TRIBHUVAN UNIVERSITY
INSTITUTE OF ENGINEERING
PULCHOWK CAMPUS

B-10-BAS-2018/23

FABRICATION OF PARTICLE IMAGE VELOCIMETRY SETUP FOR EXPERIMENTATION AT LOW REYNOLDS NUMBER

by
Ganesh Dhungana (075AER014)
Sandip Gautam (075AER038)
Yukesh Karki (075AER048)

A PROJECT REPORT SUBMITTED TO THE DEPARTMENT OF MECHANICAL AND AEROSPACE ENGINEERING IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF BACHELOR IN AEROSPACE ENGINEERING

DEPARTMENT OF MECHANICAL AND AEROSPACE ENGINEERING
LALITPUR, NEPAL

MARCH, 2023

COPYRIGHT

TRIBHUVAN UNIVERSITY
INSTITUTE OF ENGINEERING
CENTRAL CAMPUS PULCHOWK
DEPARTMENT OF MECHANICAL AND AEROSPACE ENGINEERING

The undersigned certify that they have read, and recommended to the Institute of Engineering for acceptance, a project report entitled "FABRICATION OF PARTICLE IMAGE VELOCIMETRY SETUP FOR EXPERIMENTATION AT LOW REYNOLDS NUMBER" submitted by Ganesh Dhungana, Sandip Gautam and Yukesh Karki in partial fulfillment of the requirements for the degree of Bachelor of Aerospace Engineering.

Supervisor, Kamal Darlami
Assistant Professor
Department of Mechanical and Aerospace Engineering

Supervisor, Neeraj Adhikari
Assistant Professor
Department of Mechanical and Aerospace Engineering

External Examiner, Bikalpa Bomjan Gurung
Consultant, Flight Operations Engineering
3Green, UAE

Committee Chairperson, Dr. Surya Prasad Adhikari
Head, Associate Professor
Department of Mechanical and Aerospace Engineering

Date: _____

ABSTRACT

Particle Image Velocimetry (PIV) is a widely utilized experimental technique for fluid analysis, particularly in low Reynolds number flow. This paper describes the design fabrication process of the PIV setup and its application for studying the flow over a flat plate (with a chord of 76.33 mm) at low Reynolds numbers ranging from 2,000 to 10,000. The PIV system employs a class IIIB laser to illuminate a plane of seeding particles, a high-speed camera (Chronos 2.1), and a smartphone camera to capture the frames at frame rates of 100 fps and 16 fps, respectively. Sequential frames are cross-correlated using PIVlab, a MATLAB-based image processing application for PIV, to generate two-dimensional velocity fields. The velocity field data obtained are analyzed to identify low Reynolds number flow features, such as laminar separation bubble, reattachment point, turbulent mixing, and trailing edge vortex shedding pattern. Furthermore, the experimental results are compared to the results from numerical simulation carried over same geometry at similar Reynolds number. Finally, different sources of errors in the experimental setup, image processing, and image correlation are identified.

Keywords: particle image velocimetry, separation bubble, votex shedding

ACKNOWLEDGEMENT

We would like to express our gratitude to all those who have contributed to the successful completion of the project. First and foremost, we take this opportunity to express our sincere thanks to all those for helping us to get through this report. Without their supervision, guidance, and support, we could not have finished the project.

We are heartily thankful towards the **Department of Mechanical and Aerospace Engineering, IOE Pulchowk** for providing us the opportunity to work on this project.

We extend our gratefulness towards our supervisors **Asst. Prof. Kamal Darlami**, Deputy Head of Department and **Asst. Prof. Neeraj Adhikari**. We would also like to provide special thanks to **Asst. Prof. Sudip Bhattarai, PhD** for his supervision during various phases of the project.

We would like to acknowledge **Er. Lokesh Silwal** and **Er. Bibek Sapkota** for their constant support and guidance throughout the project.

At last, we express our sincere gratitude to **Mr. Dhirendra Baral** from **Maalika Aluminium Shop** for providing us with the workspace, tools and proper supervision during the fabrication of the setup. Besides, there were many friends and teachers to whom we are very much grateful as they provided us with their valuable suggestion and supervision throughout the project, which has driven us to this phase of the project timeline.

Lastly, we would like to acknowledge **Incubation, Innovation and entrepreneurship Center (IIEC)** for the workspace and room for our setup.

Authors:
Ganesh Dhungana (PUL075AER014)
Sandip Gautam (PUL075AER038)
Yukesh Karki (PUL075AER048)

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

R_e	Reynolds Number
τ_e	Relaxation time
d_p	Particle diameter
ρ_p	Particle density
μ	Viscosity
V_T	Settling Velocity
g	Acceleration due to gravity
ρ_m	Fluid density
C_s	Light scattering cross-section
P_s	Scattered power of light
I_o	Incident laser intensity on particle
λ	Light wavelength
n	User specified constant
σ	Standard deviation
Γ	Circulation
v	Velocity
L	Lift
L'	Lift per unit span
ρ_∞	Free stream density
v_∞	Free stream velocity
P	Pressure
I	Identity matrix
τ	Stress tensor
A	Area
V	Volume
a_p	Constant in momentum discretization
u_{nb}	Velocity in neighbouring cell
p_f	Pressure in cell face
S	Source term
P_0	Pressure in cell center
∇	Divergence Operator
$\vec{r}$	Displacement Vector
f	Face of cell
J_f	Mass flux through face
A_f	Area of cell face
V_n	Normal velocity

S_l	Planform area
b	Span
AR	Aspect ratio
BR	Blockage ratio
c	chord
D	Pulley diameter
N	Stepper RPM
w	Angular velocity
$\phi(\tau)$	Wagner function
ε	Error
R	Correlation coefficient

LIST OF ACRONYMS AND ABBREVIATIONS

AoA	Angle of Attack
CAD	Computer Aided Design
CLAHE	Contrast Limited Adaptive Histogram Equalization
DC	Direct Current
DCC	Direct Cross Correlation
FEM	Finite Element Method
FFT	Fast Fourrier Transform
FPS	Frames Per Second
GUI	Graphical User Interface
HR-PIV	High-Resolution Particle Image Velocimetry
HSC	High Speed Camera
ITTC	International Towing Tank Conference
LE	Leading Edge
LEV	Leading Edge Vortex
LSB	Laminar Separation Bubble
MAV	Micro Aerial Vehicle
NACA	National Advisory Committee for Aeronautics
Nd:YAG	Neodymium-doped Yttrium Aluminum Garnet
PIV	Particle Image Velocimetry
PLA	Poly Lactic Acid
PTV	Particle Tracking Velocimetry
RMS	Root Mean Square
ROI	Region of Interest
RPM	Revolutions Per Minute
SIMPLE	Semi Implicit Method for Pressure-Linked Equations
SPIV	Stereoscopic PIV
TE	Trailing Edge
TR-PIV	Time-resolved PIV

CHAPTER ONE: INTRODUCTION

1.1 Background

Flow visualization is a crucial process in fluid dynamics for obtaining both qualitative and quantitative information about flow patterns. This involves making the flow patterns visible through various techniques such as surface flow visualization, particle tracer methods, and optical methods.

While qualitative visualization provides the necessary observation of the flow field, it may be deceptive in its flow patterns. To address this issue, quantitative flow visualization is used to study the flow field accurately and quickly. Among these techniques, Particle Image Velocimetry (PIV) is one of the best methods for rendering the global velocity field with high reliability and ease. PIV involves the quick acquisition of several thousand image frames of a given flow domain, enabling the rendering of the global velocity field [1]. PIV has been widely used since its first experimental setup in 1977 by researchers Dudderar and Simpkins, Barker and Fourney, and Grausson and Mallick to view the velocity profile in a laminar pipe flow using double exposure photographs and planar light sheet illumination. PIV is a multi-point technique, meaning that the flow is captured based on location of multiple points distributed throughout the flow. The points are tracer particles floating in the fluid that a High Speed Camera captures in photos. The PIV approach assumes that the tracer particle acts similarly to fluid particles. Cross-correlation methods are used to determine their location over two time frames. The distance travelled by the particle divided by the time it takes to go that distance is the particle's velocity, hence a velocity field can be generated.

1.2 Problem statement

The problem statement for this project is to develop a Particle Image Velocimetry (PIV) setup and use it to conduct low Reynolds flow experiments. The goal is to accurately measure the velocity field of the fluid under investigation using PIV, which involves acquiring and analyzing a large number of images of the flow field. The challenge lies in setting up the PIV system correctly, calibrating it, and optimizing the image acquisition and analysis processes to obtain reliable and accurate results. Additionally, the low Reynolds number flow presents its own unique challenges, such as the need for careful control of the flow rate and the avoidance of turbulence. The successful development of the PIV system and conducting the low Reynolds flow experiments will advance our understanding of fluid mechanics and contribute to the development of more efficient and effective fluid systems.

1.3 Objectives

1.3.1 Main Objective

The main objective of the project is to design and build a working setup of Particle Image Velocimetry (PIV) for low Reynolds number flows experiment.

1.3.2 Specific Objective

1. (a) To develop a comprehensive understanding of the physical principles and techniques behind PIV in general.
   (b) To check the feasibility of using glass powder as PIV tracer particles.
2. (a) To design and build a Particle Image Velocimetry (PIV) setup capable of measuring velocity fields in low Reynolds number flows.
   (b) To conduct experiments on flat plate to investigate the flow characteristics at low Reynolds number flows, such as Leading Edge Vortex, separation bubble and vortex shedding.
   (c) To analyze the experimental data obtained from the PIV setup to extract meaningful insights about the flow behavior, such as velocity profiles and vorticity contours.
   (d) To compare the experimental results with numerical simulation.
3. To identify the limitations and sources of uncertainty in the PIV measurements and propose strategies to mitigate them.

1.4 Application

- **Aerospace engineering:** PIV can be used to study the airflow around aircraft wings and improve their design for better aerodynamics and fuel efficiency.
- **Automotive engineering:** PIV can be used to study the flow of air and fuel in engines, optimizing their combustion process for higher performance and lower emissions.
- **Biomedical engineering:** PIV can be used to study the flow of blood in veins and arteries, helping to diagnose and treat cardiovascular diseases.

- **Environmental science:** PIV can be used to study the flow of water in rivers and oceans, understanding the effects of climate change and improving flood prediction models.
- **Fluid mechanics research:** PIV can be used to study the fundamental properties of fluids, such as turbulence and mixing, advancing our understanding of this field and leading to new discoveries and innovations

1.5 Features

- User-friendly software for providing input to the gantry system.
- Robust structure of the setup and removable parts for easy maintenance.
- Experimentation up to Reynolds number of 10,000.

1.6 Feasibility Analysis

Building a Particle Image Velocimetry (PIV) setup depends on various factors such as the availability of resources, budget, and the intended use of the setup. The feasibility analysis is described below:

1.6.1 Economical Feasibility

PIV setups can range from simple, low-cost designs to more complex and expensive ones. The cost of building a PIV setup can include the cost of components such as the laser, camera, optics, and software. Additionally, the cost of consumables such as tracer particles should be considered. Including all the materials and operational cost, the estimated cost for the setup could range from Rs. 75,000/- to Rs. 2,00,000/-. Considering the availability of the major components such as continuous plane sheet laser, high speed camera and other electronic items, building the setup would be economically feasible.

1.6.2 Technical Feasibility

PIV setup requires many low cost to expensive technical components. Also the post processing tools falls under software requirement. We are planning to use the laser and

camera from the Department of Mechanical and Aerospace Engineering, locally available glass powder as tracer particles and open source software called PIVlab for image processing. Hence, it can be concluded that developing a PIV setup is technically feasible with the available resources and technology.

1.6.3 Operational Feasibility

There are many problems regarding the aspect of operation. One of the major factors in operational feasibility is user training. With proper hands-on training regarding the operation of the PIV setup and interpretation of the results, any user can perform the experiment.

1.7 System Requirements

1.7.1 Software Requirements

Softwares and programming languages required for the completion of the project are listed below:

- CATIA - PIV setup 3D modeling
- ANSYS - PIV setup FEA and CFD validations, CFD of test specimen
- MATLAB - PIVlab program
- Python - Image segmentation, PIV image masking
- PIVlab (MATLAB Toolbox)

1.7.2 Hardware Requirements

Hardwares required for the completion of the project are listed below:

- Class IIIB Laser
- Tracer Particles
- High Speed Camera

- Electronic Components such as NodeMCU, Stepper Motor, etc.

CHAPTER TWO: LITERATURE REVIEW

Over the past couple of years, particle image velocimetry has been widely adopted to analyze various problems related to fluid dynamics. The first experimental setup for PIV traced back to 1977 when a group of researchers in [2, 3, 4] carried out an experiment using double exposure photographs and planar light sheet illumination to view the velocity profile in a laminar pipe flow. By 1983, the researchers in [5, 6, 7, 8, 9, 10] working with this method demonstrated that it can be applied to laminar as well as turbulent fluid flow of liquid and gases as well. The significant use of PIV technique in recent days can be noticed in fluid mechanics and aerodynamic research and developments [11].

PIV technique has been widely used in the field of aerodynamics to investigate unsteady flow phenomenon by capturing the instantaneous flow velocity field in a plane of flow within a very short period of time [11, 12, 13]. Particularly, low Reynolds aerodynamics has been of keen interest with the significant development in the field of micro aerial vehicles (MAVs). At low Reynolds flow, there will be certain reduction in lift with corresponding increment in drag due to the formation of laminar separation bubble (LSB). To understand the low Reynolds number aerodynamics, Genç et. al. in [14] and Park et. al. in [15] conducted the experimental study to investigate the flow over the NACA2415 and DAE51 airfoil respectively and the laminar separation bubble (LSB), transition and reattachment flow were investigated.

At low Reynolds flows, studies have shown that the performance of flat plate is significantly better than that of airfoils. Researchers have investigated the behavior of flat plate (flapping wings) and airfoils in the low range of Reynolds number of $< (10^4)$ and concluded that the performance and characteristics of the flat plate to be superior [16]. Sunada et. al. in [17] investigated the aerodynamics of various airfoils at $Re = 4 \times 10^3$ using towing tank facility. They found larger lift slope of flat plate with a thickness ratio of 5% than conventional airfoils. Ford et. al. in [16] performed experiment on a flat plate using towing tank by impulsively starting from rest at an AoA of 15° to study the unsteady flow. Leading and trailing edge vortex circulation was captured using quiver plots obtained using PIV.

2.1 Theoretical Background

2.1.1 Particle Image Velocimetry

Particle image velocimetry is an experimental technique to determine the velocity field of fluid flow by cross-correlating between images captured by the high-speed camera of the tracer particles in a fluid.

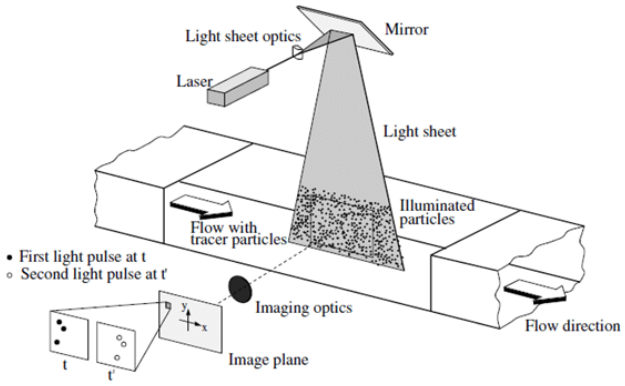


Figure 2.1: Particle Image Velocimetry Setup [11]

Figure 2.1 shows the basic experimental setup to carry out particle image velocimetry. Its setup consists of a Laser - continuous or pulsed, light sheet optics – a combination of cylindrical and spherical lens, a camera, and a post-processing unit.

(a) **Light Source:** The light source used in PIV is often a monochromatic light with a high energy density which is then converted into a thin light sheet. The high energy density monochromatic light can be obtained from Laser. Laser stands for 'Light Amplification by Stimulated Emission of Radiation', which is achieved by exciting material into a non-equilibrium state in which photons propagating through a system are amplified and emitted simultaneously. The main features to be looked upon while selecting a laser for PIV includes good beam profile – for near, mid and far-field, Gaussian beam intensity distribution, and hole-free intensity distribution. The beam from the laser is converted into light sheet using the combination of a spherical and cylindrical lens. Three different lens configurations: using three cylindrical lenses (one lens with negative focal length), using two spherical and one cylindrical lens, and three cylindrical lenses are used for this application [11].

(b) **Seeding Particles:** Seeding particles are used to track the fluid motion in a flow. These are small micron particles made from materials with same density as that of the fluid. While carrying out PIV measurements, the distance covered by the seeding particles in a small time interval Δt is measured by the cross-correlation between different images. Therefore, the measurement accuracy is greatly affected by the shape of seeding particles, their size, density ratio of the particle and the liquid, and fluid viscosity.

(i) Relaxation time

It is the measure of the tendency of particles to attain velocity equilibrium

with the fluid. For a spherical seeding particle, the relaxation time τ_s is given by:

$$\tau_s = d_p * \frac{\rho_p}{18\mu} \tag{2.1}$$

A seeding particle with a low relaxation time and smaller diameter is preferred for PIV applications [11].

(ii) Settling time

From stokes law the settling velocity of a particle can be expressed in terms of the density difference between the particle and the fluid which is given as,

$$V_T = g d_p^2 (\rho_p - \rho_f)/18\mu \tag{2.2}$$

where,

V_T = Terminal velocity of the particle

g = Acceleration due to gravity

d_p = Particle diameter

ρ_p = True density of particle

ρ_f = Density of fluid

μ = Dynamic viscosity of fluid

(iii) Light scattering behaviour

The light scattered by a particle is measured in terms of scattering cross section C_s given as,

$$C_s = P_s/I_o \tag{2.3}$$

Where,

P_s = Scattered power

I_o = Incident laser intensity on the particle

Also, [18] has given the variation of C_s as a function of the ratio of particle diameter(d_p) and incident light wavelength (λ) as shown in Figure 2.2

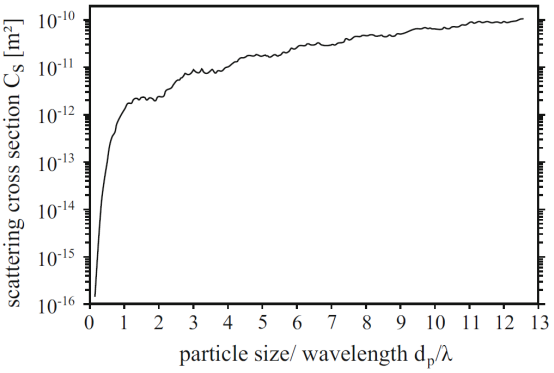


Figure 2.2: Light scattering cross section as a function of the particle size d_p [18]

The particles generally used in the case of liquid flows are listed in the table 2.1 with their mean diameter.

Table 2.1: Seeding particles in liquid [11]

Type	Material	Mean diameter in μm
Solid	Polystyrene	10-100
	Aluminum flakes	2-7
	Hollow glass spheres	10-100
	Granules for synthetic coatings	10-500
Liquid	Different oils	50-500
Gaseous	Oxygen bubbles	50-1000

2.1.2 Particle Image Analysis

PIV requires a streneous amount of work on particle image processing to generate a correct velocity field. Many algorithms have been developed so far for carrying out these task. These softwares are either proprietary or open source. The most widely accepted software in scientific community for analysing particle image are PIVlab and openPIV. Both of these software rely on the 3 steps given below:

(a) **Image Pre-Processing:** Image pre-processing is one major step in which some adjustment is made on the captured images. Sometimes the image is not perfectly illuminated with tracer particles alone. Therefore, it needs to be adjusted for background image illumination, unequal illumination of particles, and out of plane particle illumination using image segmentation and morphological algorithms.

PIVlab features various tools related to image segmentation to carry out this task. These are:

(i) CLAHE

The contrast and brightness of digital photographs can be enhanced using the common image enhancement method known as CLAHE (Contrast Limited Adaptive Histogram Equalization) designed by [19]. The fundamental principle of CLAHE is to perform histogram equalization to each tile individually after dividing the image into small sections or tiles. The key benefit of CLAHE over conventional histogram equalization is that it restricts the maximum pixel intensity value for each tile, preventing over-amplification of noise and maintaining the natural appearance of the image. By clipping the histogram at a specific threshold value, which is chosen based on the desired level of contrast enhancement, the maximum pixel intensity value is limited. The clipped values are then evenly dispersed throughout the histogram after the histogram has been equalized.

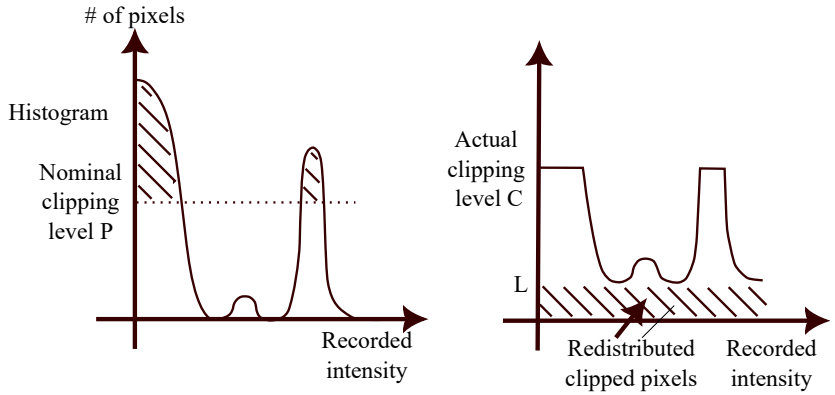


Figure 2.3: Image histogram before(left) and after(right) CLAHE [19]

(ii) Highpass filter

Highpass filter are used to conserve the high frequency information and attenuate the low frequency information in an image. The high frequency information in an image are mostly from seeding particles and low frequency information are either background noise or surface reflections. Therefore, it helps to conserve only particle information in an image.

(iii) Intensity capping

Intensity capping is the method applied to the grayscale image in which the intensity value of the pixel greater than the upper limit are replaced by the upper limit. If I_o is the grayscale median intensity of the image and σ is the standard deviation of the intensity values then the upper limit can be expressed

as $I_o + n\sigma$ where, n is user specified constant. The value of $n = 0.5 - 3$ generally yields better results after cross-correlation but to choose the optimum value, n should be selected depending on the experiment as shown in [20]. Their findings on intensity capping yields better result compared to min/max filtering, histogram equalization and CLAHE in terms of removing spurious vectors after cross-correlation and increasing the measurement accuracy with lower computational cost.

(iv) Wiener2 denoise

The Wiener2 algorithm is a digital image processing technique that removes noise from digital photographs. It uses local filtering, where each pixel is processed individually. The local window surrounding the pixel is utilized to calculate the local noise power and the signal-to-noise ratio (SNR). The local noise power is estimated based on the intensity variation of the pixel values in the window, while the SNR is calculated by dividing the local signal power by the local noise power. The calculated SNR and local noise power are then utilized by the Wiener2 algorithm to apply a filter that minimizes the mean squared error between the actual and estimated signals.

(v) Contrast Stretching

Contrast stretching enhances image contrast by increasing the dynamic range of intensity values between the minimum and maximum values of the original image. It achieves this by mapping the original intensity levels to a new range of values that can fully use the dynamic range of the display device. Both linear and non-linear functions can be used for this mapping. Linear scaling maps the minimum and maximum intensity values of the original image to 0 and 255, respectively, for an 8-bit image, and linearly extrapolates the intermediate values to fit the new range. This is known as normalization. Non-linear contrast stretching functions, like logarithmic or power-law functions, can also be used to extend specific intensity values and offer better control over the contrast enhancement process.

Apart from these algorithm, it also has the ability to subtract out the background image by calculating the mean intensity of the image frames. Sometimes, the above algorithms are not sufficient to generate a noise free particle image therefore, it has to be pre-processed before feeding it to the PIVlab.

(b) **Cross-correlation of Particle Images:** Cross-correlation is a technique used to find the similarity between two signals or images. Cross-correlation can be used in the context of image processing to determine how closely one image resembles another or a certain pattern. Cross-correlation is the most important step for car-

rying out PIV analysis. Several cross-correlation algorithms have been developed for PIV itself. PIVlab also features 3 PIV algorithms ie. FFT window deformation, Ensemble Correlation, and DCC [21, 22, 23].

(c) **Image Post-Processing:** Results after the cross-correlation contains spurious vectors that need to be removed and validated with other neighboring vectors before any further analysis is to be made. PIVLAB features 2 post-processing methods (image based validation and velocity based validation) to carry out this task. [21, 22]

2.1.3 Sources of Errors in PIV

(a) Tracer particles diameter

The optimum diameter of tracer particles to minimize the RMS uncertainty obtained from cross correlation is about 2 pixels depending upon the interrogation window sizes. Also, increasing the interrogation window size reduces the measurement uncertainty [11]. This is shown in figure 2.4.

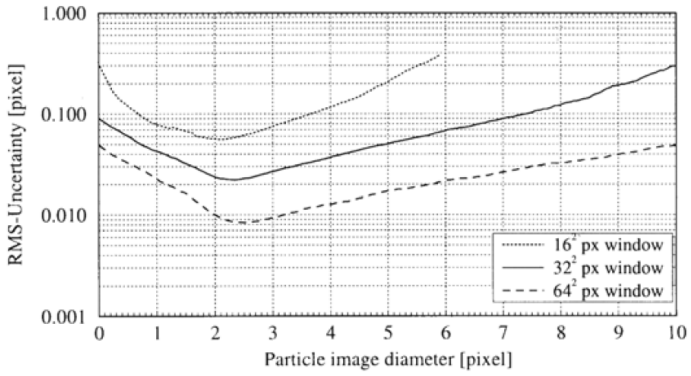


Figure 2.4: Measurement uncertainty From Tracer Particle Diameter [11]

(b) Tracer particles density

For a 32X32 interrogation window and particle diameter of 2.2 Pixels, the RMS uncertainty is lower for highly seeded than low seeded flow having same particle displacement [11]. This is shown in figure 2.5.

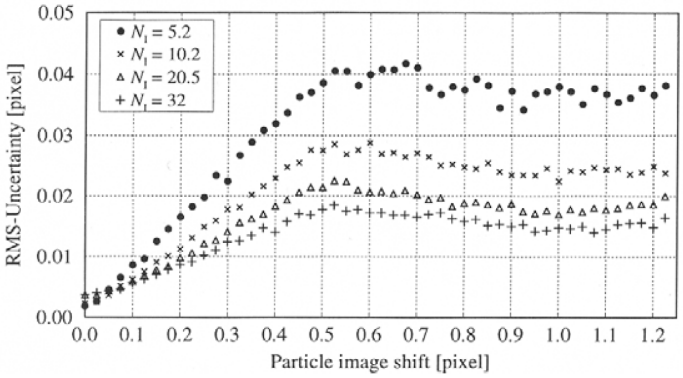


Figure 2.5: Measurement uncertainty due to Tracer Particles Density
[11]

(c) Particle Displacement

Monte carlo simulation result demonstrate the effect of particle displacement on measurement uncertainty as shown in figure 2.6. For a particle shift greater than 0.5 pixels, there is linear increase in measurement uncertainty. Also the measurement uncertainty is larger for larger particle diameter having same interrogation window and smaller for same particle diameter but larger interrogation window [11].

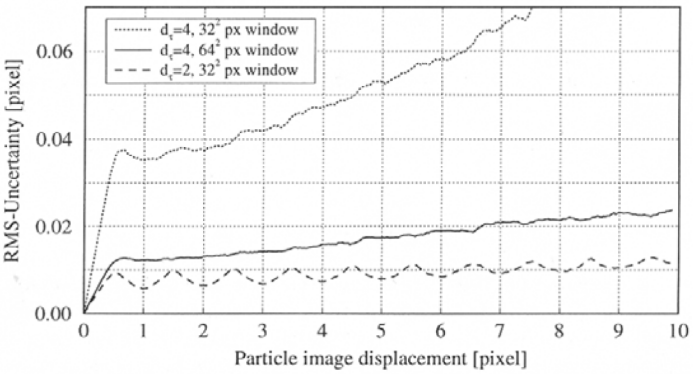


Figure 2.6: Measurement uncertainty due to Particle Displacement
[11]

(d) Displacement gradients

The effect of displacement gradient on measurement uncertainty as shown in figure 2.7. In case of lower interrogation window, measurement uncertainty reduces for highly seeded flow. But for larger interrogation windows, measurement uncertainty gets converged in both low and high seeded flow if the displacement gradient increases [11].

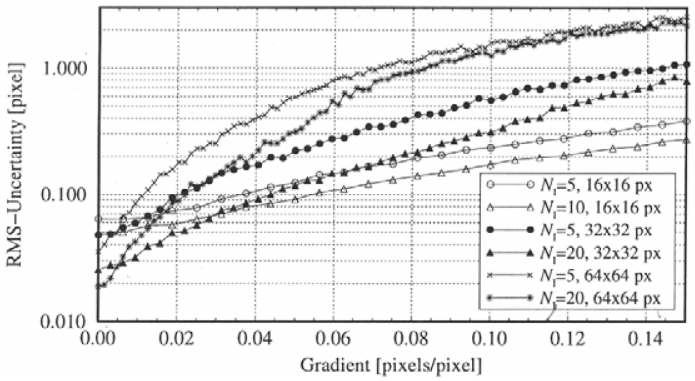


Figure 2.7: Measurement uncertainty due to Displacement Gradient [11]

2.1.4 Review on Aerodynamics

(a) **Starting Vortex** The formation of starting vortex is well explained by Kelvin's Circulation theorem. The fluid particles at the trailing edge of airfoil attains infinite velocity (theoretically) for some instant of time and tend to curl up when an airfoil dipped in fluid is initially brought to motion from rest. This small vorticity (curl of fluid particles) spread downstream as the airfoil marches forward, thus forming a vortex. This vortex is the starting vortex [24]. The major reason of the beginning of circulation around an airfoil, and hence the generation of lift due to circulation, is the formation of a growing vortex. The lift per span generated by the airfoil can be calculated from Kutta-Joukowski theorem as:

$$L^{'} = \rho_{\infty} V_{\infty} \Gamma \tag{2.4}$$

Such vortices are also necessary for studying vortex behavior downstream of a flow obstruction. To see and investigate such vortices, many approaches have been used. Aluminum tracer particles were utilized to visualize and photographically document the formation of vortex eddies [25]. Dye was employed to visualise the creation of vortices behind wedge-shaped objects at various attack angles [26], while spark shadow-graph method was use in [27].

(b) **Low Reynolds Flow:** Study of low Reynolds Number flow is important for the better understanding of micro air vehicles (MAVs). Several studies have been carried out in order to better understand the low Reynolds number flow. Sun et. al. [28] demonstrated the flow over 5% thick (of chord) flat plate at very low Reynolds Numbers ranging from 1 to 200. They suggested that the fluid does not behave as continuum at very low Reynolds number ($Re < 10$) as they have very high Knudsen

number (ratio of mean free path of fluid particles to characteristics length of flow). As a result, rarefied effects need to be considered due to which Navier-Stokes equation can't govern such flows. However, for flow with ($Re > 10$), Knudsen number is comparably low and the fluid behaves as continuum and hence can be modeled by Navier-Stokes equation.

Particulary, in our case we are interested with intermediate Reynolds number flow ($10^4 < Re < 10^5$) which we will term in upcoming literature as low Reynolds number flow.

At this low Reynolds number range ($10^4 < Re < 10^5$), a simple geometrical flat plate is more aerodynamically efficient than a scaled high Reynolds number airfoil [29]. Furthermore, Winslow et. al. [29] implied following results from their research:

- Cambered plate airfoils provide superior lift-to-drag characteristics than conventional airfoils below Reynolds number 10^5.
- Thicker airfoils provide superior lift-to-drag characteristics than thin flat or cambered plates above the Reynolds number of 10^5.
- Most airfoils' lift-to-drag characteristics cannot be expected to remain constant when the Reynolds number decreases below 10^6.
- The Reynolds number has little effect on the performance of zero-cambered flat plates.

Stevens et. al. [30] suggested that lift in low Reynolds number flow is highly effected by leading edge vortex (LEV) formation. They also studied the effect of timescales needed to achieve steady state flow starting from an airfoil at rest.

(c) **Laminar Separation Bubble:** At higher Reynolds number flow ($Re > 500000$), laminar flow over flat plate is subjected to large pressure gradient and hence separates from the surface forming a shear layer. The shear layer, however, cannot expand further because it gets energy from free stream and hence re-attaches as turbulent boundary layer, thus forming a laminar separation bubble on the surface. This boundary layer attaches along the surface till the trailing edge of the plate. Such is the case for reasonable (low) angles of attack. However, for lower Reynolds number flow ($50000 < Re < 100000$), the shear layer expands to larger bubble but still gets enough energy from free stream to re-attach as turbulent boundary layer. As a result, larger laminar separation bubble is formed. For even lower range of Reynolds number ($10000 < Re < 50000$), there is no flow separation at lower angle of attack. But as the angle of attack increases, the flow separation point moves

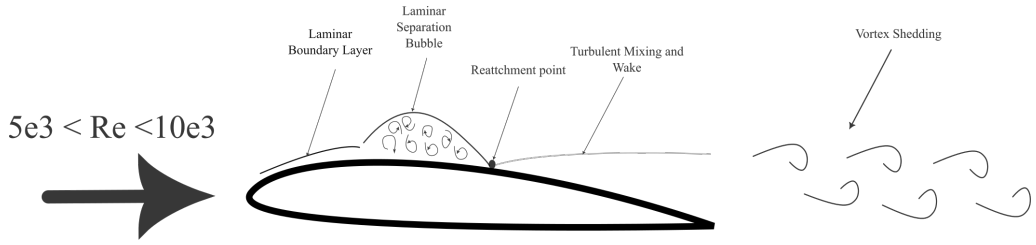


Figure 2.8: Flow over airfoil at low Reynolds number

forward towards leading edge and there is no re-attachment due to low energy of the flow [31].

(d) **Ground Effects:** Near the walls of the PIV setup, where fluid tend to remain still, if brought into motion by moving the airfoil, a phenomena similar to ground effect becomes significant. Prothin et. al. [32] studied various effects of wall present at different direction of a MAV rotor. He observed, both experimentally and computationally, that the force generated by the rotor and velocity field around it both varies a little for different configurations of the rotor and wall.

2.1.5 Computational Fluid Dynamics

The integral form of governing equations for a flat plate in incompressible viscous flow are given as,
Momentum equation,

$$\oint \rho \vec{v}\vec{v}.d\vec{A} = -\oint pI.d\vec{A} + \oint \bar{\bar{\tau}}.d\vec{A} + \oint_V \vec{F}dV \tag{2.5}$$

Continuity equation,

$$\oint \rho \vec{v}.d\vec{A} = 0 \tag{2.6}$$

where I is the identity matrix, $\bar{\bar{\tau}}$ is the stress tensor, and $\vec{F}$ is the force vector. The above equations can be solved using pressure based solvers available in ANSYS Fluent. It uses projection method algorithm to split the operators into viscous and pressure terms and hence solve the integral form of governing equation in each control volume

1. **Discretization of the Momentum Equation**

$$a_P u = \sum_{nb} a_{nb} u_{nb} + \sum p_f A.\hat{i} + S \tag{2.7}$$

Second-Order Upwind Scheme for pressure interpolation at cell faces[33]: Pressure value at the face is taken from upstream cell center pressure.

$$P_f = P_0 + \nabla P.\vec{r} \tag{2.8}$$

where P_0 and ∇P are the cell centered value and its gradient in the upstream cell. $\vec{r}$ is the displacement vector from upstream cell centroid to the face centroid.

2. **Discretization of the Continuity Equation**

$$\sum_{f}^{N_{faces}} J_f A_f = 0 \tag{2.9}$$

where J_f is the mass flux through face $f, \rho v_n$. [33]

3. **SIMPLE Algorithm:** Following steps are taken to solve the discretized equation using SIMPLE algorithm [34]:

   - Start with guessed pressure and velocity fields.
   - Solve momentum equation to obtain new velocity field.
   - Update mass flow rates using new velocity field.
   - Solve pressure correction equation using updated mass flow rates to obtain pressure correction field.
   - Update pressure and velocity fields to obtain continuity-satisfying field.

CHAPTER THREE: METHODOLOGY

The flowchart depicting the methodology of the project is shown in figure 3.1. The poject is divided into two parts i.e.the design and building of the PIV setup (Phase I) and conducting experiments and numerical simulations (Phase II).

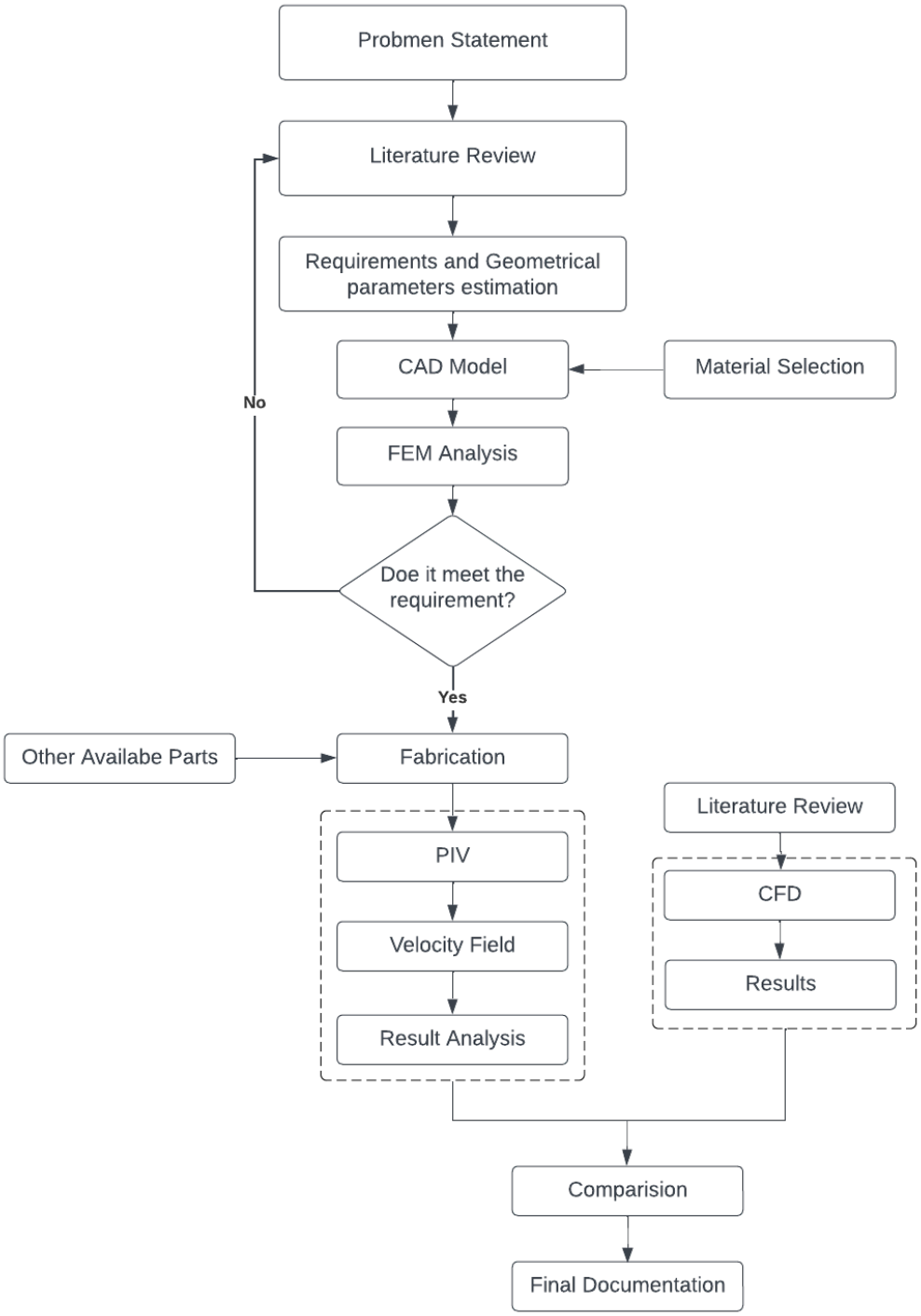


Figure 3.1: Methodology Flowchart

Figure 3.2 depicts a flow chart that explains the complete procedure involved in a flow field measurement using PIV.

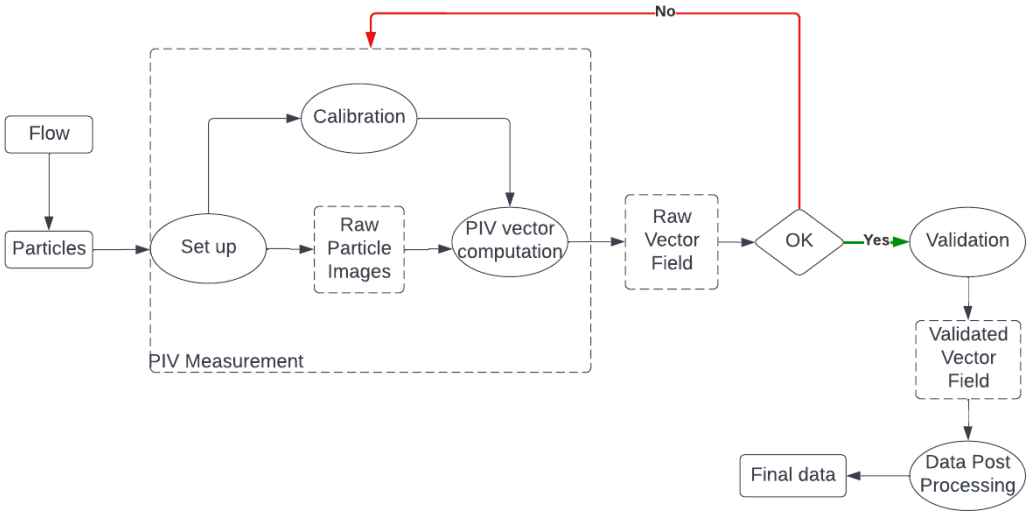


Figure 3.2: PIV Process Flowchart
(Source:"IITC - Recommended Procedures and Guidelines")

Figure 3.3 illustrates the process involved in the numerical simulation.

Figure 3.3: CFD Process Flowchart

3.1 Fabrication of the Setup

3.1.1 Design parameters

(a) Tank and plate dimensions

Tank dimensions of 2 m x 0.5 m x 0.6 m were fixed initially and based on that other parameters were calculated. The effect of blockage ratio is shown in [35]. The presence of wall will not have any effects on the pressure distribution and drag coefficient for the blockage ratios less than 6% [35]. So, the dimension of test specimen (flat plate) was calculated considering the blockage ratio of 5% as shown

below:

$$Aspect\ ratio = \frac{b^2}{S_1}$$

$$S_1 = \frac{b^2}{AR}$$

The cross section of tank is already fixed,

$$S_2 = 0.5 * 0.5 = 0.25$$

$$BR = \frac{S_1}{S_2} \Rightarrow 0.05 = \frac{\frac{b^2}{AR}}{0.25} \Rightarrow AR = 80b^2$$

But also,

$$AR = \frac{b}{c},\ where\ C = chord\ of\ plate = 0.0763m$$

$$So,\ \frac{b}{0.0763} = 80b^2 \Rightarrow b = 0.16382m$$

$$And,\ AR = \frac{b}{c} = 2.147$$

The dimensions for flat plate obtained form the above calculation is tabulated below:

Table 3.1: Dimensions for flat plate

Parameters	Value	Unit
Chord(c)	0.07634	m
Aspect Ratio (AR)	2.145	
Span (b)	0.16375	m
Planform Area (S_1)	0.0125	m^2

(b) Maximum achievable speed

The maximum achievable speed is the speed at which the flat plate moves in the stationery fluid. This depends on the type of actuation system and actuation power of the motor. Due to the constraint on the power and availability of the stepped motor, 42BYGHW811 Bipolar Stepper motor was selected. A gantry system was then used to actuate the flat plate in flow. Calculations for gantry system is tabulated below:

Table 3.2: Gantry system parameters

Variable	Value	Units
Required Speed	0.3	m/s
Pulley Diameter (D)	0.012	m
RPM of Stepper (N)	600	rev/min
Angular Velocity (w)	3769.911184	rad/min
Attainable speed	0.12	m/s

(c) Wagner and ground effects

Wagner effect on an airfoil occurs when it starts impulsively from rest. It causes circulation around an airfoil and takes some time to die out. This can lead to incorrect result therefore, needs to be considered. Figure 3.4 illustrates the wagner function plotted in terms of distance traveled in semichords for a flat plate.

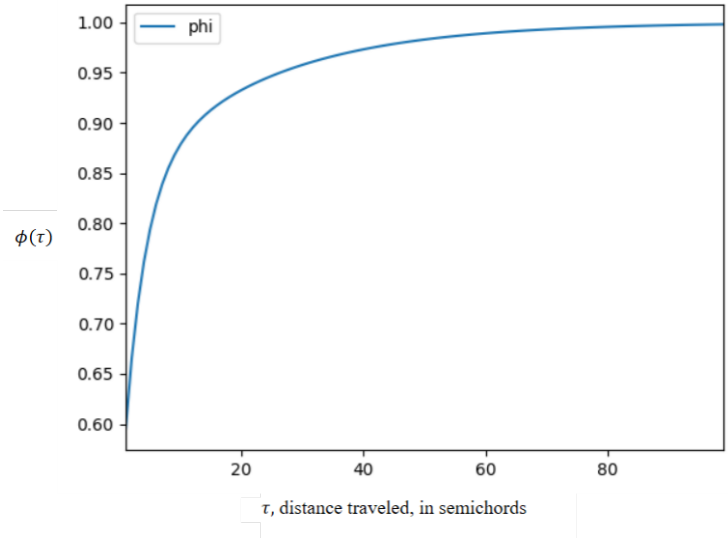


Figure 3.4: Wagner's function for an impulsively started airfoil in an incompressible fluid.

3.1.2 CAD Model

The experimental setup was divided into 3 sub-assemblies: frame, tank and gantry system. Each sub-assemblies were designed separately and final assembly of frame, tank and gantry system were carried out.

(a) Tank

Tank was designed incorporating the parameters fixed initially. Tank material of

glass and thickness of 10mm was chosen based on analysis results, cost and market research.

(b) Frame
Squared channel rod was used for frame design. The length and width of the frame were equal to the tank length and width. Other dimensions of frame can be found in appendix.

(c) Gantry
Computer Aided Design of the gantry system assembly were carried out in CATIA V5.

The overall assembly of the support, tank and gantry system was then carried out after completion of their individual designs. This is shown in figure 3.5.

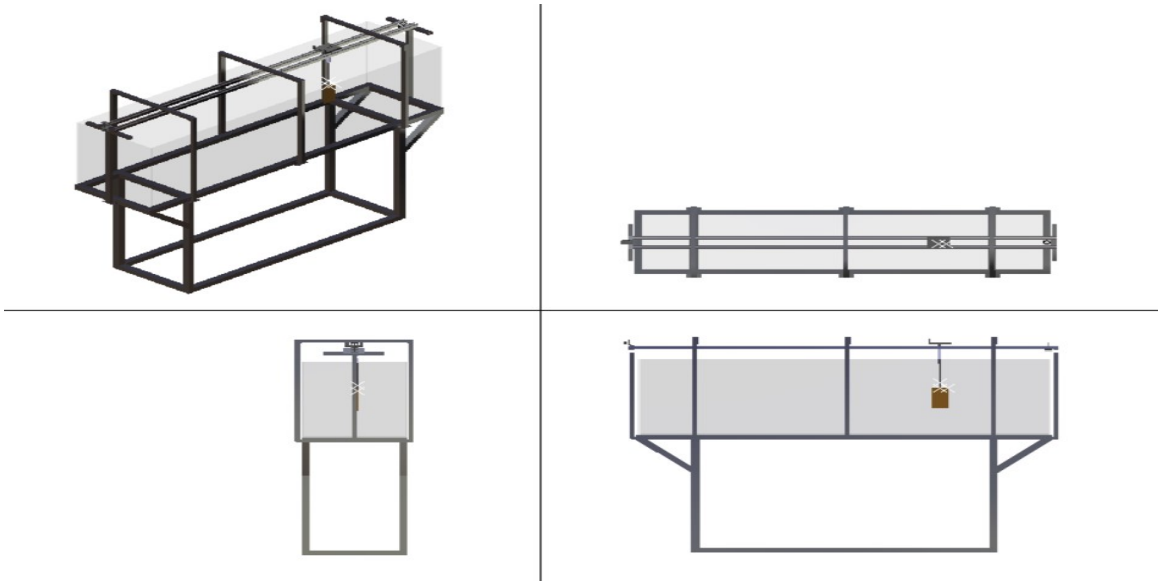
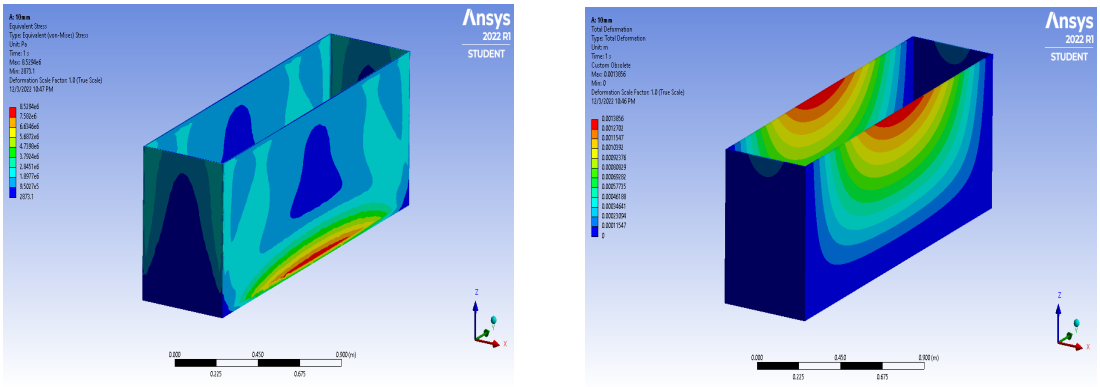


Figure 3.5: PIV Setup Assembly

3.1.3 Structural Analysis

A structural analysis was performed on a 10mm thinckness tank to simulate the effect hydrostatic pressure caused due to water. All the edges were fixed as a boundary condition and hydrostatic pressure was assigned. Total deformation of 1.38 mm was observed at the center of upper edge as shown in 3.6 and the maximum equivalent stress of 8.53 Mpa was exerted on the lower fixed edge as shown in 3.6. The safety factor was calculated to be 15. Hence, 10 mm glass was finalized for the fabrication.



Equivalent stress Total Deformation

Figure 3.6: Tank FEM Analysis

Figure 3.7 illustrates all the results obtained from the analysis for various thickness of glass.

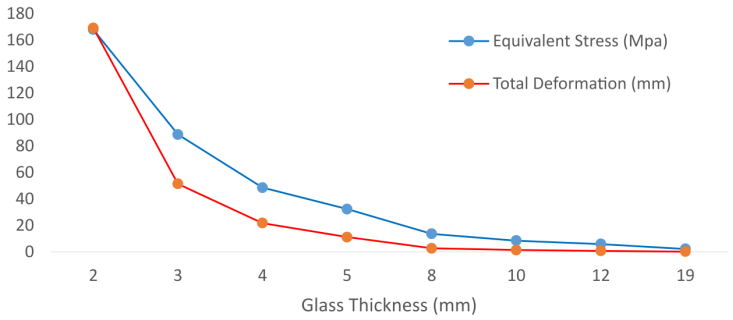


Figure 3.7: Analysis Results

3.1.4 Electronics

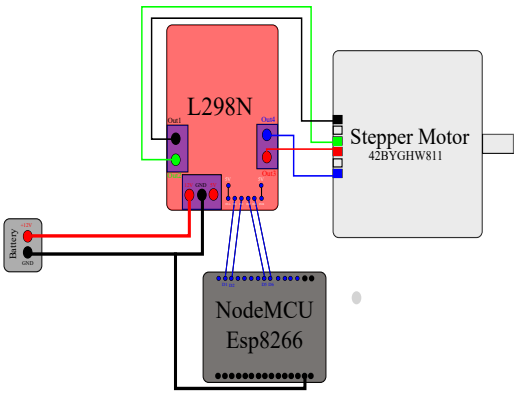
The list of components used in the setup is shown in the table 3.3:

Table 3.3: List of Components

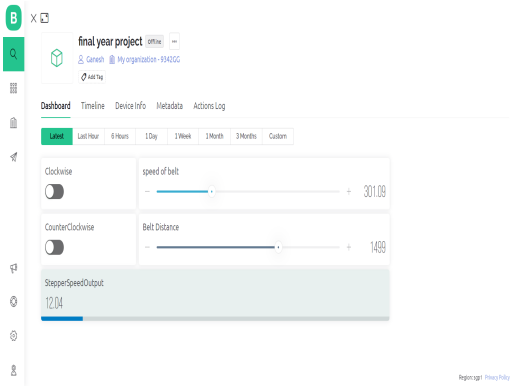
S.No	Particulars	Quantity
1	ESP8266 NodeMCU	2
2	Stepper Motor	1
3	L298N Motor Driver	1
4	Load Cell (1 Kg)	1
5	HX711 Amplifier Module	1

Stepper Motor was controlled using L298N motor driver and ESP8266 NodeMCU. A

graphics user interface was created using Blynk platform for controlling the direction and speed of the motor which is shown is figure 4.14. The code for the operation of the stepper motor can be found on appendix section.



ESP8266 with L298N Motor Driver and Stepper Motor



Graphic Interface on Blynk

Figure 3.8: Electronics setup and GUI

3.1.5 Tank Fabrication

The setup was fabricated on the basis of the cad model.

(a) Frame

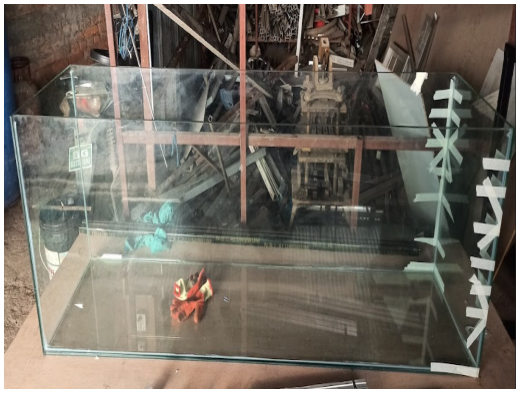
At first, 1500mm long frame was fabricated and the support were added later at the end.

(b) Glass Tank

Glass tank was made by gluing five glass sheets of thickness 10mm with silica gel. The tank was placed on the rectangular table. A 18 mm thick ply wood was placed and glass was placed on it to maintain the level. 2 aluminum dye was made to maintain the right angle position of side plates. Right angle clamp was used at the corner edges and masking tape was used all over the edge to maintain the position. Finally, after fixing the glasses in its position, silicone gel was put from inside and it was let to dry for 24 hours.

Finally silicone sealant was place from outside another day for better hold. Thus the final tank was fabricated.

Fabricated Frame



Fabricated Tank Model

Figure 3.9: Frame and Tank Fabrication

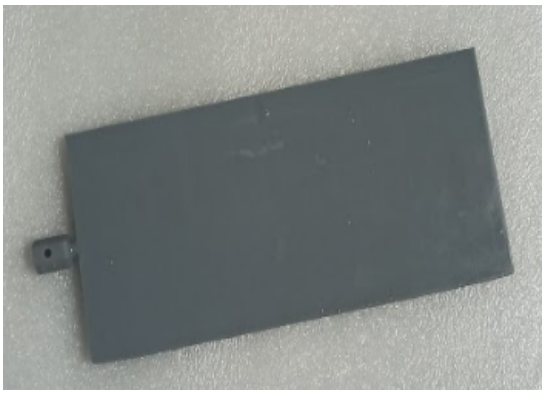
(c) Gantry System

Gantry system was fabricated using two 2100 mm long C-channel aluminium rail. Door roller was placed in the roller to move the gantry head where the load cell and flat plate is attached. Stepper motor was attached in the middle of two rail using a 2 mm sheet metal. Other end of the channel was supported using 3D printed model. Gantry head was driven by 2GT 6 mm belt. Here the belt is fixed on the curved path and is allowed to move freely on the straight path which allows the slider to move on the channel.

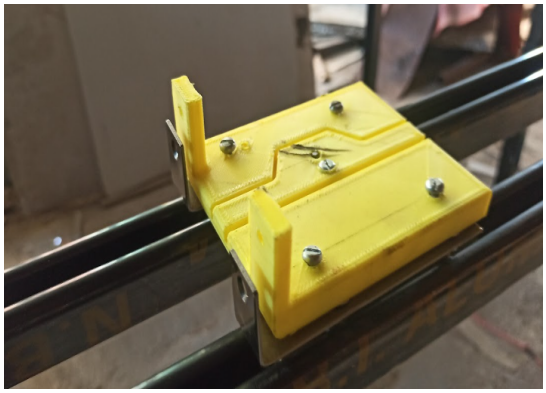
(d) 3D Print

The flat plate for experimentation was resin printed which is shown in figure 3.10.

Also the gantry slider and the end support for the C-channel was designed and printed using polylactic acid (PLA).



Resin Printed Flat Plate



PLA printed gantry head

Figure 3.10: 3D Printed Components

After completing sub-assemblies fabrication, those were assembled as shown in the figure 3.11.

Figure 3.11: Final Setup Fabricated Model

3.2 PIV Components

3.2.1 Seeding Particles

Solid glass particles of density 2500 kg/m^3 were used as seeding particles due to unavailability of precision micro-particles and import problem. These particles were collected from glass cutting industry. Therefore, they have irregular shapes and huge size range. The size range of these particles were reduced to $< 190\ \mu m$ diameter using resin filter of 190 μm. Parameters calculated for 190 μm particles as shown in 3.4 will ensure that the particles below the above size range will adequately follow the flow.

Table 3.4: Seeding particles parameters

Size(μm)	Relaxation time(sec)	Settling Velocity(m/s)
190	0.00563	0.0331

3.2.2 Laser

A class IIIB laser of output power about 5 mW provided by the Department of Mechanical and Aerospace Engineering, IOE Pulchowk was used. It has a Gaussian intensity distribution and consists of an inbuilt lens combination to produce plane laser sheet.

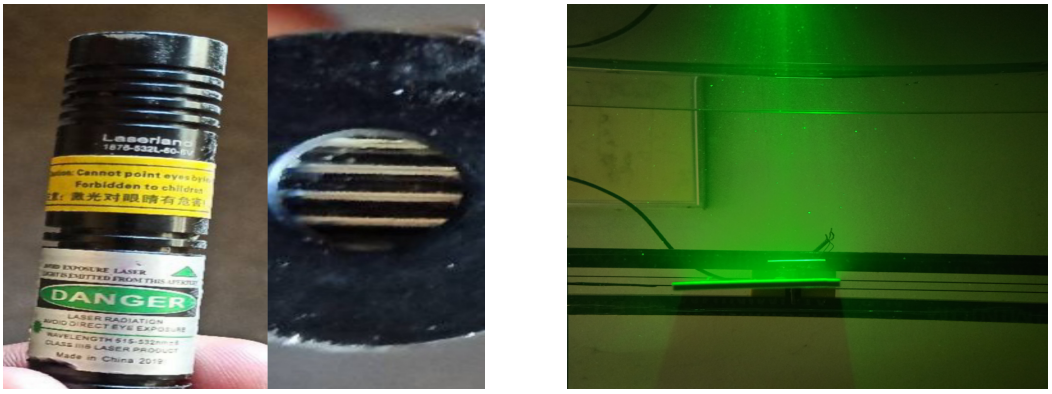


Figure 3.12: Class IIIB laser(Left) and Plane sheet(Right)

3.2.3 Camera

Two different cameras were used throughout the experimental process. They are:

- Smartphone Camera
- Chronos 2.1

3.3 Experimental Test

The whole tests were carried out over two experimental setups. One was in the small channel section and another was in the actual fabricated tank. Fluid parameters were kept same in both the experiments and are as follows:

- Fluid Density (Water, 15^{o}C): 999.13 kg/m^{3}
- Fluid Viscosity (Water, 15^{o}C): 1.138*10^{-3} Pa.s

The details about geometrical and flow parameters in both the experimental test is shown below:

(a) Test I: Test on prototype channel
Figure 4.14 shows the schematic of the test setup with camera and laser arrangement.

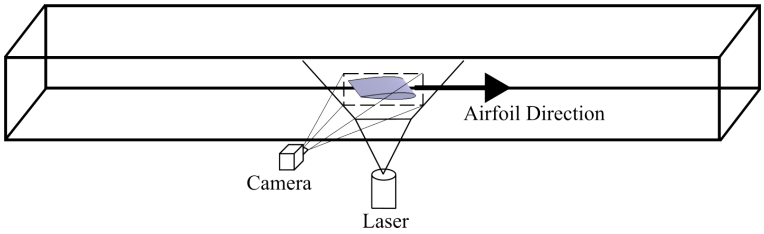


Figure 3.13: Experimental Setup

Geometrical Parameters

(a) Test specimen: Airfoil (NACA 0012)

(b) Chord: 38 mm

(c) Span: 60 mm

(d) Angle of attack: 12^{o}

Flow Parameters

(a) Airfoil velocity: 98 mm/s

(b) Reynolds number: 3270

(b) Test II: Test on fabricated tank

Figure 4.15 shows the schematic of the final PIV test setup with camera and laser arrangement.

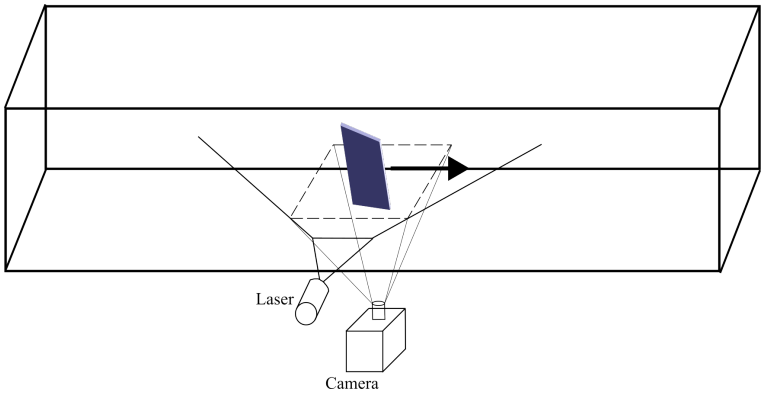


Figure 3.14: Experimental Setup

Geometrical Parameters

(a) Test specimen: Flat plate

(b) Cross section dimension: 76.33 mm x 5 mm

(c) Span: 163.75 mm

(d) Angle of attack: 10^o

(e) LE profile: flat

Flow Parameters

(a) Flat plate velocity: 90 mm/s

(b) Reynolds number: 6040

3.4 Image Generation

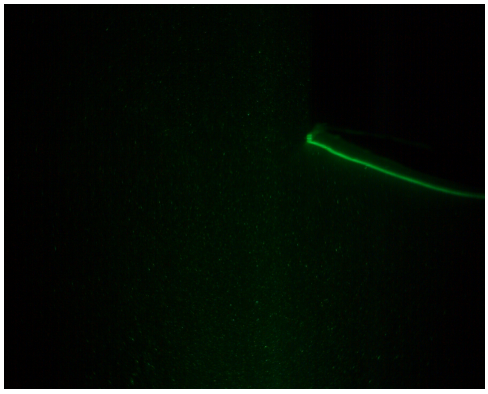
Images were taken for small channel test using Chronos 2.1 camera. The thus generated frames had frame rate of 100 fps.
For test on the fabricated tank, Chronos was unable to capture desired quality frames due to various problems which will be discussed in the upcoming sections of the report. So, smartphone camera from iPhone 13 pro was used to capture video (with better result) at 60 fps and frames were extracted from the video using ffmpeg software. However, the captured video supported up to 16 fps frame extraction after which some of the extracted frames started to duplicate. So, further processing were done for the second experiment taking the frames with frame rate of 16 fps.

3.5 PIV Processing

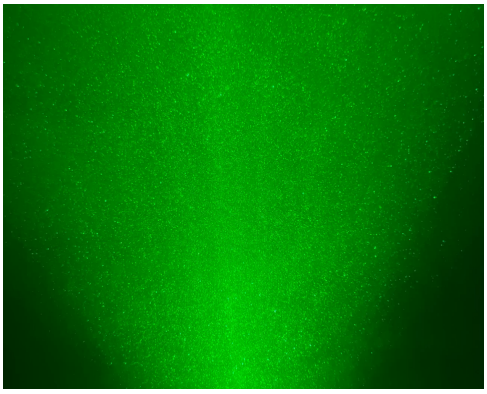
The overall PIV processing were carried out using PIVlab toolbox of MATLAB. The general procedure that was followed during PIV processing is given below.

3.5.1 Image Preprocessing

The image captured from the camera in both the tests were filled with noise from the particle dust and background illumination as shown in fig 3.15. Therefore, they need to be addressed before proceeding forward.



Raw image from Chronos 2.1



Raw image from iPhone 13 Pro

Figure 3.15: Raw images from Test I and II

3.5.1.1 Image Segmentation The inbuilt functions from the PIVLAB toolbox were used to pre-process the above images. These function given in table 3.5 operate on grayscale image of the input image to refine the image which result in higher correlation accuracy.

Table 3.5: Image segmentation functions

S.N.	Function	Settings
1	CLAHE	Window size: 64 px
2	Highbypass	Kernal size: 15 px
3	Intensity capping	Yes
4	Wiener2 denoise and Low Pass	Window size: 3 px
5	Auto Contrast Stretch	min = 0 and max = 0.00824
6	Mean Intensity Subtraction	Yes

3.5.1.2 Dynamic Mask generation Moving object like flat plate create region of non-illuminated flow where laser light is blocked and hence the seeding particle do not illuminate. These regions can interfere with the correlation map between consecutive frames and hence they need to be masked out. Masking of moving object is different to that of a stationary object as in most of PIV experiments in a sense that the required mask itself must be moving to cover the region of non-illuminated flow. There are various ways of obtaining such a mask out of which most common ones are:

1. **Edge Detection Method:** The moving object (flat plate) when illuminated by laser had higher intensity pixels in its image compared to other pixels. Hence, different edge detection methods were applied to generate suitable mask.

Sobel Edge Detection: Sobel filter based edge detection algorithm along with fill in hole algorithm for masking has been previously used in [36]. The algorithm convolves the input image with following Sobel filter kernel:

$$\begin{bmatrix} 1 & 2 & 3 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \textit{and} \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}$$

OpenCV library in python [37] was used to apply the edge detection algorithm to the range of images obtained from camera.

To generate the required mask, a fill in hole algorithm is required to fill the region within detected edge with 1 values (1 indicating white mask, 0 indicating black unmasked region). Fill-in-hole algorithm requires the detected edge to be closed boundary to work. However due to uneven illumination of the airfoil (and flat plate) and complete shadow region at one face of the plate, the closed boundary couldn't be obtained for fill in hole algorithm in our case, hence we moved to the manual masking method.

2. **Manual Masking:** Manual masking yields a good result with better mask that cover all the regions which need to be masked. However, it is a tedious process to manually mask all the frames and is very time consuming if total number of experiment and the number of frames in each experiment are both high. Hence, masks were generated using python assuming that the airfoil (flat plate) is moving with a constant velocity through the region of interest. Number of frames that the airfoil spans over was determined for each experiment and same number of images containing mask with same profile as the moving object (airfoil (flat plate)) in same orientation were generated. The thus generated masks were applied to the images to exclude undesired regions when correlating.

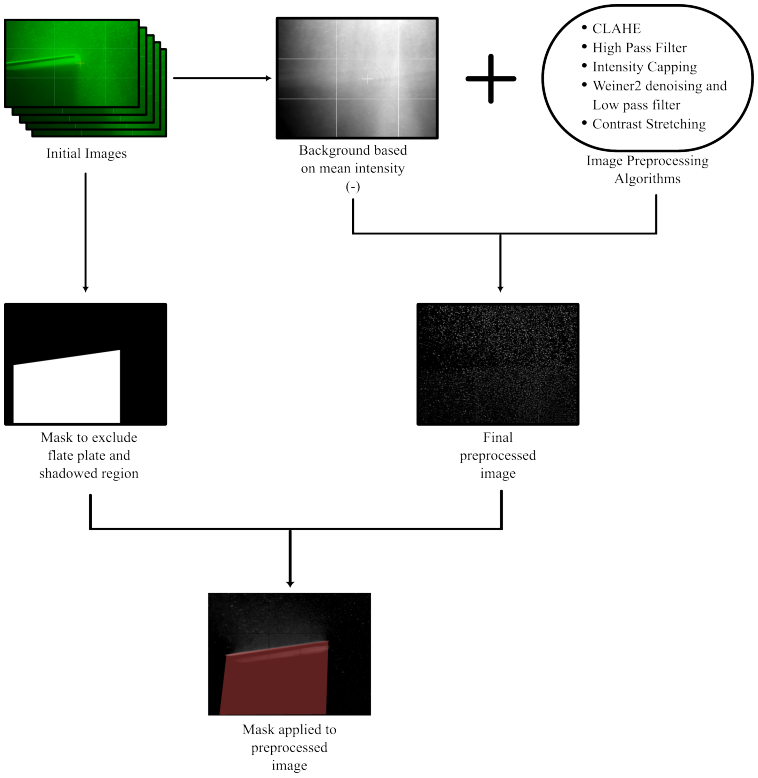


Figure 3.16: Image Pre-processing Flowchart

3.5.2 Cross Correlation

A FFT based window deformation algorithm was used to cross correlate the images in each frame. The images were correlated in multiple passes as shown in table 3.6 to increase the accuracy of the result. To further estimate the particle transition accurately, Gauss 2X3-point sub-pixel estimator was used. Finally choosing correlation robustness to standard and disabling auto correlation gives the result as shown in fig 3.17

Table 3.6: Multi-pass FFT settings

Pass	Interrogation area [px]	Step [px]
1	64	32
2	32	16
3	32	16

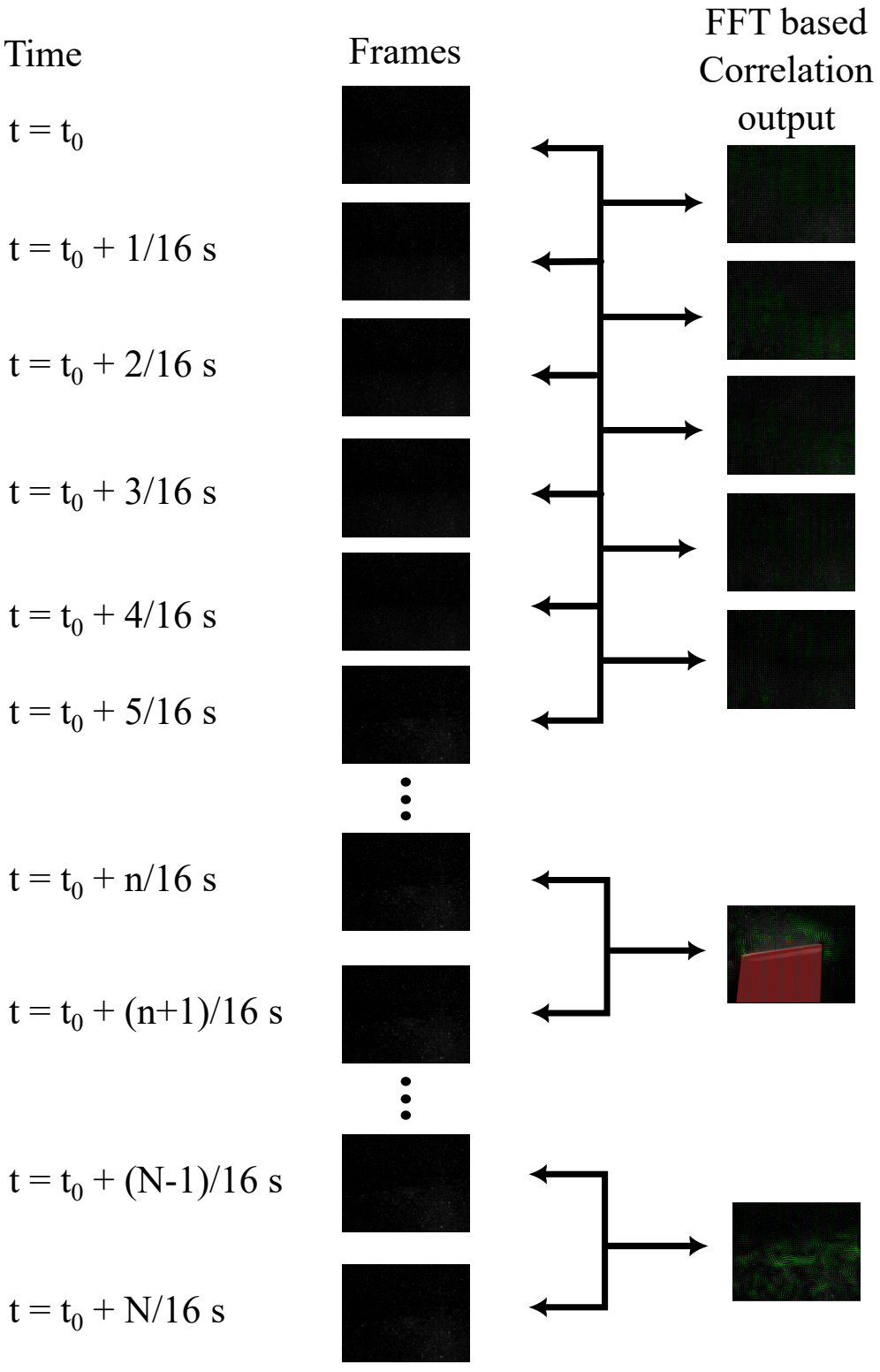


Figure 3.17: FFT window deformation result

3.5.3 Image Post-Processing

The velocity field generated by correlating consecutive frames contained erroneous vectors which needed to be filtered out. PIVlab provides velocity based vector validation and image based validation methods to filter out such undesirable vectors. Velocity based vector validation method was employed to exclude the outlier vectors from the vector field. Empty regions of such excluded vectors were interpolated with nearby vectors to compute the complete vector field. Furthermore, since the initial flow before airfoil enters the ROI was not stationary, it will be a source of background noise and will induce error in the final velocity field. Hence, mean velocity of starting frames were subtracted to minimize error from the final velocity field.

3.6 Numerical Simulation

3.6.1 Problem definition

(a) Geometry

The geometry for flat plate was designed in CATIA and imported to Ansys.

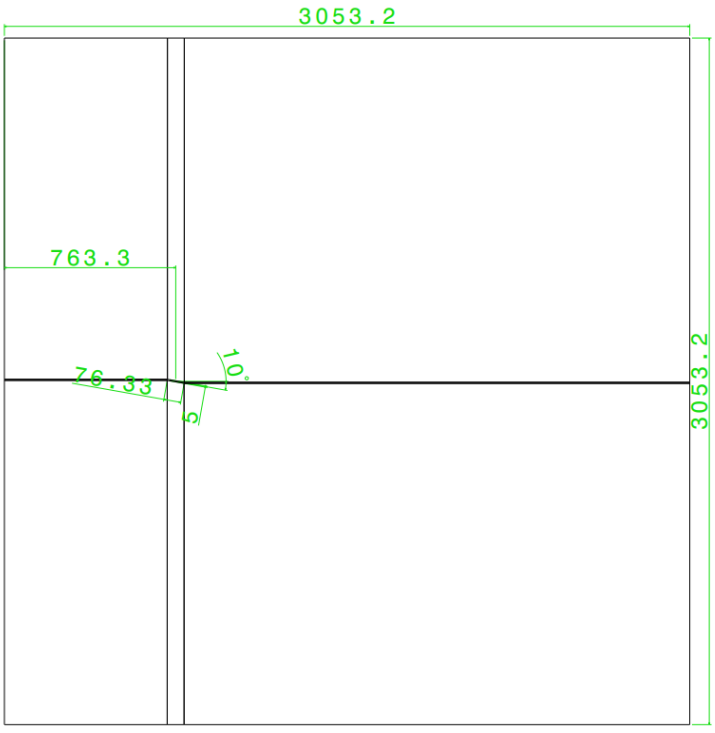


Figure 3.18: Flat Plate Domain for Numerical SImulation

(b) Boundary Conditions

The boundary conditions for both flat plate and airfoil is given in table 3.7

Table 3.7: Boundary Conditions

Parameters	Flat Plate	Airfoil
Velocity inlet	9 cm/s	10 cm/s
Pressure outlet	0 atm	0 atm
Operating Pressure	2553285 Pa	101325 Pa
Walls	Plate, Top, Bottom	Airfoil, Top, Bottom

(c) Fluid properties

The experiment was carried out in water therefore the fluid property is of water.

3.6.2 Mesh Generation

The structured mesh was generated which is shown in figure 3.19.

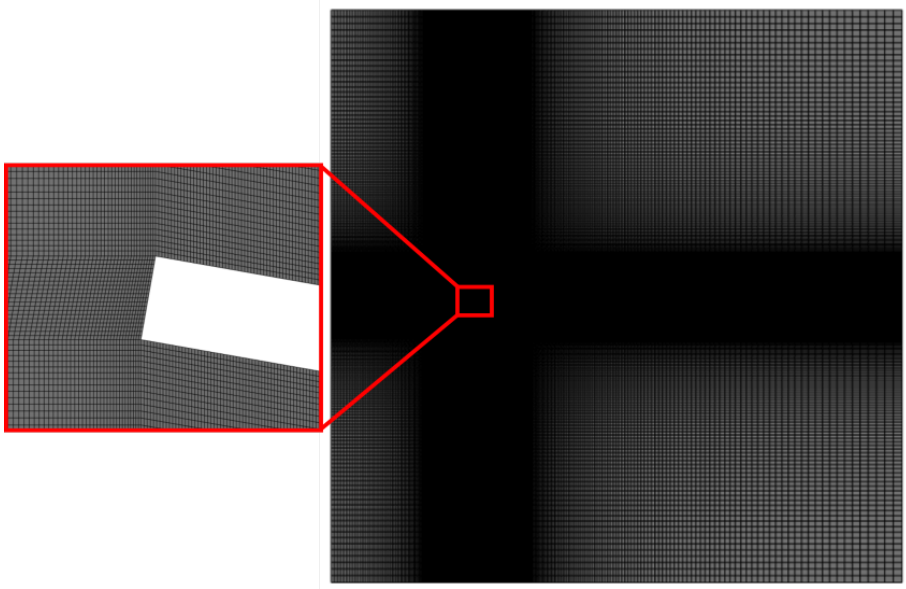


Figure 3.19: Flate Plate meshing for Numerical Simulation

The information about mesh resolution is given in table 3.8

Table 3.8: Mesh Resolution

Parameters	Flat Plate
Nodes	350690
Elements	349000

3.6.3 Setup

1. Model selection
   The laminar model which works on projection method numerical scheme was selected for both cases.

2. Solver setup
   The solver setting used for both the cases is given in table 3.9

Table 3.9: Solver Settings

Parameters	Flat Plate	Airfoil
Pressure velocity coupling	SIMPLE	SIMPLE
Pressure	Second order discretization	Second order discretization
Momentum	Second order upwind	Second order upwind
No. of time steps	10000	1000
Time step size	0.001	0.01
Solver type	Transient	Transient
Convergence criteria	error(ε) < 0.001	error(ε) < 0.001

CHAPTER FOUR: RESULTS AND DISCUSSION

4.1 Seeding Particles Concentration Result

After the selection of solid glass powder as seeding particles, particle concentration were estimated. This was done by adding seeding particles of known quantity in a small tank and capturing the image until the interrogation window of 64*64 pixels is seeded enough with the particles. The result obtained is given below:

- Small tank dimensions = 25m*20m*26m
- Water Volume = $2.25 * 10^{-3} m^3$
- Particle size = 2-4 pixels at resolution of 1920×1080
- Interrogation window = 64*64
- MinMax filter: [1]
  - Kernel size: 7*7
  - Minimum contrast: 15

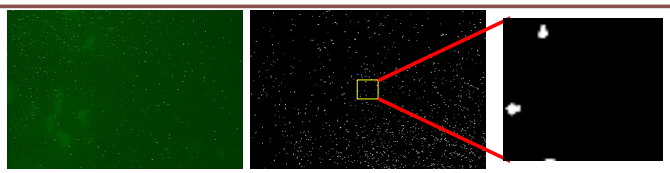
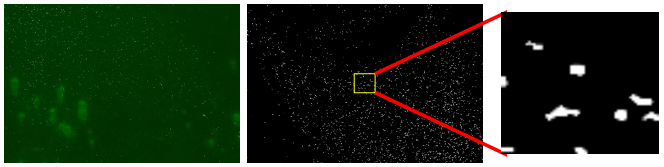
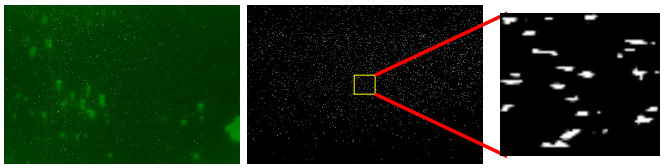
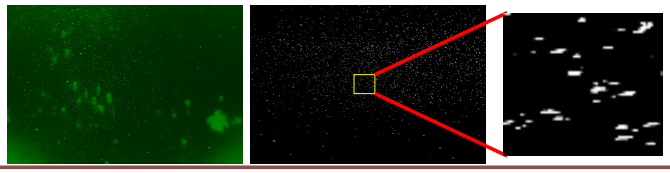
S.N	Raw Images	MinMax filter	Interrogation window (64*64)	Particle Count	Seeding Particle Wt. (gm)
1				3	0.009
2				8	0.0135
3				15	0.018
4				18	0.0225

Figure 4.1: Particle concentration

Hence, the 0.0225 gm weight of particles was selected for 2.25 litres volume of water. It contained 18 particles in the selected interrogation window of 64*64 pixels. After the determination of particle quantity in small scale, equivalent amount of particles were added to large tank model and some refinements were done by manually viewing the image. Therefore, the total particles required for large tank model of volume 0.5 m^3 was 5 gram.

Furthermore, a simple observation based test was to quantify the out of plane motion behaviour of the particle. This was done to compensate the inability of limiting the diameter range of the particles and its consequences in the out of plane motion. 5 pinches ($\approx 0.0225\ gm$) of particles were stirred in a small test section with 2.25 litres of water and left to settle for 10 min. Random particles were traced by eye until it disappear from the laser illuminated plane. The observation had following considerations:

- Only highly illuminated particles were traced.
- Particles might have been in the plane before the start of eye-tracing. So, actual

time is greater than recorded time.

- Subjected to small external disturbances.
- Particles of only midsection of tank were traced.

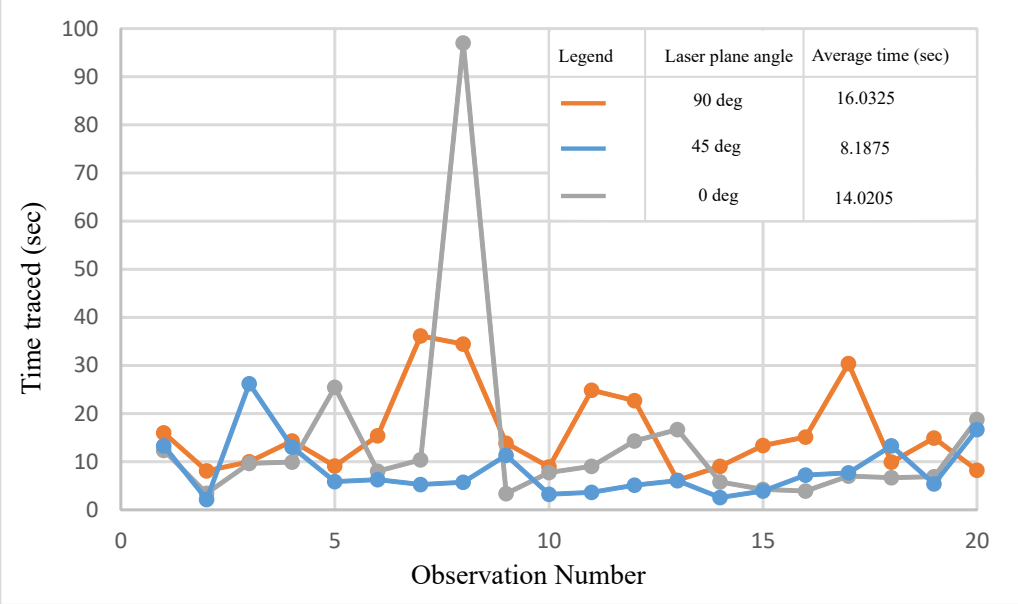


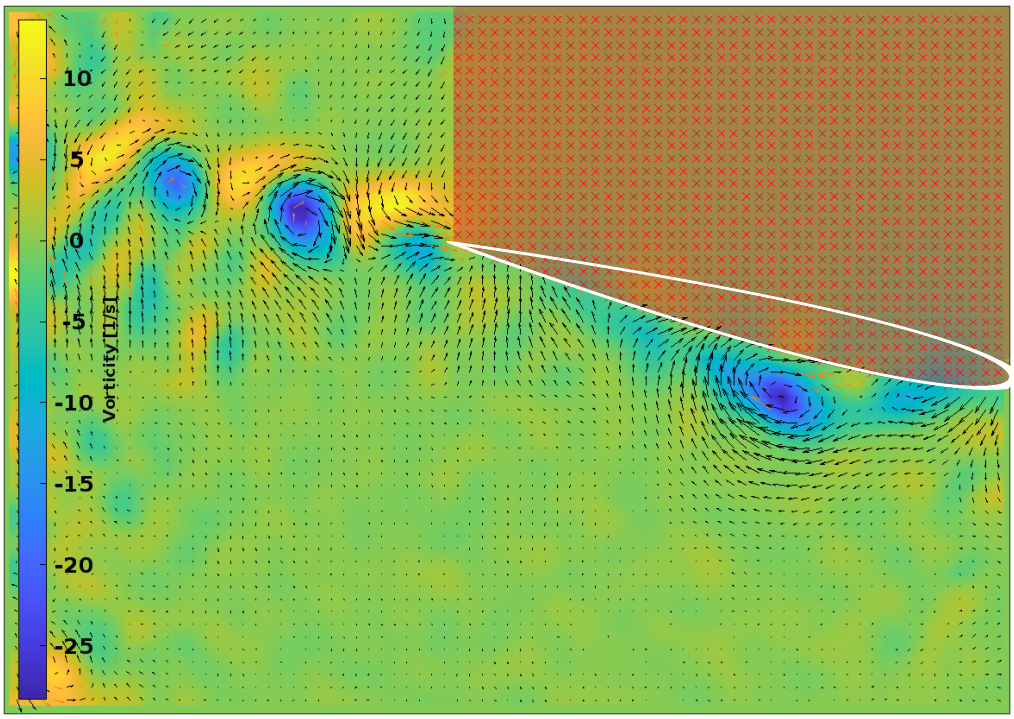
Figure 4.2: Time for which randomly selected twenty particles stay in the laser illuminated plane.

4.2 PIV Result on Test Specimen

The extracted frames were imported to PIVlab for cross-correlation between the consecutive frames. First, the images from tests in prototype channel were imported in PIVlab in time resolved format and were processed using above mentioned settings. Correlation coefficient was taken as the major parameter for quantifying uncertainty. The results from the test are presented in subsection 4.2.1. Then, experiment was carried out on the main fabricated tank. The results from the main experiment are presented in subsection 4.2.1

4.2.1 Test on prototype channel

FFT window deformation correlation with above mentioned settings was employed for correlating the sequential frames. Vector fields generated after the correlation are shown in following figure. Leading edge separation bubble formation being an important phenomena can be visibly observed in low Reynolds flow.

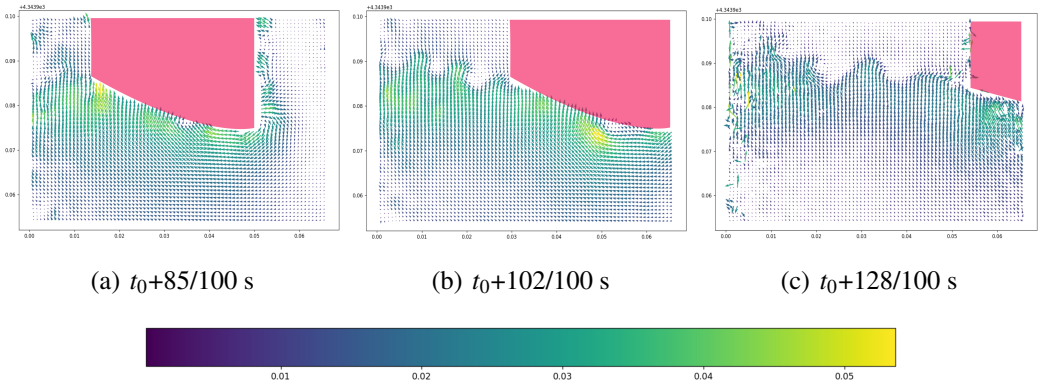


(a) Experimental PIV

Figure 4.3: Velocity (m/s) field around aifoil and vorticity contour at α = 12 degree, airfoil moving from left to right (mask applied at shadowed region)

Figure 4.3 shows the velocity field around the airfoil at α = 12 degrees. A region of circulation near leading edge above the airfoil can be seen which indicates the presence of separation bubble above the airfoil. Quantities such as separation bubble length and reattachment points could not be identified precisely due to lack of high resolution vector field. This limitation is due to various factors which will be discussed in further sections of the report. The flow is turbulent after reattachment with the airfoil and finally separates from trailing edge shedding vortices downstream the TE.

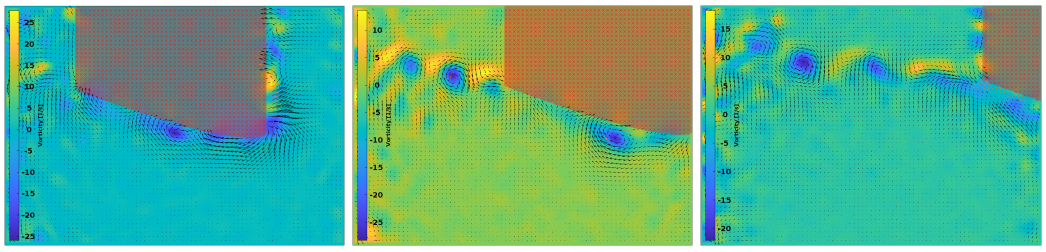
Figure 4.4 shows three different plots of velocity vectors at three different times. The first figure 4.4(a) shows the flow stabilizing over the airfoil. The second figure 4.4(b) shows the fully developed flow over the airfoil and the start of vortex shedding. The third figure 4.4(c) highlights the shedding of vortices originating from the trailing edge of the airfoil.



(a) t_0+85/100 s (b) t_0+102/100 s (c) t_0+128/100 s

Figure 4.4: Velocity (m/s) Vector field around the airfoil,
airfoil moving from left to right (mask applied at shadowed region)

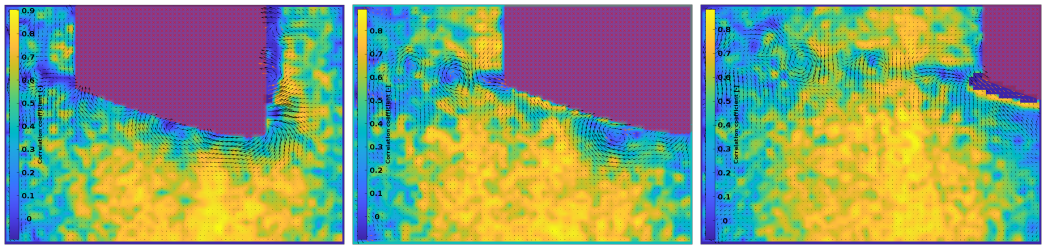
The vorticity contour shown below highlights the region of vortices in the flow. The first frame shows the flow getting stabilized over the airfoil. The flow is laminar and the bubble starts to appear. When the flow has moved further downstream along the airfoil as seen in the second figure, the separation bubble becomes clearly visible. The bubble reattaches to the airfoil surface forming a pocket of swirling fluid particles with non-laminar behavior. Beyond the trailing edge, fluid particle curls around the edge forming trailing edge vortices. Thus formed vortices shed downstream the flow forming commonly known vortex shedding pattern. The third picture shows all three phenomenon: laminar separation bubble, its reattachment and trailing edge vortex shedding in a single frame.



(a) t_0+85/100 s (b) t_0+102/100 s (c) t_0+128/100 s

Figure 4.5: Vorticity contours around the airfoil,
airfoil moving from left to right (mask applied at shadowed region)

The velocity field is generated by correlating image windows between consecutive frames. The reliability of such velocity vectors is directly dependent on how the two image windows correlate with each other when shifted by certain pixels. Hence, correlation coefficient becomes an important parameter in quantifying the reliability of generated vector field. Figures below shows the contours of correlation coefficient.



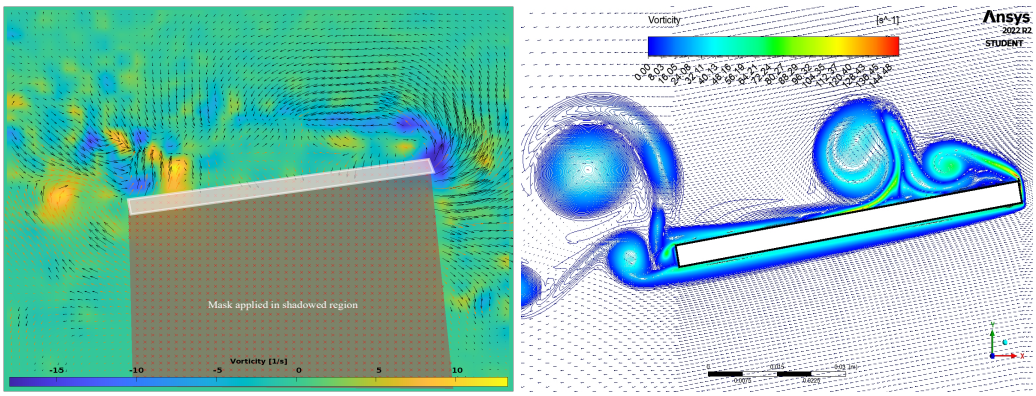
(a) t_0+85/100 s (b) t_0+102/100 s (c) t_0+128/100 s

Figure 4.6: Correlation coefficient contour plot,
airfoil moving from left to right (mask applied at shadowed region)

It can be seen that the vectors farther from airfoil are accompanied by high correlation coefficient (> 0.5). However, at regions of swirls and vortices, correlation coefficient value is fairly low (< 0.5). This may be due high displacement gradients of the particles which are difficult to capture with low speed frames. The particles escape from the correlation window during mapping due to its higher velocity, so the windows don't correlate well, hence lower correlation coefficient. Overall, flow visualization and velocity vector field computation on prototype channel test were satisfactory. Major low Reynolds number flow phenomenons like separation bubbles, reattachment points and vortex shedding from the trailing edge of airfoil were captured. Hence, we moved onward to experiment the flow over flat plate using fabricated PIV setup.

4.2.2 Test on fabricated PIV setup

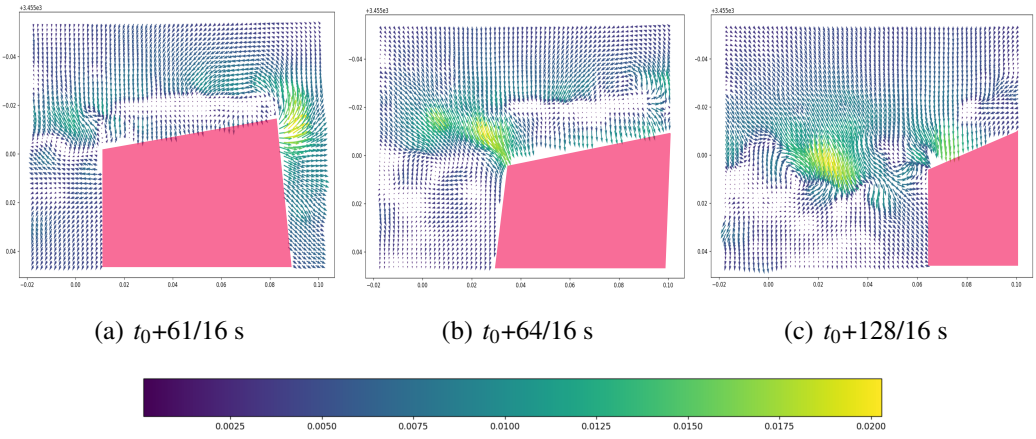
The low Reynolds flow over flat plate is also accompanied by with a region of high vorticity immediately behind the leading edge above the flat plate as shown in figure below. Due to insufficient particles and uncorrelated windows, the section of reattachment is not clearly visible. The vortex is also larger in size compared to that in airfoil. This is due to sharp turning of the flow around the leading edge in the flat plate. Figure below shows the vector field over the flat plate.



(a) Experimental PIV (b) Numerical Simulation

Figure 4.7: Velocity (m/s) vector field and vorticity contour at $\alpha = 12\,\text{deg}$, flat plate moving from left to right (mask applied at shadowed region)

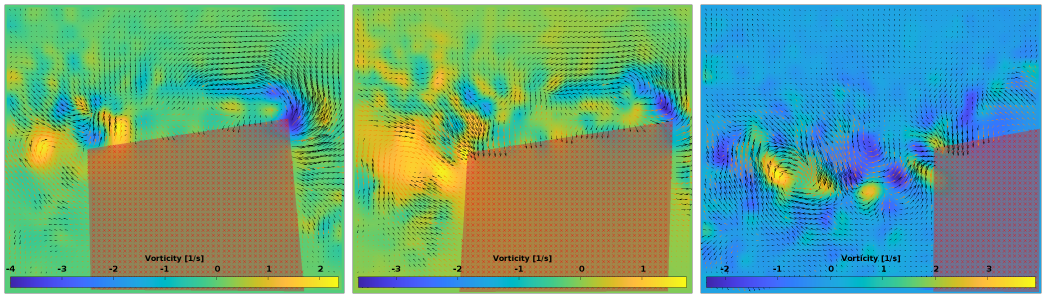
Figures below are the quiver plots of velocity field around flat plate at three different time steps. The frames denote the stabilizing flow over the plate, fully established flow over the plate and the trailing edge vortex shedding pattern. The trailing edge vortex pattern in the third frame also suggests that the laminar bubble has re-attached and the flow separates from the trailing edge giving rise to trailing edge vortices.



(a) t_0+61/16 s (b) t_0+64/16 s (c) t_0+128/16 s

Figure 4.8: Velocity (m/s) Vector field around the flat plate, flat plate moving from left to right (mask applied at shadowed region)

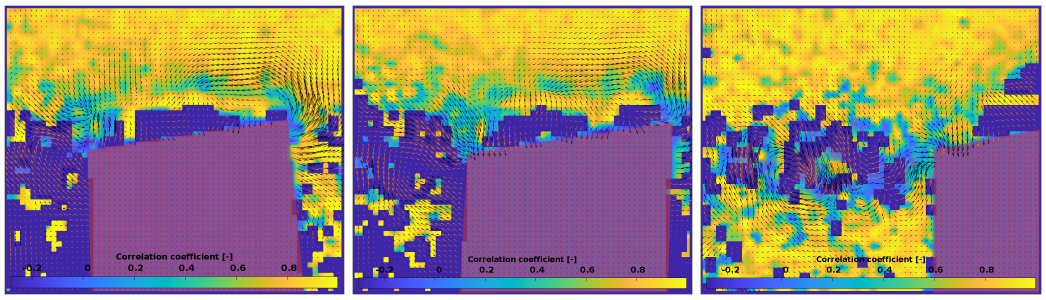
Figures below show vorticity contour for the three frames. The contours further support the argument that the region behind the trailing edge are vortex dominant region and there is a pocket of fluid with high vorticity just behind the leading edge of the plate.



(a) t_0+61/16 s (b) t_0+64/16 s (c) t_0+128/16 s

Figure 4.9: Vorticity contours around the flat plate,
flat plate moving from left to right (mask applied at shadowed region)

The figures below shows the correlation coefficient contours for the three frames. Velocity fields above the flat plate contain vectors that result from highly correlated frame windows and hence are fairly reliable. However, at the region of boundary layer just above the flat plate, the correlation coefficient value is very low ($0 < R < 0.5$) and are even negatively correlated (< 0). Thus the boundary layer information is generated with interpolation schemes and are not completely reliable.



(a) t_0+61/16 s (b) t_0+64/16 s (c) t_0+128/16 s

Figure 4.10: Correlation coefficient contour plot,
flat plate moving from left to right (mask applied at shadowed region)

4.3 Errors and error sources

1. **Fabrication induced errors:**

   (a) Non-linear input velocity profile due to non-linear movement of the gantry system.

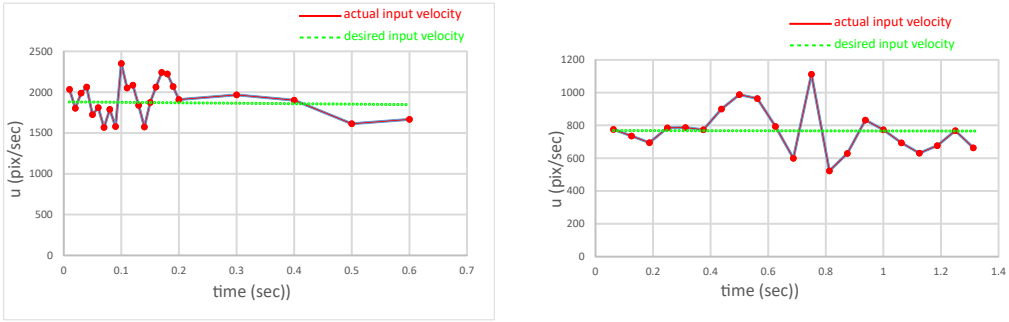


Figure 4.11: Input velocity profile and desired velocity profile, left: prototype channel test, right: fabricated tank test

Vibration induced from stepper motor used to drive the gantry head and uneven rolling surface of roller. Micro-stepping the motor or using brushless DC motor. Reduce gap between roller and trail.

(b) **Glass Refraction:** Another possible error might be due to bending of laser sheet while passing from one medium to another i.e (air to glass to liquid). This can cause misalignment between laser plane and camera.

2. **Laser insufficient illumination:** The laser used in the experiment was highly diverging and of low power than required for the PIV experiment. Hence, particles illumination was insufficient. Furthermore, the laser sheet from the laser was thick (3 mm) and hence added to the background noise in the captured frames.

3. **laser non-uniform illumination:** The light source must be of top-hat profile for uniform illumination of the particles within the region of interest. Gaussian profiled light source nearly approximates the top-hat profile with some errors in approximation. The light source in our case had very high illumination in the central region while the diverging sections had rapidly decreasing illumination so, could be arguably approximated as a Gaussian profile with low sigma value. This added to the error in pre-processing and cross correlation.

4. **Seeding Particle:** Locally available glass powder was used as seeding particle for the experiment. The particle had various undesirable properties such as:

   (a) High density (2500 kg/m^3)

   (b) Irregular shape and size: An attempt was made to bring the particle size near to desired dimension (diameter 50 μm) by filtering with 190 μm filter paper. The particle had no lower limit in its size and hence the smaller particles added to the noise. Furthermore, non-spherical particles exhibit reflection in random direction which added to the noise.

(c) Non-fluorescent particle: The particle had lower fluorescence and hence didn't reflect properly.

5. **Camera:**

   (a) Smartphone Camera: Supports low frame rate only upto 30 fps for screen recording. Furthermore, the image extraction from the recorded video only supported upto 16 fps after which frames started to duplicate.

   (b) Chronos 2.1: Unable to focus to the laser sheet due to insufficient illumination as required by the camera.

6. **Image processing:**

   (a) **Preprocessing:** The frames had very high background noise from various sources and needed to be removed to send for correlation. This process induced error in the correlation. Sources of background noise are:

      i. Small diameter seeding particles
      ii. Initial particle motion
      iii. Background illumination
      iv. Particle voids
      v. Highly illuminated airfoil (flat plate) surface

   (b) **Correlation errors:** Various factors added to the error in correlation between consecutive frames. Some of the factors are:

      i. Particle density and concentration
      ii. Particle size and image resolution
      iii. Particle shift per frame
      iv. High particle velocity gradient
      v. Settling velocity
      vi. Motion Blur

7. **Information Loss:** Due to laser illumination, the particles flowing near the wall surface were overshadowed by the laser reflecting from the surface. Therefore, the image segmentation algorithm were not able to reconstruct particles at those position due to very high background illumination. This resulted in loss of information at the near wall surface. Also, near the leading edge particle density greatly reduced due to centrifugal effect on particles. Therefore, low particle density at those spaces required larger interrogation window during cross correlation and eventually loss of information.

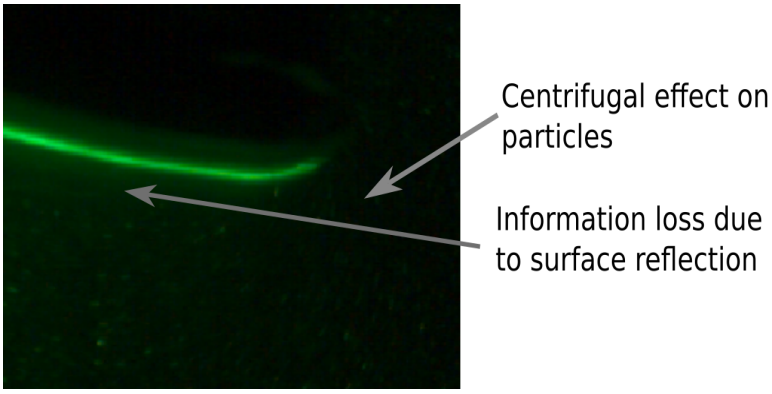


Figure 4.12: Information Loss in PIV Images

Another reason for erroneous vector field in the flow was due out of plane motion and irregular shape of the particles. Due to irregular shape the particles that has highest illumination in one frame either completely disappeared or is merely visible in an another frame. This was because in one frame particles with largest surface area was illuminating but as soon as the orientation changed in another frame illumination intensity also reduced. Out of plane particle motion can also be seen between two consecutive frame due to sudden disappearance of particles. This two effect was clearly visible in those areas where flow gradient was high.

4.4 Limitations

- The maximum achievable speed of the gantry is 0.1 m/s due to motor power limitation which limits the setup to support the study over low Reynolds number flows only.
- Calculations based on the assumption of properties of fluid at normal temperature.
- The study is limited by any sort of flow instabilities and fluctuations during experiment, which might have affected the accuracy and reliability of the data.
- Overall, the study is limited by all the error sources mentioned in previous section.

4.5 Problem Faced

The problems faced during the project timeline are listed below:

- Non uniform illumination of the glass particles due to its irregular shape and size.

- Focusing camera on the illuminated plane was difficult due to the particle used and limited laser power.

4.6 Budget Analysis

The total expense of the project is shown in the table below:

Table 4.1: Expenses

S.No	Particulars	Rate(Rs.)	Quantity	Price(Rs.)
Electronics				
1	L298N Stepper Driver	500/-	2	1,000/-
2	Stepper Motor Connector (6 Pin)	400/-	1	400/-
3	HX711	150/-	1	150/-
4	Jumper Wires	330/-	-	330/-
5	2GT-6 mm Belt	160/-	5	800/-
6	Pulley 2GT 8 mm long	160/-	2	320/-
7	Glass	500/-	50 Sq. Ft.	25,000/-
8	Aluminium C channel	-	-	2,000/-
9	Roller	1,000/-	1	1,000/-
10	1.5” square rod	-	-	5,000/-
11	1” square rod	-	-	3,000/-
12	1” angle rod	-	-	1,500/-
13	Welding Rod (3.2x300)	1,200/-	1	1,200/-
14	Cutting Wheel	-	-	1,000/-
15	Silicone Sealant	700/-	3	2,100/-
16	Nut Bolt	-	-	1,000/-
17	Paint	500/-	-	500/-
18	Man hour	-	-	10,000/-
19	Miscellaneous (Tarpin Oil, Tape, etc.)	-	-	3,000/-
	Total			59,300/-

CHAPTER FIVE: CONCLUSION AND FUTURE ENHANCEMENT

5.1 Conclusion

Successful fabrication of the test setup was completed to perform low Reynolds flow (Re<3,000) experiment using Particle Image Velocimetry technique. At first, a previously available acrylic flume prototype channel was utilized to test the locally available glass particles for its illumination using available laser source of 5mW and to identify the camera location. Also, the concentration for the available glass powder was estimated to reduce the error in cross correlation.

Vortex shedding and laminar separation bubble were visualized for an airfoil at an angle of 12 degrees using the prototype setup. Velocity and vorticity plots were generated and plotted based on the experiment. Correlation coefficient was plotted using PIVLab which helped identify the location of uncertain correlation and quantify the reliability of experiment. Flow over a flat plate at 10 degree angle of attack was experimented in the final setup and images were processed in PIVLab. The laser could not illuminate the glass particles due to its low power and irregular shape and size of the particle and its concentration. Results were obtained showing the leading edge separation bubble and reattachment point however most of the information were lost due to different error sources. This showed various room for improvements in the setup and experimentation.

Furthermore, numerical simulation of both the airfoil and the flat plate were conducted with similar Reynolds number as in the experiment. The simulation results were compared with that of the experiment. There was noticeable bias between the results from numerical study and experimental PIV. The different velocity and vorticity profiles as observed from the experimental results are due to the non-steady nature of input velocity in the experiment as shown in previous section. This might have caused the flow to behave in more complex way than it would have been for a steadily moving airfoil or flat plate. Other error sources as discussed in the previous section might also have added to the cause.

5.2 Scope for Future Enhancement

The setup aims to facilitate the experiments at low Reynolds Number (Re<30,000). The experimental setup can be further modified to make it more accurate in its function. Sug-

gestions for possible future work include:

1. Developing more robust and vibration free gantry system.
2. Incorporating AOA changing mechanism of the test specimen.
3. Improving speed range of the test specimen by microstepping the current motor driver or using brushless DC Motor.
4. Filtration of glass particles to a narrower diameter range to reduce noise in the image.
5. Improving the data acquisition system with advanced user friendly interface or visualization tool.
6. Further validation of the system with large experimental data sets.
7. Use of commercial grade PIV tracer particles and high power laser for experimentation.
8. Integration of loadcell into the setup as an alternative acquisition of lift force data and hence a validation method for the PIV results.

REFERENCES

[1] R. J. Adrian and J. Westerweel, *Particle image velocimetry*. No. 30, Cambridge university press, 2011.

[2] D. B. Barker and M. E. Fourney, "Measuring fluid velocities with speckle patterns," *Opt. Lett.*, vol. 1, pp. 135–137, Oct 1977.

[3] S. P. Dudderar, T., "Laser speckle photography in a fluid medium," *International journal of environmental research and public health*, vol. 15, no. 1, p. 16, 1977.

[4] R. Grousson and S. Mallick, "Study of flow pattern in a fluid by scattered laser light," *Appl. Opt.*, vol. 16, pp. 2334–2336, Sep 1977.

[5] R. Meynart, "Flow Velocity Measurement By A Speckle Method," in *2nd European Congress on Optics Applied to Metrology* (M. H. Grosmann and P. Meyrueis, eds.), vol. 0210, pp. 25 – 28, International Society for Optics and Photonics, SPIE, 1980.

[6] M. Roland, "Equal velocity fringes in a rayleigh-benard flow by a speckle method," *Appl. Opt.*, vol. 19, pp. 1385–1386, May 1980.

[7] Meynart, R., "Convective flow field measurement by speckle velocimetry," *Rev. Phys. Appl. (Paris)*, vol. 17, no. 5, pp. 301–305, 1982.

[8] R. Meynart, "Digital image processing for speckle flow velocimetry," *Review of Scientific Instruments*, vol. 53, no. 1, pp. 110–111, 1982.

[9] M. Roland, "Instantaneous velocity field measurements in unsteady gas flow by speckle velocimetry," *Appl. Opt.*, vol. 22, pp. 535–540, Feb 1983.

[10] R. Meynart, "Speckle velocimetry study of vortex pairing in a low-re unexcited jet," *The Physics of Fluids*, vol. 26, no. 8, pp. 2074–2079, 1983.

[11] M. Raffel, C. E. Willert, J. Kompenhans, *et al.*, *Particle image velocimetry: a practical guide*, vol. 2. Springer, 1998.

[12] J. Kompenhans, M. Raffel, L. Dieterle, T. Dewhirst, H. Vollmers, K. Ehrenfried, C. Willert, K. Pengel, C. Kähler, A. Schröder, *et al.*, "Particle image velocimetry in aerodynamics: Technology and applications in wind tunnels," *Journal of Visualization*, vol. 2, no. 3-4, pp. 229–244, 2000.

[13] Z. J. Taylor, R. Gurka, G. A. Kopp, and A. Liberzon, "Long-duration time-resolved piv to study unsteady aerodynamics," *IEEE Transactions on Instrumentation and Measurement*, vol. 59, no. 12, pp. 3262–3269, 2010.

[14] M. S. Genç, İ. Karasu, and H. H. Açıkel, "An experimental study on aerodynamics of naca2415 aerofoil at low re numbers," *Experimental Thermal and Fluid Science*, vol. 39, pp. 252–264, 2012.

[15] D. Park, H. Shim, and Y. Lee, "Piv measurement of separation bubble on an airfoil at low reynolds numbers," *Journal of Aerospace Engineering*, vol. 33, no. 1, p. 04019105, 2020.

[16] C. P. Ford and H. Babinsky, "Lift and the leading-edge vortex," *Journal of Fluid Mechanics*, vol. 720, pp. 280–313, 2013.

[17] S. Sunada, A. Sakaguchi, and K. Kawachi, "Airfoil section characteristics at a low reynolds number," 1997.

[18] A. Melling, "Tracer particles and seeding for particle image velocimetry," *Measurement science and technology*, vol. 8, no. 12, p. 1406, 1997.

[19] S. M. Pizer, E. P. Amburn, J. D. Austin, R. Cromartie, A. Geselowitz, T. Greer, B. ter Haar Romeny, J. B. Zimmerman, and K. Zuiderveld, "Adaptive histogram equalization and its variations," *Computer vision, graphics, and image processing*, vol. 39, no. 3, pp. 355–368, 1987.

[20] U. Shavit, R. J. Lowe, and J. V. Steinbuck, "Intensity capping: a simple method to improve cross-correlation piv results," *Experiments in Fluids*, vol. 42, pp. 225–240, 2007.

[21] W. Thielicke and R. Sonntag, "Particle image velocimetry for matlab: Accuracy and enhanced algorithms in pivlab," *Journal of Open Research Software*, vol. 9, no. 1, 2021.

[22] W. Thielicke and E. Stamhuis, "Pivlab–towards user-friendly, affordable and accurate digital particle image velocimetry in matlab," *Journal of open research software*, vol. 2, no. 1, 2014.

[23] W. Thielicke, *The Flapping Flight of Birds–Analysis and Application Ph. D*. PhD thesis, thesis. Rijksuniversiteit Groningen, 2014.

[24] J. Anderson, *EBOOK: Fundamentals of Aerodynamics (SI units)*. McGraw hill, 2011.

[25] O. K. G. Tietjens and L. Prandtl, *Applied hydro-and aeromechanics: based on lectures of L. Prandtl*, vol. 2. Courier Corporation, 1957.

[26] D. Pullin and A. Perry, "Some flow visualization experiments on the starting vortex," *Journal of Fluid Mechanics*, vol. 97, no. 2, pp. 239–255, 1980.

[27] R. Evans and M. Bloor, "The starting mechanism of wave-induced flow through a sharp-edged orifice," *Journal of Fluid Mechanics*, vol. 82, no. 1, pp. 115–128, 1977.

[28] Q. Sun and I. D. Boyd, "Flat-plate aerodynamics at very low reynolds number," *Journal of Fluid Mechanics*, vol. 502, pp. 199–206, 2004.

[29] J. Winslow, H. Otsuka, B. Govindarajan, and I. Chopra, "Basic understanding of airfoil characteristics at low reynolds numbers (10 4–10 5)," *Journal of Aircraft*, vol. 55, no. 3, pp. 1050–1061, 2018.

[30] R. J. Stevens and H. Babinsky, "Low reynolds number experimental studies on flat plates," in *52nd Aerospace Sciences Meeting*, p. 0743, 2014.

[31] I. H. Abbott and A. E. Von Doenhoff, *Theory of wing sections: including a summary of airfoil data*. Courier Corporation, 2012.

[32] S. Prothin, C. Fernandez Escudero, N. Doué, and T. Jardin, "Aerodynamics of mav rotors in ground and corner effect," *International Journal of Micro Air Vehicles*, vol. 11, p. 1756829319861596, 2019.

[33] ANSYS, Inc., *Ansys® Academic Research Fluent, Release 12.0, Help System, Computational Fluid Dynamics Theory Guide*, 2009.

[34] M. Darwish and F. Moukalled, *The finite volume method in computational fluid dynamics: an advanced introduction with OpenFOAM® and Matlab®*. Springer, 2016.

[35] G. West and C. Apelt, "The effects of tunnel blockage and aspect ratio on the mean flow past a circular cylinder with reynolds numbers between 104 and 105," *Journal of Fluid mechanics*, vol. 114, pp. 361–377, 1982.

[36] M. Kreizer, D. Ratner, and A. Liberzon, "Real-time image processing for particle tracking velocimetry," *Experiments in fluids*, vol. 48, pp. 105–110, 2010.

[37] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.

APPENDIX

A Drafting and Fabrication

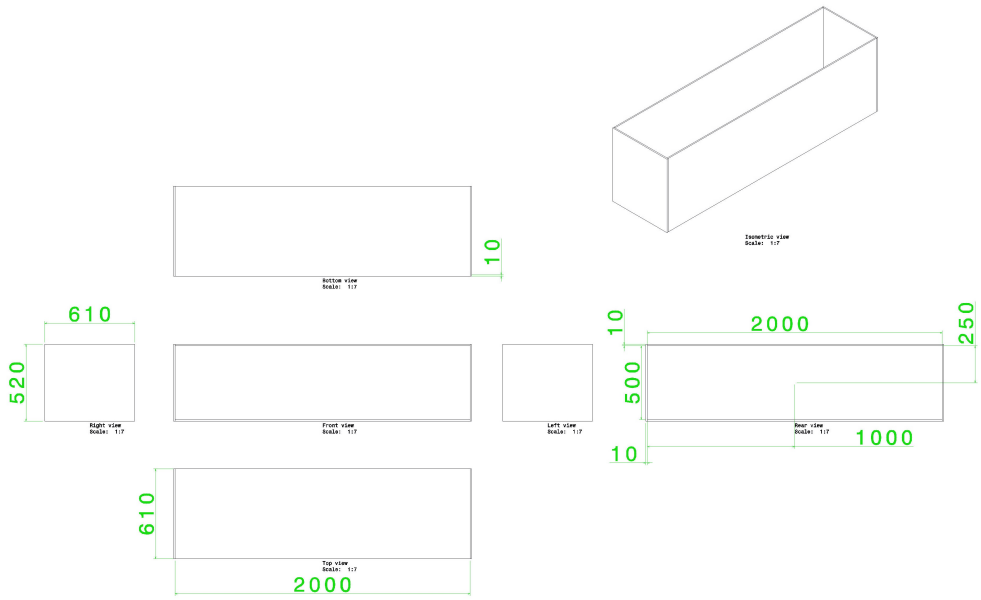


Figure A.1: Tank Drafting

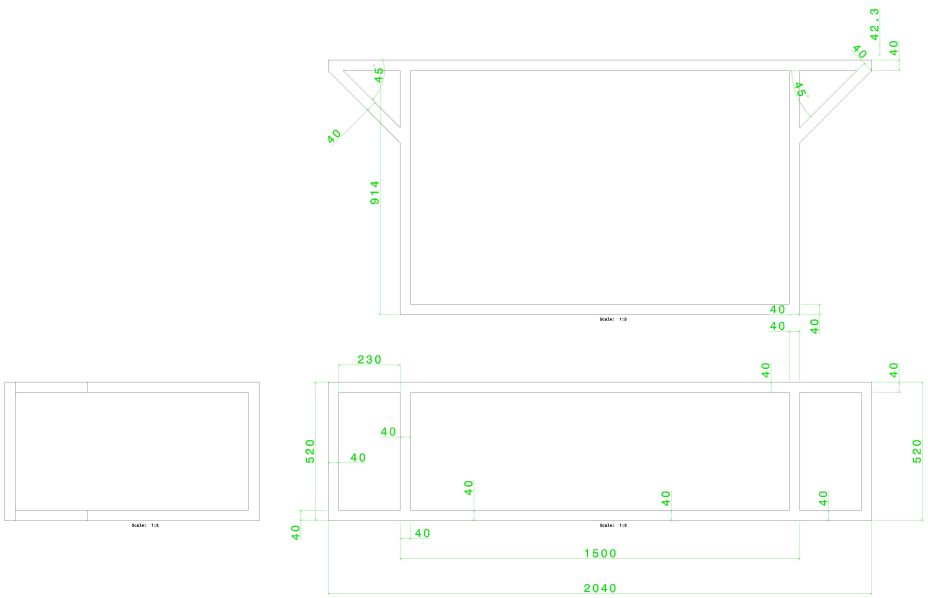
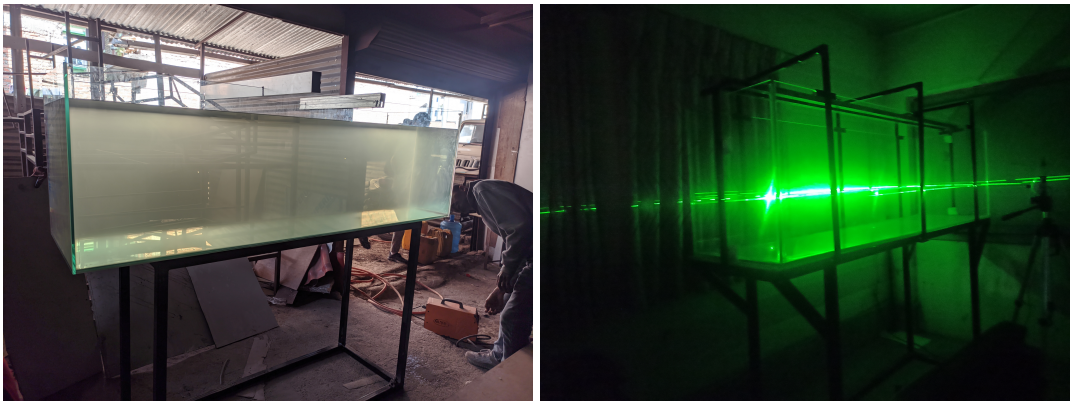


Figure A.2: Frame Drafting

(a) (b)

Figure A.3: Pictures of tank fabrication and testing



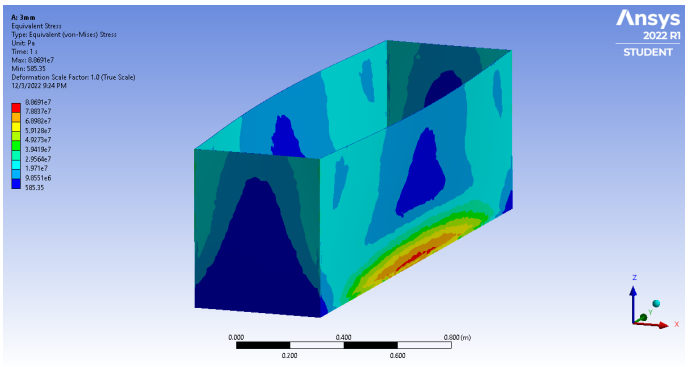
(a) (b)

Figure A.4: Pictures of tank fabrication and testing

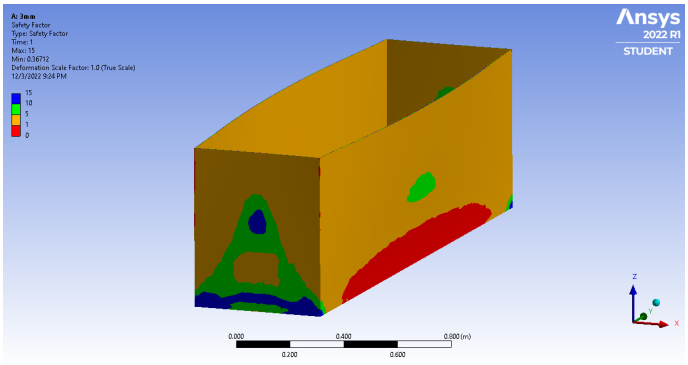
B Analysis

B.1 Tank With no support

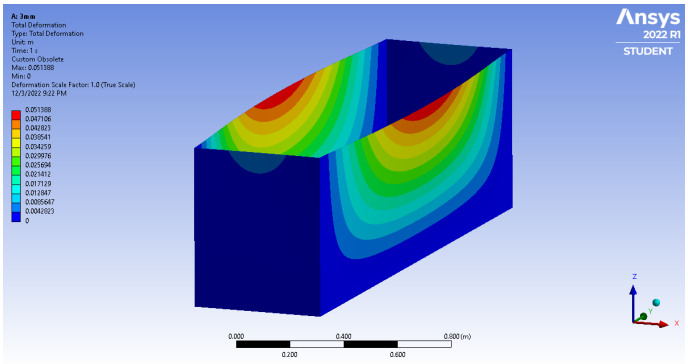
B.1.1 Glass 3mm thickness



(a)



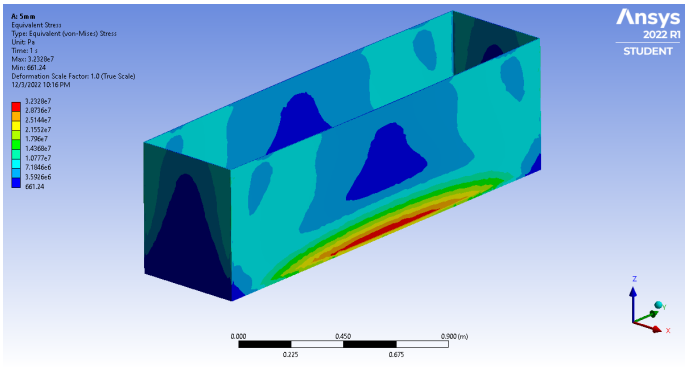
(b)



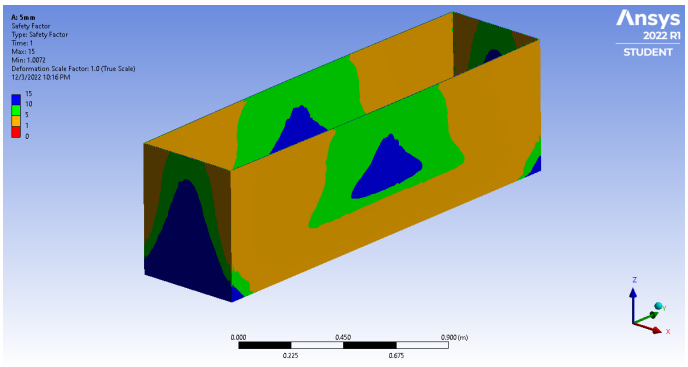
(c)

Figure B.1: Analysis on 3mm thick glass tank without any support, (a) Equivalent Stress, (b) Safety factor, (c) Total Deformation

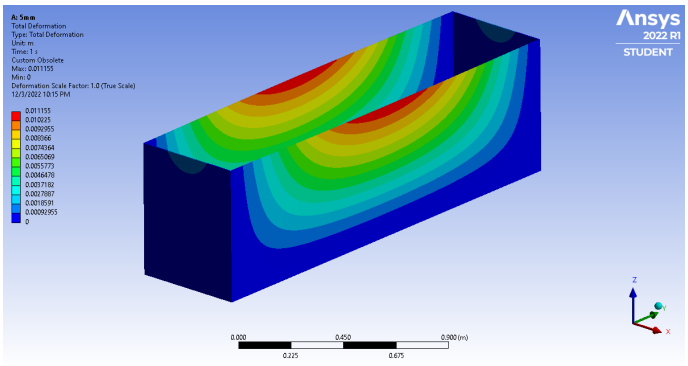
B.1.2 Glass 5mm thickness



(a)



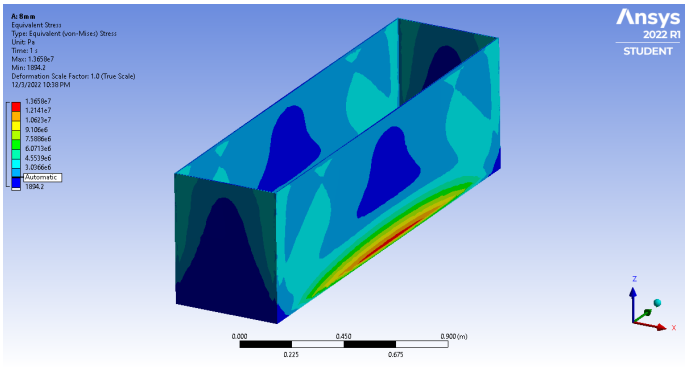
(b)



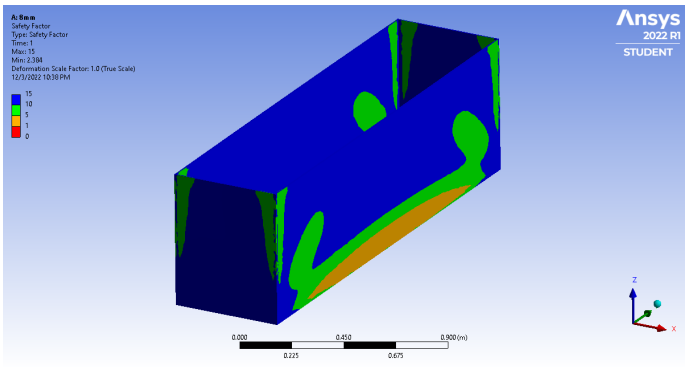
(c)

Figure B.2: Analysis on 5mm thick glass tank without any support, (a) Equivalent Stress, (b) Safety factor, (c) Total Deformation

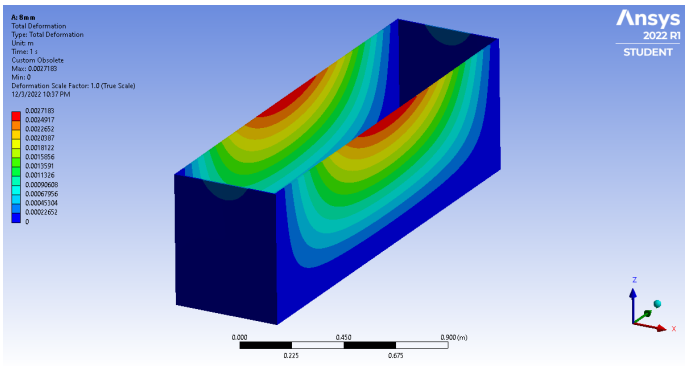
B.1.3 Glass 8mm thickness



(a)



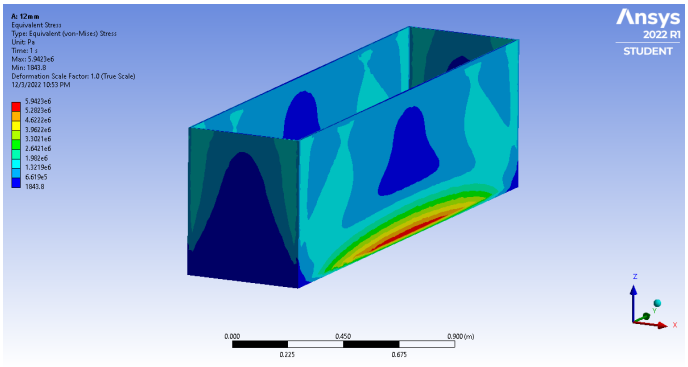
(b)



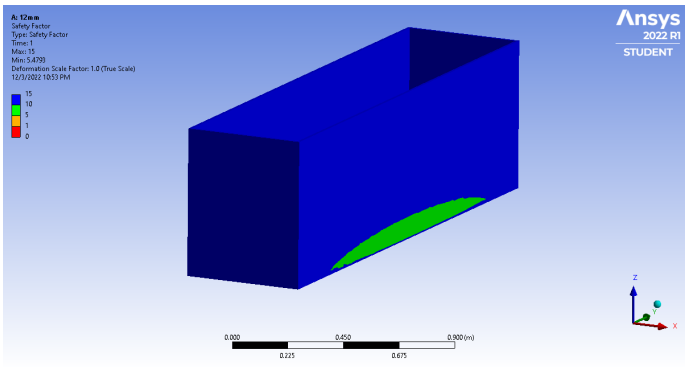
(c)

Figure B.3: Analysis on 8mm thick glass tank without any support, (a) Equivalent Stress, (b) Safety factor, (c) Total Deformation

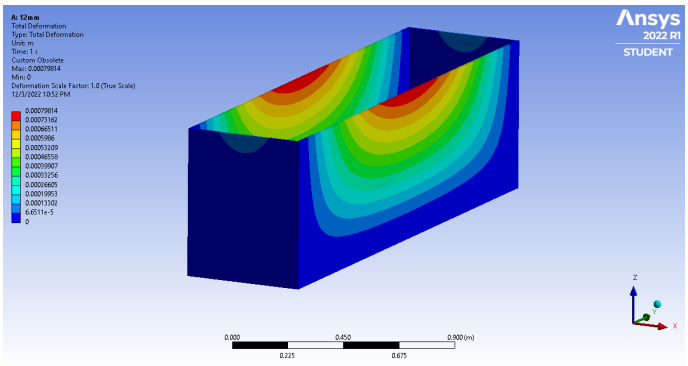
B.1.4 Glass 12mm thickness



(a)



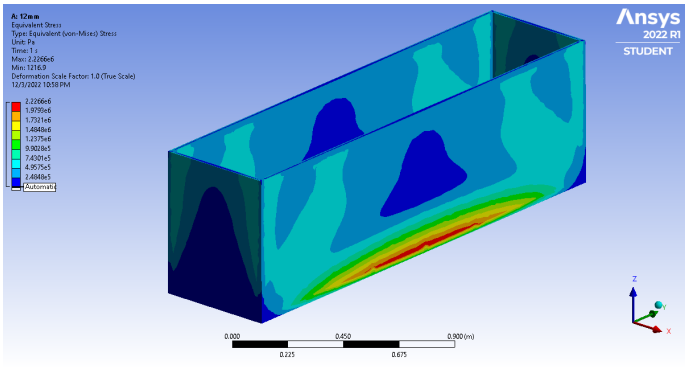
(b)



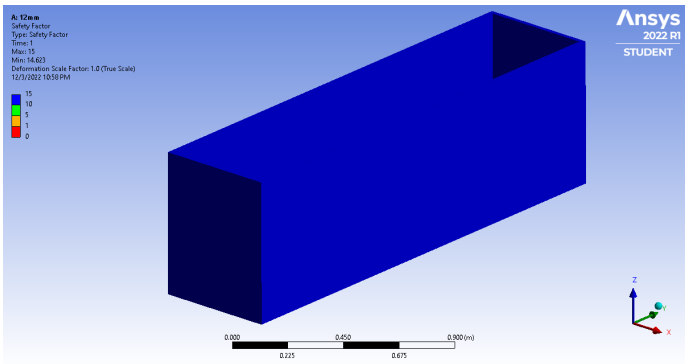
(c)

Figure B.4: Analysis on 12mm thick glass tank without any support, (a) Equivalent Stress, (b) Safety factor, (c) Total Deformation

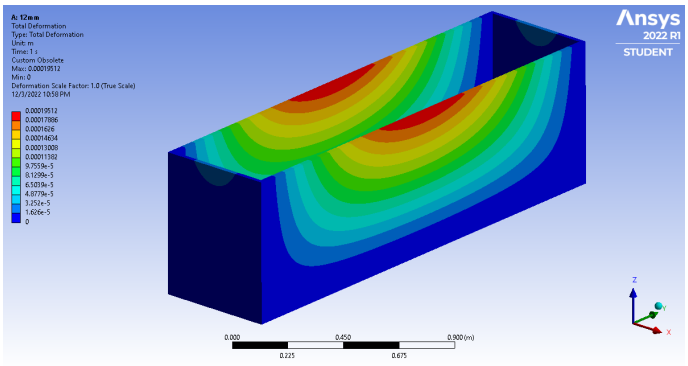
B.1.5 Glass 19mm thickness



(a)



(b)



(c)

Figure B.5: Analysis on 19mm thick glass tank without any support, (a) Equivalent Stress, (b) Safety factor, (c) Total Deformation

C Codes

C.1 Stepper Motor

```
#include <Stepper.h>
#include <BlynkSimpleEsp8266.h>
//#include <Ultrasonic.h>
#include <ESP8266WiFi.h>

#define BLYNK_TEMPLATE_ID "***********"
#define BLYNK_DEVICE_NAME "***************"
#define BLYNK_AUTH_TOKEN "*****************************"


char auth[] = BLYNK_AUTH_TOKEN;
char ssid[] = "****";//Enter your WIFI name
char pass[] = "********";//Enter your WIFI password

// Create a Stepper object
// Replace with the number of steps per revolution
for your
stepper motor
const int stepsPerRevolution = 200;

//pinout mapping
  //const int D1 = 5;
  //const int D2 = 4;
  //const int D5 = 14;
  //const int D6 = 12;
  //const int D7 = 13;
  //const int D8 = 15;

#define IN1 5
#define IN2 4
#define IN3 14
#define IN4 12
```

```
Stepper mystepper(stepsPerRevolution, IN1, IN2, IN3, IN4);

// Create an Ultrasonic object
//Ultrasonic ultrasonic(13, 15);

// Create virtual pins for Blynk
#define CLOCKWISE_BUTTON V1
#define COUNTERCLOCKWISE_BUTTON V2
#define SPEED_INPUT V5
#define DIST_INPUT V6
#define POSITION_OUTPUT V7
#define SPEED_OUTPUT V8

// Create variables to store stepper position and limit
switch status
double pulleyDia = 50;
double motorSpeed;
double motorSpeedrev;
double adjustedSpeed;
int delayVal = 2; // ms
double beltSpeed = 300; // mm/s
double totTheta = 0; // deg
int totSteps = 0;
double beltDist = 1500; // mm
bool clockwise = false;
bool counterclockwise = false;
int i = 0;

// Create variables to store the min and max values for
speed and pulley diameter
const double minSpeed = 0;
const double maxSpeed = 1000;
const double minDist = 500;
const double maxDist = 2000;

//// Callback function for the clockwise button
void clockwiseCallback(const BlynkParam& param) {
    Serial.println("clockwiseCallback called");
```

```
    int totIter = totSteps;
    int totDelay = totIter*delayVal/1000;
    adjustedSpeed = totIter/(totIter/speed-totDelay);
    // Validate the speed value and set it to the
    min or max if
    it is out of range
    if (adjustedSpeed < minSpeed) {
      adjustedSpeed = minSpeed;
    } else if (adjustedSpeed > maxSpeed) {
      adjustedSpeed = maxSpeed;
    }
    mystepper.setSpeed(adjustedSpeed);
    Serial.print("Speed adjusted to ");
    Serial.print(adjustedSpeed);
    Serial.print(" to incorporate delay of ");
    Serial.print(delayVal);
    Serial.println("ms");
    for (int i=1; i<totIter; i++) {
      mystepper.step(1);
      delay(delayVal);
    }
}
//
// Callback function for the counterclockwise button
void counterclockwiseCallback(const BlynkParam& param) {
  // If the limitReached variable is false, move the
  stepper counterclockwise
  if (!limitReached) {
      Serial.println("counterclockwiseCallback called!");
      int totIter = stepsPerRevolution*10;
      int totDelay = totIter*delayVal/1000;
      double adjustedSpeed = float(int(totIter/(totIter/
      speed-totDelay)));
      // Validate the speed value and set it to the
      min or max
      if it is out of range
      if (adjustedSpeed < minSpeed) {
        adjustedSpeed = minSpeed;
```

```
      } else if (adjustedSpeed > maxSpeed) {
        adjustedSpeed = maxSpeed;
      }
      mystepper.setSpeed(adjustedSpeed);
      Serial.print("Speed adjusted to ");
      Serial.print(adjustedSpeed);
      Serial.print(" to incorporate delay of ");
      Serial.print(delayVal);
      Serial.println("ms");
//      while (
      for (int i=1; i<totIter; i++) {
        mystepper.step(-1);
        delay(delayVal);
      }
  }
}

// Callback function for the speed input
void speedCallback(const BlynkParam& param) {
  Serial.println("speedCallback called!");
  // Set the speed variable to the value entered in
  the Blynk app
  beltSpeed = param.asDouble();
  if (beltSpeed > maxSpeed) {
    beltSpeed = maxSpeed;
  } else if (beltSpeed < minSpeed) {
    beltSpeed = minSpeed;
  }
  // Calculations
  motorSpeed = 2*beltSpeed/pulleyDia; // rad/sec
  motorSpeedrev = motorSpeed*30/PI;  // rev/min
  mystepper.setSpeed(motorSpeedrev);
  totTheta = (motorSpeed*(beltDist/beltSpeed))*180/PI;
  totSteps = int(stepsPerRevolution*totTheta/360);
  // Debuggings
  Serial.print("Belt speed set to: ");
  Serial.println(beltSpeed);
  Serial.print("Calculated motor speed is: ");
```

```
  Serial.println(motorSpeedrev);
  Serial.print("Calculated total motor steps is: ");
  Serial.println(totSteps);
  Serial.print("Currently, belt distance is: ");
  Serial.println(beltDist);
  Serial.println("");
}

// Callback function for the distance input
void distCallback(const BlynkParam& param) {
  Serial.println("distCallback called!");
  beltDist = param.asDouble();
  if (beltDist > maxDist) {
    beltDist = maxDist;
  } else if (beltDist < minDist) {
    beltDist = minDist;
  }
  Serial.print("Belt distance set to: ");
  Serial.println(beltDist);
  Serial.println("");
}



BLYNK_WRITE(CLOCKWISE_BUTTON) {
  Blynk.virtualWrite(COUNTERCLOCKWISE_BUTTON, 0);
  if (param.asInt()==1) {
    clockwiseCallback(param);
//    mystepper.step(totSteps);
  }
}
BLYNK_WRITE(COUNTERCLOCKWISE_BUTTON) {
  Blynk.virtualWrite(CLOCKWISE_BUTTON, 0);
  if (param.asInt()==1) {
  counterclockwiseCallback(param);
//  counterclockwise = true;
  }
}
BLYNK_WRITE(SPEED_INPUT) {
```

```
  speedCallback(param);
}
BLYNK_WRITE(DIST_INPUT) {
  distCallback(param);
}


void setup() {
  Serial.begin(115200);
  // Connect to Blynk server
  Serial.println("Just before!");
  Blynk.begin(auth, ssid, pass, "blynk.cloud", 80);
  Serial.println("Blynk server started");

  // Attach virtual buttons to Blynk
  Blynk.virtualWrite(CLOCKWISE_BUTTON, 0);
  Blynk.virtualWrite(COUNTERCLOCKWISE_BUTTON, 0);
  Blynk.virtualWrite(SPEED_INPUT, 300);
  Blynk.virtualWrite(DIST_INPUT, 1500);

  // Set the minimum and maximum values for the belt
  speed and belt distance
  Blynk.setProperty(SPEED_INPUT, "min", minSpeed);
  Blynk.setProperty(SPEED_INPUT, "max", maxSpeed);
  Blynk.setProperty(DIST_INPUT, "min", minDist);
  Blynk.setProperty(DIST_INPUT, "max", maxDist);
}

void loop() {
//  if (clockwise) {
//    mystepper.step(1);
//    i += 1;
//    if (i>=totSteps) {
//      clockwise = false;
//      i = 0;
//    }
//  }
//
```

```
//  if (counterclockwise) {
//    mystepper.step(-1);
//    i += 1;
//    if (i>=totSteps) {
//      counterclockwise = false;
//      i = 0;
//    }
//  }

  // Update the speed outputs to the Blynk app
  Blynk.virtualWrite(SPEED_OUTPUT, motorSpeed);
  Blynk.run();
}
```

C.2 Load Cell

1. Calibration

```
#include <Arduino.h>
#include "HX711.h"

// HX711 circuit wiring
const int LOADCELL_DOUT_PIN = 12;
const int LOADCELL_SCK_PIN = 13;

HX711 scale;

void setup() {
  Serial.begin(115200);
  scale.begin(LOADCELL_DOUT_PIN, LOADCELL_SCK_PIN);
}

void loop() {

  if (scale.is_ready()) {
    scale.set_scale();
    Serial.println("Tare... remove any weights from
```

```
    the scale.");
    delay(5000);
    scale.tare();
    Serial.println("Tare done...");
    Serial.print("Place a known weight on the scale...");
    delay(5000);
    long reading = scale.get_units(10);
    Serial.print("Result: ");
    Serial.println(reading);
  }
  else {
    Serial.println("HX711 not found.");
  }
  delay(1000);
}
```

2. Scale

```
#include <Wire.h>
#include <HX711.h>
#include <BlynkSimpleEsp8266.h>

#define BLYNK_PRINT Serial

#define BLYNK_TEMPLATE_ID "************"
#define BLYNK_DEVICE_NAME "************"
#define BLYNK_AUTH_TOKEN "************************"

#define BLYNK_TOKEN "************"
#define SSID "****"
#define PASSWORD "********"

HX711 scale;

void setup() {
  // Blynk.begin(BLYNK_AUTH_TOKEN, SSID, PASSWORD);
  Serial.begin(115200);
```

```
  // Configure load cell
  scale.begin(12, 13);
  scale.set_scale(-875.38);
  scale.tare();
}

void loop() {
  // Send load cell data to Blynk
  Serial.println(scale.get_units());
  delay(1000);
}
```

C.3 Python Code

C.3.1 Masking

```
import cv2 as cv
import numpy as np
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings("ignore")

def dist2pix(dist, conversion_factor=740/38):
    """ Conversion_factor = chord_length_in_pixels/
    chord_length_in_mm. Find chord_length_in_pixels
    from paraview.
        Draw the chord line over tiff image
        in paraview and find the length."""
    return np.int64(conversion_factor*dist)

def x2y(x, xrange, yrange):
    ix = abs(xrange-x).argmin()
    if x < xrange[ix]:
        x0, x1 = xrange[ix-1], xrange[ix]
        y0, y1 = yrange[ix-1], yrange[ix]
    else:
```

```
            x0, x1 = xrange[ix], xrange[ix+1]
            y0, y1 = yrange[ix], yrange[ix+1]

        y = y0 + (y1-y0)/(x1-x0)*(x-x0)
        if not (np.isnan(y) or np.isinf(y)):
            return int(y)
        else:
            return int(y0)

def airfoil_mask(image_dim, airfoil_coor,
start_LE_coor=(250, 300), AoA=0, flip=False):
    """
    Parameters:
    -----------
    image_dim: dimension of image file which needs
    masking.
    airfoil_coor_path: path to text file containing
    airfoil coordinates.
            order of coordinates: starts from LE,
            first of top face then of bottom face and ends at LE
    LE_coor: tuple corresponding to index(coordinates)
    of LE in the image file.
            (i.e. where the airfoil mask starts
            from in the image)
    TE_coor: tuple corresponding to index(coordinates)
    of TE in the image file.

    Returns
    -------
    A Numpy 2D array containing 1 where there is
    airfoil and 0 where there is not.
    Use np.where(img, True, False) to convert thus
    returned array to boolean mask.
    """
    # image_file = cv.imread(image)
    # img = cv.cvtColor(image, cv.COLOR_BGR2GRAY)
    x, y = airfoil_coor[:, 0], airfoil_coor[:, 1]
```

```
chord = max(abs(x))
thickness = max(y)*2

# x0, y0 = int(min(x)), y[np.abs(x-min(x)).argmin()]
# x1, y1 = max(x), y[np.abs(x-max(x)).argmin()]
x0, y0 = start_LE_coor
x1 = x0 + dist2pix(chord*np.cos(np.deg2rad(AoA)))
# x1, y1 = 1270, 300


ytop = dist2pix(thickness/2)

xrange_afl = x0 + dist2pix(x)
yrange_afl = y0 + dist2pix(y)

airfoil_img = np.zeros(image_dim)

# draw the airfoil contour
for i in range(len(xrange_afl)):
    ix, iy = xrange_afl[i], yrange_afl[i]
    airfoil_img[iy, ix] = 1

# compute coordinates of airfoil lower surface and
fill them with 1's
xfill = np.arange(x0, x1)
for i in xfill:
    ylower = x2y(i, xrange_afl, yrange_afl)
    yupper = y0
    airfoil_img[ylower, i] = 1
    airfoil_img[yupper, i] = 1
    for j in np.arange(yupper, ylower):
        airfoil_img[j, i] = 1

## Evaluate Rotation of Airfoil
rot_x = dist2pix(chord/4)
rot_center = (int(x0+rot_x), y0)
# AoA = -AoA
M = cv.getRotationMatrix2D(rot_center, -AoA, 1.0)
```

```
    # print(airfoil_img.shape)
    airfoil_img = cv.warpAffine(airfoil_img, M, airfoil_img.shape[::-1])

    # compute rotation of LE and TE about c/4 point
    x0_rot, y0_rot = int(rot_center[0]
    -np.cos(np.deg2rad(AoA))*dist2pix(chord/4))
    ,int(rot_center[1]-np.sin(np.deg2rad(AoA))*dist2pix(chord/4))
    x1_rot, y1_rot
    =int(rot_center[0]+np.cos(np.deg2rad(AoA))*dist2pix(3*chord/4))
    ,int(rot_center[1]+np.sin(np.deg2rad(AoA))*dist2pix(3*chord/4))

    # fill upper section of airfoil upto top border of image
    xfill = np.arange(x0_rot, x1_rot)
    for i in xfill:
        ylower = int(y0 + (rot_center[1]-y0_rot)
        /(rot_center[0]-x0_rot)
        *(i-rot_center[0]))
        yupper = 0
        airfoil_img[ylower, i] = 1
        airfoil_img[yupper, i] = 1
        for j in np.arange(yupper, ylower):
            # print(j, i)
            airfoil_img[j, i] = 1

    if flip:
        airfoil_img = cv.flip(airfoil_img, 1)
    roll_pixel = int(2*x0+2*dist2pix(chord))
    airfoil_img = extended_afl_mask(airfoil_img, roll_pixel,
    AoA, init=True)

    # plt.imshow(airfoil_img, cmap=plt.cm.binary)
    # plt.imshow(new_img, cmap=plt.cm.gray)
    # plt.show()

    return airfoil_img

def extended_afl_mask(airfoil_img, roll_pixel, AoA, chord=40
,horizontal_path=True, start_LE_coor=(0, 300), end_LE_coor=(1900,300)
```

```
,init=False):
    pad_width = dist2pix(chord)
    if init:
        airfoil_img = np.pad(airfoil_img, ((0,), (pad_width,))
        ,mode="constant")
    if not horizontal_path:
        path_angle = np.arctan((end_LE_coor[1]
        start_LE_coor[1])/(end_LE_coor[0]-start_LE_coor[0]))
        # returns path angle in radians
        roll_x = np.round(roll_pixel*np.cos(path_angle))
        roll_y = -np.round(roll_pixel*np.sin(path_angle))
        airfoil_imgR = np.roll(airfoil_img, shift=(int(roll_x),
        int(roll_y)), axis=(1, 0))
        print(int(roll_y))
        if int(roll_y)<0:
            print(1)
            airfoil_imgR[int(roll_y):, :] = 0
        else:
            x_coor0 = np.where(airfoil_imgR==1)[1][0]
            y_coor0 = np.where(airfoil_imgR==1)[0][0]
            x_coor1 = x_coor0+dist2pix(np.cos(np.deg2rad(AoA))*chord)
            airfoil_imgR[:int(roll_y), x_coor0:x_coor1] = 1
            # print(((x_coor0, x_coor1), y_coor0))
            # airfoil_imgR[:int(roll_y), :] = 1


        # airfoil_imgRy = np.roll(airfoil_imgRx, shift=(0, ),
        axis=(1, 0))
        # airfoil_imgR = airfoil_imgRy.copy()

    else:
        airfoil_imgR = np.roll(airfoil_img,
        shift=(int(roll_pixel), 0),
        axis=(1, 0))
    if init:
        return airfoil_imgR
    airfoil_imgC = airfoil_imgR[:, pad_width:-pad_width]
    return airfoil_imgC
```

```
def compute_LE(x0, y0, velocity, time):
    d = velocity*time
    dpix = dist2pix(d)
    return (x0+dpix, y0)

# def compute_shift_mask(ref_masked_afl, )
if __name__=="__main__":
    image = cv.imread("7r2_052_b_cut.bmp", 0)
    image_dim = image.shape
    AoA=15.77
    airfoil_coor_path = "airfoil_coor.txt"
    airfoil_coor = np.loadtxt(airfoil_coor_path, delimiter="\t"
    ,dtype=np.float32)
    chord = max(abs(airfoil_coor[:,0]))

    # Find these values from paraview
    # Note: In Numpy array, origin of image is at top left corner.
    Hence, if y_coor value in end_LE_coor
    #       is lower than that in start_LE_coor the image mask
    will be moving downwards. If confused, just
    #       draw a line joining the two points in paraview. Get
    first point from the frame where airfoil just
    #       enters and second point from the frame where airfoil
    just touches end of the picture.

    start_LE_coor = (0, 595) # coordinates of LE of airfoil when
    it just enters the frame from left.
    end_LE_coor = (1916, 300) # coordinates of LE of airfoil when
    it just touches right end of the frame.
    horizontal_path = start_LE_coor[1]==end_LE_coor[1]
    print(horizontal_path)

    a = airfoil_mask(image_dim, airfoil_coor,
    start_LE_coor=start_LE_coor, AoA=AoA, flip=1)
    # plt.imshow(a*255, cmap=plt.cm.gray)
    # plt.show()
```

```
    v_foil = 30 # mm/s
    dt_frame = 0.01
    pix_foil = dist2pix(v_foil)
    pix_shift_per_frame = pix_foil*0.01
    print(pix_shift_per_frame)
    total_iter = int((image_dim[1]+dist2pix(np.cos(np.deg2rad(AoA))*chord))
    /pix_shift_per_frame)


    # for i in range(total_iter):
    for i in range(0, 250, 5):
        cv.imwrite(f"masks_test/{str.zfill(str(i),3)}.bmp"
        ,extended_afl_mask(a, i*pix_shift_per_frame
        ,AoA=AoA,horizontal_path=horizontal_path,
        start_LE_coor=start_LE_coor, end_LE_coor=end_LE_coor)*255)
```

C.3.2 Quiver Plot Generation

```
    import numpy as np
import matplotlib.pyplot as plt
import cv2 as cv
import pandas as pd
import matplotlib.cm as cm
from matplotlib.colors import Normalize

mean_u = []
mean_v = []

for i in range(20):
    piv_flat = pd.read_csv(f"9/PIVlab{i}.txt", skiprows=2
    ,header=0,delimiter='\t')
    mean_u.append(np.mean(piv_flat['u [m/s]']))
    mean_v.append(np.mean(piv_flat['v [m/s]']))

mean_u1 = np.mean(np.array(mean_u))
mean_v1 = np.mean(np.array(mean_v))
```

```
piv_flat1 = pd.read_csv(f"9/PIVlab.txt", skiprows=2, header=0,
delimiter='\t')
x = piv_flat1['x [m]']
y = piv_flat1['y [m]']
u = piv_flat1['u [m/s]'] - mean_u1
v = piv_flat1['v [m/s]'] - mean_v1

colors = np.sqrt(u**2+v**2)
norm = Normalize()
norm.autoscale(colors)
# we need to normalize our colors array to match it colormap domain
# which is [0, 1]
colormap = cm.inferno
# print(plt.rcParams)
plt.quiver(x, y, u, v, np.sqrt(u**2+v**2), scale=900)
plt.colorbar()
plt.show()
```