Project Report: Challenging Nature—Ostraciiform Swim Gait Optimizing Via Adjoint Based Algorithms

George Kopf, Xuanhong An

August 2025

Abstract

In this work, we model the optimal gait for two-dimensional caudal swimming in Ostraciiform fish via pitch and heave. The goal is to employ an adjoint-based optimization algorithm to develop from simple sinusoidal motion towards more complex gaits that take advantage of complex vortice interactions to increase efficiency. When compared to existing models of fish biomechanics, we demonstrated preliminary evidenced that the algorithm increased swimming efficiency at low duty cycles. The high level optimization functions as a proof of concept that bio-inspired systems can be optimized for specific values using these methodologies.

1 Introduction

Bio-inspired robotics have widespread applications, and in the world of aquatic maneuvering, fish caudal structures have a number of advantages over traditional propeller systems[1, 2]. Navigating tight or complex terrain such as coral reefs, traversing environments such as dense seaweed forests that could tangle traditional propellers, generating thrust without creating unusual noise that stands out, and interacting with extant wildlife without modifying their behaviors are all prescribed advantages of using systems that simulate the swimming gait of a fish. These are swimming styles that have evolved over billions of years to work effectively in their native niches. Copying those core design principles into our robotic systems is an excellent way to utilize said optimization for our own purposes.

However, nature is not a vacuum. Animals face a number of conflicting selective pressures from environmental shifts, sexual traits, predatory threats, social traits, metabolic limitations, and more [3]. To that end, exactly copying existing fish biomechanics may not be the optimal solution when it comes to designing bio-inspired robots. If a fish needs to be effective at swimming efficiently *and* making sharp turns, we may be able to surpass it in just one category at the cost of the other.

In previous research, Dr. An et al. have developed an adjoint-based gradient descent algorithm designed to optimize the swimming gait of a 2D fish tail [4]. The pitch and

heave of the tail are used as parameters to generate a cost function that can be optimized with respect to either efficiency or thrust. However, in-depth testing of the algorithm and its practical applications has not been completed yet. The first goal of this project report is to discuss the modification of the algorithm parameters to get the system running properly at 1, 5, 10, and 30 Fourier modes.

Once the algorithm has been tuned to function effectively for our purposes, it is important to compare these optimization results with actual swimming gaits found in fish, such as Body-Caudal Swim gaits and Burst-and-Coast swimming [5, 6]. Ideally all variables would be fully controlled to match the simulated environment and methodology fully, but due to the limited scope and time of this program, we instead focused on creating a preliminary comparison between our optimized results and relevant biomechanical analyses found in existing literature.

2 Numerical Design

The full descriptive numerical methods of the algorithm's design are discussed in Dr. An's paper [4]. For the sake of brevity and to avoid repetition, this report will only briefly overview the important mathematical principles involved.

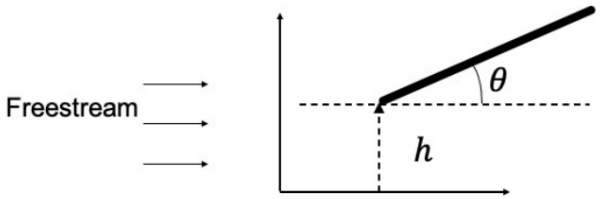


Figure 1: A drawing of the plate in the fluid.

The swimming gait is described by two series of Fourier modes representing pitch and heave of a two-dimensional flat plate experiencing continuous freestream current, where coefficients An, Bn, Cn, and Dn are the initial parameters of our swimming gait. τ is the period. These Fourier modes are what will be used to construct a cost function that relates our initial parameters to cost J, be it efficiency or thrust.

$$h(t) = \sum_{n=1}^{3} \left[A_n \sin(\frac{2\pi nt}{\tau}) + B_n \cos(\frac{2\pi nt}{\tau}) \right], n = 1, 3 \tag{1}$$

$$\theta(t) = \sum_{n=1}^{3} \left[C_n \sin(\frac{2\pi nt}{\tau}) + D_n \cos(\frac{2\pi nt}{\tau}) \right], n = 1, 3 \tag{2}$$

The two-dimensional Navier-Stokes solver employed uses the IBPM to solve the spatially discretized vorticity equation with vorticity ω.
The discrete operators are described in table 1.

Operator	Maps	Definition
C	$\psi \mapsto q$	curl of a scalar
C^T	$q \mapsto \omega$	curl of a vector in 2D
$L = -C^T C$	$\omega \mapsto \omega$	Laplacian, analogous to $\Delta u = \nabla(\nabla \cdot u) - \nabla \times \nabla \times u$
E	$q \mapsto u_B$	restriction of fluxes everywhere to velocity at boundary
E^T	$f \mapsto q$	regularization of forces at boundary points to fluxes everywhere

Table 1: Description of discrete operators.

$$\begin{cases} \dfrac{d\omega}{dt} + C^T E^T f = \nu L\omega + C^T(q \times \omega), \quad \omega|_\delta = bc_\omega \\ L\psi = -\omega, \quad \psi|_\delta = bc_\psi \\ EC\psi = u_B. \end{cases} \tag{3}$$

Optimization is commonly performed via gradient descent, but for our applications computing the gradient of this high-dimensional parameter space is unfeasible, even utilizing super computer clusters. To that end, an adjoint system is derived to solve for the gradient cost function.

$$\begin{cases} \dfrac{d\omega}{dt} + C^T E^T f = -\nu C^T C\omega + C^T \left\{[C(C^T C)^{-1}\omega + q_{pot}] \times \omega\right\}, \quad \omega|_\delta = bc_\omega \\ EC(C^T C)^{-1}\omega = u_B. \end{cases} \tag{4}$$

This adjoint method gives us a way to compute the gradient of the cost function with respect to the pitch and heave parameters. By looping through this algorithm, we can continually update the parameters as we optimize J.

The gradient of J with respect to A_n is

$$\int_0^\tau \eta^T \sin(\frac{2\pi nt}{\tau})dt, \tag{5}$$

the gradient of J with respect to B_n is

$$\int_0^\tau \eta^T \cos(\frac{2\pi nt}{\tau})dt, \tag{6}$$

the gradient of J with respect to C_n is

$$\int_0^\tau \zeta^T \sin(\frac{2\pi nt}{\tau})dt, \tag{7}$$

the gradient of J with respect to D_n is

$$\int_0^\tau \zeta^T \cos(\frac{2\pi nt}{\tau})dt, \tag{8}$$

and the gradient of J with respect to τ is

$$\begin{aligned}
&\int_0^\tau \frac{\partial g}{\partial \tau} dt \\
&+ \int_0^\tau \eta^T \left\{ \sum_{n=1}^{N} \left[A_n \cos(\frac{2\pi nt}{\tau}) 2\pi nt(-1)\tau^{-2} - B_n \sin(\frac{2\pi nt}{\tau}) 2\pi nt(-1)\tau^{-2} \right] \right\} dt \\
&+ \int_0^\tau \zeta^T \left\{ \sum_{n=1}^{N} \left[C_n \cos(\frac{2\pi nt}{\tau}) 2\pi nt(-1)\tau^{-2} - D_n \sin(\frac{2\pi nt}{\tau}) 2\pi nt(-1)\tau^{-2} \right] \right\} dt \\
&+ \xi(\tau)^T (C^T C)^{-1} \left\{ \frac{d\omega(\tau)}{dt} + C^T E^T f(\tau) + \nu C^T C\omega(\tau) - C^T \left[C(C^T C)^{-1} \omega(\tau) \times \omega(\tau) + q_{pot} \times \omega(\tau) \right] \right\} \\
&+ \phi(\tau)^T \left[EC(C^T C)^{-1} \omega(\tau) - u_B(\tau) \right] \\
&+ \left\{ \alpha(\tau)^T \left[\frac{dh(\tau)}{dt} - a(\tau) \right] + \beta(\tau)^T \left[\frac{d\theta(\tau)}{dt} - b(\tau) \right] \right\} \\
&+ \eta(\tau)^T \left\{ \sum_{n=1}^{N} \left[A_n \sin(\frac{2\pi n\tau}{\tau}) + B_n \cos(\frac{2\pi n\tau}{\tau}) \right] - h(\tau) \right\} \\
&+ \zeta(\tau)^T \left\{ \sum_{n=1}^{N} \left[C_n \sin(\frac{2\pi n\tau}{\tau}) + D_n \cos(\frac{2\pi n\tau}{\tau}) \right] - \theta(\tau) \right\} \\
&+ g(\tau). \qquad (9)
\end{aligned}$$

3 Methods

The bulk of the time spent on this research project involved testing a variety of parameters and settings in the optimization algorithm to generate viable results on an efficient time scale. Our first goal was to optimize the swimming gaits of the fish for efficiency, in order to capture a swimming gait that could be compared to the gaits collected or modeled in the current bioanatomical literature.

Part of the struggle was due to the limited time span assigned to the project, meaning that multi-day run times were an unacceptable iteration rate in terms of debugging and data collection. It was a balance between resolution and efficiency. In the end, setting the algorithm to a 100x100 grid size, with a Reynold's number of 100, could run for 30 iterations at the maximum Fourier mode count of 30 in under 8 hours (more Fourier modes means more complex waveforms are being calculated—they can be considered as analogous to more variables in a simulation). In theory it should run slightly faster than that, but in practice this is the best that could be achieved.

Another way to decrease iteration time while debugging was to stagger the optimization process. We began running input guesses into a single Fourier mode optimization, then took the output values for each parameter and input them back into the algorithm set to 5 Fourier modes. That output 5 values for each parameter, which we input into the 10 Fourier mode system. Then from 10 we input to 30. In this way, the optimization

algorithm could perform the bulk of the process at lower Fourier modes, allowing for fast iteration times (under 3 hours!) as well as interesting results that will be discussed later.

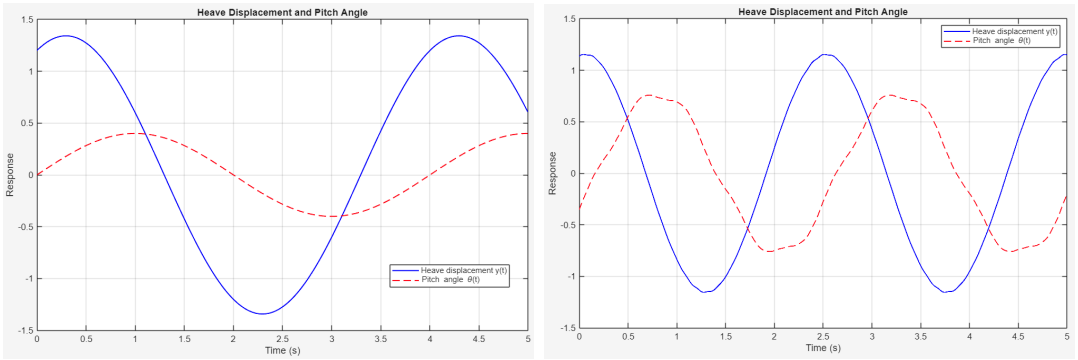
Unfortunately, this project had to sacrifice the linear search function from Dr. Xuanhong's original algorithm. With our setup, it kept causing values to spike violently or give nonreal answers for unknown reasons, and it was determined that the extensive time invested in debugging was no longer worth it. Thus, we moved on, setting the algorithm to use standard step sizes for the learning rate. As the number of Fourier modes used increases, the learning rate must decrease in turn so as to not miss minute optimizations. Without the linear search functionality, we simply performed this manually, brute force testing different values to see which worked best.

One last interesting thing to note here: Most of the analysis was done with the goal of optimizing for efficiency, but the system can also optimize for pure thrust regardless of the energy cost. We used this functionality as a sanity test, because optimizing for thrust is an unbounded system. In this scenario, we should see all parameters spike to their maximum or minimum values very quickly. This would be an easy way to confirm the basic logic of the algorithm is functioning.

4 Results

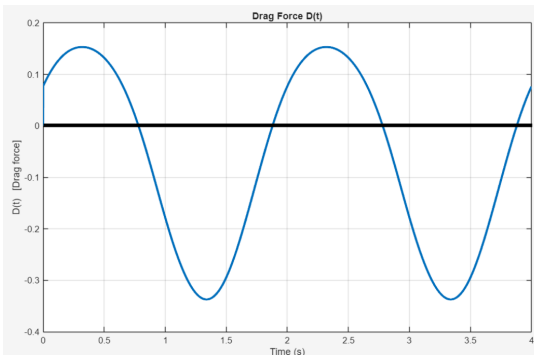
4.1 Efficiency Optimization Data

The following section overviews the results from the optimization of swim gait for efficiency. The initial guess, taken from An et al. [4], consists of basic sinusoidal waveforms, with heaving leading pitching by a phase difference of 90^o. This is shown in figure 2. It is immediately apparent that the optimization process had a strong effect on the pitching motion of the tail. Both pitch and heave also experienced increases in frequency, but the heave remained relatively unchanged aside from that.

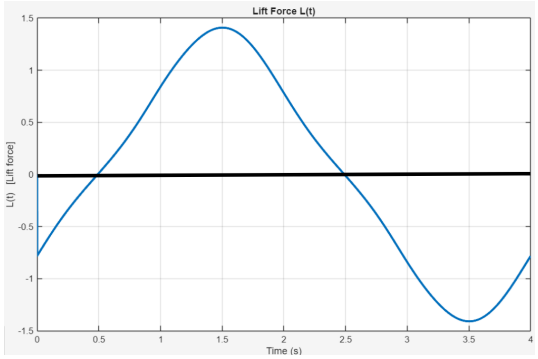


(a) Initial Pitch and Heave dynamics. (b) Final Pitch and Heave dynamics.

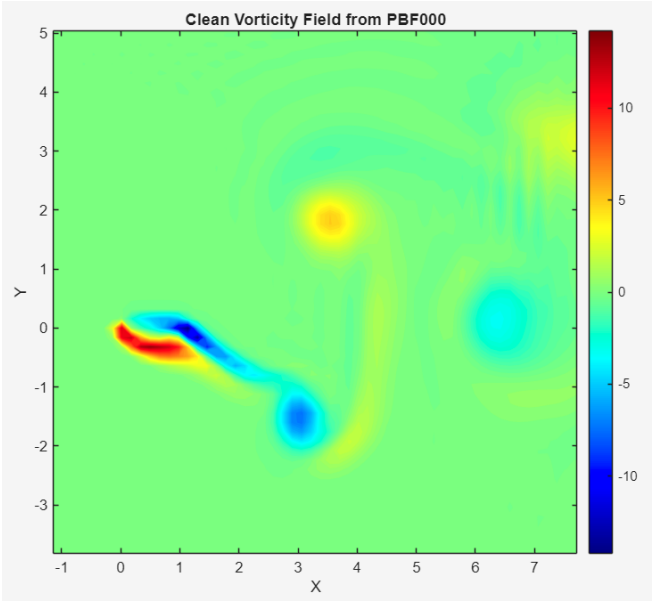
Figure 2: Comparison of the development of the pitch and heave motion describes by the algorithm throughout the optimization process.



(a) Initial drag waveform



(b) Initial lift waveform



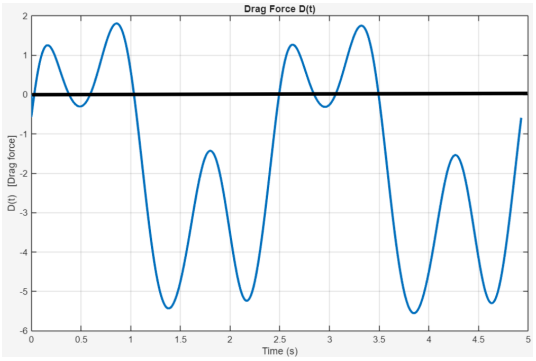
(c) Vorticity field for initial swim gait after a single pass at 1 Fourier mode.

Figure 3: The initial waveform outputs for the swimming gait efficiency optimization.

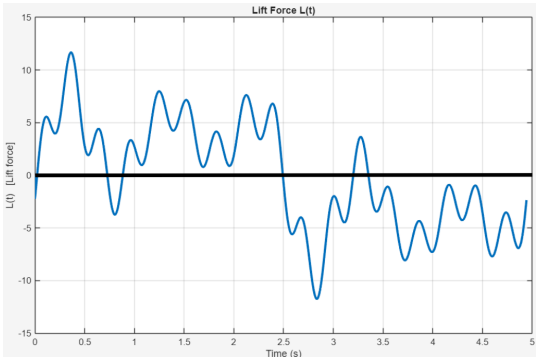
The initial drag and lift values are as expected, and the vorticity field clearly displays the simple swimming pattern giving off single trailing edge vortices with each stroke. Take note of the minimum drag and the absolute value of the lift force in both directions. Decreasing average drag and increasing lift force magnitude are indicators of more efficient swimmers.

It should be noted here that the vorticity field shown in figure 3 is not a representation of the initial guess, but rather is the visualization the optimization algorithm created after the system's first pass through a minimal length optimization cycle at 1 Fourier mode. This is because the algorithm does not generate vorticity fields until it has completed an optimization run. Thus the given vorticity field is a good representation the swimming behavior we should see given the initial conditions, but is not

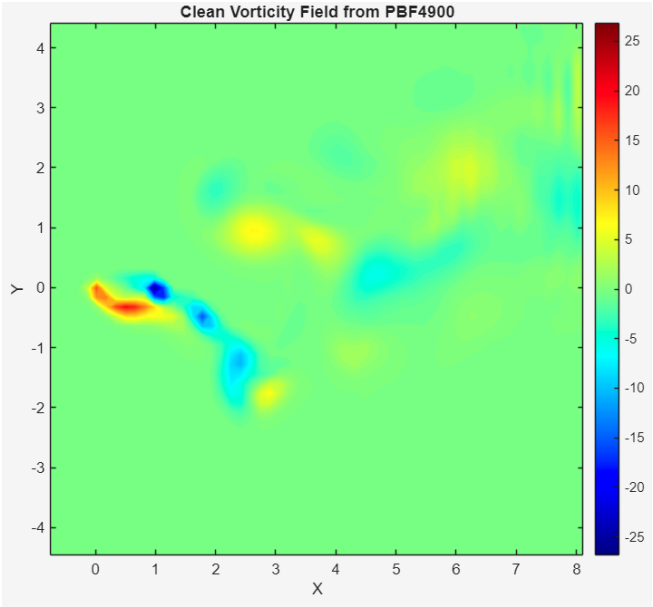
perfectly exact due to slight modifications the optimization process applied.



(a) Final optimized drag waveform



(b) Final optimized lift waveform

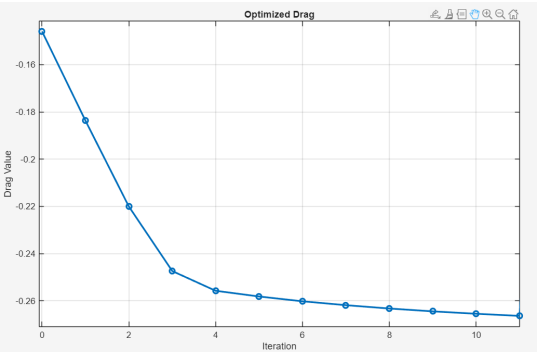


(c) Final vorticity field for optimized swimming gait.

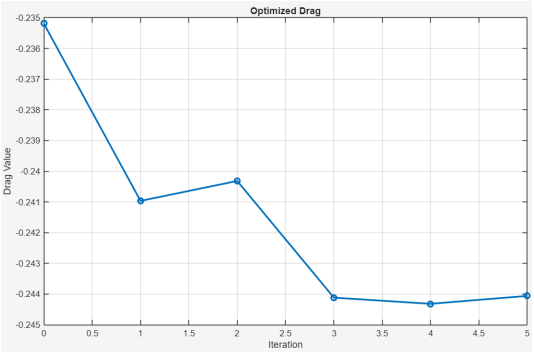
Figure 4: The final waveform outputs for the swimming gait efficiency optimization after full completion, up to 30 Fourier mode analyses.

Comparison of figure 4 to figure 3 shows a stark change. The average drag is much lower and the lift force has a much higher absolute magnitude. These values show that the optimized swimming gait generates more lift with less drag than before, increasing the efficiency by approximately 10 times.

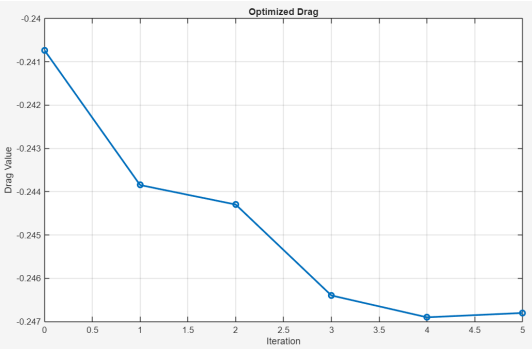
Additionally, it can be noted that drag and lift have significantly less sinusoidal forms, and the vorticity graph demonstrates each tail stroke generates 3 vortices in its wake. As the algorithm worked, it found unique swimming patterns that apparently take advantage of minute vortex dynamics to create a more efficient swimming gait in this triple-vortex fashion.



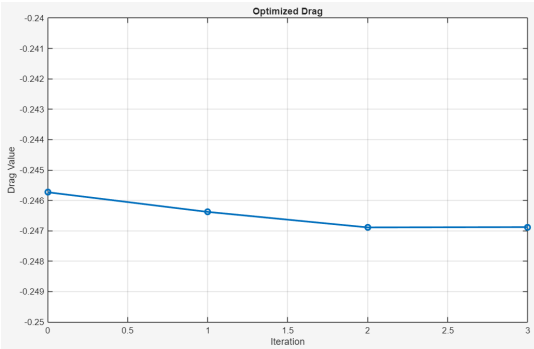
(a) Drag over time throughout 1 Fourier mode run.



(b) Drag over time throughout 5 Fourier mode run.



(c) Drag over time throughout 10 Fourier mode run.



(d) Drag over time throughout 30 Fourier mode run.

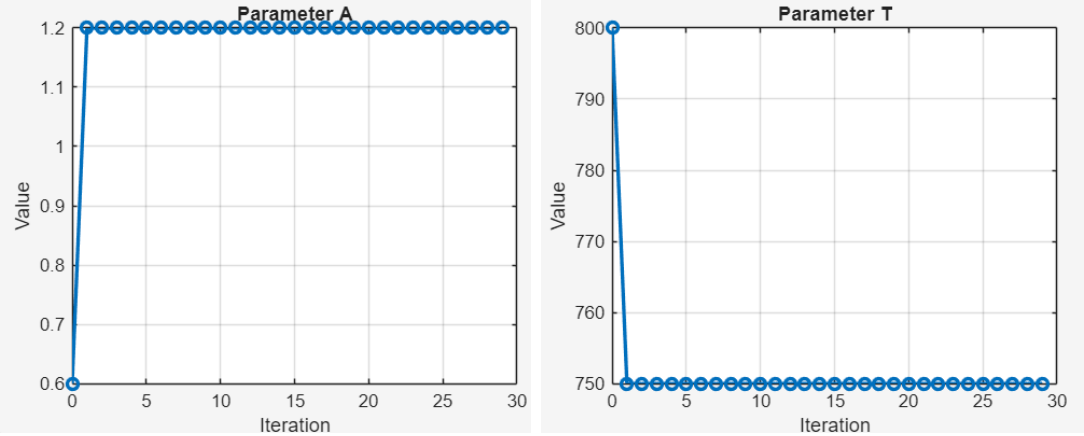
Figure 5: Graphs of how the drag changes throughout the course of each run through the optimization algorithm. Note how the majority of the optimization occurs in the 1 Fourier mode system, and very little occurs by the time we run at 30 Fourier modes.

The output parameters of each run are used as the input parameters for the next run at a higher Fourier mode, with the initial guess being used as the input of the 1 Fourier mode optimization. Each run stops when the algorithm detects that the drag value between two iterations stays within a certain threshold (i.e. the system is fully optimized at that number of Fourier modes). In figure 5 (a), it can be seen that a lot of optimization occurs, while each consecutive run has less and less change in drag (note the change in y-axis scale). In (d), the swim gait barely changes at all, and the algorithm only completes 3 iterations before finishing.

4.2 Thrust Optimization Data

Again, the purpose of the thrust optimization test was to make sure the algorithm would break in the expected way. Pure thrust is an unbounded equation, and optimizing for thrust with no limit on energy costs results in a swimming gait that tries to beat infinitely fast. When we ran the 1 Fourier mode test (with the same initial conditions

given to the efficiency optimization test), all of the relevant parameters immediately spiked to their hard-coded maximum or minimum values in a single iteration. Two of these parameters are sampled in figure 6 as examples.



(a) Heave amplitude parameter A.

(b) Period parameter T.

Figure 6: Graphs of how the parameters shown instantly spike in the thrust optimization test. Parameters A and T immediately jump to their manually set limits, upper bound 1.2 and lower bound 750 respectively. This makes sense, because the algorithm attempts to maximize amplitude and minimize period.

5 Discussion

5.1 Data Analysis

First and foremost, it can be stated that the optimization algorithm effectively increased lift while decreasing drag, resulting in a more efficient swim gait as a final result. By dipping into complex fluid mechanics, the optimized gait seems to swim more efficiently with its triple-vortex trail. The algorithm found and engaged specific vortex interactions that improved over the original guess by roughly 10 times.

The algorithm could be run at a higher resolution, and given a project with a longer timeline, there is clear value in rerunning the algorithm for 100+ hour run times at maximum resolution. We would not expect the final result to be vastly different than the rougher simulations, but a more fine grid may allow the optimization process to take advantage of similar vortex interactions to an even greater degree.

One very interesting detail is how much of the optimization process occurred in the first run at 1 Fourier mode. Each run at higher Fourier modes subsequently had less and less of an impact on the drag value, meaning change in efficiency was minor (cost is held constant, so as drag decreases it directly correlates to higher efficiency). The 30 Fourier mode run, despite taking the longest, only finished 3 iterations before

completing and had little effect on the swim gait it received from the 10 Fourier mode run. Arguably it had no effect at all.

There are a number of possible explanations behind this, which require further testing to properly understand. The most obvious answer is that for the given problem, 10 Fourier modes is enough depth to properly optimize the system, and increasing to 30 is mostly unnecessary. This may be the case for the specific parameters and settings we ran for this efficiency optimization, or it may be a general rule for this kind of application. Another possibility is that the resolution of the simulation itself was too low, so the 30 Fourier mode run did not properly engage. Like using a fine comb to sift through coarse sand too large to fit through the teeth, no meaningful actions happened here. Or it could be something else entirely. Further testing is required here to properly characterize the rates of optimization between different amounts of Fourier modes. However, for the purposes of this report, we will move forward under the assumption that the output from the 10 Fourier mode run was simply already fully optimized to the same level as the 30 Fourier mode can.

5.2 Literature Review and Biomechanics Comparison

We've established that the optimization algorithm does indeed take a simple initial swimming gait and produce a complex, optimized swimming gait that is several times more efficient. However, this does not mean much unless we can compare this to the biomechanical analyses of actual fish swim gaits that are used in robotics applications. In other words, is our swim gait better than the robotic swim gaits being modeled off of real fish?

The model fish being simulated by the optimization algorithm consists of a single caudal fin experiencing an oncoming freestream of incompressible fluid. Due to the extreme computational costs of simulating complex geometries in CFDs multiplying against the computational costs of optimization systems, our model has no body, and the caudal fin itself is fully rigid. While it does an excellent job allowing the optimization process to be physically computable, this is, pretty clearly, not a very accurate representation of a real fish or fish-inspired robot [7, 8, 9]. This means that in order to get a meaningful comparison between our optimized swim gait and existing swim gaits, we must make a few key assumptions.

First of all, real fish swim in a broad variety of ways determined by their morphology [8, 10]. On extreme end of the spectrum, anguilliform swimmers like eels undulate their entire bodies to swim. Towards the middle, carangiform swimmers like salmon involve their rear bodies and tails in their caudal swimming motion. On the other end, ostraciiform swimmers like puffer fish only use their tails, with no engagement from the body at all.

The project was originally designed to simulate Body and Caudal Fin (BCF) swimmers like thunniform tuna [11] who engage their tail and a very small amount of rear body in the caudal gait, but given the fact that our model is just a tail with no body at all, we decided that the swimming gaits of ostraciiform swimmers would be a more

relevant comparison [12]. It's not perfect, because the pectoral and dorsal fins, as well as the body itself, are not being simulated. However, for the purposes of specifically optimizing the swimming gait of just the caudal fin for pure efficiency, the comparison holds some weight.

Additionally, fish do not have rigid fins. When swimming, different fish will flex their caudal fins into various geometric structures depending on what is required in the situation [7]. Our model is fully rigid and engages in none of this. However, most bio-inspired swimming robots do have rigid fins due to material limitations [9]. Regardless, it should be remembered that a possible source of discrepancy between the swimming behavior our model employs and the swimming behavior of real fish is this caudal flexibility (or lack thereof).

One more thing to consider before we can begin choosing which biomechanical data to compare against our model: not all fish swim in a generally sinusoidal pattern. A commonly employed swim gait by BCF swimmers is the burst and coast method, where a single stroke of the tail is followed by a period of passive coasting [5]. Unlike our continuous model of tail movement, burst-coast swimming minimizes active time in the gait.

However, burst-coast swimming is not always more efficient. In fact, as can be seen in Han, Wang, & Dong [6], continuous swimming gaits are more efficient at lower duty cycles. Below a certain threshold, it is better to swim continuously. Thus there is an application for our optimized continuous swimming model: low speed, high efficiency swimming in non-traditional environments where a tail is more apt than a propellor [1, 2].

The paper we decided to use as a biomechanical basis for comparison against our optimized swim gait is Coe & Gutschmidt [8]. They match the established criteria for a valid comparison, and most importantly they calculate the coefficients of drag and lift using dimensionless coefficients of the form:

$$C_x = \frac{F_x}{0.5\,\rho U^2 L} \tag{10}$$

while our algorithm calculates those same coefficients with:

$$C_T = \frac{F_x}{0.5\,\rho U_\infty^2 c} \tag{11}$$

These equations are mostly identical, except Coe & Gutschmidt use the body length of the model fish as L while we used chord length as c. Luckily, both equations are normalized L and c to 1 respectively for model testing, meaning the coefficients in the data can be compared directly 1:1.

The data Coe & Gutschmidt provide is very comparable to the format in which our data is calculated, but the important data in their paper lies within figures 25 and 28, not shown here for copyright reasons [8].

In comparing our data, we can see that our initial conditions are actually slightly worse in terms of drag and lift compared to what Coe & Gutschmidt modeled for

actual ostraciiform fish, but after the 30 Fourier mode optimization process, drag has been decreased by over 10x and lift has been increased by almost 10x, showing a clear improvement in terms of maximizing efficiency. Thus, when compared to a valid model of fish biomechanics, the optimized swimming gait demonstrates definitive advantages over natural gaits in certain situations. This project acts as a proof of concept of the value and efficacy of using adjoint-based optimization algorithms to optimize natural systems for specific scenarios and values.

5.3 Future Steps

Moving forward, the adjoint-based optimization algorithm could use a variety of deeper characterization testing, both with the current parameters and at higher resolutions. Additionally, physical testing might be very beneficial. Programming various swimming gaits into robotic BCF swimmers could provide valuable insight into the efficacy of the optimization algorithm. Also, performing our own biomechanical analysis on actual fish would allow us to perform comparisons with a much higher degree of certainty in the number of constrained variables. Using Coe & Gutschmidt's work has just been a preliminary comparison, and should be taken as such.

6 Acknowledgments

The author appreciates the support of the National Science Foundation's REU Program and Kent State University's College of Aeronautics & Engineering.

References

[1] Jennifer A. Cardenas, Zahra Samadikhoshkho, Ateeq Ur Rehman, Alexander U. Valle-Pérez, Elena Herrera-Ponce de León, Charlotte A.E. Hauser, Eric M. Feron, and Rafiq Ahmad. A systematic review of robotic efficacy in coral reef monitoring techniques. *Marine Pollution Bulletin*, 202:116273, 2024.

[2] Benjamin Peter, Roman Ratnaweera, Wolfgang Fischer, Cedric Pradalier, and Roland Siegwart. Design and evaluation of a fin-based underwater propulsion system. *Proceedings - IEEE International Conference on Robotics and Automation*, pages 3751–3756, 05 2010.

[3] Kelly M Diamond, Raphaël Lagarde, J Gill Griner, Dominique Ponton, Kara E Powder, Heiko L Schoenfuss, Jeffrey A Walker, and Richard W Blob. Interactions among multiple selective pressures on the form–function relationship in insular stream fishes. *Biological Journal of the Linnean Society*, 134(3):557–567, 07 2021.

[4] Xuanhong An, Daniel Floryan, and Clarence W. Rowley. Optimization of swimming gaits. *AIAA SCITECH 2022 Forum*, 2022.

[5] Emre Akoz and Keith Moored. Unsteady propulsion by an intermittent swimming gait. *Journal of Fluid Mechanics*, 834, 03 2017.

[6] Pan Han, Junshi Wang, and Haibo Dong. Effects of intermittent swimming gait in fish-like locomotion. *AIAA Scitech 2020 Forum*, 2020.

[7] Brooke Flammang and George Lauder. Caudal fin shape modulation and control during acceleration, braking and backing maneuvers in bluegill sunfish, lepomis macrochirus. *The Journal of experimental biology*, 212:277–86, 02 2009.

[8] Michael Coe and Stefanie Gutschmidt. Ika-flow : A flexible body overset mesh implementation for fish swimming. *OpenFOAM® Journal*, 3:75–119, 07 2023.

[9] Christopher Esposito, James Tangorra, Brooke Flammang, and George Lauder. A robotic fish caudal fin: Effects of stiffness and motor program on locomotor performance. *The Journal of experimental biology*, 215:56–67, 01 2012.

[10] Eric Tytell, Iman Borazjani, Fotis Sotiropoulos, T Baker, Erik Anderson, and George Lauder. Disentangling the functional roles of morphology and motion in the swimming of fish. *Integrative and comparative biology*, 50:1140–54, 12 2010.

[11] Keith Korsmeyer, John Steffensen, and Jannik Herskin. Energetics of median and paired fin swimming, body and caudal fin swimming, and gait transition in parrotfish (scarus schlegeli) and triggerfish (rhinecanthus aculeatus). *The Journal of experimental biology*, 205:1253–63, 06 2002.

[12] J Hove, L O'Bryan, Malcolm Gordon, P Webb, and Daniel Weihs. Boxfishes (teleostei: Ostraciidae) as a model system for fishes swimming with many fins: Kinematics. *The Journal of experimental biology*, 204:1459–71, 05 2001.