

# Advanced Laboratory II

---

## Flow Instabilities

Luke Kilmartin

---

Student ID: 18302711

---

**Supervisor:** Assoc. Professor Brian Vohnsen



UCD School of Physics  
University College Dublin

October 18, 2021

# Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
1.1	Background . . . . .	2
1.2	Theory . . . . .	2
1.2.1	Fluid Dynamics . . . . .	2
1.2.2	Kolmogorov flow . . . . .	3
<b>2</b>	<b>Apparatus and Experimental Procedure</b>	<b>4</b>
2.1	Apparatus . . . . .	4
2.1.1	Magnet array . . . . .	4
2.1.2	The electrolyte . . . . .	4
2.1.3	Recording the fluid flow . . . . .	5
2.2	Experimental Procedure . . . . .	5
2.2.1	Measurement of the effect of viscosity on the fluid instability . . . . .	5
<b>3</b>	<b>Analysis &amp; Results</b>	<b>6</b>
3.1	Vorticity Fields . . . . .	6
3.1.1	Particle Tracking . . . . .	6
3.1.2	Velocity Fields . . . . .	6
3.1.3	Determining the vorticity field . . . . .	7
3.2	Investigation of the effect of viscosity on fluid instability - Extension to the experiment	10
3.3	Investigation of the instability onset . . . . .	12
<b>4</b>	<b>Discussion</b>	<b>13</b>
<b>5</b>	<b>Conclusions</b>	<b>14</b>
<b>6</b>	<b>Appendix</b>	<b>15</b>
6.1	Finding the velocity field from the image files . . . . .	15
6.1.1	Obtaining background image . . . . .	15
6.1.2	Obtaining the particle tracks . . . . .	15
6.1.3	Plotting the velocity field . . . . .	16
6.2	Finding the vorticity field given the velocity field . . . . .	16
6.2.1	Velocity components interpolated over a regular grid . . . . .	16
6.2.2	Using the spatial gradients of the velocity components to plot the vorticity . .	19
6.3	Investigating the root mean squared $u_x$ velocity component . . . . .	20
6.4	Investigating the FPS of the camera, and attempts made to obtain only one image per frame . . . . .	22

## Abstract

A classical fluid instability was studied in three unique mixtures of CuSO<sub>4</sub>, glycerol and deionised water. This was done using seeding particles, a camera and particle tracking methods. Electromagnetically driven thin-layer flow was used as a quasi-two-dimensional approximation for Kolmogorov flow. The spatial structure of the flow was measured, and the effect of the drive current was observed. The effect of the fluid viscosity on the fluid instability was determined, and the onset of instability in the fluids was investigated.

# 1 Introduction

## 1.1 Background

Andrey Kolmogorov was a mathematician who contributed to many fields, including the study of turbulence - a branch of fluid dynamics. In the late 1950s, Kolmogorov was interested in the stability criteria of two-dimensional incompressible shear flows using spatially periodic forcing. [1] This system is now referred to as Kolmogorov flow, and the instability has been the subject of many studies over the years.

Flow instabilities are fundamental in the understanding of flow pattern formation. The differential equations which govern fluid dynamics (such as the Navier-Stokes equations), have no unique solution. As a result, the stability of a system determines which solution will be observed. A stable system is one in which the flow pattern is unchanged despite perturbations, and the stability of a system changes with the flow parameters, such as velocity, pressure, and density. If the system's sensitivity to perturbations increases, an instability may occur, resulting in a global transition in the flow. [2]

Flow stability is important in many real-world applications, such as in understanding the billions of kilometers of pipes for utilities across the world. If turbulence can be reduced in such systems, less energy need be expended to pump fluids.

## 1.2 Theory

### 1.2.1 Fluid Dynamics

The Navier-Stokes equations are second-order nonlinear PDEs which describe the motion of viscous fluids. The conservation of momentum is described by the incompressible momentum Navier-Stokes equation given in Equation 1. [2] The assumption of incompressible flow holds well with fluids at low Mach numbers (i.e. with speeds much less than the speed of sound), and so is ideal for this experiment.

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} = -\frac{1}{\rho} \nabla p + \nu \nabla^2 \mathbf{u} + \mathbf{f} \quad (1)$$

The velocity field is  $\mathbf{u}$ ,  $t$  is time,  $\rho$  is the mass density,  $p$  is the pressure field,  $\nu$  is the kinematic viscosity, and  $\mathbf{f}$  is the applied external forces per unit mass.  $\frac{\partial \mathbf{u}}{\partial t}$  is the change of velocity with time,  $(\mathbf{u} \cdot \nabla) \mathbf{u}$  is the convective term,  $-\frac{1}{\rho} \nabla p$  is the pressure gradient, and  $\nu \nabla^2 \mathbf{u}$  is the diffusion term.

The continuity equation for incompressible fluids

$$\nabla \cdot \mathbf{u} = 0 \quad (2)$$

must also be satisfied.

The curl of Equation 1 gives the vorticity equation [3]

$$\frac{\partial \boldsymbol{\omega}}{\partial t} + (\mathbf{u} \cdot \nabla) \boldsymbol{\omega} - (\boldsymbol{\omega} \cdot \nabla) \mathbf{u} = \nu \nabla^2 \boldsymbol{\omega} + \mathbf{F} \quad (3)$$

where the vorticity is  $\nabla \times \vec{u} \equiv \vec{\omega}$  and the final term is the forcing  $\mathbf{F} = \nabla \times \mathbf{f}$ .

### 1.2.2 Kolmogorov flow

Kolmogorov flow (k-flow) is generated by a stationary sinusoidal force that varies in space, as in Equation 4. [1]

$$\mathbf{F} = F_0 \sin \frac{2\pi x}{L} \hat{z} \quad (4)$$

A solution [4] to Equations 1-4 is given by

$$\mathbf{u}_0 = \sqrt{3}U \cos \frac{2\pi x}{L} \hat{\mathbf{y}} \quad (5)$$

with a vorticity field given by

$$\boldsymbol{\omega}_0 = \nabla \times \mathbf{u}_0 = -\sqrt{3}U \frac{2\pi}{L} \sin \frac{2\pi x}{L} \hat{z} \quad (6)$$

where  $U = \langle \mathbf{u}_0 \cdot \mathbf{u}_0 \rangle^{1/2}$  is the root-mean-squared velocity and  $\frac{L}{2}$  is the width of the stripes. This solution describes the velocity as a repeating pattern of steady stripes of alternating velocity, with associated stripes of alternating vorticity.

Kolmogorov flow can be studied in the laboratory by creating an approximation of two-dimensional flow via the use of electromagnetically driven thin-layer flow, as is discussed in .

## 2 Apparatus and Experimental Procedure

### 2.1 Apparatus

The set-up of the apparatus is shown below in Figure 1.

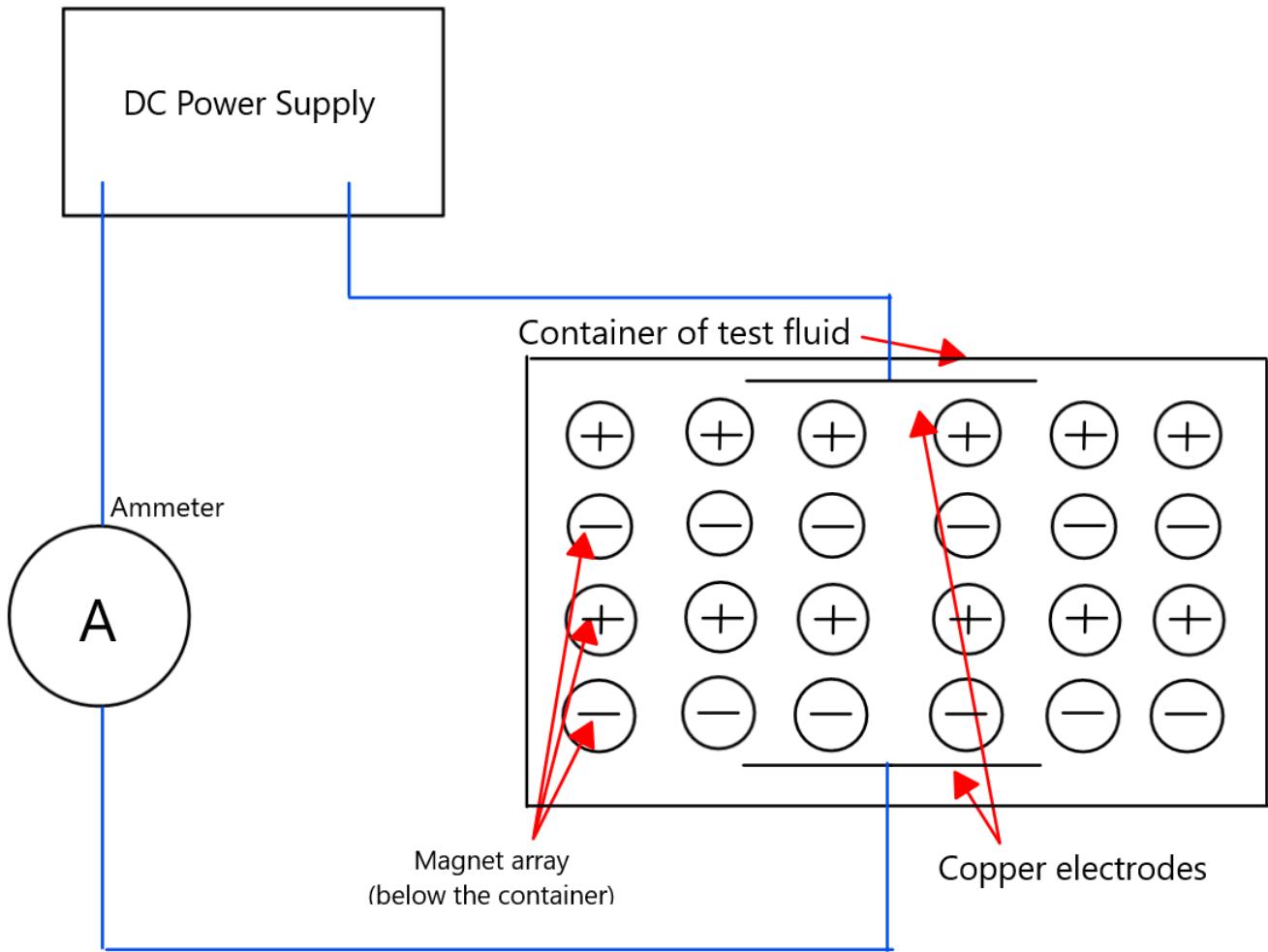


Figure 1: Schematic diagram of the apparatus used in the investigation of the classical fluid instability of a quasi-two-dimensional flow pattern. The bottom of the container was covered in a black, rubbery material to minimise light reflection.

The apparatus also included glycerol, deionised water and Copper(II) Sulfate ( $\text{CuSO}_4$ ), along with a 500  $ml$  glass beaker, a 100  $ml$  plastic graduated cylinder, and a tub of 50  $\mu\text{m}$  round Polyamide Seeding Particles. Nitrile gloves were worn at all times when dealing with chemicals.

#### 2.1.1 Magnet array

A  $6 \times 5$  array of permanent magnets was kept between acrylic sheets, under the plastic container (Figure 1 above shows a  $4 \times 6$  array, for illustrative purposes). They produced a magnetic field of  $B = 0.33$  T at their surface. Each magnet was 5  $mm$  thick with a diameter of 13  $mm$ . Alternating rows differed in polarity, with the topmost row of magnets facing positive side up.

#### 2.1.2 The electrolyte

An electrolyte was made up of 10%  $\text{CuSO}_4$  by mass in a 300  $ml$  mixture of de-ionised water (90% by volume) & glycerol (10% by volume), then mixed in the beaker using a magnetic stirrer at 400

RPM for  $\sim 5$  minutes to ensure homogeneity. The fluid was poured to a depth of  $\sim 10\text{ mm}$  into the plastic container of lateral dimensions  $170\text{ mm} \times 165\text{ mm}$ .

A rectangular copper electrode was set up horizontally on each side of the fluid layer, connected to the DC power supply (max. 30 V) and to a multimeter used as an ammeter.

As a current  $I$  is passed through the electrolyte over the magnetic array, a Lorentz force per unit mass of

$$\mathbf{f}_B = \frac{\mathbf{J} \times \mathbf{B}}{\rho} \quad (7)$$

is caused.  $\mathbf{J}$  is the current density. This setup is known as electromagnetically driven thin-layer flow, and the Lorentz force per unit mass produces bulk motion in the fluid, which can be observed.

### 2.1.3 Recording the fluid flow

In order to track the fluid flow,  $50\text{ }\mu\text{m}$  round Polyamide Seeding Particles (with densities of  $1.03\text{ g/cm}^3$ ) were added to float atop the fluid. [5] The seeding particles absorb light most strongly in the blue ( $468\text{ nm}$ ). A clamp held a blue LED bank, positioned to illuminate the experiment from above, allowing the seeding particles to be tracked easily.

A monochrome PCB camera was held in a clamp directly above the centre of the container, facing down at the test fluid. Care was taken to ensure that the camera was parallel to the electrolyte, and aligned with the axis of the magnet array. The monochrome PCB camera had a manual focus lens, and recorded frames with a height of 576 pixels and a width of 720 pixels. It is suspected that the camera had a variable frame rate (see limitations discussed in section 4), which made accurate measurements very difficult.

## 2.2 Experimental Procedure

Ensuring that (excluding the blue LED bank), the lights in the room were off, the power supply's voltage was increased from 0 V until the ammeter read a current  $\mathbf{I}$  of  $20\text{ mA}$ . After a few seconds to allow the fluid to settle into its new flow, a  $\sim 15\text{ s}$  video was recorded (in '.mp4' file format) using VLC Media Player.

This step was repeated in steps of  $10\text{ mA}$  up to a maximum of  $110\text{ mA}$ . Limiting conditions on the maximum current were the maximum voltage of the power supply, as well as a dramatic steady decrease in current on the ammeter once the current goes above  $\sim 120\text{ mA}$ . The cause for this decrease is unknown.

### 2.2.1 Measurement of the effect of viscosity on the fluid instability

The container and beakers were cleaned, and the process was repeated for two more electrolytes of differing viscosities:

- 10%  $\text{CuSO}_4$  by mass in a  $300\text{ ml}$  mixture of de-ionised water (80% by volume) & glycerol (20% by volume)
- 10%  $\text{CuSO}_4$  by mass in a  $300\text{ ml}$  mixture of de-ionised water (70% by volume) & glycerol (30% by volume)

### 3 Analysis & Results

#### 3.1 Vorticity Fields

##### 3.1.1 Particle Tracking

In order to analyse the raw data, the videos of the particle movement were converted from ‘.mp4’ files to ‘stacks’ of ‘.tiff’ image files, with an aim to have one ‘.tiff’ file per frame of the video. Unfortunately, this proved extremely difficult due to the suspected variable fps of the camera (see limitations discussed in section 4), and multiple ‘.tiff’ files were obtained per frame. The reasoning behind suspecting a variable frame rate is discussed in section 6.4 in the Appendix, alongside efforts to solve the issue.

A choice was made between preserving the full data quality, but not the correct quantity (not removing any frames from the video) and having a closer approximation of the data quantity, but losing data quality (removing frames from the video by decreasing the images taken per second, which would not discriminate between removing duplicate images and unique images).

Although technically it would have been possible to perform this work manually and achieve both quality and quantity, such a task would have been infeasible, involving going through tens of thousands of image files, and deleting any duplicates. It was decided to work with the full data quality, and incorrect quantity, as overall less information of importance would be lost in the analysis (the velocities would be presented as a smaller magnitude than they should be).

Tracking of the raw data was done by making use of the freely available (fixed and updated) post-processing Matlab software, to identify and follow seeding particles using Lagrangian particle tracking. This was taken from Raghunandan et al. (2021) [6], which Kelley co-authored. Kelley and Ouelette’s (2011) original post-processing Matlab software [2] contained bugs, and did not allow the use of lossless, high quality ‘.tiff’ image files.

##### 3.1.2 Velocity Fields

The code (with steps) for getting the velocity field of the particles from the image files can be found in section 6.1 of the Appendix. An example velocity field is shown below in Figure 2:

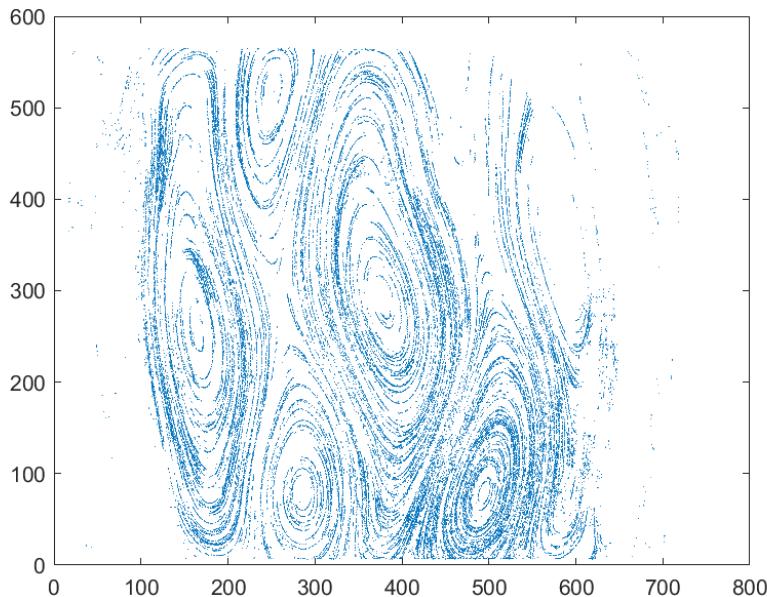


Figure 2: The measured velocity field for a current of  $20\text{ mA}$  in the fluid with 10% glycerol by volume.

### 3.1.3 Determining the vorticity field

Once the velocity field was found, the vorticity (twice the average angular velocity [7]) field could be calculated using spatial gradients of the measured velocity field.

By definition, if  $\mathbf{F} = \langle M, N \rangle$  then the two dimensional curl of  $\mathbf{F}$  is  $\text{curl } \mathbf{F} = N_x - M_y$ . [8]

The x- and y-components of the velocity field can be expressed in terms of their positions and time:

$$\mathbf{u} = [v_x, v_y] = [\mathbf{v}_x(x, y, t), \mathbf{v}_y(x, y, t)]$$

Thus the two-dimensional vorticity can be found by Equation 8: [9]

$$\omega = \nabla \times \mathbf{u} = \frac{\partial v_y}{\partial x} - \frac{\partial v_x}{\partial y} \quad (8)$$

As the velocity components gained from particle tracking were irregularly scattered, in order to get the spatial derivatives, the velocity components were interpolated on a uniform grid of query points. The result of this are shown in section 6.2.1, Figures 12 & 13.

The spatial gradients of the velocity components are then found by Matlab's `gradient()` function, which uses a central differences method, with backwards and forwards differences methods for the edges of the surface [10], and the vorticity can be plotted in accordance with Equation 8. An example vorticity plot is shown below in Figure 3 below:

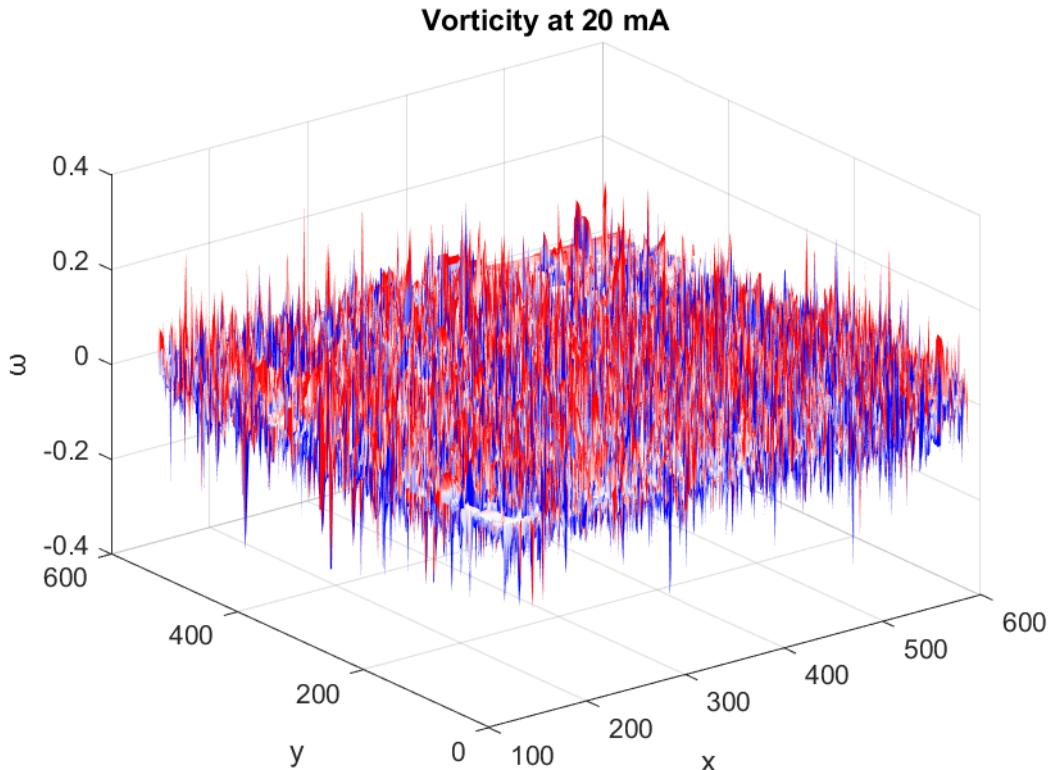


Figure 3: The vorticity plot (isometric view) for a current of 20 mA in the fluid with 10% glycerol by volume.

The vorticity plot (when viewed looking perpendicular to the x and y axes) is shown below in Figure 4, next to the same plot smoothed by a convolution with a 50x50 kernel. The vorticity typically has units of inverse time. However, since multiple image files were obtained per recorded frame, it is not possible to convert from the number of frames to seconds. As a result, the units of the vorticity field are 1/image rather than 1/s. The velocity field appears weaker, and thus the magnitude of the vorticity field also appears lower than what it actually is. See section 4 for further discussion.

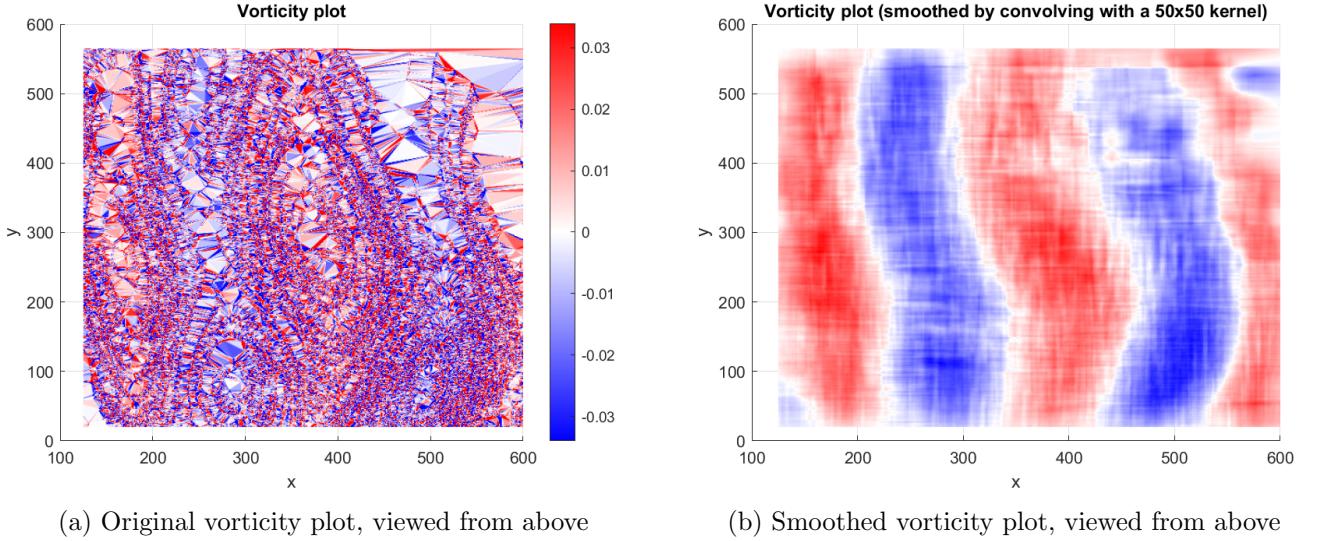


Figure 4: The vorticity plot for a current of 20 mA in the fluid with 10% glycerol by volume. Note that the original vorticity plot is full of sharp peaks and valleys, which are smoothed out via convolution.

The progression of the fluid's vorticity field as the current was increased is illustrated below in Figure 5:

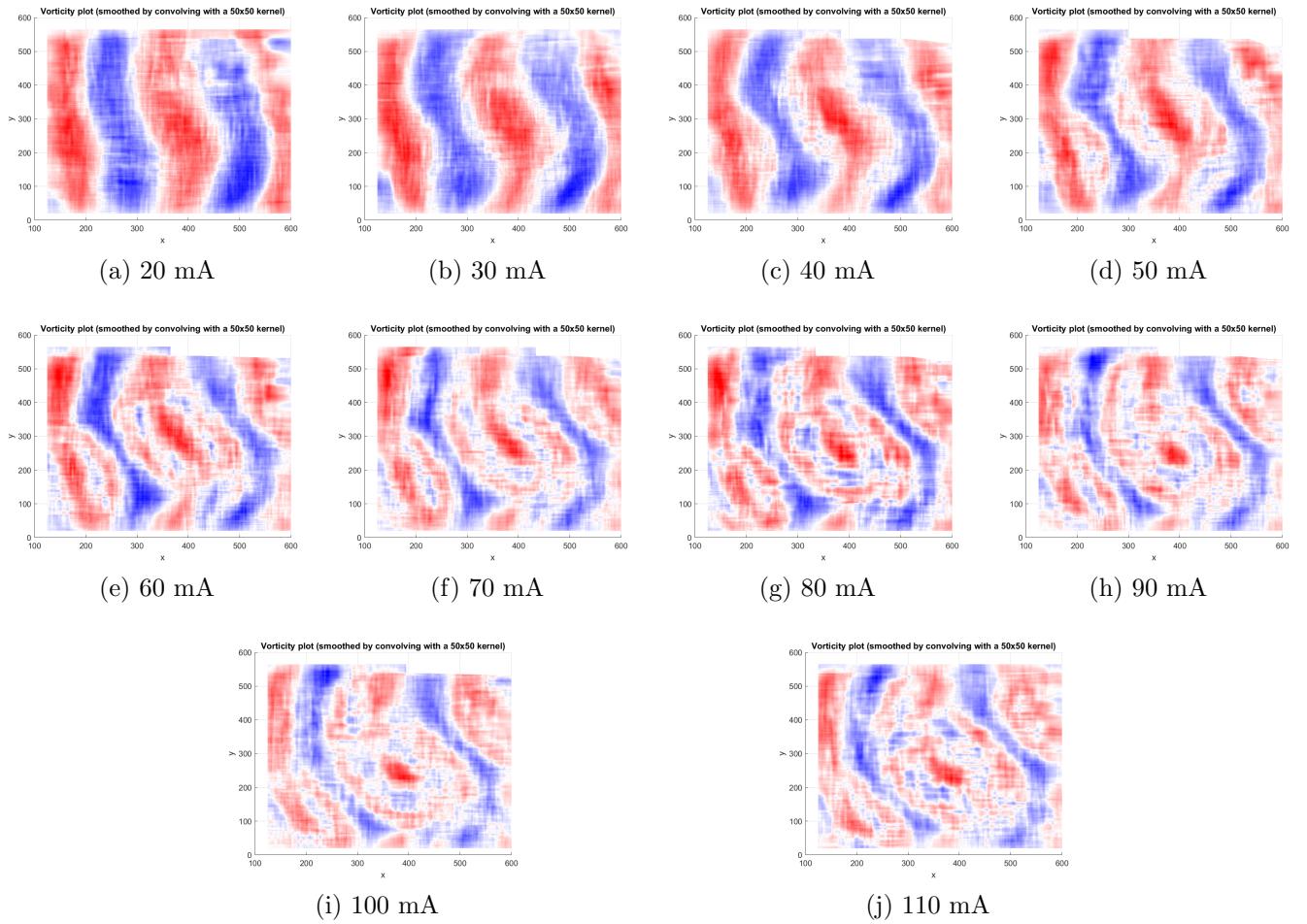


Figure 5: The progression of the vorticity field of the fluid flow as the current was increased from  $20\text{ mA}$  to  $110\text{ mA}$  (10% glycerol by volume).

The vorticity of the stable flow approximates Kolmogorov flow in the beginning, and is clearly shown to become less stable as the current (and thus the applied Lorentz force) is increased. This confirms that as the force of the perturbations increases, an instability occurs, in which the stripes are replaced by an array of steady vortices.

### 3.2 Investigation of the effect of viscosity on fluid instability - Extension to the experiment

By the same process as the first fluid (10% glycerol by volume), two more fluids were investigated. The progression of their vorticity fields as the current is increased is shown in Figures 6 and 8 below:

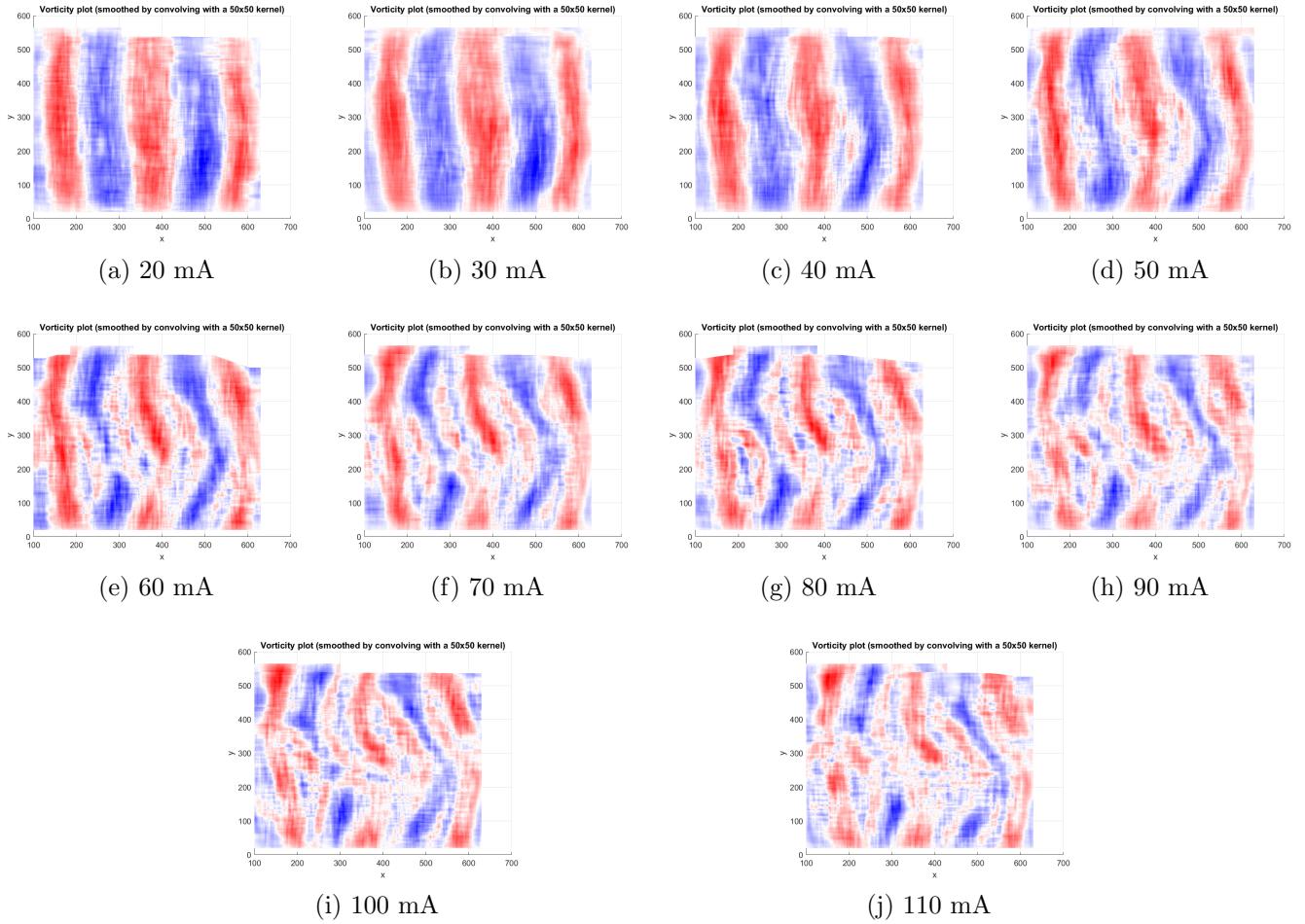
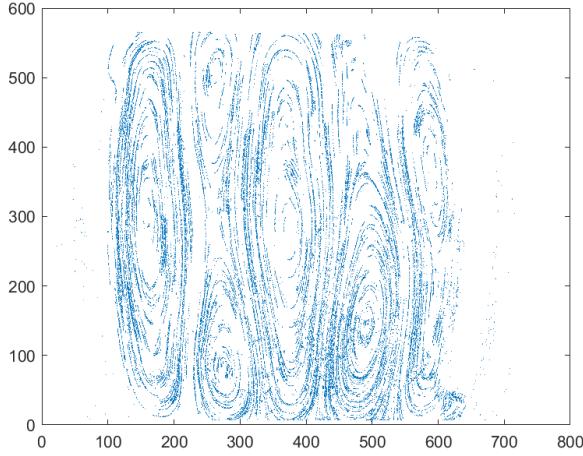
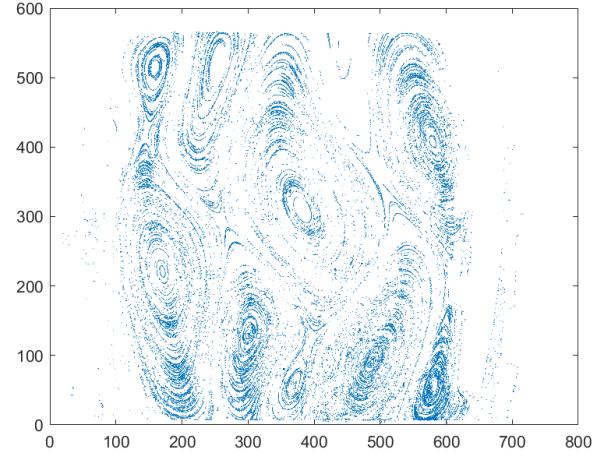


Figure 6: The progression of the vorticity field of the fluid flow as the current was increased from  $20\text{ mA}$  to  $110\text{ mA}$  (In the fluid with 20% glycerol by volume).

Similarly to the least viscous 10% glycerol by volume fluid, the 20% glycerol initially has a stable flow, with a pattern of stripes that was far more distinct, illustrating a good approximation of stable Kolmogorov flow. Whereas the 10% glycerol fluid was only stable at currents below  $40\text{ mA}$ , the 20% glycerol fluid remained stable for currents below  $60\text{ mA}$ . As the applied perturbations increased in force, an instability occurred and the pattern of stripes gave way to an array of steady vortices. This can also be seen in the velocity field for  $20\text{ mA}$  compared to that of  $110\text{ mA}$ , shown in Figure 7.



(a) Velocity plot at  $20\text{ mA}$



(b) Velocity plot at  $110\text{ mA}$

Figure 7: Comparing the velocity plots for different drive currents. (20% glycerol)

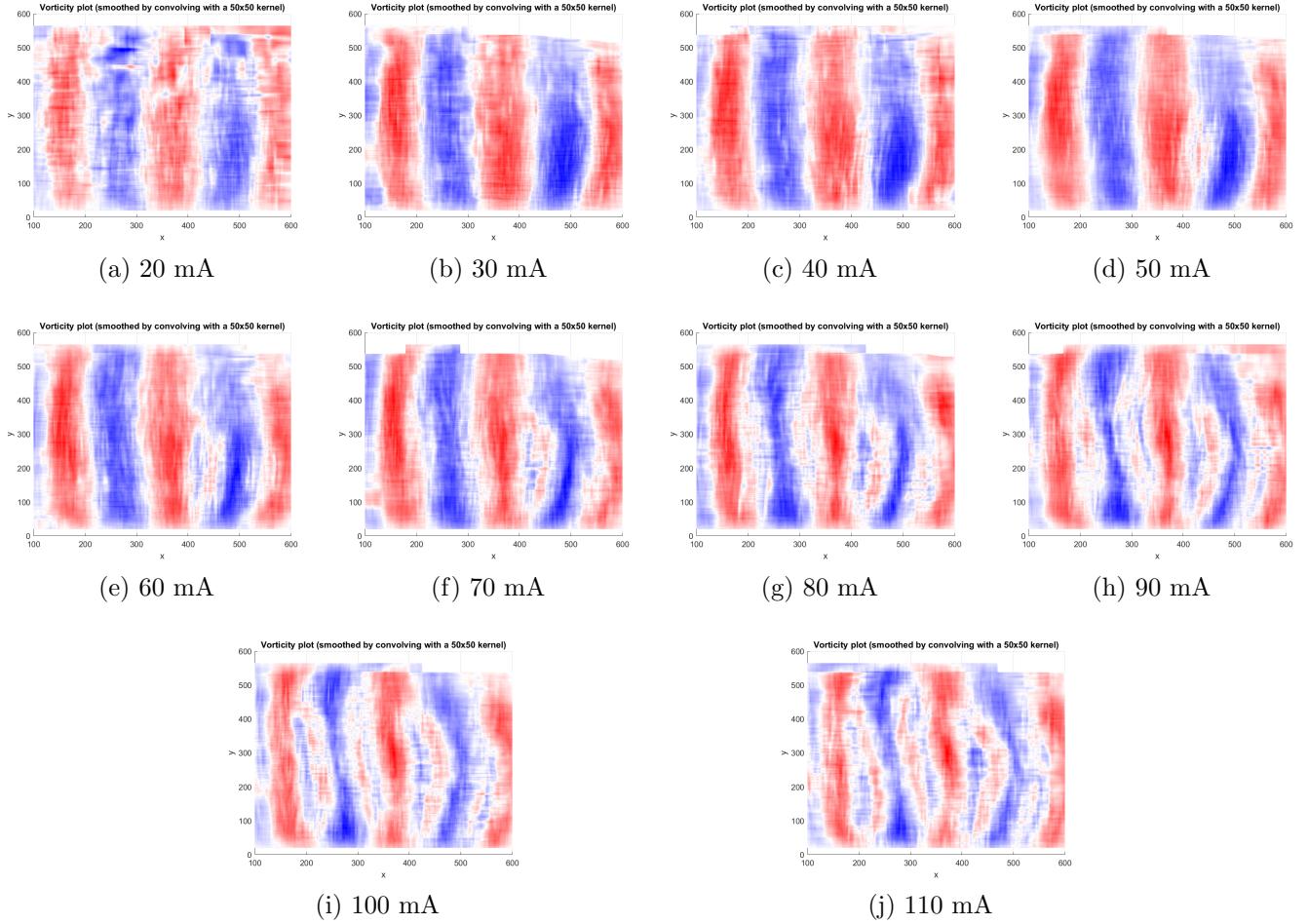


Figure 8: The progression of the vorticity field of the fluid flow as the current was increased from  $20\text{ mA}$  to  $110\text{ mA}$  (In the fluid with 30% glycerol by volume).

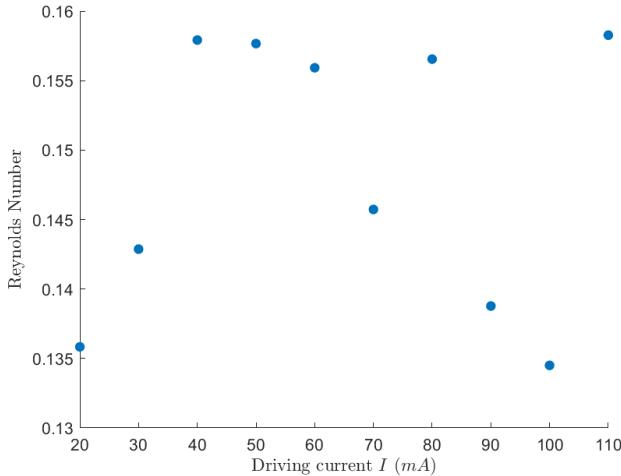
Likewise, the 30% glycerol fluid was more stable than the 20% glycerol, and remained stable for longer as the perturbations increased, only becoming unstable towards the highest currents. As the applied perturbations increased in force, an instability occurred and the pattern of stripes gave way to an array of steady vortices.

### 3.3 Investigation of the instability onset

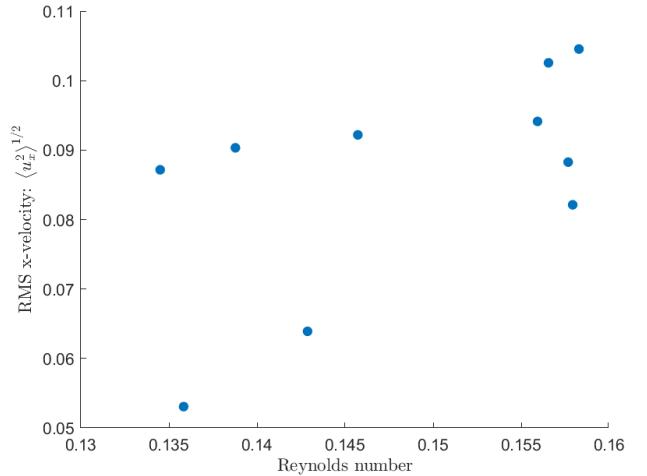
Having seen the instabilities occur, an attempt was made to plot the onset of the instability. The Reynolds number ( $Re$ ) is measured as the total Root-Mean-Square (RMS) velocity (see section 6.3 in the Appendix) and describes the relative importance of fluid inertia and viscous dampening. Small values of  $Re$  indicate that viscous effects can damp perturbations.

As the theoretical Kolmogorov flow contains no flow in the  $\hat{x}$  direction, the onset of instability was searched for using the RMS x-component of the velocity:  $\langle u_x^2 \rangle^{1/2}$  (see section 6.3 in the Appendix).

The Reynolds numbers were plotted against the drive currents, and the RMS  $u_x$  plotted against  $Re$  for the fluids in Figures 9, 10 and 11 below.

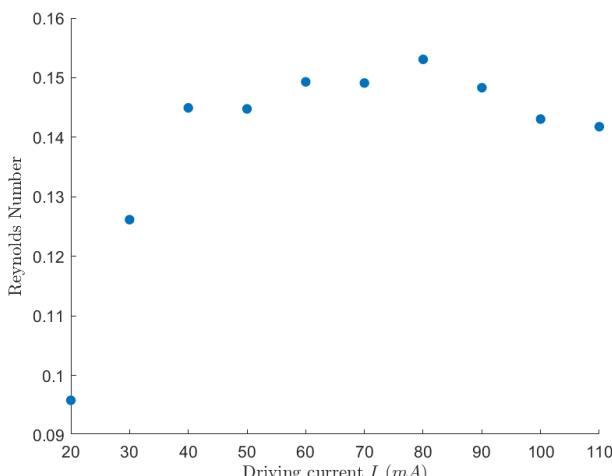


(a)  $Re$  is expected to be initially low, and increases sharply, then flattens out. The points at  $I=70,90,100$  mA do not follow this (expected) pattern.

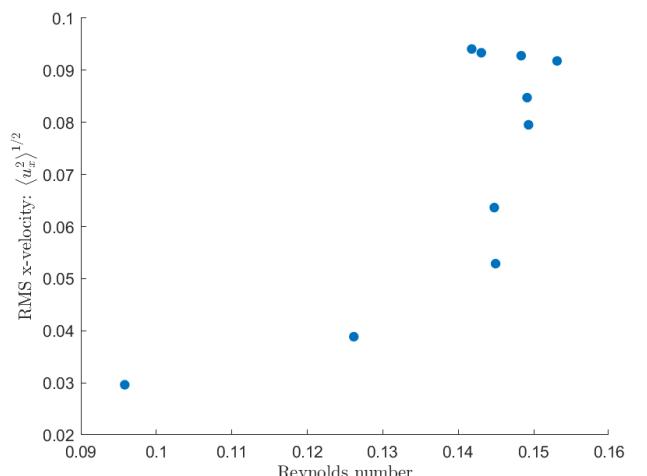


(b) Instability onset is seen as low flow speed, which increases sharply after a critical  $Re$  value. The values of  $Re$  associated with  $I=70,90,100$  mA are exceptions to this expected pattern.

Figure 9: The RMS  $u_x$  against Reynolds numbers for 10% glycerol fluid. The units of  $u_x$  are in pixels/image file. The plots barely follow the expected pattern.



(a)  $Re$  is initially low, and increases sharply, then flattens out.



(b) Instability onset is seen as low flow speed, which increases sharply after a critical  $Re$  value.

Figure 10: The RMS  $u_x$  against Reynolds numbers for 20% glycerol fluid. The units of  $u_x$  are in pixels/image file. The plots generally follow the expected pattern.

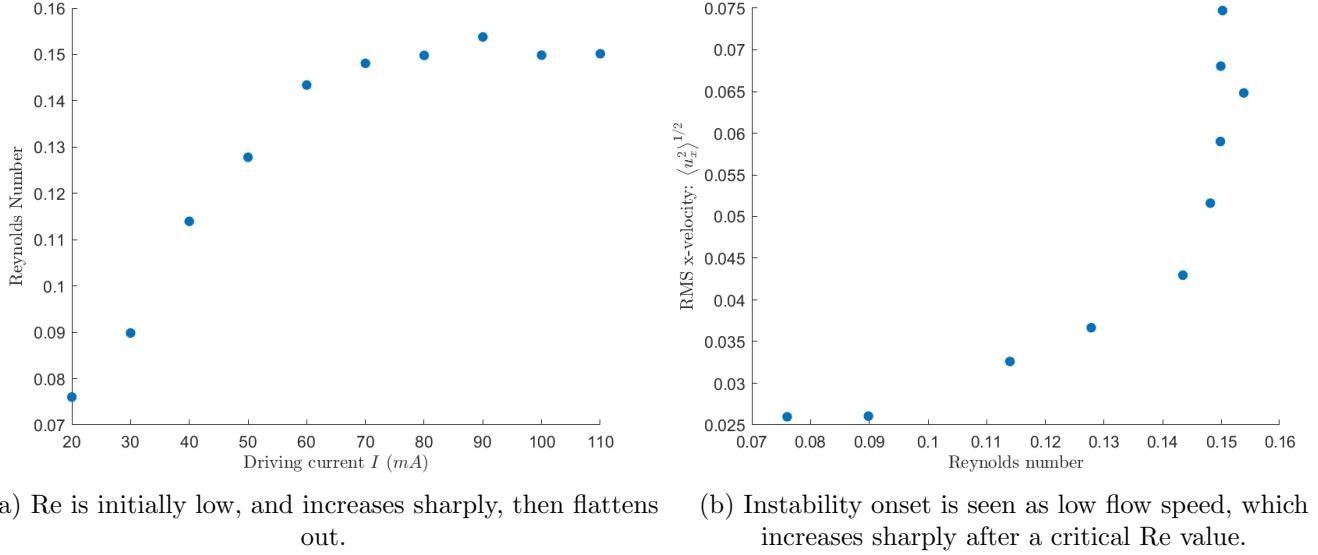


Figure 11: The RMS  $u_x$  against Reynolds numbers for 30% glycerol fluid. The units of  $u_x$  are in pixels/image file. The plots almost perfectly follow the expected pattern.

## 4 Discussion

For each fluid, it was observed that generally the fluid flow was stable for small drive currents (low force perturbations). The vorticity fields approximated Kolmogorov flow, with a stationary pattern of stripes. For greater drive currents (larger force perturbations), the stationary striped pattern bifurcated to a stationary lattice of vortices. This global transition in the flow is characteristic of an instability having occurred.

The differences in the viscosity of each fluid were found to have a significant effect on the spatial structure of the flow. The higher viscosity the fluid, the more stable it was, with a more distinct stationary pattern of stripes at low driving currents. This was as expected, as the more viscous the fluid, the greater the effect viscous damping can have on perturbations.

The parameters at which instability occurs were investigated by plotting the RMS horizontal velocity component against the Reynolds numbers. It was found that for the fluids with 20% and with 30% glycerol, the instability onset could be clearly seen, with an initially low horizontal flow speed, which increased sharply and steadily after a critical  $Re_c$  value. No attempt was made to determine the exact value of  $Re_c$  (for reasons discussed in the next paragraph), but upon visual inspection it seemed to be in the range of  $0.1 \rightarrow 0.11$  for the 30% glycerol fluid. For the 10% glycerol fluid, outliers were observed, potentially due to errors associated with the image files obtained from the associated videos.

This study did have its limitations. It is suspected that the camera used had a variable frame rate (section 6.4), which cause a host of issues. More than one image file was obtained per frame, meaning that many image files were duplicates. This resulted in it being impossible to convert velocities from pixels/image file into SI units, and so the vorticities and Reynolds numbers did not have the correct values. Because of this, no quantitative comparisons to the existing literature could be made, and attempts to determine the exact parameters at instability ( $Re_c$ ) were made futile. Days of laboratory time were lost due to the copper electrodes being inset vertically instead of horizontally.

## 5 Conclusions

The primary aim of this experiment was to use particle tracking methods to investigate a classical flow instability. The effect of the drive current was observed, and the spatial structure of the flow was measured. The effect of the fluid viscosity on the fluid instability was determined, and the onset of instability in the fluids was investigated.

For each fluid, it was observed that generally the fluid flow was stable for small drive currents (low force perturbations). For greater drive currents (larger force perturbations), a global transition in the flow was observed, characteristic of an instability having occurred.

The differences in the viscosity of each fluid were found to have a significant effect on the spatial structure of the flow. The higher viscosity the fluid, the lower the Reynolds numbers at given drive current, and the more stable it was.

It was found that for the fluids with 20% and with 30% glycerol, the instability onset could be clearly plotted, with an initially low horizontal flow speed, which increased sharply and steadily after a critical  $Re$  value ( $Re_c$ ). The instability onset was not as clear for the 10% glycerol fluid, despite appearing to be visually illustrated.

Possible further extensions to this research could involve investigating the effect of changing the  $\text{CuSO}_4$  percentage weight by mass in the fluid. Any future studies should take care to ensure that their camera records at a fixed framerate, and that their copper electrodes are inset horizontally.

## References

- [1] E. Fylladitakis. “Kolmogorov Flow: Seven Decades of History”. In: 6 (2018), pp. 2227–2263. DOI: [10.4236/jamp.2018.611187](https://doi.org/10.4236/jamp.2018.611187).
- [2] Douglas H. Kelley and Nicholas T. Ouellette. “Using particle tracking to measure flow instabilities in an undergraduate laboratory experiment”. In: *American Journal of Physics* 79.3 (2011), pp. 267–273. DOI: [10.1119/1.3536647](https://doi.org/10.1119/1.3536647).
- [3] MIT OpenCourseware, *Marine Hydrodynamics*. URL: <https://ocw.mit.edu/courses/mechanical-engineering/2-20-marine-hydrodynamics-13-021-spring-2005/lecture-notes/lecture9.pdf>.
- [4] A. M. Obukhov. “Kolmogorov flow and laboratory simulation of it.” In: 38 (1983), pp. 113–126.
- [5] Dantec Dynamics - Seeding Particles. URL: <https://www.dantecdynamics.com/components/seeding-materials/>.
- [6] Aditya Raghunandan et al. “Bulk flow of cerebrospinal fluid observed in periarterial spaces is not an artifact of injection”. In: *eLife* 10 (Mar. 2021). Ed. by Ronald L Calabrese, Claire Wyart, and Per K Eide, e65958. ISSN: 2050-084X. DOI: [10.7554/eLife.65958](https://doi.org/10.7554/eLife.65958). URL: [https://gitlab-public.circ.rochester.edu/araghuna/bulk-flow-is-not-an-artifact\\_raghunandan\\_et\\_al\\_2021](https://gitlab-public.circ.rochester.edu/araghuna/bulk-flow-is-not-an-artifact_raghunandan_et_al_2021).
- [7] Roger Stevens. *University of Leeds, Fluid Dynamics 1 Lecture Notes*. URL: [http://www1.maths.leeds.ac.uk/~kersale/2620/Notes/chapter\\_3.pdf](http://www1.maths.leeds.ac.uk/~kersale/2620/Notes/chapter_3.pdf).
- [8] Prof. Denis Auroux. *MIT OpenCourseware*. URL: [https://ocw.mit.edu/courses/mathematics/18-02sc-multivariable-calculus-fall-2010/3.-double-integrals-and-line-integrals-in-the-plane/part-b-vector-fields-and-line-integrals/session-64-curl/MIT18\\_02SC\\_notes\\_33.pdf](https://ocw.mit.edu/courses/mathematics/18-02sc-multivariable-calculus-fall-2010/3.-double-integrals-and-line-integrals-in-the-plane/part-b-vector-fields-and-line-integrals/session-64-curl/MIT18_02SC_notes_33.pdf).
- [9] Stephen Childress. *An Introduction to Theoretical Fluid Mechanics*. American Mathematical Society, 2009. Chap. 3. ISBN: 978-1-4704-1762-8. DOI: <https://doi.org/10.1090/cln/019>.
- [10] Dmitri Pelinovsky. *Numerical Methods for Differential Equations, Lecture 3*. 2007. URL: <https://dmpeli.math.mcmaster.ca/Matlab/Math4Q3/NumMethods/Lecture3-1.html>.

# 6 Appendix

## 6.1 Finding the velocity field from the image files

**Note:** Functions taken from Raghunandan et al. are `BackgroundImage_open()` and `PredictiveTracker_open()`. These functions are too long to be reasonably included in this report, but can be found at the URL associated with their citation [6]. The usage and descriptions of the functions are provided.

### 6.1.1 Obtaining background image

The background image is created as an average of every image file from a single 15 second video of fluid flow, at a constant current (in this example code, 20 mA). It's required for the next step.

```
1 % Retrieve all the image files in a directory, and average over them to get an
  ↳ average background image
2 backgroundImage_open("C:\Users\Luke\Desktop\Advanced Laboratories I & II\Adv Labs
  ↳ 2\1 Flow Instabilities\tiff files\1010\20\*.tiff", '20avg.tiff');
```

#### Usage of function

Usage: `bg=BackgroundImage\_open(inputnames, [outputname])`

Given a movie, `BackgroundImage_open` calculates the mean pixel values over time, returning the result in "bg" and saving it as an image with the filename "outputname". The movie must be saved as a series of image files, an image stack in .tif or .gif format, or an uncompressed .avi file.

### 6.1.2 Obtaining the particle tracks

The `PredictiveTracker_open()` function was provided with the list of image files and the created background image file (to be 'subtracted' from each individual file). Parameters of interest were changed until an ideal setting was found: the brightness threshold by which a pixel of interest (tracking particle) differs from the background was set to 10, the function was set to seek single-pixel particles by comparing brightness to adjacent pixels, a track was broken when no particle lay within 1 pixel of the predicted location, and pixels of the predicted location

```
1 PredictiveTracker\_open()\_open("C:\Users\Luke\Desktop\Advanced Laboratories I &
  ↳ II\Adv Labs 2\1 Flow Instabilities\tiff files\1010\20\*.tiff", 10,1 ...
2   ,           "C:\Users\Luke\Desktop\Advanced Laboratories I & II\Adv Labs 2\1
  ↳ Flow Instabilities\tiff files\1010\Averages\20avg.tiff",1,0)
3 save("1010_20.mat", 'ans')
4 clear("ans")
```

#### Usage of function

Usage: `[vtracks,ntracks,meanlength,rmslength] = PredictiveTracker_open( inputnames,`  
`threshold,max_disp,[bground_name],[minarea],[invert],[noisy])`

Given a movie of particle motions, `PredictiveTracker_open()` produces Lagrangian particle tracks using a predictive three-frame best-estimate algorithm. The movie must be saved as a series of image files, an image stack in .tif or .gif format, or an uncompressed .avi file; specify the movie in "inputnames" (e.g., '0\*.png' or 'stack.tif', or 'movie.avi'). To be identified as a particle, a part of the image must have brightness that differs from the background by at least "threshold". If

`invert==0`, `PredictiveTracker_open()` seeks particles brighter than the background; if `invert==1`, `PredictiveTracker_open()` seeks particles darker than the background; and if `invert==1`, `PredictiveTracker_open()` seeks any sort of contrast. The background is read from the file "bgroun\_name"; see `BackgroundImage`. If `minarea==1`, `PredictiveTracker_open()` seeks single-pixel particles by comparing brightness to adjacent pixels (fast and good for small particles); otherwise `PredictiveTracker_open()` seeks particles having areas larger than "minarea" (in square pixels; this method is better for tracking large particles). Once identified, each particle is tracked using a kinematic prediction, and a track is broken when no particle lies within "max\_disp" pixels of the predicted location. The results are returned in the structure "vtracks", whose fields "len", "X", "Y", "T", "U", and "V" contain the length, horizontal coordinates, vertical coordinates, times, horizontal velocities, and vertical velocities of each track, respectively. If `minarea =1`, "vtracks" is returned with an additional field, "Theta", giving the orientation of the major axis of the particle with respect to the x-axis, in radians. The total number of tracks is returned as "ntracks"; the mean and root-mean-square track lengths are returned in "meanlength" and "rmslength", respectively. If `noisy =0`, the movie is repeated with overlaid velocity quivers and the tracks are plotted. If `noisy==2`, each movie frame is also saved to disk as an image.

### 6.1.3 Plotting the velocity field

A function was written to plot the velocity field, given the particle tracks output.

```

1  function [] = Velocity_Field()
2  %VELOCITY_FIELD
3  for i = 20:10:110
4      clearvars -except i
5      disp(i)
6      load(['1010_',num2str(i),'.mat'])
7      %structs:
8      % Note to self:
9      %If any value input is an empty cell array, {}, then the output is an empty
10     %→ structure array.
11     % %To specify a single empty field, use [].
12     x=[ans.X];
13     y=[ans.Y];
14     u=[ans.U]; %horizontal vel.
15     v=[ans.V]; %vertical vel.
16     quiver(x,y,u,v) %vector plot
17     saveas(gcf,['V_F_',num2str(i),'.png'])
18 end
19 clear all
20 close all
21 end

```

## 6.2 Finding the vorticity field given the velocity field

### 6.2.1 Velocity components interpolated over a regular grid

In order to find the vorticity field, the velocity components are interpolated over a uniform grid of query points.

```

1  clearvars
2  i=20;

```

```

3 load(['1010_','num2str(i),' .mat'])
4
5 x = [ans.X];
6 y = [ans.Y];
7 u = [ans.U]; %horizontal vel.
8 v = [ans.V]; %vertical vel.
9
10 strvar = 'linear' %The type of interpolation done - found to be best after
    %extensive testing
11
12 [x_grid,y_grid] = meshgrid(xi,yi);
13 u_gd = griddata(x,y,u,x_grid,y_grid,strvar);
14 v_gd = griddata(x,y,v,x_grid,y_grid,strvar);
15
16 %Plotting the velocity interpolations onto a regular grid:
17     %Horizontal velocity:
18 figure
19 mesh(x_grid,y_grid,u_gd)
20 colormap(jet)
21 camlight right
22 lighting phong
23 title(['u\gd (method == ',strvar,',) on peaks surface'])
24 xlabel('x')
25 ylabel('y')
26 zlabel('u')
27     %Vertical Velocity
28 figure
29 mesh(x_grid,y_grid,v_gd)
30 colormap("jet")
31 camlight right
32 lighting phong
33 title(['v\gd (method == ',strvar,',) on peaks surface'])
34 xlabel('x')
35 ylabel('y')
36 zlabel('v')

```

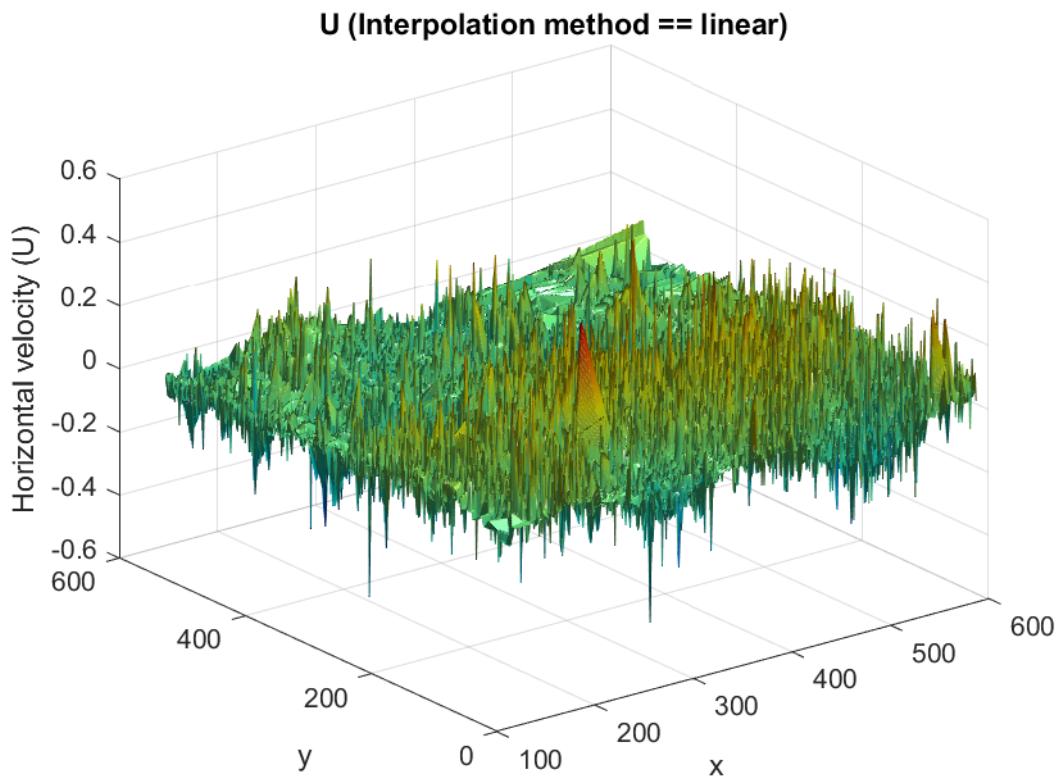


Figure 12: The horizontal velocity component interpolated over a regular grid for a current of  $20\text{ mA}$  (for fluid with 10% glycerol by volume)

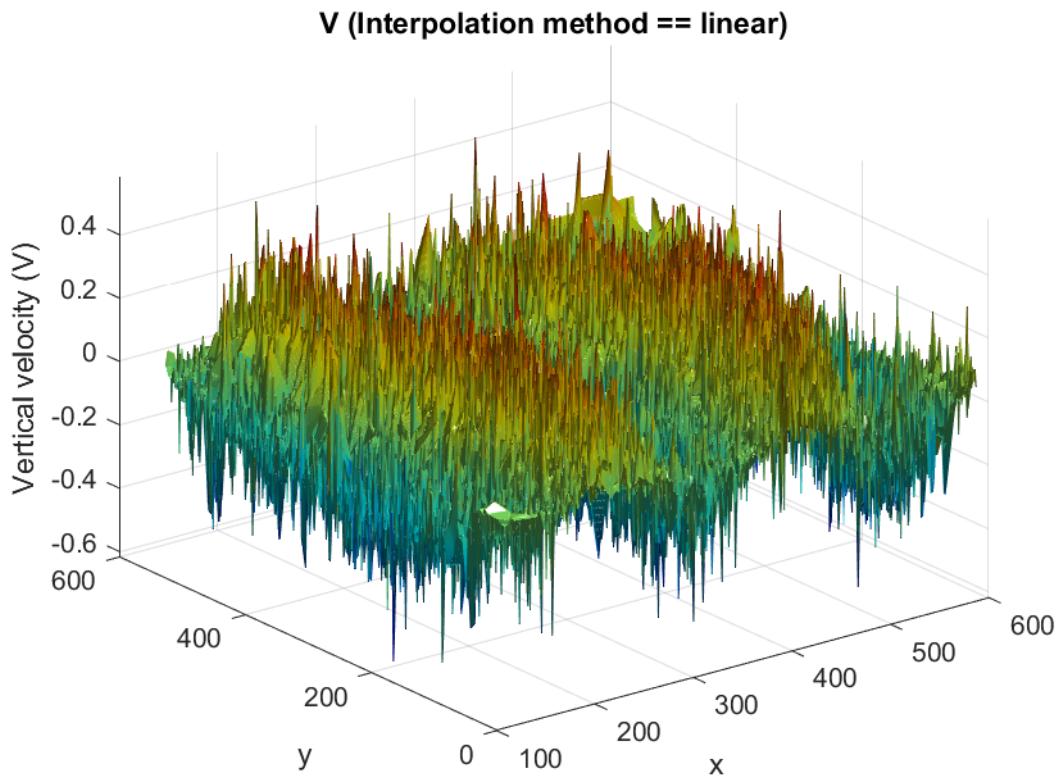


Figure 13: The vertical velocity component interpolated over a regular grid for a current of  $20\text{ mA}$  (for fluid with 10% glycerol by volume)

### 6.2.2 Using the spatial gradients of the velocity components to plot the vorticity

A function to plot the vorticity, as well as plotting the vorticity smoothed by convolving the data with different sized kernels (3x3, 5x5, 10x10, 50x50). Larger kernels result in smoother output.

```

1  function [] = Vorticity_Field()
2  %VORTICITY_FIELD
3  strvar = 'linear';
4  map=redblue(256);
5  for i = 20:10:110
6      clearvars -except i map strvar
7      disp(i)
8      load(['1010_',num2str(i),'.mat'])
9
10     x = [ans.X];
11     y = [ans.Y];
12     u = [ans.U];%horizontal vel.
13     v = [ans.V];%vertical vel.
14
15     %%{
16     %1010-specific parameters (chosen based on velocity fields):
17     xi = linspace(125,600,476); %x-range over which we interpolate the
18     %→ velocity-values to a regular grid
19     yi = linspace(20,565,546); %y-range over which we interpolate the
20     %→ velocity-values to a regular grid
21     %}
22     %{
23     %1020:
24     xi = linspace(100,630,531);
25     yi = linspace(20,565,546);
26     %}
27     %{
28     %1030:
29     xi = linspace(100,600,501);
30     yi = linspace(20,565,546);
31     %}
32
33     %Interpolating the velocity components over regular grids
34     [x_grid,y_grid] = meshgrid(xi,yi);
35     u_gd = griddata(x,y,u,x_grid,y_grid,strvar);
36     v_gd = griddata(x,y,v,x_grid,y_grid,strvar);
37
38     %The spatial gradients of the velocity components:
39     %gradient() only works as long as the spacing in each direction is 1,
40     %otherwise would have to input spacing as an argument
41     [dvdx, dvdy] = gradient(v_gd);
42     [dudx, dudy] = gradient(u_gd);
43     vort = dvdx - dudy; %vorticity is the curl of the velocity field
44
45     %Plotting the vorticity (without smoothing)
46     %%{
47     figure;
48     colormap(map)

```

```

47 camlight right;
48 surf(x_grid,y_grid,vort,'EdgeColor','None');
49 lighting phong;
50 lim=caxis*0.1;
51 lim=(abs(lim(1))+abs(lim(2)))/2;
52 caxis([-lim, lim]);
53 title(['Vorticity (' ,strvar,' velocity field interpolation')']);
54 xlabel('x');
55 ylabel('y');
56 view(2)
57 saveas(gcf,['Vort_',num2str(i),'_1010.png'])
58 close all
59 %}

60
61 %Smoothing with convolution to smooth 2D data that contains high
62 %frequency components:
63 for kvar = [3,5,10,50] %It was decided that 50x50 was the best smoothing
64 %kernel
65 K = (1/(kvar^2))*ones(kvar);
66 Zsmooth1 = conv2(vort,K,'same');
67 %Plotting
68 figure;
69 colormap("redblue")
70 camlight right;
71 surf(x_grid,y_grid,Zsmooth1,'EdgeColor','None')
72 lighting phong;
73 %keep colour axis centered on zero (white)
74 lim=caxis;
75 lim=(abs(lim(1))+abs(lim(2)))/2;
76 caxis([-lim, lim]);
77 title(['Vorticity plot (smoothed by convolving with a
78 % ',num2str(kvar), 'x',num2str(kvar), ' kernel)']);
79 xlabel('x')
80 ylabel('y')
81 view(2)
82 colorbar
83 saveas(gcf,['Vort_',num2str(i),'_1010_Kern',num2str(kvar), '.png'])
84 close all
85 end
86 end
87 end

```

### 6.3 Investigating the root mean squared $u_x$ velocity component

```

1 clearvars
2 strvar = 'linear';
3 rmsu = zeros(1,10);
4 rmsvel = zeros(1,10);
5 for i = 20:10:110
6     clearvars -except i rmsu strvar rmsvel

```

```

7 %disp(i)
8 load(['1010_',num2str(i),'.mat'])
9
10 x = [ans.X];
11 y = [ans.Y];
12 u = [ans.U];%horizontal vel.
13 v = [ans.V];%vertical vel.
14
15 %%{
16 %1010-specific parameters (chosen based on velocity fields):
17 xi = linspace(125,600,476); %x-range over which we interpolate the
18 %→ velocity-values to a regular grid
19 yi = linspace(20,565,546); %y-range over which we interpolate the
20 %→ velocity-values to a regular grid
21 %}
22 %{
23 %1020:
24 xi = linspace(100,630,531);
25 yi = linspace(20,565,546);
26 %}
27 %{
28 %1030:
29 xi = linspace(100,600,501);
30 yi = linspace(20,565,546);
31 %}
32
33 %Interpolating the velocity components over regular grids
34 [x_grid,y_grid] = meshgrid(xi,yi);
35 u_gd = griddata(x,y,u,x_grid,y_grid,strvar);
36 v_gd = griddata(x,y,v,x_grid,y_grid,strvar);
37
38 vsqr=v_gd.^2;
39 usqr=u_gd.^2;
40 %total velocity:
41 velocity_magnitude = sqrt(vsqr+usqr);
42 velocity_mag_sqr = velocity_magnitude.^2;
43 rms_velocity = sqrt(nanmean(velocity_mag_sqr(:)));
44 rmsvel(i/10 - 1) = rms_velocity;
45
46 %horizontal
47 mean_usqr = nanmean(usqr(:));
48 sqrt_mean_usqr = sqrt(mean_usqr);
49 rmsu(i/10 - 1) = sqrt_mean_usqr;
50
51 end
52 scatter([20,30,40,50,60,70,80,90,100,110],rmsvel,"filled")
53 xlabel("Reynolds Number",'Interpreter','latex');
54 ylabel('Driving current $I$ ($mA$)', 'Interpreter', 'latex');
55 saveas(gcf,'re_vs_I_1010.png')
56
57 scatter(rmsvel,rmsu,"filled")

```

```

56 ylabel("RMS x-velocity: $\left\langle u_x \right\rangle^2 \right\rangle^{1 / 2}$", 'Interpreter', 'latex');
57 xlabel('Reynolds number', 'Interpreter', 'latex');
58 saveas(gcf, 'ux_vs_Re_1010.png')
59 close all

```

## 6.4 Investigating the FPS of the camera, and attempts made to obtain only one image per frame

The 15-second videos were saved as .mp4 format by the camera. When converted to .tiff files using an online converter (default, unchangeable 60fps), multiple .tiff files per frame were discovered (No on-hand application could view .tiff files, so this was only discovered after converting them to .jpeg to take a look).

This issue warranted a closer inspection, and looking at the properties of the .mp4 files showed that the videos had an fps of 24.85. After trying many unsuccessful methods to obtain image files from .mp4 files while also being able to specify the fps, an open-source software project called FFMPEG was discovered, which utilised a command-line tool to process video and audio files. After learning how to use FFMPEG on the command-line, the following code was used to loop over my video files and obtain .tiff images at a rate of 24.85 per second in-video:

```

1 for \%i in (*.mp4) do
  ↳ "C:\Users\Luke\Downloads\ffmpeg-4.4-essentials_build\ffmpeg-4.4-essentials_build\
  ↳ bin\ffmpeg.exe" -i "%i" -r 24.85 "C:\Users\Luke\Desktop\Advanced
  ↳ Laboratories I & II\Adv Labs 2\1 Flow
  ↳ Instabilities\newtiff\1010\%~ni\%04d.tiff"

```

Unfortunately, although the math was done to make sure exactly 24.85 images per second were obtained, duplicate images were still being obtained, but only for some files. Other files would have three unique images one after the other. Extensive research and testing provided no results for methods to delete duplicate images with different file names, so no solution was found to fix the issue. It was considered as a possibility that the camera may just have a lower framerate, but this would not adequately explain the many duplicates in a row, followed by multiple unique images. In addition, were lower framerates to be manually tested (putting aside the length of time that this would take), the risk would be run of unintentionally cutting out unique frames - an arguably bigger problem.