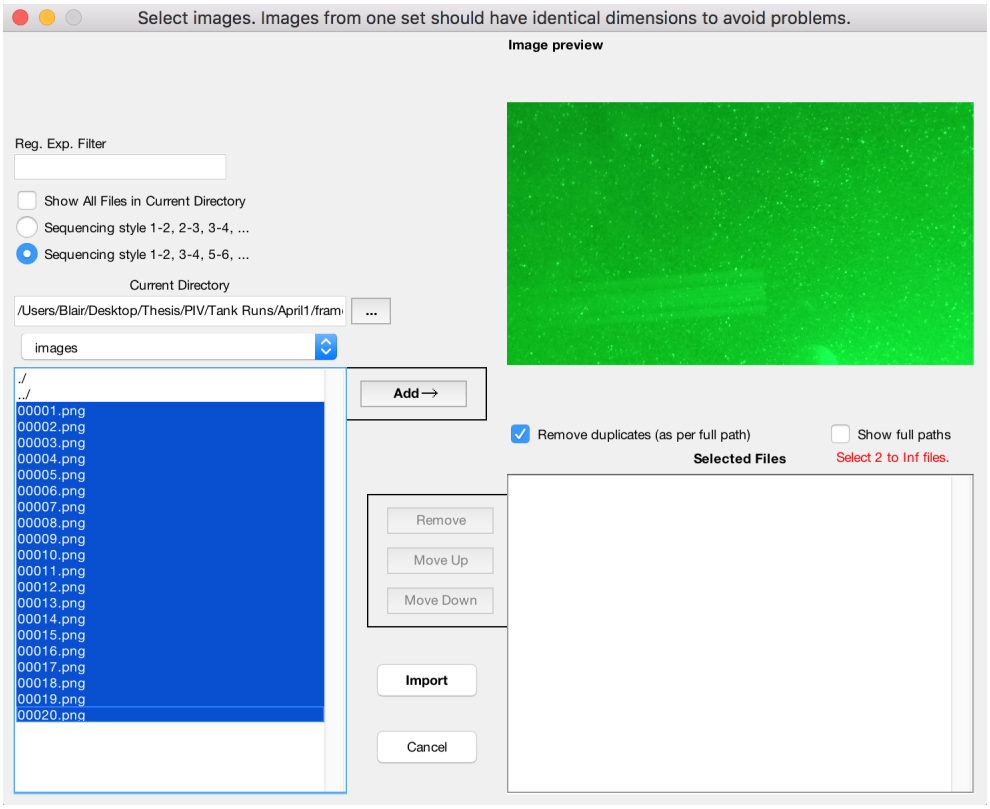


Figure B.3: Screen to import your frames for analysis into PIVlab.

noting the number of frames in the bottom right. Uploading 20 images results in 10 frames in PIVlab, because its pair is hidden from view for comparison.

Next, we need to draw masks for our image (figure B.3). A mask defines regions we don't want to analyze. The rod itself, for example, should not be analyzed. Select 'Analysis Settings → Exclusions' from the top bar, then 'Draw mask(s) for current frame' on the right hand side. You can now click on points which will draw a polyline. Enclose the rod (or whatever object you are masking) by completing a closed loop then double click inside the loop to exit drawing mode. Select 'Apply current mask(s) to all frames' so that the rod is masked in every image. We can also define a region of interest here which helps speed up analysis by only analyzing the parts of the image inside of the user-selected region.

Under 'Analysis Settings → Image Pre-Processing' we can apply the filters to our images (see sections 2.2.4 and 4.2.1 for more information on these). We want to apply CLAHE, highpass, and

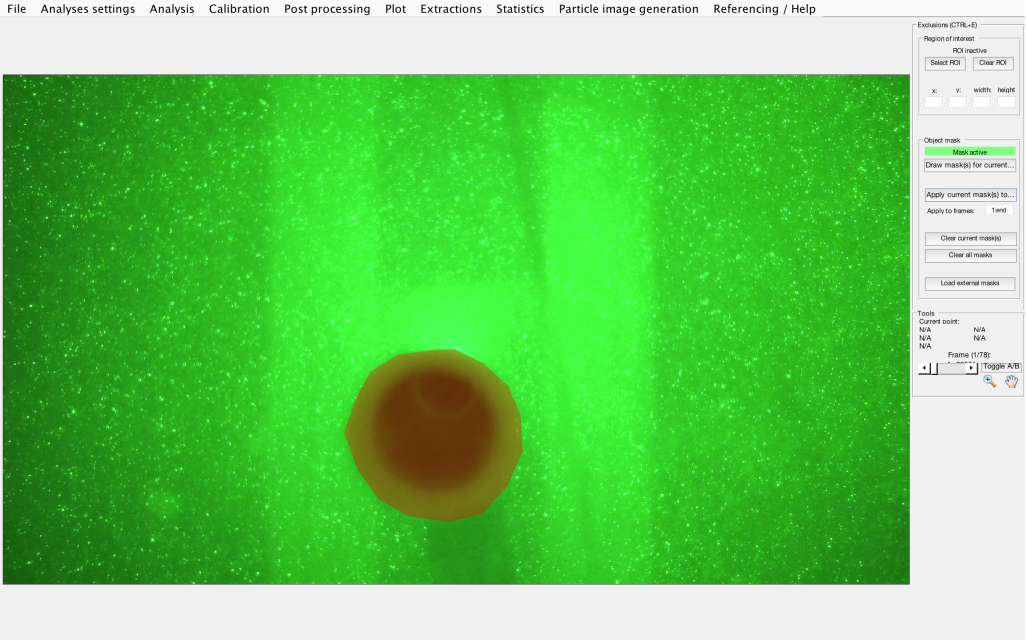


Figure B.4: Applying masks to our frames. Make a closed loop and double-click inside of it to exit drawing mode.

intensity capping. Selecting 'preview current frame' will result in something like figure B.4.

'Analysis Settings → PIV Settings' will let you set up the size of interrogation windows for PIV analysis (figure B.5). The optimal window sizes will depend on the image. More passes will yield more accurate analysis, so 4 passes is best. Start with a large interrogation window of 128 pixels and step size of 50% (64 in this case), and decrease until the smallest window contains about 7-10 particles (see section 4.1 for more information). The other settings here should stay on the defaults.

We're ready to analyze! Navigate to 'Analysis → Analyze' and select 'Analyze all frames'. PIVlab's progress will be tracked on the right hand side. Even with four passes, analysis is not too time consuming: 80 frame pairs (160 total frames) takes around 20 minutes. After the analysis is complete, select Calibration from the top bar. Upload the calibration image you snapshotted earlier and select a reference distance. I generally use one centimeter on the ruler. Next, enter the distance (in mm) and time step of the images (in ms), which is dictated by the framerate of the camera. For the Sony data being shown here, this is 16.67 ms.

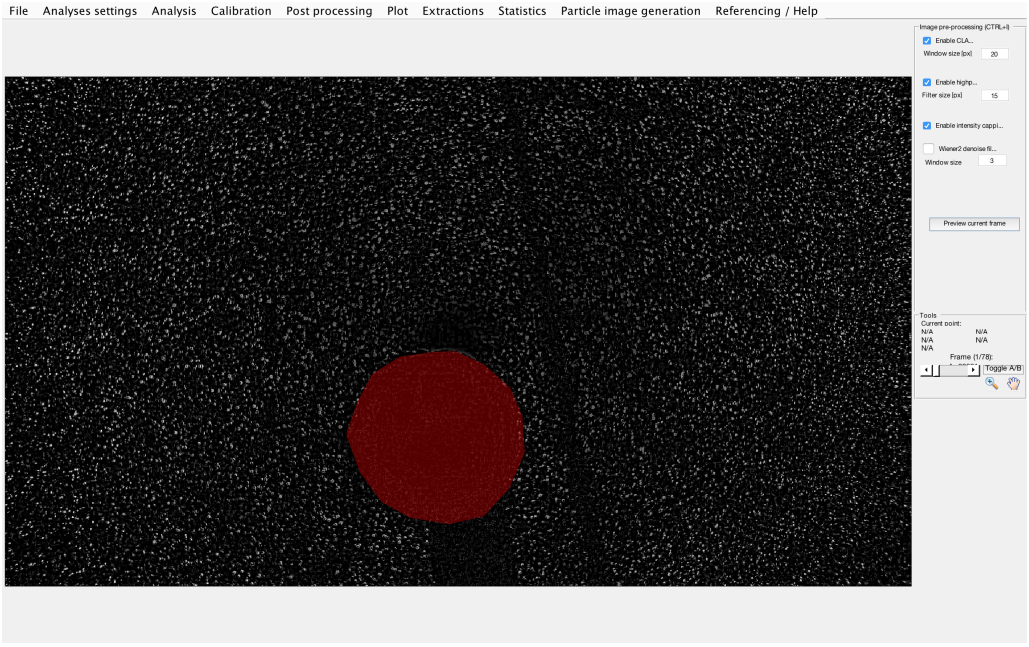


Figure B.5: What our images look like after applying pre-processing techniques.

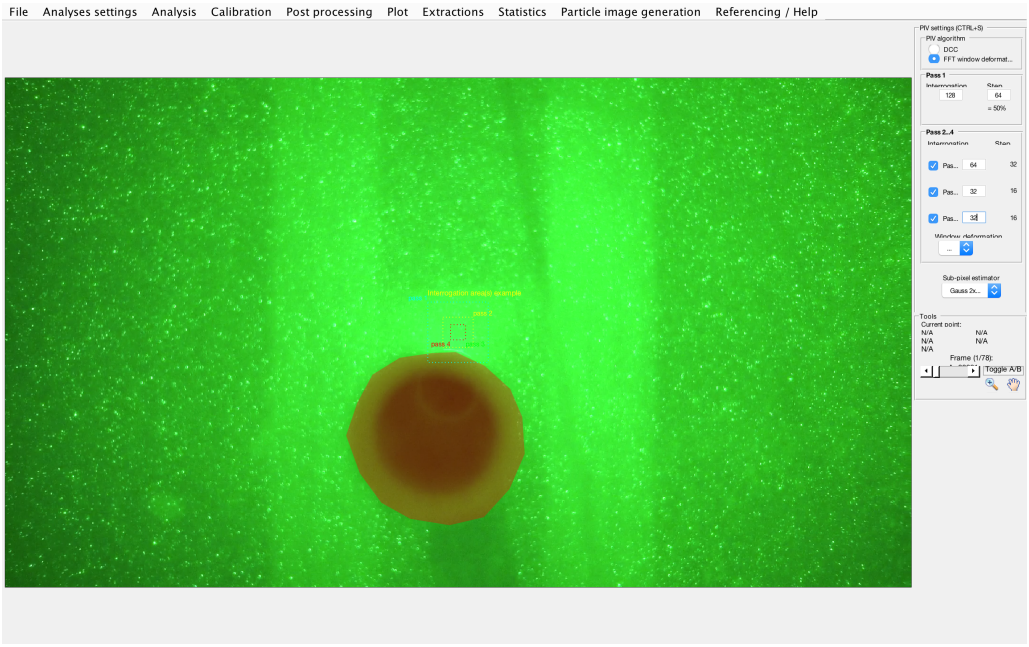


Figure B.6: Setting up the window sizes for analysis under PIV settings. Use four passes. Start large and decrease until the last pass(es) contain 7-10 particles per window.

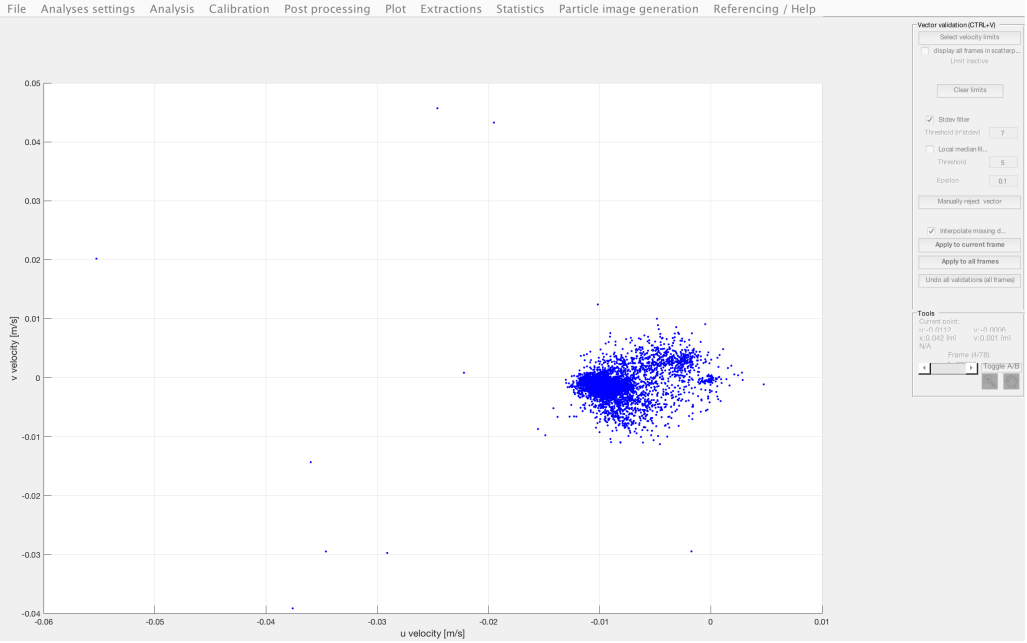


Figure B.7: Select velocity limits in post-processing. Be careful to exclude only erroneous vectors given the operating speed, not interesting flow data!

After calibration, enter 'Post-Processing → Vector Validation' to select velocity limits (figure B.6). We wish to exclude points that are erroneous given the speed we are operating at, but be careful not to exclude points that may give interesting information about our flow. After selecting your region, select 'Apply to all frames' so that these limits are set for every image.

Data analysis from this point is open ended. Many options are available under 'Plot' and 'Extractions'. To derive drag data, we first want average velocity vectors for all frames. Go to 'Plot → Derive parameters/modify data' and on the right hand side, select calculate mean vectors for 1:end. This will make a new frame at the very end that contains average vectors. Go to 'Extractions → Parameters from poly-line' to define the line over which the control volume method will be applied. Select 'Draw' and select two points to draw a line (figure B.7). Press 'z' to exit draw mode. We wish to derive the u-component of velocity from this line, so select this option from the drop-down menu. 'Plot data' will give a PIVlab generated graph of this data.

After drawing your line, select 'Save result as an ASCII chart' and save this file in the same place as the python script in Appendix A.4. Change the file name in this script to the name of the ASCII

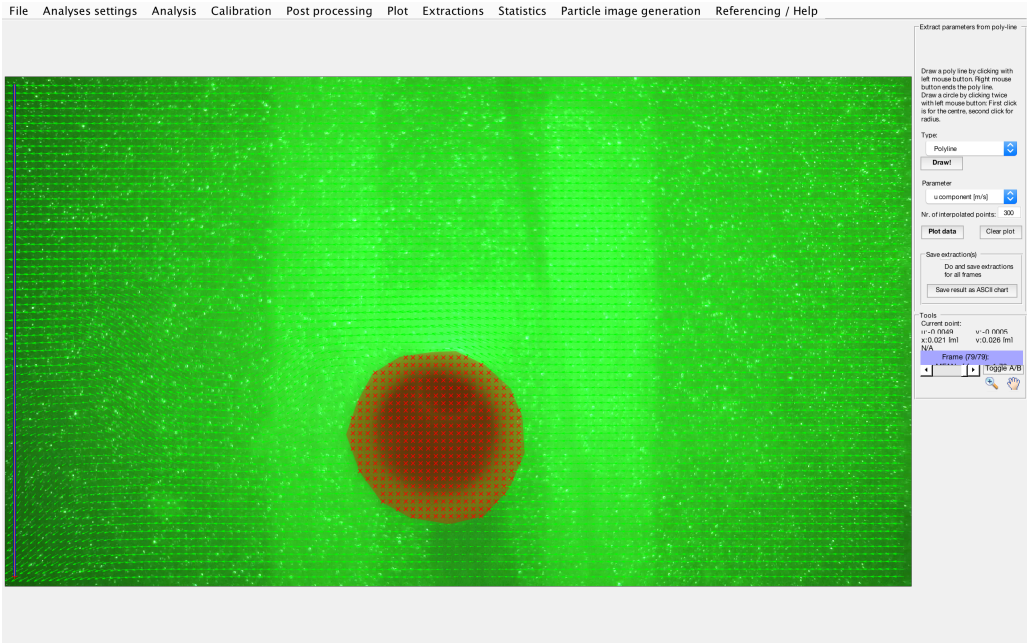


Figure B.8: Draw a poly-line over which the u-component of velocity can be derived. This will be used to make drag coefficient estimations. Note that, in the bottom right, we are in a new frame highlighted in blue to indicate these is our average velocity vectors.

file just saved and run it to compute the wake survey method integrals outlined in equation 4.1. This will print the value of this integral (an estimate of the drag coefficient) and produce a plot of the integrand.

B.3 VFD Settings

Various VFD settings were changed in order to drive the motor from the Arduino and add in new components. The following table consolidates these changes. Titles that begin with ‘A’ apply to the settings of the VFD, those that begin with ‘F’ are information about the VFD and those that begin with ‘C’ assign the terminal with that number a particular function.

Setting	Function	Description
A001	02	Run motor from current input, disables ‘RUN’ button
A001	01	Run motor from VFD, ignores current input
C001	00	Input to terminal 1 will run the tank forwards
C002	01	Input to terminal 2 will run the tank in reverse
C003	07	Input to terminal 3 will apply brakes
C004	07	Input to terminal 4 will apply brakes
F001	N/A	Display the frequency of the motor

Table B.1: A reference for the various settings on the VFD