OpenFOAM® Journal

Volume 3 [Full Papers], Pages 75–119
ISSN: 2753-8168

IKA-FLOW : A Flexible Body Overset Mesh Implementation for Fish Swimming

Michael Coe[1], * and Stefanie Gutschmidt[1]

[1]University of Canterbury, Private Bag 4800, Christchurch 8140, New Zealand
Email address: michael.coe@canterbury.ac.nz

DOI: https://doi.org/10.51560/ofj.v3.89
Results with version(s): OpenFOAM® v2206
Repository: https://github.com/michaelcoe/Ika-Flow

Abstract. Simulation of inertial aquatic swimmers requires fluid structure interactions with temporal body geometry deformation. In practice, this results in a change of the computational domain boundaries that represent the "swimmer." These simulations are traditionally done using body-fitted mesh and mesh morphing methods, but have drawbacks of negative cell volumes and small time-steps to account for the complex swimming motion. In contrast, the overset mesh method, also provided by OpenFOAM®, overcomes most of the drawbacks of the mesh morphing method at the expense of interpolation error. The current OpenFOAM® overset motion library only supports rigid body motion and cannot be used to resolve a body undergoing undulation. A modified motion solver is presented that allows for the complex mesh motion of an overset mesh for four body-caudal fin (BCF) virtual swimmers. The results of this solver are compared with published data of body-fitted meshes. The effect of different simulation parameters (including number of solving iterations, time delay, and temporal resolution) is investigated. Additionally, a novel simulation and comparison of the Ostraciiform locomotion mode with Anguilliform, Carangiform, and Thunniform modes are made investigating the wake, drag and lift. It is concluded that fish undulation has a marked effect on reducing lift generation. Lastly, a comparison of turbulence models (Spalart-Allmaras, $k-\omega$ SST, and $k-k_L-\omega$) at multiple Reynolds numbers shows that all three models have similar performance at lower Reynolds numbers but diverge at higher numbers.

1. Introduction

The efficiency and mechanics of fish motion have been a long standing topic of study in the science and engineering communities since the time of Aristotle [1]. A more formal and recent mechanical analysis of locomotion energies was started following the work of Breder [2] and Gray [3]. The latter work became known as Gray's paradox which states that the amount of power a dolphin could exert based on its muscle mass would not be able to overcome water drag forces. Since then, many scientists have worked on models for fish energetics, with a significant contribution being the "elongated-body theory" developed by Lighthill [4, 5].

There have been a number of numerical studies of swimming fish using analytical and potential flow models that extend Lighthill's theory [6–9]. As Lighthill suggested [10], a promising three-dimensional theory to analytically solve for fish energetics is lifting line theory. Here the local flow around each cross-section is made two-dimensional but considers the local angle of attack and wake vorticity. This has led to scientists utilizing vortex panel methods to model fish motion [11–13]. With the advent of Computational Fluid Dynamics (CFD), there have been a number of both two- and three-dimensional studies of fish locomotion and energetics [14–21]. These studies deal with Anguilliform and Carangiform locomotion, while none model Ostraciiform and few address Thunniform swimming [22]. Furthermore, these studies are generally limited to the laminar flow regime due to most aquatic animals operating within this region [23]. There are, however, aquatic animals and manmade aquatic robots regularly operated at larger Reynolds numbers [24, 25] as shown in Fig. 1.

A common way to analyze fish locomotion using CFD is to model the fish body as a National Advisory Committee for Aeronautics (NACA) airfoil. There have been many studies of NACA airfoil using overset meshes and specifically OpenFOAM® implementation [26–31]. These studies don't include the airfoil's undulation and are focused on the airfoil being static or undergoing rigid body motion. In the

* Corresponding author

Received: 4 August 2022, Accepted: 2 June 2023, Published: 7 July 2023

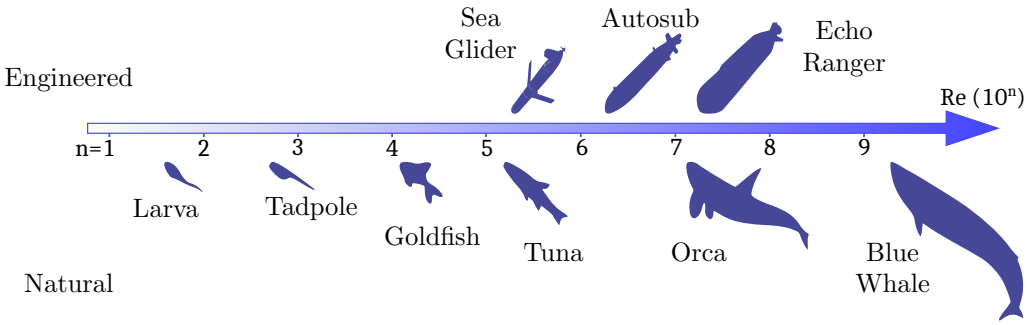


Figure 1. Aquatic animals and autonomous underwater robots at different Reynolds scales. Modified from [25].

study of undulating airfoils with overset meshes, there are a few studies [32–34], but none performed with OpenFOAM®, with these works confined to commercial CFD codes, such as Ansys Fluent [35] or Star-CCM+ [36]. This work seeks to utilize OpenFOAM® and a new overset motion library as a prerequisite step in modeling fish locomotion at a range of Reynolds numbers from $5 \cdot 10^3$ to $4 \cdot 10^7$.

When using overset mesh, errors are introduced into the simulations that must be accounted for. As pointed out by Chandar [31], these errors are caused by the interpolation of the overset mesh to the background mesh when compared to a single mesh, but the solutions are comparable to Arbitrary Mesh Interface and Generalized Grid Interface techniques [30, 31]. The interpolation error is dependent on the interpolation method with the inverse distance method having the highest error and the polynomial interpolant having the least [31].

This paper describes how the existing solid body motion solver from OpenFOAM® Computational Fluid Dynamics library (CFD) is rewritten for use with an overset grid. The current overset library only supports rigid body motion of an object and not the complex motion needed to resolve fish locomotion. Besides fish motion, this library can be used to deform an overset mesh to accommodate any motion desired. Overset grids, or chimera grids, are CFD meshes that have more than one computational domain joined together by an interpolation region, explained in more detail in section 4. Since the overset region can move independently from the background mesh, it allows for more complex motion without the use of a diffusion equation to solve for mesh points at each time step. This allows for complex mesh motion without running into negative cell volumes especially when running at larger temporal time steps.

The studies presented in this work share two things in common: they utilize body-fitted meshes for their simulations and focus on low Reynolds numbers. The highest Reynolds number used in a fish simulation is 50,000 by Yu *et al.* [14]. This work is integrated into a broader research project that aims to model fish locomotion utilizing an overset mesh, and to determine forces on different fish locomotions at several decades of Reynolds numbers: $5 \cdot 10^3$ to $1 \cdot 10^8$. This paper focuses on the first goal of this study, which is validating an overset mesh solver for fish motion for OpenFOAM®.

In this research, four body-caudal fin (BCF) locomotion modes are implemented in an overset motion library called Ika-Flow. This paper is organized in the following way: section 2 explains how fish locomotion is classified and differs from propeller motion. section 3 gives a mathematical description of the four locomotion modes in this research. section 4 reviews overset grids and section 5 is a description of the computational domain. section 6 describes how the OpenFOAM® solidBodyMotion solver is modified to simulate fish motion. section 7 and section 8 describe the numerical schemes used and how forces, power, and efficiencies are calculated. section 9 shows the effect of varying simulation parameters (PIMPLE iterations, instituting a time delay, max Courant number) on lift and drag. Finally, an investigation is provided in subsection 9.6 investigates different turbulence models (Spalart-Allmaras, $k-\omega$ SST. $k-k_L-\omega$) in the laminar region and the transition region for a Carangiform swimmer.

2. Fish Motion

Fish motion was reviewed in 1999 by Sfakiotakis *et al.* [37] and the hydrodynamics of fish compared to propeller propulsion was reviewed in 2011 by Govardhan and Arakeri [24]. Fish are categorized into two different swimming modes: Body-Caudal Fin (BCF) and Median-paired Fin (MPF). To understand the reason for this naming, the general anatomy of fish is presented in Fig. 2.

Momentum transfer from the fish to the surrounding fluid is either done by the undulation of the entire body (BCF) or by using only fins (MPF) which leads to the classification of the two swimming modes. Each mode is further separated into sub-swimming modes depending on the body or fin motion from

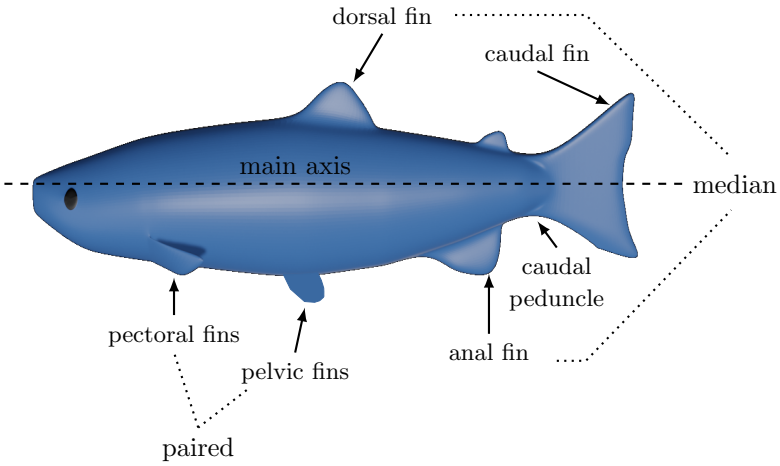


Figure 2. Sketch of the anatomy of a fish showing the terminology in the text for fins and other features. Adapted from [37].

undulation to pure oscillation, as shown in Fig. 3. Not depicted in Fig. 3 are lift-based and jet swimming modes. Lift-based swimming is the locomotion mode (e.g. turtles and sea lions) is one in which their flippers move the fluid past their bodies to create lift [38]. Jet swimming (e.g. jellyfish and cuttlefish) is when a part of the body fills with fluid and ejects it out of a nozzle for thrust [24].

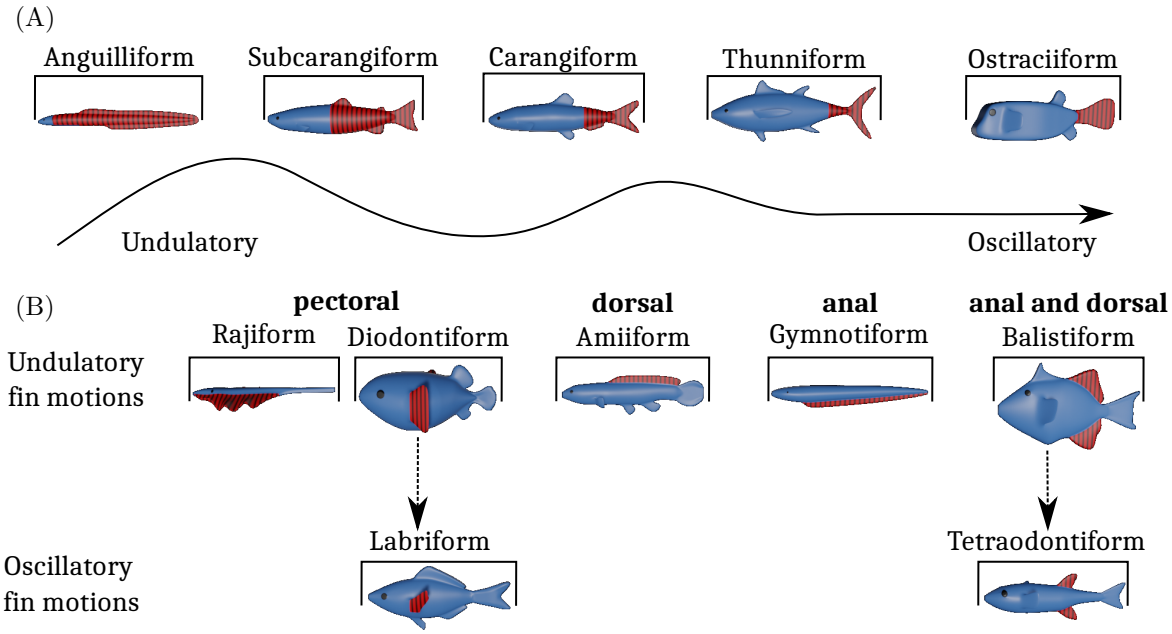


Figure 3. Locomotion modes of fish Swimming. Top row (A) represents BCF swimmers and bottom two rows (B) represent MPF propulsion. Red and black shaded areas show areas that contribute to thrust generation. Adapted from [1].

This research focuses on the BCF locomotion modes with the fish body modeled as a NACA 0012 airfoil. To understand the difference between the different types of BCF swimming modes, a visual comparison of the BCF swimmers is given in Fig. 4. BCF propulsion is characterized by a undulating

wave that travels down the body. The difference between the different BCF swim modes is the amount of the body that contributes to propulsion. The far left of Fig. 4 is the Anguilliform mode. This mode has the whole body undulating versus the Ostraciiform on the far right of Fig. 4, where only the last 15% of the body oscillates. The modes become increasingly oscillatory from left to right.

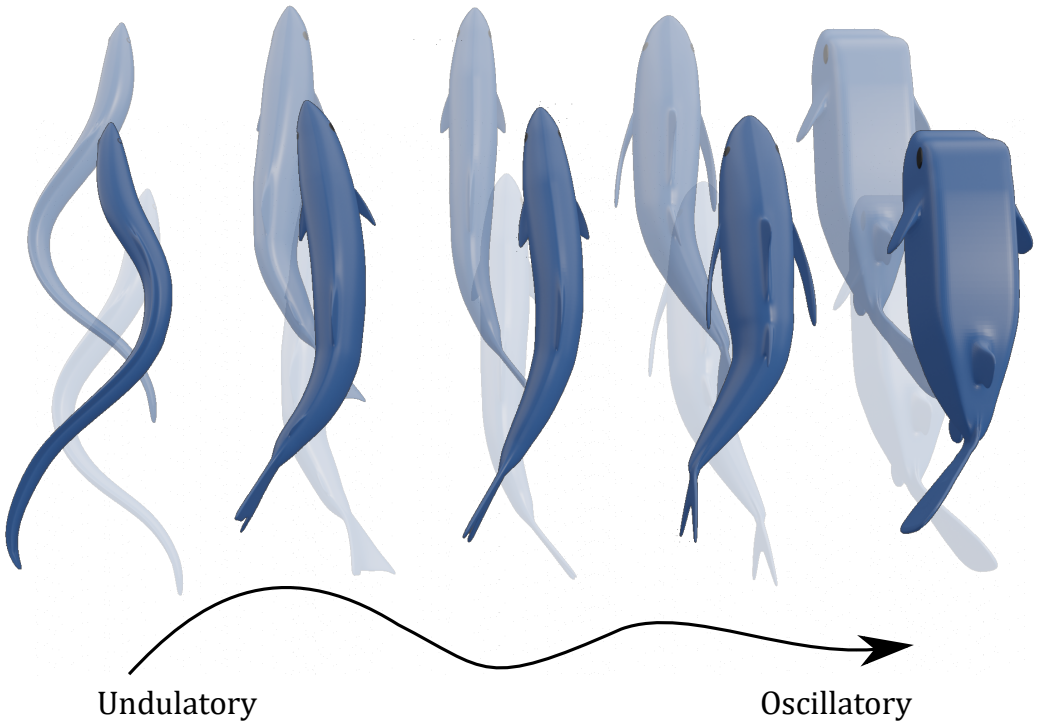


Figure 4. Comparison of one cycle of the all BCF swimmers.

Compared to conventional propeller propulsion, body undulation propulsion differs in how thrust is produced and how wake vortices are formed in the wake. Fig. 5 shows the vortices in the wake of both swimmers and how the shear layer forms across their bodies. Fish generate thrust both laterally (perpendicular to the body) and tangentially (in the line of propagation) to the body as they undulate. As opposed to fish motion, propeller thrust is only generated in one direction with the wake shed as helical vortices equal to the number of blades instead of a reverse Kármán street. The detailed review of both types of propulsion is given by Govardhan and Arakeri [24] and summarized in the following paragraph.

For a self propelled body using a propeller, the work done by the motor, $(\tau\omega)$, where τ is torque of the motor, in Nm, and ω is the angular speed of rotation, in rev/s, is converted to thrust axially via a kinetic energy into the fluid. This energy flows into the wake in the form of viscous dissipation and heat [24]. As shown in Fig. 5, the trailing vortices from a propeller propagate from the tip of the propeller blades in a helical shape.

In contrast to a propeller driven vehicle, a biological animal, or bio-inspired vehicle, converts the work done by a tail into thrust at a angle normal to the tail movement. As the caudal peduncle moves side-to-side, a low pressure region develops at the posterior of the body that sucks in a bolus of liquid laterally. This bolus is shed by the tail and rolls up into two or more same sign vortices [40]. These vortices create a jet flow as the array of vortices move downstream from the tail. The array of vortices with alternating signs in the wake forms what is known as a reverse Kármán street [41]. Triantafyllou and Triantafyllou [42] performed experiments with an oscillating foil producing reverse Kármán streets in the presence of induced vortices. The group identified three operating modes of an oscillating foil in the presence of vortices, dependent on the timing of the oscillation:

(1) Vortices created by the foil meets the incoming vortices spinning in the opposite direction. This weakens the vortices in the wake meaning energy is captured and thus, causes an increase in efficiency.
(2) The induced vortices spin in the same direction as the oncoming vortices and reinforce one another. The two concluded that this had no practical application.

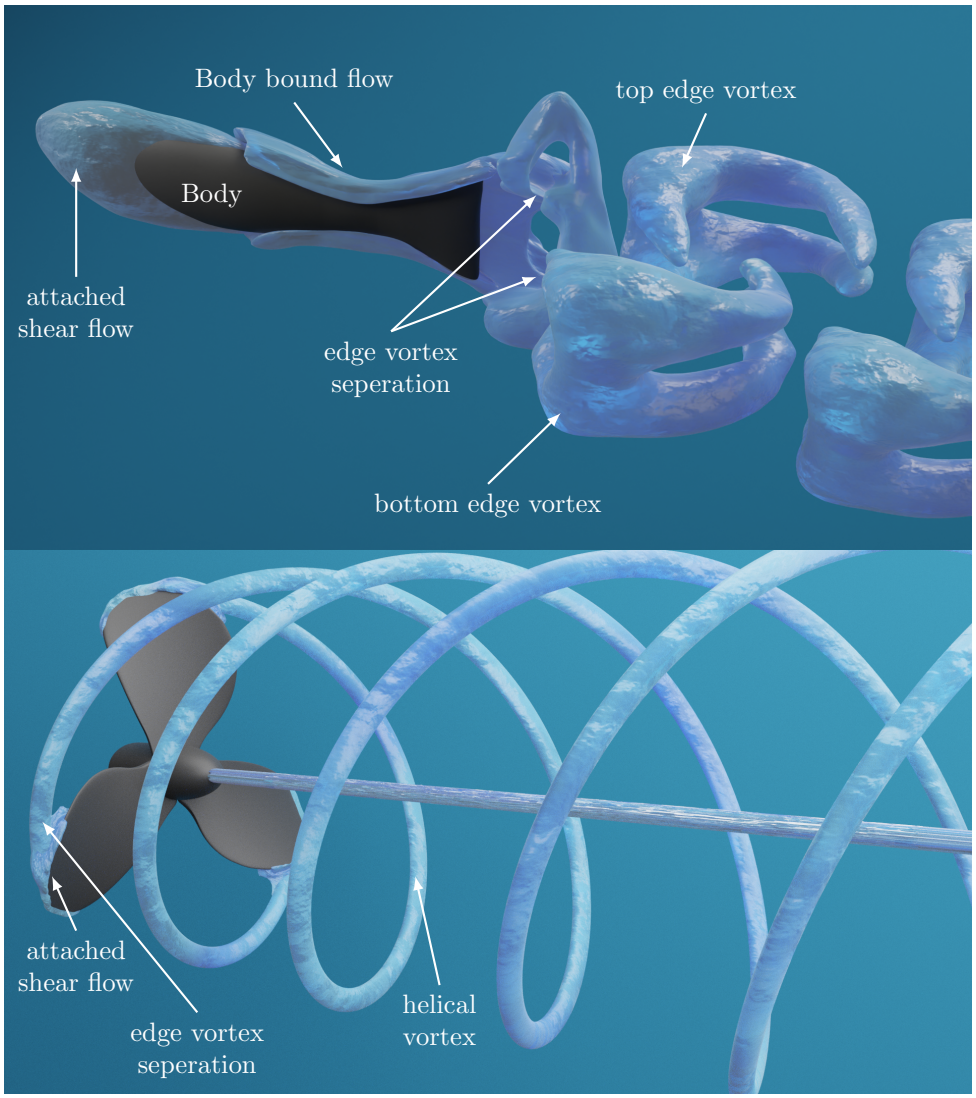


Figure 5. Wake profiles for fish (top) and propeller (bottom) during locomotion. Figure was generated using Blender [39].

(3) The last mode is that oscillations create a pair of counter rotating vortices which produce mushroom-shaped eddies aft of the swimmer, which would slow the swimmer down.

These three modes are relevant for being considered when comparing the wake of the four swimming modes modeled in this work. The total work by the fish is a summation of the viscous dissipation in the boundary layer of the body surface and the kinetic energy dissipated into the wake [24].

3. Mathematical Description of Fish Motion

The motion of the fish can be described as a prescribed deformation of the midline, shown in Fig. 4, in the transverse direction. Tytell [43] provides kinematic data for Anguilliform motion using high speed film and digital partical image velocimetry (DPIV) of American eels. Videler [44] proposed a much earlier model using the high speed film technique for subCarangiform and Carangiform swimmers. For Thunniform motion, Dewar [45] was able to perform a high speed film study on tuna.

In general, Videler and Nolet [44] describe the modeling BCF locomotion by fitting an envelope function using the least squares method with a backward traveling wave. The general form of this equation is

$$h\left(x,t\right)=\left(c_0+c_1x+c_2x^2\right)\sin\left(kx-\omega t\right), \tag{1}$$

where c_0, c_1, and c_2 are the least squares fit coefficients. x is the coordinate down the fish midline from the snout to the tail, k is the wave number that describes the amplitude of undulation given by

$$k = \frac{2\pi}{\lambda}, \tag{2}$$

where λ is the wavelength, ω is the frequency of undulation or oscillation, given by

$$\omega = \frac{2\pi V}{\lambda}, \tag{3}$$

where V is the body wave speed, and t is time. A simple way to fit these coefficients to an envelope are presented by Xia [46], by formulating the coefficients into a matrix and multiplying by its inverse, which is described as follows:

$$\begin{bmatrix} 1 & x_h & (x_h)^2 \\ 1 & x_c & (x_c)^2 \\ 1 & x_t & (x_t)^2 \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ c_2 \end{bmatrix} = \begin{bmatrix} A_h \\ A_c \\ A_t \end{bmatrix}, \tag{4}$$

where A_h is the the amplitude of the snout, A_c is the amplitude of the mass center, and A_t is the tail amplitude at the peduncle. The location of the snout is x_h, the location of the mass center is x_c, and the location of the tail peduncle as x_t. The coefficients of Eqn. 1 can be obtained by solving the set of linear equations, or by inverse matrix multiplication. Different literature suggest different coefficients for the envelopes depending on the motion types.

Maertens *et al.* [15] proposed a different format for Eqn. 1 that takes the form:

$$h(x,t) = a\left(1 + c_1(x-1) + c_2(x^2-1)\right)\sin\left(kx - \omega t\right), \tag{5}$$

where a is half the peak-to-peak amplitude of the tail. This formulation allows for the adjustment of the tail amplitude without resolving the coefficients. Coefficients used in Eqn. 5 will be discussed in the following sections and Eqn. 5 will be used for each type of motion. For clarity, a shape function in terms of x is given as $A(x)$ which represents the amplitude function. In this form, each equation for the fish midline takes the form of

$$h(x,t) = A(x)\sin\left(kx - \omega t\right), \tag{6}$$

where k is defined in Eqn. 2 and ω is defined in Eqn. 3 and $A(x)$, the envelope function, is shown in Fig. 6a and compared for different locomotion modes in Fig. 7.

3.1. **Anguilliform Motion.** From the kinematic data given by Tytell [43], the amplitude for Anguilliform swimmers can be modeled as

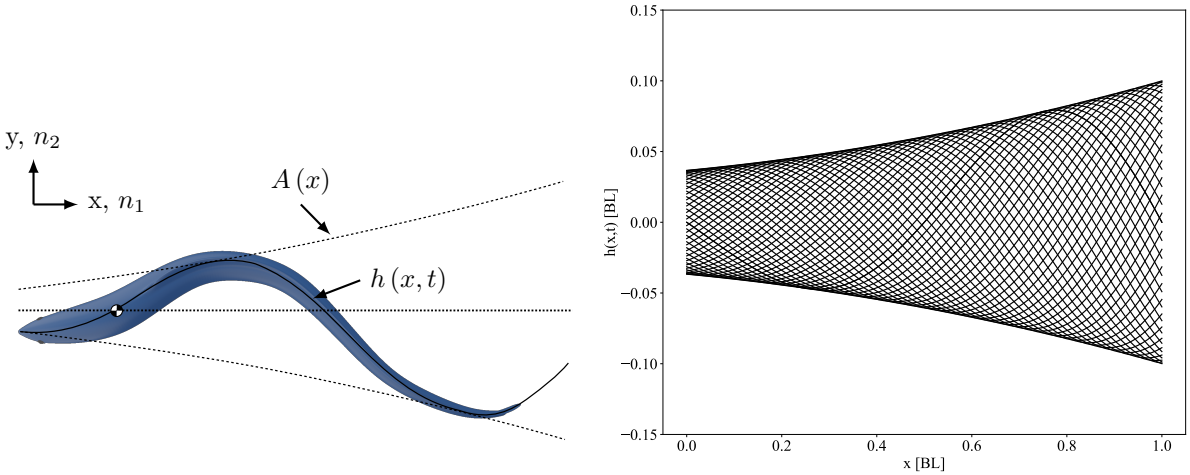
$$A(x) = ae^{\alpha\left(\frac{x}{L}-1\right)} \tag{7}$$

where a is the tail beat amplitude, α is the amplitude growth rate, x is the contour length down the midline of the swimmer starting at the head, and L is the body length. Here a large α gives a rapid increase in undulation towards the tail and a small α implies more undulation near the head. Fig. 6b shows the time progression of an Anguilliform swimmer's midline motion as defined in Eqn. 7.

The literature around numerical experiments of Anguilliform swimmers shows that a reasonable approximation for modeling Anguilliform swimming is to set A to 0.1 and α to 1 [14, 15, 17, 47, 48]. While only [15,17] used the data from Tytell directly for their simulations, the others performed a least squares fit on the envelope and adapted it to the polynomial envelope proposed by Videler and Nolet [44]. In the notation proposed by Maertens *et al.* [15], the coefficients from the least squares fit are

$$a = 0.1 \qquad c_1 = 0.323 \qquad c_2 = 0.310. \tag{8}$$

These coefficients are analogous to setting A to 0.1 and α to 1 and a comparison of the two amplitudes is provided in Fig. 7. There are two options available to model the amplitude of Anguilliform motion. The first is to use the Carangiform motion function and the coefficients in Eqn. 8. The second option is to use the Anguilliform motion function that implements the equation from Tytell. Translating these equations to an airfoil for visualization purposes is shown in Fig. 8.



(a) **(b)**

Figure 6. (A) Diagram showing the envelope and motion of an Anguilliform swimmer. (B) Midline motion of an Anguilliform swimmer at different times steps for a is set to 0.1, α is set to 1.

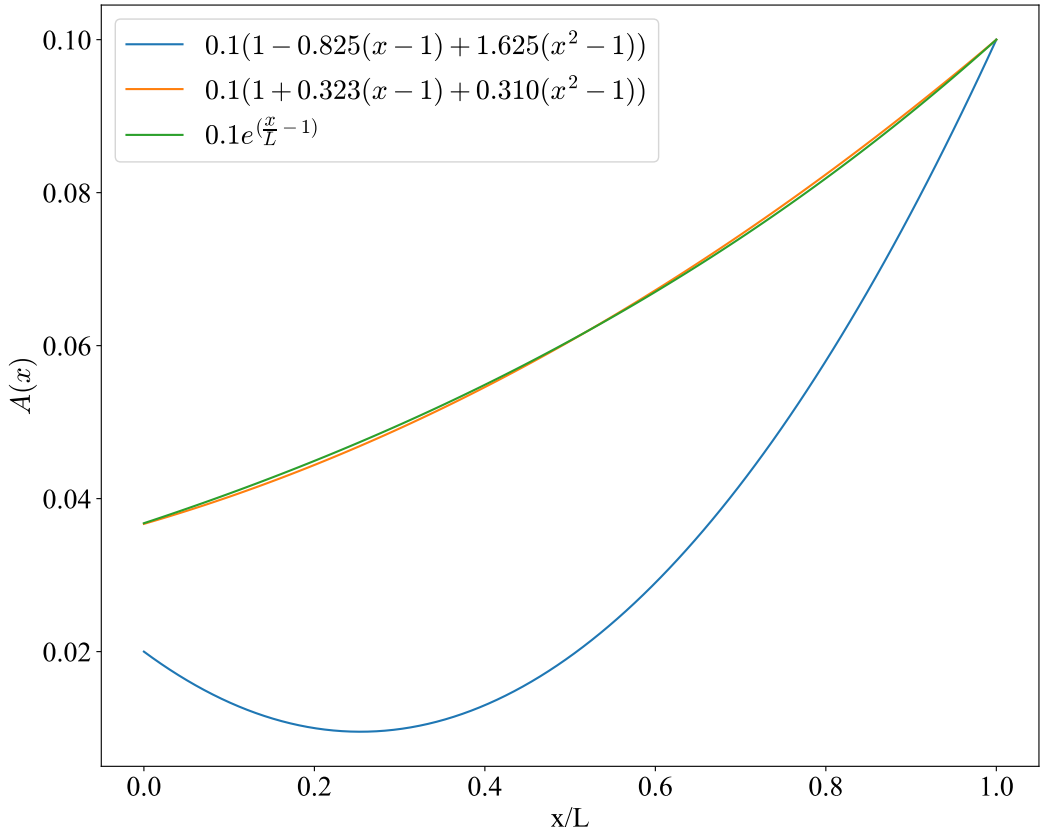


Figure 7. Comparison of shape functions for Anguilliform and Carangiform mode. Amplitude function given in legend.

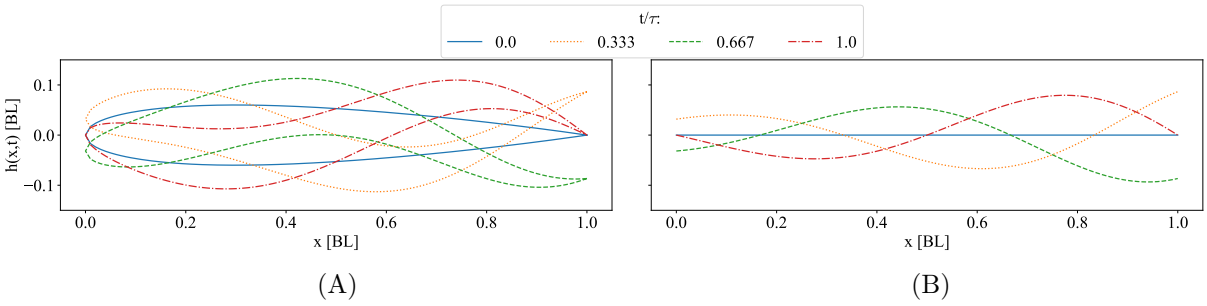


Figure 8. Visualization showing (A) shape of an airfoil swimming and (B) shape of the midline during Anguilliform motion. t/τ represents the time (t) over 1 tail beat (τ).

3.2. **Carangiform Motion.** Carangiform and sub-Carangiform motion is the most common form of BCF locomotion. This mode is characterized by the undulation of approximately 25 to 75% of the total body and caudal fin [4]. The amplitude envelope for sub-Carangiform and Carangiform swimmers was first given by Videler and Hess [49], using high speed images of fish swimming. The midline motion can be modeled as

$$A(x) = a\left(1 + c_1(x-1) + c_2(x^2-1)\right), \tag{9}$$

with the coefficients used for the amplitude envelope as

$$c_1 = -0.825 \qquad c_2 = 1.625. \tag{10}$$

In a majority of numerical studies for Carangiform motion, $\frac{\lambda}{L}$ is set to 1.0, where L is the length of the foil, and $\frac{V}{\lambda}$ are grouped into a frequency of undulation f with units Hz [14, 15, 17, 20, 50].

From data collected by Videler [23], the tail amplitude varies between $0.05L$ and $0.14L$ with an average value of $0.09L$. The average wave length of the fish λ is $0.92L$ with a standard deviation of $0.10L$. Fig. 9 shows the traveling motion wave and the midline motion at multiple time steps.

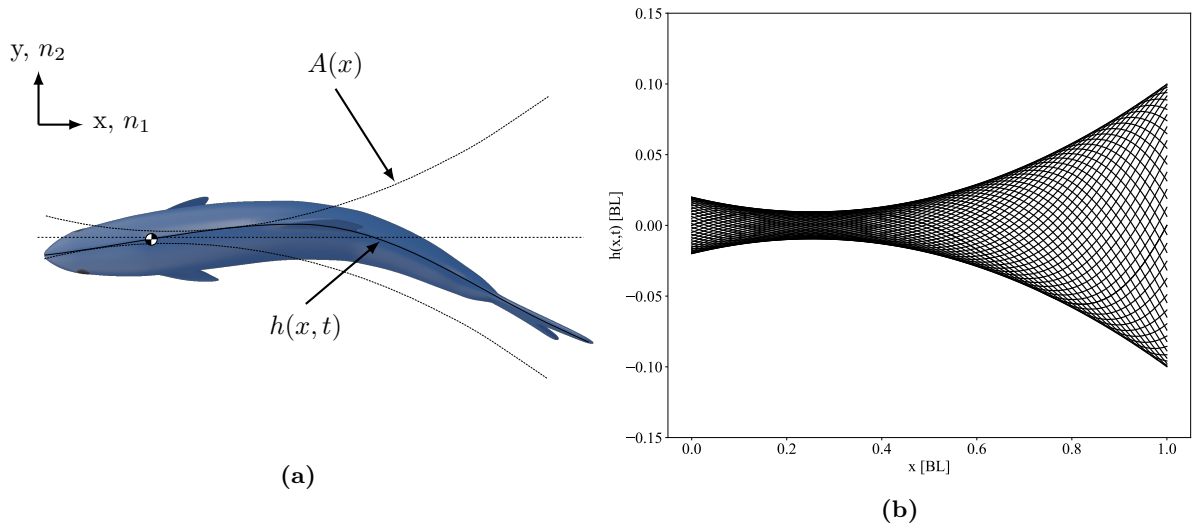
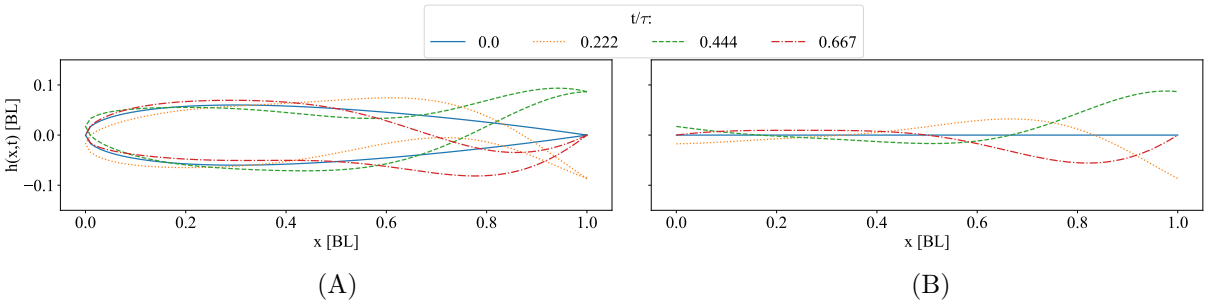


Figure 9. (A) Diagram showing the envelope and motion of an Carangiform swimmer. (B) Midline motion of an Carangiform swimmer at different times steps. a is 0.1 and coefficients of Eqn. 10

3.3. **Thunniform Motion.** Thunniform motion is widely considered the most efficient form of fish locomotion [50, 51]. This mode is characterized by 90% of the propulsive force being generated by the caudal fin, with the rest being generated by body undulations [50]. The caudal fin is modeled as pitching foil with angle $\phi(t)$, as done by Barrett [51] and Zhu *et al.* [52]. The body undulations are modeled the



(A) (B)

Figure 10. Visualization showing (A) shape of an airfoil swimming and (B) shape of the midline during Carangiform motion. t/τ represents the time (t) over 1 tail beat (τ).

same as Carangiform motion Eqn. 5, which makes the total body motion given by

$$h(x,t) = A(x)\sin(kx - \omega t), \tag{11}$$

$$\phi(t) = \phi_{max}\sin(kx_{pivot} - \omega t + \psi), \tag{12}$$

where c_1, c_2, k, and ω are defined the same as Eqn. 5 and Eqn. 10, ϕ_{max} is the maximum pitching angle, ψ is the phase angle, and x_{pivot} is defined by the user but represents the pivot point of the caudal fin normalized by the fish body length.

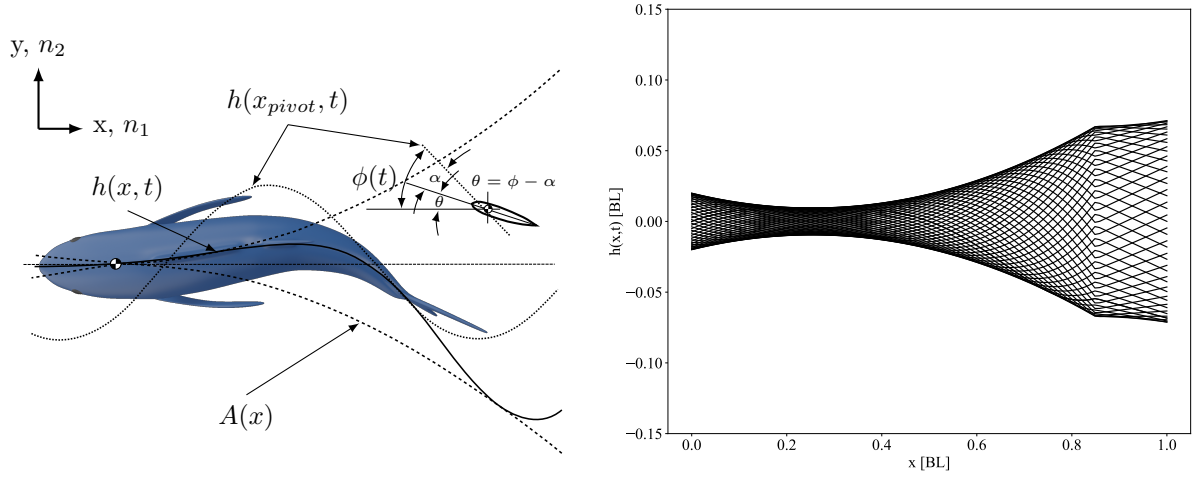
The $h(x_{pivot}, t)$ position is the lateral deflection of the pivot point and is given as

$$h(x_{pivot},t) = a(1 + c_1(x_{pivot} - 1) + c_2(x_{pivot}^2 - 1))\sin(kx_{pivot} - \omega t), \tag{13}$$

and the midline deflection of the caudal fin is given by

$$h(x_{foil},t) = h(x_{pivot},t) + x_{pivot}\tan\phi(t) \tag{14}$$

Fig. 11 shows the diagram of Thunniform motion and the midline displacement and a visualization of the airofil undergoing this type of motion is given in Fig. 12.



(a) **(b)**

Figure 11. (A) Diagram showing the envelope and motion of an Thunniform swimmer. The pitching foil is shown inset to the right. (B) Midline motion of an Thunniform swimmer at different times steps. a is 0.1 and coefficients of Eqn. 10

3.4. **Ostraciiform Motion.** Ostraciiform motion is characterized by a flapping of just the caudal fin. Fish using this locomotion mode normally utilize MPF propulsion, but it is popular among roboticists as it has an easily implemented mechanism for propulsion. The motion of the caudal fin is given in Eqn. 14 with the same parameters. To account for the motion of the head due to inertia, a slope function given by the lateral deflection of the head and pivot point is used. Here $h(x_{head}, t)$ and $h(x_{pivot}, t)$ are given by

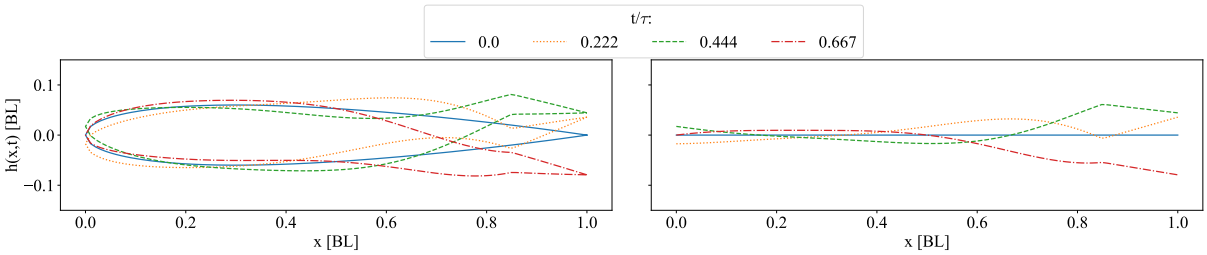


Figure 12. Visualization showing (A) shape of an airfoil swimming and (B) shape of the midline during Carangiform motion. t/τ represents the time (t) over 1 tail beat (τ).

$$h(x_{head} = 0, t) = A(0)\sin(0 - \omega t), \tag{15}$$

$$h(x_{pivot}, t) = A(x_{pivot})\sin(kx_{pivot} - \omega t), \tag{16}$$

where the lateral deflection of the body can be solved using the point slope form for a linear line

$$h(x_{body}, t) = \frac{h(x_{pivot}, t) - h(0, t)}{(x_{pivot})}x + h(0, t). \tag{17}$$

Fig. 13 shows the implementation of this model onto the Ostraciiform swimmer and Fig. 14 shows the deflection of the airfoil.

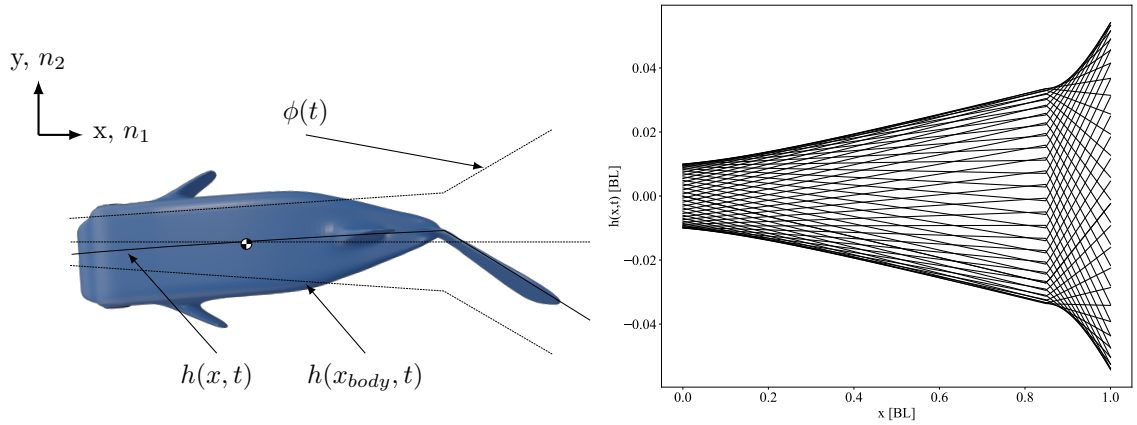


Figure 13. (A) Diagram showing the envelope for the body and tail. (B) Midline motion of an Ostraciiform swimmer at different times steps.

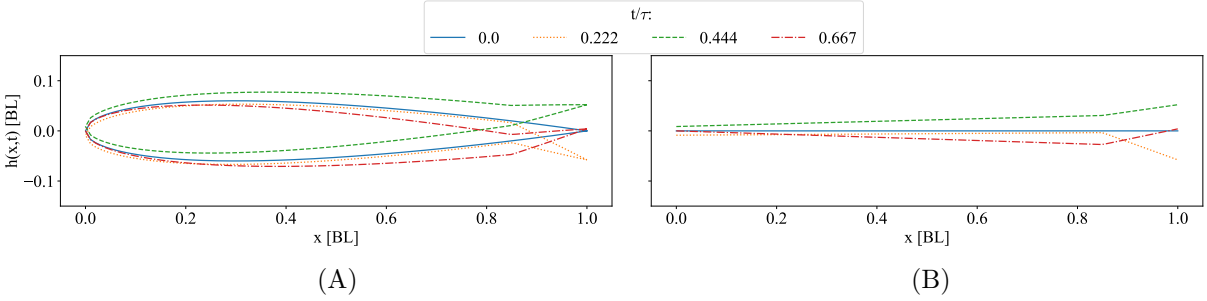


Figure 14. Visualization showing (A) shape of an airfoil swimming and (B) shape of the midline during Carangiform motion. t/τ represents the time (t) over 1 tail beat (τ).

4. **Overset Grid**

Much of the CFD simulations involving fish locomotion utilize a body fitted mesh [14–21]. This is a single mesh that conforms to the body of the swimmer and expands into the free stream. In contrast, this research utilizes an overset mesh. Overset meshes involve two or more meshes that are merged on top of each other [27, 28]. The meshes are considered as layers and a stencil is used to interpolate the values of one mesh to the other. An overview of the type of cells is given in the OpenFOAM® documentation [53] as follows:

- Donor cells: cells that provide values
- Acceptor cells: cells whose value gets set from interpolation
- Hole cells: inactive cells

Overset mesh was chosen because using body fitted meshes for patch motion are less numerically stable when it comes to complex fish motions [54], personal correspondence, and seen in other complex motions [28, 55]. The diffusion equation that solves for mesh motion causes negative volume where the meshes close to the swimmer overlap with neighboring meshes. The authors note that this could be achieved with point patch movement using a spring like motion solver or remeshing techniques as would be done in Fluent [29]. This research seeks to perform simulations without any extra tools and therefore an overset mesh is the best solution.

There are a number of interpolation methods that can be used to interpolate from one mesh to another. This research uses the inverse distance method for overset interpolation. Tisovska [56] gives a description of how the inverse distance method is implemented in OpenFOAM®. The procedure for interpolation first determines the weights to each cell center by summing the inverse distances as follows:

$$S = \sum_{i}^{n} \frac{1}{|d_i|}, \tag{18}$$

where n is the total number of donors and d_i is the distance from the donor cell center to the center of the acceptor cell. The weights for each cell can then be calculated using:

$$w_i = \frac{1}{d_i S}, \tag{19}$$

where w_i is the weight of the -ith cell on the acceptor cell. The interpolated value is then calculated using:

$$\phi = \sum_{i}^{n} = w_i \phi_i, \tag{20}$$

where ϕ is the field to be interpolated and n is the number of donors, as in Eqn. 18. Fig. 15 shows a schematic of this cell interpolation.

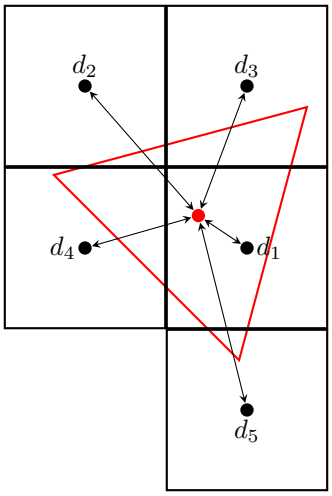


Figure 15. Schematic showing overset interpolation from the source grid (donor cells in black) to the destination grid (acceptor cell in red).

Fig. 16 shows the "airfoil" mesh with the overset cell types colored. Blue cells are calculated, white cells are interpolated, and red cells are holes.

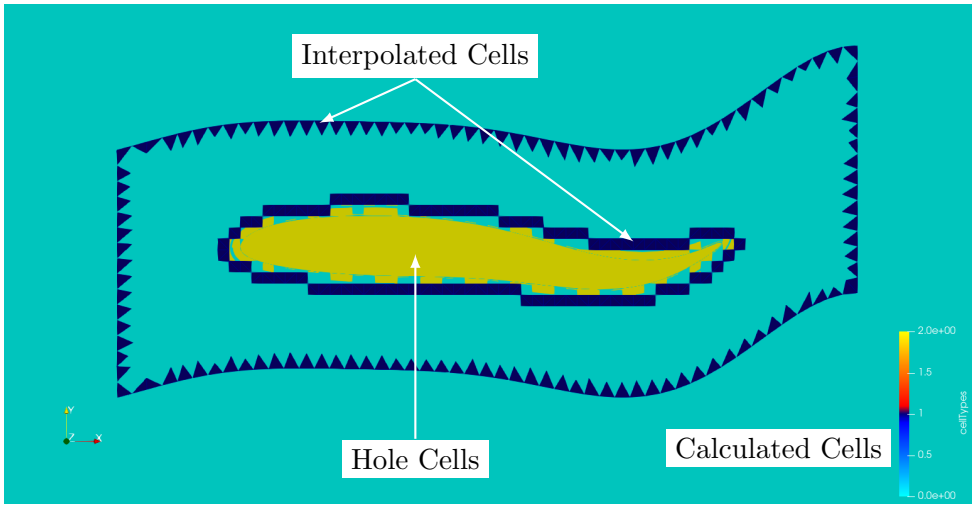


Figure 16. Overset mesh showing the types of cells as rendered by Paraview [57]. Turquoise cells represent the calculated cells, orange cells are the holes, and dark blue cells represent the interpolation region.

5. Computational Domain

The computational domain is made up of the background mesh region, two refinement regions, and an airfoil mesh region. The refinement regions here are a refinement of the background mesh in the wake of the airfoil. The background mesh is refined so that the inner refinement region is the same size as the far field airfoil mesh region. A schematic of the overall fluid domain is given in Fig. 17 and the final overset mesh is shown in Fig. 18. The overset mesh was generated Ansys meshing software [35] with the cell size and inflation layers described in Tab. 1. The background mesh, or fluid domain, is generated using OpenFOAM® blockMesh utility with the refinement regions made by running the refineMesh utility twice with different bounds on the background mesh. This has the effect of creating two mesh regions around the airfoil with finer resolutions. The background mesh was then manually adjusted so that the smallest refinement region cell size was the same size as the overset mesh region cell size.

Guidelines for overset mesh generation say that the cell size from the overset region to the background mesh should be the same size [27, 58]. For this reason, the overset mesh was constructed first. The size of the overset domain was made to be as small as possible but still allowing for a smooth transition into the background mesh. The background mesh with refinement regions was changed to match as closely to the size of the overset mesh as possible. Since the overset mesh is made of tetrahedral elements and the background mesh is constructed using hexahedral elements, the area of each cell in the overset region is roughly twice that of the background refined region. Following the guidelines of Guerrero [59], the overset mesh is built to ensure that there are more than three layers of elements between the hole region and the background mesh.

Following the guidelines for external flow simulations given by Goetten *et al.* [60], the trailing edge of the foil is made round and the "airfoil" is forced to be 250 elements for the top and the bottom. Goetten *et al.* further offers guidelines on the extent of the fluid domain. The group recommends having longitudinal and lateral domain extents greater than 200 reference lengths [60]. Many of the current undulating airfoil simulations do not follow the recommended guidelines for their simulation. For example, a common validation case given by Dong and Lu [16] has domain extents of [-2, 25] chord lengths in the x-direction and [-6, 6] chord lengths in the y-direction with all lengths given in meters. In lieu of using extents greater than 200 chord lengths, the current domain is [-11, 22] chord lengths in the x-direction and [-10, 10] chord lengths in the y-direction. The lateral dimensions result in a blockage ratio of 0.6%. The reason for a smaller domain is that the number of mesh elements increases by 30 times when comparing to a domain with extents of 200 chord lengths giving an element count of around ≈ 4.2 million severely increasing the computational demand. A summary of the final grid is given in Tab. 1 that includes the dimensions of the mesh as well as mesh metrics. More information on mesh metrics is provided by Knupp [61].

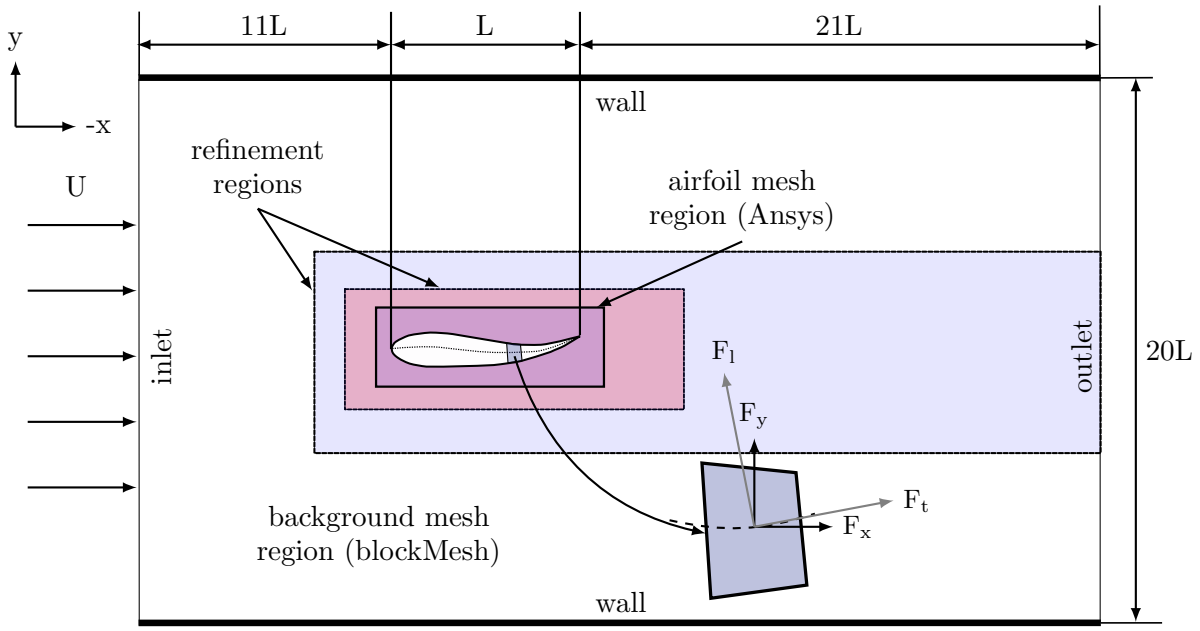
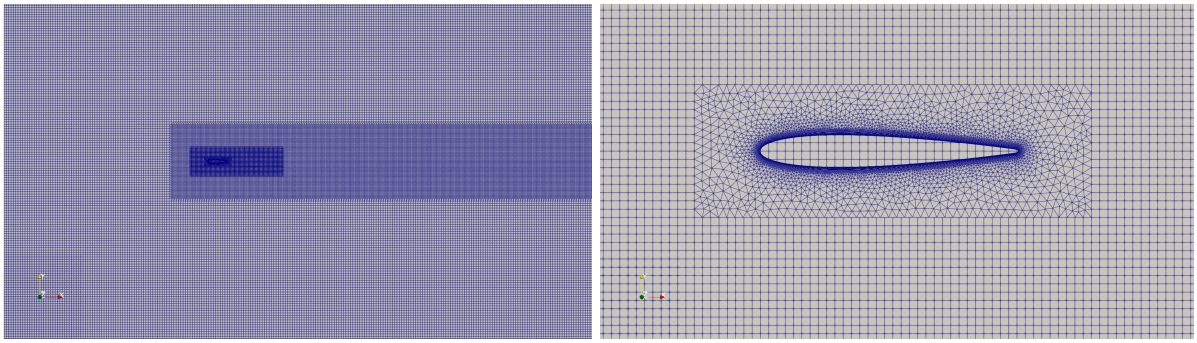


Figure 17. Schematic of computational fluid domain. Overset region is shown around the "airfoil", two refinement regions are used as shaded in red and blue. These refinement regions are only applied to the background mesh and are not separate component meshes. Note that the coordinate system is set so that -x is in the direction of flow as is customary for fish simulations.



(a) **(b)**

Figure 18. (A) Overview of the mesh with background mesh and two refined regions encompassing the overset mesh and (B) a zoomed view of the overset mesh showing the interpolation region of the overset to the background mesh being near the same size.

6. **Ika-Flow Implementation**

To implement the motion solver in OpenFOAM®, the solid body motion solver is rewritten to give the prescribed motion of fish as described in section 2. The solid body motion solver is chosen because it properly determines the overset mesh zone from the background mesh. The changes to the solid body motion class are shown in Fig. 19 and Fig. 20 using the unified modeling language (UML) [62].

Fig. 19 and Fig. 20 shows that the call to *transformField* is replaced by a custom fish motion method. The solid body motion solver was used because it allows for the accurate selection of the overset grid points which is facilitated by the **points0MotionSolver** and **zoneMotion** classes. The motion implementation in the *tranformationPoints* method for the Thunniform motion is given in (Listing 1).

All variables are provided in the **dynamicMeshDict** dictionary file within the constant directory of the simulation case. Coefficients and variables for Anguilliform and Carangiform swimming modes are provided by well established research [15,23,43]. Thunniform and ostraciform coefficients and parameters

Table 1. Summary of the mesh composition and metrics for the overset and background mesh. Metrics were given by OpenFOAM® **checkMesh** utility with *allGeometry* and *allTopology* flags turned on.

Airfoil Mesh	**Size**	
Airfoil domain [m]	0.02	
Airfoil edges (top/bottom) [divisions]	250	
Inflation first layer [m]	5.0e-05	
Number of inflation layers	35	
Inflation growth rate	1.12	
Metric	**Airfoil Mesh**	**Background Mesh**
Number cells	22,556	121,396
Max aspect ratio	80.20	1.00
Max skewness	1.68	0.33
Max non-orthogonality	40.78	18.43
Average non-orthogonality	15.91	1.52

```
Foam::tmp<Foam::pointField>
Foam::fishBodyMotionFunctions::ThunniformMotion::
transformationPoints(pointField& p0) const
{
    const scalar tm = time_.value();

    if (delay_ <= tm)
    {
        scalar t = tm - delay_;

        scalar maxAngle = maxAngle_ * 2 * M_PI / 180;
        scalar phase = phase_ * 2 * M_PI / 180;

        forAll(p0, pointI)
        {
            const scalar x = (p0[pointI].component(0)-origin_[0])/length_;
            const scalar y = p0[pointI].component(1)-origin_[1];
            const scalar z = p0[pointI].component(2)-origin_[2];

            scalar yr = 0;

            // check if the x-coordinate is larger than the pivot
            if (x >= pivot_)
            {
                //new value by equation
                const scalar xPivot = x - pivot_;

                const scalar localAmplitude = amplitude_ * (1 + (coefficients_[0]*(pivot_-1)) + (coefficients_[1]*(
                    ↪ pivot_*pivot_-1)));
                const scalar yEnd = localAmplitude * sin(waveNumber_*pivot_ - omega_*t) * length_;

                // calculate the tail angle
                const scalar thetaT = maxAngle * sin(waveNumber_*pivot_ - omega_*t + phase);

                yr = y + yEnd + xPivot * tan(thetaT) * length_;
            }
            else
            {
                // normal Carangiform body equation
                const scalar localAmplitude = amplitude_ * (1 + (coefficients_[0]*(x-1)) + (coefficients_[1]*(x*x
                    ↪ -1)));
                yr = y + localAmplitude * sin(waveNumber_*x - omega_*t) * length_;
            }
            p0[pointI] = vector(x, yr, z);
        }
        return p0;
    }
    else
    {
        return p0;
    }
}
```

Listing 1. Ika-Flow motion solver showing the Thunniform motion.

are based on the robotic operating parameters given by Zhu *et al.* [52]. Tab. 2 gives a summary of the entries used in the **dynamicMeshDict** for this research.

The ramp entry pertains to a logistics function applied to mesh in the tail region of the overset mesh. The ramp parameter can be tuned to control the deflection of mesh after the tail. This helps decrease

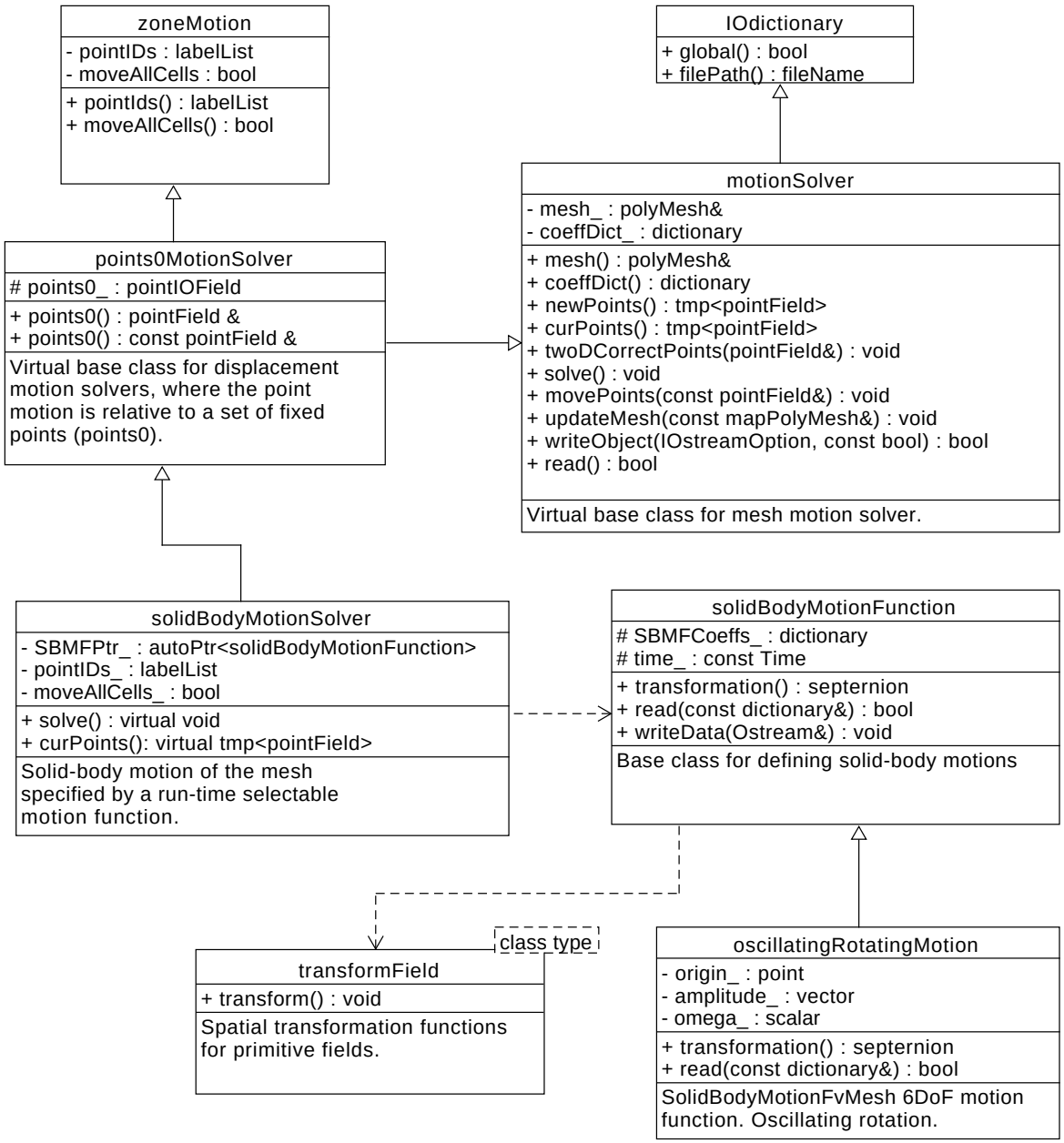


Figure 19. Original Solid Body class in UML

high skew regions that occur in this tail region. The logistic function is described as

$$0.5 - 0.5 \tan\left(\text{ramp} \cdot x - (\text{ramp} + 9)\right), \tag{21}$$

where ramp is the entry in the **dynamicMeshDict** and x is the x-coordinate of the mesh. This is used to limit the movement at the tail end of the mesh, to reduce mesh skewness problems. This function is not implemented in the Thunniform and Ostraciiform functions, as the mesh skewness is linear in the tail region for these two swimming modes. The right overset mesh in Fig. 21 shows an overdamped mesh where the left mesh shows the one used for this work. A ramp factor of 10 was experimentally found to be the optimal value in terms of limiting mesh skewness in the tail region without interfering with the tail movement. The

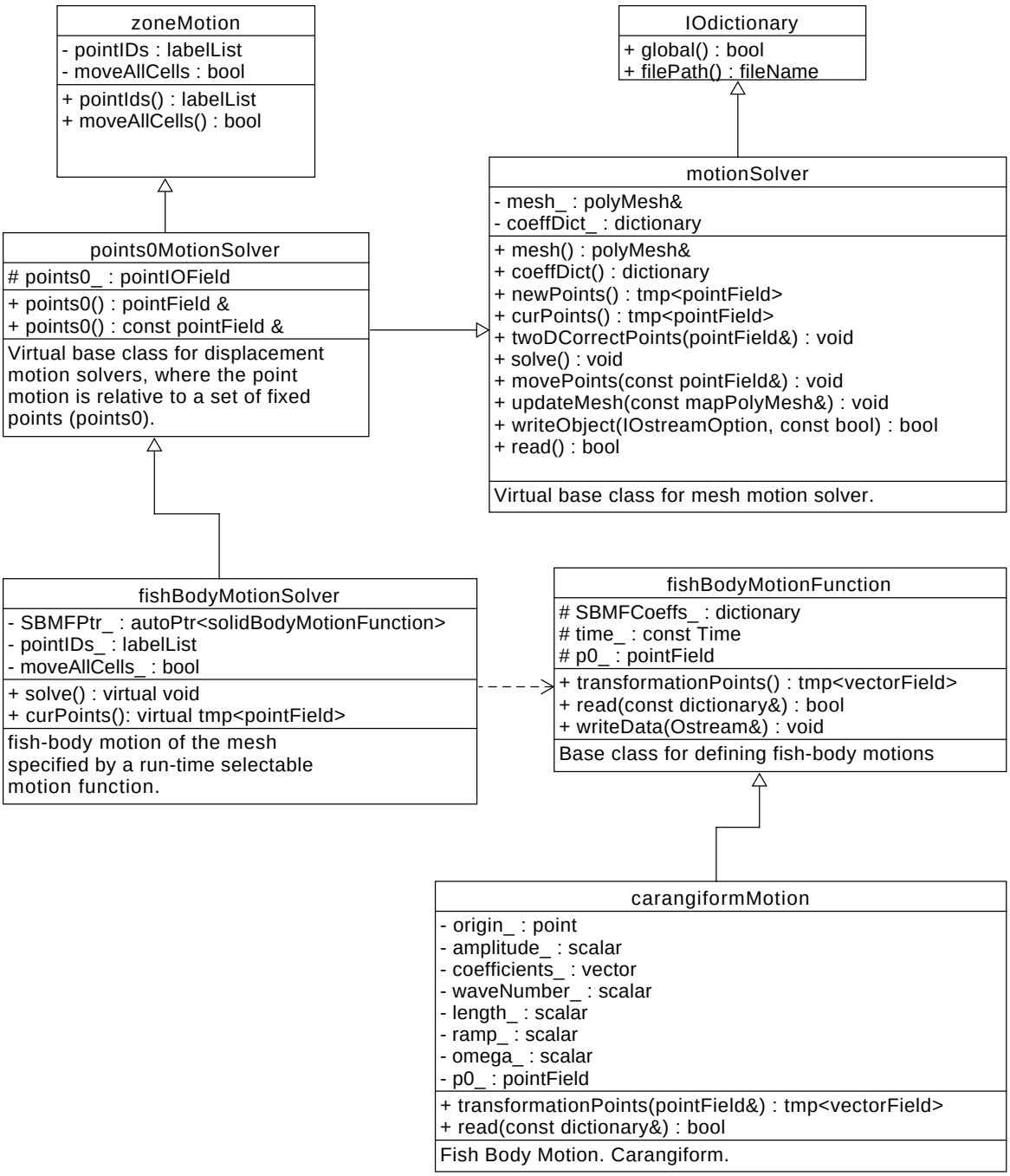


Figure 20. Modified fish body class in UML.

Table 2. Summary of **dynamicMeshDict** entries for all four BCF locomotion modes. All angles are in degrees.

Entry	Anguilliform	Carangiform	Ostraciiform	Thunniform
origin	(0.0 0.0 0.0)	(0.0 0.0 0.0)	(0.0 0.0 0.0)	(0.0 0.0 0.0)
amplitude	0.10	0.10	0.10	0.10
coefficients	(0.323 0.310 0.0)	(-0.825 1.625 0.0)	(-0.825 1.625 0.0)	(-0.825 1.625 0.0)
waveNumber	6.28	6.28	6.28	6.28
length	1.00	1.00	1.00	1.00
ramp	10.00	10.00	10.00	10.00
omega	12.57	12.57	12.57	12.57
delay	5.00	5.00	5.00	5.00
pivot			0.85	0.85
maxAngle			16.00	16.00
phaseAngle			90.00	90.00

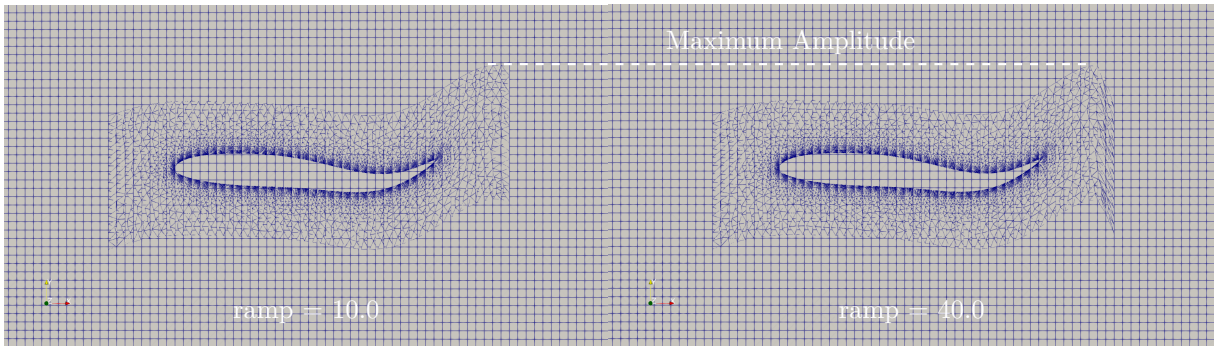


Figure 21. Two meshes showing the effect of the damping factor, controlled by the ramp entry in the dynamicMeshDict. The left mesh shows a ramp factor of 10.0 and the right mesh shows a ramp factor of 40.0.

7. Numerical Setup

For the simulations performed in this research, the 2003 variant of the k-ω shear stress turbulence (SST) model presented by Menter *et al.* [63] is used. Here, k, ω are calculated using guidance from This model has been widely used for previous fish CFD simulations [14–21] and has shown superior results compared to other Unsteady Reynolds Averaged Navier Stokes models such as k-ϵ and Spalart-Allmaras in a study on vertical axis wind turbine blades by Meana-Fernández *et al.* [64]. In section 9, there is a comparison between three well known turbulence models: the k-ω SST model, the k-k_L-ω given by Fürst *et al.* [65], and the Spalart-Allmaras model given by Spalart and Allmaras [66]. The Spalart-Allmaras model is a well used model for external aerodynamic flows and the k-k_L-ω model is a recently developed model that shows accurate results especially in transitional flows [65, 67]. To the author's knowledge, there is not another comparison of these three models in the context of fish swimming.

The initial conditions used are given in Tab. 3. The simulations were all run with the inlet velocity (V_x, V_y, V_z) set to uniform (1.0 0.0 0.0). The Reynolds number was varied by varying the kinematic viscosity using

$$\nu = \frac{UL}{\mathrm{Re}}, \tag{22}$$

where ν is the kinematic viscosity, U is the free stream velocity, L is the length of the airfoil, which is kept at 1 m, and Re is the desired Reynolds number.

Table 3. Summary of initial condition entries for simulations used in this work. Velocity and pressure fields are kept the same with turbulence values calculated based on Reynolds number. The turbulence values in this table correspond to the validation case with a Reynolds number of $5 \cdot 10^3$.

File	Inlet	outlet	walls	overset
U	fixedValue	zeroGradient	movingWallVelocity	overset
	uniform (1.0 0.0 0.0)	uniform (1.0 0.0 0.0)	uniform (0.0 0.0 0.0)	uniform (0.0 0.0 0.0)
p	zeroGradient	fixedValue	zeroGradient	overset
		uniform 0		uniform 0
k	turbulentItensityKineticEnergyInlet	inletOutlet	kqRWallFunction	overset
	intensity 0.05	uniform 3.75e-03	uniform 0	uniform 0
omega	fixedValue	inletOutlet	omegaWallFunction	overset
	uniform 8.75e-01	uniform 8.75e-01	uniform 8.75e-01	uniform 0
epsilon	fixedValue	inletOutlet	epsilonWallFunction	overset
	uniform 2.95e-06	uniform 2.95e-06	uniform 2.95e-06	uniform 0
nut	zeroGradient	zeroGradient	nutkWallFunction	overset
	uniform 4.29e-03	zeroGradient	uniform 4.29e-03	uniform 0
nuTilda	zeroGradient	zeroGradient	zeroGradient	zeroGradient

Calculation of the turbulent boundary conditions starts with a turbulence intensity, I, a length scale, L, and a reference velocity, u_{ref} [53, 68–70]. A turbulence intensity of 5%, a length scale of 1 m, and a reference velocity of, u_{ref} is used for this case. Different Reynolds numbers were achieved by keeping the reference velocity constant and changing the kinematic viscosity, ν. The turbulent energy k is then estimated as

$$k = \frac{3}{2}(I|u_{ref}|)^2. \tag{23}$$

The specific turbulence dissipation rate, ω, is defined as

$$\omega = \frac{k^{0.5}}{C_\mu^{0.25} L}, \tag{24}$$

where C_μ is a constant equal to 0.09. If needed, the turbulence dissipation rate, ϵ, can be found using

$$\epsilon = C_\mu \frac{k^{3/2}}{L}. \tag{25}$$

Specific guidelines for numerical schemes when utilizing overset simulations in OpenFOAM® are provided by Guerrero [59]. It states that there should be at least 5 PIMPLE iterations for dynamic meshes. PIMPLE iterations refers to the number of iterative marching steps performed after the PISO algorithm. This is controlled in the fvSolution file as the nOuterCorrectors entry. An overview of the difference between the PISO and PIMPLE algorithm is given in Fig. 22 and Fig. 23. Tab. 4 gives the discretization

schemes used as specified in the fvSchemes file for this research. For overset mesh, it is recommended that the turbulent values (k, omega, nut) be interpolated explicitly [59].

Table 4. Numerical schemes and their respective term. Schemes are limited to second order except for time scheme. x_j represents the partial derivative where $1-3$ is a Cartesian coordinate direction (i.e.) $1 = \partial x, 2 = \partial y, 3 = \partial z$.

Term		**Scheme**	**OpenFOAM Entry**
transient	$\frac{\partial u_i}{\partial t}$	1st order implicit	Euler
gradients	$\frac{\partial}{\partial x_j}$	1st order cell limited	cellLimited Gauss linear 1
advective U	$u_j \frac{\partial u_i}{\partial x_j}$	2nd order single limiter	Gauss linearUpwindV
advective turbulence	$u_j \frac{\partial (k\|\omega)}{\partial x_j}$	2nd order	Gauss linearUpwind
laplacian	$\frac{\partial^2}{\partial x_j^2}$	2nd order limited	Gauss linear limited 1
interpolation	$\phi P \phi L$	2nd order linear	linear
surface normal gradients	$\frac{\partial}{\partial n}$	explicit non-orthogonal corrected	limited 1
wallDistance	$\nabla^2 \psi$	Poisson Equation	Poisson

The temporal control of the simulation was performed using the adjustable time step feature and keeping the max Courant number below 0.5. The CourantFriedrichsLewy, CFL, condition was introduce by Courant, Friedrichs, and Lewy in 1928 and later translated to English in [71]. It can be written as

$$CFL = \frac{U \Delta t}{\Delta x} \leq CFL^{max}, \tag{26}$$

where U is the fluid velocity, Δt is the time step, and Δx is the dimension of a mesh cell in the direction of fluid flow. In practice, this CFL number is normally kept to be less than 1.0, which means that the fluid cannot jump more than one mesh cell within a timestep. A comparison of different maximum Courant numbers, CFL^{max}, is given in section 9.

8. **Forces, Power, and Efficiencies**

The forces library for OpenFOAM® is used to calculate the lift and drag coefficients. The forces are broken down into a normal pressure contribution as [53]

$$F_p = \sum_i \rho_i s_{f,i} \left(p_i - p_{ref} \right) \tag{27}$$

and a viscous component of

$$F_\nu = \sum_i s_{f,i} \cdot \left(\mu \mathbf{R}_{dev} \right) \tag{28}$$

where ρ is the density, $s_{f,i}$ is the face area vector. p is the pressure, μ is the dynamic viscosity, and $\mathbf{R}_{dev}$ is the stress tensor [53]. In 2D, these components add up to a total force, F_{total}, which form the non-dimensionalized drag, C_D, and lift, C_L, defined as

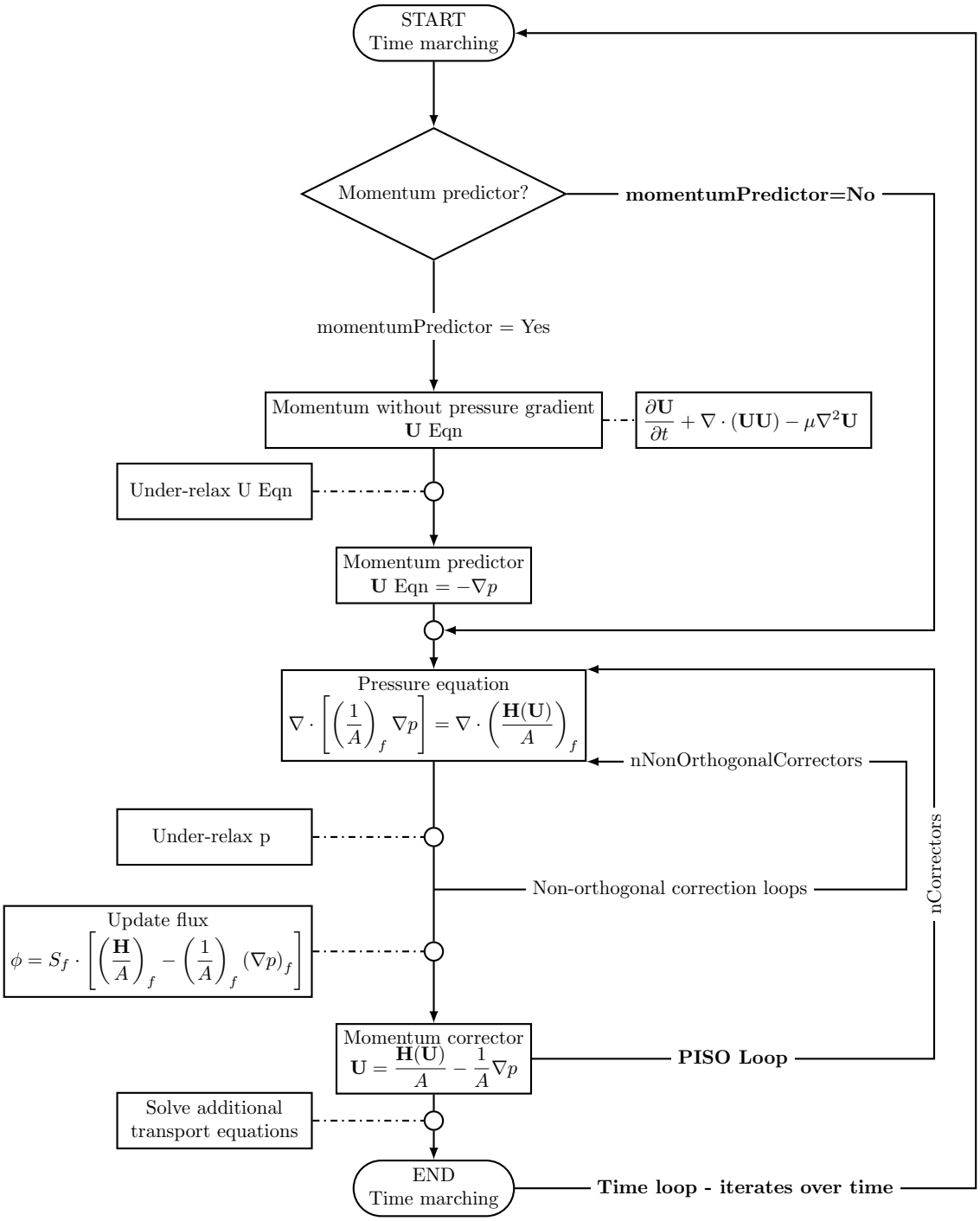


Figure 22. PISO algorithm with non-iterative marching

$$C_D = \frac{F_{total,x}}{\frac{1}{2}\rho U^2 L}, \quad \text{and} \tag{29}$$

$$C_L = \frac{F_{total,y}}{\frac{1}{2}\rho U^2 L}, \tag{30}$$

where the x and y are the component of force, U is the free stream velocity and L is the length of the airfoil.

8.1. **Post-processing.** Post-processing is needed for the results of a simulation because of the presence of high frequency noise. This is a common occurrence when simulating fish motion, and occurs even when using a body-fitted mesh with OpenFOAM® [54]. This observed noise is not limited to OpenFOAM®, as

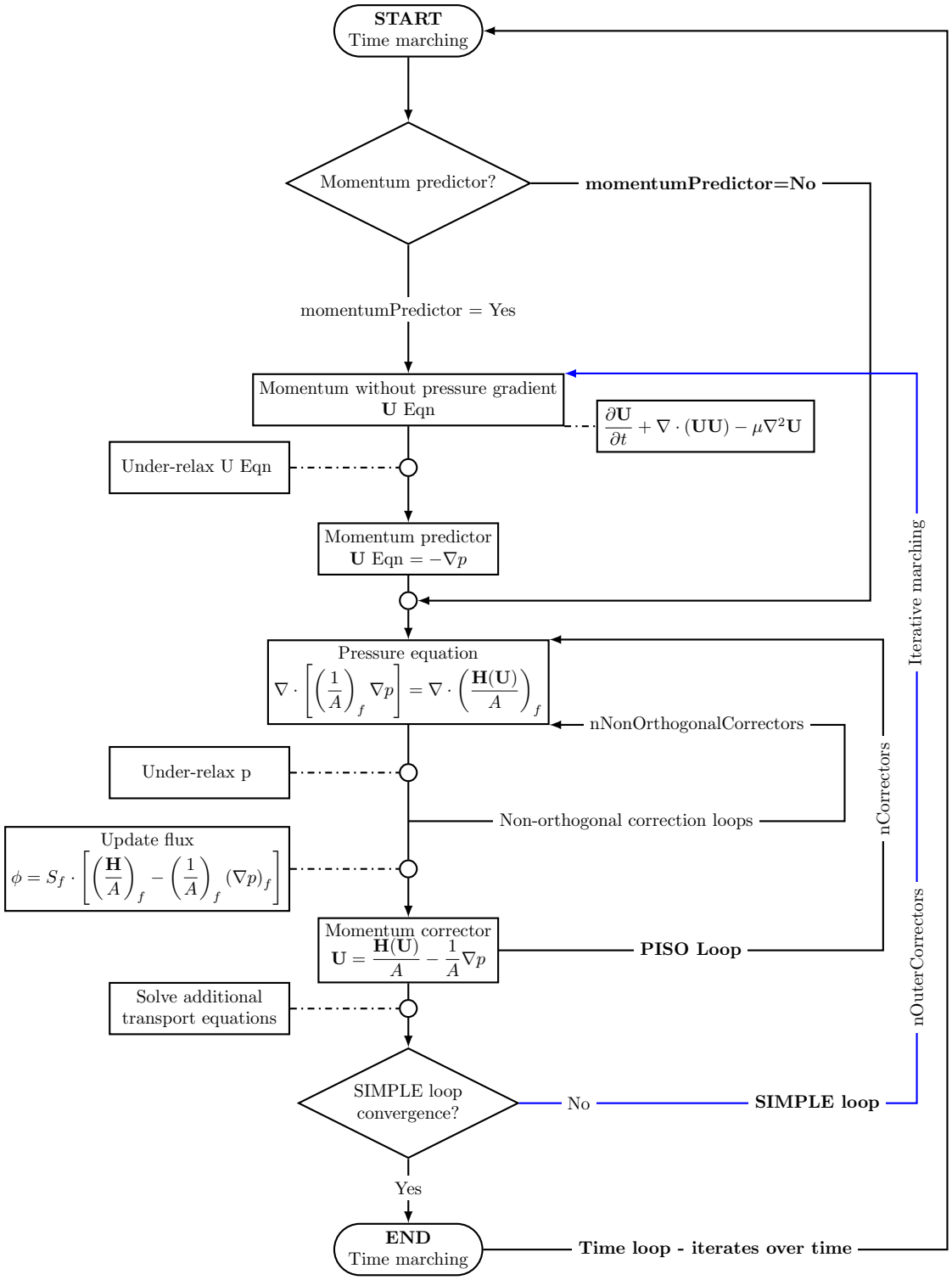


Figure 23. PISO algorithm with iterative marching (PIMPLE). Adapted from Guerrero [59]

it is shown to occur using a body fitted mesh in Fluent [72] and non-commercial code [17]. Although these are present for this type of simulation, the interpolation of the overset mesh amplifies this phenomenon, therefore, care must be taken when analyzing and visualizing the results. The origins of this noise for these types of simulations is definitely an interesting topic to explore further.

To filter this noise, a convolve function using a Hanning filter, as described by Harris *et al.* [73], is used on the raw force output. The Hanning window is defined as:

$$w(n) = 0.5 - 0.5\cos\left(\frac{2\pi n}{M-1}\right) \quad \text{for} \quad 0 \leq n \leq M-1, \tag{31}$$

where $w(n)$ is the one-dimensional window with a maximum value normalized to 1, n is the index of window function (equal to 1 for this case), and M is the length of the kernel (101 for this work), which is odd for the implementation in this work. Fig. 24 shows the raw data with the filtered data superimposed.

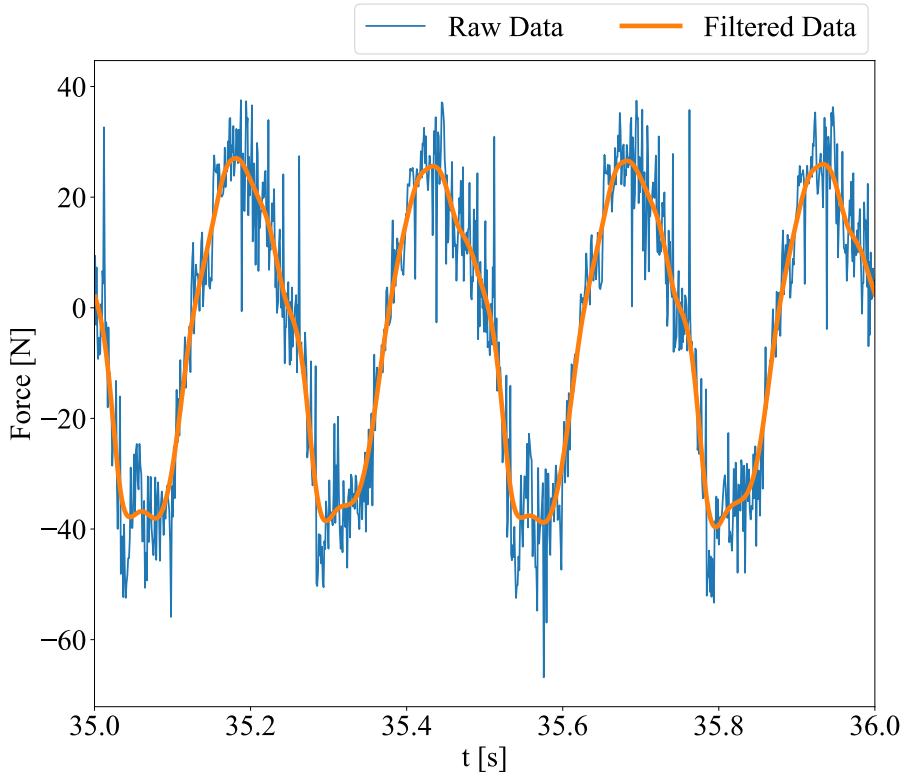


Figure 24. Comparison of forces in the x-direction before and after filtering using a Hanning filter. Data is from the lowest Reynolds case of $5 \cdot 10^3$.

The use of the adjustable time step and limiting the simulation to a CFL number gives a non-uniform temporal output for the raw forces. Uniform temporal sampling was performed using the Scipy [74] linear interpolation method in order to ensure that time averaging could be taken using the arithmetic mean, and to decrease the overall amount of data. For all figures, time is normalized by

$$\tau = t/T, \tag{32}$$

where T is the period for one tail beat cycle.

9. **Limitations and Comparisons**

This section compares the following simulation parameters: number of PIMPLE iterations, the use of a time delay, the max Courant number, different overset meshes, and three different turbulence models. The simulation results as well as the clock time (real time to finish a simulation) are used as a performance metric. This is useful in order to determine which parameters can be changed to speed up simulation times, without effecting results, and also how each parameter effects results. In the following comparisons, the time averaged values are the ones being compared and are shown as dashed lines.

In literature [14, 16–18] concerning fish motion, a unifying dimensionless number to describe the fish tail motion is the Strouhal number (St) is defined as

$$\mathrm{St} = \frac{fA}{U}, \tag{33}$$

where f is the frequency of the tail beat, A is the peak-to-peak amplitude of the fish tail, and U is the speed of the fish. In general, swimming fish operate at a Strouhal number range of $0.2 < \mathrm{St} < 0.4$ for

Reynolds numbers greater than $5 \cdot 10^3$ [25]. Below this critical Reynolds number, St $\sim$ Re$^{-1/4}$. For this work, U is fixed at 1 m/s and A at 0.2 m making St a measure of tail beat frequency. All simulations are performed at a Reynolds number of $5 \cdot 10^3$ and a Strouhal number of 40, unless otherwise specified.

9.1. **Grid Independence and Validation.** To the author's knowledge, there are no 2D overset studies involving fish motion. Previously referenced overset studies of NACA 0012 airfoils are with no motion [28–31] or only look at pitching motion [27]. Another limitation with the existing research is that there is no research at Reynolds numbers higher than 10^4. Two studies, one by Dong and Lu [16], and the other by Yu *et al* [14], were chosen as validation cases. Dong and Lu looked at two-dimensional airfoils that were in a side-by-side arrangement. Their work focuses on a wake effect from one swimmer to another. However, it included results for a single airfoil and was used extensively as a validation case for 2D swimming simulations. Their simulations were done at a fixed Reynolds number of $5 \cdot 10^3$, and Strouhal numbers of $0.2 < \text{St} < 0.4$ [16].

In contrast to Dong and Lu, Yu *et al.* focused on developing a mathematical law for the power scaling of a single fish across a range of Reynolds and Strouhal numbers. The group utilized Dong and Lu for their validation case, but their Reynolds numbers were $5 \cdot 10^3 < \text{Re} < 5 \cdot 10^4$ and Strouhal numbers of $0.2 < \text{St} < 0.4$ [14]. This validation is combined with a grid independence study in accordance with the guidance of Celik *et al.* [75].

To estimate the drag error, the Root Mean Square Difference (RMSD) method is used as follows:

$$\text{RMSD} = \sqrt{\frac{\sum_{i=1}^{N} (\hat{y}_i - y_i)}{N}}, \tag{34}$$

where $\hat{y}$ is the validation data from Dong and Lu [16], y is the simulated data, and N is the number of data points. For the validation data, linear interpolation over the time series is used.

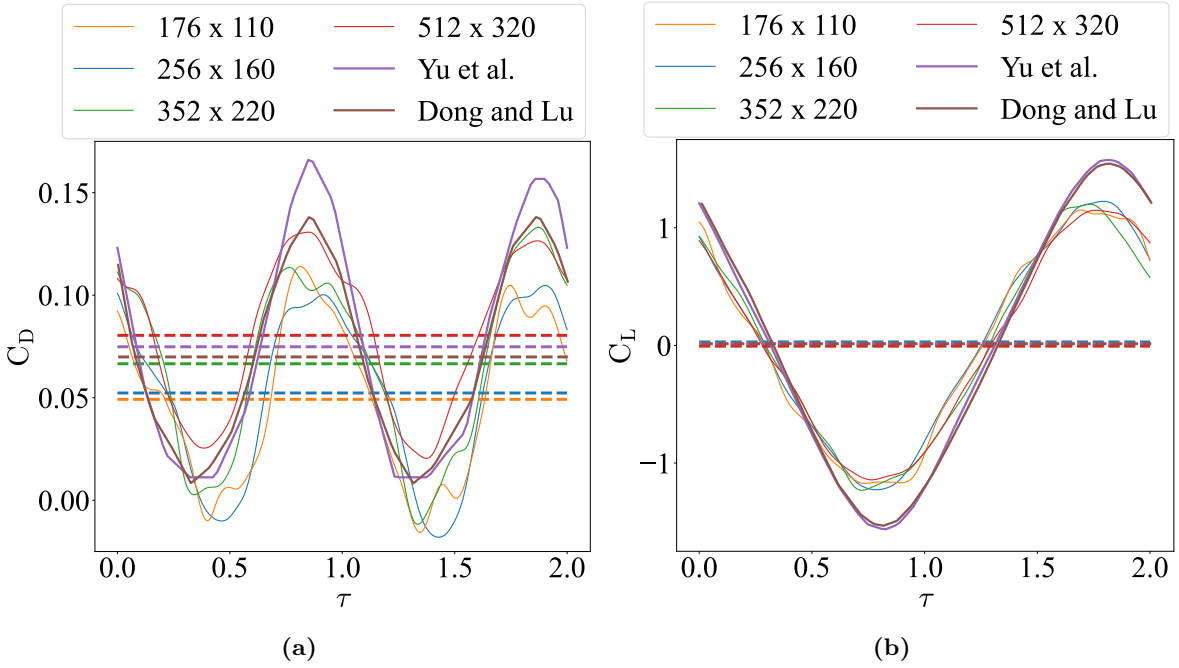


Figure 25. (A) C_D and (B) C_L for the same simulation data from Dong and Lu [16] and Yu *et al.* [14]. Both Dong and Lu and Yu *et al.* represent body-fitted morphing meshes. Plot is only of the Carangiform swim mode. Dashed lines of the same color represent the time averaged values.

Fig. 25 shows variations between the drag plots while negligible difference in lift data. For reference, the coarsest mesh is $\approx 39\%$ different from the finest mesh. Looking at Tab. 5, the difference between the second mesh and the next finest mesh is 4.42 days of simulation time. When looking at the time averaged values, shown as dashed lines in Fig. 25, there is significant variation in C_D but C_L is near zero. C_L near zero is expected as a normal symmetric airfoil would have 0 lift at an angle of attack at 0, or free stream velocity only in the x-direction. The error between the body-fitted results of Dong and Lu [16] and Yu *et al.* [14] can be partially explained by the overset interpolation error discussed in section 4. Other factors

Table 5. Summary of simulations for increasing grid sizes. Error is reported with the results from Dong and Lu [16]. Drag error is calculated using the Root Mean Square Error method. All simulations were performed using 8 CPU cores.

Dimensions	No. Elements	Clock Time [s]	Clock Time [hr]	Clock Time [day]	Drag Error [%]
176 x 110	51508	200568.00	55.71	2.32	34.67
256 x 160	85760	338924.00	94.15	3.92	32.00
352 x 220	143952	720498.00	200.14	8.34	6.67
512 x 320	282002	1203972.00	334.44	13.93	5.33

like the larger fluid domain and choice of numerics also contribute to the difference between the results of this research and the validation cases. The 6% difference between third mesh in Table Tab. 5 and the validation case was acceptable for this work, and so this mesh density was chosen for the proceeding simulations. This mesh has a longer computational time than the first 2 counterparts, but is much closer to the validation cases.

9.2. **The Effect of Max Courant Number.** The simulation temporal discretization is controlled via the maximum Courant number. In theory, the benefit of using the PIMPLE algorithm allows for the use of a higher Courant number in simulations thus lowering simulation time. Here, three Courant numbers: 0.5, 1.0, 2.0 are compared. These simulations are run at a Reynolds number of $5 \cdot 10^3$ and a Strouhal number of 40.0, with the number of PIMPLE iterations set to 5. The number of PIMPLE iterations (nOuterCorrectors) must be greater than 1 (the PISO algorithm) to take advantage of the time marching method. Fig. 26 shows the results from this comparison.

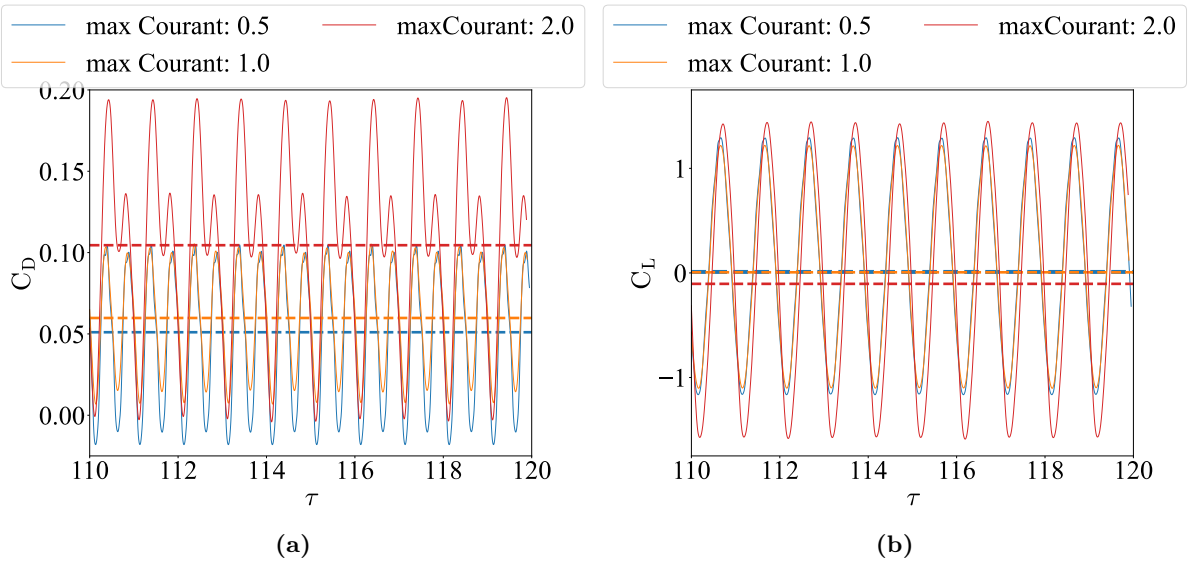


Figure 26. (A) C_D and (B) C_L for max Courant numbers of 0.5, 1.0, and 2.0. Dashed lines of the same color represent the time averaged value.

The data shows a second oscillation occurs with the 2.0 max Courant number that is not present at lower Courant numbers. Furthermore, the drag increases with the higher Courant number which is explained when examining the residuals of the simulations. By the 5^{th} PIMPLE iteration for each time step, the 0.5 Courant simulations converge to an order of magnitude smaller than that of the 2.0 Courant number, with the 1.0 Courant case somewhere in between. These results might improve with a higher number of PIMPLE iterations, but this is out of scope for this research, and the simulation times shown in Table Tab. 6 support using the smaller Courant number versus more PIMPLE iterations.

9.2.1. *Courant of 2.0 with 50 Outer Corrector Iterations.* The behavior of C_D and C_L in Fig. Fig. 26 is investigated by increasing the number of maximum outer correctors (nOuterCorrectors) to 50. From Fig. Fig. 23, this number corresponds to the iterative marching in the SIMPLE loop, which will allow the solution to converge each time step as long as it does so within 50 iterations of the SIMPLE algorithm.

Table 6. Summary of simulations for increasing grid sizes. All simulations were performed using 8 CPU cores.

Max Courant	Clock Time [s]	Clock Time [hr]	Clock Time [day]
0.5	326658.00	90.74	3.78
1.0	299850.00	83.29	3.47
2.0	314158.00	87.27	3.64

For the comparison, only the outer correctors was changed along with the maxCo number. Fig. 27 shows the results from this comparison.

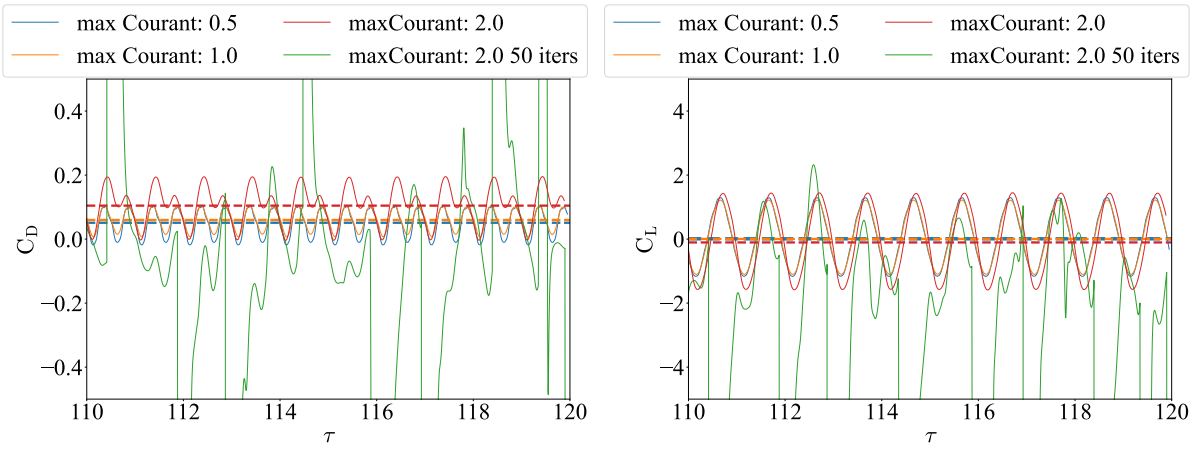


Figure 27. (A) C_D, (B) C_L zoomed into the last 10 tail beat cycles. All plots show max Courant numbers of 0.5, 1.0, 2.0, and 2.0 with 50 outer corrector iterations (labeled 50 iters)

As with Fig. Fig. 26, there is a secondary oscillation on the data not present with the cases where CFL < 1.0. Furthermore, the data is far lower and less oscillatory as would be expected from this particular use case. The output from the solver shows that the the solution did not converge for most of the time steps after 50 iterations. As Table Tab. 7 shows, this causes the simulation time to increase dramatically.

Even more interesting is that when comparing Fig. 27 and Fig. 37, the non-convergence behavior is not seen even at 25 iterations. The difference between these simulations is the maxCo used for Fig. Fig. 37 is below 1.0 for all cases. A deeper investigation into this behavior is planned for the near future. For this work and the validation case, the maxCo was set to be 0.5.

Table 7. Summary of simulations for increasing grid sizes. All simulations were performed using 8 CPU cores.

Max Courant	Clock Time [s]	Clock Time [hr]	Clock Time [day]
0.5	326658.00	90.74	3.78
1.0	299850.00	83.29	3.47
2.0	314158.00	87.27	3.64
2.0 50 Outer	1391934.00	453.67	18.90

9.3. **Reaching Quasi-Steady-State.** A common limitation with simulating fish undulation movement is that the simulation requires many cycles of undulation ≈ 75 in order to reach a steady-state. Here steady-state or quasi-steady-state is defined when the forces on the body (lift and drag) during a tail beat cycle do not vary in amplitude from the proceeding cycle. This is also a phenomenon that is seen in

Borazjani and Sotiropoulos [17] and is not covered by most existing fish CFD literature. Fig. 28 shows the lift and drag for all four BCF modes of locomotion run for 60 seconds of flow time run at a strouhal number of 40 which represents the fastest tail beat amplitude used in this research.

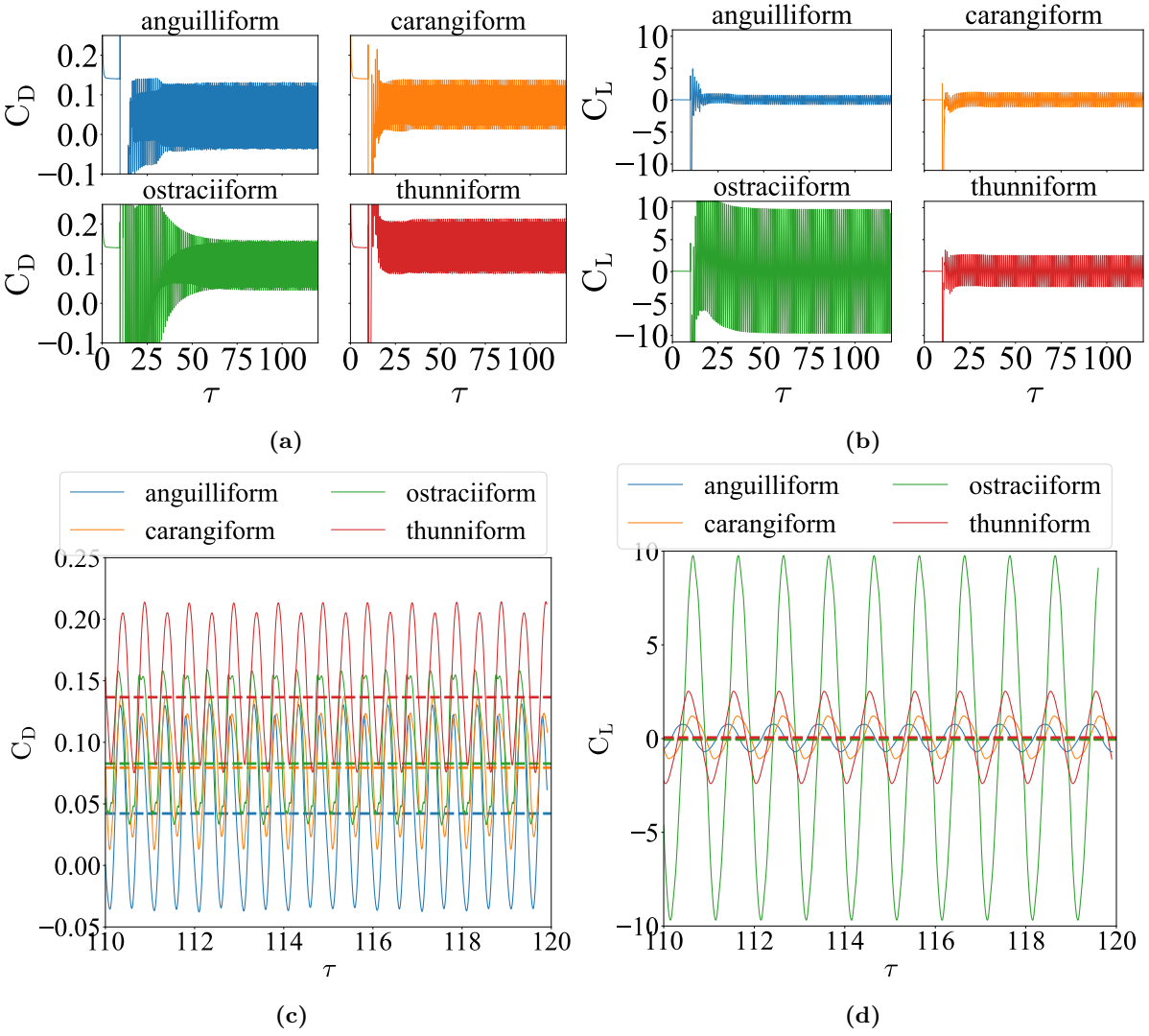


Figure 28. (A) C_D, (B) C_L for a full 120 tail beat cycles for all swimmers. (C) C_D zoomed in, and (D) C_L zoomed in for all four BCF swimming modes for the last 10 cycles using the PISO algorithm (1 nOuterCorrectors).

Fig. 28 show that Anguilliform and Ostraciiform swimming modes reach steady-state at near the same flow time at ≈ 40 seconds, corresponding to 80 tail beat cycles. Carangiform and Thunniform similarly reach steady-state at ≈ 25 seconds, corresponding to 50 tail beat cycles. Interestingly, the Anguilliform swim mode shows the least amount of drag while the Thunniform shows the largest. The only difference between the Thunniform and Carangiform swim modes is a rigid tail 85% down the body that pitches with a phase offset. This means that the rigid tail and phase offset contribute a lot to the overall drag. It would be expected that the Thunniform drag is closer to the Carangiform drag. However, the tail phase offset causes an abrupt change in flow direction, which can contribute to drag.

The lift plots in Fig. Fig. 28d show that the lift is nearly the same for all four locomotion modes. It is expected that lift hovers around 0 for a symmetric airfoil, so this is consistent with previous works [76]. The largest amplitude oscillations occur with the Ostraciiform motion due to the airfoil effectively pitching its angle of attack, while having a phase offset trailing edge. This pitching motion is what causes the largest amplitude oscillations when compared to the other three BCF modes. In order to determine the number of computer hours for each simulation, the simulation times are given in Table Tab. 8. Interestingly, the Ostraciiform mode took almost twice as long as the other three modes which can be explained by the solvers taking more iterations to reach their tolerance in the Ostraciiform case than in

the Anguilliform counterpart. The cause for this is the abrupt change in the flow field that causes a large pressure jump at the pivot point of the tail, for the Thunniform and Ostraciiform modes.

Comparing the drag and lift plots for the BCF swimmers in Fig. 28c and Fig. 28d, it is seen that the Thunniform has around twice the amount of drag and the Ostraciiform has about 3 times the amount of lift. In nature, Thunniform is considered the most efficient locomotion form for BCF swimmers [52, 77]. For these simulations, the parameters of all the swimmers were kept the same as the Carangiform swimmer in order to establish a one-to-one comparison. These are non-optimal parameters for these two swimming modes and that is a cause for the higher lift and drag plots for both the Thunniform and Ostraciiform modes.

Table 8. Summary of simulation times reaching a quasi steady-state condition and completion of 60 seconds of flow. All simulations were performed using 8 CPU cores.

Case	**Steady-State [s]**	**Clock [s]**	**Clock [hr]**	**Clock [day]**
Anguilliform	40.0	213322	59.26	2.47
Carangiform	25.0	185032	51.40	2.14
Ostraciiform	40.0	350270	97.30	4.05
Thunniform	25.0	207611	57.67	2.40

9.3.1. *Pressure Coefficients.* To understand the reason for this phenomenon, the pressure coefficient is investigated for a swimming cycle. The pressure coefficient, C_P is the non-dimensionalized pressure on the swimmer is given by

$$C_P = \frac{(P_{incident} - P_{freestream})}{0.5\rho U^2}, \tag{35}$$

where $P_{incident}$ is the pressure on the swimmer, $P_{freestream}$ is the far field pressure, ρ is the fluid density, and U is the free stream velocity. Post processing of values was performed using Paraview software [78].

Fig. 29 and Fig. 30 show the swimmer body position and corresponding C_P values along the chord length done at Reynolds number of $5 \cdot 10^3$ and Strouhal number of 40. The figures show that throughout a swim cycle, the Ostraciiform and Thunniform locomotion modes have larger pressures incident on the swimmer, with the Thunniform's pressure distribution being relatively smaller than the Ostraciiform's. This gives a direct correlation between the pressure distribution on the swimmer and the amount of power needed for thrust. An interesting observation is that the Ostraciiform mode does not have a leading edge pressure distribution that increases exponentially like the other three modes. This is a result of this mode being in a pure pitching motion versus an undulating mode of the other three which effectively provides a more blunt leading edge.

A further observation made in Fig. 29 and Fig. 30 is the abrupt change in pressure distribution at the trailing edge for the Ostraciiform and Thunniform cases. This is in contrast to the smoother transition seen in the Anguilliform and Carangiform cases. This is caused by the change in geometry for the pitching tail, which gives a drastic change in fluid direction versus the smoothly transitioning tails. This causes a high pressure region on one side of the tail that is not present in the Anguilliform and Carangiform cases. This high pressure region adds to the total power needed to overcome the drag force for self propulsion.

9.3.2. *Velocity Results.* Fig. 31, Fig. 32,Fig. 33, and Fig. 34 display the velocity contours for all swimming modes. These simulations were performed at a Reynolds number of $5 \cdot 10^3$ and a Strouhal number of 40. The results show that for the Anguilliform, Carangiform, and Ostraciiform locomotion modes, a reverse Kármán street is formed behind the swimmer, which agrees with similar simulations of fish motion in literature [14–21, 25]. For the Thunniform swim mode, there is a definite alternating flow pattern. However, the Kármán street is not as well defined as the other locomotion modes. Experiments and simulations show that there is a definite reverse Kármán street in the wake of Thunniform swimmers [52, 77]. The diffusion of vortices into the wake may be a consequence of prescribed motion parameters not being optimized for this type of locomotion. An investigation into this swimming mode and how the different parameters affect the wake is planned for a future study.

A further interesting result is the formation of faster vortices behind the Ostraciiform locomotion as shown in Fig. 33. This indicates that more energy is transferred to the free stream. The Ostraciiform case also shows larger pressure fields forming around the swimmer's body during locomotion when compared to other swimming modes. Comparing Fig. 33 and Fig. 34, the only difference between the running

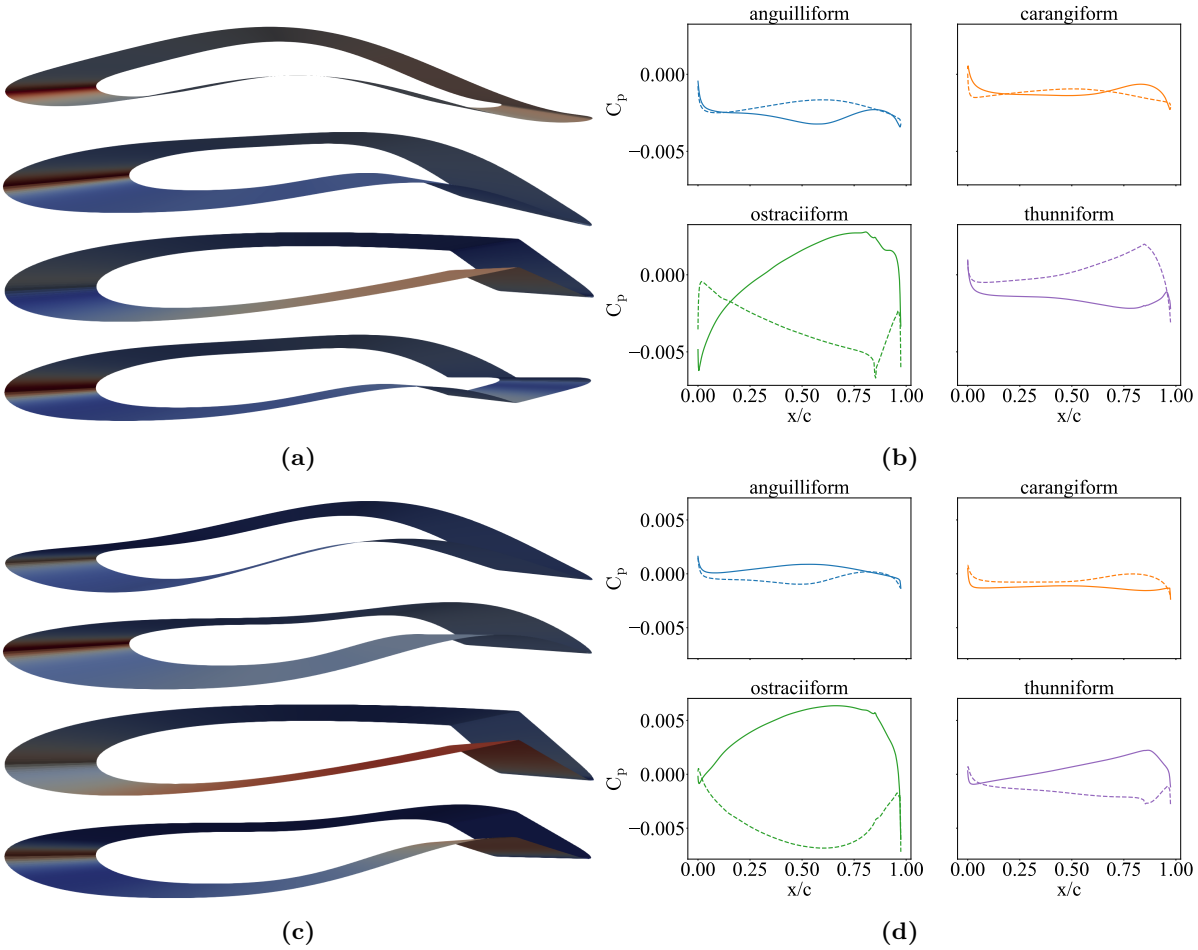


Figure 29. (A), (C) Swimmer at first half of cycle time, τ =0.2, 0.4. Anguilliform, Carangiform, Ostraciiform, and Thunniform modes from top to bottom. (B), (D) upper and lower pressure coefficient, C_P, for each locomotion mode plotted over the chord length. Solid lines represent the lower airfoil and dashed lines represent the upper airfoil.

parameters of these two cases is that the body undulates instead of pitching. This comparison suggests that the undulation of the body minimizes pressure changes around the body. This can also be seen in the span-wise pressure distribution in Fig. 29 and Fig. 30. The effects on span-wise C_D and C_L are shown in Fig. 36b and discussed in the next section.

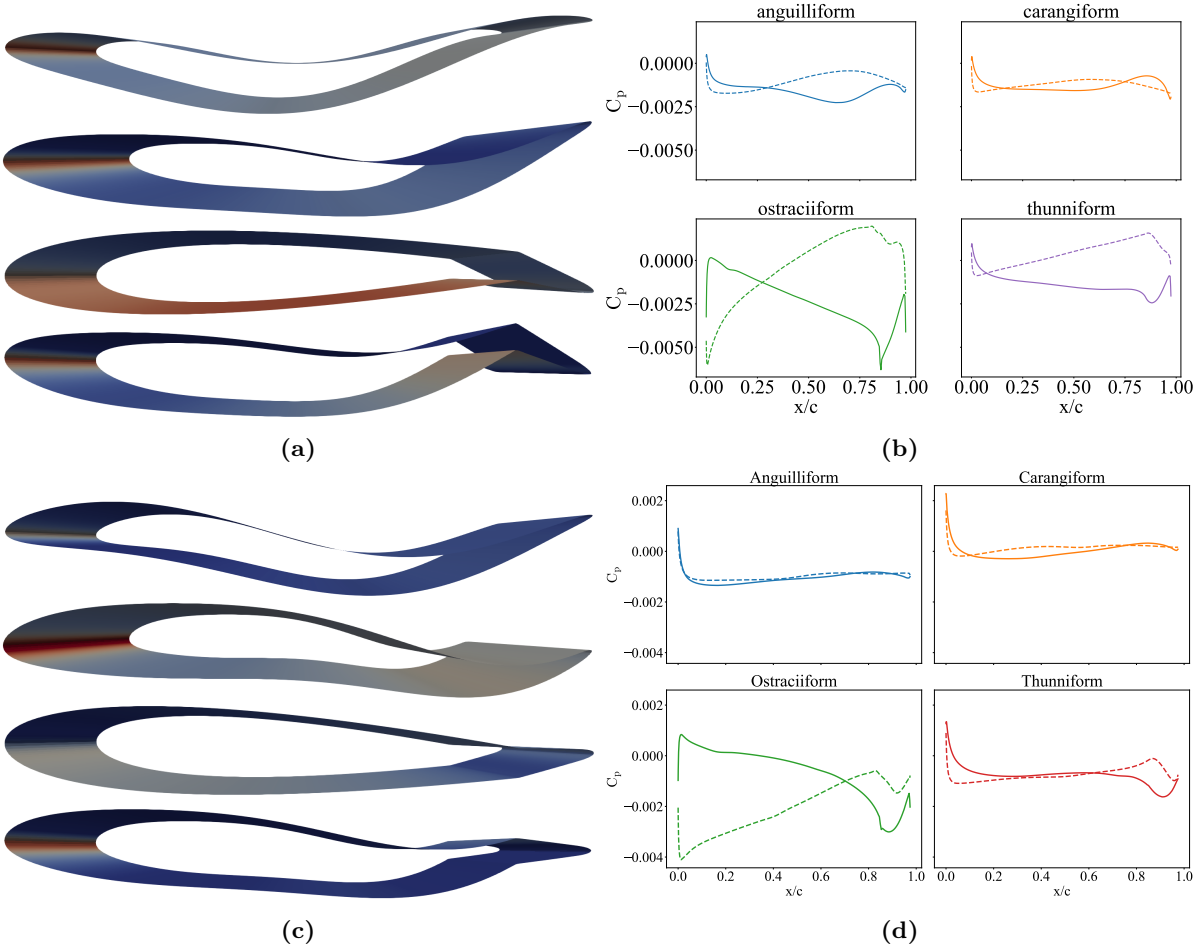
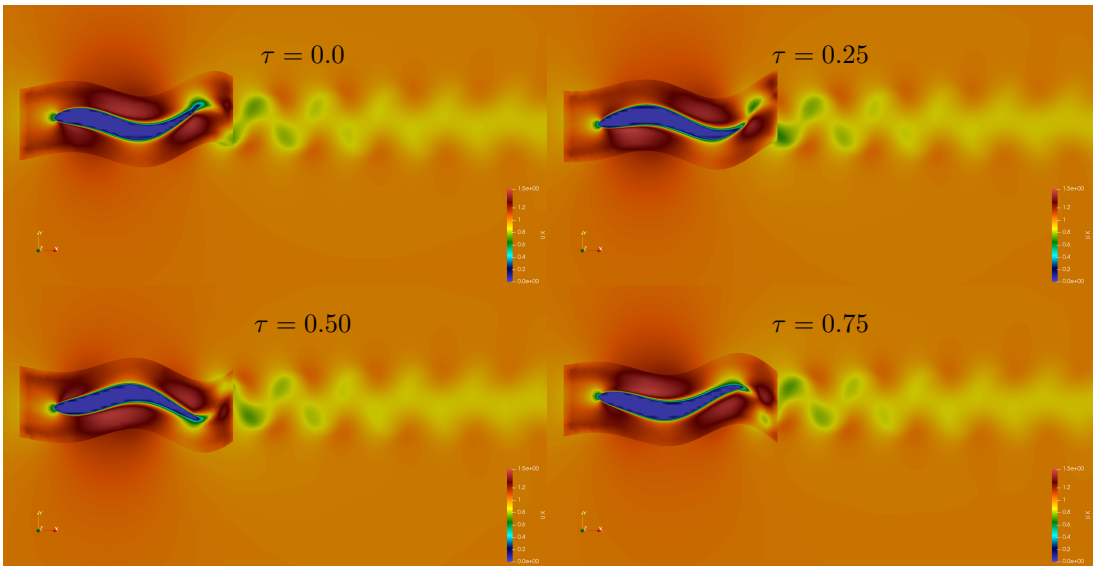
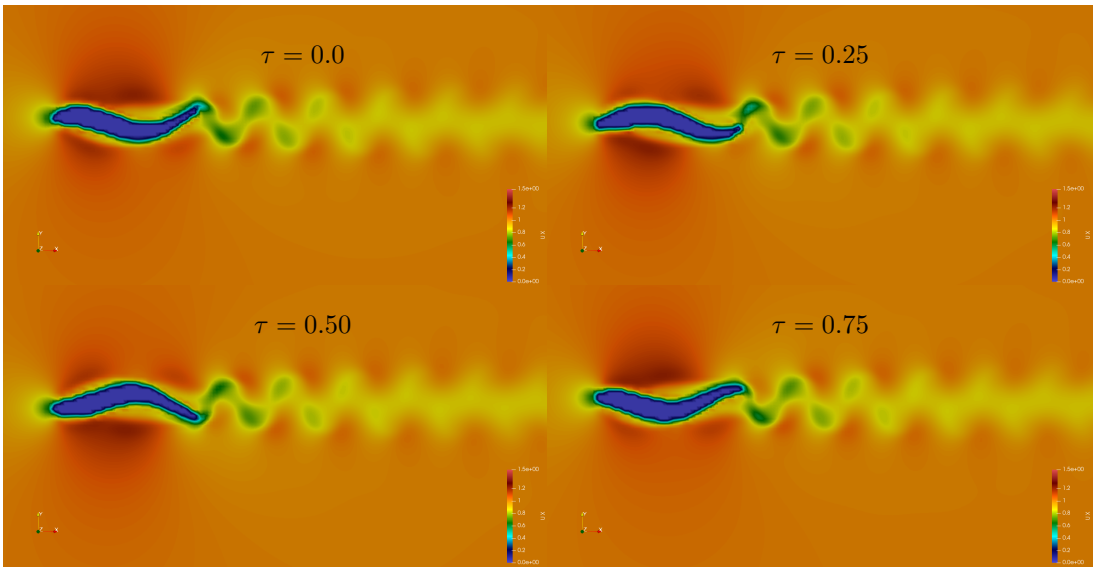


Figure 30. (A), (C) Swimmer at first half of cycle time, τ =0.6, 0.8. Anguilliform, Carangiform, Ostraciiform, and Thunniform modes from top to bottom. (B), (D) upper and lower pressure coefficient, C_P, for each locomotion mode plotted over the chord length. Solid lines represent the lower airfoil and dashed lines represent the upper airfoil.

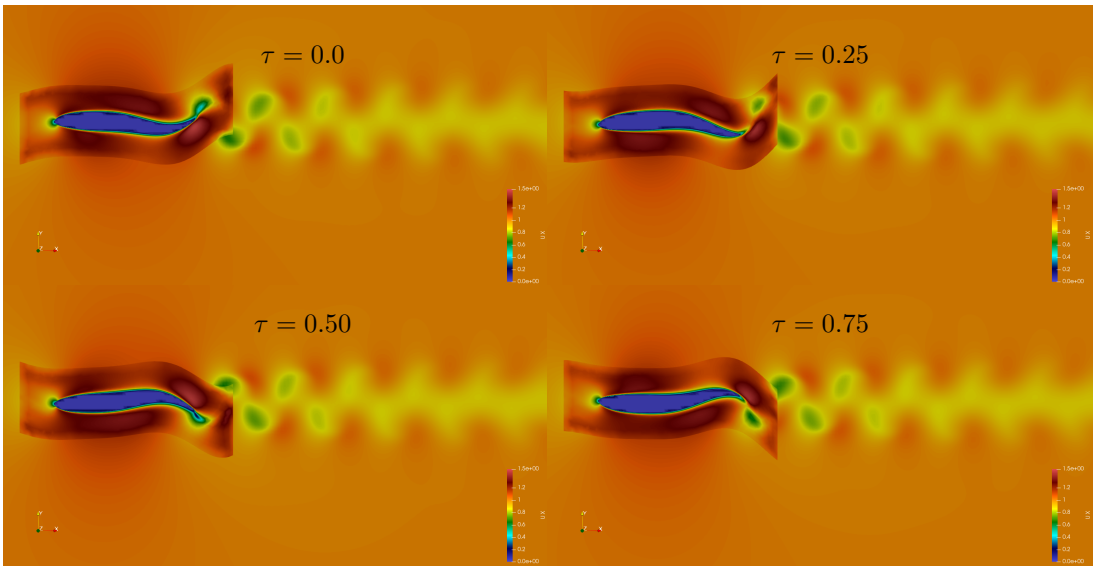


(a)

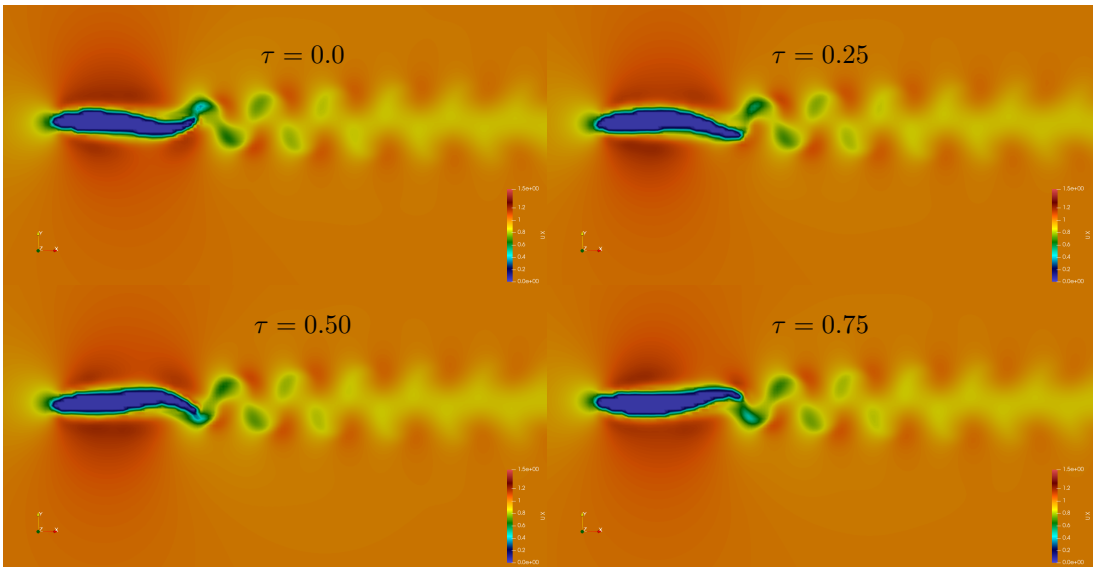


(b)

Figure 31. Velocity contours for Anguilliform swimmer with (A) overset mesh and (B) only background mesh.

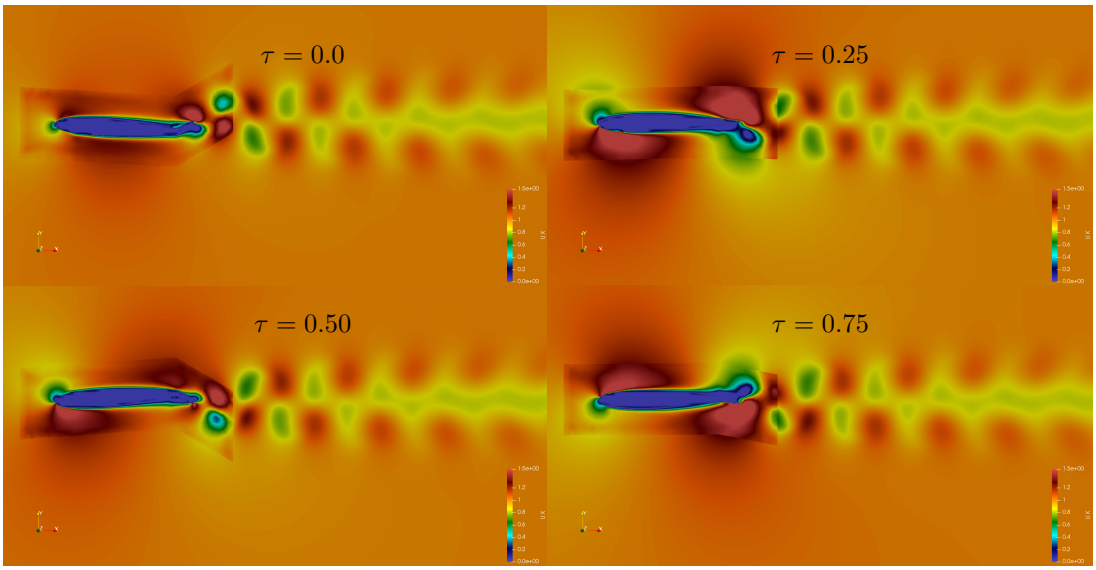


(a)

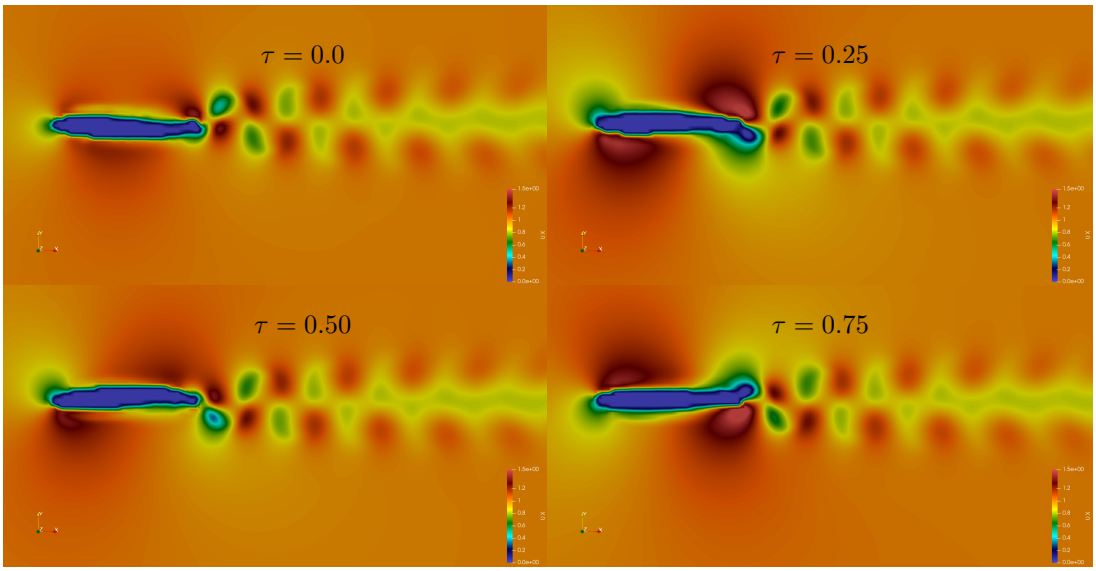


(b)

Figure 32. Velocity contours for Carangiform swimmer with (A) overset mesh and (B) only background mesh.

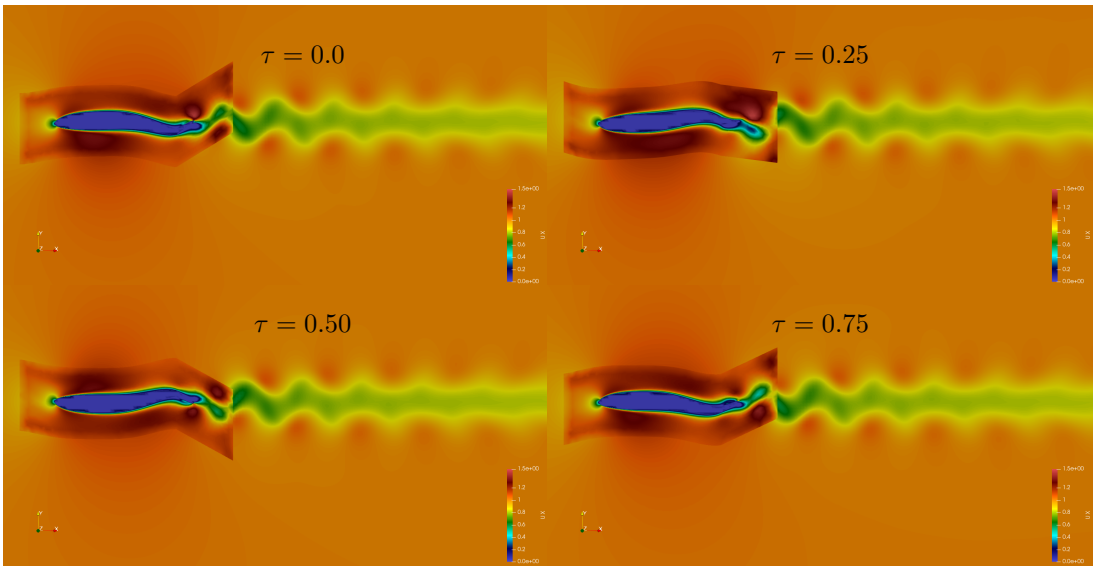


(a)

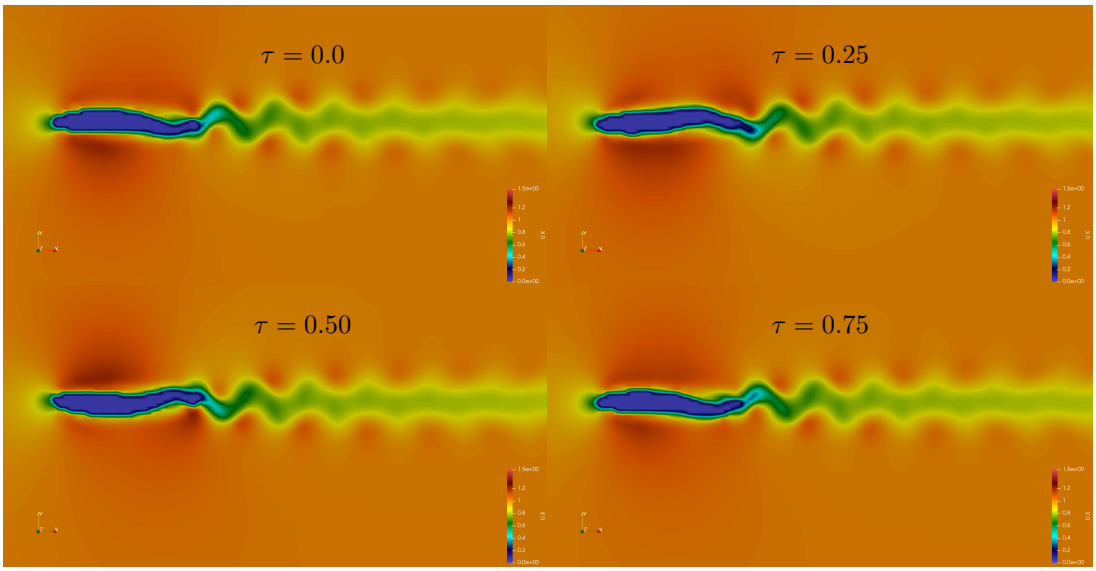


(b)

Figure 33. Velocity contours for Ostraciiform swimmer with (A) overset mesh and (B) only background mesh.



(a)



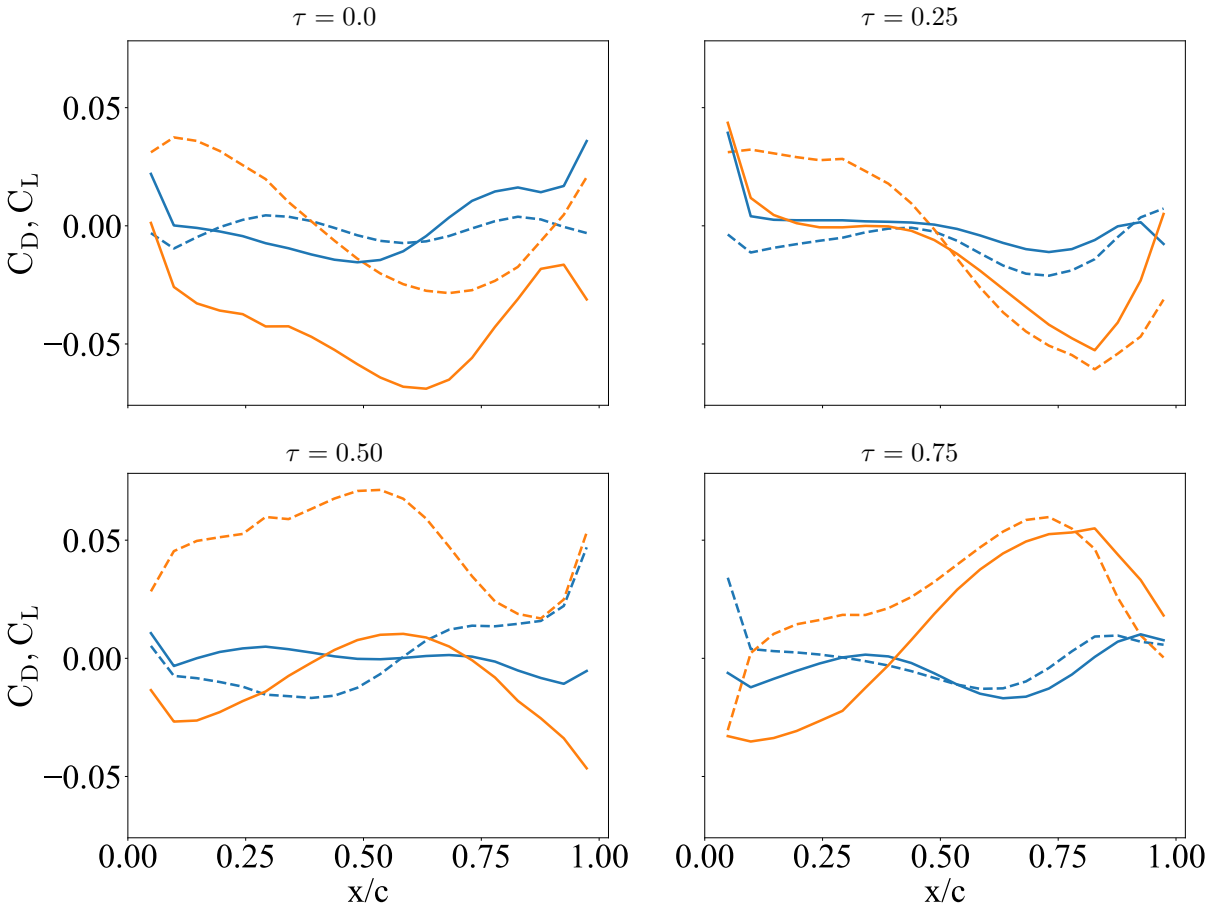
(b)

Figure 34. Velocity contours for Thunniform swimmer with (A) overset mesh and (B) only background mesh.

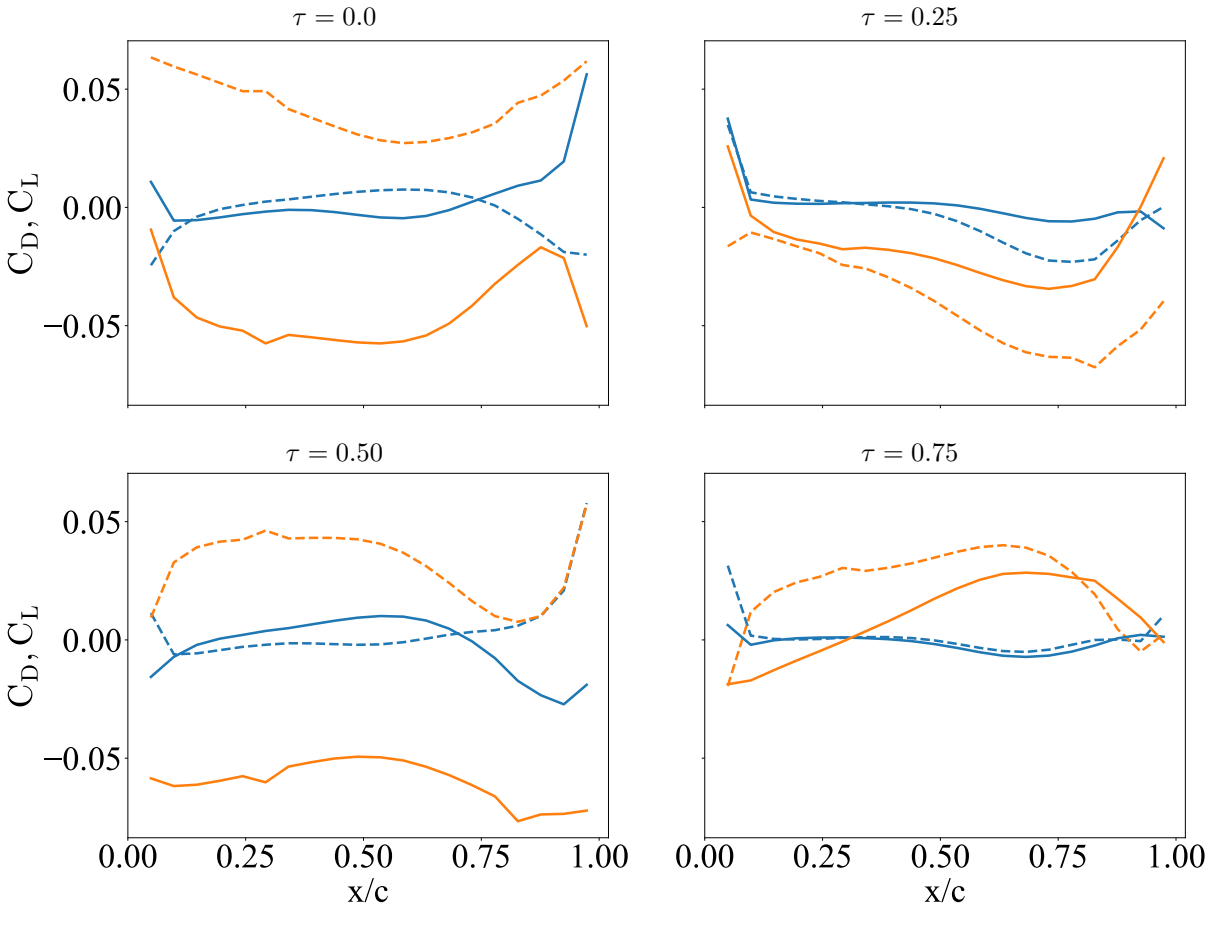
9.3.3. *Lift and Drag for Tail Beat Cycle.* To investigate the correlation between the wake field, C_D, and C_L, the span-wise lift and drag is plotted at different times in a tail beat cycle. Data was evenly sampled from the last cycle and forces were calculated using OpenFOAM® force function described by Eqn. 27 and Eqn. 28. Forces were then converted to a force coefficient using Eqn. 29.

Fig. 35 shows the C_D and C_L for the Anguilliform and Carangiform swimmers. Both swim modes have comparable magnitudes of lift and drag through a tail beat cycle. This can also be seen in the similar wake profile in the velocity contours of Fig. 31 and Fig. 32. Similar results are seen in the C_D profile for each swim mode, especially in the region of the head and tail. The Carangiform mode has some instances, in panel 1 of Fig. 35b, that show a higher drag on both the top and bottom airfoil. These instances contribute to the higher overall drag value of the Carangiform swim mode when compared with the Anguilliform swimmer in Fig. 28c.

Fig. 36 gives the C_D and C_L for the Ostraciiform and Thunniform locomotion modes. Like the Anguilliform/Carangiform case, these two modes have similar drag profiles through one tail beat cycle. The difference between these two is that the Ostraciiform produces around twice the amount of lift as the Thunniform mode, and around threefold the Anguilliform and Carangiform swim modes. This is confirmed when comparing Fig. 36a with Fig. 28d. This suggests that the undulation of the swimmer has more of an effect on lift than drag. This can be seen in Fig. 36, as the difference between the Ostraciiform and Thunniform modes is that the Thunniform swimmer undulates versus the Ostraciiform swimmer undergoing a small pitching moment.

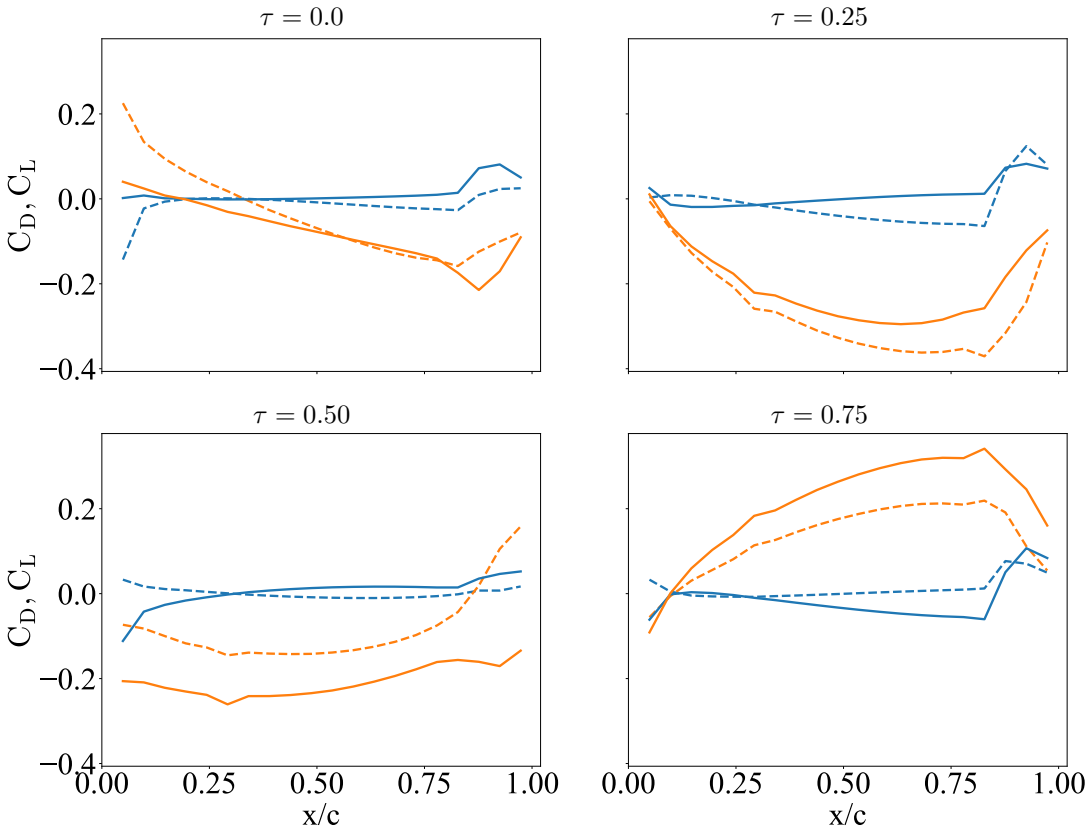


(a)

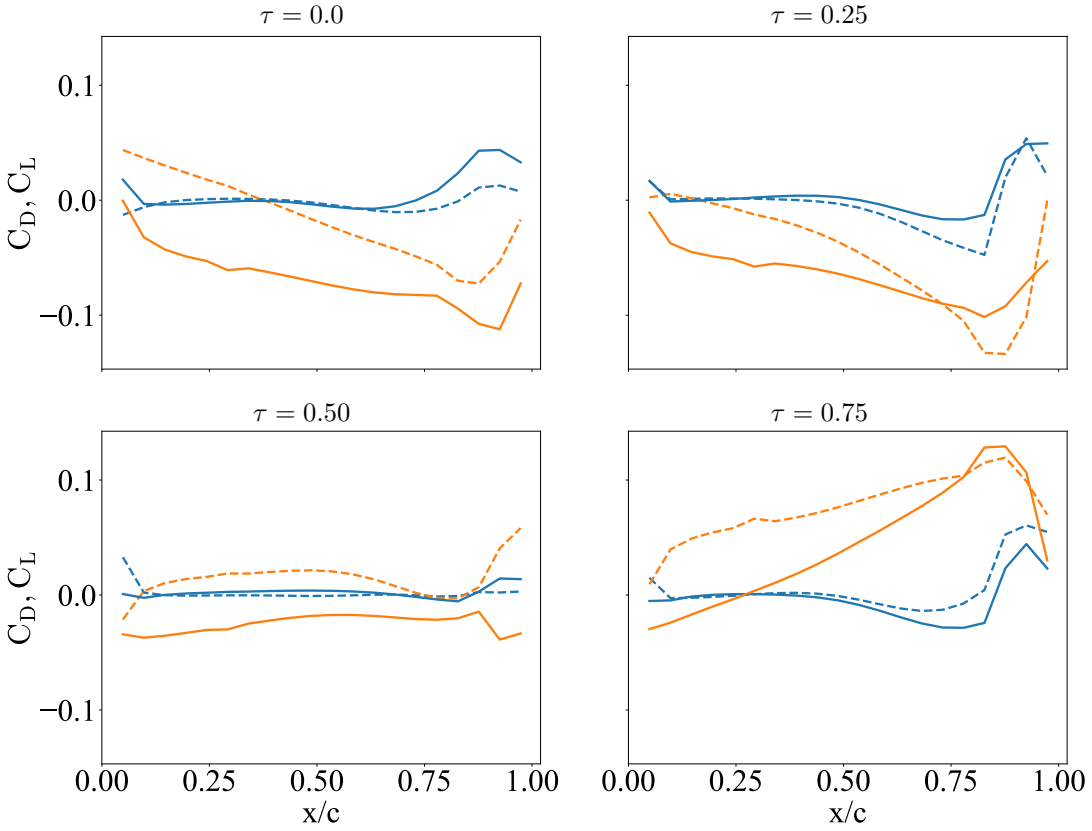


(b)

Figure 35. Span-wise C_D and C_L for (A) Anguilliform and (B) Carangiform swimmers. Solid lines represent the lower airfoil and dashed lines represent the upper airfoil. Blue lines represent C_D and the orange lines represent C_L.



(a)



(b)

Figure 36. Span-wise C_D and C_L for (A) Ostraciiform and (B) Thunniform swimmers. Solid lines represent the lower airfoil and dashed lines represent the upper airfoil. Blue lines represent C_D and the orange lines represent C_L.

9.4. **PIMPLE Iterations.** As discussed in section 7, the time marching scheme used in this research is the PISO or PIMPLE algorithm. Fig. 37 shows a comparison of the number of time marching steps in the PIMPLE algorithm, shown in Fig. 23 of section 7. In OpenFOAM®, the number of time marching steps is controlled by the nOuterCorrectors entry in the fvSolution file, under the system folder. The figure shows that after around 45 seconds of flow time, all simulations reach a steady state oscillation at close to the same time. Interestingly, the 5 and 25 iteration cases follow a similar trajectory with the 25 iteration case having an abnormal divergence at around 30 seconds. This indicates that steady state is independent of the number of time march iterations.

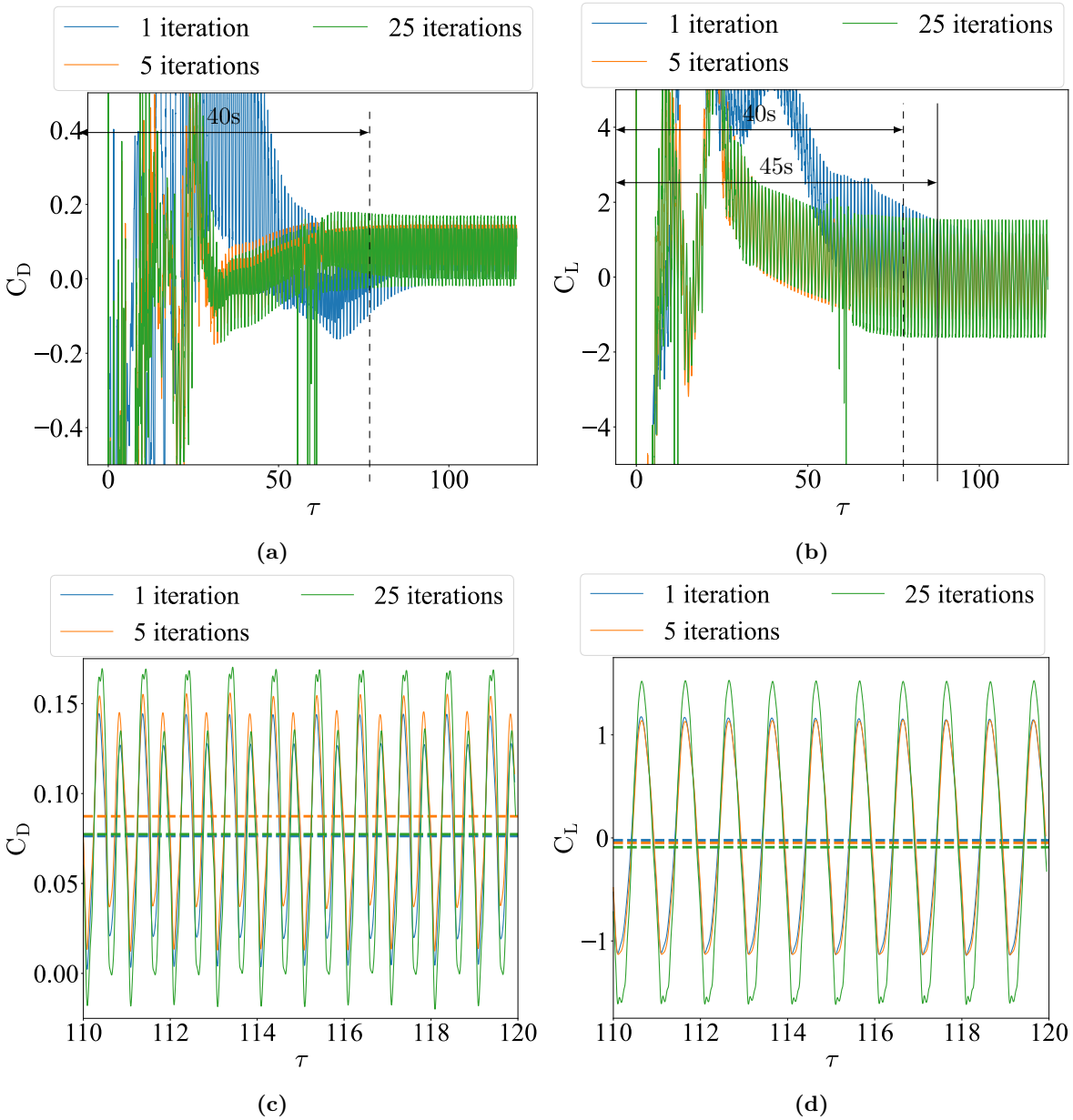


Figure 37. (A) C_D, (B) C_L, (C) C_D zoomed in, and (D) C_L zoomed in of Carangiform swimming for 60 seconds utilizing 1, 5, and 25 pimpleFoam iterations. Simulations are performed at Reynolds number $5 \cdot 10^3$ and at Strouhal number of 40.

Comparing the steady state oscillations of each case, all cases have similar average C_L. C_D shows the 1 iteration and 25 iteration cases having nearly the same average drag, with the 5 iteration case being slightly greater. In both drag and lift, the 25 iteration case has a larger envelope of oscillation. This may be caused by better convergence of the steady state solution compared to the 5 iteration case. The 1 iteration case is similar to the 5 iteration case, but with an offset, which may indicate that using time marching is not beneficial in this case unless using a larger number of iterations.

To determine the most appropriate number of iterations, the simulation time for each case is shown in Tab. 9. All simulations were performed on the same 32 core virtual machine using 8 cores. This is true for all reported times related to this research.

Table 9. Summary of simulation times for 1, 5, and 25 PIMPLE iterations (nOuterCorrectors). All simulations were performed using 8 CPU cores.

Case	Clock Time [s]	Clock Time [hr]	Clock Time [day]
1	223838	62.18	2.59
5	453748	126.04	5.25
25	1133369	314.82	13.12

Tab. 9 shows that the run time for each simulation increases as expected. Due to the amount of simulations needed to be performed and the amount of resources available, the remainder of this research uses the PISO algorithm or 1 PIMPLE iteration.

9.5. **Time Delay.** The amount of time it takes to reach steady state is a limitation on computational resources. To mitigate this, a time delay for fish movement was coded into the Ika-Flow solver package. The reasoning behind introducing a time delay is that the solution stabilizes to a steady-state value with no movements and then movement begins, therefore, there would not be large fluctuations in flow, as seen in the beginning of Fig. 37. Fig. 38 shows the same simulation with a delay of 5 and 15 seconds.

Interestingly, Fig. 38 indicates that C_D does reach steady state sooner with a time delay but C_L seems to be invariant to the delay. In both cases, the time delay does stabilize the forces quicker versus the case with no time delay shown in Fig. 37. This has the added benefit of reducing simulation time as shown in Tab. 10.

Table 10. Summary of simulation times the same simulation with no delay, a 5 second delay, and a 15 second delay. All simulations were performed using 8 CPU cores.

Case	Steady-state [s]	Clock Time [s]	Clock Time [hr]	Clock Time [day]
No Delay	40	453748.00	126.04	5.25
5s	20	338924.00	94.15	3.92
15s	35	337924.00	93.87	3.91

As the times for each simulation show, the clock time for the delayed cases is about 1.5 days shorter than for the non delayed case. The reason for the simulation time for the 15 s delayed case being shorter than the 5 s delayed case is that the simulation spent more flow time with zero movement, which is faster numerically. For the rest of the simulations, a delay time of 2 seconds was chosen due to being when the non-movement simulation stabilizes. The force data is then taken after 35 seconds of flow time for 5 tail beat cycles. This allows for the solution to reach steady state and gives enough cycles for averaging.

9.6. **Turbulence Models.** The discussion of turbulence models is considered because this research is concerned with the lift and drag of fish locomotion at Reynolds numbers between $5 \cdot 10^3 < \mathrm{Re} < 1 \cdot 10^8$. For this reason, a comparison is made between three turbulence models; the k-ω SST [63], the k-k_L-ω [65], and the Spalart-Allmaras (S-A) model [66]. The purpose of this comparison is to determine if one model is computationally faster than the other and if their results are comparable. The models are tested at two Reynolds numbers: $5 \cdot 10^3$ and $4 \cdot 10^5$.

9.6.1. *Model Comparisons.* Fig. 39 shows the comparison between the three turbulence models at different Reynolds numbers. The second Reynolds number, $4 \cdot 10^5$, is chosen due to being in the transitional region for flat plate flow [70]. At lower Reynolds numbers, all three models perform nearly the same, but in the transitional regime, the models diverge in drag, while being consistent in lift. Interestingly, higher Reynolds numbers show more oscillations in the forces than at lower Reynolds numbers. This may be caused by the resolution of the grid normal to the surface of the surface, known as the y+ value. For reference, the grid has a y+ value of less than 1 for Reynolds numbers up to $Re = 4 \cdot 10^7$ from flat plate theory discussed in Schlichting [70].

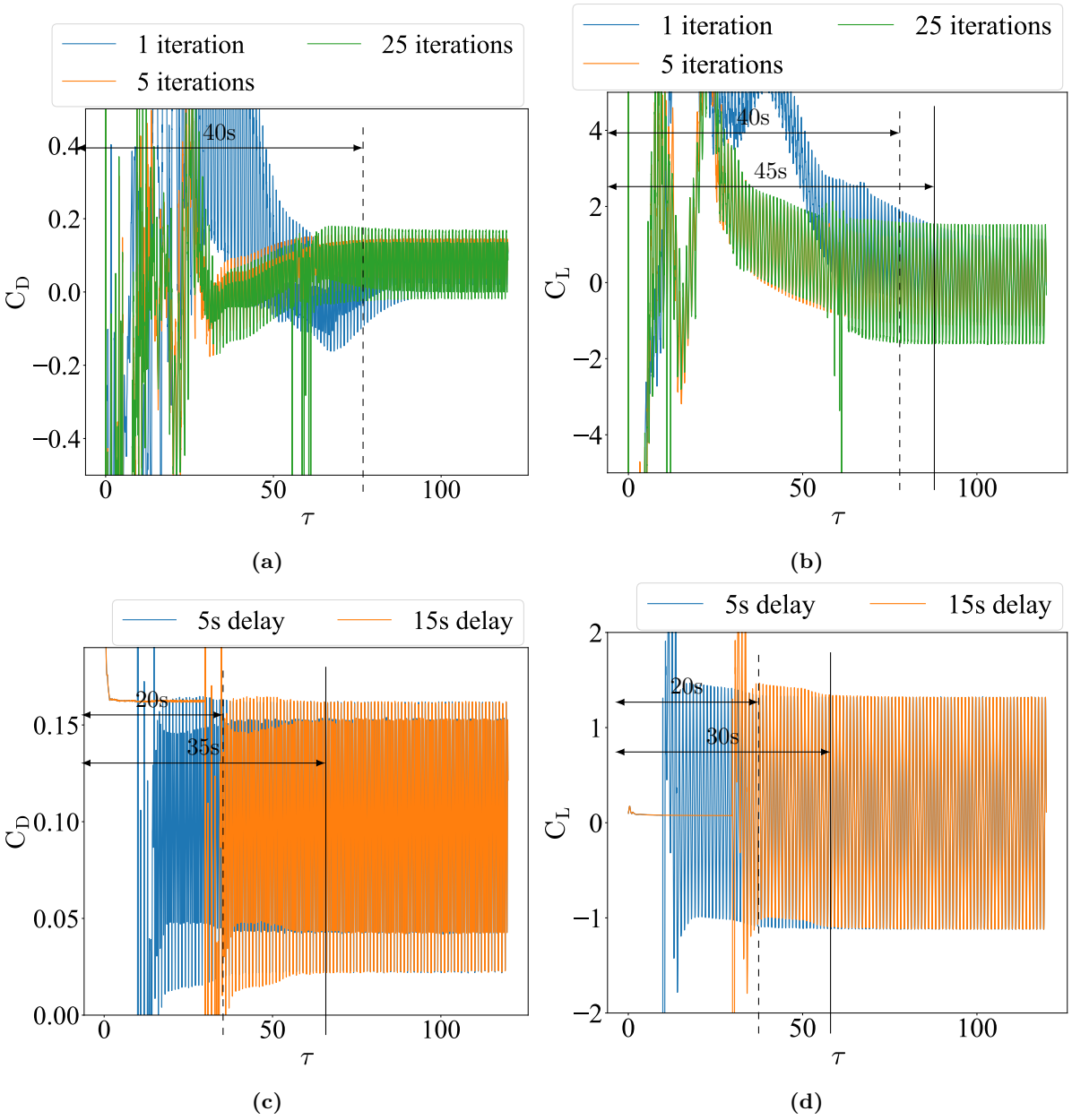
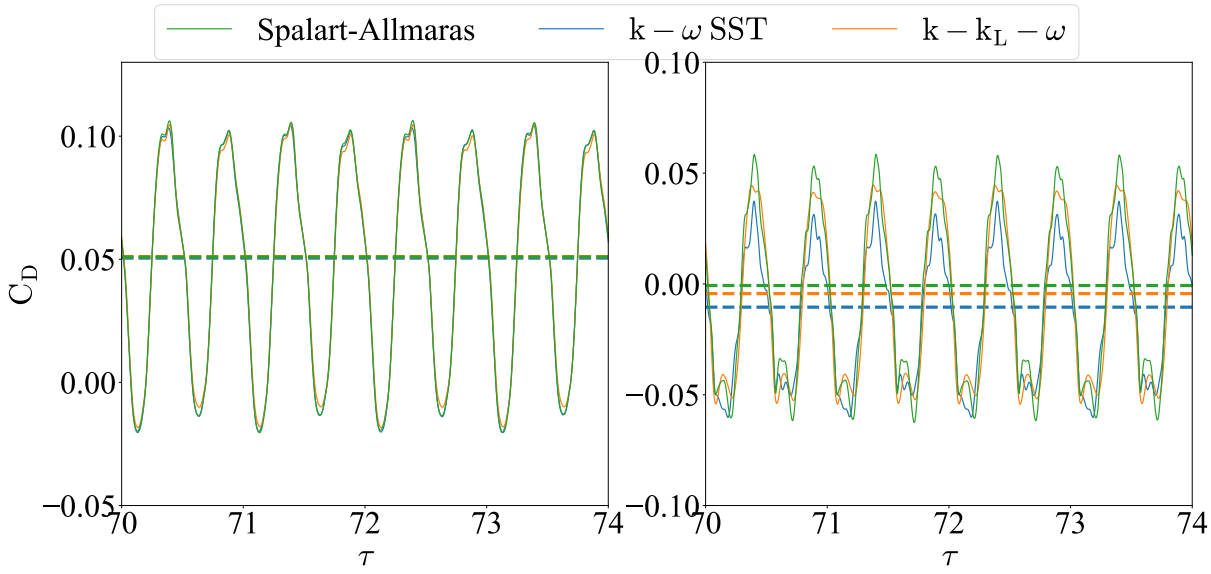


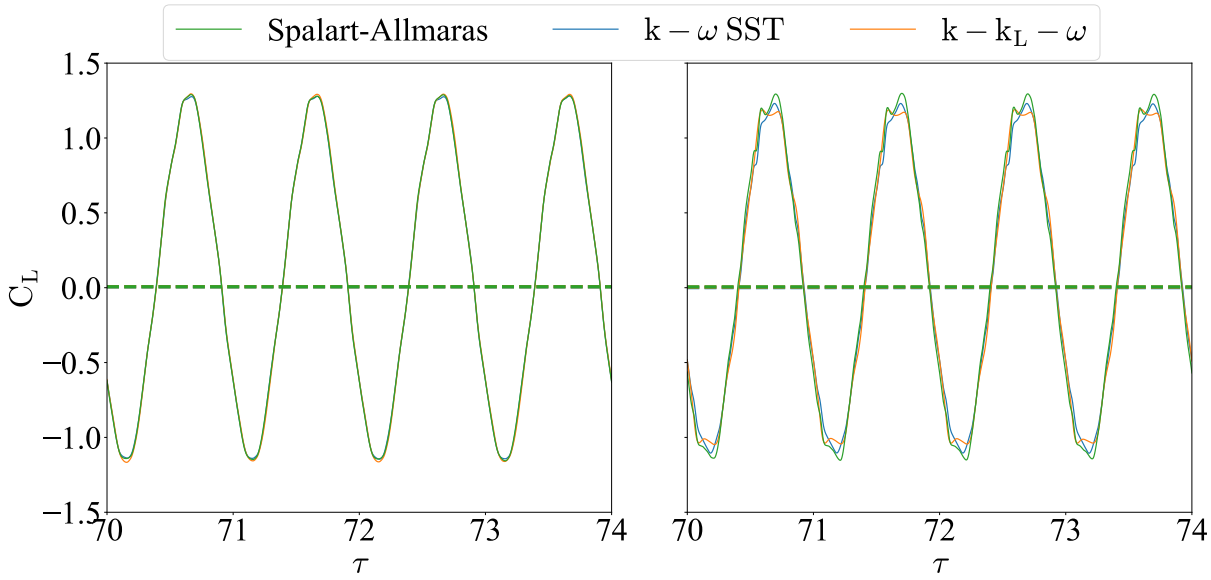
Figure 38. (A) C_D, (B) C_L for Carangiform swimming for 60 seconds utilizing 1, 5, and 25 pimpleFoam iterations. (C) C_D and (D) C_L for the same simulation with a delay of 5 and 15 seconds.

To the author's knowledge, there are no studies of fish swimming at Reynolds numbers exceeding $1 \cdot 10^5$ and, therefore, there is no data to validate these simulations. Experimental data on a rigid NACA 0012 airfoil at a Reynolds number of $Re = 6 \cdot 10^6$ are given in McCrosky [76]. McCrowsky's data shows that lift is nearly 0 and drag is about 0.0085 for an airfoil at 0 angle of attack. Here the XFOIL program developed by Drela [79], which can be used to determine the lift and drag at the specific Reynolds number of interest. It should be noted that XFOIL uses panel methods and the airfoil is not undulating for the XFOIL calculations. Tab. 11 shows the average lift and drag for the three turbulence models, XFOIL, and data from McCroskey [76].

Comparing the simulation data, the trend is a decreasing drag for an increasing Reynolds number. For the higher Reynold's case, the $k-\omega$ SST and Spalart-Allmaras model are significantly higher than the $k-k_L-\omega$ model. The expected trend for an airfoil is that drag will decrease as Reynolds number increases as given by empirical data in Hoerner [80]. All models follow this expected trend but $k-k_L-\omega$ model decreases significantly more than the others. It is difficult to discern which model is more accurate and further simulations will be needed to be done in future research. Discrepancies aside, the $k-\omega$ SST



(a)



(b)

Figure 39. (A) C_D and (B) C_L for different turbulence models. Simulations performed at Reynolds numbers 5e03 (left plots) and 4e05 (right plots) with a Strouhal number of 40.0.

model has been successfully used in a wide variety of engineering cases including fish hydrodynamics [14–21], which increases the confidence in this model. In order to assess if there are any time savings with the different models, the simulation times are presented in Tab. 12.

Tab. 12 show that all three turbulence models have similar computational times for low Reynolds numbers, but in higher Reynolds number applications, the Spalart-Allmaras model performs the most efficiently. This is consistent with a one equation model versus a two or three equation model as there are less equations to solve. Comparing the other two models, the $k - k_L - \omega$ is faster than the $k - \omega$ SST, but for reasons mentioned above, this research is utilizing the $k - \omega$ SST model.

Concerning different turbulence models for this application, the purpose of this work is to introduce the overset motion solver library. Fully characterizing multiple turbulence models would certainly be interesting but is out of scope for this research. A more complete characterization of each solver's results throughout a wide range of Reynolds numbers is a question which will be pursued in a future investigation.

Table 11. Lift C_L and drag C_D for three different turbulence models, XFOIL, and data from McCrosky [76] at Reynolds numbers of $5 \cdot 10^3$ and $4 \cdot 10^5$. All simulations were performed using 8 CPU cores.

Model	**Reynolds Number**	C_D	C_L
$k-\omega$ SST	$5 \cdot 10^3$	0.0656	0.0056
$k-k_L-\omega$	$5 \cdot 10^3$	0.0649	0.0045
Spallart-Allmaras	$5 \cdot 10^3$	0.0654	0.0004
$k-\omega$ SST	$4 \cdot 10^5$	0.0101	0.0072
$k-k_L-\omega$	$4 \cdot 10^5$	0.0039	0.0057
Spalart-Allmaras	$4 \cdot 10^5$	0.0137	0.0065
XFOIL	$4 \cdot 10^5$	0.0062	0.0000
McCrosky [76]	$6 \cdot 10^6$	0.0085	0.0000

Table 12. Summary of simulation times for three different turbulence models at Re = $[5 \cdot 10^3, 4 \cdot 10^5]$. All simulations were performed using 8 CPU cores.

Simulation	**Re**	**Clock Time [s]**	**Clock Time [hr]**	**Clock Time [day]**
$k-\omega$ SST	$5 \cdot 10^3$	220480.00	61.24	2.56
$k-k_L-\omega$	$5 \cdot 10^3$	214672.00	59.63	2.48
Spallart-Allmaras	$5 \cdot 10^3$	216337.00	60.09	2.50
$k-\omega$ SST	$4 \cdot 10^5$	335537.00	93.20	3.88
$k-k_L-\omega$	$4 \cdot 10^5$	310976.00	86.38	3.60
Spallart-Allmaras	$4 \cdot 10^5$	302305.00	83.97	3.50

10. **Concluding Remarks**

A motion library to simulate the BCF swimming modes of fish is presented. The library is able to accurately prescribe the locomotion for four types of swimmers. While this motion library was used to model fish motion, it can be adapted to model any complex motion that can be mathematically prescribed to the mesh points. This type of utility is currently not available in the OpenFOAM® overset mesh library, as it only supports rigid body motion.

Concerning the simulation of fish locomotion, it is shown that each swimming mode takes many cycles in order to reach steady-state, a behavior also observed by Borazjani and Sotiropoulos [17]. To overcome this limitation, a feature added to the new solver that requires a delay, to first stabilize the calculations, is proven to be effective at limiting the amount of flow time needed for the swimmer to reach a quasi-steady-state. This will decrease the amount of time needed to simulate fish motion for future studies.

A novel finding on the effect of undulation versus pitching on lift and drag is demonstrated by the comparison of the Ostraciiform and Thunniform swimming modes. To the authors' knowledge, the Ostraciiform swimming mode has not been simulated and compared in this manner. The lift and drag plots for these two swimmers show a marked effect on C_L over one tail beat cycle. Furthermore, the velocity field around and wake of the Ostraciiform swimmer, in Fig. 33, displays higher magnitude fluctuations than that of the Thunniform swimmer, in Fig. 34. This indicates that more energy is transferred into the free stream by the Ostraciiform swimmer.

It is shown that an overset mesh setup can be utilized to provide similar results to standard morphing meshes in comparison with literature [14, 16]. There exists a slight offset between the results using the overset versus body-fitted meshes. The average results, shown in Fig. 25, are within 5% of each other and the benefits of using an overset mesh for this particular case outweigh the small discrepancy with body-fit meshes. Utilizing an overset mesh for this problem allows for a larger time step and no negative cell volumes, which arise due to solving the diffusion equation for the mesh motion.

For the first time, a comparison of fish locomotion with three popular turbulence models for low and transitional Reynolds numbers is investigated and discussed in Fig. 39. The most common model in literature is the $k-\omega$ SST model [14–21]. At low Reynolds numbers, all three turbulence models perform similarly, but tend to diverge in the transitional regime. As there is no experimental or simulation data currently available to validate these models with, the $k-\omega$ SST model was chosen based on its popularity for resolving turbulence in a many external aerodynamic studies.

Acknowledgements

This work was made possible by the use of the RCC facilities at the University of Canterbury. The authors further wish to acknowledge the use of New Zealand eScience Infrastructure (NeSI) high performance computing facilities, consulting support and/or training services as part of this research. New Zealand's national facilities are provided by NeSI and funded jointly by NeSI's collaborator institutions and through the Ministry of Business, Innovation & Employment's Research Infrastructure programme. URL `https://www.nesi.org.nz`. Finally, the authors would like to acknowledge Paul Southward and Adam Latham of the Univeristy of Canterbury, Department of Mechanical Engineering, technical staff for computational resources for this work.

Author Contributions: Conceptualisation, M.C.; methodology, M.C.; software, M.C; validation, M.C.; formal analysis, M.C.; investigation, M.C.; resources, S.G.; data curation, M.C.; writing—original draft preparation, M.C.; writing—review and editing, M.C. and S.G.; visualisation, M.C. and S.G.; supervision, S.G.; project administration, S.G.; funding acquisition, S.G. All authors have read and agreed to the published version of the manuscript.

References

[1] C. C. Lindsey, "1 - form, function, and locomotory habits in fish," *Fish Physiology*, vol. 7, pp. 1–100, 1978. [Online]. Available: https://doi.org/10.1016/S1546-5098(08)60163-6

[2] Breder Charles M. (Charles Marcus), "The locomotion of fishes," *Zoologica : scientific contributions of the New York Zoological Society*, vol. 4, pp. 159–297, 1926. [Online]. Available: https://doi.org/10.5962/p.203769

[3] J. Gray, "Studies in animal locomotion VI. The propulsive powers of the dolphin," *Journal of experimental Bbology*, vol. 13, no. 2, pp. 192–199, 1936. [Online]. Available: https://doi.org/10.1242/jeb.13.2.192

[4] M. J. Lighthill, "Hydrodynamics of Aquatic Animal Propulsion," *Annual Review of Fluid Mechanics*, vol. 1, no. 1, pp. 413–446, 1969. [Online]. Available: https://doi.org/10.1146/annurev.fl.01.010169.002213

[5] ——, "Large-amplitude elongated-body theory of fish locomotion," *Proceedings of the Royal Society of London. Series B. Biological Sciences*, vol. 179, no. 1055, pp. 125–138, 1971. [Online]. Available: https://doi.org/10.1098/rspb.1971.0085

[6] T. Y.-T. Wu, "Hydromechanics of swimming propulsion. Part 1. Swimming of a two-dimensional flexible plate at variable forward speeds in an inviscid fluid," *Journal of Fluid Mechanics*, vol. 46, no. 2, pp. 337–355, 1971. [Online]. Available: https://doi.org/10.1017/s0022112071000570

[7] ——, "Hydromechanics of swimming propulsion. Part 2. Some optimum shape problems," *Journal of Fluid Mechanics*, vol. 46, pp. 521–544, 1971. [Online]. Available: https://doi.org/10.1017/S0022112071000685

[8] T. Y. T. Wu, "Hydromechanics of swimming propulsion. Part 3. Swimming and optimum movements of slender fish with side fins," *Journal of Fluid Mechanics*, vol. 46, no. 3, pp. 545–568, 1971. [Online]. Available: https://doi.org/10.1017/S0022112071000697

[9] S. Jung, "Swimming, flying, and diving behaviors from a unified2D potential model," *Scientific Reports*, vol. 11, no. 1, pp. 1–11, 2021. [Online]. Available: https://doi.org/10.1038/s41598-021-94829-7

[10] M. J. Lighthill, "Aquatic animal propulsion of high hydromechanical efficiency," *Journal of Fluid Mechanics*, vol. 44, no. 2, pp. 265–301, 1970. [Online]. Available: https://doi.org/10.1017/S0022112070001830

[11] C. E. Lan, "The unsteady quasi-vortex-lattice method with applications to animal propulsion," *Journal of Fluid Mechanics*, vol. 93, no. 4, pp. 747–765, 1979. [Online]. Available: https://doi.org/10.1017/S0022112079002019

[12] H. Kagemoto, M. J. Wolfgang, D. K. Yue, and M. S. Triantafyllou, "Force and power estimation in fish-like locomotion using a vortex-lattice method," *Journal of Fluids Engineering, Transactions of the ASME*, vol. 122, no. 2, pp. 239–253, 2000. [Online]. Available: https://doi.org/10.1115/1.483251

[13] G. Xue, Y. Liu, M. Zhang, and H. Ding, "Numerical Analysis of Hydrodynamics for Bionic Oscillating Hydrofoil Based on Panel Method," *Applied Bionics and Biomechanics*, vol. 2016, 2016. [Online]. Available: https://doi.org/10.1155/2016/6909745

[14] Y. L. Yu and K. J. Huang, "Scaling law of fish undulatory propulsion," *Physics of Fluids*, vol. 33, no. 6, p. 1ENG, 2021. [Online]. Available: https://doi.org/10.1063/5.0053721

[15] A. P. Maertens, A. Gao, and M. S. Triantafyllou, "Optimal undulatory swimming for a single fish-like body and for a pair of interacting swimmers," *Journal of Fluid Mechanics*, vol. 813, pp. 301–345, 2017. [Online]. Available: https://doi.org/10.1017/jfm.2016.845

[16] G. J. Dong and X. Y. Lu, "Characteristics of flow over traveling wavy foils in a side-by-side arrangement," *Physics of Fluids*, vol. 19, no. 5, pp. 1–11, 2007. [Online]. Available: https://doi.org/10.1063/1.2736083

[17] I. Borazjani and F. Sotiropoulos, "Numerical investigation of the hydrodynamics of carangiform swimming in the transitional and inertial flow regimes," *Journal of Experimental Biology*, vol. 211, no. 10, pp. 1541–1558, 2008. [Online]. Available: https://doi.org/10.1242/jeb.015644

[18] ——, "On the role of form and kinematics on the hydrodynamics of self-propelled body/caudal fin swimming," *Journal of Experimental Biology*, vol. 213, no. 1, pp. 89–107, 2010. [Online]. Available: https://doi.org/10.1242/jeb.030932
[19] M. Abbaspour and M. Ebrahimi, "Comparative numerical analysis of the flow pattern and performance of a foil in flapping and undulating oscillations," *Journal of Marine Science and Technology (Japan)*, vol. 20, no. 2, pp. 257–277, 2015. [Online]. Available: https://doi.org/10.1007/s00773-014-0297-7
[20] M. Ebrahimi and M. Abbaspour, "Numerical investigation of the forward and backward travelling waves through an undulating propulsor: performance and wake pattern," *Ships and Offshore Structures*, vol. 11, no. 5, pp. 517–539, 2016. [Online]. Available: https://doi.org/10.1080/17445302.2015.1030248
[21] S. Gupta, A. Sharma, A. Agrawal, M. C. Thompson, and K. Hourigan, "Hydrodynamics of a fish-like body undulation mechanism: Scaling laws and regimes for vortex wake modes," *Physics of Fluids*, vol. 33, no. 10, 2021. [Online]. Available: https://doi.org/10.1063/5.0062304
[22] N. Li, H. Liu, and Y. Su, "Numerical study on the hydrodynamics of thunniform bio-inspired swimming under self-propulsion," *PLOS ONE*, vol. 12, no. 3, pp. 1–36, 2017. [Online]. Available: https://doi.org/10.1371/journal.pone.0174740
[23] J. J. Videler, *Fish Swimming*. Springer Netherlands, 1993. [Online]. Available: https://doi.org/10.1007/978-94-011-1580-3
[24] R. N. Govardhan and J. H. Arakeri, "Fluid mechanics of aquatic locomotion at large Reynolds numbers," *Journal of the Indian Institute of Science*, vol. 91, no. 3, pp. 329–353, 2011.
[25] M. Gazzola, M. Argentina, and L. Mahadevan, "Scaling macroscopic aquatic locomotion," *Nature Physics*, vol. 10, no. 10, pp. 758–761, 2014. [Online]. Available: https://doi.org/10.1038/nphys3078
[26] Y. Zheng and M. S. Liou, "A novel approach of three-dimensional hybrid grid methodology: Part 1. Grid generation," *Computer Methods in Applied Mechanics and Engineering*, vol. 192, no. 37-38, pp. 4147–4171, 2003. [Online]. Available: https://doi.org/10.1016/S0045-7825(03)00385-2
[27] J. Guerrero, "Overset composite grids for the simulation of complex moving geometries," in *EUA4X. European Atelier for Engineering and Computational Sciences*, 2006, pp. 1–14.
[28] D. Chandar and H. Gopalan, "Comparative analysis of the arbitrary mesh interface(AMI), generalized grid interface (GGI) and overset methods for dynamic body motions in openFOAM," *46th AIAA Fluid Dynamics Conference*, pp. 1–12, 2016. [Online]. Available: https://doi.org/10.2514/6.2016-3324
[29] D. D. Chandar and V. B. Boppana, "A Comparative Study of Different Overset Grid Solvers Between OpenFOAM, STAR-CCM+ and ANSYS-Fluent," *AIAA Aerospace Sciences Meeting, 2018*, no. 210059, 2018. [Online]. Available: https://doi.org/10.2514/6.2018-1564
[30] P. Laws, J. S. Saini, A. Kumar, T. Dadri, and T. Dadri, "Performance Analysis of Two-Dimensional NACA 0018 Airfoil Using an Overset Grid Solver," in *20th Annual CFD Symposium*, 2018, pp. 1–10.
[31] D. D. Chandar, "On overset interpolation strategies and conservation on unstructured grids in OpenFOAM," *Computer Physics Communications*, vol. 239, pp. 72–83, 2019. [Online]. Available: https://doi.org/10.1016/j.cpc.2019.01.009
[32] S. Takahash, I. Monjugawa, and K. Nakahash, "Unsteady flow computations around moving airfoils by overset unstructured grid method," *Transactions of the Japan Society for Aeronautical and Space Sciences*, vol. 51, no. 172, pp. 78–85, 2008. [Online]. Available: https://doi.org/10.2322/tjsass.51.78
[33] S. Deng, T. Xiao, B. van Oudheusden, and H. Bijl, "A dynamic mesh strategy applied to the simulation of flapping wings," *International Journal for Numerical Methods in Engineering*, vol. 106, no. January, pp. 664–680, 2016. [Online]. Available: https://doi.org/10.1002/nme
[34] G. Loubimov and M. P. Kinzel, "A cfd-based assessment of undulating propulsion," *AIAA Aviation 2019 Forum*, no. June, pp. 1–15, 2019. [Online]. Available: https://doi.org/10.2514/6.2019-3583
[35] Ansys Inc., "Ansys meshing." [Online]. Available: https://www.ansys.com/products/meshing
[36] Siemens Digital Industries Software, "Simcenter STAR-CCM+, version 2022," Siemens 2022.
[37] M. Sfakiotakis, D. M. Lane, and J. B. C. Davies, "Review of fish swimming modes for aquatic locomotion," *IEEE Journal of Oceanic Engineering*, vol. 24, no. 2, pp. 237–252, 1999. [Online]. Available: https://doi.org/10.1109/48.757275
[38] F. Fish and G. Lauder, "Passive and active flow control by swimming fishes and mammals," *Annual Review of Fluid Mechanics*, vol. 38, no. 1, pp. 193–224, 2006. [Online]. Available: https://doi.org/10.1146/annurev.fluid.38.050304.092201
[39] B. O. Community, *Blender - a 3D modelling and rendering package*, Blender Foundation, Stichting Blender Foundation, Amsterdam, 2021. [Online]. Available: http://www.blender.org
[40] E. D. Tytell, "Experimental hydrodynamics of swimming in fishes," Ph.D. dissertation, Harvard University, Massachusetts, Nov. 2005. [Online]. Available: https://ui.adsabs.harvard.edu/abs/2005PhDT.....100T
[41] M. S. Triantafyllou, G. S. Triantafyllou, and R. Gopalkrishnan, "Wake mechanics for thrust generation in oscillating foils," *Physics of Fluids A*, vol. 3, no. 12, pp. 2835–2837, 1991. [Online]. Available: https://doi.org/10.1063/1.858173
[42] M. S. Triantafyllou and G. S. Triantafyllou, "An efficient swimming machine," *Scientific American*, vol. 272, no. 3, pp. 40–48, 1995. [Online]. Available: https://doi.org/10.1038/scientificamerican0395-64
[43] E. D. Tytell, "The hydrodynamics of eel swimming II. Effect of swimming speed," *Journal of Experimental Biology*, vol. 207, no. 19, pp. 3265–3279, 2004. [Online]. Available: https://doi.org/10.1242/jeb.01139
[44] J. Videler and B. Nolet, "Costs of Swimming Measured At Optimum Speed: Scale Effects, Differences Between Swimming Styles, Taxonomic Groups and Submerged and Surface Swimming," *Comparative biochemistry and physiology*, vol. 97, no. 2, pp. 91–99, 1990. [Online]. Available: https://doi.org/10.1016/0300-9629(90)90155-l
[45] H. Dewar, "Studies of tropical tuna swimming performance: Thermoregulation, swimming mechanics and energetics," Ph.D. dissertation, University of California, San Diego, 1993.
[46] D. Xia, J. Liu, W. Chen, and L. Han, "Hydrodynamic analysis of fishlike robot swimming in the straight forward way," *2009 IEEE International Conference on Mechatronics and Automation, ICMA 2009*, pp. 3342–3347, 2009. [Online]. Available: https://doi.org/10.1109/ICMA.2009.5246298

[47] J. Carling, T. L. Williams, and G. Bowtell, "Self-propelled anguilliform swimming: Simultaneous solution of the two-dimensional Navier-Stokes equations and Newton's laws of motion," *Journal of Experimental Biology*, vol. 201, no. 23, pp. 3143–3166, 1998. [Online]. Available: https://doi.org/10.1242/jeb.201.23.3143

[48] N. Thekkethil, A. Sharma, and A. Agrawal, "Self-propulsion of fishes-like undulating hydrofoil: A unified kinematics based unsteady hydrodynamics study," *Journal of Fluids and Structures*, vol. 93, p. 102875, 2020. [Online]. Available: https://doi.org/10.1016/j.jfluidstructs.2020.102875

[49] B. Y. J. J. Videler and F. Hess, "Fast Continuous Swimming of Two Pelagic Predators, Saithe (Pollachius Virens) and Mackerel (Scomber Scombrus): a Kinematic Analysis," *Journal of Experimental Biology*, vol. 109, no. MAR, pp. 209–228, 1984.

[50] D. Scaradozzi, G. Palmieri, D. Costa, and A. Pinelli, "BCF swimming locomotion for autonomous underwater robots: a review and a novel solution to improve control and efficiency," *Ocean Engineering*, vol. 130, no. October 2016, pp. 437–453, 2017. [Online]. Available: https://doi.org/10.1016/j.oceaneng.2016.11.055

[51] D. S. Barrett, "Propulsive Efficiency of a Flexible Hull Underwater Vehicle," Ph.D. dissertation, Massachussetts Institute of Technology, 1996.

[52] J. Zhu, C. White, D. K. Wainwright, V. D. Santo, G. V. Lauder, and H. Bart-Smith, "Tuna robotics: A high-frequency experimental platform exploring the performance space of swimming fishes," *Science Robotics*, vol. 4, 9 2019. [Online]. Available: https://doi.org/10.1126/scirobotics.aax4615

[53] OpenCFD Ltd (ESI Group), "OpenFOAM® v2012." [Online]. Available: http://www.openfoam.com

[54] F. Weiler, "Numerical investigation of subcarangiform swimming as a model for future propulsion systems," Master's thesis, Hochschule Bremen City University of Applied Sciences, June 2018.

[55] O. D. Mejia, O. E. Mejia, K. M. Escorcia, F. Suarez, and S. Laín, "Comparison of sliding and overset mesh techniques in the simulation of a vertical axis turbine for hydrokinetic applications," *Processes*, vol. 9, no. 11, 2021. [Online]. Available: https://doi.org/10.3390/pr9111933

[56] T. Petra, "Description of the overset mesh approach in esi version of openfoam," in *Proceedings of CFD with OpenSource Software*, N. H., Ed., 2019. [Online]. Available: https://doi.org/10.17196/OS_CFD#YEAR_2019

[57] J. Ahrens, B. Geveci, and C. Law, *ParaView: An End-User Tool for Large Data Visualization*. Elsevier, 2005.

[58] W. M. Chan, R. J. Gomez III, S. E. Rogers, and P. G. Buning, "Best Practices in Overset Grid Generation," *American Institute of Aeronautics and Astronautics*, vol. 3191, no. June, 2002. [Online]. Available: https://doi.org/10.2514/6.2002-3191

[59] J. Guerrero, "OpenFOAM advanced training. Moving meshes, rigid body motion, adaptive mesh refinement, and overset meshes," 2022. [Online]. Available: https://doi.org/10.6084/m9.figshare.19310492.v1

[60] F. Goetten, D. Felix, M. Marino, C. Bil, M. Havermann, and C. Braun, "A review of guidelines and best practices for subsonic aerodynamic simulations using RANS CFD," *11th Asia-Pacific International Symposium of Aerospace Technology*, no. December, pp. 227–245, 2019.

[61] P. M. Knupp, "Algebraic Mesh Quality Metrics," *Jounral of Scientific Computing*, vol. 23, no. 1, pp. 193–218, 2001. [Online]. Available: https://doi.org/10.1137/S1064827500371499

[62] J. Arlow and I. Neustadt, "Uml 2 and the unified process: Practical object-oriented analysis and design," *Journal of Object Technology - JOT*, 01 2002.

[63] F. R. Menter, M. Kuntz, and R. Langtry, "Ten Years of Industrial Experience with the SST Turbulence Model," in *Proceedings of the fourth international symposium on turbulence, heat and mass transfer*, vol. 4, 2003, pp. 625–632.

[64] A. Meana-Fernández, J. Fernández Oro, K. Argüelles Díaz, and S. Velarde-Suárez, "Turbulence-Model Comparison for Aerodynamic-Performance Prediction of a Typical Vertical-Axis Wind-Turbine Airfoil," *Energies*, vol. 12, no. 3, p. 488, 2019. [Online]. Available: https://doi.org/10.3390/en12030488

[65] J. Fürst, J. Píhoda, and P. Straka, "Numerical simulation of transitional flows," *Computing*, vol. 95, no. SUPPL.1, 2013. [Online]. Available: https://doi.org/10.1007/s00607-012-0266-0

[66] P. R. Spalart and S. R. Allmaras, "One-equation turbulence model for aerodynamic flows," *Recherche aerospatiale*, no. 1, pp. 5–21, 1994. [Online]. Available: https://doi.org/10.2514/6.1992-439

[67] M. Gokdepe, "Turbulence Models for the Numerical Prediction of Transitional Flows with RANSE," *Turkish Naval Academy*, 2011. [Online]. Available: http://dspace.mit.edu/handle/1721.1/7582

[68] "Menter shear stress transport model," accessed: 2021-09-20. [Online]. Available: https://turbmodels.larc.nasa.gov/sst.html

[69] "Turbulence free-stream boundary conditions," accessed: 2021-09-20. [Online]. Available: https://www.cfd-online.com/Wiki/Turbulence_free-stream_boundary_conditions

[70] H. Schlichting and K. Gersten, *Boundary-layer Theory*, 9th ed. Heidelberg: Springer Belin,, 2017.

[71] R. Courant, K. Friedrichs, and H. Lewy, "On the Partial Difference Equations of Mathematical Physics," *IBM Journal of Research and Development*, vol. 11, no. 2, pp. 215–234, 1967. [Online]. Available: https://doi.org/10.1147/rd.112.0215

[72] C. Rogers, "Computational Fluid Dynamics Analysis of an Ideal Anguilliform Swimming Motion," Masters, University of New Orleans, 2014. [Online]. Available: https://scholarworks.uno.edu/td/1940/

[73] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant, "Array programming with NumPy," *Nature*, vol. 585, no. 7825, pp. 357–362, 2020. [Online]. Available: https://doi.org/10.1038/s41586-020-2649-2

[74] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. Jarrod Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. Carey, İ. Polat, Y. Feng, E. W. Moore, J. Vand erPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and S. . . Contributors, "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python," *Nature Methods*, vol. 17, pp. 261–272, 2020.

[75] I. B. Celik, U. Ghia, P. J. Roache, C. J. Freitas, H. Coleman, and P. E. Raad, "Procedure for estimation and reporting of uncertainty due to discretization in CFD applications," *Journal of Fluids Engineering, Transactions of the ASME*, vol. 130, no. 7, pp. 0 780 011–0 780 014, 2008. [Online]. Available: https://doi.org/10.1115/1.2960953

[76] W. J. Mccroskey, "A Critical Assessment of Wind Tunnel Results for the NACA 0012 Airfoil," *NASA technical Memorandum 100019 USAAVSCOM Technical Report 87-A-5*, 1987. [Online]. Available: https://ntrs.nasa.gov/citations/19880002254

[77] D. S. Barrett, M. S. Triantafyllou, D. K. Yue, M. A. Grosenbaugh, and M. J. Wolfgang, "Drag reduction in fish-like locomotion," *Journal of Fluid Mechanics*, vol. 392, pp. 183–212, 8 1999. [Online]. Available: https://doi.org/10.1017/S0022112099005455

[78] U. Ayachit, *The ParaView Guide: A Parallel Visualization Application*. Clifton Park, NY, USA: Kitware, Inc., 2015.

[79] M. Drela, "XFOIL: an analysis and design system for low Reynolds number airfoils." in *Low Reynolds Number Aerodynamics*, T. J. Mueller, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 1989, pp. 1–12. [Online]. Available: https://doi.org/10.1007/978-3-642-84010-4_1

[80] S. F. Hoerner, *Fluid-Dynamic Drag*, 2nd ed. P.O. Box 342, Brick Town, N.J. 08723: Hoerner Fluid Dynamics, 7 1965.