

# Computational framework for efficient high-fidelity optimization of bio-inspired propulsion and its application to accelerating swimmers



Ahmed Abouhoussein\*, Yulia T. Peet

School for Engineering of Matter, Transport and Energy, Arizona State University, Tempe, AZ 85287, USA

## ARTICLE INFO

### Article history:

Received 29 July 2021

Received in revised form 3 February 2023

Accepted 25 February 2023

Available online 6 March 2023

### Keywords:

Surrogate-based optimization

Spectral-element method

Bio-inspired propulsion

Accelerating fish

Polynomial refinement

Scaling laws

## ABSTRACT

A new computational framework for high-fidelity optimization of kinematic gaits during self-propelled undulatory swimming is developed. A computational framework utilizes a spectral-element method on moving body-fitted grids for a simulation of self-propelled swimming, and a surrogate-based optimization (SBO) procedure. A new volume-conservation method for reconstruction of a swimmer's geometry during the undulatory motion is proposed to ensure numerical stability of the fluid-structure interaction solver in an incompressible flow framework. A surrogate-based optimization algorithm that utilizes a Kriging response surface method is adopted and further developed in this work to manage the optimization process in the presence of physiological constraints on the fish body motion. A grid convergence of the optimization results is established, and the influence of the polynomial refinement on the results of optimization procedure is assessed. The increase in polynomial order does not change the optimum gaits of locomotion or relative efficiency rankings between the modes, but it results in slightly lower predicted efficiency for all the modes. The optimum solution is characterized by a kinematic gait that generates the reverse Karman vortex street associated with high propulsive efficiency. Efficiency of sub-optimum modes is found to increase with both the tail amplitude and the effective flapping length of the swimmer, and a new scaling law is proposed to capture these trends. Lastly, the SBO algorithm converged to an optimized gate with significantly less function evaluations than typically observed for evolutionary algorithms. This suggests that the SBO framework is a well suited alternative for high-fidelity optimization of fluid and structure problems.

© 2023 Elsevier Inc. All rights reserved.

## 1. Introduction

Biomimetic propulsion has triggered the interests of researchers for several decades, since it not only relates to the processes of evolution and natural selection in aquatic species, but also provides an inspiration for engineering and design of agile underwater devices [1–3]. In this aspect, identification of optimum, in some pre-defined sense, modes of locomotion is especially intriguing, since it gives a clear pathline towards design of bio-inspired vehicles that aim to provide a corresponding optimality [4]. While parametric studies of aquatic biolocomotion can give some important insights towards

\* Corresponding author.

E-mail addresses: [aabouhus@asu.edu](mailto:aabouhus@asu.edu) (A. Abouhoussein), [yppet@asu.edu](mailto:yppet@asu.edu) (Y.T. Peet).

understanding of swimming hydrodynamics [5,6], in this work we are interested in developing formal optimization techniques that can automatically select and refine the candidates for optimality, given certain design criteria [7]. Specifically, the goal of this study is to develop efficient and accurate numerical framework for optimization of bio-inspired propulsion that utilizes fully-resolved computational fluid dynamics (CFD) simulations.

There have been several studies devoted to optimization of aquatic swimmers in the previous literature [8–13]. All of them, with the exception of Ref. [13], used the so-called evolutionary algorithms (EA), which rely on a nature-inspired method to select and “evolve” potentially favorable solution candidates towards optimality [14]. Individuals, or solution candidates, in a population evolve using a variety of biologically inspired processes such as mutation, crossover and recombination [15]. However, most evolutionary, or genetic, algorithms demand a large number of function evaluations to arrive at a reasonable solution. For example, the EA used in [8] required roughly 500 function evaluations to reach an optimum, and the same algorithm reached 8000 evaluations in [9]. To cope with such substantial computational demands, the recent trend in bio-inspired aquatic optimization was to reduce the fidelity of the simulation method, with Refs. [11,12] departing from CFD and employing simple hydrodynamic models based on a potential flow solution and empirical estimates [16–18] to provide evaluation of hydrodynamic forces.

While reduced-order models based on potential theories and empirical correlations are valuable for providing basic scaling laws and dominant trends in biolocomotion [19], their applicability to a quantitative evaluation of hydrodynamic parameters, especially during the optimization procedure, is questionable for several reasons: 1) They typically underestimate the total work and overestimate the speed, thrust, useful work and efficiency [20,21]; 2) They can not reliably predict the drag of an undulating swimmer [2,20,22]; 3) Due to a lack of experimental data on sub-optimum swimming, it is unclear to which extent they are applicable to the modes of locomotion, other than the optimal; 4) They are largely based on steady-state swimming theories and data [16,20,23], and might not be extendable to other hydrodynamic regimes, such as, e.g., acceleration [24,25].

For the above-mentioned reasons, we are interested in developing an optimization framework based on high-fidelity fully-resolved simulations (FRS), in order to put forward a robust and reliable methodology, which could be used through a variety of hydrodynamic regimes, and not just for a steady swimming, and could provide a predictive capability for the purposes of comparison, refinement and further development of reduced-order models. To keep computational costs realistic, we thus pursue a different approach: instead of sacrificing the numerical method’s fidelity, we are looking for efficient optimization techniques to tackle the problem. One such technique comes from the field of surrogate-based optimization (SBO). This branch of optimization makes use of a surrogate model built with the help of true function evaluations, which is used by an optimizer for finding local or global optima [26–28]. One of the methodologies under the SBO umbrella, a surrogate management framework (SMF), relies on a flexible global search method and a fixed local pattern search method to exploit a Kriging response surface of the objective function [26,27]. Due to its flexible nature, the global search method specified in SMF aims to enable users to strike a balance between a global versus local exploration of the response surface according to the nature of the optimization problem. The convergence properties of SMF rely on its local search pattern, or the POLL step, where a proof of convergence can be found in [26]. The objective of the current study is achieved by designing an appropriate SBO optimization strategy, with proven global convergence properties, to maintain a low budget for a high fidelity FRS-based optimization. To this end, we apply a SBO algorithm based on a Kriging response surface [29], which is capable of ensuring strictly feasible (and hence realistic) design candidates during optimization of FRS while keeping the number of function evaluations low [27,30]. Exploitation of the Kriging surface relies on trust region global and local search methods, both of which guarantee convergence in the probabilistic sense [31]. We further develop this optimization framework to extend the surrogate updating strategy, previously available only for simple box-constrained optimization [31], to handle a constrained optimization procedure relevant in the studies of fish propulsion.

Maertens et al. [13] documented an optimization study of the undulatory propulsion using viscous flow CFD simulations and a BOBYQA algorithm, which performs bound-constrained optimization using an iteratively constructed quadratic approximation for the objective function [32]. BOBYQA is a local optimization method, and thus is prone to local minima traps, at least, without special modifications [32,33], as compared to the global search methods, such as the one based on a Kriging response surface employed in the current work. Apart from a difference in optimization procedure, Maertens et al. [13] also use a different computational approach, which is a second-order immersed boundary (IB) method, as compared to a spectral-element method with an arbitrary Lagrangian-Eulerian (ALE) formulation in this work. While the IB technique does not require remeshing in moving body applications, discontinuities at the immersed interface, especially for a moving-body, can produce spurious pressure and velocity modes [34]. These limitations are typically overcome by a fluid mesh with a highly refined interface boundary. As such, an IB method might require an excessively large number of computational grid points as opposed to the body fitted grid methods [35,36], especially to resolve the fluid-structure interaction at the moving surfaces, and thus lose the computational advantages related to its original simplicity of formulation. Finally, the study in [13] focuses on a steady-state swimming, while we look at an optimization of an accelerated propulsion. As pointed out in [24], while locomotion strategies for steady swimming are well characterized, far less known about acceleration [25,37]. Thus, the current work also contributes to a scarce body of literature regarding accelerated swimming.

In this paper, we present a novel computational methodology for optimization of the kinematic gaits of a self-propelled thunniform swimmer utilizing a spectral-element method Nek5000 for high-fidelity simulations on moving body-fitted grids [38,39], coupled with the SBO optimization approach. An overall schematic of the workflow during the optimization procedure is presented in Fig. 1. In addition to paying close attention to the efficiency of optimization algorithms, the

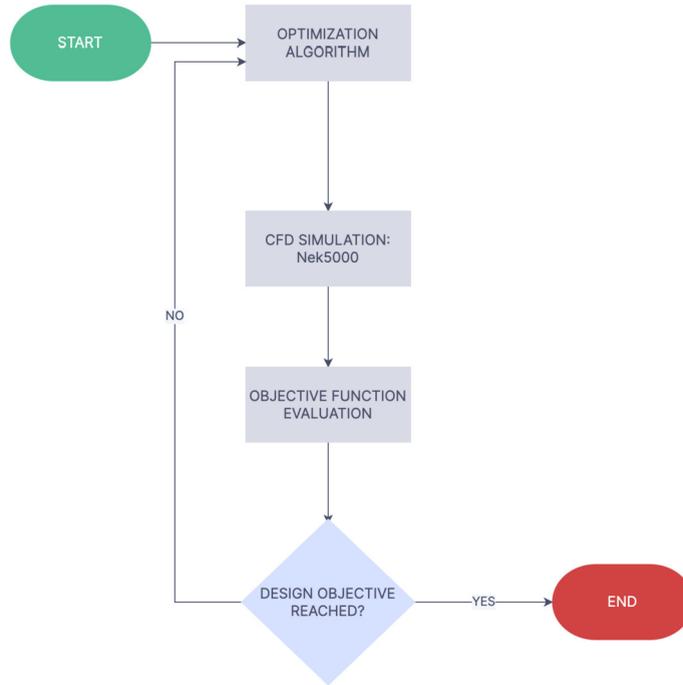


Fig. 1. Flowchart of the optimization process.

current work also presents new developments that significantly improve efficiency of fluid-structure interaction (FSI) algorithms on body-fitted grids in incompressible flow. In particular, we introduce a new volume conservation approach for a geometry reconstruction of the undulating fish body with a widely used kinematic deformation model [16,22], which allows us to avoid, otherwise necessary for stability purposes, implicit iterations between the velocity and pressure solvers [36,40], and speeds up the calculations by three to five times. Another important aspect of the current work is that it establishes a grid convergence of the optimization procedure with a polynomial refinement of the CFD solution, and investigates a sensitivity of the optimization results to a numerical resolution of the FSI simulations, an important issue, which, to the authors' knowledge, seems to have been overlooked in the previous optimization studies [8–10,13], at least in the context of bio-inspired propulsion. We also present a physical analysis of the optimum and sub-optimum modes of locomotion obtained with a high polynomial order of  $N = 9$ , and propose a new scaling law for the propulsive efficiency versus the kinematic gait parameters, applicable to both optimum and sub-optimum propulsion.

The paper is organized as follows. In section, 2 we describe the physical model of a self-propelled undulatory swimmer, including the equations governing the swimmer's shape, deformation kinematics and the volume conservation scheme, as well as the self-propulsion mechanism. In section 3, the Navier-Stokes equations governing the fluid flow are formulated and discretized using a spectral element method on moving body-fitted grids. In section 4, the simulation setup, including the computational grid and the boundary conditions are introduced. In Section 5, the optimization problem is posed, and a detailed description of the utilized and extended SBO algorithm is given. Section 6 includes a discussion of the results, while Section 7 presents the concluding remarks.

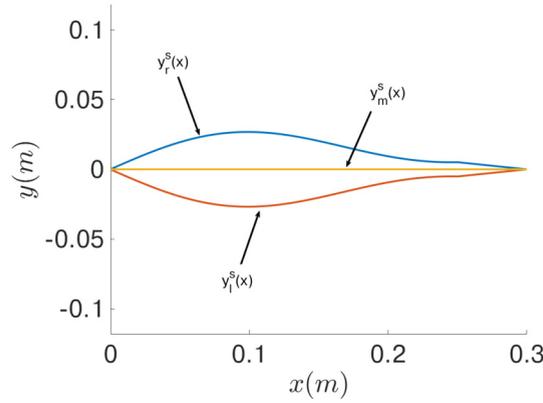
## 2. Model of self-propelled undulatory swimmer

### 2.1. Swimmer shape

The computational model of a thunniform swimmer employed in the current work considers its two-dimensional approximation in a streamwise-lateral plane. The geometrical features of the streamlined body of a thunniform swimmer in its static configuration in this two-dimensional approximation have been described using real fish morphology data [41,42],

$$y_r^s(x) = r_1 \sin(r_2 x) + r_3 \sin(r_4 x), \quad (1)$$

where  $r_1 = 0.055l_b$ ,  $r_2 = \frac{2\pi}{1.25l_b}$ ,  $r_3 = 0.08l_b$ ,  $r_4 = \frac{2}{l_b}$ , and  $l_b$  is the length of the swimmer's body up to but not including the tail. We take equation (1) to be the curve corresponding to the right boundary of the fish,  $y_r^s(x)$ , and the left boundary is given by the symmetry, such as  $y_l^s(x) = -y_r^s(x)$ . The total dimensional length,  $L = l_b + l_t$ , where  $l_t$  is the length of the tail, is chosen to be  $L = 0.3$  m to mimic the length of a realistic soft robotic thunniform swimmer [42]. We model a sharp tail



**Fig. 2.** Static thunniform swimmer.  $y_r^s(x)$  and  $y_l^s(x)$  are the right and left boundaries of the swimmer in its static configuration, respectively;  $y_m^s(x) = 0$  is the static midline.

by linearly extending the body of the swimmer across  $l_t$ , which is chosen to be  $l_t = 0.05$  m, yielding  $l_b = 0.25$  m. The static swimmer geometry can be seen in Fig. 2.

## 2.2. Swimmer kinematics

The majority of robotic swimmers are designed to mimic the body and/or caudal fin (BCF) movement of aquatic animals [41]. BCF movement characterizes a range of swimming motions that are used by fast swimmers like tuna, sharks and pikes, and hence provides superior propulsive techniques to be utilized by robots. The form of the BCF flapping motion for a slender body undulating in a lateral direction with respect to a direction of motion can be described by a traveling wave equation [16,22] for the body midline deformation of thunniform (tuna) swimmers:

$$y_m(x, t) = \left[ c_1 \frac{x}{L} + c_2 \left( \frac{x}{L} \right)^2 \right] \sin(kx - \omega t), \quad (2)$$

where  $y$  and  $x$  are the lateral and streamwise coordinates of the midline points, respectively,  $t$  is the time variable,  $L$  is the fish length,  $c_1$  and  $c_2$  are the linear and quadratic wave amplitudes,  $k$  is the wave number associated with the body motion, and  $w$  is the wave frequency. Subsequent biological studies revealed that the thunniform swimming mode of the BCF family can be characterized by the following body wave number relation [43],

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

where  $\lambda$ , the body wave length, was measured to be  $\sim 1.1$ . We fix  $\lambda$  to be 1.1 and we note that while the tail-beat frequency is observed to have a positive correlation with the swimming speed [44,45], we fix  $\omega = 2\pi$  rad/s in this study.

The midline of a swimmer, which is simply  $y_m^s(x) = 0$  in its static configuration, is discretized with the corresponding midline points with  $x$ -coordinates  $x_m$ . During the motion described by Eq. (2), the  $x$ -coordinates of the midline discretization points remain unchanged, while  $y$ -coordinates update according to Eq. (2). After the midline of the swimmer is deformed, the next step is to deform the right and left curves of the body. Two methods for the body deformation are considered: a non-conservative, and a conservative method, that are described below.

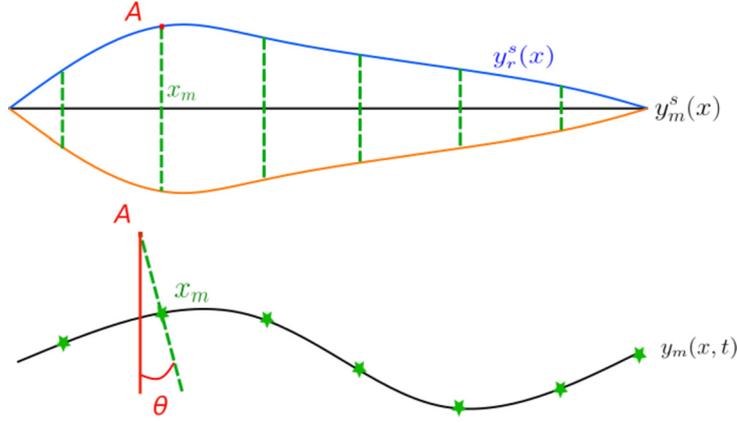
### 2.2.1. Non-conservative body deformation

The simplest way to deform the body of the fish is to adjust the boundary curves,  $y_r(x, t)$  and  $y_l(x, t)$ , to maintain orthogonality of the body cross-sections with respect to the midline while keeping the lateral distance between the corresponding points on the fish surface and the midline constant [5]. For example, consider a point,  $A$ , associated with a certain midline point with the  $x$ -coordinate of  $x_m$ , which lies on the boundary of a swimmer as depicted in its static configuration in the top view of Fig. 3. As the midline deforms, point  $A$  on the boundary moves to maintain orthogonality of the connecting segment with the midline as shown in the bottom view of Fig. 3. The geometrical location of the point  $A$ , lying, for example, on the right boundary of the swimmer,  $\{x_r^A(t), y_r^A(t)\}$ , can be described at any instance in time by the following equations:

$$x_r^A(t) = x_m - \sin(\theta(x_m, t))y_r^s(x_m), \quad (3a)$$

$$y_r^A(t) = y_m(x_m, t) + \cos(\theta(x_m, t))y_r^s(x_m), \quad (3b)$$

where the angle  $\theta(x_m, t)$  is such that



**Fig. 3.** Point A on the right boundary of a swimmer is shown in its static configuration (top view), and at a later point in time when the swimmer is moving (bottom view).

$$\theta(x_m, t) = \arctan \left( \frac{d y_m(x, t)}{d x} \Big|_{x=x_m} \right). \quad (4)$$

The corresponding surface velocity,  $\mathbf{v}_r^A(t) = \{v_{x_r}^A(t), v_{y_r}^A(t)\}$ , for the point A on the right boundary can be obtained by taking the time derivative of Eq. (3):

$$v_{x_r}^A(t) = \frac{d x_r^A}{d t} = - \frac{d}{d t} (\sin(\theta(x_m, t))) y_r^s(x_m), \quad (5a)$$

$$v_{y_r}^A(t) = \frac{d y_r^A}{d t} = \frac{d}{d t} (y_m^s(x_m, t)) + \frac{d}{d t} (\cos(\theta(x_m, t))) y_r^s(x_m). \quad (5b)$$

Similar expressions can be written for the left boundary. We remark that, potentially, with this deformation approach, it can happen that two neighboring segments may cross each other, which would lead to a mesh entanglement. However, due to a relatively low width of the fish and the motion constraints that restrict the undulation amplitude (See Section 5.3), such situations were not encountered in the current study.

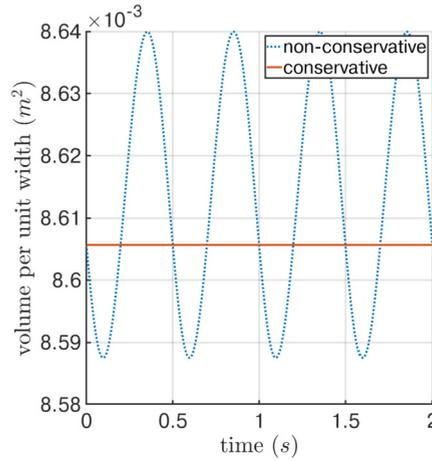
It is easily seen that holding the width of each orthogonal segment distance constant is not sufficient to maintain the volume conservation of the swimmer as the arc length of the midline fluctuates during swimming. To further illustrate this point, we plot the volume of the fish over one time period for the non-volume conserving deformation scheme (Fig. 4). It is not physiologically realizable for a neutrally-buoyant swimmer, robotic or organic, to change volume during swimming, unless other buoyancy aids are engaged, which would result in a change of the swimming altitude [46,47]. Additionally, a lack of body volume conservation leads to a sudden change in the volume of the surrounding fluid in the enclosed domain, which violates the incompressibility constraint and leads to numerical instabilities [40]. This has been previously alleviated by using additional iterations on the fluid pressure-velocity solve in the moving mesh framework [36,40], but this technique is expensive. For example, the results obtained with a non-conservative scheme in Fig. 4 required, on average, three to five fluid solver iterations per time step [48]. While the problem of volume conservation has been previously explored in relation to free-surface flows and fluid-structure interaction with passive solid objects [49,50], volume conservation of swimmers, to our best knowledge, has not received similar attention in the literature related to fish propulsion [5,8,51]. An exception is a study of Shirgaonkar et al. [52], where they imposed divergence-free deformations on an undulatory eel motion, albeit in their deformation scheme the body cross-sections were moved strictly laterally, rather than orthogonal to the midline, in which case the volume conservation is straightforward. An importance and a physiological relevance of orthogonal body deformations was discussed in Bergman et al. [5]. Therefore, in this work, we develop a volume-conservation method for a fish undulatory motion while keeping the fish body cross-sections orthogonal to the midline at all times.

### 2.2.2. Conservative body deformation

With the conservative approach, the body surface velocities obtained via Eq. (5), which by themselves do not guarantee the volume conservation, are augmented by a correction velocity  $\mathbf{v}_c(t) = v_c(t) \cdot \mathbf{n}$ , with the magnitude constant over the body surface, which acts in a surface-normal direction  $\mathbf{n}$  (with  $\mathbf{n}$  denoting the surface outer unit normal). With the correction velocity, Eqs. (5) can be re-written as:

$$v_{x_r}^A(t) = - \frac{d}{d t} (\sin(\theta_m(x_m, t))) y_r^s(x_m) + v_c(t) \cos(\theta(x_m, t)), \quad (6a)$$

$$v_{y_r}^A(t) = \frac{d}{d t} (y_m^s(x_m, t)) + \frac{d}{d t} (\cos(\theta_m(x_m, t))) y_r^s(x_m) + v_c(t) \sin(\theta(x_m, t)). \quad (6b)$$



**Fig. 4.** Volume of a swimmer undergoing an undulatory body motion over one cycle period with non-conservative and conservative deformation methods.

Correction velocity  $v_c(t)$  is found as the solution to the following optimization problem:

$$\begin{aligned} & \text{minimize} && \frac{d\mathcal{V}(v_c(t))}{dt} \\ & \text{subject to} && v_c(t) \in \mathbb{R}, \end{aligned} \quad (7)$$

where  $\mathcal{V}(v_c(t))$  is the volume of the swimmer that depends on the correction velocity  $v_c(t)$  (area in the current two-dimensional approximation). To solve the optimization problem, we use the Newton-Raphson secant method at every time step iteration, with the initial value of  $v_c(t) = 0$ . At each time step, the Newton-Raphson secant procedure converges in roughly 6–8 iterations to machine precision. The volume of a swimmer with non-conservative and conservative deformation methods as a function of time is shown in Fig. 4. The body shape at a time instance corresponding to the maximum difference in volume (approximately 0.35 s) is documented in Fig. 5a for the two methods. The two shapes are indistinguishable with a naked eye. Fig. 5b shows a difference in the lateral positions of the boundary curves between the conservative and non-conservative swimmers at  $t = 0.35$  s defined as

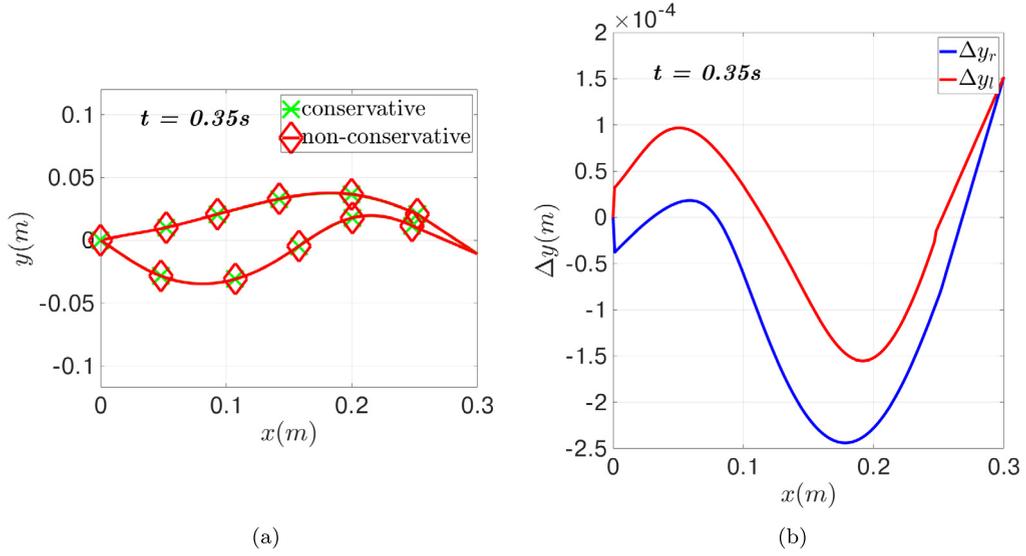
$$\Delta y_{(r,l)}(x) = y_{(r,l)}^C(x) - y_{(r,l)}^{NC}(x), \quad (8)$$

where  $y_{(r,l)}^C(x)$  and  $y_{(r,l)}^{NC}(x)$  are the volume-conserved and non-conserved right and left boundaries of the swimmer, respectively. The maximum difference is indeed very small, on the order of 0.1 mm, which is consistent with the maximum difference in volume of  $35 \text{ mm}^2$  (per unit width) and the fish length of 300 mm.

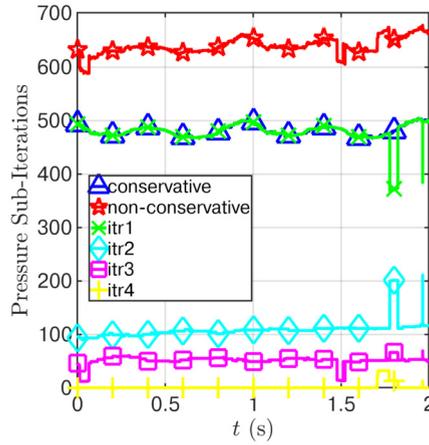
While the volume inconsistencies with the original, non-conservative scheme, are small, they are sufficient to trigger instabilities in the incompressible code that require additional fluid pressure-velocity iterations for stabilization, as described above. With the new developed correction procedure, the volume of the swimmer is conserved, and no additional fluid iterations are required on the solver to yield a stable solution. We note that the iterations on the system of Eqs. (6)–(7) in the conservative scheme are purely geometrical and are cheaper in comparison to the fluid pressure-velocity iterations of the non-conservative scheme [36,40]. Comparison of the total computational time between the two simulations, one with volume conservation and one without, shows that the conservative scheme leads to approximately 20% of the computational speed-up over the non-conservative scheme as applied to current simulations. As discussed above, the speed-up is enabled by the elimination of the fluid pressure-velocity iterations by the conservative scheme, while the non-conservative scheme requires three to four of such iterations (See Fig. 6). Each pressure-velocity iteration invokes a solution of the pressure Poisson equation, which is the most expensive part of the solver [53]. As can be seen from Fig. 6, the first pressure-velocity iteration (or the only pressure-velocity iteration in the conservative scheme) yields the highest amount of pressure sub-iterations in the Poisson solver (solved with the preconditioned GMRES algorithm [54]), while the subsequent pressure-velocity iterations in a non-conservative scheme are significantly more efficient. This explains why the savings from the conservative scheme do not scale with the amount of saved pressure-velocity iterations, but rather with the amount of saved total pressure sub-iterations in the Poisson solver. The conservative approach will be used in all the computations presented in this paper.

### 2.3. Self-propulsion

We model self-propulsion by considering the Newton's second law of motion for the undulating swimmer, while calculating the corresponding viscous and pressure forces directly from the fully-resolved simulations of the fluid-body interactions. Newton's second law of motion for the self-propelling fish can be written as



**Fig. 5.** Body shape of a swimmer undergoing an undulatory motion at  $t = 0.35$  s. a) Comparison of the body shape between conservative and non-conservative schemes. b) Difference in the lateral positions of the right and left boundary curves,  $\Delta y_r$ ,  $\Delta y_l$ , between conservative and non-conservative schemes.



**Fig. 6.** Amount of pressure sub-iterations in a Poisson solver taken by each pressure-velocity iteration in a conservative and non-conservative setting at each time step over two undulation cycles. Conservative scheme only has one pressure-velocity iteration, shown by blue triangles; for a non-conservative scheme, the total amount is shown by red stars, and the breakdown by iterations is indicated by the symbols highlighted in the figure legend.

$$m \frac{d\mathbf{U}(t)}{dt} = \mathbf{F}(t), \quad (9)$$

where  $m$  is the mass of the fish,  $\mathbf{U}(t)$  is its translational velocity, and  $\mathbf{F}(t)$  is the sum of the viscous and pressure forces acting on its body. The force vector  $\mathbf{F}(t)$  consists of a streamwise,  $F_x(t)$ , and lateral,  $F_y(t)$ , components, defined as

$$F_x(t) = \oint_{\Gamma_f} -(\sigma \mathbf{n}) \cdot \mathbf{i} d\Gamma, \quad (10)$$

$$F_y(t) = \oint_{\Gamma_f} -(\sigma \mathbf{n}) \cdot \mathbf{j} d\Gamma, \quad (11)$$

where  $\sigma$  is the total Cauchy stress tensor, which includes viscous and pressure contributions,  $\Gamma_f$  is the curvilinear boundary of the fish, and  $\mathbf{i}$ ,  $\mathbf{j}$  are the unit vectors in the streamwise and lateral directions, respectively. The streamwise force  $F_x(t)$  consists of the sum of the thrust and drag contributions, while the lateral force  $F_y(t)$  will sometimes be referred to as the lift force, to conform to the standard definitions in the airfoil literature [55].

In the current formulation, we only consider the contribution of the streamwise force,  $F_x(t)$ , i.e. thrust and drag, to the motion, and thereby the swimmer is confined to move in the streamwise direction and does not travel in the lateral direction. We also neglect propulsion effects due to moments. The same assumptions were made in [56,57]. The adopted motion mimics the case of a swimmer which leverages surrounding fluid to maintain its straight heading direction, and is beneficial for consideration of stability [56,57]. Therefore, we assume

$$m \frac{dU(t)}{dt} = F_x(t), \quad (12)$$

where  $U(t)$  is the fish forward velocity, and  $F_x(t)$  is the streamwise force defined in Eq. (10).

### 3. Numerical method

#### 3.1. Arbitrary Lagrangian-Eulerian formulation

In order to solve the moving fluid-structure interaction (FSI) problem, an arbitrary Lagrangian-Eulerian (ALE) formulation [58,59] of the incompressible Navier-Stokes equations is considered:

$$\rho \left( \frac{\delta \mathbf{u}}{\delta t} + (\mathbf{u} - \mathbf{w}) \cdot \nabla \mathbf{u} \right) + \nabla p = \mu \Delta \mathbf{u}, \quad (13a)$$

$$\nabla \cdot \mathbf{u} = 0, \quad (13b)$$

where  $\rho$ ,  $\mathbf{u} = \{u_x, u_y\}$ ,  $p$ ,  $\mu$  and  $\mathbf{w} = \{w_x, w_y\}$  are the fluid density, velocity, pressure, dynamic viscosity and the mesh velocity, respectively, while the derivative  $\delta/\delta t$  represents the ALE derivative, please, refer to [38,59] for more details. Mesh velocity has to satisfy the kinematic boundary condition

$$\mathbf{w} = \begin{cases} \mathbf{v}_{\Gamma_m}, & \text{on } \Gamma_m, \\ 0, & \text{on } \Gamma_s, \end{cases} \quad (14)$$

where  $\Gamma_m$  is the moving boundary of the fluid domain,  $\Gamma_s$  is the stationary boundary, and  $\mathbf{v}_{\Gamma_m}$  is the velocity of the moving boundary.

#### 3.2. Spatial discretization

The spectral element method (SEM) is used to solve Eqs. (13) by applying a weighted residual technique that casts the governing equations into a weak, or variational form [60]. The weak formulation of Eqs. (13) is given by: find  $\mathbf{u} \in \mathcal{H}^1(\Omega(t))^2$  and  $p \in \mathcal{L}^2(\Omega(t))$  such that  $\forall \mathbf{v} \in \mathcal{H}_0^1(\Omega(t))^2$  and  $\forall q \in \mathcal{L}^2(\Omega(t))$ ,

$$\frac{\delta}{\delta t} \int_{\Omega(t)} \mathbf{v} \cdot \rho \mathbf{u} \, d\Omega + \int_{\Omega(t)} \mathbf{v} \cdot \rho \nabla \cdot (\mathbf{u}\mathbf{u} - \mathbf{u}\mathbf{w}) \, d\Omega = \int_{\Omega(t)} (p \nabla \cdot \mathbf{v} - \mu \nabla \mathbf{v} : \nabla \mathbf{u}) \, d\Omega, \quad (15a)$$

$$- \int_{\Omega(t)} q \nabla \cdot \mathbf{u} \, d\Omega = 0, \quad (15b)$$

where  $\nabla \mathbf{v} : \nabla \mathbf{u}$  refers to a double-dot product between the two tensors,  $\mathcal{L}^2(\Omega(t))$  denotes the Hilbert space of square-integrable functions,  $\mathcal{H}^1(\Omega(t))^2$  denotes the Sobolev space of 2D vector functions which possess a square-integrable first derivative, while  $\mathcal{H}_0^1(\Omega(t))^2$  further constraints the functions to vanish on Dirichlet boundaries.

In a spectral element method, the computational domain  $\Omega(t)$  is subdivided into  $E$  non-overlapping rectilinear elements, or  $\Omega(t) = \cup_{e=1}^E \{\Omega_e(t)\}$ . Additionally, each element  $\Omega_e(t)$  is mapped to the reference domain,  $\hat{\Omega} = [-1, 1]^2$ , such that  $\mathbf{x}^e(\delta) \in \Omega_e(t) \rightarrow \delta \in \hat{\Omega}$ . The trial functions  $(\mathbf{u}, p)$  and the test functions  $(\mathbf{v}, q)$  are discretized on the reference domain using high-order polynomial expansions, which, in 2D, take the form

$$\mathbf{s}(x, y) = \sum_{i=1}^{N_x} \sum_{j=1}^{N_y} \hat{\mathbf{s}}_{ij} \phi_i(x) \phi_j(y). \quad (16)$$

Here,  $\phi_i$  and  $\phi_j$  are the basis functions in the  $x$  and  $y$  coordinates with degree  $N_x$  and  $N_y$ , respectively, and  $\hat{\mathbf{s}}_{ij}$  are the vector-valued basis coefficients. We specify the Lagrange interpolating polynomials as basis functions, which satisfy the cardinality property  $\phi_i(\xi_k) = \delta_{ik}$ ,  $\phi_j(\xi_m) = \delta_{jm}$ , with  $\delta_{ik}$ ,  $\delta_{jm}$  being the Kronecker delta functions, and  $\{\xi_k, \xi_m\}$ ,  $k = 1 \dots N_x$ ,  $m = 1 \dots N_y$ , the interior collocation points. We set the polynomial order  $N_x = N_y = N$ , with the discretization points as Gauss-Lobatto-Legendre (GLL) points for the velocity, and  $N_x = N_y = N - 2$ , with the discretization points as Gauss-Lobatto (GL) points for the pressure, to arrive at the  $\mathbb{P}_N - \mathbb{P}_{N-2}$  formulation known for its ability to suppress spurious pressure modes

[61]. The spectral element discretization ensures numerical stability and minimizes dissipative and dispersive errors [59]. For additional details, including extension of Eq. (16) to curvilinear coordinates and handling a global assembly across the elements, the reader is referred to [38,54,59].

### 3.3. Temporal discretization

#### 3.3.1. ALE equations

Equations (15) are discretized in time using explicit Adams-Bashforth scheme for nonlinear terms, and implicit backward-difference (BDF) scheme for viscous and pressure terms, yielding the following discretization

$$\frac{1}{\delta t} \left[ \left( \int_{\Omega(t^n)} \mathbf{v} \cdot \rho \mathbf{u} \, d\Omega \right)^n - \left( \int_{\Omega(t^{n-1})} \mathbf{v} \cdot \rho \mathbf{u} \, d\Omega \right)^{n-1} \right] + \sum_{j=1}^n a_j \left( \int_{\Omega(t^{n-j})} \mathbf{v} \cdot \rho \nabla \cdot (\mathbf{u}\mathbf{u} - \mathbf{u}\mathbf{w}) \, d\Omega \right)^{n-j} - \int_{\Omega(t^n)} (p \nabla \cdot \mathbf{v} \, d\Omega)^n = - \left( \int_{\Omega(t^n)} \mu \nabla \mathbf{v} : \nabla \mathbf{u} \, d\Omega \right)^n \tag{17}$$

where  $a_j$  is the  $j$ -th coefficient of the  $n^{th}$ -order Adams-Bashforth scheme ( $n = 3$  is used in the current setup), and  $\delta t$  denotes the time step. To ensure a divergence-free velocity field, the standard pressure-velocity decoupling approach resulting from the operator splitting is utilized [59,62]. As mentioned above, with a non-conservative solid deformation scheme, a nonlinear term has to be discretized implicitly via an iterative approach, while iterations are not required when the volume of the moving solid body is conserved. The SEM ALE algorithm was previously validated extensively on a variety of benchmark problems involving moving and deforming grids in its explicit and implicit formulation [36,38,39].

#### 3.3.2. Mesh velocity

To compute the mesh velocity, at the end of each time step  $t^n$ , the fish kinematics is first updated according to Eqs. (6). In solving for the correction velocity via Eq. (7), a volume change is discretized as  $dV/dt \approx (V^n - V^{n-1})/\delta t$ .

In the current situation of only one swimmer,  $\Gamma_m = \Gamma_f$ , and the velocity of the undulating fish surface, in accordance with Eq. (14), is set as the boundary condition for the mesh velocity at the fluid-body interface as

$$\mathbf{v}_{\Gamma_m}^n = \begin{cases} \mathbf{v}_r^n & \text{on } \Gamma_f : y^s \geq 0, \\ \mathbf{v}_l^n & \text{on } \Gamma_f : y^s < 0, \end{cases} \tag{18}$$

where  $\mathbf{v}_r^n, \mathbf{v}_l^n$  are velocities of the right and left boundaries of the swimmer, respectively, computed at time step  $t^n$ , and  $y^s$  is a static configuration of the fish as in Fig. 2. As noted above, the other boundary conditions on the mesh velocity are set to zero at the stationary surface  $\Gamma_s$ , which includes the inlet, the outlet, and the lateral boundaries of the fluid domain.

To find the interior mesh velocity  $\mathbf{w}^n(x, y)$  while maintaining a smooth mesh movement, a blending function,  $f_{blend}(x^s(x, y), y^s(x, y))$ , is designed, based on a static mesh configuration, to blend the mesh movement from the fish boundary  $\Gamma_f$  to the rest of the domain. We chose to design a custom mesh deformation algorithm rather than to rely on a built-in elasticity solver [36,53] to speed up the computations and to ensure a smooth mesh deformation. This is possible in our case, since the surface motion is known a-priori.

To design an appropriate mesh deformation function, we first define the undamped interior mesh velocity, by extruding the fish surface velocity into the interior of the domain as

$$\mathbf{w}_{\Gamma}^n(x, y) = \begin{cases} 0 & \text{for } (x, y) : x^s < 0, \\ \mathbf{v}_r^n(x) & \text{for } (x, y) : 0 \leq x^s \leq 0.3, y^s \geq 0, \\ \mathbf{v}_l^n(x) & \text{for } (x, y) : 0 \leq x^s \leq 0.3, y^s < 0, \\ \mathbf{v}_r^n(x_t) & \text{for } (x, y) : x^s > 0.3, \end{cases} \tag{19}$$

where  $x_t$  is the current axial position of the tail tip, and then smooth it with the help of a blending function as

$$\mathbf{w}^n(x, y) = f_{blend}(x^s(x, y), y^s(x, y)) \mathbf{w}_{\Gamma}^n(x, y). \tag{20}$$

The blending function,  $f_{blend}(x^s, y^s)$ , takes the following form:

$$f_{blend}(x^s, y^s) = \begin{cases} -y^s + y_2, & (x^s, y^s) \in \mathcal{X}^+ \\ y^s + y_2, & (x^s, y^s) \in \mathcal{X}^- \\ \frac{-(-y^s + y_2)(x^s - x_1)^2}{(x_2 - x_1)^2} + 1, & (x^s, y^s) \in \mathcal{Y}^+ \\ \frac{-(y^s + y_2)(x^s - x_1)^2}{(x_2 - x_1)^2} + 1, & (x^s, y^s) \in \mathcal{Y}^- \\ 1, & (x^s, y^s) \in \mathcal{Z} \\ 0, & \text{otherwise,} \end{cases} \tag{21}$$

where the regions  $\mathcal{X}^+$ ,  $\mathcal{X}^-$ ,  $\mathcal{Y}^+$ ,  $\mathcal{Y}^-$ ,  $\mathcal{Z}$  are defined according to the following rules:

$$\begin{aligned}\mathcal{X}^+ &= \{y^s \leq y_2, y^s \geq y_1, x^s \geq 0, x^s \leq x_1\}, \\ \mathcal{X}^- &= \{y^s \geq -y_2, y^s \geq -y_1, x^s \geq 0, x^s \leq x_1\}, \\ \mathcal{Y}^+ &= \{y^s \leq y_2, y^s \geq y_2, x^s > x_1, x^s \leq x_2\}, \\ \mathcal{Y}^- &= \{y^s \geq -y_2, y^s \leq -y_1, x^s > x_1, x^s \leq x_2\}, \\ \mathcal{Z} &= \{y^s \geq y^s(\Gamma_f), |y^s| < y_1, x^s > x_1, x^s < x_2\}.\end{aligned}\quad (22)$$

The blending function ensures that only the mesh in the vicinity of the undulating fish deforms, while the rest of the mesh remains stationary. The nodal fluid coordinates,  $\mathbf{x}$ , are updated by integration:

$$\mathbf{x}(t) = \mathbf{x}(t - \delta t) + \int_{t-\delta t}^t \mathbf{w}(t) dt, \quad (23)$$

where Eq. (23) is temporally discretized with the first-order implicit Euler scheme using the updated mesh velocity  $\mathbf{w}^n$ .

### 3.3.3. Self propulsion

The Newton's second law of motion that is utilized for a fish self propulsion in the current work, given by Eq. (12), is temporally discretized with the second-order implicit Adams-Moulton scheme:

$$U^n = U^{n-1} + \frac{\delta t}{2m} (F_x^n + F_x^{n-1}), \quad (24)$$

where  $F_x^n = F_x(t^n)$  is the total propelling force acting on the fish in the  $x$  direction at a time  $t^n$  given by Eq. (10). Instead of physically moving the fish through the domain, we model the swimmer's propulsion by adjusting the fluid velocity at the inlet,  $\mathbf{u}_{inlet}^n = \{U^n, 0\}$ . To achieve an implicit update of the Eq. (24) that depends on the force  $F_x^n$ , which subsequently depends on the fluid velocity field, a fixed-point iteration approach with Aitken relaxation [63,64] is performed as

$$\tilde{U}_i^n = U_{i_{max}}^{n-1} + \frac{\delta t}{2m} (F_{x, i-1}^n + F_{x, i_{max}}^{n-1}), \quad i \geq 1, \quad (25)$$

$$U_i^n = \omega_i \tilde{U}_i^n + (1 - \omega_i) U_{i-1}^n, \quad i \geq 1, \quad (26)$$

with  $U_0^n = U_{i_{max}}^{n-1}$ , where  $\omega_i$  is the Aitken relaxation parameter defined as

$$\omega_i = -\omega_{i-1} \frac{\tilde{U}_i^n - U_{i-1}^n}{\tilde{U}_i^n - U_{i-1}^n - \tilde{U}_{i-1}^n + U_{i-2}^n}, \quad i \geq 2, \quad (27)$$

and  $\omega_1 = 0.75$ . As can be seen from Eq. (27), the Aitken relaxation parameter is a dynamic quantity which relies on the values from the two previous FSI iterations to accelerate the convergence of the algorithm [64]. In the current application, the fixed-point method typically converges in 3–4 iterations to a set tolerance of  $\kappa_{fsi} = 1 \times 10^{-4}$ , where the tolerance is defined as  $tol = |(U_i^n - U_{i-1}^n)/U_{i-1}^n|$ , and the iterations terminate if  $tol \leq \kappa_{fsi}$ .

The need for an implicit update between the fluid-body interaction of the fish motion stems from the previous FSI studies involving incompressible flow and solid structures, which indicated that a simple sequentially partitioned coupling between the fluid and the solid (which would be equivalent here to an explicit discretization of Newton's second law) can be unstable irrespective of the time step due to the so-called "added-mass" effect [65,66]. It was shown in [64] that the use of an implicit predictor-corrector scheme based on fixed-point subiterations with Aitken relaxation alleviates the added-mass effect and stabilizes a numerical solution to the FSI problem. An adaption of this fixed-point iteration technique in a spectral-element method shows high accuracy and fast convergence for classical FSI benchmarks [36,40], and we refer the reader to these studies for a complete numerical description.

## 4. Simulation setup

### 4.1. Physical parameters

In the current simulations, the fluid and fish density are both taken to be  $1 \times 10^3$  kg/m<sup>3</sup>, which corresponds to a neutrally-buoyant swimmer. Dynamic viscosity of the fluid,  $\mu$ , is set to  $1 \times 10^{-3}$  kg/(m · s). The mass of the fish  $m$  is specified as 0.8606 kg, which corresponds to the parameters of a soft robotic thunniform swimmer prototype [42].

## 4.2. Computational parameters

The computational domain is specified as a rectangle with dimensions  $12L \times 4L$ . The fish leading edge is fixed a distance  $L$  away from the inlet and a distance  $2L$  away from the right and left lateral boundaries. Note that the leading edge (nose) of the fish does not move in the current formulation, while the trailing edge (tip of the tail) moves, see Eq. (2). A body fitted structured hexahedral mesh with  $N_{el} = 8887$  total elements is constructed to discretize the domain, with the element refinement along the streamwise direction and the lateral direction close to the fish boundary (See Fig. 7a for an illustration of the spectral element mesh in the swimmer's static configuration). For the mesh deformation in Eqs. (21), (22), the default parameters of  $\{x_1, x_2, y_1, y_2\} = \{0.3, 0.6, 0.1, 0.2\}$  are utilized. In the current paper, we perform numerical simulations for two values of the polynomial order  $N$ ,  $N = 5$  and  $N = 9$ , for the purposes of grid refinement, and to establish a sensitivity of optimization results to the degree of polynomial approximation in a high-order solver. The total gridpoint count based on the GLL velocity points is  $3.2 \times 10^5$  for  $N = 5$  and  $8.9 \times 10^5$  for  $N = 9$ . Fig. 7b shows a close-up of the numerical grid with the interior GLL points included for the polynomial order  $N = 5$ . The inset demarcated by the red rectangle in Fig. 7b further illustrates the details of the grid for the two polynomial orders,  $N = 5$  and  $N = 9$ , in Fig. 8. The static mesh shown in Figs. 7, 8 undergoes a dynamic remeshing at every time step to conform to the undulating fish geometry, according to the rules described in Section 3.3.2. The constant time step of  $\delta t = 2 \times 10^{-4}$  seconds is used throughout all the simulations, unless otherwise noted.

## 4.3. Boundary conditions

Fluid velocity at the fluid-body interface is set equal to the velocity of the moving boundary. A velocity inlet, a pressure outlet, and symmetry boundary conditions are prescribed at the inflow, outflow, and lateral boundaries, respectively. An inlet velocity is obtained at each time step from the self propulsion calculation as described in Section 3.3.3.

## 4.4. Initial conditions

A static configuration of the swimmer represented by Eq. (1) and depicted in Figs. 2, 7 does not actually correspond to any realizable fish position during swimming. To begin the undulations, we first need to move the swimmer into its starting, but deformed, position. To allow for initial transients to decay after the beginning of the undulations, we introduce the time delay  $\tau$  and evaluate the initial position of the swimmer at  $t = -\tau$ . Thus, the initial deformation of the swimmer midline is given by Eq. (2) evaluated at  $t = 0$ , while the position of fish surface at  $t = -\tau$  can be reconstructed via Eqs. (3), (4) evaluated at  $t = -\tau$ . Once the fish is moved into its starting position, the static mesh of Fig. 7 is correspondingly deformed via applying the blending function  $f_{int}$  given by Eq. (21) to the mesh deformation  $\delta \mathbf{x}$  instead of the mesh velocity as in Eq. (20). The initial fluid velocity and the mesh velocity at  $t = -\tau$  are all set to zero. Once the domain geometry and the numerical mesh are deformed, the simulations commence at  $t = -\tau$  by applying a midline deformation curve of Eq. (2) to the time  $t = -\tau + \delta t$  and following all the algorithmic steps as described above, while using the deformed start-up geometry as the initial configuration at  $t = -\tau$ . It was shown in [40] that initial transients cause pressure field to fluctuate which stabilizes after two time steps. Thus,  $\tau = 2 \delta t$  is chosen in the current simulations.

The developed computational methodology for an undulating swimmer self-propulsion is of the first-order accuracy in time and of spectral accuracy in space via polynomial refinement. Verification of the computational accuracy of the method, as well as sensitivity to the mesh deformation algorithm, are presented in Appendix A.

## 5. Optimization

### 5.1. Problem formulation

The optimization problem is stated as

$$\begin{aligned} & \text{maximize } f(\mathbf{z}) \\ & \text{subject to } \mathbf{z} \in \mathbb{R}^n, \end{aligned} \quad (28)$$

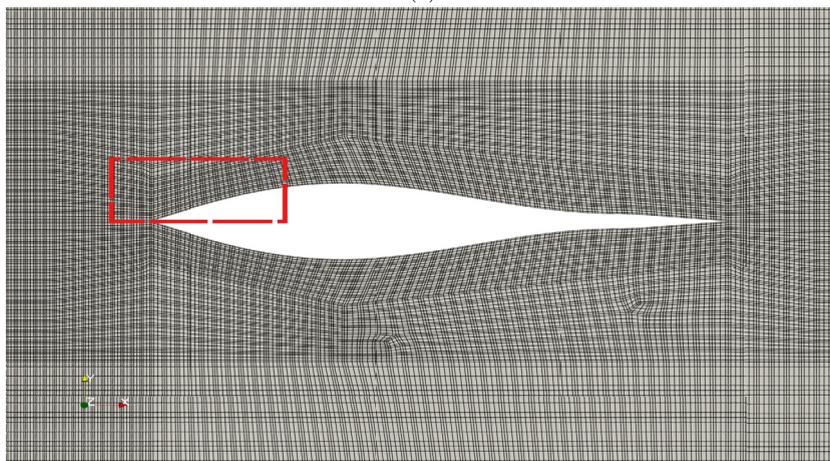
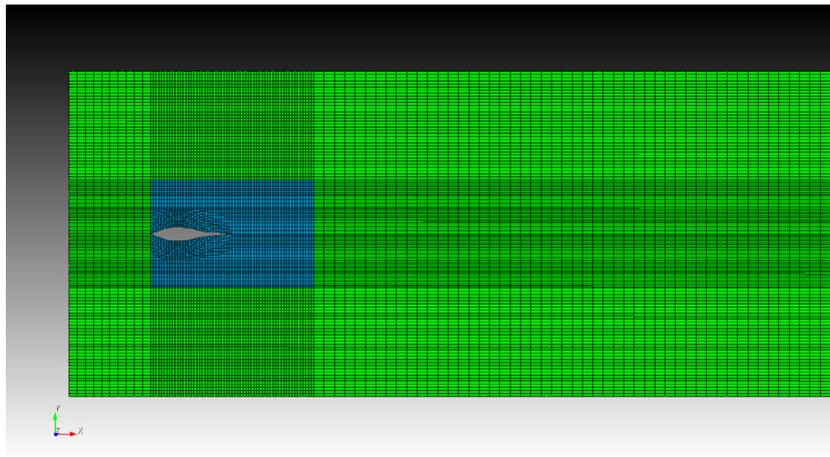
where  $f: \mathbb{R}^n \rightarrow \mathbb{R}$  is the objective function, and  $\mathbf{z} \in S \cap C$  is a vector of design parameters. The set  $S \subseteq \mathbb{R}^n$  contains the  $n$ -dimensional search space:

$$l(i) \leq z_i \leq u(i), \quad 1 \leq i \leq n, \quad (29)$$

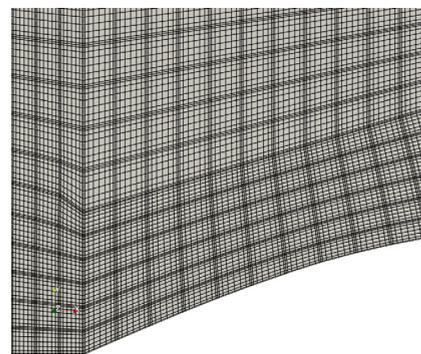
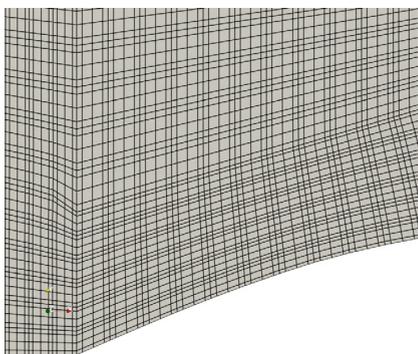
where  $l(i)$  and  $u(i)$  represent the lower and the upper bounds, respectively, on a design parameter in the  $i$ th dimension. The set  $C \subseteq \mathbb{R}^n$  contains a set of  $m \geq 0$  constraints:

$$g_r(\mathbf{z}) \leq 0, \quad r = 1, \dots, m,$$

where  $g_r(\mathbf{z})$  is referred to as an inequality constraint on the design parameter vector,  $\mathbf{z}$ .



**Fig. 7.** a) Spectral element mesh with refinement highlighted by black lines; only element boundaries are shown; The region defined by the union of  $\mathcal{X}^+$ ,  $\mathcal{X}^-$ ,  $\mathcal{Y}^+$ ,  $\mathcal{Y}^-$  and  $\mathcal{Z}$  is highlighted in blue (Eq. (22) with  $\{x_1, x_2, y_1, y_2\} = \{0.3, 0.6, 0.1, 0.2\}$ ); b) Numerical grid with GLL discretization points with  $N = 5$ . Red rectangle indicates an inset for zoom-in in Fig. 8.



**Fig. 8.** An inset of the numerical grid corresponding to a red rectangle in Fig. 7 with  $N$ -th order polynomial discretization (GLL points shown). a)  $N = 5$  and b)  $N = 9$ .

### 5.2. Design parameters

The design parameters are taken to be the amplitude coefficients  $\{c_1, c_2\}$  of the midline undulation function  $y_m(x, t)$ , see Eq. (2).

### 5.3. Constraints

In order to allow for physically realizable swimming configurations, we limit the midline movement, Eq. (2), by imposing the following constraint set,  $\mathcal{C}$ , on the design parameters  $\{c_1, c_2\}$ ,

$$\mathcal{C} = \begin{cases} |c_1 + c_2| \leq 0.1L, \\ \left| \frac{c_1^2}{4c_2} \right| \leq 0.1L, \\ c_2 \leq 0. \end{cases} \quad (30)$$

The constraint set is derived from a physical intuition used in soft swimmer design [41,42,56]. The first and second constraints ensure that the maximum tail and body motion amplitudes do not exceed  $0.1 L$ . The design domain governed by the first two constraints is symmetric about the origin  $\{c_1, c_2\} = \{0, 0\}$ , with the pairs  $\{c_1, c_2\}$  flipping sign resulting in identical swimming motions that are left-to-right reflections of each other. To avoid redundancy and to save 50% of the computational time, we introduce the third constraint restricting the design domain to one half of the coordinate space.

### 5.4. Objective function: propulsive efficiency

We define the objective function,  $f(\mathbf{z})$ , as a propulsive swimming efficiency of an undulatory swimmer. A propulsive efficiency can be defined as the ratio of a “useful” energy gained over the total work done by the swimmer over a certain time period [18,24,67], which can be stated as

$$\eta(c_1, c_2, t_0) = \frac{W_{useful}}{W_{total}} = \frac{\int_0^{t_0} \left( \oint_{\Gamma_f} -(\sigma \mathbf{n}) \cdot \mathbf{i} d\Gamma \right) U(t) dt}{\int_0^{t_0} \oint_{\Gamma_f} -(\sigma \mathbf{n}) \cdot \mathbf{v}(x, y, t) d\Gamma dt} \quad (31)$$

where  $t_0$  is an *a priori* specified variable indicating a time period, and  $\mathbf{v}(x, y, t)$  is the fish surface velocity due to undulation. While the integration starts at zero, it is ensured that initial transients are decayed by this time by specifying initial conditions at an earlier time  $t = -\tau$  as described in Section 4.4. A choice of the integration period  $t_0$  requires a careful consideration. By fixing  $t_0$  we seek a kinematic gait corresponding to a swimming mode capable of efficiently transferring undulation work to propulsive energy over a brief time span, which is referred to as the start-up propulsive efficiency. As the baseline case, we set  $t_0$  equal to  $2T$ , where  $T = 2\pi/\omega = 1$  s is the tail-beat period. A sensitivity of results to a choice of  $t_0$  is further discussed in Section 6.2. Since the objective function, Eq. (31), does not explicitly depend on the design parameters, and instead its evaluation requires a solution of the Navier-Stokes equation, this model is referred to as a “black-box” optimization problem.

### 5.5. Surrogate based optimization (SBO)

#### 5.5.1. SBO framework

In this work, a surrogate based optimization procedure for a bio-inspired aquatic propulsion is developed. SBO framework is chosen for its high efficiency without sacrificing the accuracy [26,27]. A surrogate based optimization algorithm is a gradient-free global search method which seeks to identify an optimum by building a surrogate approximation of the objective function, thereby requiring no derivative information or approximation. An initial surrogate of the objective function is created by sampling the true objective function values according to a space filling strategy. An infill criteria is then used to assess fitness of inexpensive surrogate function evaluations to determine one or more points to be evaluated using the true objective function. Those points are used to refine the surrogate, and the process is repeated until an optimum is identified.

We utilize an SBO framework which uses the DACE [29] MATLAB toolbox to sample, build and refine a surrogate model of the objective function. We define an infill strategy, which follows the Metric Stochastic Response Surface (MSRS) framework proposed by Regis and Shoemaker [31]. MSRS method ensures global convergence to an optimum while producing a competitive average best function value with a fixed iteration count [31]. In its original form, MSRS equation handles only box constraints. Since the thuniform swimmer problem involves physiological constraints, given by Eq. (30), which restrict the box-type  $n$ -dimensional search space, Eq. (29), we develop a modification to the MSRS method to handle generic linear and nonlinear constraints that can be imposed on a solution.

#### 5.5.2. Base algorithm

The steps of the SBO algorithm used in this study are described as follows. Let the feasible domain,  $D$ , be defined as  $D = S \cap C$ . Additionally, let  $M_k$  and  $s_k(x)$  denote the mesh space and the surrogate model, respectively, at iteration  $k$ .

Define maximum number of iterations,  $k_{max}$ , and a tolerance bound,  $\varkappa_{opt}$ , such that the iterations terminate if the computed tolerance.

$$tol = \left| \frac{f_{opt} - f_{old-opt}}{f_{old-opt}} \right| \quad (32)$$

reaches the value

$$tol \leq \varkappa_{opt}, \quad (33)$$

where  $f_{opt}$  and  $f_{old-opt}$  are the values of the objective function from the best and the next best iterations.

The following procedure describes the optimization algorithm:

#### Initialization

**Step 1:** Sample a finite set of evaluation points  $T \subset D$  using Latin Hypercube Sampling [68], where  $\text{card}(T) = 20$ , and  $\text{card}(T)$  refers to the size, or cardinality, of a set  $T$ . Evaluate  $f(T)$ . Identify the current best point  $\mathbf{z}^0$ . Set  $M_0 = T$ .

**Step 2:** Fit a Kriging model surrogate,  $s_0(\mathbf{z})$ , with a Gaussian correlation function and 0-th order regression polynomial. In other words, the surrogate model is assumed to have a constant mean and a stochastic error term that is modeled by a Gaussian process. This creates a flexible and reliable prediction method known as ordinary Kriging [69].

#### Iterations

**Step 3:** While ( $k < k_{max}$ )

(a) Create a set of strictly feasible candidate points,  $Z_k$ , according to the new candidate sampling algorithm developed in this work and presented in Section 5.5.3, and evaluate  $s_k(Z_k)$ .

(b) Use MSRS to assess fitness of the candidate points. The MSRS method assigns a weighted score to each point in the set  $Z_k$  based on two criteria: 1) the distance of points in  $Z_k$  to  $M_k$ , and 2) the surrogate response values,  $s_k(Z_k)$ . Since the surrogate approximates the objective function that is being maximized, high response values are sought-after. Thus, the weighted score favors the candidates associated with a high response value that are far away from already sampled points, thereby striking a balance between local and global exploration. The point with the best weighted score is identified as the next evaluation point,  $\mathbf{z}^k$ .

(c) Evaluate  $f(\mathbf{z}^k)$ . If  $tol \leq \varkappa_{opt}$  criterion is met: **break**.

(d) Set  $M_{k+1} = M_k + \mathbf{z}^k$ . Re-fit  $s_{k+1}$  with  $M_{k+1}$ . Set  $k = k + 1$ .

**Step 4:** Return  $\mathbf{z}^k$ .

#### 5.5.3. New algorithm: constrained candidate sampling

This section describes a new algorithm developed in this work to handle generic constraints with the MSRS method. The candidate points are split into two selection categories which follow a trust region approach to guarantee an optimum in a probabilistic sense [31]:

1. Uniformly sampled global points: The first set,  $U_k$ , is generated by sampling points from a continuous uniform distribution across the box-constrained domain such that  $U_k \subset S$ . Following [70], we set  $\text{card}(U) = 2000n$ .
2. Normally sampled local points: The second set  $N_k$  is generated by adding perturbations to  $\mathbf{z}^{k-1}$  drawn from a normal distribution with zero mean and unit variance. There are three perturbation rates chosen: one-tenth, one-hundredth and one-thousandth of the smallest variable range. The smallest variable range is defined as  $\min_i (u(i) - l(i))$ , where  $l(i)$  and  $u(i)$  are the lower and upper bounds on a design variable  $z_i$ . Following [70], we set  $\text{card}(N) = 2000n$ .

We define the possibly unfeasible candidate set as  $Z'_k = U_k + N_k$ . It follows that  $\text{card}(Z'_k) = 4000n$ . We note that  $Z'_k \subset S$ , however it might be that  $Z'_k \not\subset S \cap C$ . We then enforce linear and nonlinear constraints through the following algorithm:

**Step 1:** Evaluate  $g_r(Z'_k)$  for  $r = 1, \dots, m$ , where  $g_r(\mathbf{z})$  is inequality constraint, and  $m$  is the total number of inequality constraints. We only consider inequality constraints as they are sufficient for the current problem addressed. However, equality constraints, if present, can be trivially accounted for in the same manner.

**Step 2:** Define 'penalty' vector,  $J$ , for each candidate point:

$$J_i = \sum_{r=1}^m \max(0, g_r(\mathbf{z}^i)), \quad i = 1, \dots, \text{card}(Z'_k), \quad (34)$$

where  $J_i$  are the entries of the vector  $J$ .

**Step 3:** The candidate points set,  $Z_k$ , is now simply defined as:  $Z_k = Z'_k : J = 0$ . In other words, the  $Z_k$  candidates are the candidates in  $Z'_k$  with a zero penalty.

**Table 1**

Computational time taken by  $N = 5$  and  $N = 9$  cases on 84 cores of the ASU high-performance computing cluster Agave with  $t_0 = 2T$ .

Case	Total time per simulation cycle	Time per cycle spent on constraint evaluations	Total time of optimization
$N = 5$	$3.57 \times 10^3$ seconds	0.021 seconds	$1.86 \times 10^5$ seconds
$N = 9$	$1.10 \times 10^4$ seconds	0.021 seconds	$6.27 \times 10^5$ seconds

**Table 2**

Results for the swimmer propulsive efficiency optimization with two SEM polynomial orders for  $t_0 = 2T$ .

Case	Evaluations	Efficiency ( $\eta_{opt}$ )	Optimum $\{c_1/L, c_2/L\}$	tol
$N = 5$	52	12.90%	{0.2106, -0.1108}	$5.9 \times 10^{-5}$
$N = 9$	57	12.04%	{0.2104, -0.1106}	$4.5 \times 10^{-5}$

This method does not give an a-priori estimate of a cardinality of a candidate set  $Z_k$ , but it guarantees that the final candidate set  $Z_k$  is always feasible for all possible candidate solutions. If it happens that  $Z_k \subset M_k$ , then the candidates are resampled.

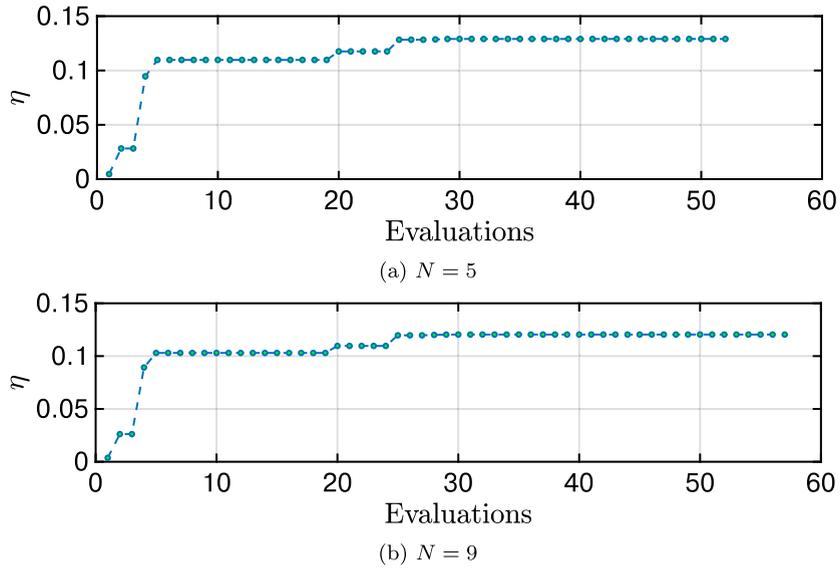
The developed SBO algorithm was validated using several benchmark problems, including analytical functions and a black-box optimization [30,71]. Compared to other optimization algorithms, it consistently showed a competitive performance in terms of the error bounds on the optimum solution and the number of function evaluations. As compared to evolutionary algorithms, SBO was typically able to find an optimum solution within the specified error bounds using a significantly less number of function evaluations [30,71].

## 6. Results

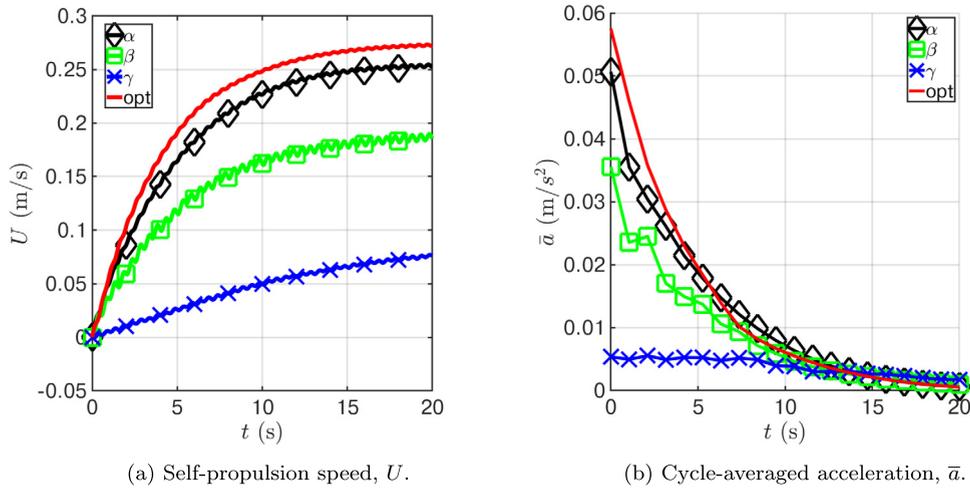
The simulations were performed on 6 nodes (84 cores) of a parallel computing cluster Agave at ASU with each compute node containing multi-core Intel Xeon E5-2680 v4 CPUs running at 2.40GHz. Table 1 lists the computational time taken by  $N = 5$  and  $N = 9$  cases (using  $t_0 = 2T$ ), both for a single CFD simulation, and for the total time of optimization considering the number of evaluations listed in Table 2. We also list the time it took to evaluate all the constraints per single simulation cycle. It is seen that the speed-up between  $N = 9$  and  $N = 5$  cases is approximately three-fold, as expected, owing to a strong parallel efficiency of the code that keeps a constant time per gridpoint per timestep between the simulations [72], and a  $(9/5)^2 \approx 3.24$  decrease in the number of gridpoints from  $N = 9$  to  $N = 5$  case. We also note that the time spent on evaluation of  $4000 \times 2 = 8000$  constraints per simulation (2 is the dimension of the optimization problem) takes an insignificant portion of the total simulation time.

### 6.1. Optimum parameters at $t_0 = 2T$

In this section, results of the optimization procedure as applied to an accelerated self-propulsion of a thunniform swimmer that maximizes its propulsive efficiency over two swimming periods are presented for two polynomial orders,  $N = 5$  and  $N = 9$ . We specify the following termination parameters for the optimization algorithm:  $k_{max} = 700$ ,  $\varkappa_{opt} = 1 \times 10^{-4}$ . Fig. 9 documents the evolution of the best identified value of the propulsive efficiency  $\eta$  as the function evaluations progress, and Table 2 lists the final results of the optimization procedure, including the total number of evaluations, the final value of the propulsive efficiency, the optimum motion parameters  $\{c_1/L, c_2/L\}$ , and the actual error tolerance norm  $tol$  from Eq. (32) at the end of optimization. We note that Fig. 9 shows efficiency from all the iterations which include both the initial sampling (the first 20 iterations, ordered with the increasing efficiency) and the optimization iterations (21–52, when  $N = 5$ , and 21–57, when  $N = 9$ ). Consequently, this affects the slope of the efficiency function, which increases drastically in the beginning, as it starts from some clearly inefficient motions, while it increases more slowly after it reaches a certain acceptable value of efficiency and works its way up to satisfy a relatively tight tolerance of  $1 \times 10^{-4}$ . We see that in both cases the tolerance bound  $\varkappa_{opt} = 1 \times 10^{-4}$  is reached with less than 60 iterations. The  $N = 5$  case results in a slightly higher tolerance norm but a lower function evaluation count, and yields a slightly higher value of an optimum propulsive efficiency  $\eta_{opt} \approx 12.90\%$  as compared to  $N = 9$  answer of  $\eta_{opt} \approx 12.04\%$ . While the efficiency values are slightly different, both polynomial orders produce essentially identical solutions in terms of the optimum mode of locomotion  $\{c_1/L, c_2/L\}$ , differing only in the fourth significant digit. The computed optimum efficiency agrees well with the biological data for the propulsive efficiency measured in the live fishes for the given tailbeat amplitude. For example, the authors of [24] survey a large class of accelerating fish, 51 species from 20 taxonomic orders with different body shapes and sizes, morphologies and ecologies. The accelerating swimmers swam in flows with Reynolds number ranging from  $1.47 \times 10^5$  to  $2.46 \times 10^5$ . By defining the efficiency as the ratio of the propulsive work to the total input work over 10 swimming periods, the authors report propulsive efficiencies for accelerating swimmers which reach up to 25%. Using a similar definition of efficiency (Eq. (31)) integrated over 10 periods (as seen in Table 3), we see that the propulsive efficiency ( $\eta \approx 20\%$ ) falls within the range reported in [24].



**Fig. 9.** Propulsive efficiency  $\eta$  versus the number of function evaluations for two SEM polynomial orders for  $t_0 = 2T$ : (a)  $N = 5$  and (b)  $N = 9$ . The  $\eta$  value reported corresponds to the best solution found over the considered number of iterations.



**Fig. 10.** The self propulsion speed, (a), and the cycle-averaged acceleration, (b), of the optimum and three selected sub-optimum modes across  $20T$  with  $N = 5$ .

For comparison, we also perform a similar optimization study using a popular evolutionary optimization approach [8,11,12], utilizing COLINY-EA algorithm from the state-of-the-art DAKOTA optimization framework [73], which, after  $k_{max} = 700$  iterations reached a tolerance  $tol = 1.4 \times 10^{-3}$  and produced a mode with only 11.6% efficiency [30]. This, again, confirms that the SBO optimization procedure might be a more efficient alternative to use with expensive function evaluations, at least within the scope of the considered design problem.

### 6.2. Sensitivity to integration period

The results presented in Section 6.1 assumed a fixed integration period of  $t_0 = 2T$ . Since our objective is to elucidate on effective propulsive modes that correspond to the most efficient acceleration from rest, a predilection in the current study is to keep  $t_0$  small. The choice of “how small” is however somewhat judicious, as there are no set guidelines on when exactly a “propulsive mode” of locomotion does or should transition to a less vigorous “quasi-steady mode”, as this process is gradual. To understand this further, a sensitivity of results to the choice of  $t_0$  is important to evaluate.

To evaluate the sensitivity of results to the integration period, we consider the efficiency of the optimum mode and three selected sub-optimum modes, named as the modes  $\alpha$ ,  $\beta$  and  $\gamma$ , across different integration periods, which range from  $1T$

**Table 3**

Calculated propulsive efficiency of the optimum and selected sub-optimum modes across different integration periods with  $N = 5$ .

Mode-iter	$\{c_1/L, c_2/L\}$	1T	2T	4T	10T	15T	20T
opt-52	{0.21,-0.11}	7.88%	12.90%	18.84%	20.19%	17.51%	14.85%
$\alpha$ -29	{0.09,-0.17}	5.81%	8.49%	12.86%	15.00%	13.01%	10.88%
$\beta$ -22	{0.35,-0.45}	2.33%	3.04%	4.36%	4.56%	3.72%	3.02%
$\gamma$ -9	{0.20,-0.24}	0.47%	0.75%	1.51%	3.29%	3.82%	3.87%

**Table 4**

Results for the propulsive efficiency optimization using two different integration periods,  $t_0 = 2T$  and  $t_0 = 4T$ , with  $N = 5$ .

Case	Evaluations	Efficiency ( $\eta_{opt}$ )	Optimum $\{c_1/L, c_2/L\}$	tol
2T	52	12.90%	{0.2106, -0.1108}	$5.9 \times 10^{-5}$
4T	63	18.84%	{0.2089, -0.1091}	$6.8 \times 10^{-5}$

to 20T. Table 3 lists the kinematic parameters of the modes,  $\{c_1/L, c_2/L\}$ , the iteration number when these modes were encountered, and their efficiency over selected integration periods with  $N = 5$ .

Fig. 10 shows the evolution of the swimming speed,  $U$ , as well as the cycle-averaged acceleration,  $\bar{a}$ , for the optimum,  $\alpha$ ,  $\beta$  and  $\gamma$  modes over 20 tailbeat periods. It is seen that acceleration is high at the beginning (apart from a very inefficient  $\gamma$  mode), while it gradually decreases, asymptotically approaching a value of zero, which would correspond to a steady swimming, as swimming time increases to 20T. It can be seen that if one is interested in a regime of active propulsion (high acceleration), it is natural to restrict ourselves to small integration periods. For example, at  $t_0 = 2T$ , acceleration decays by 30% in the optimum mode.

While physics and kinematics of  $\alpha$ ,  $\beta$  and  $\gamma$  modes will be discussed further in the manuscript, Table 3 shows that although the propulsive efficiency changes with  $t_0$ , the relative efficiency rankings between the modes remain largely consistent irrespective of the integration period. Specifically, the efficiency rankings between the four modes remain exactly the same up to the integration period of 10T, and the optimum mode remains the highest ranked among the four all the way through 20T. After the integration period of 10T, the propulsive efficiency of the optimum mode as well as the modes  $\alpha$  and  $\beta$  starts to decrease, while the efficiency of the least optimum mode of the four, mode  $\gamma$ , starts to pick up and eventually overcomes that of  $\beta$ , which may indicate the end of the effective “start-up” period. Given this, as long as the value of the integration period  $t_0$  is below 10T, this should in principle represent the regime of efficient acceleration.

To further assess this presumption, we perform another optimization study with  $N = 5$  and  $t_0 = 4T$ . Results of the  $t_0 = 4T$  case, as shown in Table 4, yield an optimum efficiency value that is 18.84%. This is the same efficiency as the 2T-optimized mode was showing at the integration period of 4T (Table 3). Moreover, the kinematic gaits of both 2T and 4T-optimized modes are essentially the same. These results indicate that the trends captured by optimization with two swimming periods are representative of a start-up acceleration process while maintaining low computational overhead.

### 6.3. Analysis of the optimum versus sub-optimum modes of locomotion

In this section, we perform an analysis of the optimum as compared to sub-optimum modes of locomotion, using  $N = 9$ ,  $t_0 = 2T$  case as a reference.

#### 6.3.1. Hydrodynamic variables

Fig. 11a shows a contour plot of the objective function, the propulsive efficiency  $\eta$ , using Kriging interpolation between evaluated data points. Fig. 11b documents the final attainable swimming speed at the end of the considered period 2T,  $U_{max} = U(t = 2T)$ . The quantities, whose ratio defines the propulsive efficiency in Eq. (31), namely  $W_{useful}$  and  $W_{total}$ , are plotted in Figs. 11c and 11d. To draw a complete picture, Figs. 11e and 11f present the time-averaged streamwise and lateral forces,  $\bar{F}_x$ ,  $\bar{F}_y$ , acting on the swimmer, defined as

$$\bar{F}_x = \frac{1}{t_0} \int_0^{t_0} F_x(t) dt, \quad (35)$$

$$\bar{F}_y = \frac{1}{t_0} \int_0^{t_0} F_y(t) dt, \quad (36)$$

with the forces  $F_x(t)$ ,  $F_y(t)$  given by Eqs. (10), (11), respectively, and  $t_0 = 2T$ .

At a glance, the contours of all the five hydrodynamic variables, apart from the lift force, seem rather similar. Another interesting observation is that efficiency is very low in the interior of the design domain, with the line  $c_1 + c_2 = 0$  corre-

**Table 5**

Hydrodynamic quantities of selected kinematic modes with  $N = 9$ ,  $t_0 = 2T$ : a) Absolute quantities, and b) Relative quantities of the sub-optimum modes normalized by the respective quantity of the optimum mode.

	$\eta$	$W_{\text{useful}}$ (mJ)	$W_{\text{total}}$ (mJ)	$U_{\text{max}}$ (m/s)	$\bar{F}_x$ (N)	$\bar{F}_y$ (N)	$A$ (m)	$Re_{\text{max}}$
opt	12.04%	4.411	36.64	0.101	0.044	0.080	0.030	$3.03 \times 10^4$
$\alpha$	7.99%	3.132	39.20	0.085	0.037	-0.054	0.029	$2.55 \times 10^4$
$\beta$	2.74%	1.506	54.99	0.063	0.026	-0.024	0.028	$1.89 \times 10^4$
$\gamma$	0.64%	0.040	6.25	0.010	0.004	-0.004	0.010	$3.00 \times 10^3$

(a)

	$\eta_r$	$W_{\text{useful}_r}$	$W_{\text{total}_r}$	$U_{\text{max}_r}$	$\bar{F}_{x_r}$	$\bar{F}_{y_r}$	$A_r$
$\alpha$	0.66	0.71	1.07	0.84	0.84	-0.67	0.97
$\beta$	0.23	0.34	1.50	0.62	0.58	-0.30	0.93
$\gamma$	0.05	0.01	0.17	0.10	0.10	-0.05	0.33

(b)

sponding to a near-zero efficiency. Indeed, as can be seen from Eq. (2), the line  $c_1 + c_2 = 0$  represents a zero tail amplitude motion, and thus corresponds to a zero theoretical propulsive efficiency [19,23,74].

To better understand the relationships and the correlation properties between different hydrodynamic variables associated with propulsive swimming, Fig. 12 presents the correlation matrix of the six considered hydrodynamic variables, and documents the respective Pearson correlation coefficient  $R$ . It can be seen that the correlation between efficiency  $\eta$  and the useful (propulsive) work  $W_{\text{useful}}$  ( $R = 0.98$ ), and, consequently, between efficiency  $\eta$  and the propulsive force  $\bar{F}_x$  ( $R = 0.97$ ) are the highest. It is interesting to note that, despite the fact that the total work  $W_{\text{total}}$  enters the denominator of Eq. (31), the correlation between efficiency  $\eta$  and the total work  $W_{\text{total}}$  is still positive ( $R = 0.71$ ), mainly because the increased total work leads to an increased useful work (correlation of  $R = 0.79$ ) and an increased propulsive force ( $R = 0.85$ ). For the lateral force, we present the correlations with its absolute value, since its sign (indicating whether the total lift acts to the right or to the left with respect to the fish motion) does not matter in regard to efficiency. We note that the correlation between  $\eta$  and  $|\bar{F}_y|$  is high ( $R = 0.98$ ), mainly, because high lift is associated with more energetic undulations, which also positively correlate with the propulsive force and the useful work.

### 6.3.2. Relation of efficiency to kinematics

We further analyze the details of the optimum and sub-optimum swimming by selecting three specific kinematic modes,  $\alpha$ ,  $\beta$  and  $\gamma$ , which were previously introduced in Table 3, and comparing their kinematic and hydrodynamic trends with the optimum mode. The modes  $\alpha$ ,  $\beta$  and  $\gamma$  were selected for their distinct kinematic characteristics, which will allow us to highlight important relationships between the body motion and efficiency, as discussed below.

The kinematics of the four selected modes is shown in Fig. 13, where the shape of the swimmer's midline is illustrated for each  $1/10^{\text{th}}$  of a period during one undulation cycle  $T$ . It can be seen that the three more efficient modes, optimum,  $\alpha$  and  $\beta$ , all have comparably large tail amplitudes, while the least efficient mode,  $\gamma$ , out of the four analyzed, has a very small tail amplitude motion. The first three modes, however, apart from having similar tail amplitudes, exhibit other marked differences in the body line kinematics, which, as will be seen below, effect their efficiency.

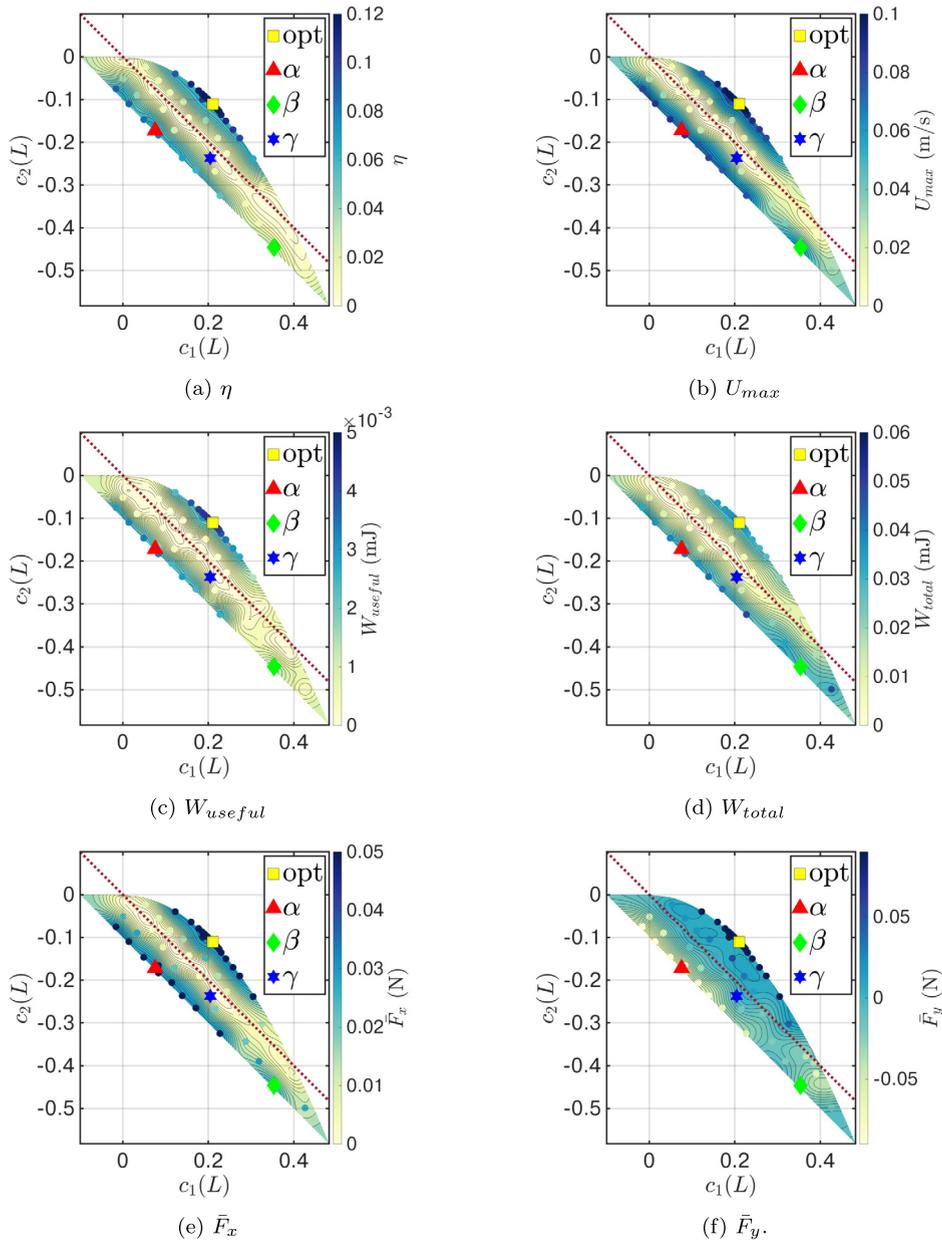
Table 5 reports on the hydrodynamic quantities of the optimum and  $\alpha$ ,  $\beta$ ,  $\gamma$  modes with  $N = 9$ , including the Reynolds number at the maximum speed  $Re_{\text{max}}$ , defined as

$$Re_{\text{max}} = \frac{U_{\text{max}}L}{\nu}, \quad (37)$$

where  $\nu = \mu/\rho = 1 \times 10^{-6}$  ( $\text{m}^2/\text{s}$ ) is the kinematic viscosity of water. In addition to the absolute quantities for each mode, Table 5 also documents the relative quantities of the sub-optimum modes normalized by the respective quantity of the optimum mode. In Table 5,  $A$  denotes the tail amplitude of a corresponding kinematic mode, with the other hydrodynamic quantities as defined previously. The results indicate that, indeed, a small tail amplitude ratio ( $A_\gamma/A_{\text{opt}} \approx 0.33$ ) is associated with a reduced efficiency ( $\eta_\gamma/\eta_{\text{opt}} \approx 0.05$ ), consistent with the findings in [1]. Regarding the relation of efficiency to the amplitude, it was previously established that the thrust coefficient scales with the amplitude squared,  $A^2$ , for the flapping foil propulsion [75–77], and these arguments were extended to derive the scaling laws for the fish locomotion [19,23,78].

However, quadratic scaling with the tail amplitude can not explain the efficiency of all the realizable modes. For example, if we look back at Table 5, we recognize that the optimum,  $\alpha$  and  $\beta$  modes all have a comparable tail amplitude, however, their efficiency varies by as much as five times.

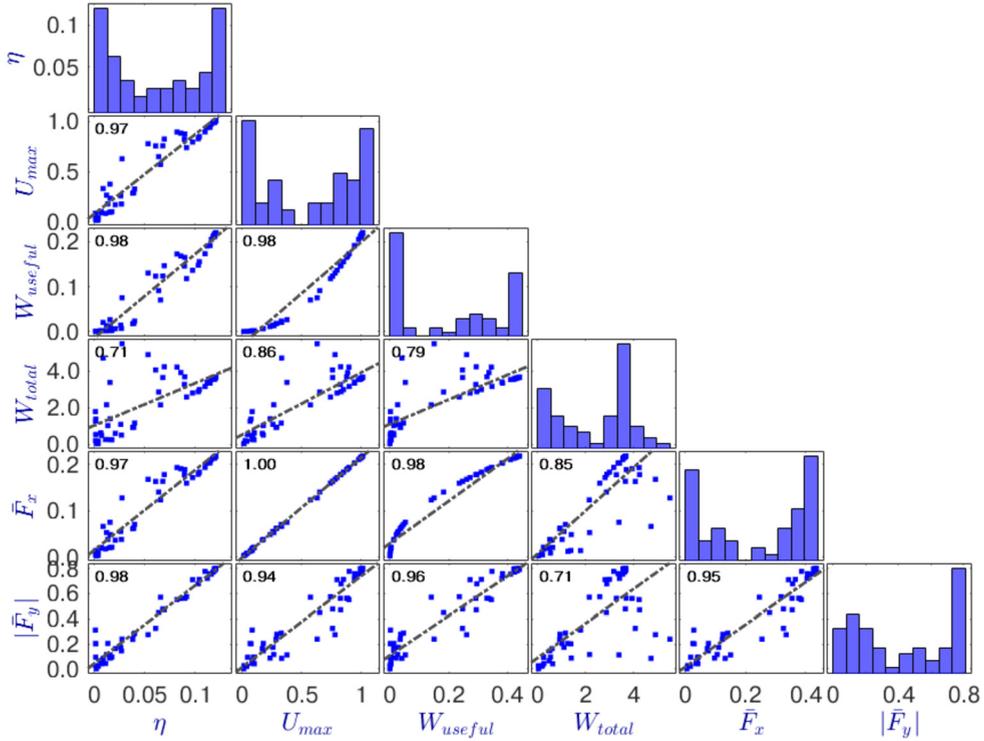
To illustrate this phenomenon, we consider the mode  $\beta$ , which has a tailbeat amplitude of  $A_\beta/A_{\text{opt}} \approx 0.93$ , however only one fifth of efficiency of the optimum mode ( $\eta_\beta/\eta_{\text{opt}} \approx 0.23$ ). To understand where the efficiency losses might come from, we look at the total work and the useful work ratios of the mode  $\beta$  provided in Table 5. We see that the efficiency losses are not fully attributed to a larger amount of the total work ( $W_{\text{total}_\beta} = 1.50$ ), but are instead mostly effected by a lower propulsive work ( $W_{\text{useful}_\beta} = 0.34$ ). The lower propulsive work in the sub-optimum  $\beta$ -swimmer corresponds to a lower propulsive force ( $\bar{F}_{x_\beta} = 0.58$ ) and a lower maximum speed ( $U_{\text{max}_\beta} = 0.62$ ). This is consistent with the results of the correlation analysis in Fig. 12.



**Fig. 11.** Hydrodynamic quantities,  $t_0 = 2T$ : (a)  $\eta$ ; (b)  $U_{max}$ ; (c)  $W_{useful}$ ; (d)  $W_{total}$ ; (e)  $\bar{F}_x$ ; (f)  $\bar{F}_y$ ; evaluated for the entire design domain using Kriging interpolation between the computed CFD data points with  $N = 9$ , shown as circles color-coded by the corresponding values. The colored symbols in each figure represent the optimum (square),  $\alpha$  (triangle),  $\beta$  (diamond) and  $\gamma$  (star) modes selected for comparison in Table 3. Red dotted line corresponds to  $c_1 + c_2 = 0$ .

### 6.3.3. New scaling law for optimum and sub-optimum propulsion

As seen from the previous section, while scaling of the thrust with the tailbeat square amplitude  $A^2$  might explain the optimum propulsion between the species, it is not necessarily applicable to a sub-optimum motion, as the efficiency results might be very different for the same tailbeat amplitude. To understand this phenomenon, we look back at the shape of the midline deformation of the swimmer over the undulation cycle presented in Fig. 13. It can be seen that, between the optimum,  $\alpha$  and  $\beta$  modes, while the tail amplitudes are similar, there are marked differences in the body line kinematics. We note, for example, an existence of a “fixed point” in the midline deformation curves of sub-optimum modes, where the amplitude of motion is zero. This fixed point essentially separates the motion of the tail from the motion of the body. Since in thunniform swimming, most of the propulsion comes from the tail [43,79,80], we argue that the existence of this fixed point essentially reduces the effective propulsive length of the fish from its full length  $L$  to a “flapping length”  $L_f$ , defined



**Fig. 12.** Correlation matrix of hydrodynamic quantities with the respective Pearson correlation coefficient ( $R$ ) in black inside the corresponding squares. Histograms of each variable are found in the diagonal elements. Histograms are composed of 10 uniformly-distributed bins between the minimum and the maximum value for each variable.

as the distance between the fixed point and the tip of the tail. Conferring with the equation of the midline deformation, Eq. (2), we identify the axial coordinate where the deformation is zero (fixed point) as

$$x_f = -\frac{c_1}{c_2} L, \quad (38)$$

and therefore we can define the flapping length as

$$L_f = \begin{cases} (1 + \frac{c_1}{c_2}) L, & c_1 + c_2 \leq 0 \\ L, & c_1 + c_2 > 0. \end{cases} \quad (39)$$

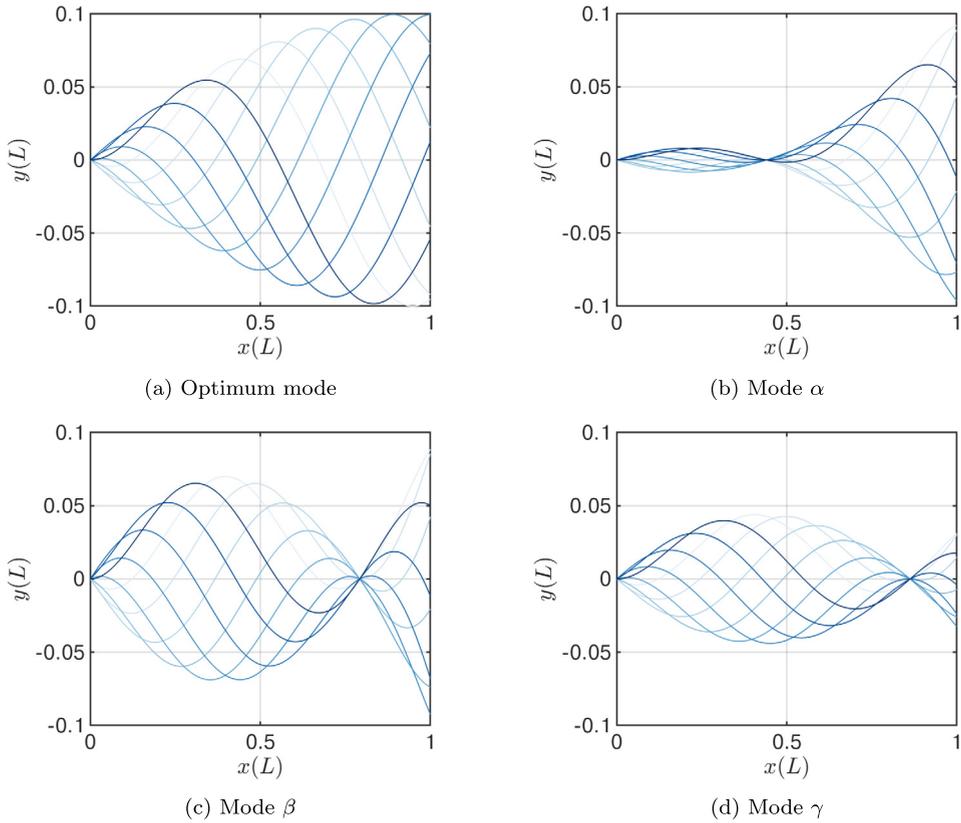
Eq. (39) signifies that if the fixed point defined by Eq. (38) lies outside of the fish body (if  $c_1 + c_2 > 0$ , such as, for example, in the optimum mode), we constraint the flapping length to be equal to the fish length  $L_f = L$ .

Following the scaling arguments presented by Gazzola et al. [23], we define the mass of the fluid set in motion by the fish tail propulsion as  $\rho L_f^2$  per unit depth, acceleration of the fluid as  $A\omega^2$ , and the local angle between the tail and the direction of motion as  $A/L_f$ . Thus, we arrive at the thrust force scaling as  $T \sim \rho \omega^2 L_f A^2$ . Since drag is independent of the tail amplitude [23,74], propulsion velocity  $U$  is proportional to the net thrust, and the total work in the denominator of Eq. (31) roughly scales with the hydrodynamic forces, we can conclude that the efficiency  $\eta$  should obey the scaling  $\eta \sim C_\eta L_f A^2$ , where  $C_\eta$  is some dimensional constant of proportionality.

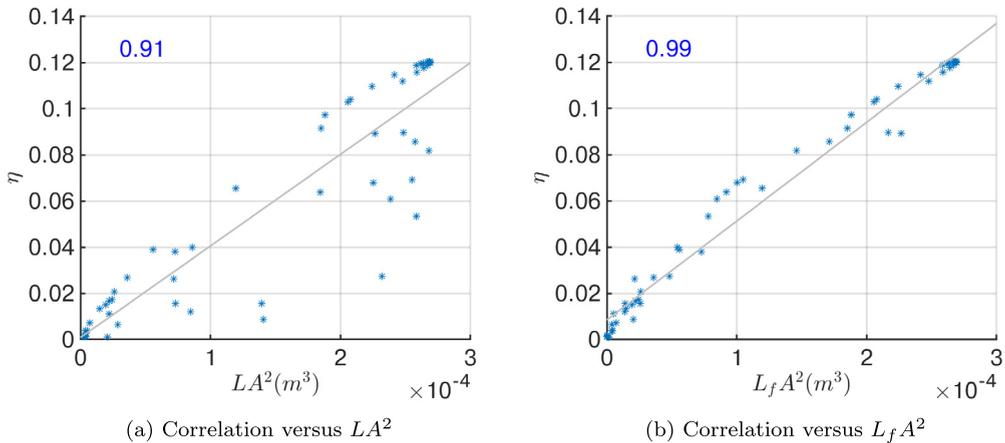
Indeed, using a standard propulsion law  $\eta \sim LA^2$ , we obtain a relative efficiency for the mode  $\alpha$  with respect to the optimum mode as  $\eta_{\alpha 1} = 0.94$  instead of the observed  $\eta_\alpha = 0.66$ . However, scaling with  $\eta \sim L_f A^2$  results in  $\eta_{\alpha 2} = 0.56$ , which is much closer to the observed efficiency. Likewise, for the mode  $\beta$ , we would have  $\eta_{\beta 1} = 0.86$  from the original scaling law, and  $\eta_{\beta 2} = 0.18$  with the modified scaling, which is much closer to the actual  $\eta_\beta = 0.23$ .

Plotting a correlation of efficiency with  $LA^2$  (unmodified propulsion law) in Fig. 14a shows that a correlation with  $LA^2$  is a reasonable rule of thumb ( $R = 0.91$ ), but it still has a lot of outliers, such as the modes  $\alpha$  and  $\beta$  discussed here. However, correlating efficiency with  $L_f A^2$  in Fig. 14b provides a perfect agreement with the correlation coefficient of  $R = 0.99$ . Putting this result into a perspective of Fig. 11a, we see that the modes that lie close to the line  $c_1 + c_2 = 0$  have a low tail amplitude,  $A$ , while the modes that additionally lie below this line (modes with  $c_1 + c_2 \leq 0$ ) have a restricted flapping length,  $L_f < L$ . These two criteria define a region of the lowest efficiency as a narrow strip bordering the line  $c_1 + c_2 = 0$  and extending immediately below it, with the mode  $\gamma$  being one of the least efficient modes as it belongs to this region.

This result, first, shows that simple fluid mechanics scaling laws, with a slight modification, are applicable to a wide range of realizable (prescribed) undulatory motions of the aquatic swimmer in the viscous flow, optimum as well as sub-



**Fig. 13.** Swimmer midline deformation across one time cycle for four propulsive modes: (a) Optimum; (b)  $\alpha$ ; (c)  $\beta$ ; and (d)  $\gamma$ . Deformations of the midline in time are encoded every  $1/10^{th}$  of the period in the different shades of blue from lightest ( $t = 0$ ) to darkest ( $t = T$ ).

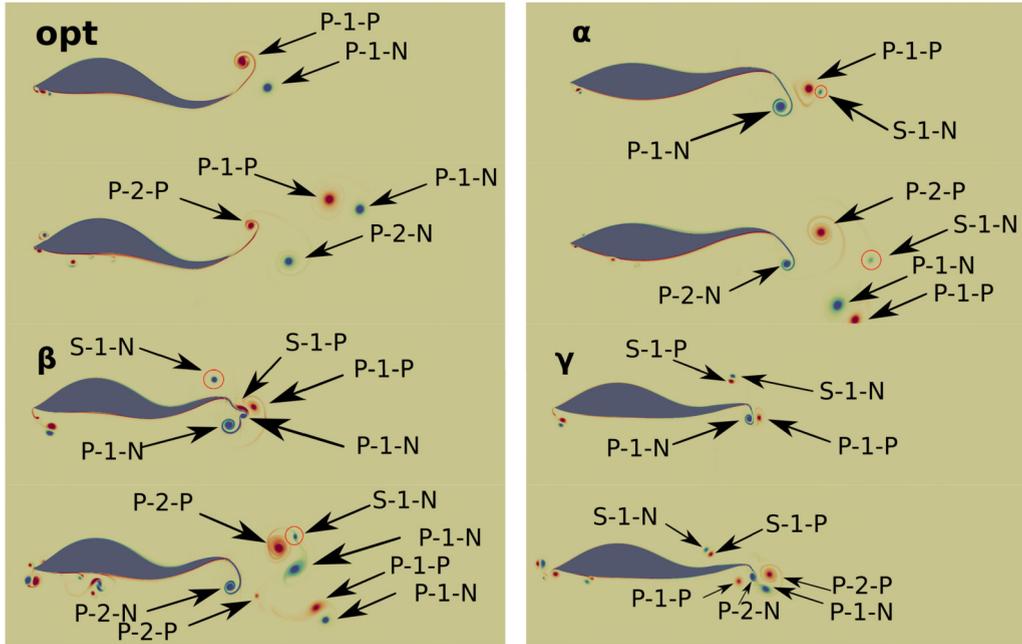


**Fig. 14.** Correlation of efficiency  $\eta$  versus a)  $LA^2$  and b)  $L_fA^2$ . Pearson correlation coefficient (R) in shown in blue inside each plot.

optimum. And, second, it provides the guidance for the optimum design of the undulatory motions of a swimmer: while maximizing the tailbeat amplitude under allowable constraints, it should also maximize the propulsive area and avoid the fixed points in the body deformation curve.

6.3.4. Vortex wakes

We now analyze the vortex wakes produced by the four selected kinematic modes, to understand the relationships between the swimming efficiency and vorticity dynamics. The vortex wakes produced by the four modes are shown in Fig. 15. We present two snapshots, at  $t = T = 1$  s and  $t = 2T = 2$  s, respectively, to show the dynamics of the pair of positive and negative vortices resulting from each stroke. For the optimum mode, after 1 s, the first pair of primary positive (P-1-P)



**Fig. 15.** Vorticity (red – positive, blue – negative) in the wake of the fish for the four modes with  $N = 9$ . Maximum and minimum values of vorticity are  $100 \text{ s}^{-1}$  and  $-100 \text{ s}^{-1}$ , respectively. Top row is at  $t = T = 1 \text{ s}$  and bottom row is at  $t = 2T = 2 \text{ s}$  for each mode. Vortices are named with a 'A-B-C' convention, where A is either a 'P' primary or 'S' secondary vortex, B is an indication that a vortex appears after either the '1' first or the '2' second time cycle and 'C' is either a 'P' positive or 'N' negative vortex.

and negative (P-1-N) vortices behind the optimum swimmer is advected at an angle, which is expected since the swimmer generates its first stroke from an equilibrium position. After 2 s, the P-1-P and P-1-N vortices are of comparable magnitude and they behave similarly to a pair of point vortices, which propel each other in the negative streamwise direction, thus propelling the fish in the positive streamwise direction. The second primary vortex pair (P-2-P and P-2-N) begins to shed at  $t = 2 \text{ s}$  in a fashion similar to the reverse Karman street wake associated with high propulsive efficiency [1,21,81].

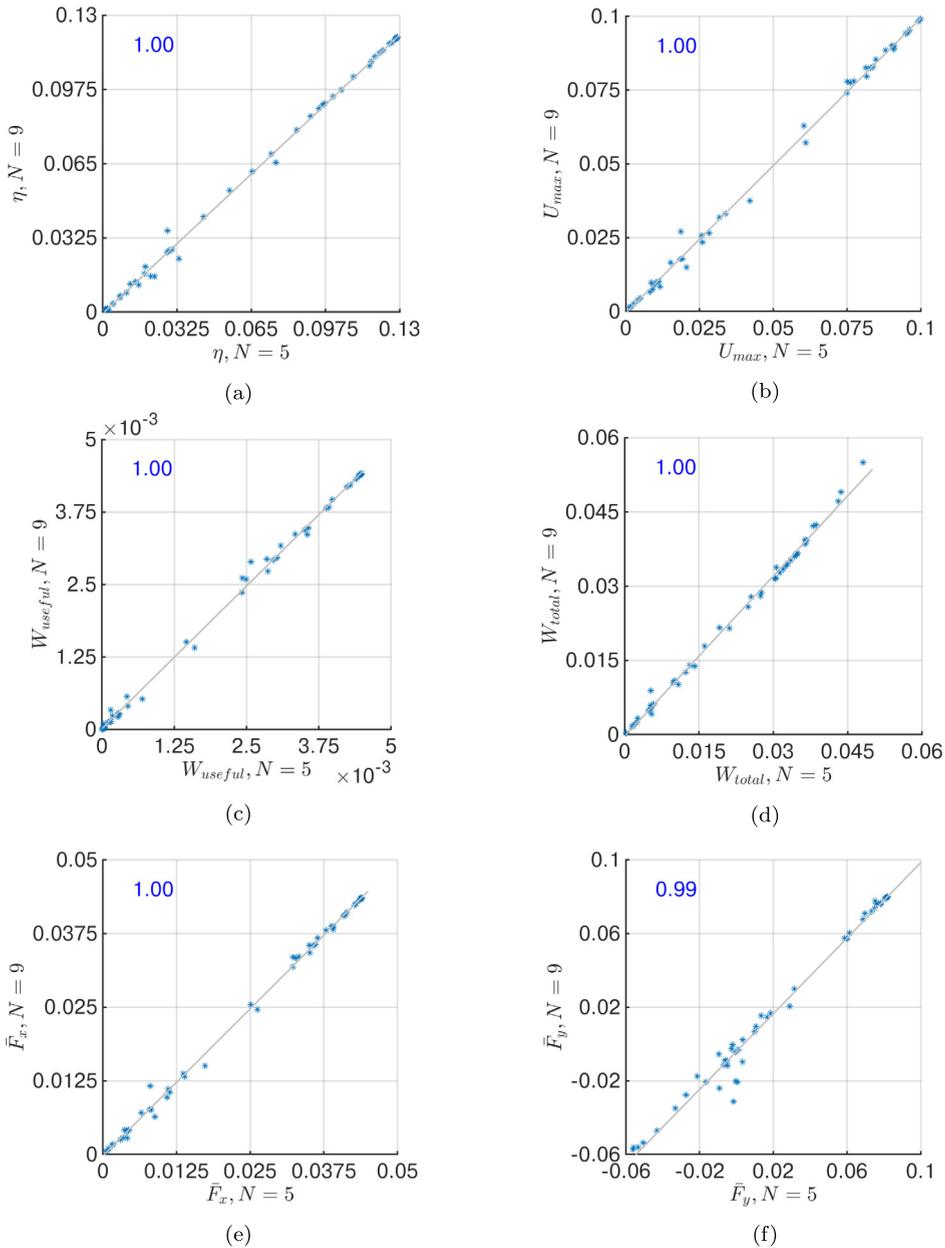
Out of the three additional modes considered, only mode  $\alpha$ , second in rank with respect to efficiency after the optimum mode, results in a similar reverse Karman street. However, a smaller secondary negative vortex (S-1-N) is observed after the first stroke of the  $\alpha$ ,  $\beta$  and  $\gamma$  swimmers. Effect of the S-1-N vortex varies between the modes. For the  $\alpha$  mode, S-1-N is first attached to P-1-P at 1 s but is observed to be isolated at 2 s. However for the  $\beta$  mode, the secondary vortex effects are more prominent, and in fact disrupt the wake by separating and redirecting of the primary vortices (note, for example, two instances of P-1-N and P-2-P vortices at the end of the second cycle, which got split up and steered away from each other during the undulation). Despite having a sufficient tail amplitude ( $A_{\beta_r} = 0.93$ ), mode  $\beta$  fails to produce a reverse Karman street wake and consequently results in a low propulsive efficiency. This might be associated with a lower effective flapping length, as discussed above. Mode  $\gamma$  leaves the wake with minute vortices due to small tail oscillations that lack a propulsive power. These small oscillations result in the lowest propulsive efficiency between the four modes ( $\eta_{\gamma_r} = 0.05$ ). Since the secondary vortex formation is observed in all the considered modes except for the optimum one, we can conclude that secondary vortices act to disrupt the formation of an efficient reverse Karman vortex street and thus interfere negatively with the efficient propulsion. A negative effect on propulsion can also be attributed to a formation of the head vortices, especially noticeable in the two least efficient,  $\beta$  and  $\gamma$ , modes. While head vortices do not interfere with the wake, they presumably affect the hydrodynamic forces and the energy expenditure of the swimmer. Finally, it is worth noting that secondary tail vortex formation is only observed during the first cycle for all the modes, and might be associated with the onset of acceleration, while the shedding of the head vortices is noticed for both the first and the second cycles.

#### 6.4. Comparison between $N = 5$ and $N = 9$ cases

A lower polynomial order  $N = 5$  case is compared here to  $N = 9$  case at  $t_0 = 2T$  to establish grid convergence and to analyze the sensitivity of the optimization results to a polynomial degree in a high-order method.

##### 6.4.1. Hydrodynamic quantities

The correlations of the hydrodynamic quantities between  $N = 5$  and  $N = 9$  cases are presented in Fig. 16. To compute the correlations, we interpolate the data obtained from the CFD evaluation points of  $N = 9$  case to the evaluation points of  $N = 5$  case yielded by the optimization procedure.



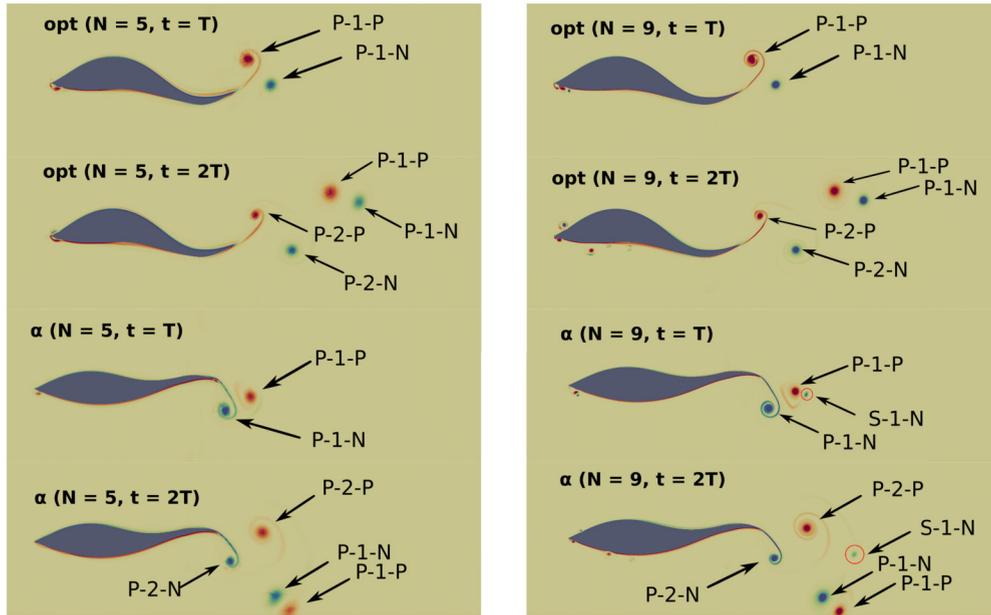
**Fig. 16.** Correlation between hydrodynamic quantities obtained with  $N = 9$  (y-axis) and  $N = 5$  (x-axis) spectral element method for  $t_0 = 2T$ . Pearson correlation coefficient is highlighted in blue on each plot.

We observe an excellent agreement between the hydrodynamic data of  $N = 5$  and  $N = 9$  cases, with the correlation coefficient of  $R \geq 0.99$ . Among all presented quantities, lift exhibits the most scatter, although it still yields a correlation coefficient of  $R = 0.99$ . The scatter is primarily in the region close to zero, where the actual value of the lift is so low, that it is most prone to numerical uncertainties.

Table 6 includes the hydrodynamic quantities for the optimum and the three selected sub-optimum modes, see Table 3, for  $N = 5$  case, which can be compared with the corresponding values for  $N = 9$  case presented in Table 5. A most notable difference is a slight upward shift of the efficiency values computed for all the four modes with  $N = 5$ , accompanied by (and perhaps attributed to) a slight downward shift of the total work. The other quantities, including propulsive and lift forces, useful work, and maximum velocity, do not show consistent deviations. The fact that the viscous forces are predicted well by the lower polynomial order solution but the total work deviates slightly might be an indication that it is the resolution of the interaction between the viscous forces and the local deformations of the moving surface, which defines the total work, that benefits the most from a higher-order approximation. Note that, since the resulting shift in efficiency is monotonic, the

**Table 6**  
Hydrodynamic quantities of selected kinematic modes with  $N = 5$ ,  $t_0 = 2T$ .

	$\eta$	$W_{\text{useful}}$ (mJ)	$W_{\text{total}}$ (mJ)	$U_{\text{max}}$ (m/s)	$\bar{F}_x$ (N)	$\bar{F}_y$ (N)	$Re_{\text{max}}$
opt	12.90%	4.492	34.81	0.102	0.044	0.082	$3.06 \times 10^4$
$\alpha$	8.49%	3.091	36.42	0.085	0.037	-0.055	$2.55 \times 10^4$
$\beta$	3.04%	1.462	48.80	0.060	0.025	-0.010	$1.80 \times 10^4$
$\gamma$	0.75%	0.044	5.90	0.011	0.005	-0.004	$3.30 \times 10^3$



**Fig. 17.** Vorticity (red – positive, blue – negative) in the wake of the fish for the optimum and  $\alpha$  modes with  $N = 5$  (left) and  $N = 9$  (right). Maximum and minimum values of vorticity are  $100 \text{ s}^{-1}$  and  $-100 \text{ s}^{-1}$ , respectively. The vortices follow the naming convention of Fig. 15.

lower-order  $N = 5$  approximation predicts the correct relative efficiency rankings between the modes, illustrates a perfect correlation with the high-order  $N = 9$  case for all the hydrodynamic quantities considered, and identifies the same optimum mode, as seen from Table 2.

#### 6.4.2. Vortex wakes

Fig. 17 shows the wakes produced by the optimum mode and the mode  $\alpha$  at  $t = 1 \text{ s}$  and  $t = 2 \text{ s}$  computed with  $N = 5$  and  $N = 9$  polynomial orders. For the optimum mode, at  $t = 1 \text{ s}$ , the wakes behind each swimmer appear nearly identical. Towards the head, however, there are less small vortices resolved using a lower order approximation. At  $t = 2 \text{ s}$ , a similar pattern is observed where the smaller vortices are missing at  $N = 5$ . The reverse von Karman street wake appears after the second stroke both for  $N = 5$  and  $N = 9$  cases, and the first pair of vortices are able to propel each other due to closely matched circulation strengths in both cases. A better capture of small vortices in a higher-order approximation case is in line with the prediction of a higher total work of the optimum swimmer and, consequently, a lower maximum efficiency by  $N = 9$  solution. Since these small vortices do not interfere with the wake, an almost identical kinematic gait mode is recovered as the optimum with both polynomial resolutions.

The wake of the  $\alpha$  mode similarly remains unchanged with one notable exception: the negative secondary vortex (S-1-N) is less visible in  $N = 5$  case at both time instances. Before  $t = 1 \text{ s}$ , the smaller negative vortex (S-1-N) generates and stays attached to the primary positive vortex (P-1-P) as the swimmer moves from its static position to an upward stroke position (not shown here). As the swimmer completes its stroke, at  $t = 1 \text{ s}$ , the secondary negative vortex (S-1-N) appears to have been stretched around the positive vortex (P-1-P) and is barely visible. At  $t = 2 \text{ s}$ , the first pair of vortices (P-1-N and P-1-P) float in the negative streamwise direction while the second pair (P-2-N and P-2-P) initiate a reverse von Karman street wake, as in the  $N = 9$  case. Since the appearance of S-1-N vortex in  $\alpha$ ,  $\beta$  and  $\gamma$  modes was previously associated with a lower efficiency, its under-resolution in  $N = 5$  case might, again, explain the increased efficiency trends with a lower polynomial order.

In the  $\beta$  mode, shown in Fig. 18, while a weak S-1-N vortex can be seen at  $t = 1 \text{ s}$  with  $N = 5$ , the overall vorticity pattern is notably simpler as compared to the  $N = 9$  case, where a complicated pattern of vortex breakdown and merging was observed. While the S-1-N vortex is primarily dissipated by  $2 \text{ s}$ , its original formation still caused the primary vortex (P-1-P) to separate from the wake. Therefore, the  $\beta$  mode was similarly unable to produce an efficient reverse von Karman

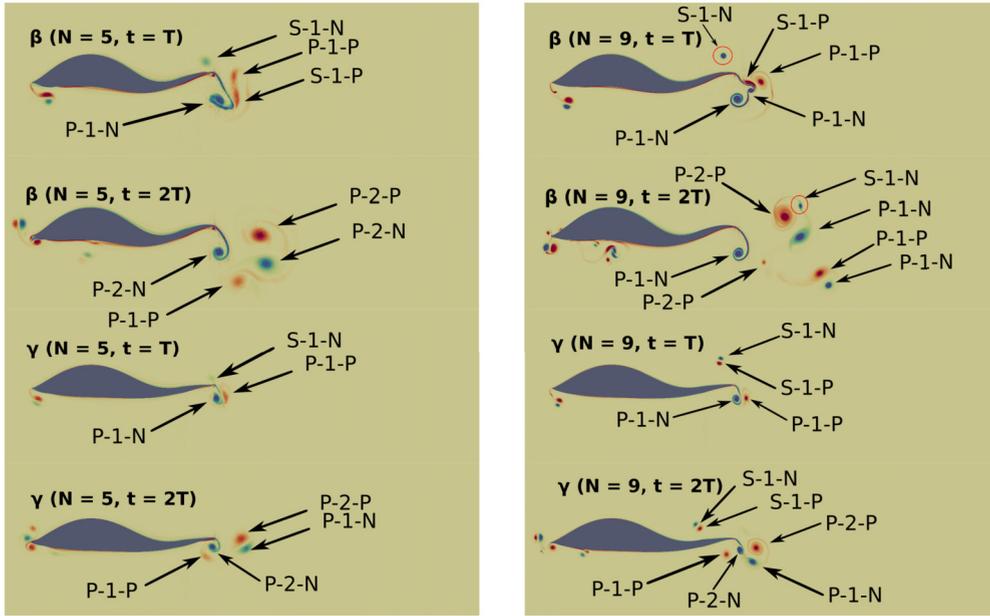


Fig. 18. Vorticity (red – positive, blue – negative) in the wake of the fish for the  $\beta$  and  $\gamma$  modes with  $N = 5$  (left) and  $N = 9$  (right). Maximum and minimum values of vorticity are  $100 \text{ s}^{-1}$  and  $-100 \text{ s}^{-1}$ , respectively. The vortices follow the naming convention of Fig. 15.

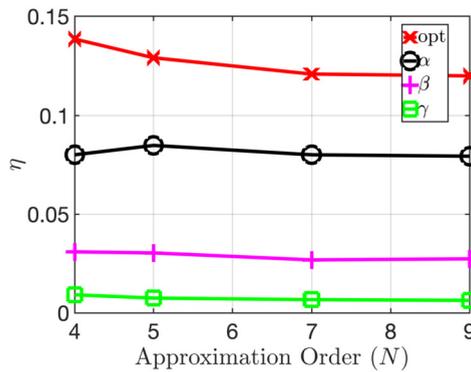


Fig. 19. Propulsive efficiency,  $\eta$ , for the optimum,  $\alpha$ ,  $\beta$ , and  $\gamma$  kinematic modes evaluated using various polynomial orders,  $N$ , for  $t_0 = 2T$ .

wake with  $N = 5$ . As before, a somewhat weaker S-1-N vortex and an under-resolution of a vortex breakdown with a low polynomial order resulted in a slightly higher predicted efficiency in  $N = 5$  case. The same trend persists for the  $\gamma$  mode, where, while the overall wake structure is similar between  $N = 5$  and  $N = 9$  cases, the  $N = 5$  case resolves significantly less details regarding the secondary vorticity both around the head and the tail of the fish, resulting in a slightly higher predicted efficiency.

### 6.5. Comparison with other polynomial orders

To extend comparison to other polynomial orders, including lower orders, Fig. 19 presents the relative ranking of the optimum mode and the three selected sub-optimum modes ( $\alpha$ ,  $\beta$  and  $\gamma$  of Table 3) across the polynomial orders of  $N = 4, 5, 7$  and  $9$  at  $t_0 = 2T$ . This figure reveals that for all the polynomial orders tested, including  $N = 4$ , the relative efficiency rankings between the four modes do not change, implying that optimization with as low as fourth-order accuracy simulations would still produce reliable results within the current SEM framework. Lower values of  $N$  can not be tested with Nek5000, at least with the current  $\mathbb{P}_N - \mathbb{P}_{N-2}$  formulation, as the first-order polynomials for pressure are inconsistent with the existing algorithms. Third- and lower-order accuracy solutions with optimization procedure should be tested in the future work with conventional finite-difference and finite-volume codes to establish an acceptable accuracy threshold on a CFD solver to yield reliable results with optimization.

## 7. Conclusions

This paper presents a development of a new computational framework for high-fidelity optimization of the kinematic gaits of locomotion for a self-propelled undulatory swimmer. The framework is based on a spectral-element solution of the underlying fluid-structure problem, coupled to an efficient and accurate derivative-free surrogate based optimization framework. A new volume-conservation scheme is developed for a fish geometry update during the undulations. Not only it ensures that a physiologically relevant constraint on a neutrally-buoyant swimmer is satisfied during the simulations [46,47], but allows one to increase the efficiency of the fluid-structure solver by avoiding implicit pressure-velocity iterations in the moving mesh formulation, otherwise required for stability purposes [40,48]. With the volume conservation approach, a traditional explicit operator-splitting scheme in the arbitrary Lagrangian-Eulerian formulation [38,39] remained stable for all the realizable swimmer motions emerged during the optimization process, while three to four pressure-velocity iterations were required in a non-conservative scheme. While the total computational savings from the volume conservation scheme reported in the current problem were 20%, they can be larger if, for example, more pressure-velocity iterations are required by a non-conservative approach, or if the pressure convergence within the iterations is slow. The developed framework has been applied to an optimization of accelerated propulsion of the flexible swimmers, which received significantly less attention in the literature as compared to a steady swimming [24,25]. To focus our attention on the regime of intense propulsion (high acceleration), the investigated swimming modes were optimized over two swimming cycles, although doubling of the optimization period did not change the optimum mode (up to a tolerance of 1%). As the swimmer accelerates, however, the efficiency of the optimum propulsive mode first increases up to a period of approximately  $10T$ , when it peaks and starts gradually decreasing. This occurs since the relatively energetic optimum start-up motions, while keep requiring high energy inputs, start yielding decreased efficiency gains at lower values of acceleration (thus lower useful work). It is envisioned that the swimming time of approximately  $10T$  may demarcate the regime of efficient acceleration from the regime of efficient (quasi)-steady swimming. It may therefore be recommended, depending on the goals, that a mode of locomotion be switched from that of efficient propulsion to that of efficient steady swimming (or a low cost of transport) [8,48] as swimming progresses.

Another important contribution of this work is an assessment of the sensitivity of the optimization results to the accuracy of approximation of the CFD solution, which is a key consideration, nonetheless lacking in modern bio-inspired optimization studies [9,10,13]. It was shown that an increase in the polynomial order of approximation of the CFD solution in a spectral-element method does not change the optimum gaits or the predicted efficiency rankings between the modes. However, it does result in the overall lower predicted efficiency. The reason for that is that the lower-order solution misses some details of the interaction between the viscous and pressure forces resulting from the swimmer's undulation, with the undulating motion itself, which leads to an under-prediction of the total work and over-prediction of efficiency. This conclusion is consistent with the previous observations that lower-order models overpredict the swimming speed and efficiency, and underpredict the input power [20,21], but otherwise are capable of qualitatively capturing the trends of locomotion. However, more work needs to be done on assessment of the accuracy of CFD and optimization results in this complicated problem, since the spectral-element method with a "low polynomial order" of  $N = 5$  still corresponds to a higher fidelity than most of the methods previously used for the computational studies of fish propulsion. In this regard, a comparison of the optimum mode of locomotion and three selected sub-optimum modes did not change in the current study for the polynomial orders down to  $N = 4$ . Lower-accuracy solutions could not be assessed here due to restrictions of the current methodology, and need to be evaluated using traditional low-order codes.

In regards to the vortex wakes, the optimum mode shows an expected reverse Karman vortex street wake conducive to high propulsive efficiency [1,21,81]. The wakes of sub-optimum modes vary considerably, with more efficient modes showing a resemblance of a reverse Karman street wake, albeit less effective. The less efficient modes do not exhibit reverse Karman street wake and shed disorderly vortices that sweep around the fish and are failed to propel away. A higher polynomial order solution resolved more details regarding a formation of small secondary vortices around the fish head and body, and a vortex breakdown around the tail, which ultimately showed a decreased efficiency of the propulsive wake, consistent with the fact that the lower-order methods under-resolve the details of the complex fluid-body interactions and overestimate efficiency.

Regarding the relation of efficiency to the swimmer's kinematics, a new scaling law is proposed in this paper, that takes into account the flapping length of the swimmer, as opposed to the total length, which can be quite different from the total length in inefficient propulsion. With this modification, the new scaling law shows an almost ideal  $R = 0.99$  correlation between the efficiency and the mode kinematics. This provides a useful guidance for the design of efficient flexible propellers, suggesting that the tail amplitude must be maximized given the available constraints, but also that the body motion must be designed such that the flapping length is maximized, and there are no "fixed points" in the body deformation curve. We would like to caution the reader that, again, while the scaling laws based on very rudimentary fluid mechanics principles provide a very good interpretation of the efficiency trends, it is the actual values of efficiency which can not be well predicted by the scaling laws and require a high-fidelity resolution of the pertinent fluid-body interactions.

Finally, an SBO algorithm was shown to be well suited to the optimization problem at hand and was able to provide optimal solutions, converged within a specified tolerance, with a low computational budget.

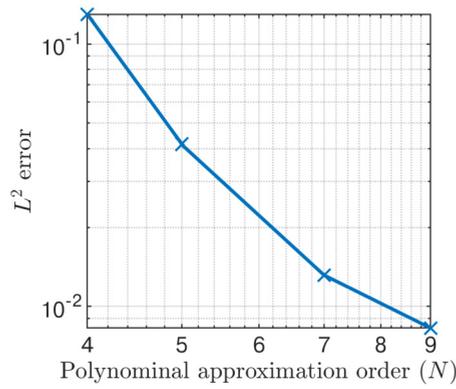


Fig. A.20.  $L^2$  error norm of streamwise velocity under polynomial refinement,  $\delta t = 2 \times 10^{-5}$  seconds.

### CRediT authorship contribution statement

**Ahmed Abouhussein:** Investigation, Methodology, Software, Validation, Writing – original draft. **Yulia T. Peet:** Funding acquisition, Resources, Supervision, Writing – review & editing.

### Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

### Data availability

Data will be made available on request.

### Acknowledgements

This work has been supported by NSF CMMI-1762827 grant. Computational time has been provided by the ASU Research Computing Center on the parallel computing cluster Agave.

## Appendix A. Verification of the numerical algorithm

### A.1. Spatial and temporal convergence analysis

This section presents a spatial and temporal convergence study of the numerical procedure outlined in Section 3 using the simulation setup of Section 4. The full algorithm, including the mesh deformation and the self-propulsion, is evaluated in the convergence analysis. All verification studies are performed using the optimum mode obtained with  $N = 5$  and  $t_0 = 2T$  (see Table 2).

The spatial convergence is studied through a polynomial refinement (“p-refinement”), using a baseline computational grid of Section 4.2. P-refinement is a standard convergence verification procedure as applied to spectral and high-order finite-element methods [59,82]. The error metrics in the current study is taken to be the  $L^2$  error norm of streamwise velocity calculated as

$$L^2(u_x) = \sqrt{\left. \frac{\int_{\Omega_f} (u_x^e(\mathbf{x}, t) - u_x(\mathbf{x}, t))^2 d\Omega}{\mathcal{V}_f} \right|_{t=2T}}, \quad (\text{A.1})$$

where  $u_x^e$  is the “exact” (reference) streamwise velocity,  $\mathcal{V}_f$  is the fluid volume of the domain  $\Omega_f$  and the notation  $|_{t=2T}$  signifies that the error is evaluated at the time  $t = 2T$ . Due to an absence of an analytical solution in the current problem, the solution at the polynomial order  $N = 10$  is taken as the reference solution. Lagrange interpolation is used to interpolate a reference solution to a lower-order polynomial grid for each  $N$  for the error computation. The time step used is  $\delta t = 2 \times 10^{-5}$  seconds. Fig. A.20 documents the results of the polynomial refinement. Overall, the convergence rate is consistent with a spectral approximation, especially at low orders, while it slightly drops at higher  $N$ , which can be associated either with the challenges of high-order approximation on highly irregular deforming meshes, or an artifact of a self-convergence test.

Temporal convergence study is performed with the polynomial order of  $N = 5$ . The error is calculated the same way, using Eq. (A.1), albeit the reference solution is taken from the same polynomial order  $N = 5$  but at a finer time step of

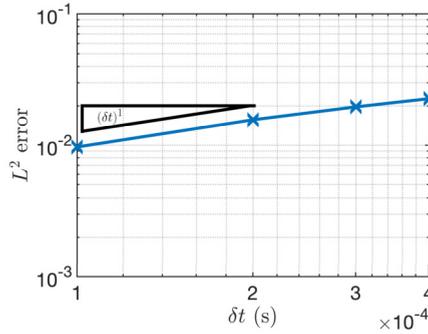


Fig. A.21.  $L^2$  error norm of streamwise velocity under temporal refinement,  $N = 5$ .

**Table A.7**  
Numerical parameters for the baseline and the test mesh deformation schemes, Eqs. (21), (22), and the errors between the two simulations.

Deformation scheme	$x_1$	$x_2$	$y_1$	$y_2$
Baseline	0.3	0.6	0.1	0.2
Test	0.6	0.9	0.2	0.3
Error	$L^2(u_x)$		$L^1(\eta)$	
	$2.74 \times 10^{-2}$		$1.6 \times 10^{-3}$	

$\delta t = 2 \times 10^{-5}$  seconds, to separate spatial from temporal errors. Results are presented in Fig. A.21. The temporal convergence is of the expected first order, set by the first-order ALE geometry update in Eq. (23).

### A.2. Sensitivity to mesh deformation scheme

To assess sensitivity of results to the mesh deformation algorithm, we compare two moving mesh simulations, both defined by the blending function of Eqs. (21), (22), but with two different set of parameters as listed in Table A.7. The same computational setup of a self-propelled undulatory swimmer with  $N = 5$  and the optimum mode of  $t_0 = 2T$  is utilized, as in A.1. In a similar way, we define an  $L^2$  error of streamwise velocity as

$$L^2(u_x) = \sqrt{\left. \frac{\int_{\Omega_f} (u_x^t(\mathbf{x}, t) - u_x(\mathbf{x}, t))^2 d\Omega}{\mathcal{V}_f} \right|_{t=2T}} \tag{A.2}$$

where  $u_x$  refers to a streamwise velocity obtained with the baseline deformation scheme, and  $u_x^t$  refers to a streamwise velocity obtained with a test deformation scheme. We also define an  $L^1$  error in efficiency as

$$L^1(\eta) = \left| \frac{\eta^t(t_0) - \eta(t_0)}{\eta(t_0)} \right|, \tag{A.3}$$

with  $\eta^t$  signifying the efficiency of the test deformation scheme, and  $t_0 = 2T$ . Both the  $L^2(u_x)$  and  $L^1(\eta)$  metrics in Table A.7 indicate that the numerical results are fairly insensitive to the choice of a mesh deformation algorithm.

### References

- [1] M.S. Triantafyllou, G. Triantafyllou, D. Yue, Hydrodynamics of fishlike swimming, *Annu. Rev. Fluid Mech.* 32 (1) (2000) 33–53.
- [2] F. Fish, G.V. Lauder, Passive and active flow control by swimming fishes and mammals, *Annu. Rev. Fluid Mech.* 38 (2006) 193–224.
- [3] S. Coyle, C. Majidi, P. LeDuc, K.J. Hsia, Bio-inspired soft robotics: material selection, actuation, and design, *Extrem. Mech. Lett.* 22 (2018) 51–59.
- [4] D.A. Paley, N.M. Wereley, *Bioinspired Sensing, Actuation, and Control in Underwater Soft Robotic Systems*, Springer, 2021.
- [5] M. Bergmann, A. Iollo, R. Mittal, Effect of caudal fin flexibility on the propulsive efficiency of a fish-like swimmer, *Bioinspir. Biomim.* 9 (4) (2014) 046001.
- [6] Q. Zhu, M. Wolfgang, D. Yue, M. Triantafyllou, Three-dimensional flow structures and vorticity control in fish-like swimming, *J. Fluid Mech.* 468 (2002) 1–28.
- [7] J. Nocedal, S. Wright, *Numerical Optimization*, Springer Science & Business Media, 2006.
- [8] S. Kern, P. Koumoutsakos, Simulations of optimized anguilliform swimming, *J. Exp. Biol.* 209 (24) (2006) 4841–4857.
- [9] M. Gazzola, W.M. Van Rees, P. Koumoutsakos, C-start: optimal start of larval fish, *J. Fluid Mech.* 698 (9) (2012) 5–18.
- [10] W.M. Van Rees, M. Gazzola, P. Koumoutsakos, Optimal shapes for anguilliform swimmers at intermediate reynolds numbers, *J. Fluid Mech.* 722 (R3) (2013).
- [11] C. Eloy, On the best design for undulatory swimming, *J. Fluid Mech.* 717 (2013) 48.

- [12] G. Tokić, D.K. Yue, Optimal shape and motion of undulatory swimming organisms, *Proc. - Royal Soc., Biol. Sci.* 279 (1740) (2012) 3065–3074.
- [13] A.P. Maertens, A. Gao, M.S. Triantafyllou, Optimal undulatory swimming for a single fish-like body and for a pair of interacting swimmers, *J. Fluid Mech.* 813 (2017) 301–345.
- [14] T. Back, *Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms*, Oxford University Press, 1996.
- [15] N. Hansen, A. Ostermeier, Completely derandomized self-adaptation in evolution strategies, *Evol. Comput.* 9 (2) (2001) 159–195.
- [16] M. Lighthill, Note on the swimming of slender fish, *J. Fluid Mech.* 9 (2) (1960) 305–317.
- [17] M.J. Lighthill, Large-amplitude elongated-body theory of fish locomotion, *Proc. R. Soc. Lond. B, Biol. Sci.* 179 (1055) (1971) 125–138.
- [18] P.W. Webb, Hydrodynamics and energetics of fish propulsion, in: *Bulletin—Fisheries Research Board of Canada*, 1975.
- [19] M. Saadat, F.E. Fish, A. Domel, V. Di Santo, G. Lauder, H. Haj-Hariri, On the rules for aquatic locomotion, *Phys. Rev. Fluids* 2 (8) (2017) 083102.
- [20] W.W. Schultz, P.W. Webb, Power requirements of swimming: do new methods resolve old questions?, *Integr. Comp. Biol.* 42 (5) (2002) 1018–1025.
- [21] G.S. Triantafyllou, M. Triantafyllou, M. Grosenbaugh, Optimal thrust development in oscillating foils with application to fish propulsion, *J. Fluids Struct.* 7 (2) (1993) 205–224.
- [22] D. Barrett, M. Triantafyllou, D. Yue, M. Grosenbaugh, M. Wolfgang, Drag reduction in fish-like locomotion, *J. Fluid Mech.* 392 (1999) 183–212.
- [23] M. Gazzola, M. Argentina, L. Mahadevan, Scaling macroscopic aquatic locomotion, *Nat. Phys.* 10 (10) (2014) 758–761.
- [24] O. Akanyeti, J. Putney, Y.R. Yanagitsuru, G.V. Lauder, W.J. Stewart, J.C. Liao, Accelerating fishes increase propulsive efficiency by modulating vortex ring geometry, *Proc. Natl. Acad. Sci.* 114 (52) (2017) 828–13 833.
- [25] T.N. Wise, M.A. Schwalbe, E.D. Tytell, Hydrodynamics of linear acceleration in bluegill sunfish, *lepomis macrochirus*, *J. Exp. Biol.* 221 (23) (2018) 1–12.
- [26] A.J. Booker, J.E. Dennis, P.D. Frank, D.B. Serafini, V. Torczon, M.W. Trosset, A rigorous framework for optimization of expensive functions by surrogates, *Struct. Optim.* 17 (1) (1999) 1–13.
- [27] A.L. Marsden, M. Wang, J.E. Dennis, P. Moin, Optimal aeroacoustic shape design using the surrogate management framework, *Optim. Eng.* 5 (2) (2004) 235–262.
- [28] Z.-H. Han, K.-S. Zhang, Surrogate-based optimization, in: O. Roewa (Ed.), *Real-World Applications of Genetic Algorithms*, 2012.
- [29] S.N. Lophaven, H.B. Nielsen, J. Søndergaard, *DACE: a Matlab kriging toolbox*, Citeseer 2 (2002).
- [30] A. Abouhusein, N. Islam, Y. Peet, A constraint handling approach with guaranteed feasibility for surrogate based optimization, arXiv preprint arXiv: 2107.10190, 2021.
- [31] R.G. Regis, C.A. Shoemaker, A stochastic radial basis function method for the global optimization of expensive functions, *INFORMS J. Comput.* 19 (4) (2007) 497–509.
- [32] M.J. Powell, *The BOBYQA algorithm for bound constrained optimization without derivatives*, Cambridge NA Report NA2009/06, University of Cambridge, Cambridge, 2009, pp. 26–46.
- [33] C. Cartis, L. Roberts, O. Sheridan-Methven, Escaping local minima with local derivative-free methods: a numerical investigation, *Optimization* (2021) 1–31.
- [34] J. Lee, J. Kim, H. Choi, K.-S. Yang, Sources of spurious force oscillations from an immersed boundary method for moving-body problems, *J. Comput. Phys.* 230 (7) (2011) 2677–2695.
- [35] M.E. Rosti, L. Brandt, Numerical simulation of turbulent channel flow over a viscous hyper-elastic wall, *J. Fluid Mech.* 830 (2017) 708–735.
- [36] Y. Xu, Y.T. Peet, Verification and convergence study of a spectral-element numerical methodology for fluid-structure interaction, *J. Comput. Phys.* 10 (100) (2021) 100084.
- [37] U.K. Müller, J.L. Van Leeuwen, Undulatory fish swimming: from muscles to flow, *Fish Fish.* 7 (2) (2006) 84–103.
- [38] B.E. Merrill, Y.T. Peet, Moving overlapping grid methodology of spectral accuracy for incompressible flow solutions around rigid bodies in motion, *J. Comput. Phys.* 390 (2019) 121–151.
- [39] S. Patel, P. Fischer, M. Min, A. Tomboulides, A characteristic-based spectral element method for moving-domain problems, *J. Sci. Comput.* 79 (1) (2019) 564–592.
- [40] Y. Xu, Y.T. Peet, Accuracy and performance of fluid-structure interaction algorithms with explicit versus implicit formulations of the fluid solver, *AIAA Paper 2017-3449*, Denver, CO, 2017.
- [41] R. Du, Z. Li, K. Youcef-Toumi, P.V. y Alvarado, *Robot Fish: Bio-Inspired Fishlike Underwater Robots*, Springer, 2015.
- [42] P.P.A. Valdivia y Alvarado, *Design of biomimetic compliant devices for locomotion in liquid environments*, PhD thesis, Massachusetts Institute of Technology, 2007.
- [43] J.M. Donley, K.A. Dickson, Swimming kinematics of juvenile kawakawa tuna (*euthynnus affinis*) and chub mackerel (*scomber japonicus*), *J. Exp. Biol.* 203 (20) (2000) 3103–3116.
- [44] R. Bainbridge, The speed of swimming of fish as related to size and to the frequency and amplitude of the tail beat, *J. Exp. Biol.* 35 (1) (1958) 109–133.
- [45] R.L. McMasters, C.P. Grey, J.M. Sollock, R. Mukherjee, A. Benard, A.R. Diaz, Comparing the mathematical models of Lighthill to the performance of a biomimetic fish, *Bioinspir. Biomim.* 3 (1) (2008) 016002.
- [46] B.W. Lindsey, F.M. Smith, R.P. Croll, From inflation to flotation: contribution of the swimbladder to whole-body density and swimming depth during development of the zebrafish (*danio rerio*), *Zebrafish* 7 (1) (2010) 85–96.
- [47] R.M. Alexander, *Principles of Animal Locomotion*, Princeton University Press, 2013.
- [48] Y. Xu, Y.T. Peet, Optimum gaits of 2D thunniform locomotion for efficient swimming and performance of fish pair, *AIAA Paper 2018-2915*, Atlanta, GA, 2018.
- [49] A. Franci, M. Cremonesi, On the effect of standard PFEM remeshing on volume conservation in free-surface fluid flow problems, *Comput. Part. Mech.* 4 (3) (2017) 331–343.
- [50] A. Khayyer, N. Tsuruta, Y. Shimizu, H. Gotoh, Multi-resolution MPS for incompressible fluid-elastic structure interactions in ocean engineering, *Appl. Ocean Res.* 82 (2019) 397–414.
- [51] I. Borazjani, F. Sotiropoulos, Numerical investigation of the hydrodynamics of carangiform swimming in the transitional and inertial flow regimes, *J. Exp. Biol.* 211 (10) (2008) 1541–1558.
- [52] A.A. Shirgaonkar, M.A. MacIver, N.A. Patankar, A new mathematical formulation and fast algorithm for fully resolved simulation of self-propulsion, *J. Comput. Phys.* 228 (7) (2009) 2366–2390.
- [53] P. Fischer, J. Lottes, S. Kerkemeier, O. Marin, K. Heisey, A. Obabko, E. Merzari, Y. Peet, *Nek5000 User's Manual*, Technical Report ANL/MCS-TM-351, Argonne National Laboratory, 2015, <http://nek5000.mcs.anl.gov>.
- [54] P.F. Fischer, An overlapping Schwarz method for spectral element solution of the incompressible Navier–Stokes equations, *J. Comput. Phys.* 133 (1) (1997) 84–101.
- [55] J.D. Anderson Jr, *Fundamentals of Aerodynamics*, Tata McGraw-Hill Education, 2010.
- [56] I. Borazjani, F. Sotiropoulos, On the role of form and kinematics on the hydrodynamics of self-propelled body/caudal fin swimming, *J. Exp. Biol.* 213 (1) (2010) 89–107.
- [57] H. Yu, X.-Y. Lu, H. Huang, Collective locomotion of two uncoordinated undulatory self-propelled foils, *Phys. Fluids* 33 (1) (2021) 011904.
- [58] L.W. Ho, A Legendre spectral element method for simulation of incompressible unsteady viscous free-surface flows, PhD thesis, Massachusetts Institute of Technology, 1989.

- [59] M.O. Deville, P.F. Fischer, E. Mund, *High-Order Methods for Incompressible Fluid Flow*, Cambridge University Press, 2002.
- [60] Y. Maday, A.T. Patera, Spectral element methods for the incompressible Navier-Stokes equations, in: *State-of-the-Art Surveys on Computational Mechanics* (A90-47176 21-64), American Society of Mechanical Engineers, New York, 1989, pp. 71–143.
- [61] C. Bernardi, Y. Maday, A collocation method over staggered grids for the Stokes problem, *Int. J. Numer. Methods Fluids* 8 (1988) 537–557.
- [62] A. Quarteroni, F. Saleri, A. Veneziani, Factorization methods for the numerical approximation of Navier–Stokes equations, *Comput. Methods Appl. Mech. Eng.* 188 (1–3) (2000) 505–526.
- [63] A.C. Aitken, On Bernoulli's numerical solution of algebraic equations, *Proc. R. Soc. Edinb.* 46 (1927) 289–305.
- [64] U. Küttler, W.A. Wall, Fixed-point fluid–structure interaction solvers with dynamic relaxation, *Comput. Mech.* 43 (1) (2008) 61–72.
- [65] P. Causin, J.-F. Gerbeau, F. Nobile, Added-mass effect in the design of partitioned algorithms for fluid–structure problems, *Comput. Methods Appl. Mech. Eng.* 194 (42–44) (2005) 4506–4527.
- [66] C. Förster, W.A. Wall, E. Ramm, Artificial added mass instabilities in sequential staggered coupling of nonlinear structures and incompressible viscous flows, *Comput. Methods Appl. Mech. Eng.* 196 (7) (2007) 1278–1293.
- [67] E.D. Tytell, G.V. Lauder, The hydrodynamics of eel swimming: I. Wake structure, *J. Exp. Biol.* 207 (11) (2004) 1825–1841.
- [68] M.D. McKay, R.J. Beckman, W.J. Conover, Comparison of three methods for selecting values of input variables in the analysis of output from a computer code, *Technometrics* 21 (2) (1979) 239–245.
- [69] A.A. Eldeiry, L.A. Garcia, Comparison of ordinary kriging, regression kriging, and cokriging techniques to estimate soil salinity using LANDSAT images, *J. Irrig. Drain. Eng.* 136 (6) (2010) 355–364.
- [70] J. Mueller, MATSuMoTo: the MATLAB surrogate model toolbox for computationally expensive black-box global optimization problems, arXiv preprint arXiv:1404.4261, 2014.
- [71] A. Abouhoussein, N. Islam, Y.T. Peet, Constraint enforcement to guarantee strictly feasible solutions in a surrogate based optimizer, AIAA Paper, San Diego, CA, 2022.
- [72] N. Offermans, O. Marin, M. Schanen, J. Gong, P. Fischer, P. Schlatter, A. Obabko, A. Peplinski, M. Hutchinson, E. Merzari, On the strong scaling of the spectral element solver Nek5000 on petascale systems, in: *Proceedings of the Exascale Applications and Software Conference 2016*, 2016, pp. 1–10.
- [73] B.M. Adams, W.J. Bohnhoff, K. Dalbey, J. Eddy, M. Eldred, D. Gay, K. Haskell, P.D. Hough, L.P. Swiler, Dakota, a multilevel parallel object-oriented framework for design optimization, parameter estimation, uncertainty quantification, and sensitivity analysis: Version 5.0 User's Manual, Tech. Rep. SAND2010-2183, Sandia National Laboratories, 2009.
- [74] F. Gibouin, C. Raufaste, Y. Bouret, M. Argentina, Study of the thrust–drag balance with a swimming robotic fish, *Phys. Fluids* 30 (9) (2018) 901.
- [75] T. Theodorsen, General theory of aerodynamic instability and the mechanism of flutter, NASA Technical Report No. 46, 1935.
- [76] I. Garrick, Propulsion of a flapping and oscillating airfoil, NACA Report No. 567, 1937, pp. 419–427.
- [77] K.W. Moored, D.B. Quinn, Inviscid scaling laws of a self-propelled pitching airfoil, *AIAA J.* 57 (9) (2019) 3686–3700.
- [78] M. Lighthill, Hydromechanics of aquatic animal propulsion, *Annu. Rev. Fluid Mech.* 1 (1) (1969) 413–446.
- [79] G.V. Lauder, E.D. Tytell, Hydrodynamics of undulatory propulsion, *Fish Physiol.* 23 (2005) 425–468.
- [80] F.E. Fish, N. Rybczynski, G.V. Lauder, C.M. Duff, The role of the tail or lack thereof in the evolution of tetrapod aquatic propulsion, *Integr. Comp. Biol.* (2021).
- [81] T. Van Buren, D. Floryan, A.J. Smits, Bioinspired underwater propulsors, in: L. Daniel, W. Soboyejo (Eds.), *Bioinspired Structures and Design*, Cambridge University Press, Cambridge, 2018.
- [82] G.E. Karniadakis, S. Sherwin, *Spectral/Hp Element Methods for Computational Fluid Dynamics*, Oxford University Press, 2005.