University of Bristol

Faculty of Engineering
Department of Mechanical Engineering

Fluids 3

Jet Dynamics

Authors:

Jack Sharpe

Kate Mactear

Warot Lilahajiva

Edward McLaughlin

Supervisors:

Dr. C. Snider

Prof. S. Burgess

Dr. M. Tierney

Dr. D. Di Maio

In partial fulfillment of the requirements
for the degree of
Master of Engineering in the subject of
Mechanical Engineering

May 5, 2018

Abstract

This reports looks to compare a turbulent jet modelled using an incompressible flow solver (MOBILE) to the results of a practical experiment. The practical experiment was recorded with three different inlet velocities whilst the simulations were computed with a constant inlet velocity but three different resolutions. Post-processing methods were used in order to calculate the different jet characteristics; inlet velocity, spread rate, virtual origin, attenuation constants, dilution rate, entrainment velocity and Reynolds number. Discussions include the validity of these post-processing methods and the software available to engineers. Further comments are made on the errors that occur with both the practical experiment and simulation. Suggestions are also made on the statistical confidence of the results obtained.

Contents

Abstract i

1 Introduction 1

2 Method 2
- 2.1 Experimental Laboratory . . . 2
- 2.2 Computational Laboratory . . . 3

3 Results and Discussion 4
- 3.1 Bernoulli's Equation . . . 4
- 3.2 Spread Rate and Virtual Origin . . . 6
- 3.3 Statistical Confidence . . . 10
- 3.4 Dilution Rate . . . 11
  - 3.4.1 Calculation of Attenuation Constants . . . 11
  - 3.4.2 Rearranging the logarithmic equation for concentration . . . 13
  - 3.4.3 Assuming $b \neq 0$ . . . 15
  - 3.4.4 Calculating the Dilution Rate . . . 15
  - 3.4.5 Dilution Rate - Experimental . . . 16
  - 3.4.6 Dilution Rate - Simulated . . . 17
- 3.5 Entrainment Velocity . . . 19
- 3.6 Reynolds Number . . . 21

4 Conclusion 23

References i

Appendices i
- Calculation of Spread Rate and Virtual Origin . . . i
- Calculation of attenuation constants . . . iii
- Experimental Dilution Rate . . . viii
- Simulated Dilution Rate . . . xi
- Kolmogorov Dimensional Analysis . . . xv

List of Figures

1 Schematic Diagram of a Jet 1
2 Jet Velocity Schematic 4
3 Practical experiment 1 8
4 Practical experiment 2 8
5 Practical experiment 3 9
6 Simulated Images of Jet 10
7 A graph showing the mean relative light intensity against s, the distance from the ray origin. 12
8 A graph showing the exponential fit for the mean relative light intensities. 13
9 A graph showing the concentration aC_1 versus the distance along the ray, s. 14
10 A graph showing the $\ln(I_n)$ versus distance along the ray, s. 16
11 A contour plot showing the light intensity field for the experimental jet. The black line represents the centreline of the jet whereas the red line represents the centreline of the ray. 17
12 A contour plot showing the light intensity field for a time-averaged simulated jet at a resolution of 128 pixels. 17
13 The dilution rate along the centreline of the simulated jet (black line). The blue line represents an exponential fit to the curve and the dotted line represents a 5th order polynomial fit. 18
14 a.) First Sample Image at t=0 s and b.) Second Sample Image at t=0.25 s 19
15 Variation of Entrainment Velocity With the Distance from The Nozzle 20

1 Introduction

A turbulent jet occurs when a moving fluid enters an undisturbed body of the same fluid via a nozzle or an orifice [1]. Shear velocity is created between the nozzle of the jet and the ambient fluid thereby resulting in chaotic property changes, i.e. turbulence. The most practical defined jets are produced when a liquid waste is discharged into the river or sea through a narrow channel such as an industrial pipe. Understanding of turbulent jets is presently limited to simple analysis. However, modelling a submerged jet according to a jet's behaviour may expand the scope of our understanding and help to further examine the dynamics of a submerged turbulent jet.

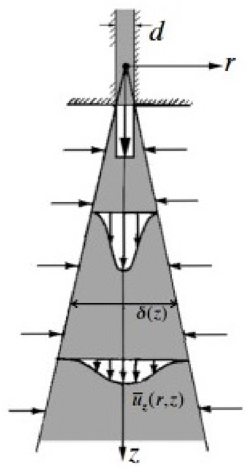


Figure 1: Schematic Diagram of a Jet

This report aims to present various methods used to assess the behaviour and properties of a turbulent jet modelled in the experimental and supercomputer laboratories. The assessment encompasses the application of Bernoulli's equation to produce estimations of inlet velocities, the calculation of spread rate and virtual origin, the calculation of attenuation constants of the Rhodamine B dye and water used in the experimental laboratory, the use of Particle image velocimetry (PIV) to illustrate the variation of entrainment velocity along the edge of the turbulent jet and Reynolds number analysis.

2 Method

2.1 Experimental Laboratory

There are three parts to the experimental laboratory: calibration of light attenuation along the laser ray path [2], laser induced fluorescence and Particle Image Velocimetry (PIV). All parts except PIV undergo the same initial procedures as listed below:

1. Designate a task for each team member: one to operate the camera, one to handle rhodamine contaminated apparatus, one to operate the laser and one to go through the instructions
2. Fill the tank gently with water so that the orifice is submerged. Bubbles formed on the undersurface of the underslung are removed to ensure undisturbed attenuation.
3. Allow the water in the tank to settle while pushing a syringe into the connector pipe in the header tank above. The syringe acts to stop the fluid from draining from the header tank into the glass tank below.
4. Fill the header tank with Rhodamine solution.
5. Allow the eyes to adjust to the darkness when the laboratory lights are turned off. Adjust the camera settings, mainly aperture and shutter speed to ensure maximum exposure before the syringe can be removed
6. Remove the syringe from the connector pipe to begin draining Rhodamine solution into the glass tank

For calibration of laser light attenuation, the procedure is as follows:

- The header tank is allowed to drained complete such that the jet has run dry. A small pump is then used to homogenise the fluid in the glass tank [2].
- Turn on the laser - this should show a fan shape of rays - and focus the camera lens such that the contrast is maximised. Take a clear photo of the attenuation of laser light
- Empty the tank and repeat process 1-6 and follow through the calibration procedure twice to obtain a total of three images.

Laser induced fluorescence part aims to analyse the two types of images of the jet: instantaneous flow and time-mean using fast and slow shutter speeds respectively. The procedure is as follows:

- Turn on the laser as Rhodamine exits the orifice. Take a photograph of the instantaneous flow using a fastest shutter speed with a widest aperture.
- Immediately after the instantaneous flow picture has been take, a time-mean photograph is taken using a slow shutter speed with narrow aperture.
- Empty the tank and repeat process 1-6 and this procedure for different heights of the header tank to analyse the effects the height has on the jet.

Particle image velocimetry (PIV) is aimed at estimating the velocity field of the horizontal entrainment flow. This part of the experimental laboratory does not follow process 1-6 and the procedure is as follows:

- Fill the glass tank with fresh water and inject a particle solution into it. The particles should glisten when laser ray is projected through the tank.
- Homogenise the fluid in the tank using a pump.
- Adjust the camera settings so that the particles appear sufficiently bright for the image(s) to be post-processed
- Take a long exposure photograph, i.e. slow shutter speed to produce an image where the particles displacement is shown through streaks.

2.2 Computational Laboratory

This laboratory exercise aims to run simulations of the submerged turbulent jet using the BlueCrystal supercomputer.

- Create a new working directory in WinSCP. Copy *fluids3jet32.tar.gz* file from *Blackboard* which contains working parameters into the working directory and unzip the file.
- Replace *.bashrc* file with an updated version on *Blackboard* and run *MOBILE.x* file outside the queue to find potential error.
- Change the settings in *input setup.dat*. The only settings concerned are *tnx* which is the resolution of the simulated images and *bcbaseflow* which is the inlet velocity.
- Run simulations for the three inlet velocities for the three corresponding heights with *tnx:=assign(128)*.
- Calculate the average inlet velocity from the three experimental velocities and run three simulations with *tnx* 64, 128 and 256.

3 Results and Discussion

3.1 Bernoulli's Equation

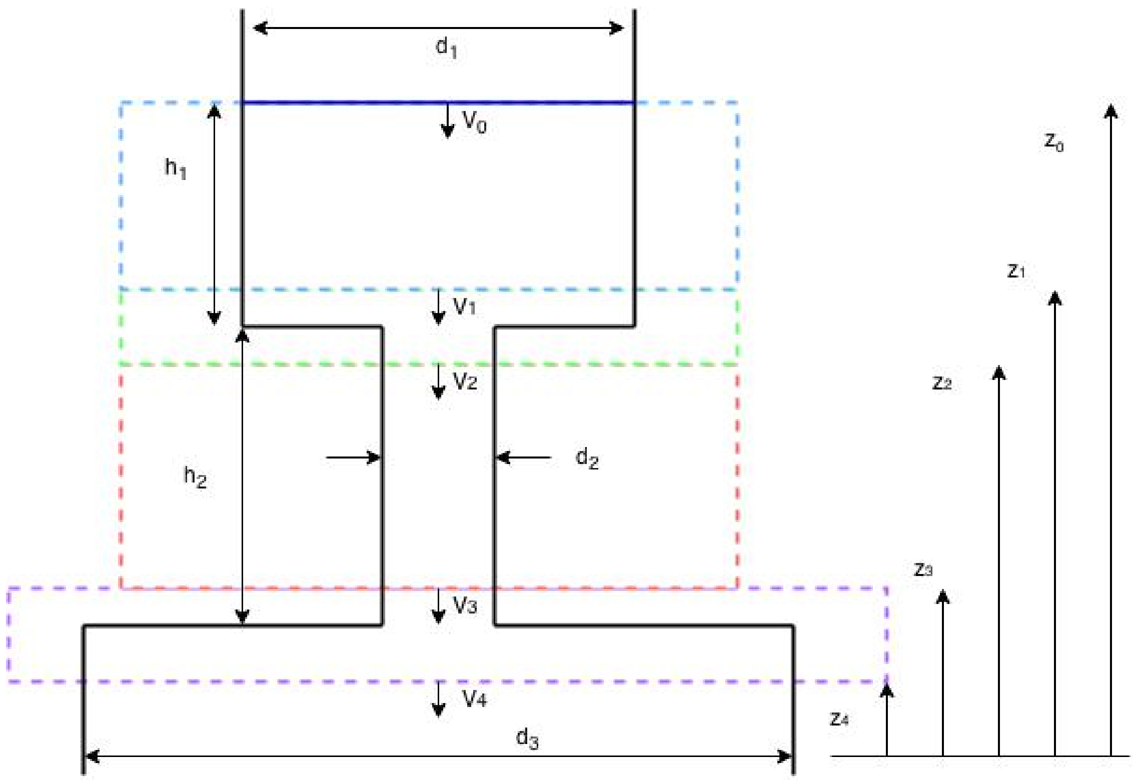


Figure 2: Jet Velocity Schematic

The inlet velocities of the jet at different heights can be estimated using Bernoulli's equation, shown below. Four control volumes are defined in the experiment schematic shown in Figure 2. Each control volume will use Bernoulli's equation (or a slight variation on it) to compute the outlet velocity of that control volume given its inlet velocity and various assumptions. All assumptions will be stated upon use.

$$P_0 + \frac{\rho V_0^2}{2} + \rho g z_0 = P_1 + \frac{\rho V_1^2}{2} + \rho g z_1 = constant \tag{1}$$

For the first control volume the following assumptions are made:

- $P_0 = P_1$ - As static pressure is equal on both sides
- $V_0 = 0$ - As the ratio of diameters is sufficiently large.

This gives the outlet velocity from the first control volume, V_1 as:

$$V_1 = \sqrt{2g\Delta z} \tag{2}$$

where $\Delta z = h_1$; the height of the header tank. In the second control volume the aim is to find the entrance velocity inside the pipe by considering the pressure loss due to the abrupt contraction in the cross-sectional area. This pressure loss is given in [3] as:

$$P_{loss} = \frac{\rho V_2^2 k^2}{2} \tag{3}$$

where $k = \frac{1}{0.63+0.37(\frac{A_2}{A_1})^3} - 1$ for as given in [4]. A_1 and A_2 are the areas before and after the contraction respectively. This changes the Bernoulli equation (Equation 1) as follows:

$$P_1 + \frac{\rho V_1^2}{2} + \rho g z_1 = P_2 + \frac{\rho V_2^2}{2} + \rho g z_2 + \frac{\rho V_2^2}{2}\left(\frac{1}{0.63 + 0.37(\frac{A_2}{A_1})^3} - 1\right)^2 = constant \tag{4}$$

The following assumptions are made for the second control volume:

- $P_1 = P_2$ - As static pressure is equal on both sides
- $\Delta z \approx 0$

Applying these assumptions gives:

$$V_2 = \frac{V_1}{\sqrt{1 + \left(\frac{1}{0.63+0.37(\frac{A_2}{A_1})^3} - 1\right)^2}} \approx 0.938 V_1 \tag{5}$$

for the experimental set-up used. The third control volume takes into account the head loss due to friction in a pipe. This head loss is given by the Darcy-Weisbach equation:

$$h_f = \frac{f h_2 V^2 \rho}{2 d_2} \tag{6}$$

where f, is the friction factor, V is the average flow velocity given by $V = \frac{V_1+V_2}{2}$, h_2 the length of the pipe, ρ is the fluid density and d_2 is the diameter of the pipe. This gives an adjusted Bernoulli equation of:

$$P_2 + \frac{\rho V_2^2}{2} + \rho g z_2 = P_3 + \frac{\rho V_3^2}{2} + \rho g z_3 + \frac{f h_2 V \rho}{2 d_2} = constant \tag{7}$$

The following assumptions are made for the third control volume:

- $P_2 = P_3$ - As static pressure is equal on both sides
- The friction factor, f, for a laminar flow may be assumed to be $f = \frac{64}{Re}$ where Re is Reynolds Number given by $Re = \frac{V d_2}{\upsilon}$ where υ represents the kinematic viscosity. Assuming laminar flow is valid as $\upsilon = 8.9 \times 10^{-4}$ Pa s and $d_2 = 0.0025$ m for water. Thus V would need to be greater than 750 m s^{-1} to have an $Re > 2100$ i.e. turbulent flow.

- $\Delta z = h_2$

This gives a final quadratic of:

$$V_3^2 + \left(\frac{32v}{d_2}\right)V_3 - \left(V_2^2 - \left(\frac{32\epsilon}{d_2}\right)V_2 + 2gh_2\right) = 0 \tag{8}$$

The fourth control volume is concerned with the pressure drop at a sudden expansion of a pipe. This changes the Bernoulli equation to:

$$P_3 + \frac{\rho V_3^2}{2} + \rho g z_3 = P_4 + \frac{\rho V_4^2}{2} + \rho g z_4 + \frac{\rho V3}{2}\left(1 - \left(\frac{A_3}{A_4}\right)^2\right) = constant \tag{9}$$

Making the following assumptions it can be shown that the outlet velocity of the jet into the tank is given by equation 10, noting that $V_{jet} \approx V_{center} \approx V_{edge}$:

- $P_3 = P_4$ - As static pressure is equal on both sides
- $\Delta z \approx 0$
- Pressure loss at a sudden expansion is given by $\frac{\rho V3}{2}\left(1 - \left(\frac{A_3}{A_4}\right)^2\right)$, [3].

$$V_4 = \sqrt{1 - \left(1 - \frac{A_3}{A_4}\right)^2} V_3 \tag{10}$$

After following this process for the three different heights, the velocities in Table 1 were calculated.

Table 1: A table showing the calculated velocities of the jet with different header height tanks using Bernoulli's equation.

Height of header tank, h_1 (mm)	Bernoulli's Velocity(ms^{-1})
210	0.42
280	0.83
380	0.98

3.2 Spread Rate and Virtual Origin

In order to calculate the spread rate and virtual origin, results from the practical experiment and computer simulations were analysed. The analysis was carried out in MATLAB using the code presented in the Appendix. For the experimental images, the code was used to analyse an already time-averaged image taken over 0.4 seconds. For the simulation images, the code was used to generate a time-averaged image using 8 images with 0.5 seconds between each one and then analysed the image in the same way it did the experimental images.

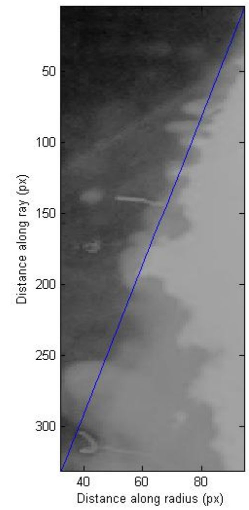
The spread rate is defined as the gradient of the slope of the jet in the direction of the flow. For the practical experiment, time averaged images for each of the three different heights were used: 380mm, 280mm and 210mm. The actual images were constrained to one side of the jet and focuses on the flow directly from the nozzle. It was important that the images being analysed were in steady state in order to get a clear image of the flow. For the computer simulation, an average was taken for the two different resolutions used: 64x128 and 128x256. The time-averaged images were then used to generate a line of best fit of the jet which was then used to calculate both the spread rate and virtual origin. The line of best fit was calculated using least squares and linear regression. The fact that the line of best fit is linear, indicates that the jet width increases linearly with distance from the jet orifice. At large distances, it appears that the influence of the nozzle becomes less and less important.

The virtual origin of the jet, defined as the point where r(z) crosses the z-axis, is the theoretical position from where the jet begins. The virtual origin was found using the best fit line of the jet and calculating where it crossed the z-axis. It is generally found that the virtual origin is located behind the actual nozzle.

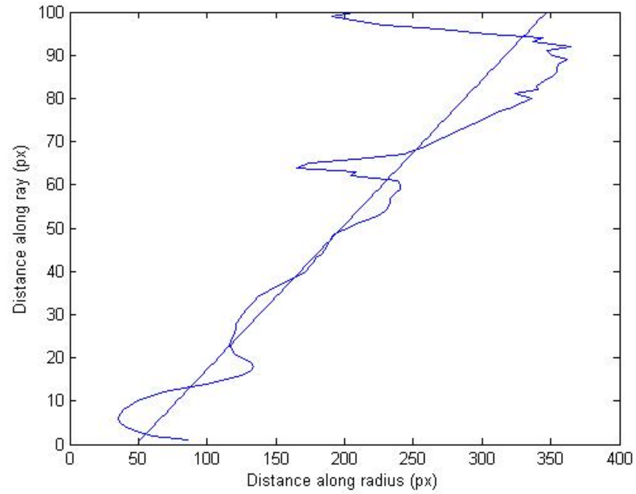
The conversion of pixels to millimeters (mm) was calculated by comparing the width of the nozzle in mm to the width of the pixel length. This varied for each image depending on the resolution of the image. The virtual origin is shown in Table 2 in both pixels and millimeters.

Table 2: Table to show the results of Spread Rate and Virtual Origin

	Inlet Velocity (ms^{-1})	Spread Angle (degrees)	Spread rate (r(z))	Virtual origin (px)	Virtual origin (mm)
Experiment 1 (380mm)	0.98	10.72	0.33	28.47	9.05
Experiment 2 (280mm)	0.83	12.53	0.34	49.88	22.10
Experiment 3 (210mm)	0.42	15.95	0.58	48.00	21.26
Simulation 1 (64x128)	0.74	16.36	0.16	36.53	15.22
Simulation 2 (128x256)	0.74	7.80	0.20	147.7	30.77

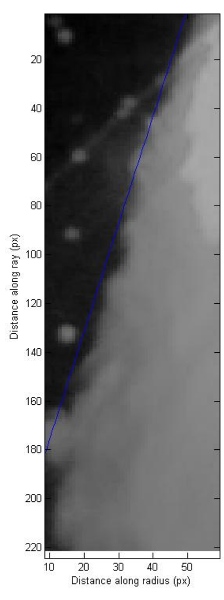


(a) Calculation of the Virtual origin

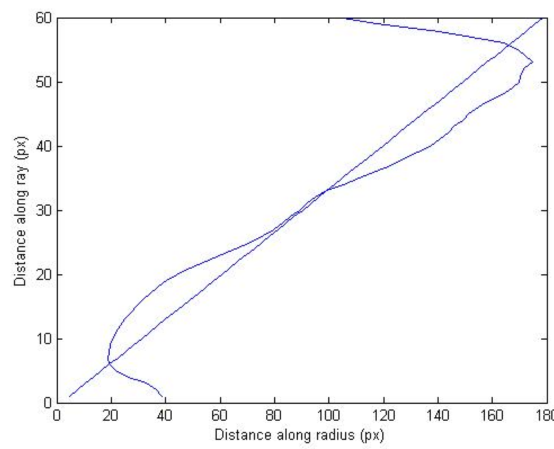


(b) Calculation of the Spread rate

Figure 3: Practical experiment 1

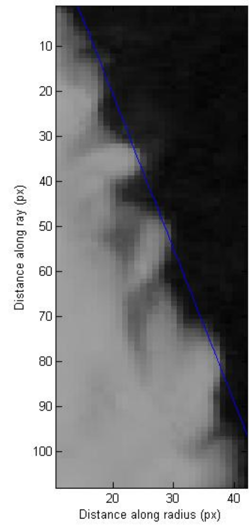


(a) Calculation of the Virtual origin

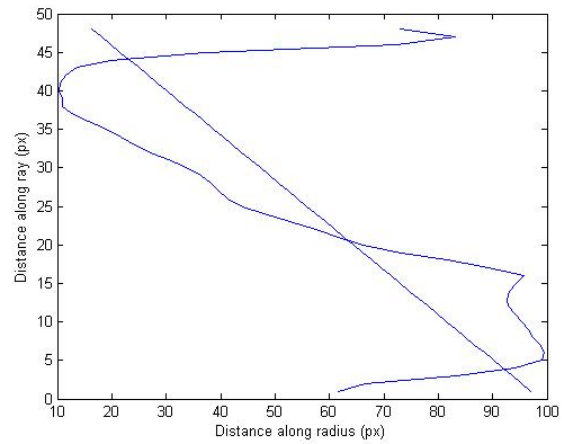


(b) Calculation of the Spread rate

Figure 4: Practical experiment 2



(a) Calculation of the Virtual origin



(b) Calculation of the Spread rate

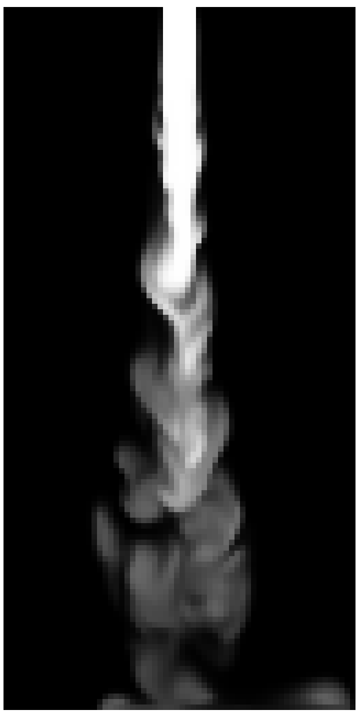
Figure 5: Practical experiment 3

Figures 3 - 5 show the processing of the spread rate and virtual origin for each practical experiment.

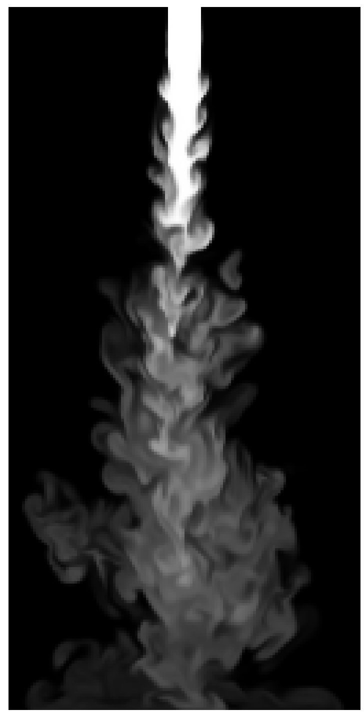
According to a report written on *Environmental Fluid Mechanics* by *Benoit Cushman-Roisin*, the spread angle should always be the same, regardless of the fluid and other variables such as the diameter of the orifice and the inlet velocity. The report states the universal angle is equal to 11.8° which is equivalent to a spread rate of 0.209 [1]. The average value obtained during the analysis of the practical experiment for the spread angle was 13.07° which is 11% more than the universal value. The reason the values obtained are higher than the expected value could be due to many different errors. The laser light that was being used to illuminate the tank of water was not as bright as it should have been. Consequently the images that were taken were not clear enough and therefore the analysis of the jet was more difficult. There was also a problem with laser alignment with the jet orifice. This means that if the jet wasn't illuminated properly and turbulence of the jet starts before the laser reaches the jet, information might be lost. There was also the issue with air bubbles being present underneath the jet orifice. Although the area was cleared of the larger air bubbles, it was impossible to ensure the smallest ones were no longer present. This in turn may have affected the path of the jet.

Illumination of the nozzle is not necessary for the practical experiment, as the measurements of spread rate and virtual origin are taken from the section of the jet that is in steady state. At the orifice, the jet is just beginning to form and is in what is called the transient phase which occurs before steady state. For an accurate measurement, the jet needs to be fully formed hence illuminating the nozzle is not necessary as no useful information can be gained from the beginning of the flow.

The average value obtained during the analysis of the simulated images for the spread angle was 12.08° which is only 2.37% higher than the universal value. However the difference between the two spread angles is 8.56°. This difference in the spread angle for the simulated images could potentially be due to errors when finding the line of best fit along the jet. The difference could also be due to the truncation error. This is the error than arises when higher order terms in the partial differential equations governing the flow are ignored. Figure 6 shows the light intensity of a simulated jet, produced by BlueCrystal. For the lower resolution image, the fine detail and smaller eddies that are captured in the higher resolution image are not computed which means more errors occur when analysing the image.



(a) Low resolution simulation of jet: 64x128



(b) High resolution simulation of jet: 128x256

Figure 6: Simulated Images of Jet

The lower resolution image shows a much smoother flow which is a poor representation of a real life turbulent jet.

3.3 Statistical Confidence

The Confidence Interval is a statistical estimate that is used to indicate the reliability of an estimate. In this report, it has been used to calculate the reliability of the virtual origin prediction. Taking the confidence level at 99% the following equation can be used to calculate the margin of error:

$$Z_{\frac{n}{2}} \times \left(\frac{\sigma}{\sqrt{n}}\right) \tag{11}$$

where $Z_{\frac{n}{2}}$ is the confidence coefficient, a is the confidence level which has been selected as 0.99, σ is the standard deviation and n is the sample size. A value of ± 17.66 is calculated as the margin of error. To find the upper and lower bounds of the confidence interval the margin of error is added and subtracted from the mean of the virtual origin. The mean of the virtual origin is calculated as 42.12 which means all values for the virtual origin calculated from the experimental images lie within the confidence interval.

3.4 Dilution Rate

It is known that the dye concentration of a jet decreases as the fluid moves further from the nozzle. The change in concentration with respect to s, the distance along the jet, is called the dilution rate. The dilution rate is a useful metric in jet dynamics as it can be used to find an estimate of the entrainment velocity and the mass flux. Although it is extremely useful, it is also impossible to measure directly. Instead, it is derived from the light intensity.

Due to the attenuation of light throughout the dye, the light intensity cannot be represented directly by the concentration. Therefore attenuation constants need to be calculated from calibration experiments.

3.4.1 Calculation of Attenuation Constants

The first task carried out during the experimental laboratory was a calibration exercise. This was done to calibrate the rate of light attenuation along the laser ray path. It is necessary to do this in order to calculate the concentration from light intensity while the jet is running.

The calibration was done by filling the tank of water with a quantity of of Rhodamine B which was then mixed using a pump. This ensured that the dye concentration was consistent throughout the tank. Photographs were then taken making sure that the images were exposed correctly to ensure the direction of the ray was known and to maximise the contrast between the light of the laser and the dark areas where all the laser light had been attenuated.

To ensure that the correct value light intensity was calculated, the mixing and photographing process was repeated twice with additional amounts of dye added, equal to the first.

The three photographs were then processed using a MATLAB script in order to calculate the relative intensity of light at 256 points along a ray. This was done for each concentration with the light intensity from 121 rays accumulated for each. The mean was then taken for each set of data to minimise the effect of photograph noise and random error. These means were then plotted against the distance along the ray which can be seen by the black lines in Figure 7.

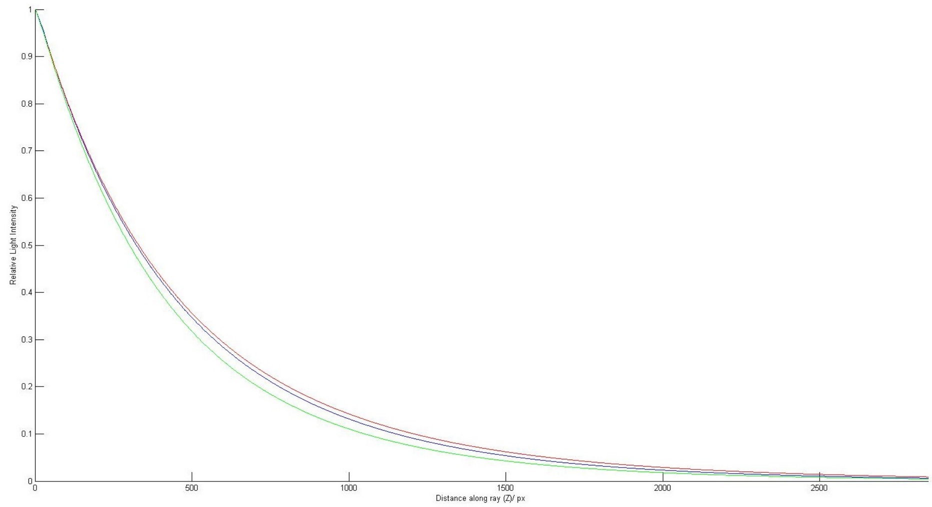


Figure 7: A graph showing the mean relative light intensity against s, the distance from the ray origin.

The Beer-Lambert law, given in Equation 12, can be used to model the attenuation of light through a material, in this case water.

$$\frac{\delta I}{\delta s} = -f(C)I \tag{12}$$

where I is the light, s is the distance along the ray and C is the concentration. As the light attenuation through water follows a linear relationship, $f(C)$ can be modelled as, $f(C) = aC + b$. From this the attenuation of the laser can be modelled using the following equation:

$$I(s) = I_o e^{-(aC+b)s} \tag{13}$$

where $I(s)$ represents the intensity at a distance s from the rays origin, I_0 represents the laser intensity at the origin, C is the dye concentration and a and b are attenuation coefficients for Rhodamine B and water respectively.

In order to calculate the product of the attenuation coefficient a and C, several assumptions had to be made. The first being that the same quantity of Rhodamine B was added to the tank for each image, therefore $C_2 = 2C_1$ and $C_3 = 3C_1$. An assumption was also made that the light absorption by the water was considered negligible, there $b = 0$. Due to the relative light intensity being plotted it can be stated that $I_o = 0$.

By following these assumptions it was possible to use the 'fit' function in MATLAB to fit an exponential curve to the mean relative light intensities in order to find aC_n. The

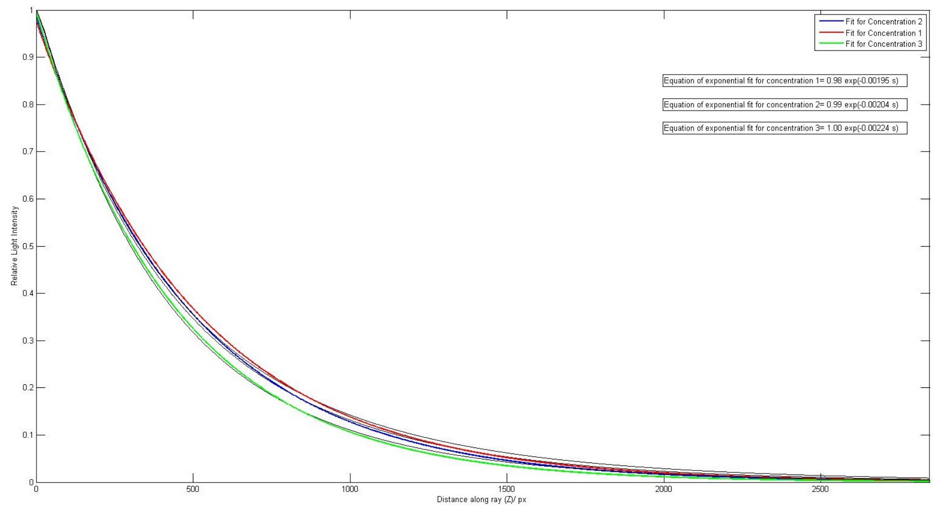


Figure 8: A graph showing the exponential fit for the mean relative light intensities.

'fit' function uses a non-linear least squares regression technique with a 95% confidence interval which gives the plots given in Figure 10.

All values for aC_n and the confidence interval can be found in Table 3.

Table 3: A table showing aC_n and the intensity at the laser origin for different concentrations of Rhodamine B

	aC_n	Original Intensity, I_0	95% Confidence Interval
Concentration 1	-0.001954	0.9765	(-0.001958, -0.001950)
Concentration 2	-0.002044	0.9862	(-0.002047, -0.002040)
Concentration 3	-0.002237	0.9954	(-0.002241, -0.002234)

It can be seen that there is a small amount of uncertainty present in the results for all three concentrations. This uncertainty could be present due to a number of errors from both the practical experiment and computational errors. For example, the experimental errors could have occurred from the dye not mixing fully or reflections being present.

Also given in Table 3 are the values of I_0 for each concentration. It can be seen that the assumption that $I_0 = 1$ holds true as the actual values for I_0 are only 2.35%, 1.38% and 0.46% out respectively.

3.4.2 Rearranging the logarithmic equation for concentration

For understanding jet behaviour, dye concentration is a more useful output than light intensity. In order to extract the dye concentration from the experimental data, we need to track along each light ray and reverse engineer the light attenuation by rearranging the logarithmic equation in terms of concentration [2]. We can do this by studying the relationship between the intensities at two concentrations, see Equation 14 below.

$$\frac{I_1(s)}{I_2(s)} = \frac{I_0 e^{-(aC_1+b)s}}{I_0 e^{-(aC_2+b)s}} \tag{14}$$

By using the assumption that $b = 0$, this can be rearranged to give:

$$ln\Big(\frac{I_1(s)}{I_2(s)}\Big) = as(C_2 - C_1) \tag{15}$$

As we know that $C_2 = 2C_1$, it can be stated that:

$$aC_1 = \frac{1}{s} ln\Big(\frac{I_1(s)}{I_2(s)}\Big) \tag{16}$$

A plot of aC_1 against the distance along the ray can then be plotted. As the mixture of Rhodamine B and water is assumed to be thoroughly mixed, the concentration is assumed to be constant. It can be seen from Figure 10 that this is not the case and there is in fact some variation in the concentration along the distance of the ray.

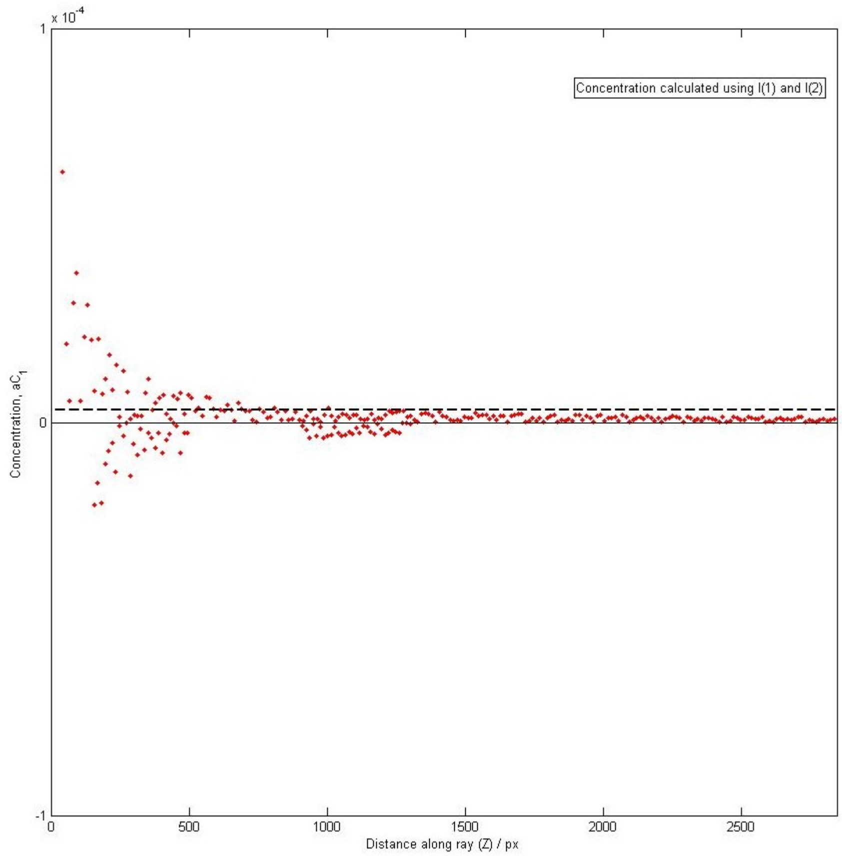


Figure 9: A graph showing the concentration aC_1 versus the distance along the ray, s.

By plotting a line of best fit through the points it becomes more clear that the points lie on a lie with a gradient very close to zero, therefore supporting the assumption that the mixture was fully mixed. From Figure 10 it can be approximated using the line of best fit that the value of aC_1 is 2.5753×10^{-6}.

3.4.3 Assuming $b \neq 0$

Throughout the previous two sections, the assumption that $b = 0$ has been made. With the data available it is in fact possible to calculate b, as well as aC_1. In order to do this, Equation 13 must be applied to the first and second concentrations, giving the following equations:

$$I(s) = I_o e^{-(aC_1+b)s} \tag{17}$$

and

$$I(s) = I_o e^{-(aC_2+b)s} \tag{18}$$

By taking the natural logarithm to each equation and assuming that $I_0 = 1$, the natural logarithm of relative light intensity can be found, see Equations 19 and 20.

$$log(I_1) = -(aC_1 + b)s \tag{19}$$

$$log(I_2) = -(aC_2 + b)s \tag{20}$$

When plotted on axes labelled $ln(I_n)$ against s, the following plots are produced. By re-arranging these equations for b, the following relationship can be given:

$$\frac{-log(I_1)}{s} - aC_1 = \frac{-log(I_2)}{s} - aC_1 = b \tag{21}$$

By letting $aC_2 = 2aC_1$ and taking values for I_1 and I_2 at a s value close to 110, Equation 21 can be re-arranged to give a value of aC_1 equal to 3.7454×10^{-6}. By substituting this back into Equation 21, the value of b is found to be 0.0022. These values for aC_1 and b very much replicate the results obtained in the previous section. Although this is the case they do not replicate the results obtained from using the 'fit' function within MATLAB.

It should be stated that a and C_n do not need to be known independently of each other and in fact it is only necessary to know them as a product of each other. Granting this it is important that the correct units are specified for each variable so that the units of aC_n match the units of b. It is a possibility that a mismatch of units is responsible for the misalignment of the values of aC_1 generated using different methods.

3.4.4 Calculating the Dilution Rate

The dilution rate can be described as the change in concentration with respect to z. It is important to know the dilution rate as from it, an estimation of the mass flux and entrainment velocity can be calculated. It can be said that the concentration of a jet decreases as the fluid gets further away from its origin.

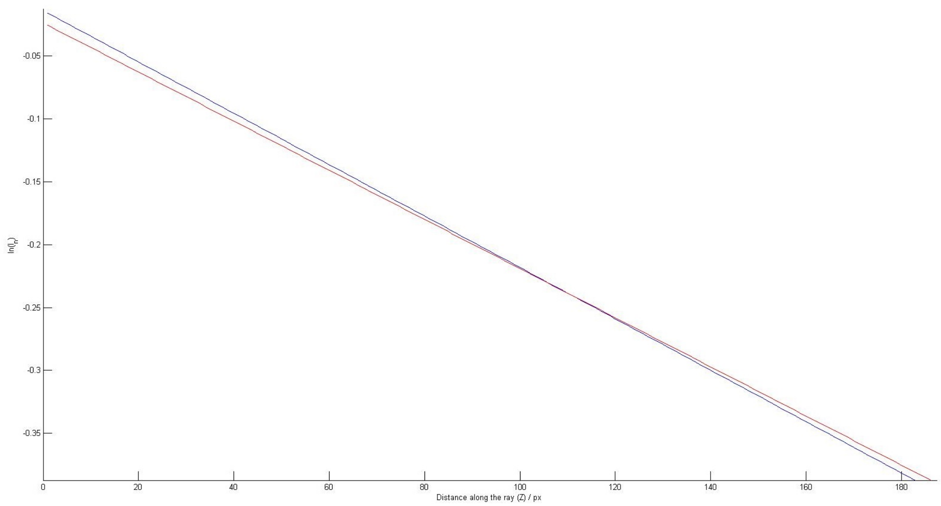


Figure 10: A graph showing the $\ln(I_n)$ versus distance along the ray, s.

By taking a time-averaged image when the height of the header tank was 280mm and using the approximated value of aC_1, it is possible to estimate the light intensity field of the jet. Equation 12 can be used to calculate the light intensity field although only when the concentration throughout the tank is constant. In this scenario when the concentration is not constant, s must be discretised into Δs where the concentration during each step is assumed to be constant. By doing this, Equation 12 can be rearranged to give the concentration as:

$$C(s) = \frac{1}{a\Delta s} ln\Big(\frac{I(s)}{I(s-1)}\Big) \tag{22}$$

3.4.5 Dilution Rate - Experimental

Due to several reasons it is difficult to accurately calculate the concentration and the dilution rate of the experimental data, many of which are generated through the set up of the experiment. As well as the previously mentioned reflections on the tank glass it must also be noted that the origin point of the laser and the jet nozzle are not coincident. This has several affects including that the top of the jet is not illuminated and that many of the laser rays pass through the jet at an angle. These angled rays produce a larger amount of light attenuation on one side of the light intensity field. Figure 11 shows a contour plot of the light intensity field for a time-averaged experimental jet. The black line on the plot represents the centreline of the jet. As predicted previously, there is a change in in light intensity when the angled rays pass through the jet, which can be seen from the edge created in the contour plot. The drop in light intensity can also be viewed from the contour plot. This is represented by the contour lines lying closer to the y-axis below the jet centreline. Although there is a small drop in light intensity it is not as much as expected in this case.

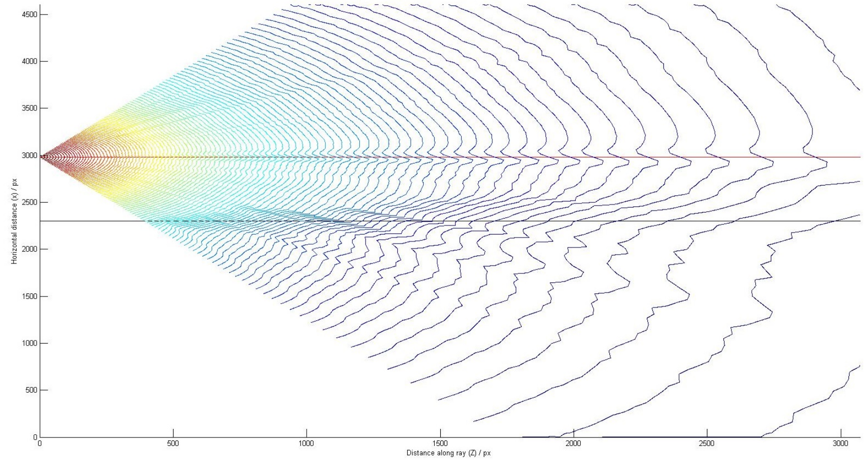


Figure 11: A contour plot showing the light intensity field for the experimental jet. The black line represents the centreline of the jet whereas the red line represents the centreline of the ray.

3.4.6 Dilution Rate - Simulated

Unlike with the experimental data, it is far easier to calculate the dilution rate with the simulated jet data. The reason for this is that the laser in the simulation can be positioned so that the laser origin and the jet nozzle can be placed coincidental. This is visible in Figure 12 which shows there is a light intensity field which is approximately equal on either side of the nozzle/jet.

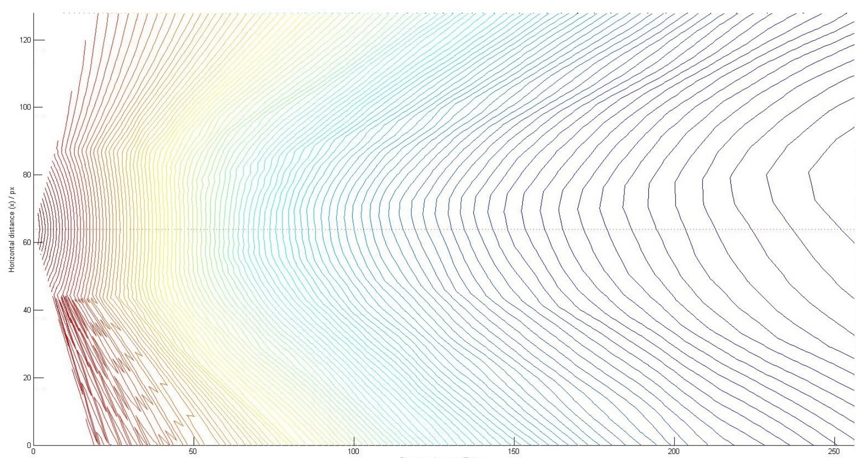


Figure 12: A contour plot showing the light intensity field for a time-averaged simulated jet at a resolution of 128 pixels.

Although this is the case, it can be seen that there is a small area of the light intensity field which has become jagged. This is due to numerical errors in the simulation becoming present in the data.

Similar to what was previously done, the relative light intensity can be plotted against the distance along the ray to give the dilution rate along the centreline of the time-averaged simulated jet, given in Figure 13.

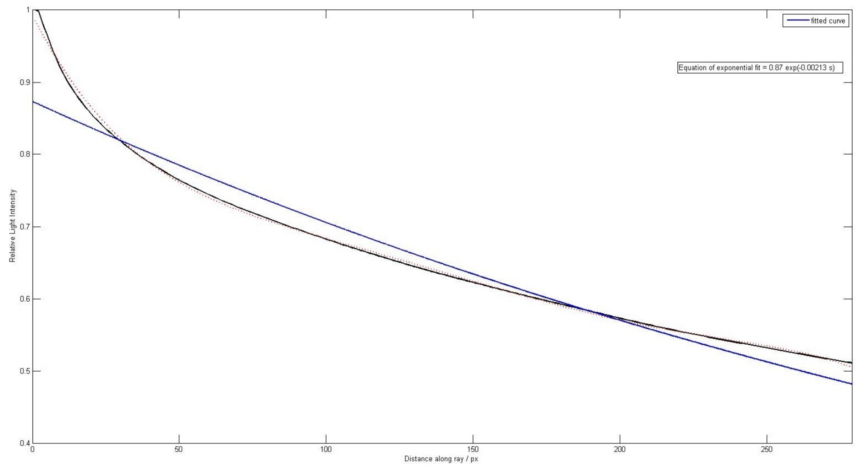


Figure 13: The dilution rate along the centreline of the simulated jet (black line). The blue line represents an exponential fit to the curve and the dotted line represents a 5th order polynomial fit.

In order to gather the dilution rate at any point along the centreline, $C(0, z)$, a curve must be fitted to the true plot. Both an exponential curve using least-squares regression and a 5th order polynomial curve have been fitted; the latter giving a more accurate representation of $C(0, z)$.

It should be noted that time-averaging the concentration field is generally not the same as time-averaging the light intensity field; although the concentration is the derivative of the light intensity field with respect to s. This is due to the way that light attenuates.

In spite of this there is a single scenario when a time-averaged light intensity field can equal a time-averaged concentration field. If the attenuation of light is assumed to be really small then the two would be equal. This would hold true as at every point the light intensity would be the same and since the light emitted through Rhodamine B is proportional to the concentration then the light intensity field would be same as the concentration field. It must be considered that these assumptions only hold true if the inverse square law of radiation is ignored and instead of assuming that $I(s) \propto \frac{1}{s^2}$, $I(s) \propto \frac{1}{s}$ is used instead.

3.5 Entrainment Velocity

As the jet from the nozzle develops, an increasing amount of fluid does not originate from the nozzle. Further downstream, a substantial amount of fluid in the jet is the ambient fluid which has been engulfed by a shear induced turbulent flux [1] and therefore, it is considered to be fluid from the jet nozzle. From conservation of momentum, the momentum flux across the horizontal plane is constant for all heights. The turbulence transfers this momentum onto the ambient fluid, which has no initial momentum, surrounding the turbulent edge. Engulfed ambient fluid is displaced axially thereby creating a gap. As the jet behaviour varies very slowly in the vertical plane, ambient fluid in the horizontal plane replaces the engulfed fluid. The velocity at which ambient fluid replaces the engulfed molecules is known as entrainment velocity, $u_{entrain}$. The entrainment velocity of the analysed jet generated from the experimental data and computational simulations is measured using a technique known as Particle Image Velocimetry (PIV).

PIV is a non-intrusive laser optical measurement technique whereby tracer particles are introduced into the tank. These particles are illuminated when a laser ray is projected through the tank. The velocity at which they entrain, i.e. the velocity at which the particles are moving towards the jet to replace the engulfed particles, can be approximated using:

$$v = \frac{d}{t} \tag{23}$$

where d=distance travelled and t=time taken. For this particular experiment, sample images provided are analysed. This is due images taken in the experimental laboratory showing insufficient contrast and focus. The images were taken 0.25 seconds apart. Since the displacement time is known, the particle displacement can be approximated by measuring the pixels - which will later be translated into mm - each particles move within that time frame. The images considered are shown in Figure 14.

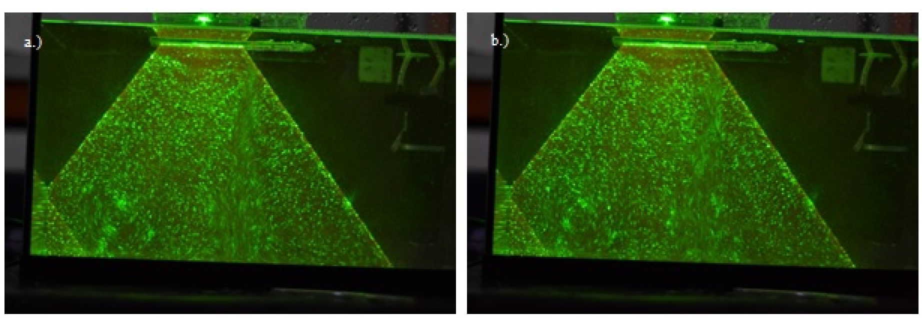


Figure 14: a.) First Sample Image at t=0 s and b.) Second Sample Image at t=0.25 s

The entrainment velocities are measured in the direction of increasing z along the edge of the jet where eddies are yet to be formed, i.e. velocity data which are less affected by eddies along the edge. On average, the entrainment velocity decreases as the distance from the nozzle increases. This is shown in Figure 15.

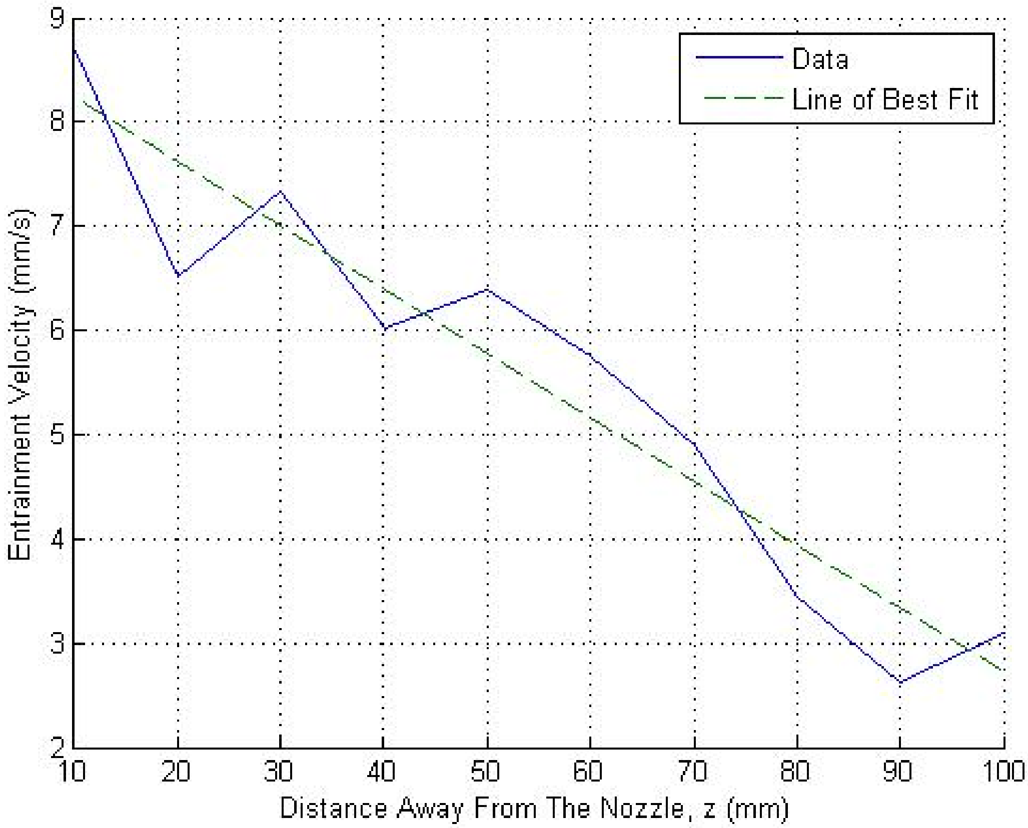


Figure 15: Variation of Entrainment Velocity With the Distance from The Nozzle

It can be seen that the trend line shows a linear decay of entrainment velocity with distance from the nozzle with equation:

$$u_{entrain} = 8.846875 - 0.0612875z \tag{24}$$

It is assumed that momentum flux in the axial direction is constant and mass flux is variable and that in a two-dimensional planar jet analysis, mass flux varies per unit width of the jet, $\delta(z)$. Thus,

$$\frac{\partial \dot{m}}{\partial z} \approx \alpha \frac{\dot{m}}{\delta(z)} \tag{25}$$

where α is the coefficient of entrainment and is an undetermined constant [2]. The width of the jet increases with, and is proportional to, the distance from the nozzle. This means that as the jet develops away from the nozzle, the mass flux must increase as centreline velocity decreases to maintain momentum flux along the jet. The increase in mass flux along the edge of the jet is a result of an increase in entrainment velocity,

$$M = \dot{m}u_z = constant \tag{26}$$

this implies that there is a direct proportionality between entrainment velocity and width of the jet.

By employing three-dimensional axi-symmetric jet analysis, mass flux varies with the increase in circumference of the jet and that the particles entrained are circumferential. The main reason for the decrease in entrainment velocity as the distance from the nozzle increases as shown is that the rate at which the entrainment velocity increases is outweighed by the rate at which the circumference of the jet expands.

In theory, the entrainment velocity is perfectly horizontal. However, Figure 14 illustrates two images where tracer particles are moving in all directions with a whirling motion. This is caused by shear velocity in the flow along the jet which changes the direction of the entrained particles or the particles present in the entrainment domain. Additionally, the assumption that the jet flows into an infinite domain may justify why the experimental jet behaves in the manner shown in Figure 14. The glass tank used is a finite domain and therefore affects the flow of the jet, particularly the fluid further down the jet which is pulled towards the sides of the tank. The fluid here forms a swirling motion and becomes a part of the entrained fluid.

3.6 Reynolds Number

In 1941, Russian scientist Andre Kolmogorov wrote a paper (K41) concerning turbulent flows on large and small scales. The premise for his paper is that all of the energy held by large eddies is dissipated to smaller and smaller eddies until the flow is no longer 'turbulent' i.e. Re < 1. Once the eddies become small enough the energy is dissipated as heat, this scale is called the Taylor micro-scale. For his analysis he made three assumptions about a turbulent flow:

- On the (Kolmogorov) micro-scale the turbulence in the flow is isotropic;
- For very high Reynolds numbers the characteristics of the small scale flow is dependent only on the fluid viscosity, υ, and the energy dissipation rate, ϵ;
- For high Reynolds numbers the characteristics of the intermediate flow (neither small or large scale) is determined by its length scale, r, and ϵ, where $L_0 >> r >> \eta$ (for the duration of this section the subscript '0' will denote the largest scale and η will denote the Kolmogorov micro-scale).

After some dimensional analysis it can be shown that the Kolmogorov scale for a flow is given by $\eta = \left(\frac{\upsilon^3}{\epsilon}\right)^{\frac{1}{4}}$. Furthermore, through the calculations shown in the appendix, the following relationships may also be determined:

$$
\begin{aligned}
&\text{Small Scale} && \text{Large Scale} \\
\epsilon_\eta &= \frac{\upsilon}{t_\eta^2} & \epsilon_0 &= \frac{u_0^3}{L_0} \\
Re_\eta &= \frac{u_\eta \eta}{\upsilon} & Re_0 &= \frac{u_0 L_0}{\upsilon}
\end{aligned}
$$

Finally we can sub the equations above into the expression given for η. This gives an expression relating η and L_0:

$$\eta \sim \frac{L_0}{Re^{\frac{3}{4}}} \tag{27}$$

Equation 27 has the implication that when investigating flows with very high Reynolds numbers, the eddies at the Kolmogorov micro-scale are so small that there is too much data to analyse them accurately. Thus, it appears as though a complete understanding of turbulent flow is still a long way off.

From this arises the problem of accurately measuring Reynolds numbers from experimental images. Is it best to try to measure the small scale Reynolds numbers or the large scale ones? Moreover, the accuracy of the length scale chosen is dependent upon the resolution of the photo.

The turbulent length scale is often estimated as $0.07l$ where l is a characteristic length. For the purpose of this report we'll take l as the radius of the inlet jet, $l = 1.25mm$ thus $L = 0.0875mm$. Taking the entrainment velocity at the base of the jet as $8.7mm/s$ from the PIV calculation and a dynamic viscosity of 8.9 Pa s the Reynolds number can be calculated as:

$$Re = \frac{uL}{\upsilon} = 9.87 \tag{28}$$

Although the colloquial definition for turbulent flow is a Reynolds number greater than 2100, any Reynolds number greater than 1 has inertial forces greater than viscous forces and thus is somewhat turbulent. In relation to this experiment, the speeds that are being dealt with are remarkably low, and therefore the Reynolds number of the flow is not expected to match that of the flow in the wake of a body travelling through water such as a cruise-liner or a sailing boat.

The manner in which the energy disperses through the different scales of eddies is a fundamental property of a turbulent flow. This property is called the *Energy Spectrum Function*, $E(\kappa, \epsilon)$. Through dimensional analysis, it can be shown that

$$E(\kappa, \epsilon) = \kappa^{\frac{-5}{3}} \epsilon^{\frac{2}{3}} \tag{29}$$

This is one of the most prominent results from Kolmogorov's paper as it has been somewhat confirmed by lots of experimental data.

4 Conclusion

The study of turbulent jets is very complex and therefore relies on a number of assumptions in order to make any conclusions. The aim of this report was to assess the validity of these assumptions by investigating a number of jet characteristics.

The inlet velocity was calculated by splitting the jet path into four control volumes: the first to find the speed at the bottom of the inlet tank, the second and fourth to take into account losses at abrupt contractions and sudden expansions and the third control volume to account for pipe losses. Although there are many ways to go about assessing each control volume, the means used in this report best replicate any course material from the lecture courses over the last three years.

Calculation of the spread rate and virtual origin required use of post-processing software and our own judgement on where the line of best fit should lie. This shows that we cannot solely rely on the computer software as it does not always give the best results. Despite this, the experimental results show good correlation with the theory. The average value for both the spread rate and virtual origin were not far off the expected theoretical value. Furthermore, the simulated images show that when the higher resolution images are used, much more accurate results are obtained. This is because the higher resolution images represent a real life turbulent jet much better than the lower resolution images. The low resolution image shows late onset of turbulence proving that this image is a poor representation of a real turbulent jet.

It was discovered during calculating the dilution rate how difficult it is to do this accurately. Whether experimental or simulated data was used, errors were present. Experimental data presented calibration errors, errors with the equipment and systematic measurement errors. The simulated data also produced computational errors. The consequence of this was that attenuation constants could not accurately be calculated, meaning that the function produced for the concentration cannot be accurate.

Entrainment velocity is found to decrease with increasing distance from the nozzle due to the rate of expanding circumference of the jet being faster than the rate of increase of mass flux. Therefore, entrainment velocity is directly proportional to the centreline velocity. However, it is difficult to determine centreline velocity accurately as tracer particles are moving too fast for the camera to determine their positions.

References

[1] Benoit Cushman-Roisin, Environmental Fluid Mechanics Thayer School of Engineering, Dartmouth College, Chapter 9, 2014

[2] A. Lawrie, Submerged Turbulent Jets, 2014

[3] D. Jones, Control Volume Notes, 2012

[4] Herbert Oertel (2004). Prandtl's Essentials of Fluid Mechanics. United States: Springer-Verlag New York Inc. . 152.

Appendices

Calculation of Spread Rate and Virtual Origin

```
% FOR DSC_5900.NEF

nfiles = 0;

for i=5900:5902
    nfiles=nfiles + 1

    infile=sprintf('DSC_%4d.NEF.jpg',i);
    outfile=sprintf('bw_0074.bmp',i);

    imdata=imread(infile); % Read image from file

    %image(imdata); % output image to screen

    imdata=double(imdata);
    sz=size(imdata); % Find the size of the image
    imbw=zeros(sz(1),sz(2)); % Set up a new B&W image array of the same size
    imbw=sqrt(imdata(:,:,1).*imdata(:,:,1)+...
        imdata(:,:,2).*imdata(:,:,2)+...
        imdata(:,:,3).*imdata(:,:,3)); % Compute intensity value from colours
    mx=max(max(imbw)); % Find max value
    mn=min(min(imbw)); % Find min value

    if (mx-mn)>0 % Protect in case image is all zeros
        imbw=(imbw-mn)/(mx-mn); % Normalise to [0,1]
    end

    %image(imbw); %plot the scalar image in matlab colour scheme
    %this may need editing to ensure immean is still created even if the file ...
        list starts after i=1

    if i==5900
        immean=zeros(sz(1),sz(2)); % initialise the cumulative count to zero
    else
        immean=immean+imbw; %add one image each time.
```

```
    end

    immono=zeros(sz(1),sz(2),3);
    immono(:,:,1)=imbw; %use the same value for Red
    immono(:,:,2)=imbw; % Green
    immono(:,:,3)=imbw; % Blue
    %imwrite(immono,outfile,'bmp'); %write out new greyscale file
    image(immono);

end

immean=immean/nfiles; %take the average intensity (outside the loop)
%image(immean) %plot the mean scalar... x256?
imwrite(immean,'mean_2.bmp','bmp');
%process time-average image
clc
immean=imread('mean_2.bmp');
immean=double(immean(:,:,1)); %pick any colour
sz=size(immean);

%constrain the image to the interesting bits in this camera shot
immean=immean; %row/column swap to physical space, origin in top left corner
sz=size(immean); % update size
%take the better illuminated half of the jet
image(immean)
%smooth field before plotting the intensity gradient
nx=sz(1);
ny=sz(2);
%elliptic smoothing to get rid of pixel-frequency noise
for it=1:10
    for ix=2:nx-1
        for iy=2:ny-1
            immean(ix,iy)=0.25*(immean(ix+1,iy)+...
                immean(ix,iy+1)+...
                immean(ix-1,iy)+...
                immean(ix,iy-1));
        end
    end
end
imdx=immean(2:nx,2:ny)-immean(1:nx-1,2:ny); %divided by dx...
imdy=immean(2:nx,2:ny)-immean(2:nx,1:ny-1); %divided by dy...
imgrad=imdx.*imdx+imdy.*imdy;
%image(imgrad); %smooth image

%constrain to 600 x 200 to eliminate artefacts from experimental reflections
%imgrad=imgrad;
sz=size(imgrad);
nx=sz(1);
ny=sz(2);
image(imgrad);
%create an index field
xindex=zeros(nx,ny);
for ix=1:nx
    for iy=1:ny
        xindex(ix,iy)=ix;
    end
```

```
end

%find the (normalised) moment
xmoment=zeros(1,ny);
for iy=1:ny
    num=sum(imgrad(:,iy).*xindex(:,iy));
    den=sum(imgrad(:,iy));
    xmoment(iy)=num/den;
end
figure(2);
axis([0 nx 0 ny]);
yidx=ny:-1:1;
plot(xmoment,yidx);
hold on
%find line of best fit with least squares:
%gradient=(bar(x*y)-bar(x)*bar(y))/(bar(x*x)-bar(x)*bar(x))

b=(mean(xmoment.*yidx)-mean(xmoment)*mean(yidx))/(mean(yidx.*yidx)-mean(yidx)*mean(yidx));
%a=150-b*200; %You should also find a... using Least Squares Regression.
%a = (mean(xmoment)-b)/mean(yidx);
a=mean(xmoment)-b*mean(yidx);

%Solve the simultaneous equations in the slides from the lecture on data ...
    post-processing
xfit=(b*yidx)+a;

plot(xfit,yidx);
%plot(yidx,xfit);
hold off

x=[-1000:1:1000];
%y =((b*x) + a);

z = b*1.75;
%y =((z*x)+60);
y1 =((-z*x) +500);

figure(3)
%set(gca,'Ydir','reverse')
I = imread('mean_2.bmp');
imshow(I)
image(I)
hold on
%plot(x,y)
%set(gca,'Xdir','reverse')

plot(x,y1)

hold off
```

Calculation of attenuation constants

```
clear; clc;
filemin=1; %put your own favourite file number in here...
filemax=3;
```

```
ifile=0;
xlaser=2442; %source position
zlaser=0; %source position
divergence=120; %spread angle of laser sheet
nrays=121; %refinement of ray casting
raylength=2847; %how many increments along ray
dzray=1; % vertical increment of ray on grid
a=0.01; %attenuation coefficient proportional to concentration
b=0.001; %background level of attenuation

for i = filemin:filemax
    filename=sprintf('1_%d.jpg',i);
    if (exist(filename)==2)
        ifile=ifile+1;
        %This is where the hard work begins
        scal=imread(filename); % read image
        scal=double(scal);
        scal=scal(:,:,1); %greyscale, so take only one component
        scal=scal/2847;
        %image(scal*256); %needs range 0<p<256 to plot picture
        scal=scal'; %x is now horizontal, z vertical
        %scal(:,:)=0.0;
        %scal(32:96,:)=1.0;
        %create array for light ray positions
        aray=(-divergence/2:divergence/2)*pi/180; %angle of ray
        dxray=dzray*tan(aray); %the dx across for each dz down
        xray=zeros(raylength,nrays); %position of ray
        zray=zeros(raylength,nrays); %position of ray
        pray=zeros(raylength,nrays); %intensity of ray
        xdom=size(scal);
        xdom=xdom(1);
        xray(1,:)=xlaser; %initial position
        zray(1,:)=zlaser;
        pray(1,:)=1.0; %normalised intensity
        %following each ray along its length
        for ir=2:raylength
            %update the current x and z positions
            xray(ir,:)=xray(ir-1,:)+dxray(1,:);
            zray(ir,:)=zray(ir-1,:)+dzray;
            %stop rays running of the left and right edges of the domain
            for jr=1:nrays
                if xray(ir,jr) < 3
                    xray(ir,jr)=3;
                end
                if xray(ir,jr) > xdom-2
                    xray(ir,jr)=xdom-2;
                end
            end

            xf=int16(floor(xray(ir,:))); %decide which cell ray leaves from
            xfm=int16(floor(xray(ir-1,:))); %decide which cell ray arrives
            for jr=1:nrays %stop rays running off the edge
                if xf(jr) < 3
                    xf(jr)=3;
                end
                if xf(jr) > xdom-2
```

```
                xf(ix)=xdom-2;
            end
            if xfm(jr) < 3
                xfm(jr)=3;
            end
            if xfm(jr) > xdom-2
                xfm(jr)=xdom-2;
            end
        end
        divsin=sin(aray);
        divsin=1.0./divsin;
        ltmp=sqrt(dxray.*dxray+dzray*dzray);
        ptmp=pray(ir-1,:);
        conc=scal(:,ir-1);
        conc=conc';

        for jr=1:nrays
            if xf(jr)-xfm(jr) == 0 %crosses no faces (one cell only)
                eta=(a*conc(xf(jr))+b); %use calibration to calculate ...
                    attenuation rate
                ptmp(jr)=ptmp(jr)*exp(-eta*ltmp(jr)); %exponential decay ...
                    of intensity
            end
            if xf(jr)-xfm(jr) == 1 %crosses one face to the right
                ltmp1= (double(xf(jr))-xray(ir-1,jr))*divsin(jr); %hypotenuse
                eta=(a*conc(xfm(jr))+b);
                ptmp(jr)=ptmp(jr)*exp(-eta*ltmp1);
                ltmp2= ltmp(jr)-ltmp1;
                eta=(a*conc(xfm(jr)+1)+b);
                ptmp(jr)=ptmp(jr)*exp(-eta*ltmp2);
            end
            if xf(jr)-xfm(jr) == 2 %crosses two faces to the right
                ltmp1= (double(xf(jr))-1-xray(ir-1,jr))*divsin(jr);
                eta=(a*conc(xfm(jr))+b);
                ptmp(jr)=ptmp(jr)*exp(-eta*ltmp1);
                ltmp2= (double(xf(jr))-xray(ir-1,jr))*divsin(jr);
                ltmp2= ltmp2-ltmp1;
                eta=(a*conc(xfm(jr)+1)+b);
                ptmp(jr)=ptmp(jr)*exp(-eta*ltmp2);
                ltmp3= ltmp(jr)-ltmp1-ltmp2;
                eta=(a*conc(xfm(jr)+2)+b);
                ptmp(jr)=ptmp(jr)*exp(-eta*ltmp3);
            end
            if xf(jr)-xfm(jr) == -1 % crosses one face to the left
                ltmp1= (xray(ir-1,jr)-double(xfm(jr)))*divsin(jr);
                eta=(a*conc(xfm(jr))+b);
                ptmp(jr)=ptmp(jr)*exp(-eta*ltmp1);
                ltmp2= ltmp(jr)-ltmp1;
                eta=(a*conc(xfm(jr)-1)+b);
                ptmp(jr)=ptmp(jr)*exp(-eta*ltmp2);
            end
            if xf(jr)-xfm(jr)== -2 % crosses two faces to the left
                ltmp1= (xray(ir-1,jr)-double(xfm(jr)))*divsin(jr);
                eta=(a*conc(xfm(jr))+b);
                ptmp(jr)=ptmp(jr)*exp(-eta*ltmp1);
                ltmp2= (xray(ir-1,jr)-double(xfm(jr)+1))*divsin(jr);
```

```
                    ltmp2= ltmp(jr)-ltmp1;
                    eta=(a*conc(xfm(jr)-1)+b);
                    ptmp(jr)=ptmp(jr)*exp(-eta*ltmp2);
                    ltmp3= ltmp(jr)-ltmp1-ltmp2;
                    eta=(a*conc(xfm(jr)-2)+b);
                    ptmp(jr)=ptmp(jr)*exp(-eta*ltmp3);
                end
            end
            pray(ir,:)=ptmp; %update final intensity array
        end
    end
    av(i,:)=mean(pray,2);
end


x=1:2847;
x=x';
y1 = (av(1,:))';
y2 = (av(2,:))';
y3 = (av(3,:))';
f1 = fit(x, y1, 'exp1');
f2 = fit(x, y2, 'exp1');
f3 = fit(x, y3, 'exp1');

p11 = predint(f1,x,0.95,'observation','off');

mycoeffs_1 = coeffvalues(f1);
mycoeffs_2 = coeffvalues(f2);
mycoeffs_3 = coeffvalues(f3);
coeff_1_1 = mycoeffs_1(1);
coeff_1_2 = mycoeffs_1(2);
coeff_2_1 = mycoeffs_2(1);
coeff_2_2 = mycoeffs_2(2);
coeff_3_1 = mycoeffs_3(1);
coeff_3_2 = mycoeffs_3(2);

ac_1 = -coeff_1_2;
ac_2 = -coeff_2_2;
ac_3 = -coeff_3_2;

figure(2);
hold on
plot(x, y1, 'k')
h1 = plot(f1, 'b');
plot(x, y2, 'k')
h2 = plot(f2, 'r');
plot(x, y3, 'k')
h3 = plot(f3, 'g');
set(h1, 'LineWidth', 1.5);
set(h2, 'LineWidth', 1.5);
set(h3, 'LineWidth', 1.5);
axis([ 0 2847 0 1]);
ylabel('Relative Light Intensity');
xlabel('Distance along ray (Z)/ px');
legend([h1 h2 h3],{'Fit for Concentration 2','Fit for Concentration 1', 'Fit ...
    for Concentration 3'})
```

```
line([.62 .7],[1 1],'Linestyle',':','color','k')
text(2000,.85,[ 'Equation of exponential fit for concentration 1= ' ...
    num2str(coeff_2_1,'%.2f'),' exp(-' num2str(-coeff_2_2,'%.5f') ' s)    ' ...
    ],'EdgeColor','k')
text(2000,.80,[ 'Equation of exponential fit for concentration 2= ' ...
    num2str(coeff_1_1,'%.2f'),' exp(-' num2str(-coeff_1_2,'%.5f') ' s)    ' ...
    ],'EdgeColor','k')
text(2000,.75,[ 'Equation of exponential fit for concentration 3= ' ...
    num2str(coeff_3_1,'%.2f'),' exp(-' num2str(-coeff_3_2,'%.5f') ' s)    ' ...
    ],'EdgeColor','k')
set(gca,'box','on')
hold off


for f= 1:2847
    rsquared1(f) = 1 - (norm(f1(f) - av(1,f))/norm(av(1,f) - (mean(av(1,:)))))^2;
    rsquared2(f) = 1 - (norm(f2(f) - av(2,f))/norm(av(2,f) - (mean(av(2,:)))))^2;
    rsquared3(f) = 1 - (norm(f3(f) - av(3,f))/norm(av(3,f) - (mean(av(3,:)))))^2;

end

mean(rsquared1);
mean(rsquared2);
mean(rsquared3);

f1l = ones(1, 2847);
f2l = ones(1, 2847);
f3l = ones(1, 2847);
for l = 1: 2847
    f1l(l) = log(coeff_1_1*exp(coeff_1_2*l));
    f2l(l) = log(coeff_2_1*exp(coeff_2_2*l));
    f3l(l) = log(coeff_3_1*exp(coeff_3_2*l));
end

w = 115
f1l(w)
f2l(w)

ac_1_new = (-f1l(w)/w) - (-f2l(w)/w)
b_new = -f2l(w)/w + ac_1_new

figure(5)
hold on
plot(x, f1l, 'b');
plot(x, f2l, 'r');
%plot(x, f3l, 'g');
ylabel('ln(I_{n})');
xlabel('Distance along the ray (Z) / px');


relative_1_2 = zeros(1,2847);
relative_2_3 = zeros(1,2847);
relative_1_3 = zeros(1,2847);
for s = 1:2847
    relative_1_2(s) = ...
        log(coeff_1_1*exp(coeff_1_2*s)/coeff_2_1*exp(coeff_2_2*s))/s^2;
```

```
    relative_2_3(s) = ...
        log(coeff_2_1*exp(coeff_2_2*s)/coeff_3_1*exp(coeff_3_2*s))/s^2;
end


x=1:2847;
x=x';
rand1 = rand(1,2847);
rand2 = rand(1,2847);
r1 = -rand1 .* relative_1_2;
r2 = -rand2 .* relative_2_3;
r1plot = r1(1:13:end);
xplot = x(1:13:end);
av_1_2 = mean(r1);
av_2_3 = mean(r2);
av12=zeros(219,1);
av12(:,1) = av_1_2;
av23=zeros(219,1);
av23(:,1) = av_2_3;
figure(3);
hold on
plot (xplot,r1plot,'r.')
plot (x, 0, 'k-');
plot(xplot, av12,'--k', 'LineWidth', 2);
ylabel('Concentration, aC_1');
xlabel('Distance along ray (Z) / px');
line([.62 .7],[1 1],'Linestyle',':','color','k')
text(1900,.000085, 'Concentration calculated using I(1) and I(2)' ...
    ,'EdgeColor','k')
set(gca,'box','on')
hold off
axis([0 2847 -0.001 0.001]);
axis square
```

Experimental Dilution Rate

```
clear; clc;
filemin=11; %put your own favourite file number in here...
filemax=11;
ifile=0;
xlaser=(4608*22)/34; %source position
zlaser=0; %source position
divergence=120; %spread angle of laser sheet
nrays=121; %refinement of ray casting
raylength=3072; %how many increments along ray
dzray=1; % vertical increment of ray on grid
a=0.01; %attenuation coefficient proportional to concentration
b=0.001; %background level of attenuation

for i = filemin:filemax
    filename=sprintf('1_%d.jpg',i);
    if (exist(filename)==2)
        ifile=ifile+1;
        %This is where the hard work begins
        scal=imread(filename); % read image
```

```
        scal=double(scal);
        scal=scal(:,:,1); %greyscale, so take only one component
        scal=scal/3072;
        %image(scal*256); %needs range 0<p<256 to plot picture
        scal=scal'; %x is now horizontal, z vertical
        %scal(:,:)=0.0;
        %scal(32:96,:)=1.0;
        %create array for light ray positions
        aray=(-divergence/2:divergence/2)*pi/180; %angle of ray
        dxray=dzray*tan(aray); %the dx across for each dz down
        xray=zeros(raylength,nrays); %position of ray
        zray=zeros(raylength,nrays); %position of ray
        pray=zeros(raylength,nrays); %intensity of ray
        xdom=size(scal);
        xdom=xdom(1);
        xray(1,:)=xlaser; %initial position
        zray(1,:)=zlaser;
        pray(1,:)=1.0; %normalised intensity
        %following each ray along its length
        for ir=2:raylength
            %update the current x and z positions
            xray(ir,:)=xray(ir-1,:)+dxray(1,:);
            zray(ir,:)=zray(ir-1,:)+dzray;
            %stop rays running of the left and right edges of the domain
            for jr=1:nrays
                if xray(ir,jr) < 3
                    xray(ir,jr)=3;
                end
                if xray(ir,jr) > xdom-2
                    xray(ir,jr)=xdom-2;
                end
            end
            %during the descent by dzray the most extreme ray at 60 degrees
            %could at most traverse three square cells (tan -1 2 = 63 deg)
            xf=int16(floor(xray(ir,:))); %decide which cell ray leaves from
            xfm=int16(floor(xray(ir-1,:))); %decide which cell ray arrives
            for jr=1:nrays %stop rays running off the edge
                if xf(jr) < 3
                    xf(jr)=3;
                end
                if xf(jr) > xdom-2
                    xf(ix)=xdom-2;
                end
                if xfm(jr) < 3
                    xfm(jr)=3;
                end
                if xfm(jr) > xdom-2
                    xfm(jr)=xdom-2;
                end
            end
            divsin=sin(aray);
            divsin=1.0./divsin;
            ltmp=sqrt(dxray.*dxray+dzray*dzray);
            ptmp=pray(ir-1,:);
            conc=scal(:,ir-1);
            conc=conc';
```

```
            %eta=(a*conc(xf)+b);
            %ptmp=ptmp.*exp(-eta.*ltmp);
            %work out geometry of face-crossings
            for jr=1:nrays
                if xf(jr)-xfm(jr) == 0 %crosses no faces (one cell only)
                    eta=(a*conc(xf(jr))+b); %use calibration to calculate ...
                        attenuation rate
                    ptmp(jr)=ptmp(jr)*exp(-eta*ltmp(jr)); %exponential decay ...
                        of intensity
                end
                if xf(jr)-xfm(jr) == 1 %crosses one face to the right
                    ltmp1= (double(xf(jr))-xray(ir-1,jr))*divsin(jr); %hypotenuse
                    eta=(a*conc(xfm(jr))+b);
                    ptmp(jr)=ptmp(jr)*exp(-eta*ltmp1);
                    ltmp2= ltmp(jr)-ltmp1;
                    eta=(a*conc(xfm(jr)+1)+b);
                    ptmp(jr)=ptmp(jr)*exp(-eta*ltmp2);
                end
                if xf(jr)-xfm(jr) == 2 %crosses two faces to the right
                    ltmp1= (double(xf(jr))-1-xray(ir-1,jr))*divsin(jr);
                    eta=(a*conc(xfm(jr))+b);
                    ptmp(jr)=ptmp(jr)*exp(-eta*ltmp1);
                    ltmp2= (double(xf(jr))-xray(ir-1,jr))*divsin(jr);
                    ltmp2= ltmp2-ltmp1;
                    eta=(a*conc(xfm(jr)+1)+b);
                    ptmp(jr)=ptmp(jr)*exp(-eta*ltmp2);
                    ltmp3= ltmp(jr)-ltmp1-ltmp2;
                    eta=(a*conc(xfm(jr)+2)+b);
                    ptmp(jr)=ptmp(jr)*exp(-eta*ltmp3);
                end
                if xf(jr)-xfm(jr) == -1 % crosses one face to the left
                    ltmp1= (xray(ir-1,jr)-double(xfm(jr)))*divsin(jr);
                    eta=(a*conc(xfm(jr))+b);
                    ptmp(jr)=ptmp(jr)*exp(-eta*ltmp1);
                    ltmp2= ltmp(jr)-ltmp1;
                    eta=(a*conc(xfm(jr)-1)+b);
                    ptmp(jr)=ptmp(jr)*exp(-eta*ltmp2);
                end
                if xf(jr)-xfm(jr)== -2 % crosses two faces to the left
                    ltmp1= (xray(ir-1,jr)-double(xfm(jr)))*divsin(jr);
                    eta=(a*conc(xfm(jr))+b);
                    ptmp(jr)=ptmp(jr)*exp(-eta*ltmp1);
                    ltmp2= (xray(ir-1,jr)-double(xfm(jr)+1))*divsin(jr);
                    ltmp2= ltmp(jr)-ltmp1;
                    eta=(a*conc(xfm(jr)-1)+b);
                    ptmp(jr)=ptmp(jr)*exp(-eta*ltmp2);
                    ltmp3= ltmp(jr)-ltmp1-ltmp2;
                    eta=(a*conc(xfm(jr)-2)+b);
                    ptmp(jr)=ptmp(jr)*exp(-eta*ltmp3);
                end
            end
            pray(ir,:)=ptmp; %update final intensity array
        end
    end
    av(i,:)=mean(pray,2);
end
```

```
jetline = 4608/2;
figure(1)
hold on
plot(1:raylength, jetline, '-k');
plot(1:raylength, xlaser, '-r');
contour(zray,xray,pray,100);
axis([ 0 3072 0 4608]);
ylabel('Horizontal distance (x) / px');
xlabel('Distance along ray (Z) / px');
```

Simulated Dilution Rate

```
clear; clc;
filemin=13; %put your own favourite file number in here...
filemax=13;
ifile=0;
xlaser=227/2; %source position
zlaser=0; %source position
divergence=120; %spread angle of laser sheet
nrays=121; %refinement of ray casting
raylength=279; %how many increments along ray
dzray=1; % vertical increment of ray on grid
a=0.01; %attenuation coefficient proportional to concentration
b=0.001; %background level of attenuation

for i = filemin:filemax
    filename=sprintf('1_%d.jpg',i);
    if (exist(filename)==2)
        ifile=ifile+1;
        %This is where the hard work begins
        scal=imread(filename); % read image
        scal=double(scal);
        scal=scal(:,:,1); %greyscale, so take only one component
        scal=scal/279;
        %image(scal*256); %needs range 0<p<256 to plot picture
        scal=scal'; %x is now horizontal, z vertical
        %scal(:,:)=0.0;
        %scal(32:96,:)=1.0;
        %create array for light ray positions
        aray=(-divergence/2:divergence/2)*pi/180; %angle of ray
        dxray=dzray*tan(aray); %the dx across for each dz down
        xray=zeros(raylength,nrays); %position of ray
        zray=zeros(raylength,nrays); %position of ray
        pray=zeros(raylength,nrays); %intensity of ray
        xdom=size(scal);
        xdom=xdom(1);
        xray(1,:)=xlaser; %initial position
        zray(1,:)=zlaser;
        pray(1,:)=1.0; %normalised intensity
        %following each ray along its length
        for ir=2:raylength
            %update the current x and z positions
            xray(ir,:)=xray(ir-1,:)+dxray(1,:);
            zray(ir,:)=zray(ir-1,:)+dzray;
```

```
%stop rays running of the left and right edges of the domain
for jr=1:nrays
    if xray(ir,jr) < 3
        xray(ir,jr)=3;
    end
    if xray(ir,jr) > xdom-2
        xray(ir,jr)=xdom-2;
    end
end
%during the descent by dzray the most extreme ray at 60 degrees
%could at most traverse three square cells (tan -1 2 = 63 deg)
xf=int16(floor(xray(ir,:))); %decide which cell ray leaves from
xfm=int16(floor(xray(ir-1,:))); %decide which cell ray arrives
for jr=1:nrays %stop rays running off the edge
    if xf(jr) < 3
        xf(jr)=3;
    end
    if xf(jr) > xdom-2
        xf(ix)=xdom-2;
    end
    if xfm(jr) < 3
        xfm(jr)=3;
    end
    if xfm(jr) > xdom-2
        xfm(jr)=xdom-2;
    end
end
divsin=sin(aray);
divsin=1.0./divsin;
ltmp=sqrt(dxray.*dxray+dzray*dzray);
ptmp=pray(ir-1,:);
conc=scal(:,ir-1);
conc=conc';
for jr=1:nrays
    if xf(jr)-xfm(jr) == 0 %crosses no faces (one cell only)
        eta=(a*conc(xf(jr))+b); %use calibration to calculate ...
            attenuation rate
        ptmp(jr)=ptmp(jr)*exp(-eta*ltmp(jr)); %exponential decay ...
            of intensity
    end
    if xf(jr)-xfm(jr) == 1 %crosses one face to the right
        ltmp1= (double(xf(jr))-xray(ir-1,jr))*divsin(jr); %hypotenuse
        eta=(a*conc(xfm(jr))+b);
        ptmp(jr)=ptmp(jr)*exp(-eta*ltmp1);
        ltmp2= ltmp(jr)-ltmp1;
        eta=(a*conc(xfm(jr)+1)+b);
        ptmp(jr)=ptmp(jr)*exp(-eta*ltmp2);
    end
    if xf(jr)-xfm(jr) == 2 %crosses two faces to the right
        ltmp1= (double(xf(jr))-1-xray(ir-1,jr))*divsin(jr);
        eta=(a*conc(xfm(jr))+b);
        ptmp(jr)=ptmp(jr)*exp(-eta*ltmp1);
        ltmp2= (double(xf(jr))-xray(ir-1,jr))*divsin(jr);
        ltmp2= ltmp2-ltmp1;
        eta=(a*conc(xfm(jr)+1)+b);
        ptmp(jr)=ptmp(jr)*exp(-eta*ltmp2);
```

```
                    ltmp3= ltmp(jr)-ltmp1-ltmp2;
                    eta=(a*conc(xfm(jr)+2)+b);
                    ptmp(jr)=ptmp(jr)*exp(-eta*ltmp3);
                end
                if xf(jr)-xfm(jr) == -1 % crosses one face to the left
                    ltmp1= (xray(ir-1,jr)-double(xfm(jr)))*divsin(jr);
                    eta=(a*conc(xfm(jr))+b);
                    ptmp(jr)=ptmp(jr)*exp(-eta*ltmp1);
                    ltmp2= ltmp(jr)-ltmp1;
                    eta=(a*conc(xfm(jr)-1)+b);
                    ptmp(jr)=ptmp(jr)*exp(-eta*ltmp2);
                end
                if xf(jr)-xfm(jr)== -2 % crosses two faces to the left
                    ltmp1= (xray(ir-1,jr)-double(xfm(jr)))*divsin(jr);
                    eta=(a*conc(xfm(jr))+b);
                    ptmp(jr)=ptmp(jr)*exp(-eta*ltmp1);
                    ltmp2= (xray(ir-1,jr)-double(xfm(jr)+1))*divsin(jr);
                    ltmp2= ltmp(jr)-ltmp1;
                    eta=(a*conc(xfm(jr)-1)+b);
                    ptmp(jr)=ptmp(jr)*exp(-eta*ltmp2);
                    ltmp3= ltmp(jr)-ltmp1-ltmp2;
                    eta=(a*conc(xfm(jr)-2)+b);
                    ptmp(jr)=ptmp(jr)*exp(-eta*ltmp3);
                end
            end
            pray(ir,:)=ptmp; %update final intensity array
        end
    end
    average = mean(pray,2);

end
figure(10)
plot(pray)

jetline = 227/2;
figure(1)
hold on
plot(1:raylength, jetline, '-k');
plot(1:raylength, xlaser, '-r');
contour(zray,xray,pray,100);
axis([ 0 256 81.5 145.5]);
ylabel('Horizontal distance (x) / px');
xlabel('Distance along ray (Z) / px');

x=1:279;
x=x';
f1 = fit(x, average, 'exp1');
mycoeffs_1 = coeffvalues(f1);
coeff_1_1 = mycoeffs_1(1);
coeff_1_2 = mycoeffs_1(2);
f1x = differentiate(f1,x);
figure(2);
hold on
plot(x, average, 'k', 'LineWidth', 2);
co = polyfit(x, average, 5);
newy = polyval(co, x);
```

```
plot(x, newy, ':r', 'LineWidth', 2);
h1 = plot(f1, 'b');
line([.62 .7],[1 1],'Linestyle',':','color','k')
text(220,.92,[ 'Equation of exponential fit = ' num2str(coeff_1_1,'%.2f'),' ...
    exp(-' num2str(-coeff_1_2,'%.5f') ' s)    ' ],'EdgeColor','k')
set(gca,'box','on')
set(gca,'box','on')
set(h1, 'LineWidth', 1.5);
axis([ 0 279 0.4 1]);
ylabel('Relative Light Intensity');
xlabel('Distance along ray / px');
hold off
```

Kolmogorov Dimensional Analysis

Below is the detailed workings for the Kolmogorov dimensional analysis applicable to the section on Reynolds numbers.

$$\epsilon = \frac{u^2}{T} = \frac{L^2}{T^3} \tag{30}$$

$$\upsilon = \frac{L^2}{T} \tag{31}$$

Therefore, from equations 30 and 31 we can see that:

$$\frac{\upsilon}{\epsilon} \sim T^2 \tag{32}$$

$$\upsilon\epsilon \sim \frac{L^4}{T^4} \tag{33}$$

$$T \sim \left(\frac{\upsilon}{\epsilon}\right)^{\frac{1}{4}} \tag{34}$$

$$u \sim (\upsilon\epsilon)^{\frac{1}{4}} \tag{35}$$

Thus from Speed = $\frac{\text{Distance}}{\text{Time}}$ and the equations 34 and 35:

$$\eta = u_\eta T_\eta \sim \left(\frac{\upsilon^3}{\epsilon}\right)^{\frac{1}{4}} \tag{36}$$

From equations 34 and 35 expressions for ϵ can be found for the small and large scales.

$$\epsilon_\eta = \frac{\upsilon}{t_\eta^2} \tag{37}$$

$$Re_\eta = \frac{u_\eta \eta}{\upsilon} \tag{38}$$

$$\epsilon_0 = \frac{u_0^3}{L_0} \tag{39}$$

$$Re_0 = \frac{u_0 L_0}{\upsilon} \tag{40}$$