Design and Experimentation of Complex Dynamical Systems for Intelligent Navigation

Nathan Phan[1] and Ani Hseih[2]

[1]*Dept. of Chemical Engineering, The Ohio State University, SUNFEST Fellow*
[2]*Dept. of Mechanical Engineering and Applied Mathematics, The University of Pennsylvania*

***Abstract*—Managing the consumption of resources exists regardless of discipline. In the case of oceanic endeavors, this principle manifests due to both energetic and physical constraints, including fuel and motor capabilities. In order to advance our understanding of robotic systems and interactions, we must learn to overcome these obstacles, one method being more intelligent planning through the various complex flows found in practice. Evolving to meet this goal involves three discrete steps: fabrication of these complex flows, implementation of algorithms to build intelligence, and experimentation using the previously developed miniature autonomous surface vehicles (mASV). This paper details the first two steps, exploring the process, results, and other applications of each. Numerical integration of complex flows were developed within Python, improving our intuition detailing their characteristics and properties of formation. Once sufficient, miniature flows were conceived, translating theory into experiment, finally scaling into the Multi-Robot Tank. After modeling these complex flows, focus shifted to quantitative and qualitative analysis of them, with hopes of translating the trends into a data structure, whereupon algorithms for intelligent navigation to be implemented. Implications of the findings further our capabilities with the limited resources at hand, allowing more efficient boats and testing.**

I. Introduction

In practice, studying fluid flows is non-trivial, as they make their systems non-ideal. Consideration of such complexities drives the necessary cost of intellectual exploration to unfeasible lengths, both computationally and physically [1]. Even the data provided by these methods may be null, as the minute errors evolve into an uncertainty cascade [2]. One area severely affected by this phenomenon of particular interest to this paper is the study of robots in dynamical systems, specifically oceans and other bodies of water. The successful operation of autonomous surface vehicles (ASV) hinges on the understanding of their environments, as they can make more intelligent locomotive decisions, saving both time and energy.

Currently, researchers turn to computer simulations to model flows. This requires descriptions, or models, of the flow that can be evaluated by a computer, *e.g.* vector field (Fig. 3). Numerical integration is one way to evaluate flow models. While the production of simpler flows poses little issue, the introduction of obstacles and imperfections in the container, both of which are found in practice, brings with them complexities that greatly increase the mathematics needed, as well as computational requirements that may be out of reach. The method of physical development and testing in this paper challenges this approach, proposing an alternative that can greatly reduce the time required to see quality and, more importantly, realistic data to be examined.

To better understand energy efficient navigation for robot teams we consider three steps: fabrication of these complex flows, implementation of navigation algorithms, and experimentation using miniature autonomous surface vehicles (mASV). The objective of this summer is to fabricate flow environments, and use the information derived from them to advance intelligent navigation efforts. First, we explore the capabilities of numerical integration, modeling a point particle over a vector field. This helps build intuition for flows before attempting any experimentation. Next, to allow for fast development iteration, miniature flows are produced. Finally, the miniature flows are scaled into the larger tank, where data is collected about the flow. The three flows examined during this project include the single gyre, the double gyre, and a flow with an obstacle. Finally, potential algorithms are considered for mASV navigation in flows.

The contributions of this research project are:

1) Reliable single and double gyre environments for further testing of the mASV.
2) Quantitative data of the flow's properties that could be used later on.
3) Simulations and data-driven graphs for future presentations and reports.
4) Foundational knowledge for algorithms to be implemented into intelligent path planning.

II. Background

This summer, we explored a wide variety of fields to further our objective of understanding flow environments to improve robotic planning.

A. Numerical Integration

Numerical Integration includes computational methods to approximate an integral through its integrands. One example is the Taylor series expansion, and a common approximation method is to use Euler's Method. Euler's Method is,

$$x_{n+1} = x_n + f(x_n)\Delta t \tag{1}$$

which evaluates first order ordinary differential equations (ODEs). It approximates the next point, x_{n+1}, using its predecessor, x_n, allowing for an estimate to be made towards the actual shape of the curve. While Euler's method works well in some cases, certain ODEs require higher order to capture the true dynamics. Alternative numerical integration methods can include these terms and result in approximations that are closer to reality. One explored is the Refined Euler's Method, which evaluates the average of the two points given by Euler's method,

$$\tilde{x}_{n+1} = x_n + f(x_n)\Delta t, \tag{2}$$

$$x_{n+1} = x_n + \frac{1}{2}[f(x_n) + f(\tilde{x}_{n+1})]\Delta t \tag{3}$$

where $\tilde{x}$ is the trial point and x is the actual step taken from averaging the start and the trial.

B. Dynamical Systems

Time-dependent systems, otherwise known as dynamical systems, offer insight into our changing world. This paper plans to discuss them through the usage of ordinary differential equations (ODEs), using them as a framework to detail the simulation portion of the project.

For example, the simple harmonic pendulum provides a reliable platform to establish a strong intuition for dynamics. This toy problem can be extended to include additional forces such as damping [4].

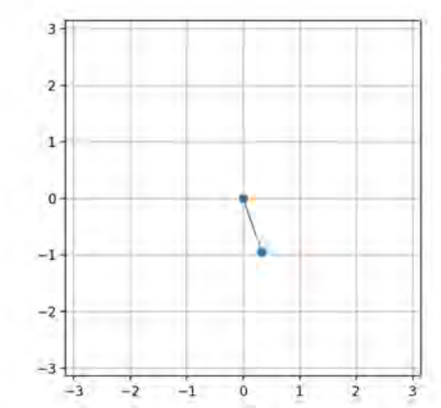

Fig. 1. Simulation of a Simple Pendulum Using Euler's Method

Using Euler's method (1) we simulate a simple pendulum. The dynamics of the simple pendulum are,

$$\frac{d^2\theta}{dt^2} = -\frac{g}{L}\sin(\theta) \tag{4}$$

where g is gravity, L is length of the rod, θ is the state of the pendulum. In this project, we focus on simulating various gyres to improve intuition for these complex dynamic environments.

C. Data Structure and Algorithms

Dijkstra is a foundational graph search algorithm used in intelligent robotic navigation and planning [7]. It is a greedy method that traverses a weighted graph, picking the edge with the smallest value at that specific point. This approach suffers as the graphs become more complex, since it explores every possible connection at every node. This greatly increases the computational time and power required as the number of nodes and edges grow.

There are extensions of Dijkstra to improve graph search. For example, A* adds a heuristic to Dijkstra, which accounts for the location of the goal, picking nodes that carry the path towards it. In this work, we implement Dijkstra as a first step to understand ways to perform robot path planning in flows.

III. Methodology

The objective of this project is to generate environments where robots require intelligent navigation. Complex environments, such as oceans and rivers, may cause losses in efficiency if traversed improperly, limiting the robot's capability to complete its intended objective.

A. Simulation of Dynamical Systems

Circular oceanic patterns, known as gyres are the focus of this project. Gyres are observed throughout the world, and understanding their properties and, more specifically, the interactions they have with objects such as ships or floaters, may provide key insights needed to improve navigational needs.

First, we simulated flows to better understand the theoretical underpinnings of the environments we were trying to generate. The principles learned from integrating the simple pendulum were then translated to both the singe gyre and double gyre. These time independent flows are seen in Figures 2 and 3 below. The ODEs for the double gyre are,

$$\dot{x} = -\frac{\partial \psi}{\partial y}, \tag{5}$$

$$\dot{y} = \frac{\partial \psi}{\partial x} \tag{6}$$

where ψ is

$$\psi(x,y) = \sin(\pi x)\sin(\pi y). \tag{7}$$

To properly display the simulation, an initial plot of the single gyre vector graph was conceived, who's vectors produced from the partial derivation of the double gyre equation, shifting its frame such that it only shows a singular gyre. The vectors were composed using the components provided by the partial differentiation of the time independent double gyre equation by x and y.

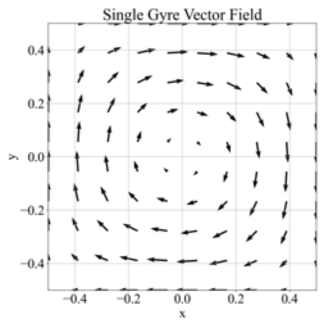

Fig. 2. Vector field of a time independent single gyre flow

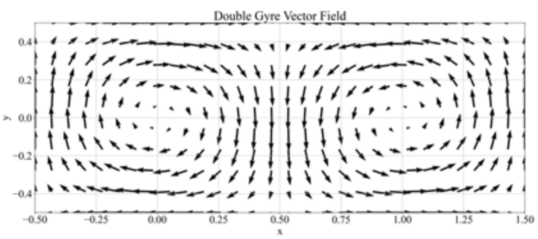

Fig. 3. Vector field of a time independent double gyre flow

Adding Euler's method lead to the production of a particle moving about a single gyre flow, however this particular model faced drawbacks.

Euler's method, as stated before, suffers from issues with accuracy, revealing themselves though the particle's lack of stability. The theory suggests that the center of a gyre is its calmest part [5], but within the simulation, the particle readily escaped. A simple fix would be adjusting the time step. By shortening Δt in Equation (**??**), Euler's method suggests that its accuracy increases. While true up to a certain point, the method falls short yet again in one major respect. First, decreasing the time step increases the computational expenses of the program. While minimal increase of computational time was seen, with more complex systems, this can rapidly spiral [6].

To combat this, we use a refined Euler's method. As seen in Equation (2) , the method relies on the average of the two steps, increasing the accuracy while maintaining the same initial step size. This manifested in an expected greater stability within the core of the gyre, as the particle was more inclined to remain in its position.

With a practical understanding of simulating flows the next step is to attempt to experimentally generate small flow like environments.

B. Experimental Testing

Mini flows are generated in order to obtain physical intuition for flows before applying insights to a larger tank. This requires understanding where to place propellers, as well as what velocity of flow results in strong visibility.

Due to the volume of ScalAR's Multi-Robot Tank, initial prototyping of the desired flows would be impractical. The time required to fill or drain the tank would force the project to a halt, as testing water voltage ratios would be imperative to further testing. Additionally, double gyre and obstacle driven flows were novel to the group, with none having ever been generated before. Lacking intuition on how to build these flows, what water to voltage ratios would be most optimal, and time constrictions, it was imperative that flow testing started in a smaller, more manageable environment.

The intuition gained from these flows is then applied to the Multi-Robot tank. We place drifters in the flow, and track their trajectories using an OptiTrack motion capture system.

The ScalAR lab is interested in having robots operate in flow like environments. This requires modeling the environment, generating these environments experimentally, and intelligent navigation methods for robots in the flow. Knowledge of this environment, qualitatively observed from the simulated and physical flows, provides insight on potential intelligent navigation strategies.

C. Navigation in flows

Given a single gyre, we built a graph model exploiting the gyre shape. The graph is a 6-node complete graph, connected with weighted edges. The weights for these edges followed two presumptions derived from the data collected from the single gyre flow.

These assumptions are:

1) Traveling with the direction of the gyre would be vastly more efficient than vice versa
2) Traveling about the flow to reach the desired location would be vastly superior as compared to through the gyre, as no drifter was seen to travel across the gyre.

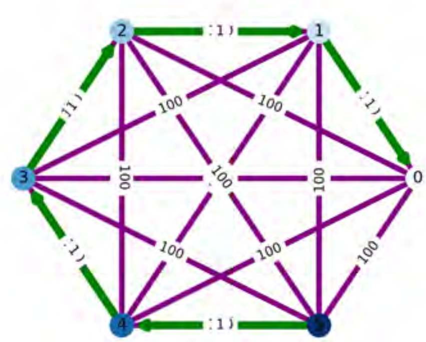

Fig. 4. Graph representation of a CW single gyre, highlighting the most efficient path

In our specific iteration, Dijkstra looks for the path with the lowest value, representing the least energetically taxing path. These graphs were created using dictionaries, with keys being the node names and values being the edges. Each edge was directional, allowing us to assign differing values depending on the direction of travel. As previously mentioned, assumptions are made that traveling about the flow will be more efficient than against it, and that traveling through the center of the gyre would be more difficult than about it. These two assumptions lead the edges that follow the direction of the gyre to be weighted significantly less than others. This can be seen in Figure 4

IV. Experimental Setup

Flows are generated in two experimental environments: the miniature flows and the Multi-Robot tank.

Below, the miniature single gyre flow is pictured (Fig. 5). This flow features two vibration motors, directing the flow of water in the direction stipulated in the figure. The particle's movement provided a basis in which we were able to determine that the flow had been created, despite its non-ideal shape.

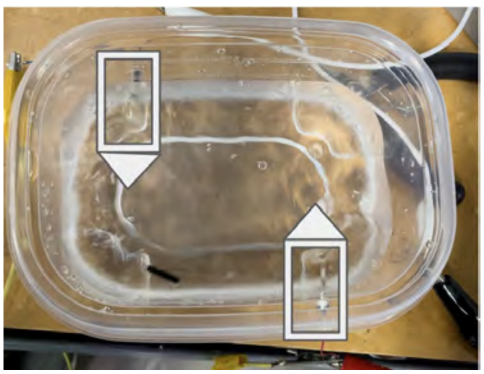

Fig. 5. Miniature Single Gyre flow

The double gyre configuration saw more iterations, due to competing views for its construction. Three (Fig. 6) and four (Fig. 7) propeller designs of the flow were made and tested, both configured in a way such that the tank geometry itself was leveraged in order to produce a satisfactory flow. Upon final evaluation, the four propeller was chosen for future experimentation.

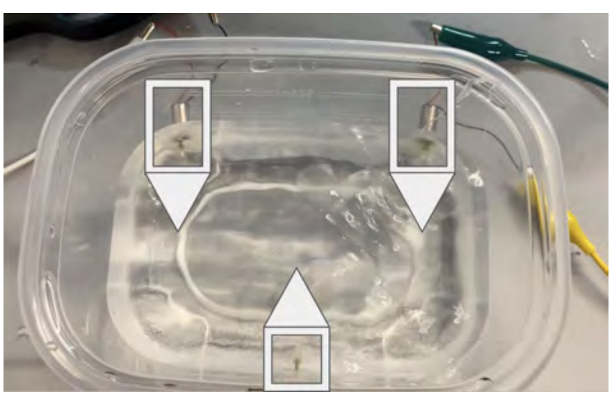

Fig. 6. Three propeller configuration of the double gyre flow

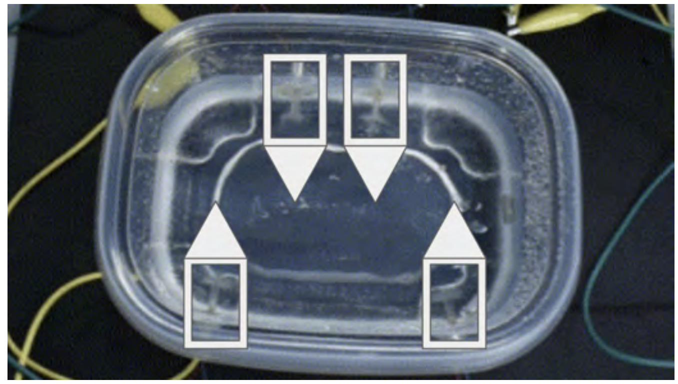

Fig. 7. Four propeller configuration of the double gyre flow

While a miniature obstacle driven flow was produced, the scale of the obstacles to the flow size proved to be a major issue. Due to experimental setup issues, these results are excluded from this report.

With each miniature flow proving successful, they were scaled to the dimensions of the MultiRobot tank. This process was seamless, as the miniature flow provided strong intuition regarding how to arrange propellers such that the geometry of the tank was leveraged. The scaled products are as followed:

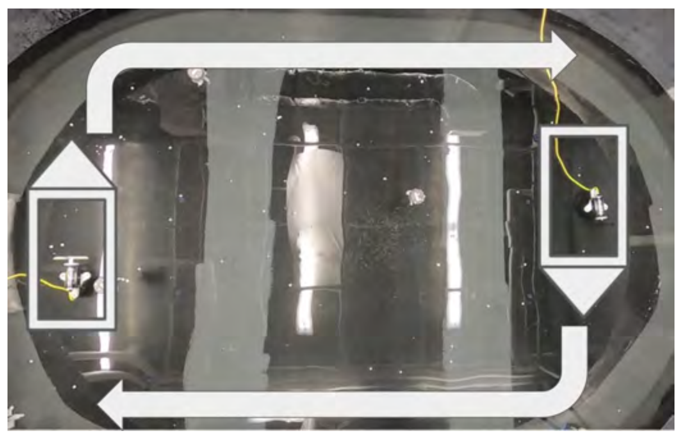

Fig. 8. Multi-Robot tank single gyre

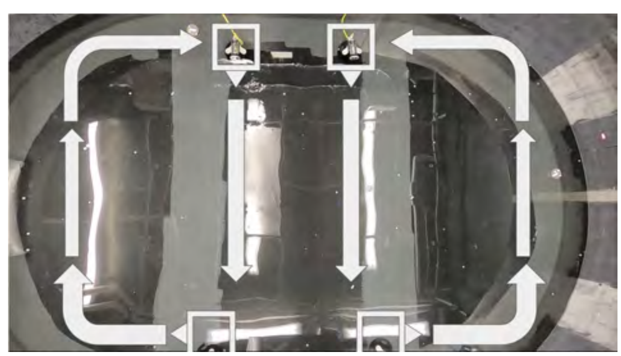

Fig. 9. Multi-Robot tank double gyre

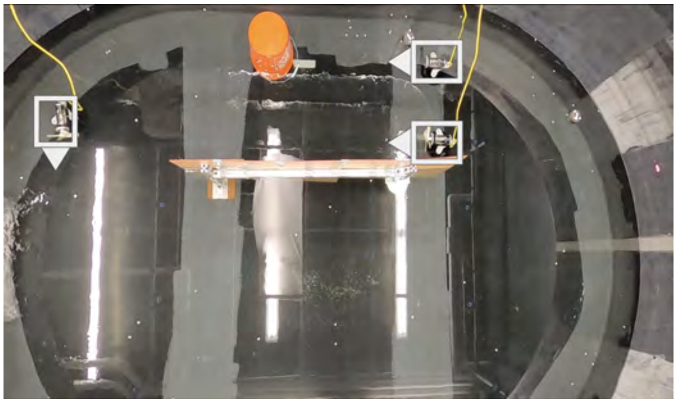

Fig. 10. Multi-Robot tank obstacle driven flow

Figures 8, 9, and 10 each highlight the location of the propellers, as well as the direction in which the water travels about them. In the cases of the single and double gyres, the intended flow pattern is depicted as well.

Fig. 11. Passive Drifter

Within each of the Multi-Robot tank flows, drifters (Fig. 11) were used in conjunction with the lab's Optitrack system, collecting data which was then plotted using Python. The data observed from the flows built upon previous intuition that would later be used in the graph modeling process. The visual data collected is shown in Figures 13, 13, and 14.

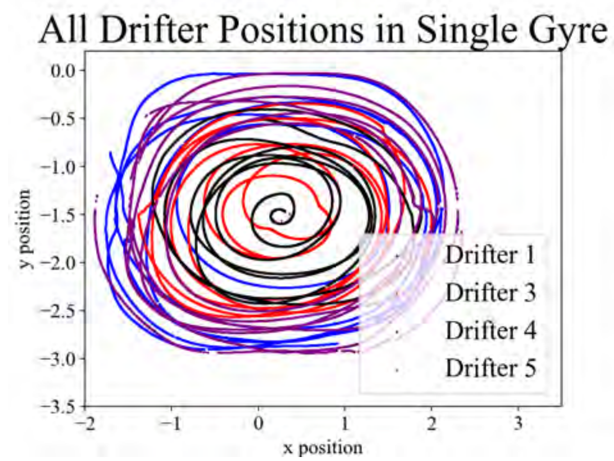

Fig. 12. Drifter Data for Single Gyre flow

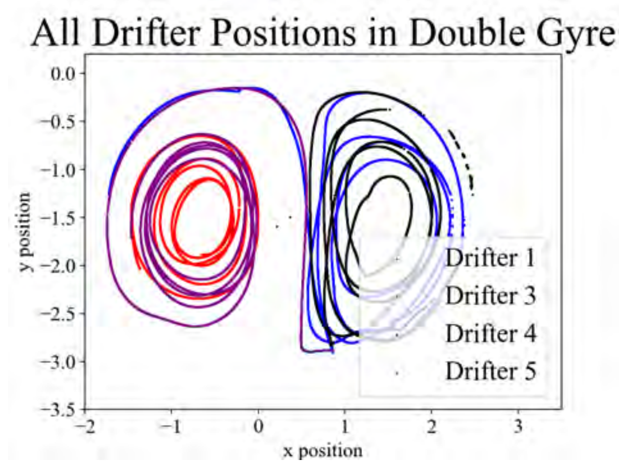

Fig. 13. Drifter Data for Double Gyre flow

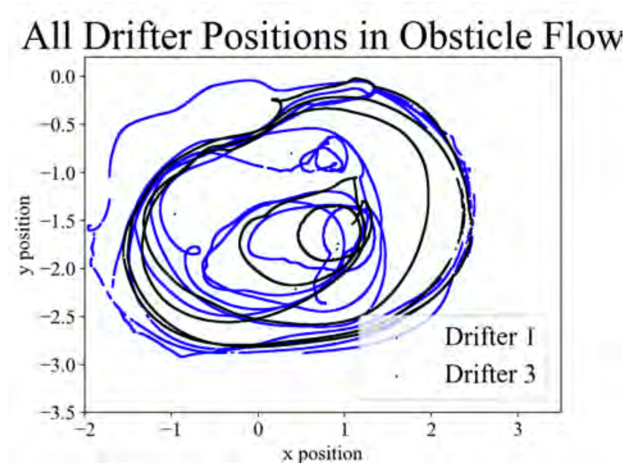

Fig. 14. Drifter Data for Obstacle Driven flow

V. Results and Discussion

Initial testing of the both the single and double gyres, and obstacle driven flow, all proved successful, with clear flow paths being produced. Throughout the process, initial intuition regarding these structures were both broken and reinforced.

A. Navigation in flows

The path planning implementation in the single gyre was successful, as it it gave a path that aligns with both the vector field (Fig. 2) and single gyre direction. No matter the start or end node, the algorithm gave the least energetically taxing path, even if the total distance was longer.

There are two issues with this graph model of the single gyre flow, both stemming from its simplicity.

As seen in the overlaid image (Fig. 15), the lack of nodes generalizes the flow to the point of potential inaccuracy. It neglects the nuances found within the flow, as the edges found within the graph may not align with the vectors of the flow as seen in the vector field (Fig. 2). These discrepancies would likely cause losses in efficiency.

Another issue would be the weights assigned to the edges. Though practical, clearly displaying the most natural path about the flow, they lack specificity. Intuition states that the flow speed around a propeller would be greater than the regions without it, but the graph fails to account for that phenom. Like with the lack of nodes, lack of edge weight accuracy may cause for unwanted inaccuracies in the ideal path about a flow.

A solution to both issues might be a re imagining of the graph itself. Rather than using a 6-node directional complete graph, a mesh may be used to represent the entirely of the tank, rather than the flow itself. The mesh would allow for more nodes and with it more edges to accurately depict the nuances found within a single gyre flow.

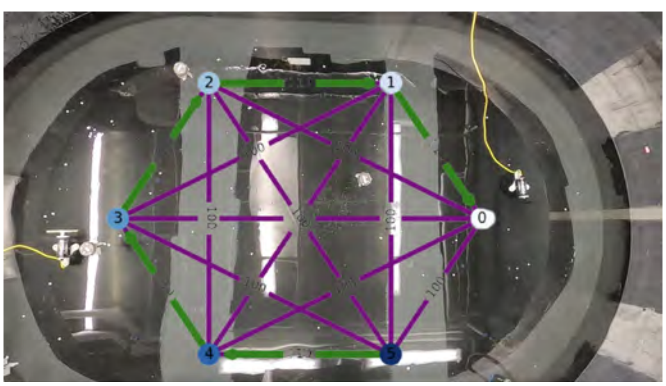

Fig. 15. Single Gyre Graph overlaid onto Multi-Robot Tank gyre

B. Overall construction

Suspicions around the construction of these flows were prevalent, but what remained consistent was the leveraging of the container's geometry. The walls in particular, posed as both a benefit and a hindrance, depending on the container used. In the miniature flows, the corners of the container seem to have produced unwanted vertices within the flow, causing the particles to become entrapped, temporarily halting their movement. This was later rectified once scaled to the Multi-Robot tank, which possesses a more ideal shape. This ideal shape allowed for more flexible configurations of the propellers. It was also determined that the clarity of the flow was a function of the flow speed, which in our case was determined by the voltage supplied to the propeller, and the amount of water in the tank. With faster speeds, indicated by higher voltage, the surface of the water begins to distort, muddying the flow. The opposite was seen with high water levels. The assumption made was that an increase of water essentially increases the amount of mass needed to be moved by the propellers. The optimal combination was learned to be a low flow speed combined with a low water level, which can be seen in Figure 16.

		Voltage						
		0.5	1	1.5	2	2.5	3	3.5
Volume	235	Weak flow	Good flow	Best flow	Unsteady flow	Too much turbulence	too much turbulence	
	295	Weak flow	good flow	Best flow	Unsteady flow	too much turbulence	too much turbulence	
	356	No flow	weak flow	Weak flow	Best flow	Unsteady flow	too much turbulence	
	476	No flow	No flow	Weak flow	Best flow	Unsteady flow	Unsteady flow	Too much turbulence

Fig. 16. Comparison of varying flow speeds (determined by voltage) and water volumes (determining the mass needed to be moved)

C. Three vs. Four propeller configurations for the double gyre

The three and four propeller configuration comparison posed as an interesting question. At its core, it seems to investigate the importance of a distinct median within the double gyre. The three propeller double gyre argues that the median is trivial, as a successful generation would indicate that one might not need to be fostered. The four supports the idea that the same median is in fact non-trivial to the generation of the double gyre, and that two the two sides must have a distinct region separating them. It was found that this median zone was non-trivial, and that separation of the two gyres has an impact on the clarity of the flow itself, inviting further testing of this separation.

D. Variations in flow velocities through position

Supported by the numerical integration of the velocity vector of a single gyre (Fig. 2), intuition states that the center of the gyre experiences a slower flow than the outer edges. Experimentally, this notion was supported. Below, the individual drifter paths (Fig. 17) and velocities (Fig. 18) are shown.

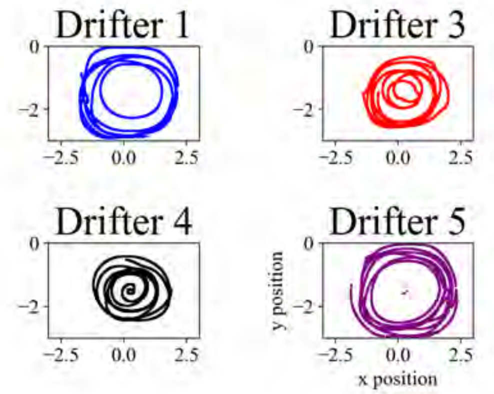

Fig. 17. Single Gyre drifter positions

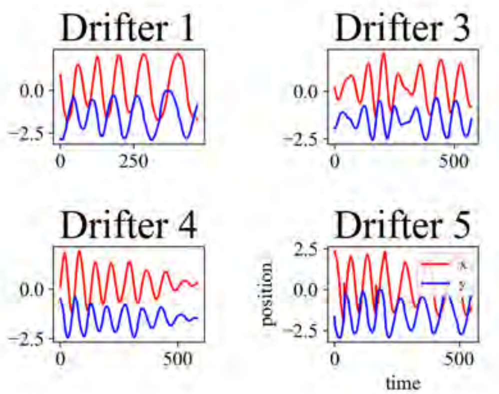

Fig. 18. Single Gyre drifter velocities

Drifter four is of interest, as it clearly displays the assumption made through the vector field. As the drifter approaches the center of the gyre, both the x and y velocities deteriorate. Compare that with the graphs of drifter five, which sees minimal fluctuation in its position vs time graph, presumably due to its relatively consistent path.

VI. Conclusion

The project concluded in an exciting area, one that allows for a variety of pivots, each proposing great value to the group. Of particular interest is continuing the graph implementation of the project. The graph implementation currently lacks accuracy, as there are simply not enough edges capable of being modified, leading to an overly simplified rendition of what it should be. A stronger graph would feature more nodes and with it more edges, so that more minute movements and shifts in velocity are accounted for. Additionally, intuition only leads to so much accuracy. While it is true that traveling along the flow is most efficient, this generalization may lead to poor path planning and efficiency.

VII. Acknowledgements

I would like to give immense thanks to my partner Arriella Mafuta, mentors Victoria Edwards, Thales C. Silva, principle investigator, Dr. Ani Hsieh, and everyone else in the ScalAR lab group for making this summer so memorable. Truly, I would be unable to accomplish this without you all. I would also like to thank the National Science Foundation for supporting SUNFEST through NSF REU grant no. 1950720.

References

[1] D. Kularatne, S. Bhattacharya and M. A. Hsieh, ”Optimal Path Planning in Time-Varying Flows Using Adaptive Discretization,” in IEEE Robotics and Automation Letters, vol. 3, no. 1, pp. 458-465, Jan. 2018, doi: 10.1109/LRA.2017.2761939.

[2] K. Szwaykowska and F. Zhang, “Controlled lagrangian particle tracking: Error growth under Feedback Control,” IEEE Transactions on Control Systems Technology, vol. 26, no. 3, pp. 874–889, 2018. doi:10.1109/tcst.2017.2695161

[3] R. A. Nelson and M. G. Olsson, “The pendulum—rich physics from a simple system,” American Journal of Physics, vol. 54, no. 2, pp. 112–121, 1986. doi:10.1119/1.14703

[4] H. E. Fiedler, “Coherent structures in turbulent flows,” Progress in Aerospace Sciences, vol. 25, no. 3, pp. 231–269, 1988. doi:10.1016/0376-0421(88)90001-2

[5] S. Fadugba, B. Ogunrinde, and T. Okunlola, “Euler's method for solving initial value problems in ordinary differential equations.,” Afe Babalola University Repository, http://eprints.abuad.edu.ng/244/ (accessed Aug. 2, 2023).

[6] Huijuan Wang, Yuan Yu, and Quanbo Yuan, “Application of dijkstra algorithm in Robot Path-Planning,” 2011 Second International Conference on Mechanic Automation and Control Engineering, 2011. doi:10.1109/mace.2011.5987118