SIGNATURE PAGE

This thesis for honors recognition has been approved for the

Department of Mathematics.

Director — Date: 2-15-16

Reader — Date: 2/15/16

Reader — Date: 2/15/16

Development of Physically Realistic Numerical Model for Blinking Vortex System

Dawson Osborn

February 15, 2016

Abstract

The blinking vortex system defines a method of mixing where two vortices turn on and off alternately. Research suggests that systems of this nature are sensitive to the initial conditions. This thesis aims to build off the authors previous paper, "Experimental and Numerical Analysis of Chaotic Advection in 2-D Blinking Vortex Flow System", to create a more accurate numerical model of a blinking vortex system [9]. The model previously created in the PHYS 452: Advanced Physics Lab course did not account for multiple important physical characteristics of the fluid system, which resulted in the model being unusable to predict physical behavior. Therefore, a new set of differential equations, based on the Navier-Stokes equations is used to model the blinking vortex system.

Using the experimental results obtained in PHYS 452, the improved numerical model is compared to physical results. Using the positive Lyapunov exponent as a point of comparison, it is revealed that the improved numerical model is very accurate in predicting physical behavior. By accounting for the fluid's momentum and viscosity with the Navier-Stokes equations, a viable numerical model for the blinking vortex system was created.

Contents

List of Figures

List of Tables

1 Acknowledgements

I would like to thank my thesis director, Dr. Eric Sullivan. I am not alone when I say that his patience and enthusiasm for mathematics is an inspiration. I would to thank Dr. Kelly Cline for his assistance in numerical modeling; without his instruction in the Numerical Methods and Computational Physics course, I would not have the knowledge to complete this thesis. I would like to thank Dr. Mary Keeffe for her assistance in completing the experiment in Advanced Physics Lab. I would be remiss if I did not mention my lab partners Austin Powell and Mark Griffith for their contributions to completing the blinking vortex lab. I would like to thank the Mathematics and Physics faculty that instructed during my time at Carroll College: Dr. Holly Zullo, Dr. Tim Melvin, Dr. Phil Rose, Dr. Ted Wendt, Dr. Anthony Szpilka, and Mr. Pat Judge.

Finally, I would like to thank my family for their constant love and support. I am truly blessed.

2 Introduction

The blinking vortex system, developed by Hassan Aref, exhibits behavior that is classified as chaotic [1]. This system, in the most basic form, consists of two vortices; the vortices alternate periodically, meaning that only one vortex is active at any given time. This alternation, or blinking, creates sensitivity to initial conditions where sections of the fluid move throughout the system very differently, which results in the system mixing.

2.1 Foundations and Terminology

The following concepts and terms are provided to assist with understanding the nature of the blinking vortex system and the accompanying experiments and numerical models.

2.1.1 Tracers

In order to observe both experimental and numerical systems, individual particles are often observed to understand the behavior of the system as a whole. Experimentally, small, light polystyrene spheres can act as tracers, as they float and follow the flow of the system. Numerically, tracers are defined as points, which are acted upon by the numerically calculated velocity fields. For the purposes of this thesis, a tracer refers to an indicator of the system, which can be experimentally observed or numerically modeled. For more information regarding the computer program that tracks tracers in the experimental environment, see 2.1.8.

2.1.2 Chaos and Chaotic Advection

The term chaos, in the English language, is defined as, “A state of utter confusion or disorder” [5]. The term chaos, in fluid dynamics, is understood to define a system

that is sensitive to initial conditions. Strogatz defined chaos as, "Aperiodic long-term behavior in a deterministic system that exhibits sensitive dependence on initial conditions" [11]. Wiggins and Ottino developed a more pointed definition for chaos: "The orbits have some positive Lyapunov exponents" [12]. The term, orbits, refers to the path that a particle follows as a system develops. In other words, the orbit defines where the particle has been and where it will go in the future [12].

Certain issues arise in dealing with systems that are chaotic. How do you quantify this arguably qualitative term [12]. Given the scope of this thesis, understanding chaotic to mean that there is a positive Lyapunov exponent associated with the system suffices. For more information regarding the positive Lyapunov exponent, see 2.1.6.

2.1.3 Electrode

The term electrode is frequently used in this thesis in reference to the origin of the mixing. For the blinking vortex experimental system, two electrodes produce mixing via the Lorentz force. In other physical systems, the forcing apparatus could be a paddle wheel or other common mixing apparatus. See figure 2 for location of electrodes in experimental system.

2.1.4 Blinking

Blinking refers to the behavior of the electrodes within the blinking vortex system. As only one electrode is active at any given moment, the electrodes activate alternatively back and forth, or blink.

2.1.5 Efficient Mixing

For the purposes of the original paper by the author, the topic of efficient mixing was paramount [9]. The question of what defined efficient mixing remains relevant

to this thesis. For the purpose of the original paper, efficient mixing was defined by the area that a single tracer would cover over in a certain time period; this area was then divided by the total area of the system, which resulted in a proportion. This proportion is known as the Area Sweep Proportion, or ASP. The calculation of the ASP takes into account two factors that define efficient mixing: velocity of the tracer and change in distance from the electrode [9]. For more information regarding the area calculation, see 3.2.1 and A.

2.1.6 Lyapunov Exponent

Strogatz states that, in a n-dimensional system, there are n different Lyapunov exponents [11]. This thesis aims to gain a better understanding of the 2-dimensional blinking vortex system, which means that there are two different Lyapunov exponents: a positive and a negative exponent. If two tracers were placed close to each other in a system, the distance between the tracers grows at an approximately exponential rate. The exponential separation is defined by equation (1).

$$\Delta r(t) = \Delta r_0 e^{\lambda t} \tag{1}$$

where Δr_0 is the initial distance between the tracers and λ is the positive Lyapunov exponent [6]. This positive Lyapunov exponent can be determined by analyzing the plot of the natural log of the distance separating the tracers versus time. The value of the positive Lyapunov exponent can be found by examining the slope of a semi-logarithmic plot of the distance between tracers over time. An example of a semi-logarithmic such plot can be found in figure 1.

As evidenced by figure 1, the plot is not a smooth, straight line; however, the positive Lyapunov exponent can be approximated through a linear regression. The calculation of linear regressions is accomplished through either MATLAB or Excel.

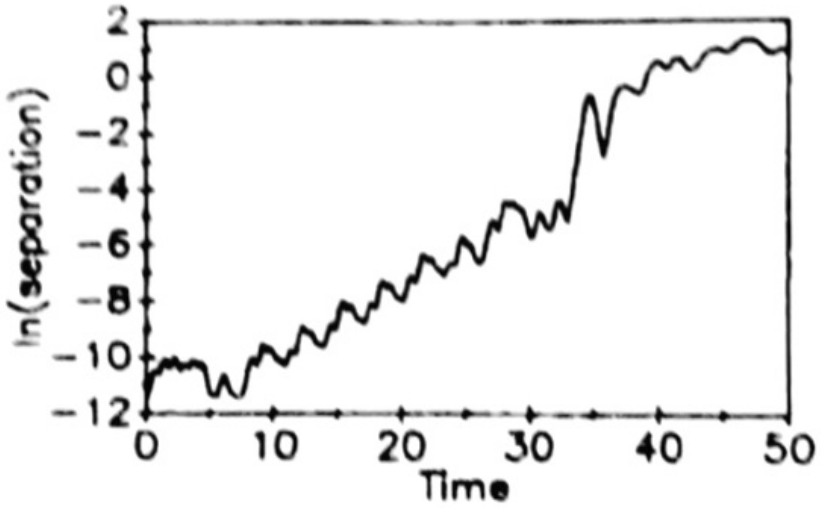

Figure 1: Example of semi-logarithmic plot of the separation between two tracers moving in a blinking vortex system, [6]

The positive Lyapunov exponent allows for quantitative description of chaos in the blinking vortex system. This thesis will not attempt to determine levels of chaos; rather, experimental results will reveal the presence of a positive Lyapunov exponent. If a positive Lyapunov exponent is observed, the exponent will assist in determining if numerical models accurately predict the behavior of the blinking vortex system.

2.1.7 LabVIEW and DAQ

LabVIEW is a computer software that allows hardware integration. This software allows the user to manipulate instruments such as voltage and current sources, digital oscilloscopes, and other technical instruments. Using this software, code was developed in the PHYS 452 course that allowed electrical parameters the blinking vortex system to be user-defined.

DAQ is short for data acquisition; a DAQ refers to a device that is able to communicate with the LabVIEW code. The usage of the DAQ, paired with the previously described LabVIEW code, created a current source and function generator that al-

lowed for variation of the parameters. In studying the variation of the parameters, along with the use of the previously mentioned area proportion calculation, the parameters corresponding to optimal mixing efficiency were determined in the PHYS 452 course. For more information regarding the results of the variation in parameters, see 3.3.

2.1.8 Tracker

Tracker is a computer software that allows the user to upload video files and track the movement of points within the video environment. Using the timing information provided by the video file, the software is able to calculate positions, velocities, and accelerations of points. For the purpose of the observing the blinking vortex system, tracers were placed in the system. The movement of these tracers was mapped out by Tracker, which allowed for the calculation of the previously mentioned area proportion and the positive Lyapunov exponent.

2.1.9 MATLAB

MATLAB is a coding environment that allows for matrix-based calculations. This software is advantageous in simulating physical systems, such as the blinking vortex system. This software was used to create the author's original numerical model and the new numerical model presented in this thesis [9].

2.2 Purpose

As a result of the semester-long PHYS 452: Advanced Physics Lab course, Arefs blinking vortex was experimentally observed and numerically modeled. An accompanying paper created by the author, entitled, Experimental and Numerical Analysis of Chaotic Advection in 2-D Blinking Vortex Flow System, summarized the results of this experimentation and modeling [9]. Following a brief presentation of the results

of the course, several questions arose regarding the validity of the numerical model from the Mathematics and Physics faculty of Carroll College. The course was at its conclusion, which did not allow for correction of the numerical model. This thesis will revisit the results found in the Spring 2015 PHYS 452 course, address the issues found in the authors, Experimental and Numerical Analysis of Chaotic Advection in 2-D Blinking Vortex Flow System, explore a new numerical model of the blinking vortex, and compare the new numerical model to the original model.

3 Experimental Results

This section will both explain how the blinking vortex system was created experimentally and give a brief overview of the results found in the PHYS 452 course.

3.1 Experimental System

The system is contained by a petri dish; this petri dish contains a wire ring that runs around the bottom of the dish. Two electrodes enter below the dish, through holes. A strong ($\sim$1 Tesla) neodymium magnet is placed under the petri dish, creating a magnetic field that runs approximately perpendicular to the bottom surface of the petri dish. Figure 3 shows the magnetic field lines in the context of the system. A salt water solution is placed into the petri dish; the added salt allows current to flow through the solution. To create the vortices at the electrodes, an electromagnetic force is created by allowing current to flow from the wire ring to the active electrode. This resulting force, defined by the Lorentz force, causes the solution to mix. Figure 2 shows the experimental set-up with the aforementioned components labeled.

To create the blinking vortex, a DAQ along with LabVIEW code created a blinking system that could be manipulated. The code allowed for the current flowing and the period of blinking to be user defined. The wiring schematic of this blinking system is

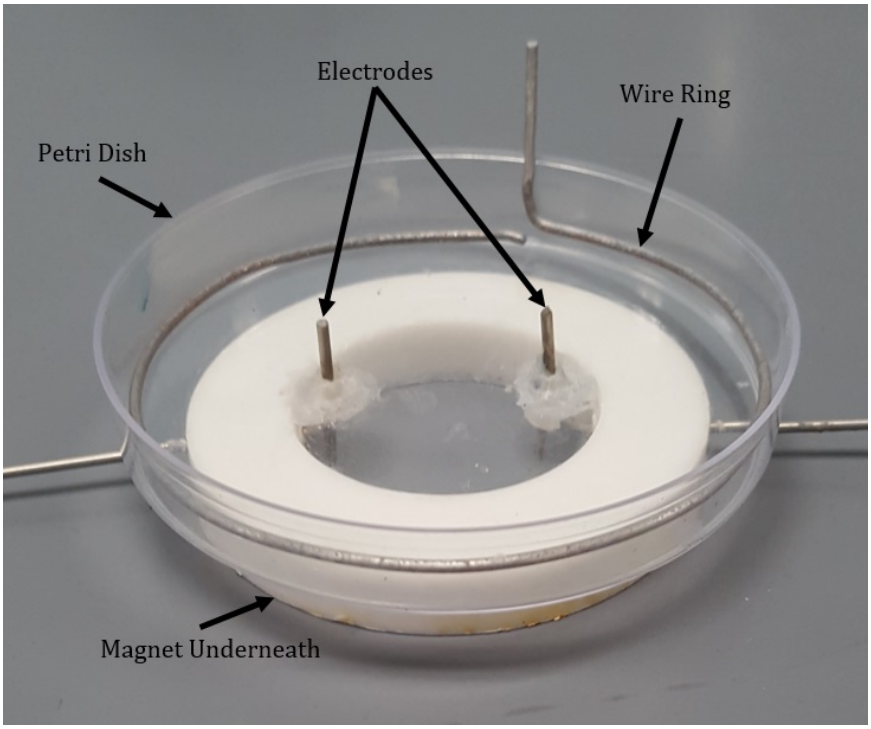

Figure 2: Photograph of blinking vortex system experimental set-up, magnet not shown in photograph

displayed in figure 4.

To observe the behavior of the system, tracers, or small, light polystyrene spheres, were placed in the solution. A video camera was placed above the blinking vortex system, creating an overhead view of the evolution of the system. Resulting videos were then exported to the Tracker program, where the behavior of the tracers could be quantitatively defined.

3.2 Experimental Analysis

3.2.1 Single Tracer

Using the previously described area sweep proportion, the effectiveness of the mixing given the parameters was quantified. We determined the efficiency of the mixing by examining the area that a tracer covered in the first 45 seconds of experimentation.

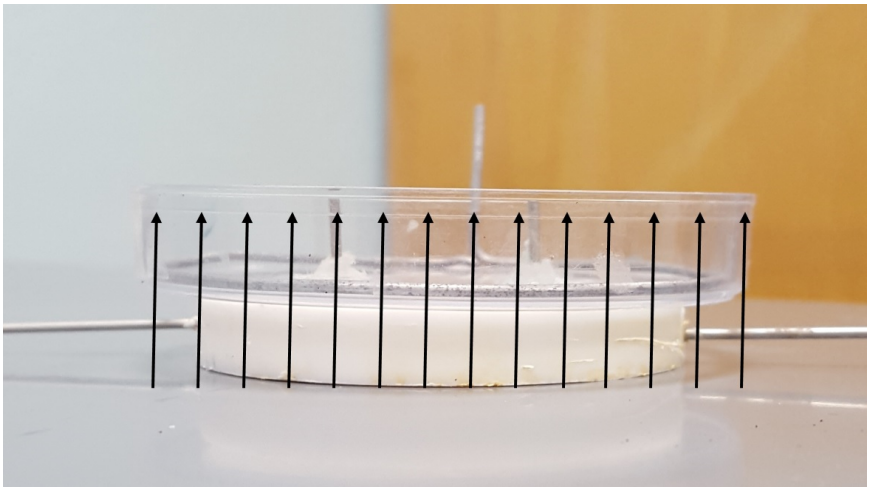

Figure 3: Visualization of magnetic field lines in blinking vortex system experimental set-up, line direction can be reversed by changing magnet's orientation

The tracers path was converted into x and y coordinates using Tracker. This data was then transferred to Area Sweep code (see A), which highlights points under a defined radius from the tracer over the entire 45 second interval. The highlighted area is then divided by the total area to give the proportion, the ASP. The interpretation, of the author, of the ASP was that larger proportions corresponded to more efficient mixing [9]. This proportion allowed for the comparison of results where certain parameters of the system were varied. This procedure was used for the analysis of all experiments that involved single tracers.

3.2.2 Two Tracer and Positive Lyapunov Exponent

In the cases where two tracers were placed in the system to determine the positive Lyapunov exponent, a slightly different procedure was followed. The implications and calculation of this positive Lyapunov exponent will be discussed in depth in several following sections. As previously stated, Tracker allows the x and y position of the tracers to be calculated frame by frame. Accurate values of position are obtained by placing a reference, in this case a ruler, within the frame of the video and defining that

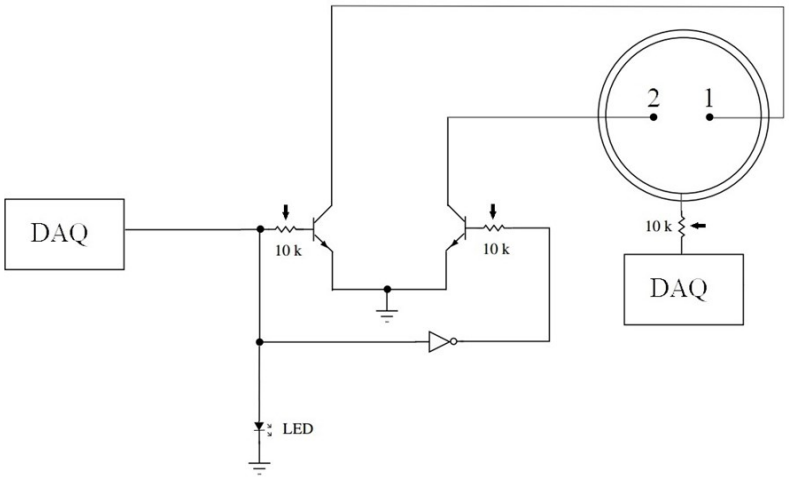

Figure 4: Circuit schematic for experimental setup; DAQ serves as constant current source and function generator for blinking

distance within Tracker. After the location of each tracer has been defined at every frame, the time and position data are exported to Excel, where the distance between the tracers can be calculated and in turn, the positive Lyapunov exponent. More specifically, once the distance between tracers has been calculated, the natural log of that distance is found. By combining the time information from Tracker with the natural log of the separation between tracers, semi-logarithmic plots of the distance are created. A linear regression is applied to this plot, which gives the value of the positive Lyapunov exponent. An example of the resulting plot from an experimental trial is shown in figure 5; this plot has been truncated to the first 20 seconds of experimentation. The coefficient of the regression, 0.1396 Hz, corresponds to the positive Lyapunov exponent.

3.3 Results of Varying Parameters

The results of the parameter variation will briefly be given. For more detailed explanation of the following results, see the author's (2015) Experimental and Numerical Analysis of Chaotic Advection in 2-D Blinking Vortex Flow System.

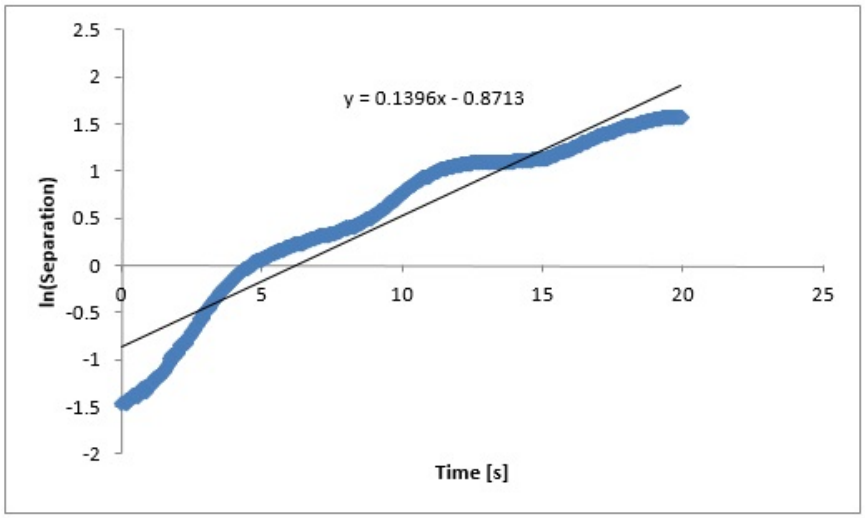

Figure 5: Semi-logarithmic plot displaying separation, in centimeters, of tracers over time, from experimental results, coefficient of regression represents positive Lyapunov exponent

3.3.1 Varying Salt Concentration in Solution

By varying the amount of concentration of salt in the solution, it was determined that increasing the concentration results in an increase in current flowing between the wire ring and the electrodes, up to a certain point. Once the solution becomes saturated with salt, the current able to flow through the system is maximized.

3.3.2 Varying Current Supplied by DAQ

By varying the amount of current supplied by the DAQ, it was determined that increasing the current results in more efficient mixing as defined by the ASP. This confirms the view that larger fluid velocities result in more efficient mixing, as the greater force, created by the greater current, creates a higher ASP.

3.3.3 Varying Period of Blinking

By varying the period of blinking, it was determined that mixing was optimized by a period of blinking between approximately 16 and 25 seconds, meaning each electrode

is active for approximately 8 to 12.5 seconds.

3.4 Observed Positive Lyapunov Exponents

For five trials, the behavior of two tracers placed in the blinking vortex system were observed. Using the previously defined procedure for determining the positive Lyapunov exponent. In the authors original work, Experimental and Numerical Analysis of Chaotic Advection in 2-D Blinking Vortex Flow System, the semi-logarithmic plots were truncated in a manner that only took into account selected sections of the data [9]. This subjective selection of the positive Lyapunov exponent for each trial undoubtedly added a bias to the results. The results of the experimentation, with the subjective truncation, are found in table 1.

Trial	Positive Lyapunov Exponent (Hz)
1	0.130
2	0.165
3	0.140
4	0.132
5	0.062
Average	0.126

Table 1: Table of biased positive Lyapunov exponent results

To correct for this subjective bias, the first 25 seconds of each trial was examined. No truncation was performed and the resulting positive Lyapunov exponent was strictly based on the regression of the first 25 seconds. The time period of 25 seconds was chosen based on Strogatz comments that, "we can't predict longer than a few multiples of $\frac{1}{\lambda}$" [11]. Table 2 displays the corrected values of the positive Lyapunov exponent for the same five trials. The corrected, non-biased results for the positive Lyapunov exponent will be used to analyze the validity of the numerical models.

Trial	Positive Lyapunov Exponent (Hz)
1	0.1230
2	0.0098
3	0.1041
4	0.0882
5	0.0622
Average	0.0775

Table 2: Table of unbiased positive Lyapunov exponent results

4 Original Numerical Model

4.1 Development of Model

The MATLAB code created in the PHYS 452 course is a top-down simulation (i.e. as if the code takes the place of the camera positioned above the experimental system) of the two-dimensional position of the two tracers. Along with the initial positions of the tracers, several parameters can be modified to mimic an experimental system. The simulation then solves for the tracers positions over time with a Runga-Kutta 4 numerical solution using equations (2) and (3). These differential equations were defined by Nugent, Quarles, and Solomon [8].

$$\frac{dx}{dt} = -\frac{A * y}{(x_s)^2 + y^2} \tag{2}$$

$$\frac{dy}{dt} = -\frac{A * x_s}{(x_s)^2 + y^2} \tag{3}$$

The results of the numerical calculations give the tracers positions, which are plotted in real time. See figure 6 for example of code output. The red line represents the orbit of a tracer.

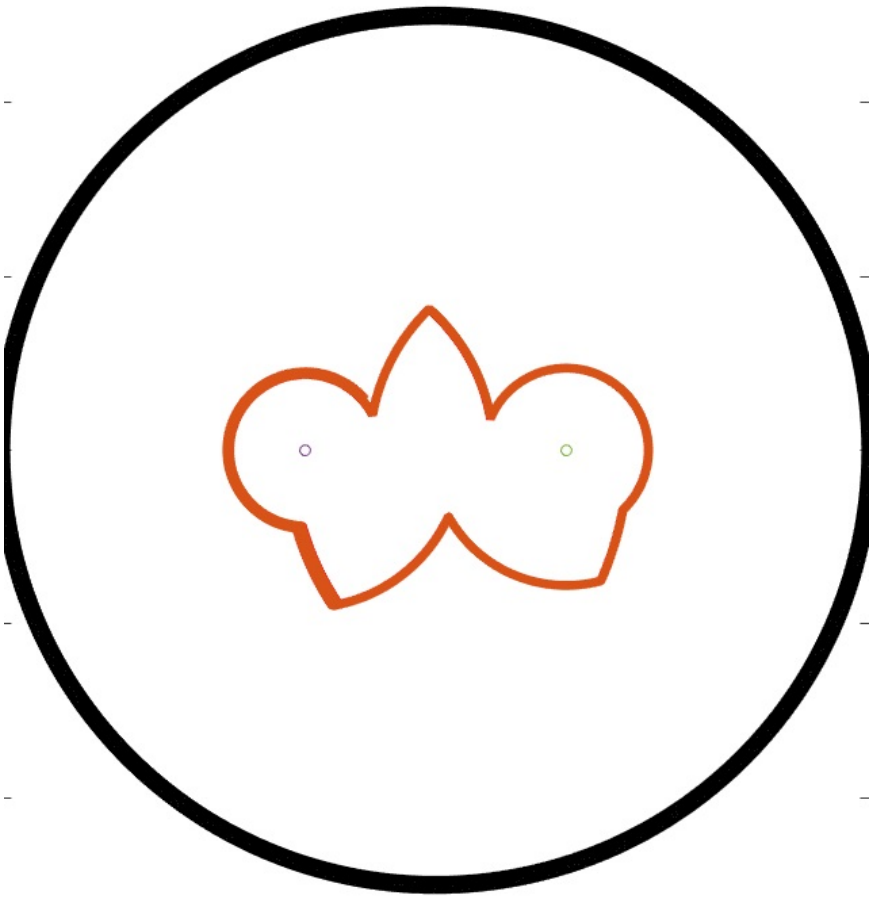

Figure 6: Output of Blink code, red orbits represent path of tracer, note "broken-ness" of orbits, sharp changes in direction are result of blinking

4.1.1 User-Defined Parameters

The Blink code requires an initial position, both x and y, for both tracers to be modeled. There are also four parameters which can be adjusted to change the behavior of the code. These parameters are:

- Period of blinking—This parameter corresponds to the duration of a blink. In other words, this is how it takes for the system to have both electrodes alternate.
- Amplitude—This parameter can be modified to mimic an experimental system. It is the combination of the force created through the current through the solution, the strength of the magnetic field, and the viscosity of the solution.

- Separation between the electrodes—This parameter corresponds to half the distance, in centimeters, between the two electrodes. Adjusting this distance will change the radius of rotation; thus, the overall path of the tracer.

- Number of time steps—This parameter corresponds to the length of time we want the simulation to run.

The code itself takes into account the initial conditions along with the parameters defined by the user and repeatedly performs calculations based on equations (2) and (3). Using these calculations, the code finds the new position one time step further down the line until the tracers have been defined for the user-defined time period.

4.1.2 Outputs

Once the code has calculated the position at all times, it moves on to visually displaying the tracers. The program plots the positions of both tracers on two side-by-side plots. Thus, we can see the movement of the tracers, in real time, as they are acted upon by the changing velocity field. We can also use the code to display the entire paths of the two tracers. However, the basic behavior of the numerical model is very comparable to the experimentally observed results.

4.1.3 Calculation of Numerical Positive Lyapunov Exponent

By slightly modifying the code previously developed for numerically modeling the behavior of two tracers, the positive Lyapunov exponent was able to be numerically calculated. The full modified code can be found in Appendix C; however, a brief description will suffice to explain how these modifications calculate the positive Lyapunov exponent. The modifications allow for the maximum and average positive Lyapunov exponent to be calculated:

- The addition of an additional *for* loop to numerically simulate multiple trial

results in a user-defined number of trials. By conducting multiple trials, random variation between simulations is accounted for.

- The addition of initial condition restrictions caused the two tracers to always start very close to each other, regardless of their random initial location. This means that no matter where the first tracer was randomly located by the code, the second tracer would always be placed very close initially.

- The addition of distance calculations results in the values of distance between the two tracers to be defined at every time step. This is the most important addition to the code, as it allows the positive Lyapunov exponent calculations to take place.

- The addition of linear regression calculations results in the calculation of the positive Lyapunov exponent itself. The linear regression is calculated individually for each trial, as well as for the average of every trial. This allows for discovery of the maximum positive Lyapunov exponent along with the average positive Lyapunov exponent of all the trials.

With these four main additions to the existing Blink code along with a few other minor changes, parameters that mimicked the experimental set-up were defined and the positive Lyapunov exponent was calculated. The experimental parameters mimicked included the following:

- 3 cm separation between the electrodes

- Strong magnet under system

- 10 mA of current flowing from wire ring to active electrodes

- Period of blinking of 10 seconds

- Data collection for 50 seconds

4.2 Results

100 trials were numerically simulated; the average positive Lyapunov exponent was calculated to be 0.043 Hz. A hypothesis test is conducted. The null hypothesis is that there is no difference between the experimentally observed positive Lyapunov exponent and the numerically calculated positive Lyapunov exponent; the alternative hypothesis is that the experimentally observed positive Lyapunov exponent is greater than the numerically calculated positive Lyapunov exponent. Equations (4) and (5) give the hypotheses.

$$H_0 : \lambda_{experimental} = \lambda_{numerical} \tag{4}$$

$$H_1 : \lambda_{experimental} > \lambda_{numerical} \tag{5}$$

Using a difference in means t-test, a p-value of 0.026 is found, which is significant at the α=0.05 level. This p-value is based on the following values:

- $\lambda_{experimental} = 0.0775\,\text{Hz}$
- $\sigma_{\bar{\lambda_e}} = 0.0439\,\text{Hz}$
- $\lambda_{numerical} = 0.0430\,\text{Hz}$
- $\sigma_{\bar{\lambda_n}} = 0.0391\,\text{Hz}$

Thus, we have evidence that the experimental positive Lyapunov exponent is greater than the numerical positive Lyapunov exponent predicted by this model.

4.3 Issues with Original Numerical Model

The differential equations (2) and (3) that went into the numerical code were not sufficient to describe the motion of a tracer or the velocity field of the fluid, as they did not account for fluid momentum or viscosity of the solution. The way that the

fluid moves in not broken as seen in 6, meaning that the velocity field experimentally does not behave in the manner defined by Equations 2 and 3. The issues arising from this numerical model indicate that further modeling with new differential equations is necessary to predict the positive Lyapunov exponent accurately.

5 Navier-Stokes Model

5.1 Development of Model

To better model the positive Lyapunov exponent, a change in the differential equations to model the velocity field was necessary. The Navier-Stokes equations were utilized to create a more realistic model of the blinking vortex system, as their use in fluid dynamics is extensive. Equation (6), similar to the equation given by Burger, defines the convective term of the Navier-Stokes equations [4]. The function, $\vec{F}$ represents the forcing function, in this case the Lorentz force that acts on the fluid. The function, $\vec{u}$ represents the velocity function of the fluid at all points in the model. The parameter, ν, represents the kinetic viscosity of the fluid.

$$\partial_t \vec{u} + \left(\vec{u} \cdot \vec{\nabla}\right) \vec{u} = -\frac{1}{\rho}\vec{\nabla}\vec{p} + \nu\vec{\nabla}^2\vec{u} + \vec{F} \tag{6}$$

This equation can be further simplified by assuming that the fluid is incompressible. This assumption is valid, according to Panton, when the Mach number is low ($M \to 0$) [10]. The speed of sound in water, at room temperature, is 1,481 m/s. The highest fluid velocities found in experimental trials were of the 10^{-2} m/s magnitude. With this information, a Mach number of magnitude 10^{-6} defines the experimental blinking vortex system. It is reasonable to assume that this is an example of an incompressible fluid. With that information, equation (6) can be simplified to equation

(7).

$$\partial_t \vec{u} = \nu \vec{\nabla}^2 \vec{u} + \vec{F} - \left(\vec{u} \cdot \vec{\nabla}\right) \vec{u} \tag{7}$$

The convection term and diffusion term of equation (7), which are defined as follows:

- Convection Term: $\left(\vec{u} \cdot \vec{\nabla}\right) \vec{u}$
- Diffusion Term: $\nu \vec{\nabla}^2 \vec{u}$

These two terms allow for the inertia and interaction of the fluid to be properly modeled. This improvement from equations (2) and (3) allows for the blinking vortex system to be modeled in a more physically realistic manner.

Using an Euler time step and the centered finite difference approximations for the first and second derivatives, a numerical model of the velocity field is created. From this velocity field, initial conditions can be defined and the path of the tracers according to the evolution of the velocity field is calculated using bilinear interpolation. In a similar manner to the original numerical model, the distance between two tracers in calculated and a semi-logarithmic plot is created. A linear regression is used to define the numerical positive Lyapunov exponent. Figure 7 displays an example of semi-logarithmic plot for one of the numerically modeled experimental trials; this plot includes a linear regression. The coefficient of the linear regression corresponds to the positive Lyapunov exponent.The full code for the Navier-Stokes model can be found in C.

5.2 Results

Using the experimental initial conditions of the five trials, five numerical positive Lyapunov exponents were calculated. Table 3 shows the results for each trial.

A hypothesis test is conducted. The null hypothesis is that there is no difference

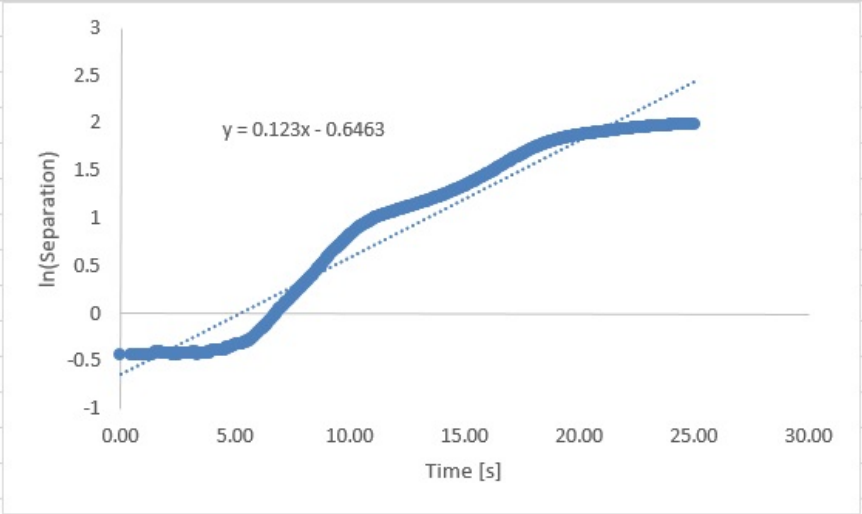

Figure 7: Semi-logarithmic plot produced by Navier-Stokes code calculations, separation in centimeters, coefficient of regression represents positive Lyapunov exponent

Trial	Positive Lyapunov Exponent (Hz)
1	0.1004
2	0.0593
3	0.1223
4	0.0459
5	0.0439
Average	0.0743

Table 3: Table of positive Lyapunov exponent results from Navier-Stokes numerical simulations

between the experimentally observed positive Lyapunov exponent and the numerically calculated positive Lyapunov exponent; the alternative hypothesis is that the experimentally observed positive Lyapunov exponent is greater than the numerically calculated positive Lyapunov exponent. Equations (8) and (9) give the hypotheses.

$$H_0 : \lambda_{experimental} = \lambda_{numerical} \tag{8}$$

$$H_1 : \lambda_{experimental} > \lambda_{numerical} \tag{9}$$

Using a difference in means t-test, a p-value of 0.451 is found, which is not significant at the α=0.05 level. This p-value is based on the following values:

- $\lambda_{experimental} = 0.0775\,\text{Hz}$
- $\sigma_{\bar{\lambda_e}} = 0.0439\,\text{Hz}$
- $\lambda_{numerical} = 0.0743\,\text{Hz}$
- $\sigma_{\bar{\lambda_n}} = 0.0352\,\text{Hz}$

Thus, have no evidence that the Navier-Stokes model does not accurately predict the positive Lyapunov exponent.

6 Conclusion

The usage of the Navier-Stokes equations modeled the blinking vortex system in a physically realistic manner. The equations account for the momentum and viscosity of the fluid. The original models differential equations, equations (2) and (3), did not allow for these important physical characteristics to be modeled. The assumption that the fluid is incompressible allows for significantly easier calculations, while keeping the model physically realistic.

The average positive Lyapunov exponent of the Navier-Stokes numerical model, 0.0743 Hz, is comparable to the experimentally observed positive Lyapunov exponent, 0.0775 Hz. The average positive Lyapunov exponent of the original numerical model, 0.0430 Hz, is different from the observed results.

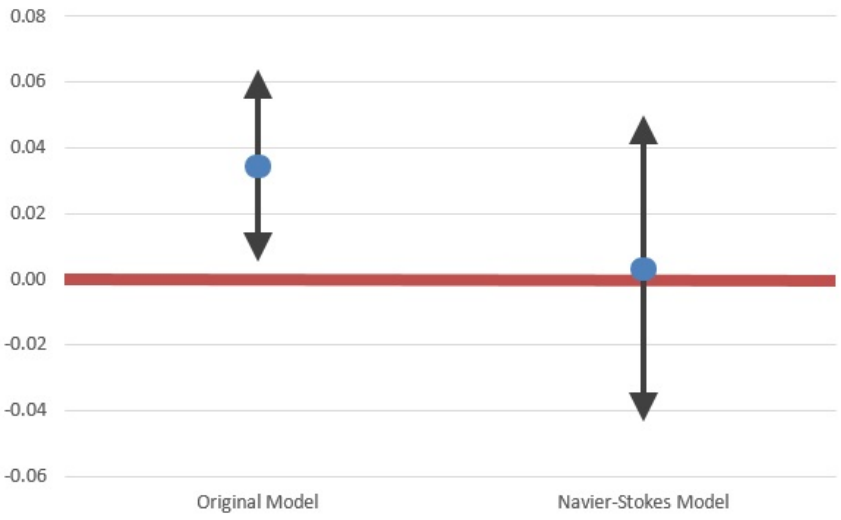

Figure 8: Intervals of the two difference of means t-tests calculated in 4.2 and 5.2, y-axis units are Hz

By applying statistical analysis, we find that these differences, or lack of differences, provide insight on the validity of each model; the statistical significance of the original model, at the α=0.05 level, indicates that this model is not physically realistic, where the lack of significance, at the α=0.05 level, indicates there is no

evidence the Navier-Stokes model is not physically realistic. See figure 8 for a visual representation of the difference of means t-test analysis; note that the interval of the original model does not include 0, which gives evidence that the model is significantly different from the experimentally observed positive Lyapunov exponent.

References

[1] Aref, H. (1984). Stirring by chaotic advection. *Journal of Fluid Mechanics*, 143, 1-21.

[2] Aref, H. (1990). Chaotic Advection of Fluid Particles. *Philosophical Transactions: Physical Sciences and Engineering*, 333(1631), 273-288.

[3] Aref, H. (2002). The development of chaotic advection. *Physics of Fluids*, 14(4), 1315-1325.

[4] Burger, M. (n.d.). Numerical Methods for Incompressible Flow. Retrieved January 30, 2016 from ftp://ftp.math.ucla.edu/pub/camreport/cam04-12.pdf

[5] Chaos. (n.d). Dictionary.com Unabridged. Retrieved January 30, 2016 from http://dictionary.reference.com/browse/chaos

[6] Chaotic Mixing and Hamiltonian Phase Space. (n.d). Retrieved January 30, 2016 from http://www.compadre.org/advlabs/bfy/files/chaotic_mixing1.pdf

[7] Karolyi, G. & Tel, T. (1997). Chaotic tracer scattering and fractal basin boundaries in a blinking vortex-sink system. *Physical Reports*, 290, 125-147.

[8] Nugent, C. R., Quarles, W. M., & Solomon, T. H. (1997). Experimental Studies of Pattern Formation in a Reaction-Advection-Diffusion System. *Physical Review Letters*, 93(21), 1-4.

[9] Osborn, D. (2015). Experimental and Numerical Analysis of Chaotic Advection in 2-D Blinking Vortex Flow System.

[10] Panton, R. L. (1984). *Incompressible Flow*. New York: John Wiley & Sons.

[11] Strogatz, S. H. (1994). *Nonlinear Dynamics and Chaos*. Cambridge, MA: Westview Press.

[12] Wiggins, S. & Ottino, J. M. (2004). Foundations of chaotic mixing. *The Royal Society*, 362, 937-970.

A Appendix A: Area Sweep Code

```
clc;
clear all;

trace = xlsread('Video30 Data.xlsx','Origin Top Left','B2:C1311');

dx = 0.01; % [cm]
dy = 0.01; % [cm]
pdd = 8.75; % petri dish diameter [cm]
pd = zeros(pdd/dx+1,pdd/dy+1);

tol = 0.2; % radius to sweep [cm]
tf = size(trace,1);

for n=1:tf
    for i=0:dx:pdd
        for j=0:dy:pdd
            if sqrt((trace(n,1)-i)^2+(trace(n,2)+j)^2) <= tol
                pd(round(i/dx+1),round(j/dy+1)) = 1;
            end
        end
    end
end

p_filled = sum(sum(pd))/((pdd/dx+1)^2)*4/pi % proportion of points filled

contourf(pd);
```

B Appendix B: Lyapunov Code

Simulation of a blinking vortex system

D. Osborn, Carroll College, March 17, 2015

```
clear all;
clc;
clf;

T=10; % Period of blinking
A=2*0.318; % Amplitude [m^2/s]
b=1.5; % 1/2 Seperation [cm]
cr=0.5; % Approximation of vortice region surrounding electrode
N=10000; % Number of time steps
trials=100; % Number of simulations to run
wormlength=100; % Number of points in worm
dt=0.005; % Time step [s]

time=0:dt:N*dt;
storage=zeros(N+1,trials);
lyapunov=zeros(trials,1);

figure('units','normalized','outerposition',[0 0 1 1]); % Opens full screen
for p=1:trials
rx=7*rand-3.5;
ry=7*rand-3.5;
    while or(sqrt((rx-b)^2+(ry)^2)<cr,sqrt((rx+b)^2+(ry)^2)<cr)
```

```
        rx=7*rand-3.5;
        ry=7*rand-3.5;
    end

qx=rx+0.005*rand-0.0025;
qy=ry+0.005*rand-0.0025;
distance=zeros(N,1); % Distance storage vector
distance(1)=log(sqrt((rx-qx)^2+(ry-qy)^2)); % Initial distance, ln

r=zeros(N,2); % Position storage matrix, 1st tracer
worm1=zeros(wormlength,2,N-wormlength); % Worm storage matrix
r(1,:)=[rx,ry]; % Initial tracer position, 1st tracer

q=zeros(N,2); % Position storage matrix, 2nd tracer
worm2=zeros(wormlength,2,N-wormlength); % Worm storage matrix
q(1,:)=[qx,qy]; % Initial tracer position, 2nd tracer

    for i=1:N
        t=(i-1)*dt; % time value
        x_sr=r(i,1)+b*sign(sin((2*pi/T)*t)); % blinking function, 1st tracer
        v_xr=(-A*r(i,2))/((x_sr)^2+(r(i,2))^2); % x-component of velocity, 1st tracer
        v_yr=(A*x_sr)/((x_sr)^2+(r(i,2))^2); % y-component of velocity, 1st tracer

        x_sq=q(i,1)+b*sign(sin((2*pi/T)*t)); % blinking function, 2nd tracer
        v_xq=(-A*r(i,2))/((x_sq)^2+(q(i,2))^2); % x-component of velocity, 2nd tracer
        v_yq=(A*x_sr)/((x_sq)^2+(q(i,2))^2); % y-component of velocity, 2nd tracer
```

```
%RK4
%k1
k_1xr=v_xr;
k_1yr=v_yr;

k_1xq=v_xq;
k_1yq=v_yq;
%k2
x_s2r=(r(i,1)+0.5*dt*k_1xr)+b*sign(sin((2*pi/T)*(t+dt/2)));
k_2xr=(-A*(r(i,2)+0.5*dt*k_1yr))/((x_s2r)^2+((r(i,2)+0.5*dt*k_1yr))^2);
k_2yr=(A*x_s2r)/((x_s2r)^2+((r(i,2)+0.5*dt*k_1yr))^2);

x_s2q=(q(i,1)+0.5*dt*k_1xq)+b*sign(sin((2*pi/T)*(t+dt/2)));
k_2xq=(-A*(q(i,2)+0.5*dt*k_1yq))/((x_s2q)^2+((q(i,2)+0.5*dt*k_1yq))^2);
k_2yq=(A*x_s2q)/((x_s2q)^2+((q(i,2)+0.5*dt*k_1yq))^2);
%k3
x_s3r=(r(i,1)+0.5*dt*k_2xr)+b*sign(sin((2*pi/T)*(t+dt/2)));
k_3xr=(-A*(r(i,2)+0.5*dt*k_2yr))/((x_s3r)^2+((r(i,2)+0.5*dt*k_2yr))^2);
k_3yr=(A*x_s3r)/((x_s3r)^2+((r(i,2)+0.5*dt*k_2yr))^2);

x_s3q=(q(i,1)+0.5*dt*k_2xq)+b*sign(sin((2*pi/T)*(t+dt/2)));
k_3xq=(-A*(q(i,2)+0.5*dt*k_2yq))/((x_s3q)^2+((q(i,2)+0.5*dt*k_2yq))^2);
k_3yq=(A*x_s3q)/((x_s3q)^2+((q(i,2)+0.5*dt*k_2yq))^2);
%k4
x_s4r=(r(i,1)+dt*k_3xr)+b*sign(sin((2*pi/T)*(t+dt)));
k_4xr=(-A*(r(i,2)+dt*k_3yr))/((x_s4r)^2+((r(i,2)+dt*k_3yr))^2);
k_4yr=(A*x_s4r)/((x_s4r)^2+((r(i,2)+dt*k_3yr))^2);
```

```
        x_s4q=(q(i,1)+dt*k_3xq)+b*sign(sin((2*pi/T)*(t+dt)));
        k_4xq=(-A*(q(i,2)+dt*k_3yq))/((x_s4q)^2+((q(i,2)+dt*k_3yq))^2);
        k_4yq=(A*x_s4q)/((x_s4q)^2+((q(i,2)+dt*k_3yq))^2);
        % New values of x and y
        r(i+1,1)=r(i,1)+(dt/6)*(k_1xr+2*k_2xr+2*k_3xr+k_4xr);
        r(i+1,2)=r(i,2)+(dt/6)*(k_1yr+2*k_2yr+2*k_3yr+k_4yr);

        q(i+1,1)=q(i,1)+(dt/6)*(k_1xq+2*k_2xq+2*k_3xq+k_4xq);
        q(i+1,2)=q(i,2)+(dt/6)*(k_1yq+2*k_2yq+2*k_3yq+k_4yq);

        distance(i+1)=log(sqrt((r(i+1,1)-q(i+1,1)).^2+(r(i+1,2)-q(i+1,2)).^2));

            if i>=wormlength % Saves worm to worm storage matrix
            worm1(:,1,i-(wormlength-1))=r(i-(wormlength-2):i+1,1); % Worm x-values
            worm1(:,2,i-(wormlength-1))=r(i-(wormlength-2):i+1,2); % Worm y-values

            worm2(:,1,i-(wormlength-1))=q(i-(wormlength-2):i+1,1); % Worm x-values
            worm2(:,2,i-(wormlength-1))=q(i-(wormlength-2):i+1,2); % Worm y-values
            end

    end

fit=polyfit(transpose(time),distance,1);
lyapunov(p)=fit(1);
storage(:,p)=distance(:);
subplot(1,2,1);
```

```
plot(r(:,1),r(:,2),'LineWidth',5,'Color','r');
title(['Tracer 1 is red. Tracer 2 is blue. Tracer 1 Initial Postion is [', ...
      num2str(rx),', ',num2str(ry),']. Tracer 2 Initial Position is [', ...
      num2str(qx),', ',num2str(qy),'].']);
axis([-5,5,-5,5]);
axis equal
hold on
plot(q(:,1),q(:,2),'LineWidth',5);
circlex=5*cos(2*pi/50*time);
circley=5*sin(2*pi/50*time);
plot(circlex,circley,'LineWidth',10,'Color','k');
plot(-b,0,'o');
plot(b,0,'o');
drawnow;
hold off;
subplot(1,2,2);
plot(time,distance);
drawnow;
pause(0.1);
end


figure;
avg=mean(storage,2);
plot(time,avg,'LineWidth',3,'Color','r');
lin=polyfit(transpose(time),avg,1);
reg=lin(1)*time+lin(2);
hold on;
```

```
plot(time,reg,'LineWidth',3,'Color','k');
title(lin(1));
max(lyapunov)

figure('units','normalized','outerposition',[0 0 1 1]); % Opens full screen

plot(r(:,1),r(:,2),'LineWidth',5,'Color','r');
title(['Tracer 1 is red. Tracer 2 is blue. Tracer 1 Initial Postion is [', ...
      num2str(rx),', ',num2str(ry),']. Tracer 2 Initial Position is [', ...
      num2str(qx),', ',num2str(qy),'].']);
axis([-5,5,-5,5]);
axis equal
hold on
plot(q(:,1),q(:,2),'LineWidth',5);
circlex=5*cos(2*pi/50*time);
circley=5*sin(2*pi/50*time);
plot(circlex,circley,'LineWidth',10,'Color','k');
plot(-b,0,'o');
plot(b,0,'o');
```

C Appendix C: Navier-Stokes Code

```
clear all;

clf;

clc;


nu = 0.0000011; % kinetic viscosity of water @ 15C ...

(see Table 2.2, p. 7, Tryggeson) [m^2/s]


I = 0.010; % current [A]

B = [0,0,1]; % magnetic field strength [T]

a = 0.015; % 1/2 seperation between electrodes [m]


dx = 0.001; % x step [m]

dy = 0.001; % y step [m]

dt = 0.05; % time step [s]


length = (dx+dy)/2; % "wire length", distance between grid points [?] [m]


xf = 0.111; % x length [m]

yf = 0.111; % y length [m]

tf = 25; % t length [s]


x = 0:dx:xf; % x-grid

y = 0:dy:yf; % y-grid

t = 0:dt:tf; % time frame

P = 16.67; % period of blinking [s]
```

```
xcenter = xf/(2*dx)+1; % x-center indice
ycenter = yf/(2*dy)+1; % y-center indice

e1 = [xcenter-a/dx,ycenter]; % left electrode location
e2 = [xcenter+a/dx,ycenter]; % right electrode location

J = I/((xf/dx+1)*(yf/dy+1))*(4/pi); % current density [A/m^2] ...
dividing current equally into each grid point

u1 = zeros(round(xf/dx+1),round(yf/dy+1),round(tf/dt+1));
% x-component of fluid velocity at location and time: (x,y,t) [m/s]
u2 = zeros(round(xf/dx+1),round(yf/dy+1),round(tf/dt+1));
% y-component of fluid velocity at location and time: (x,y,t) [m/s]

% initial conditions
% u1(1:3,:,1) = 0; % speed of fluid in x direction at t = 0 [m/s]
% u2(1:3,:,1) = 0; % speed of fluid in y direction at t = 0 [m/s]
% u1(xf/dx:xf/dx+1,yf/dy:yf/dy+1,1) = 0;
% speed of fluid in x direction at t = 0 [m/s]
% u2(xf/dx:xf/dx+1,yf/dy:yf/dy+1,1) = 0;
% speed of fluid in y direction at t = 0 [m/s]

delta_u1 = zeros(tf/dt+1,1);
delta_u2 = zeros(tf/dt+1,1);

t1x0 = (e1(1)-1)*dx + 0.00329617; % tracer 1 initial x position
t1y0 = (e1(2)-1)*dy - 0.0087201; % tracer 1 initial y position
```

```
tracerp1 = zeros(tf/dt+1,2);
tracerc1 = zeros(tf/dt+1,2);
tracerp1(1,:) = [t1x0,t1y0];
tracerc1(1,:) = [t1y0,t1x0];

t2x0 = (e1(1)-1)*dx - 0.005931; % tracer 2 initial x position
t2y0 = (e1(2)-1)*dy - 0.0111218; % tracer 2 initial y position
tracerp2 = zeros(tf/dt+1,2);
tracerc2 = zeros(tf/dt+1,2);
tracerp2(1,:) = [t2x0,t2y0];
tracerc2(1,:) = [t2y0,t2x0];

% calculations

for k = 1:round(tf/dt+1) % time loop
    time = (k-1)*dt;
    current = sin(2*pi*time/P) + sin(3*(2*pi*time/P))/3 ...
    + sin(5*(2*pi*time/P))/5 + sin(7*(2*pi*time/P))/7 ...
    + sin(9*(2*pi*time/P))/9;
          if current > 0
              active = [e1, 0]; % activates left electrode
          else
              active = [e2, 0]; % activates right electrode
          end
%           u1(active(1), active(2), k) = 0;
% active electrode has no velocity [m/s]
%           u2(active(1), active(2), k) = 0;
```

```
% active electrode has no velocity [m/s]


   for i = 2:round(xf/dx) % x-loop
       for j = 2:round(yf/dy) % y-loop
           location = [i,j,0];
           xloc = (i-1)*dx;
           yloc = (j-1)*dy;
           if and(i == active(1), j == active(2))
                F1 = 0;
                F2 = 0;
           elseif norm([xcenter,ycenter,0]-location) <= xf/2/dx
                direction = (active-location)./norm(active-location);
                % unit vector pointing towards active electrode, ...
                allows for IlxB calculation
                dfe = sqrt((((active(1)-1)*dx)-(xloc))^2 + ...
                (((active(2)-1)*dy)-(yloc))^2); % distance from electrode [m]
                ratio = 4*exp(-0.75*(dfe)/(dx))+1;
                % 1/r dependence of current density [?]
                ILB = length*ratio*J*cross(direction,B); % force calculation
                F1 = ILB(1); % force in x direction [N]
                F2 = ILB(2); % force in y direction [N]
           else
               F1 = 0;
               F2 = 0;
           end

           delta_u1(k) = nu*(u1(i+1,j,k)-2*u1(i,j,k)+u1(i-1,j,k))/(dx^2)*dt ...
```

```
        + F1*dt - u1(i,j,k)*(u1(i+1,j,k)-u1(i-1,j,k))/(2*dx)*dt;
        delta_u2(k) = nu*(u2(i,j+1,k)-2*u2(i,j,k)+u2(i,j-1,k))/(dy^2)*dt ...
        + F2*dt - u2(i,j,k)*(u2(i,j+1,k)-u2(i,j-1,k))/(2*dy)*dt;

        u1(i,j,k+1) = u1(i,j,k) + delta_u1(k);
        u2(i,j,k+1) = u2(i,j,k) + delta_u2(k);
    end
end
 sf = 1000000; % scale factor
 sf1 = 1000000000; % scale factor

 quiver(u1(:,:,k+1)',u2(:,:,k+1)');
 title(time);
 drawnow;

 cu1 = u1(:,:,k+1)';
 cu2 = u2(:,:,k+1)';

 rdp = 3; % round decimal point
 rxt1 = round(tracerp1(k,1),rdp);
 if rxt1-tracerp1(k,1) > 0
     fxt1 = rxt1-dx;
     cxt1 = rxt1;
 else
     fxt1 = rxt1;
     cxt1 = rxt1+dx;
 end
```

```
ryt1 = round(tracerp1(k,2),rdp);
if ryt1-tracerp1(k,2) > 0
    fyt1 = ryt1-dy;
    cyt1 = ryt1;
else
    fyt1 = ryt1;
    cyt1 = ryt1+dy;
end

x1y1t1 = [round(fyt1/dy+1),round(fxt1/dx+1)];
x1y2t1 = [round(cyt1/dy+1),round(fxt1/dx+1)];
x2y1t1 = [round(fyt1/dy+1),round(cxt1/dx+1)];
x2y2t1 = [round(cyt1/dy+1),round(cxt1/dx+1)];

rxt2 = round(tracerp2(k,1),rdp);
if rxt2-tracerp2(k,1) > 0
    fxt2 = rxt2-dx;
    cxt2 = rxt2;
else
    fxt2 = rxt2;
    cxt2 = rxt2+dx;
end
ryt2 = round(tracerp2(k,2),rdp);
if ryt2-tracerp2(k,2) > 0
    fyt2 = ryt2-dy;
    cyt2 = ryt2;
else
```

```
    fyt2 = ryt2;
    cyt2 = ryt2+dy;
end

x1y1t2 = [round(fyt2/dy+1),round(fxt2/dx+1)];
x1y2t2 = [round(cyt2/dy+1),round(fxt2/dx+1)];
x2y1t2 = [round(fyt2/dy+1),round(cxt2/dx+1)];
x2y2t2 = [round(cyt2/dy+1),round(cxt2/dx+1)];

xyt1 = (1/(dx*dy))*((cxt1-tracerp1(k,1))*(cyt1-tracerp1(k,2))
       *[cu1(x1y1t1(1),x1y1t1(2))*sf,cu2(x1y1t1(1),x1y1t1(2))*sf] + ...
       (tracerp1(k,1)-fxt1)*(cyt1-tracerp1(k,2))
       *[cu1(x2y1t1(1),x2y1t1(2))*sf,cu2(x2y1t1(1),x2y1t1(2))*sf] + ...
       (cxt1-tracerp1(k,1))*(tracerp1(k,2)-fyt1)
       *[cu1(x1y2t1(1),x1y2t1(2))*sf,cu2(x1y2t1(1),x1y2t1(2))*sf] + ...
       (tracerp1(k,1)-fxt1)*(tracerp1(k,2)-fyt1)
       *[cu1(x2y2t1(1),x2y2t1(2))*sf,cu2(x2y2t1(1),x2y2t1(2))*sf]);

xyt2 = (1/(dx*dy))*((cxt2-tracerp2(k,1))*(cyt2-tracerp2(k,2))
       *[cu1(x1y1t2(1),x1y1t2(2))*sf,cu2(x1y1t2(1),x1y1t2(2))*sf] + ...
       (tracerp2(k,1)-fxt2)*(cyt2-tracerp2(k,2))
       *[cu1(x2y1t2(1),x2y1t2(2))*sf,cu2(x2y1t2(1),x2y1t2(2))*sf] + ...
       (cxt2-tracerp2(k,1))*(tracerp2(k,2)-fyt2)
       *[cu1(x1y2t2(1),x1y1t2(2))*sf,cu2(x1y2t2(1),x1y2t2(2))*sf] + ...
       (tracerp2(k,1)-fxt2)*(tracerp2(k,2)-fyt2)
       *[cu1(x2y2t2(1),x2y2t2(2))*sf,cu2(x2y2t2(1),x2y2t2(2))*sf]);
```

```
    deltat1x = xyt1(1)*dt;
    deltat1y = xyt1(2)*dt;
    deltat2x = xyt2(1)*dt;
    deltat2y = xyt2(2)*dt;
    tracerc1(k+1,:) = [tracerc1(k,2)+deltat1y,tracerc1(k,1)+deltat1x];
    tracerc2(k+1,:) = [tracerc2(k,2)+deltat2y,tracerc2(k,1)+deltat2x];
    tracerp1(k+1,:) = [tracerp1(k,1)+deltat1x,tracerp1(k,2)+deltat1y];
    tracerp2(k+1,:) = [tracerp2(k,1)+deltat2x,tracerp2(k,2)+deltat2y];

    hold on
    plot(tracerp1(k+1,1)/dx+1,tracerp1(k+1,2)/dy+1,'ro');
    plot(tracerp2(k+1,1)/dx+1,tracerp2(k+1,2)/dy+1,'ko');
    pause(0.01);
    hold off

end
```