



HAL
open science

Simulation of forced deformable bodies interacting with two-dimensional incompressible flows: Application to fish-like swimming

Seyed Amin Ghaffari, Stéphane Viazzo, Kai Schneider, Patrick Bontoux

► To cite this version:

Seyed Amin Ghaffari, Stéphane Viazzo, Kai Schneider, Patrick Bontoux. Simulation of forced deformable bodies interacting with two-dimensional incompressible flows: Application to fish-like swimming. *International Journal of Heat and Fluid Flow*, 2015, Theme special issue celebrating the 75th birthdays of Brian Launder and Kemo Hanjalic, 51, pp.88-109. 10.1016/j.ijheatfluidflow.2014.10.023 . hal-00967077v2

HAL Id: hal-00967077

<https://hal.science/hal-00967077v2>

Submitted on 5 May 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Simulation of forced deformable bodies interacting with two-dimensional incompressible flows: Application to fish-like swimming

Seyed Amin Ghaffari^a, Stéphane Viazzo^a, Kai Schneider^a, Patrick Bontoux^a

^aLaboratoire M2P2-UMR 7340-CNRS, Aix-Marseille Université, Centrale Marseille, France
Corresponding author: bontoux@L3M.univ-mrs.fr

Abstract

We present an efficient algorithm for simulation of deformable bodies interacting with two-dimensional incompressible flows. The temporal and spatial discretizations of the Navier-Stokes equations in vorticity stream-function formulation are based on classical fourth-order Runge-Kutta and compact finite differences, respectively. Using a uniform Cartesian grid we benefit from the advantage of a new fourth-order direct solver for the Poisson equation to ensure the incompressibility constraint down to machine zero. For introducing a deformable body in fluid flow, the volume penalization method is used. A Lagrangian structured grid with prescribed motion covers the deformable body interacting with the surrounding fluid due to the hydrodynamic forces and moment calculated on the Eulerian reference grid. An efficient law for curvature control of an anguilliform fish, swimming to a prescribed goal, is proposed. Validation of the developed method shows the efficiency and expected accuracy of the algorithm for fish-like swimming and also for a variety of fluid/solid interaction problems.

Keywords: Fluid interaction with forced deformable bodies, Compact fourth-order direct Poisson solver, Volume penalization, Fish swimming/turning

1. Introduction

The quantification and simulation of the flow around biological swimmers is one of the challenges in fluid mechanics [40]. At the same time bio-inspired design of swimming robots are in growth [25]. The costs of experimental studies [38] lead the researchers to develop for efficient predictive numerical algorithms for the hydrodynamic analyses of fish swimming. Difficulties of numerical simulations of fish-like swimming are due to different reasons. One problem is efficient quantification of the kinematics of different species which seems to be far from the proposed simple laws in different studies. However, the main swimming mechanism in the majority of anguilliform fishes consists of a sinusoidal wave enveloped by a profile, which is created by the backbone of the fish and moves from head to tail. The tail beat creates a reversed Kármán street of vortices and generates thrust, leaving thus a momentumless wake back. Efficient simulation of incompressible flows is also an important problem, where the efficiency of the elliptic solver is crucial. The third bottleneck in numerical simulations of fish-like swimming is the coupling of the fluid solver with deformable, moving and rotating bodies. Anguilliform fishes add a constant curvature to their backbone for turning, i.e., they use their body like a rudder for torque generation. Yeo et al. [31] studied numerically the straight swimming/cruising and sharp turning manoeuvres in two-dimensions. A carangiform-like swimmer was shown to execute a sharp turn through an angle

of 70° from straight coasting within a space of about one body length. Bergman and Iollo [34] performed numerical simulations of fish swimming/rotation, looking for food at a prescribed point. They added a radius to the backbone of the fish to perform a rotation. The considered fish is constructed by a complex valued mapping like the Kutta-Joukowski transform. Here we will present a simple law for turning of an anguilliform fish. Our rotation control law [41] is similar to that presented in [31] and [34], in which the feedback is based on the angle between the line-of-sight and the direction of advance. But instead of adding a radius to the backbone we envisage to work with curvature which seems to be more efficient. We are using the method proposed by Boyer et al. [21] which is based on quaternions for efficient description of the fishes backbone kinematics. We are going to apply the rotation control to two-dimensional swimming. Even if due to the shape and deformation style of the fish-like swimmers the surrounding flow is fully three dimensional, most of the fundamental features of swimming are included in two-dimensional analyses. For two dimensional incompressible flows the Navier-Stokes equations can be reformulated in terms of vorticity and stream-function which is more efficient. The choice of a finite difference method (FDM) in this paper is related to the use of an immersed boundary method in which a Cartesian grid can be used. Hence the use of FDM is straightforward. Among finite difference methods high order compact methods [5, 10] are more advantageous in terms of accuracy and reasonable cost. We

refer to [19] and [35] for high-order compact discretizations of the incompressible Navier-Stokes equations in primitive variables and to [6]-[7] and [12] for the vorticity and stream-function formulation. Solving the incompressible Navier-Stokes equations typically implies an elliptic Poisson equation which is the most time consuming part of the algorithm. Direct methods like diagonalization or iterative methods (e.g. red-black PSOR, multi-grid and Krylov subspace methods) can be used. Using high-order discretizations iterative methods are less attractive because the resulted matrices are less sparse, thus the rate of convergences are slow. However iterative methods can cover all types of boundary conditions, see Spitz and Carey [12] for a fourth-order compact discretization of Poisson equation. On the other hand in direct methods the memory limitation is restrictive for simulations on fine grids. Therefore decoupling of the directions by FFT based methods can be advantageous, however this method implies some limitations in the boundary conditions. We are presenting a new fourth-order solver for the Poisson equation which is a combination of a compact finite difference with a sine FFT. The main advantages of our method are fourth-order accuracy, efficiency, the possibility to parallelize and convergence down to zero machine precision. Other advantages and limitations of the proposed solver are discussed in the paper. A difficulty in numerical simulations of fish swimming is the analysis of fluid/solid interaction, which can be handled by strong or loose coupling according to implicit or explicit time advancement, cf. [40] for a detail discussion. We are using the volume penalization method which was proposed by Angot et al. [14] and belongs to the so called immersed boundary methods (IBMs). It consists of modeling the immersed body as a porous media, thus getting ride of the Dirichlet boundary conditions by considering both the fluid and the body as one domain with different permeabilities. So one can consider a rectangular solution domain in which the body is immersed and can even move. The penalization method will lead to first-order accuracy near the body but it is one of the most efficient methods in dealing with deformable, moving and rotating bodies in fluids. An extension to include elasticity of the solid in one direction is done in [39]. We refer to the review of Mittal and Iaccarino [20] for a complete classification and description of immersed boundary methods. In the present work we will focus on some numerical aspects of efficient turning laws a topic which is less studied so far. To this end the method of quaternions is adapted to backbone kinematics description. Starting by the code developed in [29] we are applying compact finite differences to the vorticity stream-function formulation of the Navier-Stokes equations including the penalization term. An efficient direct method is presented for solving the Poisson equation. Thus different numerical aspects of algorithm like accuracy in space and the error introduced by the penalization method will be examined. The code is developed in FORTRAN and is accessible for all [42]. The paper is organized as follows. First our methodology in-

cluding the governing equations, discretization, kinematics of a fish like-swimming and the algorithm for fluid interaction with forced deformable bodies will be presented. Then a validation of the algorithm will be done and errors convergence will be studied. Results for swimming and rotation control will be also reported. Finally, the results will be discussed and some guides for the future works will be addressed.

2. Methodology

2.1. Governing equations of incompressible flow

The governing equations of incompressible flows are the Navier-Stokes equations. In two-dimensional problems the vorticity and stream-function formulation has the advantage that it not only eliminates the pressure variable entirely, but also ensures a divergence-free velocity field (mass conservation, i.e., $\nabla \cdot \mathbf{u} = 0$), if the Poisson equation (2) properly satisfied, see [6] and [7]. One encounters two scalar valued quantities, i.e., the vorticity ω and the stream-function ψ , instead of the velocity vector and the pressure field, thus it makes the computations more efficient. With this formulation it is possible to use a collocated grid without adding any explicit numerical dissipation which reduces the arithmetics considerably. Therefore we continue with this formulation, but the concepts can also be extended to primitive variable formulation. By taking the curl of the Navier-Stokes equations, one obtains the vorticity transport equation:

$$\partial_t \omega + (\mathbf{u} \cdot \nabla) \omega = \nu \nabla^2 \omega + \nabla \times \mathbf{F}, \quad \mathbf{x} \in \Omega \in \mathbb{R}^2 \quad (1)$$

where $\omega(\mathbf{x}, t) = \nabla \times \mathbf{u} = v_x - u_y$ denotes the vorticity component which is normal to the considered two-dimensional plane, Ω is the spatial domain of interest, given as an open subset of \mathbb{R}^2 , which can be bounded or unbounded in general, $\mathbf{u}(\mathbf{x}, t)$ is the velocity field, $\nu = \mu/\rho_f > 0$ is the kinematic viscosity of the fluid, ρ_f is the density and $\mathbf{F}(\mathbf{x}, t)$ is a source term. For a complete description of a particular problem, the above equations need to be complemented to describe an initial/boundary value problem (IBVP). The equation is parabolic in time and the velocity components are $(u, v) = (\partial_y \psi, -\partial_x \psi)$, with ψ satisfying a Poisson equation

$$-\nabla^2 \psi = \omega \quad (2)$$

which is an elliptic equation in space. The penalization term for unit mass of the fluid reads,

$$\mathbf{F} = -\eta^{-1} \chi(\mathbf{u} - \mathbf{u}_P) \quad (3)$$

where $\mathbf{u}_P(\mathbf{x}, t)$ is the velocity field of the immersed body. The Navier-Stokes equations are written for unit mass of the fluid, therefore the dimension of the source term \mathbf{F} is acceleration, i.e., $[LT^{-2}]$. The penalization parameter η is the permeability (porosity) coefficient of the immersed

body with dimension $[T]$. The mask (characteristic) function χ is dimensionless and describes the geometry of the immersed body

$$\chi(\mathbf{x}, t) = \begin{cases} 1 & \mathbf{x} \in \Omega_b \\ 0 & \mathbf{x} \in \Omega_f \end{cases} \quad (4)$$

where Ω_f represents the domain of the flow and Ω_b represents the immersed body in the domain of the solution. The solution domain $\Omega = \Omega_f \cup \Omega_b$ is governed by the Navier-Stokes equations in the fluid regions and by Darcy's law in the penalized regions, when $\eta \rightarrow 0$.

2.1.1. Spatial discretization

In the present investigation an explicit second-order and an implicit fourth-order compact finite difference methods [5] is used for discretization of the spatial derivatives. The advantage of the compact methods over explicit finite differences is illustrated in terms of the scaled modified wavenumber $w = k\Delta x$ in Fig. 1. For a given periodic function $f(x) = e^{ikx}$, $x \in [0, 2\pi]$ with known analytical derivatives $f'(x) = ik e^{ikx}$, $f''(x) = -k^2 e^{ikx}$, a numerical approximation of the derivatives at point x_j have the form $f'(x_j) = ik' e^{ikx_j}$ and $f''(x_j) = -k''^2 e^{ikx_j}$. The difference between exact and numerical approximation of the wavenumber is a measure of the discretization error which is purely dispersive for the first derivative and dissipative for the second derivative, if the considered function and discretization is periodic. For the second derivative the scaled modified wavenumbers are compared in Fig. 1 (b) with analytical values given by Lele [10] for some explicit and implicit differentiation methods. A good agreement between the numerical approximations of the scaled modified wavenumbers and the analytical values can be observed. It must be noted that the error in terms of the modified wavenumber is not necessarily sensitive to the formal order of the truncation error obtained by Taylor expansion analysis. The desired characteristics of finite difference schemes are better studied by directly optimizing the scheme in Fourier space rather than looking for the lowest truncation error. For example spectral like five-diagonal finite difference schemes designed by Lele [10] or Kim [23] are formally fourth order, see Fig. 1 (a). Given the values of a function f on a uniformly spaced mesh $x_i = (i-1)h$, for $(i = 1, \dots, N)$ with $h = L_x/(N-1)$. Following [5, 10] a fourth-order approximation of the first and second derivatives are obtained by the classical Padé schemes:

$$f'_{i-1} + 4f'_i + f'_{i+1} = 3(f_{i+1} - f_{i-1})/h \quad (5)$$

$$f''_{i-1} + 10f''_i + f''_{i+1} = 12(f_{i+1} - 2f_i + f_{i-1})/h^2 \quad (6)$$

for $i = 2, \dots, N-1$, near the boundaries a third-order forward/backward stencil can be used, see [10]. A direct solver (LU-decomposition) will be applied to the tri/pentadiagonal system of linear equations along each line. The computational cost of a tridiagonal implicit method is in general three times of the one explicit method and a pentadiagonal linear system twice that of the tridiagonal.

2.1.2. Time integration

A classical fourth-order Runge-Kutta (RK4) method is used for time integration of the penalized vorticity transport equation (1), by assembling all spatial derivatives in the operator $k(\omega)$, one has

$$\omega^{n+1} = \omega^n + \frac{\Delta t}{6} (k_1(\omega) + 2k_2(\omega^*) + 2k_3(\omega^*) + k_4(\omega^*)) \quad (7)$$

where

$$k_i(\omega) = -\partial_y \psi_i \partial_x \omega + \partial_x \psi_i \partial_y \omega + \nu \nabla^2 \omega + \partial_x F_y - \partial_y F_x \quad (8)$$

for $i = 1, 2, 3, 4$. In each time step Eq. (8) must be evaluated four times where Eq. (2) must be solved for updating of stream-function ($-\nabla^2 \psi_i = \omega_i^*$). For details and technical discussions of Runge-Kutta methods see [11]. However, Δt is limited by the *CFL* (Courant-Friedrichs-Lewy) condition which implies that

$$\frac{U \Delta t}{\Delta x} \leq CFL \approx \frac{\sigma_i}{w'_{\max}} \quad (9)$$

where U is an advection velocity (or a phase speed). In the presence of the nonlinearity in space more attention must be paid. Viscous terms imply a constraint of the form

$$\frac{\nu \Delta t}{\Delta x^2} \leq VSL \approx \frac{\sigma_r}{w''_{\max}} \quad (10)$$

on the time-step, where $\sigma_r = 2.9$ and $\sigma_i = 2.85$ are limits of real and imaginary parts of the stable region on complex plan for RK4 method. $w'_{\max} = 1.74$ and $w''_{\max} = 6$ are the maximum values of the scaled modified wave numbers for the first and second derivatives calculated via fourth-order Padé scheme, plotted in Fig. 1. It can be seen that $w'_{\max} \in [1, \pi]$ and $w''_{\max} \in [4, \pi^2]$ for different approximations of spatial derivatives. Therefore with the use of a high-order method for spatial discretization smaller time-steps must be used. In the presence of moving bodies the displacement of the moving body must not exceed the grid spacing, i.e., $\Delta t \leq \Delta \mathbf{x} / \mathbf{u}_B$. Moreover, by using the explicit penalization method another constraint, $\Delta t \leq \eta$, must be respected. Among the four above-mentioned constraints, the smallest Δt must be chosen. According to Gazzola et al. [33] with the use of volume penalization method for simulation of fluid/solid interaction the order of accuracy in time is reported to be first-order independent of the time accuracy of the underlying method for time integration of the Navier-Stokes equations. However the larger stability bound of the RK4 is still attractive to enhance the time step.

2.1.3. Fourth-order fast Poisson solver

In solving the incompressible Navier-Stokes equations an elliptic Poisson equation is frequently encountered which is the most time consuming part of the algorithm. The common case is the pressure Poisson equation normally used with homogeneous Neumann boundary conditions,

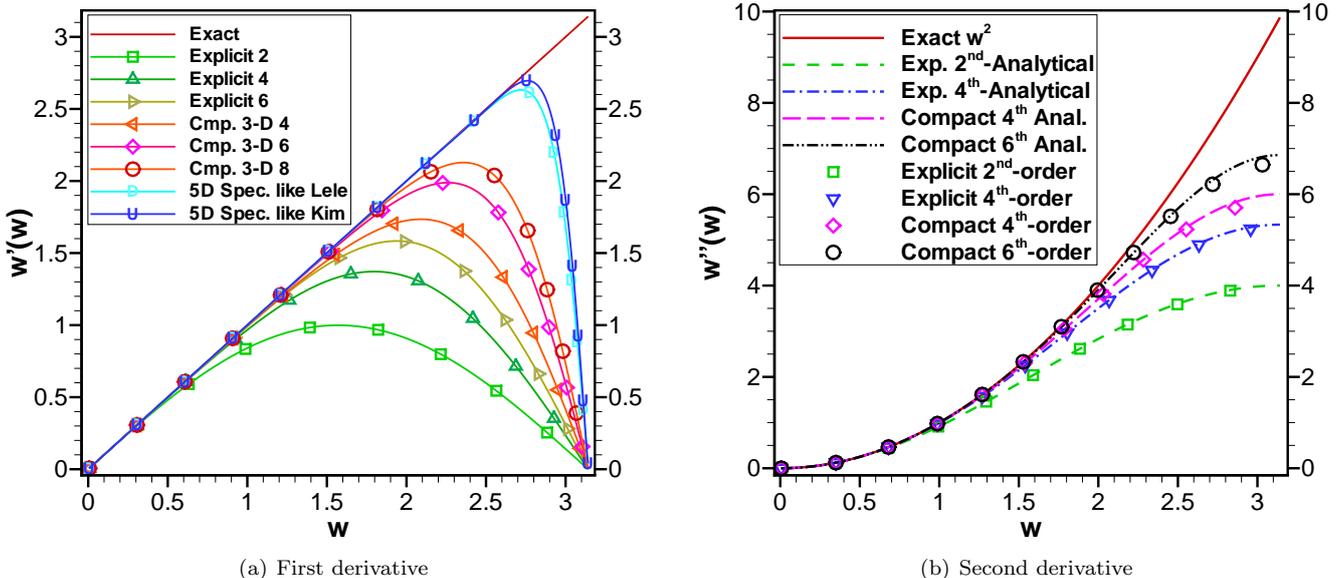


Figure 1: Plots of the scaled modified wavenumber for the first and second derivative, $w'(w)$ and $w''(w)$ respectively, versus the scaled wavenumber $w = k\Delta x$ for different central finite difference methods presented in Lele [10] and Kim [23].

for the pressure correction in projection methods. In the vorticity stream-function formulation equation (2) has to be solved with Dirichlet boundary condition for vorticity and stream-function. Free slip ($\omega = 0$) boundary conditions in a close rectangular domain ($\psi = 0$, all around) is applied in all the test cases studied in the present investigation. In the presence of periodic boundary conditions, FFT based direct solvers can be used to efficiently solve the Poisson equation with high accuracy. Even if the flow is not periodic in all directions, like most of the practical problems, in accordance with the boundary conditions for the elliptic equation (homogeneous Dirichlet/Neumann) *sine* or *cosine* FFTs can be used, see [9]-[15] and [30]. We are presenting a new direct fourth-order solver for the Poisson equation (2) which is a combination of a compact finite difference with a sine FFT. The advantages of our method are fourth-order accuracy, convergence down to zero machine precision, compact tridiagonal stencil, possibility of extension to three dimensions, reduced arithmetics and memory usage in comparison to iterative methods. Moreover the parallelization is straightforward because of decoupling of the operations in different directions. Nearly linear strong scaling (speed up) and efficiency is reported by Laizet and Lamballais [30] for a similar direct solver. They introduced a dual domain decomposition (or pencil) method, in which information along a line is accessible for a CPU by alternative decomposition of the domain in three directions. The limitation of our method (moreover the boundary conditions) is the use of a uniform grid in the direction in which the FFT is applied. When the solver of the parabolic part consists of finite-differences, it is a custom to use a FDM discretization in one direction with-

out loss of accuracy and efficiency via a direct tridiagonal solver. The advantage of this approach is the possibility of applying general boundary conditions in that direction and using a refined mesh. The second-order version of this solver can be found in [15]. For a compact fourth-order collocated discretization of the Poisson equation $-\nabla^2\psi = \omega$, over $N_x \times N_y$ grid points, by using

$$\frac{\partial^2\psi}{\partial x^2} = \delta_x^2\psi - \frac{\Delta x^2}{12}\frac{\partial^4\psi}{\partial x^4} + O(\Delta x^4) \quad (11)$$

where δ_x^2 represents a central second-order estimation of the second derivative, for the x -direction we obtain

$$\left(\delta_x^2 - \frac{\Delta x^2}{12}\frac{\partial^4}{\partial x^4} + \partial_{yy}\right)\psi = -\omega \quad (12)$$

because of the presence of the Δx^2 factor behind the fourth-order derivative, this term cannot be dropped and must be evaluated with second-order accuracy. Therefore, the hole approximation scheme yields fourth-order accuracy. The fourth-order derivative can be evaluated by using the original Poisson equation $-\nabla^2\psi = \omega$, and successive differentiation with respect to x (i.e., $\partial_{xx}\partial_{xx}\psi = -\partial_{xx}\partial_{yy}\psi - \partial_{xx}\omega$). Replacing ∂_{xx} by δ_x^2 , we find

$$\left(\delta_x^2 + \frac{\Delta x^2}{12}\delta_x^2\partial_{yy} + \partial_{yy}\right)\psi = -\omega - \frac{\Delta x^2}{12}\delta_x^2\omega \quad (13)$$

By applying Fourier transform in y direction over Eq. (13) and replacing second derivatives by $-k_y^2\hat{\psi}$ in Fourier space, we have

$$\left(\delta_x^2 - \frac{\Delta x^2}{12}\delta_x^2k_y'^2 - k_y'^2\right)\hat{\psi} = -\hat{\omega} - \frac{\Delta x^2}{12}\delta_x^2\hat{\omega} \quad (14)$$

Usually the exact wavenumber is replaced by the modified wavenumber $k_y'^2$ which permits to evaluate the difference between the finite-difference and the spectral approximation of the second derivative [15]. For a fourth-order explicit finite-difference discretization, analytical relation for the scaled modified wavenumber of second derivative is given by Lele [10] as follows

$$k_y'^2 = \frac{1}{\Delta y^2} \left[\frac{8}{3} \left(1 - \cos\left(\frac{k_y \pi}{N_y}\right) \right) - \frac{1}{6} \left(1 - \cos\left(\frac{2k_y \pi}{N_y}\right) \right) \right] \quad (15)$$

Comparison with numerical values in Fig. 1 (b) confirm that Eq. (15) is exact. The final tridiagonal system to be solved in Fourier space for each wavenumber of ψ in the y -direction is

$$\beta \hat{\psi}_{i+1,m} - (2\beta + k_y'^2) \hat{\psi}_{i,m} + \beta \hat{\psi}_{i-1,m} = -(\hat{\omega}_{i+1,m} + 10\hat{\omega}_{i,m} + \hat{\omega}_{i-1,m})/12 \quad (16)$$

for $i = 2, \dots, N_x - 1$, where $\beta = \Delta x^{-2} - k_y'^2/12$. In summary, first a one-dimensional direct-FFT of the forcing function is performed in the y -direction. Then for each line in the x -direction the tri-diagonal system (16) must be solved to find the solution ψ in wavenumber space. Next an inverse-FFT of the solution is performed. For the real data with zero value at the boundaries (homogeneous Dirichlet boundary condition, i.e., $\psi = \omega = 0$), the natural Fourier transform to use is the *sine* transform, see e.g. [11]. The direction of FDM and FFT can be changed to consider no-slip boundary condition in the y -direction. In order to take into account inflow/outflow boundary conditions the mean flow must be reduced from the total velocity field $\mathbf{u} = \mathbf{U} - \mathbf{U}_\infty$ in the vorticity transport equation (1) to impose $\psi = 0$ at the boundaries. This is equivalent to move the grid with \mathbf{U}_∞ and writing the Navier-Stokes equations in a moving reference frame for the perturbed velocity field \mathbf{u} instead of a Galilean inertial frame [32].

2.2. Algorithm of the fluid interaction with a deformable body

2.2.1. Kinematics of the fish

The geometrically exact theory of nonlinear beams, is developed by Simo [8] and extended for fish vertebral by Boyer et al. [21]. In this theory, the beam is considered as a continuous assembly of rigid sections of infinitesimal thickness, i.e., a one-dimensional Cosserat medium. We are summarizing the exact kinematics of the fish backbone in three dimensions for interested readers and future developments, but all the cases in this paper are limited to two-dimensions. Following [21], [25] and [38] the kinematics of the backbone for Eel-like fishes can be determined by integration along the arclength $\xi \in [0, l_{\text{fish}}]$ starting with the head's orientation, position and velocities as boundary conditions. The variation of the orientation is obtained by

$$\frac{\partial Q}{\partial \xi} = \frac{1}{2} M(\Omega) Q \quad (17)$$

where $Q = (\cos \frac{\phi}{2}, a_x \sin \frac{\phi}{2}, a_y \sin \frac{\phi}{2}, a_z \sin \frac{\phi}{2})^T$ are unit normalized $(q_0^2 + q_1^2 + q_2^2 + q_3^2)^{1/2} = 1$ quaternions that represent the head frame's orientation with respect to the inertial frame and $M(\Omega)$ is an anti-symmetric tensor

$$M(\Omega) = \begin{bmatrix} 0 & -\omega_1 & -\omega_2 & -\omega_3 \\ \omega_1 & 0 & \omega_3 & -\omega_2 \\ \omega_2 & -\omega_3 & 0 & \omega_1 \\ \omega_3 & \omega_2 & -\omega_1 & 0 \end{bmatrix} \quad (18)$$

where $\Omega = (\omega_1, \omega_2, \omega_3)^T$ denotes the mean angular velocity. The geometry $R = (x, y, z)^T$ in the Galilean reference frame is given by

$$\frac{\partial R}{\partial \xi} = \text{Rot}(Q) K \quad (19)$$

where k_2 and k_3 in $K = (k_1, k_2, k_3)^T$ stand for the Eel's backbone transversal curvature and k_1 represents the rate of rotation (twist) of the section around the backbone with the normal aligned with ξ -direction. The rotation matrix in terms of the quaternions is then given by

$$\text{Rot} = 2 \begin{bmatrix} q_0^2 + q_1^2 - \frac{1}{2} & q_1 q_2 - q_0 q_3 & q_1 q_3 + q_0 q_2 \\ q_1 q_2 + q_0 q_3 & q_0^2 + q_2^2 - \frac{1}{2} & q_2 q_3 - q_0 q_1 \\ q_1 q_3 - q_0 q_2 & q_2 q_3 + q_0 q_1 & q_0^2 + q_3^2 - \frac{1}{2} \end{bmatrix} \quad (20)$$

The variation of linear $V = (v_1, v_2, v_3)^T$ and angular $\Omega = (\omega_1, \omega_2, \omega_3)^T$ velocities in the local frame, i.e., the frame attached to the body are given by

$$\frac{\partial}{\partial \xi} \begin{bmatrix} V \\ \Omega \end{bmatrix} = - \begin{bmatrix} K^\vee & \Gamma^\vee \\ 0 & K^\vee \end{bmatrix} \begin{bmatrix} V \\ \Omega \end{bmatrix} + \begin{bmatrix} \dot{\Gamma} \\ \dot{K} \end{bmatrix} \quad (21)$$

where $(\dot{\cdot})$ represents the time derivative, (\vee) stands for the anti-symmetric matrix constructed from a given vector, e.g.,

$$\Gamma^\vee = \begin{bmatrix} 0 & -\gamma_3 & \gamma_2 \\ \gamma_3 & 0 & -\gamma_1 \\ -\gamma_2 & \gamma_1 & 0 \end{bmatrix} \quad (22)$$

where $\Gamma = (\gamma_1, \gamma_2, \gamma_3)^T$ represents the local transversal shearing of the sections whose first component is the stretching rate along the ξ -direction. The accelerations can also be deduced from the time derivative of Eq. (21). For more details we refer to [21], [25] and [38]. To find the velocities in the frame attached to the head from the velocities V_G in the Galilean reference frame and inverse,

$$(v_1, v_2, v_3)^T = \text{Rot}^T (v_x, v_y, v_z)^T \quad (23)$$

can be used. By considering N ($1, \dots, N_{\text{points}}$) discrete points on the Eel's backbone, equations (17), (19) and (21) altogether must be integrated in space by a proper numerical method ($N_{\text{eq}} = 13$ in 3D). We are using a fourth-order Runge-Kutta method for integration and comparisons with a first-order Euler method shows that RK4 is more precise especially when the number of the points along the Eel's backbone is less than $N_{\text{points}} = 30$.

2.2.2. Lagrangian structured grid

The first choice to start the parameterization of the swimmer's body is a symmetric shape. The geometry of a two-dimensional swimmer can be characterized by the half width $w(\xi)$ of the body along its arclength (midline) $\xi \in [0, l_{\text{fish}}]$. Following the work of Kern and Koumoutsakos [22] and Carling et al. [13], the half width $w(\xi)$ is defined as

$$w(\xi) = \begin{cases} \sqrt{2w_h\xi - \xi^2} & 0 \leq \xi < s_b \\ w_h - (w_h - w_t)\left(\frac{\xi - s_b}{s_t - s_b}\right)^2 & s_b \leq \xi < s_t \\ w_t \frac{l_{\text{fish}} - \xi}{l_{\text{fish}} - s_t} & s_t \leq \xi \leq l_{\text{fish}} \end{cases} \quad (24)$$

where l_{fish} is the body length, $w_h = s_b = 0.04l_{\text{fish}}$, $s_t = 0.95l_{\text{fish}}$ and $w_t = 0.01l_{\text{fish}}$. In the mid part of the fish a linear function can be used as in Gazzola et al. [33]. A structured grid formed by normal to backbone lines with thickness given by (24) covers the body. The velocity components of each point on the Lagrangian grid V_{shape} with (I, J) indexes are given by

$$V_{\text{shape}}(I, J) = V_{\text{BN}}(I) + \mathbf{r}(I, J)\Omega_{\text{BN}}(I) \quad (25)$$

where V_{BN} and Ω_{BN} are the linear and angular velocities of the backbone respectively, given by Eq. (21). The radius ($-w < |\mathbf{r}| < w$) is aligned with the transversal lines of the structured grid normal to the backbone. Fig. 3 shows an example of the Lagrangian grid covering the fish after deformation in which the corresponding velocities of each point are illustrated. The information of the Lagrangian structured grid covering the deformable body must be transferred to the Eulerian-Cartesian grid by interpolation to find $\chi(i, j)$ and $\mathbf{u}_p(i, j)$. To determine $\chi(i, j)$ on the Eulerian grid whose first point $(x, y)_{(1,1)} = (0, 0)$ is located at the origin, the coordinates of each point on the Lagrangian grid $\mathbf{X}_{\text{shape}}(I, J)$ is divided by Δx and Δy . After applying a correction to the integer part of the results they give the indexes (i, j) of the mask function χ on the Eulerian grid for which $\chi = 1$ is assigned. Following [16] and [27] the mask is mollified by the Shuman [2] filter

$$\bar{\chi}_{i,j} = (2\chi_{i,j} + \chi_{i+1,j} + \chi_{i-1,j} + \chi_{i,j+1} + \chi_{i,j-1})/6 \quad (26)$$

which is equivalent to raised *cosine* filter in Fourier space, we refer to [26] for more details. An example of the transferred geometry to the Eulerian grid, after smoothing by Eq. (26) is illustrated in Fig. 4 which represents the mask function. The spacing of the grid points on the Lagrangian grid must be fine enough in comparison to ΔX and ΔY to accurately represent the deformation of the body. We are using two-dimensional linear interpolation

$$f(x, y) = axy + bx + cy + d \quad (27)$$

which leads to a 4×4 linear system. To determine the coefficients the system is solved by a direct method, i.e., Gauss-Jordan elimination from [11]. For all points in the interior of the fish we have $\chi(i, j) = 1$ on the Eulerian grid.

For each point of the Eulerian grid in which $\chi = 1$ the four nearest points of the Lagrangian grid are used to solve the system formed by (27). For some points due to mollifying by Eq. (26) we have $0 < \chi < 1$, therefore the interpolation automatically becomes an extrapolation. Some points are completely outside of the original Lagrangian shape. An example of the interpolated velocity components on the Eulerian grid is illustrated in Fig. 4. Note that the resulting velocity field \mathbf{u}_p is not divergence-free, we refer to Gazzola et al. [33] for a complete theoretical and numerical discussion about this subject. In the present investigation we are not considering this issue.

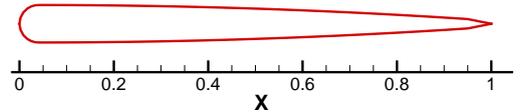


Figure 2: Shape of the fish given by Eq. (24) before deformation.

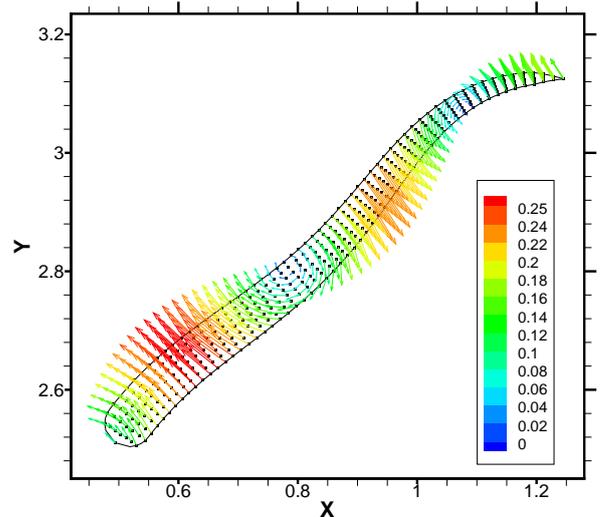


Figure 3: Lagrangian structured grid covering the fish after deformation and the corresponding velocity vectors of each point, colored by absolute velocity $\sqrt{u^2 + v^2}$.

2.2.3. Hydrodynamic coefficients evaluation

With the use of the penalization method the hydrodynamic forces and the moments acting on the body, which are usually evaluated via surface integrals of the stress tensor $\sigma(\mathbf{u}, p) = \mu(\nabla\mathbf{u} + (\nabla\mathbf{u})^T)/2 - p\mathbf{I}$, can be computed directly by integrating the penalized velocity over the considered volume (surface in two-dimensions). Thus the forces

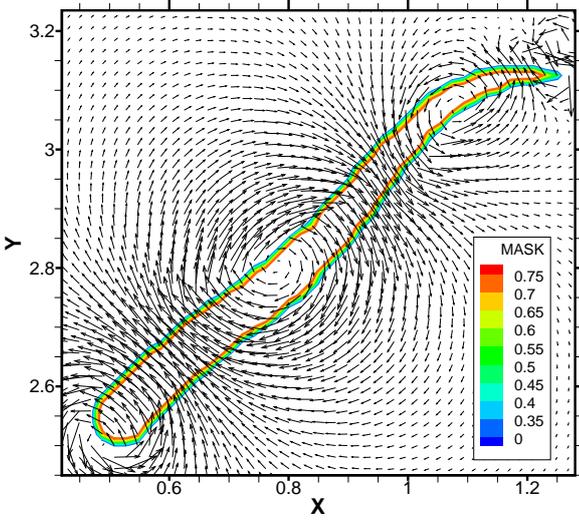


Figure 4: Interpolated smoothed mask function $\bar{\chi}$ (colored isolines) and the velocity components.

in [Newton] are given by

$$\mathbf{F}^* = \oint_{\partial\Omega_s} \boldsymbol{\sigma} \cdot \mathbf{n} \, dl = \lim_{\eta \rightarrow 0} \frac{\rho_f}{\eta} \int_{\Omega_s} \chi(\mathbf{u} - \mathbf{u}_B) \, ds + \rho_f S_{pen} \ddot{\mathbf{X}}_{cg} \quad (28)$$

for the unit mass ($m = \rho_f S_{pen}$) of the fluid. By definition $\mathbf{F} = \mathbf{F}^*/m$, we have

$$\mathbf{F} \approx \frac{1}{\eta S_{pen}} \int_{\Omega_s} \chi(\mathbf{u} - \mathbf{u}_B) \, ds + \ddot{\mathbf{X}}_{cg} \quad (29)$$

The moment in two-dimensions is evaluated by

$$M_{\mathbf{X}_{cg}} = \oint_{\partial\Omega_s} (\mathbf{x} - \mathbf{X}_{cg}) \times \boldsymbol{\sigma} \cdot \mathbf{n} \, dl = \lim_{\eta \rightarrow 0} \frac{\rho_f}{\eta} \int_{\Omega_s} \chi(\mathbf{x} - \mathbf{x}_{cg}) \times (\mathbf{u} - \mathbf{u}_B) \, ds + \frac{\rho_f}{\rho_b} J_{cg} \ddot{\theta}_{cg} \quad (30)$$

in $[N \cdot m]$, where $J_{cg} = \int \mathbf{r}^2 dm$ is the polar moment of inertia taken around center of the mass, \mathbf{n} is the unit outward vector to $\partial\Omega_s$, \mathbf{X}_{cg} is the location of the center of gravity of the immersed body, θ is the angle of rotation with respect to the center of gravity. The dots denote derivatives with respect to time and S_{pen} is the surface of the penalized area.

2.2.4. Denoising of the hydrodynamic coefficients

In dealing with fluid/solid interaction problems, the oscillation of the hydrodynamic forces and moments during successive iterations calculated from Eqs. (29) and (30) causes some trouble in correctly predicting the accelerations. The hydrodynamic forces and moments acting

on the body are used to calculate the linear and angular accelerations which in turn have impact on the predicted velocity vector and the trajectory of the solid. The oscillations are due to the nature of the penalization method see the discussion in [28], insufficient resolution, the approximative nature of Eqs. (29) and (30). The oscillations are like a noise and lead to an invalid result and even divergence of the simulations. An efficient method to eliminate them is to apply a low-pass filter like *exponential smoothing* usually used in denoising of time series. This filter is used in [22] to denoise the hydrodynamic forces and moments. Simple exponential smoothing does not perform well when there is a trend in the data. In such situations, several methods were devised like *second-order (double) exponential smoothing* [3]

$$\hat{F}^n = \alpha F^n + (1 - \alpha)(\hat{F}^{n-1} + b^{n-1}) \quad , \quad n = 3, 4, \dots \quad (31)$$

$$b^n = \beta(\hat{F}^n - \hat{F}^{n-1}) + (1 - \beta)b^{n-1} \quad , \quad (\alpha, \beta) \in [0, 1] \quad (32)$$

where $\hat{F}^1 = F^1$, for $n = 2$ one can use Eqs (31) and (32) with $\alpha = \beta = 1$. Then $\alpha = 1 - (1 - \delta)^2$ and $\beta = \delta^2/\alpha$ can be used in which δ is a small band. According to our experience $\delta = 10^{-3}$ performs well for denoising of the hydrodynamic forces, moments and centroid change of the deformable bodies. However $\delta = 10^{-3}$ have a strong damping effect, bigger values, i.e., $\delta = 5 \times 10^{-3}$ has less damping effect but there is a risk of divergence in the simulations. A sensitivity analysis must be done for each test case (see also the discussion of the results in Section 3.4).

2.2.5. Body dynamics

The dynamics of an arbitrary solid or deformable body moving in a viscous incompressible fluid is governed by Newton's second law

$$\Sigma(\mathbf{F}_H + \mathbf{F}_G) = m\ddot{\mathbf{X}} \quad (33)$$

where the applied forces can be split into two components; the hydrodynamic forces F_H and the forces due to gravity F_G . Newton's law can be integrated directly to give the position of the center of gravity as a function of time. Holding \mathbf{F} constant over the discrete physical time step (t^n, t^{n+1}) yields

$$\Delta \mathbf{X}_{cg} = \frac{1}{2} \frac{\mathbf{F}^n}{m} \Delta t^2 + \mathbf{V}^n \Delta t \quad (34)$$

and $V^{n+1} = V^n + \ddot{\mathbf{X}} \Delta t$. The rotational motion is described by Euler's equation

$$\Sigma M = \frac{d}{dt} (J_{cg} \dot{\theta}) \quad (35)$$

where M is the applied moment around the reference point. Time integration of Eq. (35) regardless of changes in the moment of inertia and M will give the new angle of the body with respect to a given reference

$$\Delta \theta = \frac{1}{2} \ddot{\theta}^n \Delta t^2 + \dot{\theta}^n \Delta t \quad (36)$$

where $\ddot{\theta} = M/J$ and $\dot{\theta}^{n+1} = \dot{\theta}^n + \ddot{\theta}\Delta t$ (the dots denote derivation with respect to time). Eqs (34) and (36) describe a motion with three degrees of freedom (3DOF) for the considered body. In these equations second-order terms can be eliminated as done in [33] but we are keeping these terms. Eqs. (29) and (30) provide the fluid forces and moment necessary to integrate the system of ODEs formed by Eqs. (33) and Eq. (35). Denoising of forces and moment is done according to (31). Appropriate initial conditions are necessary. In the present computations we use a first-order scheme for time integration of the dynamics equations which seems to be adequate because of the error introduced by penalization method which is also first-order. The same time integration method is also used in [28] and [33] for the dynamics of the body where the penalization is used. A summary of the algorithm for the fluid/structure interaction is given in Algorithm 1. The flowchart is illustrated in Fig. 5.

Algorithm 1 Fluid/structure interaction

1. START FROM AN INITIAL CONDITION
 2. BODY KINEMATICS
 - (a) (Just for the Eel) Create Eel's backbone by integrating Eqs (17), (19) and (21)
 - (b) (Just for the Eel) Cover the shape by a Lagrangian structured grid & calculate velocities of each point with Eq. (25)
 - (c) Compute the mask $\chi(i, j)$ and smooth it by Eq. (26)
 - (d) Compute the moment of inertia J around the reference point
 - (e) Compute the velocity components of the body $u_p(i, j), v_p(i, j)$ on the Eulerian grid (Lagrange \rightarrow Euler)
 3. TIME INTEGRATION OF FLOW FIELD VIA RK4
 - (a) $\omega_0^* = \omega^n$
For $i=1,2,3$
 - (b) Compute $k_i(\omega_{i-1}^*)$ from Eq. (8)
 - (c) $\omega_i^* = \omega^n + \alpha_i \Delta t k_i$
 - (d) Solve Eq. (2); $-\nabla^2 \psi_i = \omega_i^*$ for updating (u, v)
End For
 - (e) Compute $k_4(\omega_3^*)$ from Eq. (8)
 - (f) Update vorticity from Eq. (7)
 - (g) Solve Eq. (2); $-\nabla^2 \psi = \omega^{n+1}$
 4. SOLVE FOR THE BODY DYNAMICS
 - (a) Compute the hydrodynamic coefficients of the body from Eqs. (29) and (30)
 - (b) Denoise the coefficients by Eq. (31)
 - (c) Compute the displacements from Eq. (34)
 - (d) Compute the rotation from Eq. (36)
 5. WRITE NECESSARY DATA TO FILE
 6. IF $T < T_{end}$, GO TO STEP 2
 7. END
-

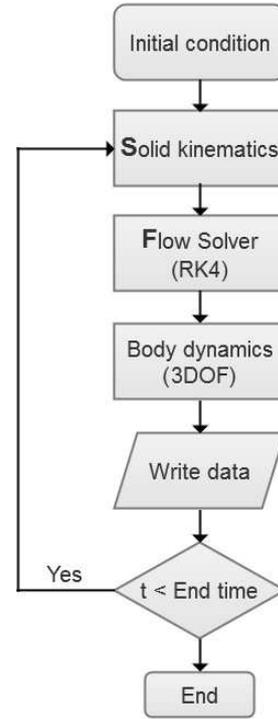


Figure 5: Flowchart of the fluid/solid interaction (FSI) algorithm.

3. Validation

The spatial convergence of the flow solver including the penalization term is verified using a Taylor-Couette flow for which an analytical solution is available. Then the solver for solid dynamics is verified via falling cylinder and ellipse in a quiescent fluid. Finally a test case of fish swimming in forward gait is compared with the results of Gazzola et al. [33].

3.1. Spatial convergence for a Taylor-Couette flow

For a rigorous study of the error due to the penalization term added to the Navier-Stokes equation in vorticity and stream-function formulation an exact solution is necessary. The Taylor-Couette configuration is a good choice, first and foremost, because of known Dirichlet boundary conditions everywhere, and secondly, because of the presence of curved walls contrary to other existing analytical solutions usually available for Cartesian domains which thus coincide with the underlying Cartesian grid used to discretize the governing equations. Although the solver is adapted to a Cartesian domain the mask function which represents the penalized area for the Taylor-Couette flow is curved (see Fig. 6) as it is the case for flow around an ellipse or complex geometries which will consider in following. An explicit second-order finite difference method and an implicit fourth-order compact finite difference are used for discretization of the governing equations including the curl of the penalization term $\nabla \times \mathbf{F}$. Taylor-Couette flow

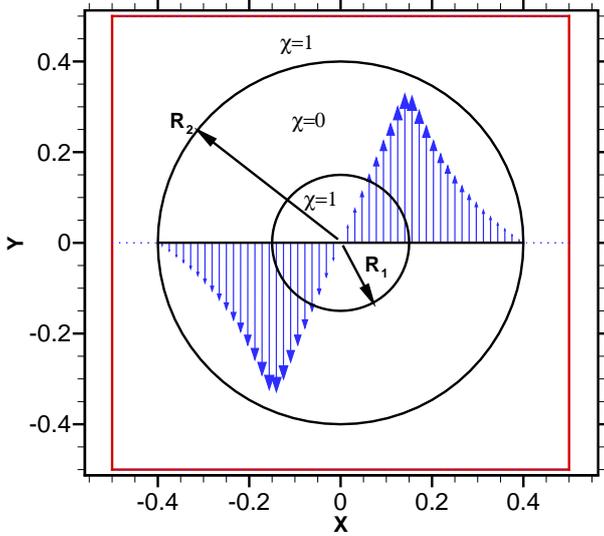


Figure 6: Schematic representation of the penalized unit square domain for modeling the Taylor-Couette flow with volume penalization method ($\chi = 0$ represents the fluid domain and $\chi = 1$ the solid domains, respectively). The radius of the inner cylinder is $R_1 = 0.15$ and that of the outer cylinder is $R_2 = 0.4$. The angular velocity of the inner cylinder is $\Omega_1 = 0.2$ and that of the outer is equal to zero, $\nu = 0.01$ and $Ta = R_1(\Omega_2 - \Omega_1)^2(R_2 - R_1)^3\nu^{-2} \approx 1$.

[1] consists of a viscous fluid confined between two concentric cylinders with radii (R_1, R_2) in rotation with different angular velocities (Ω_1, Ω_2). For Taylor numbers below the critical value $Ta_c \approx 1708$, the flow is steady and purely azimuthal, i.e., $u_z = u_r = 0$. This state is known as circular Taylor-Couette flow and for which there is an analytical solution [4]. The solution is given in cylindrical coordinates, the azimuthal velocity is $u_\theta(r) = Ar + B/r$ where $(r, \theta) \in [R_1, R_2] \times [0, 2\pi]$, $A = (\Omega_2 R_2^2 - \Omega_1 R_1^2)/(R_2^2 - R_1^2)$ and $B = R_1^2 R_2^2 (\Omega_1 - \Omega_2)/(R_2^2 - R_1^2)$ are known. The vorticity between the two cylinders is constant ($\omega_z = 2A$) and the stream-function is given by $\psi(r) = -Ar^2/2 - B \ln(r) + c_0$ where c_0 must be determined with respect to an arbitrary reference point. To use the volume penalization method, the velocity components must be enforced also in the solid regions from known angular velocities (i.e., Ω_1 and Ω_2), $u_\theta(r) = r\Omega$ where $(r, \theta) \in [0, R_1] \cup [R_2, R_{max}] \times [0, 2\pi]$. The vorticity inside the rotating regions is constant and is equal to twice of the domain's angular velocity ($\omega_z = 2\Omega$) and the stream-function is given by, $\psi(r) = \Omega/2r^2 + c$, where c must be determined for each domain in accordance with c_0 . A unit square domain is considered as the solution domain, the time-step of the RK4 method is calculated by the constraints presented in the Section 2.1.2 and the kinematic viscosity is fixed to $\nu = 0.01$. The radii are chosen $R_1 = 0.2$ and $R_2 = 0.4$, respectively. At $t = 0$ the fluid domain is at rest and the inner-cylinder is set into movement with a fixed angular velocity ($\Omega_1 = 0.2$) while the angular velocity of the outer cylinder is kept equal

to zero ($\Omega_2 = 0$). The Taylor number for this configuration ($Ta = 0.64$) is below the critical value, thus the flow is purely azimuthal. The L_1 -error $\|u^{\text{exact}} - u_\eta^{\text{num}}\|$ for u which is the x -component of the considered velocity field, is calculated for different penalization parameters η and resolutions (N in x and y directions). The simulations are carried out until a steady state is reached, so that the error is independent of the time discretization. The simulations are stopped when the time $t_{\text{end}} = 10$ is reached. Original and mollified mask function at the midline $y = 0.5$ is illustrated in Fig. 7 (a), comparison of the computed vorticity ω , stream-function ψ and the u velocity component with exact solutions for $N = 128$ grid points in each direction is plotted in Fig. 7 (a)-(b). The convergence of the L_1 -error of u versus the grid resolution is shown in Fig. 7 (c) for different penalization parameters. Suppose u_η^{num} denotes the numerical solution of the penalized equations, for quantifying the numerical error of u_η^{num} compared to u^{exact} (the solution to the original Navier-Stokes problem), the error can be estimated by

$$\|u^{\text{exact}} - u_\eta^{\text{num}}\| \leq \underbrace{\|u^{\text{exact}} - u_\eta\|}_{O(\sqrt{\eta})} + \underbrace{\|u_\eta - u_\eta^{\text{num}}\|}_{O(\Delta x^p)} \quad (37)$$

The first term at the right-hand side is the error due to the penalization term and the second term representing the discretization error (p being the formal order of accuracy of the numerical method used to discretize the equation). Where $\|\cdot\|$ is an appropriate norm. A compromise between two errors is to chose $\Delta x \approx \sqrt{\eta}$, which will lead the to a first-order convergence for the error $\|u^{\text{exact}} - u_\eta^{\text{num}}\| \leq O(\Delta x)$ [37]. The convergence of the L_1 -error of u versus different penalization parameters is shown in Fig. 7 (d) for different grid resolutions, where the order $\sqrt{\eta}$ can be observed. For these calculations the expected formal accuracy is $p = 2$ and thus the convergence is between first and second order in space as a function of the resolution N , confirming the theoretical analysis of Carbou et al. [18] and the numerical results of Morales et al. [36]. We also observe a saturation of the convergence error for large N , due to dominance of the penalization error. An optimal resolution can be found for each η . Decreasing η will lead to an accuracy enhancement in general but for an explicit integration time step $\Delta t = O(\eta)$ is limited by η as discussed in Section 2.1.2. A fine grid also needs a small η as can be seen in the Fig. 7.

3.2. Two-way fluid/solid interaction

In this section we attempt to perform a simulation of a two-dimensional cylinder falling in a quiescent fluid to validate the two-way fluid/solid interaction. We compare our results with those of Gazzola et al. [33] and Namkoong et al. [24] which have the same physical parameters. A rigid 2D cylinder of diameter $D = 0.005$ m with $\rho_b = 1.01\rho_f$, is released from rest in a fluid with density $\rho_f = 996$ kg/m³ and kinematic viscosity $\nu = 8 \times 10^7$ m²/s and accelerates due to gravity ($g = -9.81$ m/s²) until it reaches its

asymptotic terminal velocity. The domain size is set to $(x, y) \in [0, 0.04 \text{ m}] \times [0, 0.32 \text{ m}]$. The resolution in our simulations is set to 512×4096 , the penalization parameter $\eta = 10^{-3}$, the time step $\Delta t = 10^{-3}$ and filter parameter for denoising of the hydrodynamic coefficients $\delta = 0.001$. In the simulations of Gazzola et al. [33] the resolution is 1024×8192 , the penalization parameter $\eta = 10^{-4}$ and the Lagrangian $CFL = 0.01$. The snapshots of the vorticity isolines generated by the falling cylinder in a quiescent and slightly perturbed fluid are illustrated in Fig. 8 and Fig. 9, respectively. A qualitative agreement with the simulations of Gazzola et al. [33] can be observed. Comparison of the vorticity structures at $t = 13$ between the simulation with slightly perturbed initial condition ($\mathbf{u}' \approx 0.001 \times \text{random number} \times u_\infty$) represented in Fig. 9 (h) and that of quiescent initial condition represented in Fig. 8 (d) shows that presence of a perturbation in the initial condition can trigger the transition in the early stage of falling, i.e., $t \approx 3$. Particularly this is important to obtain comparable results with other simulations with different numerical methods where the added numerical dissipation is not necessarily the same. Without adding any initial perturbation the transition can be triggered (e.g., at $t \approx 10$) by the numerical errors which are performing like a perturbation (see Fig. 8). This is not controlled, it depends on grid resolution and the numerical implementation and explains the delayed streamwise velocity overshoot and the different transient flow fields. Fig. 10 shows the time evolution of the streamwise velocity obtained with the present method and those of Namkoong et al. [24] and Gazzola et al. [33]. As can be seen the streamwise velocity shows the same dynamics as the reference simulations. In particular the streamwise velocity obtained by simulation with perturbed initial condition overshoots above the terminal velocity and then slows down when the vortices start shedding. The terminal streamwise velocity of the simulation with perturbed initial condition is $u_{\text{streamwise}} = 0.023 \text{ m/s}$ resulting a Reynolds $Re \approx 144$ and that of unperturbed initial condition is $u_{\text{streamwise}} = 0.025 \text{ m/s}$ resulting a Reynolds $Re \approx 156$. In the former an overshoot can be observed in the streamwise velocity while in the later the overshoot takes place in a bigger time interval or it is entirely eliminated. The terminal velocity differs 8% from the reference terminal velocity in the case of perturbed initial condition and coincide in the case of unperturbed initial condition. The differences are due to different Poisson solvers which is unbounded in the simulation of Gazzola et al. [33], the boundary conditions which is free-slip and no-penetration in our simulations, different penalization parameters and resolutions. In the author's viewpoint the take-home message here is that the near one relative fluid/solid density leads to a small buoyancy where invalid approximation of the hydrodynamic coefficients specially in the early stages of the falling yields the simulation to a failure. To cope with this challenge the process of filtering of the hydrodynamic coefficients with a proper parameter is devised in the proposed algorithm. For bigger density

ratios $\rho_b > 1.5\rho_f$ the difficulty is not serious and can be handled even without denoising of the hydrodynamic coefficients.

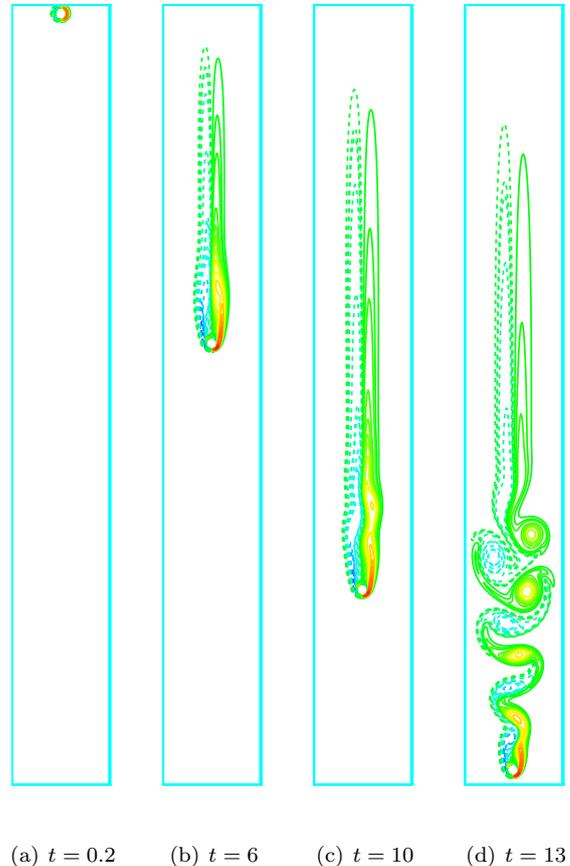


Figure 8: Vorticity isolines (dashed lines are used for negative values) of a falling cylinder in quiescent fluid, $g = -9.81 \text{ m/s}^2$, $\rho_b/\rho_f = 1.01$, $D = 0.005 \text{ m}$, $(x, y) \in [0, 0.04 \text{ m}] \times [0, 0.32 \text{ m}] = [0, 8D] \times [0, 64D]$ and $\nu = 8 \times 10^{-7} \text{ m}^2/\text{s}$ and $Re \approx 156$.

3.3. Validation of the solid dynamics via a falling ellipse

The validation of the body dynamics interacting with a fluid is done by a falling ellipse due to gravity in a fluid in rest. Different behaviors like steady falling, fluttering, tumbling and chaotic motion can be observed by varying the ellipse's geometry, density ratio and the viscosity of the fluid. These parameters can be summarized in dimensionless moment of inertia

$$J_{\text{cg}}^* = 2J_{\text{cg}}/(a^4 \rho_f) = \pi(a^2 + b^2)(b/2a^3)(\rho_b/\rho_f)$$

and the Reynolds number $Re = u_t L/\nu$, where u_t is the sedimentation average velocity estimated by

$$u_t = \sqrt{4bg(\rho_b/\rho_f - 1)}$$

see [33]. The domain of the solution is $(x, y) \in [0, 5L] \times [0, 20L]$ where $L = 2a = 1$ and $H = 2b = 0.2$ are the major

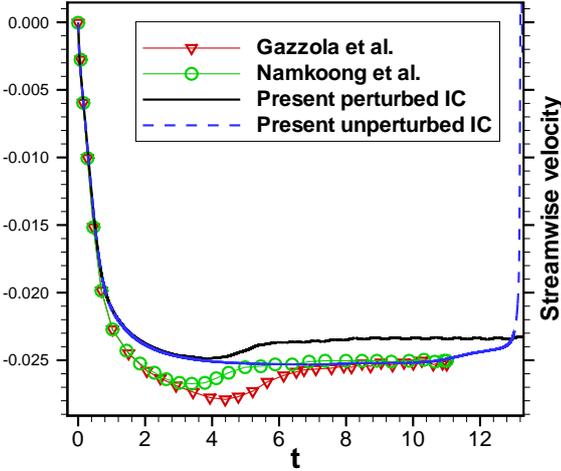


Figure 10: Comparison of the streamwise velocity of a falling cylinder via different methods/parameters. Symbols indicate the reference simulations performed by Gazzola et al. [33] (gradients) and Namkoong et al. [24] (circles). Solid and dashed lines represent the results with the proposed algorithm performed by a perturbed and unperturbed initial conditions (IC), respectively.

and minor diameters of the ellipse respectively. The resolution of the grid is 512×2048 , the polar moment of inertia around the center of gravity $J_{cg} = 0.25\pi ab(a^2 + b^2)\rho_b$, the initial position $(x_0, y_0) = (0.5L_x, L_y - 3a)$ and the initial angle of the major diameter with respect to the horizon $\theta_0 = \pi/4$. The density ratio is set to $\rho_b = 1.538\rho_f$, the filter parameter for denoising of the hydrodynamic coefficients $\delta = 0.001$, the kinematic viscosity $\nu = 0.01 \text{ m}^2/\text{s}$, the gravity in y -direction $g = -9.81 \text{ m/s}^2$ and the penalization parameter $\eta = 10^{-3}$. The trajectory and the isolines of the vorticity generated by the falling ellipse in the fluttering regime are illustrated in Fig. 11 at different instants from $t = 0.2$ up to $t = 25$. A qualitative agreement with the simulations of Kolomenskiy et al. [28] and Gazzola et al. [33] can be observed. The corresponding forces and velocity components are plotted in Fig. 12. All of the hydrodynamic coefficients show an oscillatory behavior with a principal frequency $f_1 \approx 0.24$. However in the side force a harmonic frequency with $f_2 = 2f_1 \approx 0.48$ can be seen which is due to the shedding of the vortices. The chosen reference point for the simulation of the falling ellipse is the center of gravity (cg) for calculation of the polar moment of inertia, rotation angle and the moment. This choice is advantageous for simplification of the Euler Eq (35) to not include the torque due to buoyancy. For the simulations of the fish swimming the buoyancy is set to zero, i.e. $\rho_b = \rho_f$. Thus the reference point can coincide with the head which is more suitable for creating the fish starting by the situation of the head.

3.4. Fish in forward gait

Anguilliform swimming presented in Gazzola et al. [33] is considered for further validation of the proposed algorithm. A periodic swimming law is defined by fitting the backbone of the fish to a given curve $y(x, t)$ while keeping the backbone length l_{fish} fixed. Let ξ be the arclength over the curvilinear coordinate of the deformed backbone ($0 \leq \xi \leq l_{\text{fish}}$). For points being uniformly distributed with $\Delta\xi = l_{\text{fish}}/(N - 1)$, over the backbone, y is given by

$$y(x, t) = a(x) \sin(2\pi(x/\lambda + ft)) \quad (38)$$

where λ is the wavelength, f represents the frequency of the backbone and the envelope $a(x)$ is given by

$$a(x) = a_0 + a_1x + a_2x^2 \quad (39)$$

where x is defined by inverting the arclength integral, i.e., $\Delta x = \Delta\xi/\sqrt{1 + (\partial y/\partial x)^2}$. The wavelength of the fish is defined in accordance with the geometry of the backbone in the Cartesian coordinate. The pointwise curvature of the backbone is needed to be able to use the geometrically exact theory of nonlinear beams. One must switch from the Cartesian system to the curvature, thus second derivative of Eq. (38) will lead to

$$k(\xi, t) = (2a_2 - (2\pi/\lambda)^2 a(\xi)) \sin(2\pi(\xi/\lambda + ft)) + (4\pi(a_1 + 2a_2\xi)/\lambda) \cos(2\pi(\xi/\lambda + ft)) \quad (40)$$

where $a(\xi) = a_0 + a_1\xi + a_2\xi^2$. The parameters used by Kern and Koumoutsakos [22] and Gazzola et al. [33] for the kinematics of the fish are as follows; $\lambda = 1$, $f = 1$, $a_2 = 0$, $a_1 = 0.125/(1 + c)$, $a_0 = 0.125c/(1 + c)$ and $c = 0.03125$. The buoyancy is equal to zero, i.e., $\rho_b = \rho_f$. The viscosity of the fluid is set to $\nu = 1.4 \times 10^{-4}$ resulting in a Reynolds number approximately $Re \approx 3800$, with an asymptotic mean velocity $U_{\text{forward}} \approx 0.52$. The simulations of Gazzola et al. [33] are carried out on a rectangular domain $(x, y) \in [0, 8l_{\text{fish}}] \times [0, 4l_{\text{fish}}]$ with resolution of 4096×2048 and penalization parameter $\eta = 10^{-4}$. We are performing our simulations on a rectangular domain $(x, y) \in [0, 10l_{\text{fish}}] \times [0, 5l_{\text{fish}}]$ by imposing a penalization parameter inside the body equal to $\eta = 10^{-3}$ with resolution of 2048×1024 and 1024×512 and $\Delta t = 10^{-3}$. The centroid of the fish is initially positioned at $x_{cg} = 0.9L_x$ and $y_{cg} = 0.5L_y$. The forward velocities of the center of the mass computed with different methods/parameters are compared in Fig. 14. We impose two degrees of freedom fixing the angular velocity equal to zero. The simulations start with the body and fluid at rest. The motion of the fish is initialized by gradually increasing the amplitude of the backbone through a sinusoidal function (see Fig. 13) from zero to its designated value during the first period T in the reference simulations, i.e., [22] and [33]. Here we are not considering this and start by a sudden movement given by Eq. (38). This is the reason why a deviation from the reference solution can be seen in the first period. This

deviation will continue systematically until the asymptotic velocity is reached at $t = 7$.

The reference simulation of Kern and Koumoutsakos [22] is based on a body fitted finite volume method which is first-order in time and second-order in space. The Navier-Stokes equations were solved using the commercial package STAR-CD which computes flows on arbitrary Lagrangian-Eulerian grids. The solution of Newtons equations of motion and the resulting grid deformation and motion are implemented in user defined subroutines linked to STAR-CD. The implemented explicit coupling procedure is a staggered integration algorithm proposed by Farhat and Lesoinne [17]. The simulation of Gazzola et al. [33] is done by a code based on a vortex particle method coupled with Brinkman penalization which handles arbitrarily deforming bodies and specially the corresponding divergent velocity field inside the body. A projection method is used by Gazzola et al. [33], the resulting Poisson equations for rotational (solenoidal) and potential (divergent) components of the velocity fields are solved by an unbounded domain, FFT based solver in Cartesian grids. A second and fourth order finite difference discretization is used for all other spatial derivatives. The time step is adapted by a Lagrangian CFL condition. The difference on the final forward velocity of the fish reported by Gazzola et al. [33] by taking into account the divergence of the velocity field inside the fish due to deformation of the fish is visible in Fig. 14. Even though the average divergence over the fish volume is zero (i.e. the volume is conserved) locally inside the fish the velocity field is not divergence free. We are not dealing with this issue in this paper because for the moment we are not interested in swimmers with high thickness. In our simulations a grid independent solution is obtained with 2048×1024 grid points. The difference of two simulations with 2048×1024 and 1024×512 grid points can be seen in Fig. 14. Filtering of the hydrodynamic coefficients is necessary to prevent the simulation from divergence and non-physical results. We are using a second-order exponential filtering (31) instead of the first-order filtering used in [22]. This process is like adding a damper to the system therefore a proper value for δ must be chosen via numerical tests for obtaining physical results. We propose values in the range of $\delta \in [0.01, 0.001]$ for fluid/solid interaction problems, however this can also depend on the manner of non-dimensionalization of the forces. In Fig. 14 the effect of filtering with two filter parameter, i.e., $\delta = 0.001$ and $\delta = 0.05$, can be seen. Filtering with a small filter parameter $\delta = 0.001$ will be more stable but instead will lead to smaller values in the terminal velocity and also a smaller amplitude in its oscillations. In Fig. 15 two snapshots of vorticity isolines at $t = 1$ and $t = 9$ with the aforementioned parameters are given.

4. Application and results

We attempt to propose an efficient law for rotation of an anguilliform swimmer. Fish maneuvering for track-

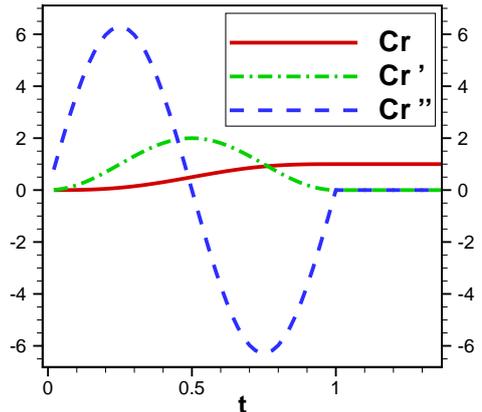


Figure 13: Smooth step function $Cr(t) = t' - \sin(2\pi t')/(2\pi)$, $t \in [t_i, t_f]$ with $t' = (t - t_i)/(t_f - t_i)$, $t_i = 0$, $t_f = 1$ for gradually starting the motion proposed by Boyer et al. [21]. At $t = 1$ the left-and right-hand limits are equal for the function Cr and its first Cr' and second Cr'' derivatives.

ing a fixed goal is done by adding a constant curvature $k_{\text{offset}}(\theta_{\text{des}}, t)$ all along the fish's backbone $\xi \in [0, l_{\text{fish}}]$, to the primary propulsion mode, i.e., $k_3 = k(\xi, t) + k_{\text{offset}}$. However the change of the added curvature k_{offset} via Eq. (42) must be gradually, i.e., $O(\Delta t)$ to perform a physically reasonable rotation. For the fish in forward gait k_{offset} is set equal to zero. For applying a rotation, first a desired curvature k_{des} must be evaluated by the following relation,

$$k_{\text{des}}(\theta_{\text{des}}) = \begin{cases} -\text{sgn}(\theta_{\text{des}}) k_{\text{max}} & |\theta_{\text{des}}| \geq \theta_{\text{limit}} \\ -\text{sgn}(\theta_{\text{des}}) k_{\text{max}} \left(\frac{\theta_{\text{des}}}{\theta_{\text{limit}}}\right)^2 & \text{else} \end{cases} \quad (41)$$

where sgn represents the sign function, i.e., $\text{sgn}(\theta_{\text{des}}) = \theta_{\text{des}}/|\theta_{\text{des}}|$. For a schematic representation of θ_{des} see Fig. 16. A desired angle θ_{des} should be calculated in each time step according to the position and direction of the head by considering the target. Then by using Eq. (41) a desired curvature k_{des} must be found. After that k_{offset} will be evaluated with the following relation,

$$k_{\text{offset}}^{n+1}(k_{\text{des}}) = \begin{cases} k_{\text{offset}}^n + \Delta k & k < k_{\text{desired}} \\ k_{\text{offset}}^n - \Delta k & \text{else} \end{cases} \quad (42)$$

where $\Delta k = \Delta t \pi/T$, k_{offset} will be added to the backbone curvature for performing a rotation. We are using $k_{\text{max}} = \pi$ which is equivalent to turning with a curvature adapted to a semicircle. As in Bergmann and Iollo [34] we are using $\theta_{\text{limit}} = \pi/4$. The time derivative of the curvature dk/dt is needed in Eq. (21) for velocity calculation and can be calculated numerically. To show the performance of the proposed method a test case of food finding with the above rotation law is performed. The domain size is $(x, y) \in [0, 5l_{\text{fish}}] \times [0, 5l_{\text{fish}}]$, the resolution is set to 1024×1024 , the penalization parameter $\eta = 10^{-3}$, filter parameter $\delta = 0.005$, tail beat frequency $f = 1$, wavelength $\lambda = 1$, kinematic viscosity $\nu = 1.4 \times 10^{-4}$,

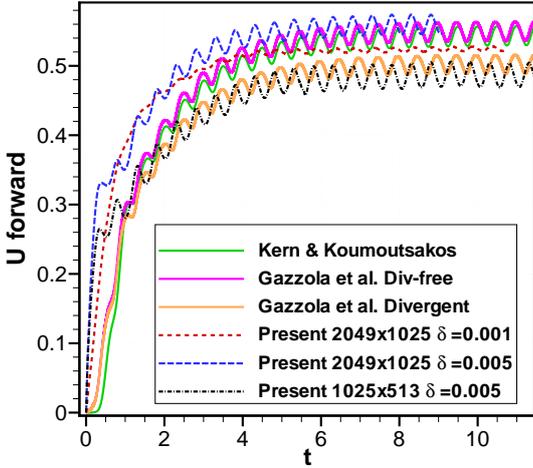


Figure 14: Forward velocity U of a 2D anguilliform swimmer's ($\lambda = f = 1$). Solid lines indicate the reference simulations performed by Kern and Koumoutsakos (green) [22] and Gazzola et al. (pink and brown) [33]. Dashed lines represent the results with the proposed algorithm.

initial position of the head $(x_0, y_0) = (0.1L_x, 0.5L_y)$ and initial angle of the head $\theta_0 = 0$. Fig. 17 shows snapshots of vorticity isolines obtained during a simulation of swimming fish for finding a food which is located at $(x_f, y_f) = (0.9L_x, 0.5L_y)$. At $t = 0$ the fish and the surrounding flow are in rest. After reaching the vicinity ($r_{\text{food}} = 0.5l_{\text{fish}}$) of the food the curvature of the backbone, given by Eq. (40), will tend to zero by multiplying it with the following function,

$$C(t) = \frac{t_f - t}{t_f - t_i} + \frac{1}{2\pi} \sin\left(2\pi \frac{t - t_i}{t_f - t_i}\right), \quad t \in [t_i, t_f] \quad (43)$$

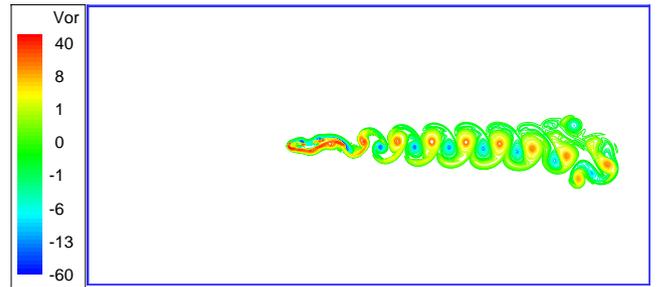
which is the mirror of the function presented in Fig. 13, with $t_i = t_{\text{reached}}$, $t_f = t_{\text{reached}} + T$ for gradually decreasing the curvature of the backbone during one period. With the proposed law for rotation which is added to the primary propulsion mode, the fish can perform a 180° change in swimming direction within an area of about its length.

5. Conclusion

In this paper an efficient algorithm for simulation of deformable bodies interacting with two-dimensional incompressible flows is proposed. By using a uniform Cartesian grid a new fourth-order direct solver for the solution of the Poisson equation is presented. For introducing a deformable body in fluid flow, the volume penalization method is applied to the solution of the Navier-Stokes equations as a forcing term. Even if the penalization method is first-order in space, an important advantage of this method is that the evaluation of the hydrodynamic coefficients is straightforward. Proper denoising of the hydrodynamic coefficients is vital for dealing with fluid/solid



(a) $t = 1$



(b) $t = 9$

Figure 15: Snapshots of vorticity isolines obtained during a simulation where $(x, y) \in [0, 10l_{\text{fish}}] \times [0, 5l_{\text{fish}}]$ by imposing a penalization parameter inside the body equal to $\eta = 10^{-3}$ with resolution of 2048×1024 , the Reynolds number is approximately $Re \approx 3800$.

interaction problems via the volume penalization method. An efficient law for curvature control of an anguilliform swimmer looking for food is proposed which is based on geometrically exact theory of nonlinear beams. Validation of the developed method shows the efficiency and expected accuracy of the algorithm for fish-like swimming control and also for a variety of fluid/solid interaction problems. Some perspectives for future developments are to develop high-order immersed boundary methods, adding a multi-resolution analyses to the algorithm for grid adaptation, enhancement of rotation law, parallelization and extension to three dimensions. The FORTRAN code is open source and is also accessible upon request [42].

Acknowledgment

We give our thanks to professor Boyer and colleagues for useful discussions and sending their code describing Eel's backbone kinematics in forward gait, to Professor Gazzola and colleagues for sending the needed data for comparisons.

References

- [1] G. I. Taylor, Stability of a viscous liquid contained between two rotating cylinders. *Philosophical Transactions of the Royal*

- Society*, A 223: 289-343, 1923.
- [2] F. G. Shuman, Numerical methods in weather prediction: II Smoothing and filtering. *Mon. Wea. Rev.*, Vol. 85, 357-361, 1957.
 - [3] C. C. Holt, Forecasting trends and seasonal by exponentially weighted averages. *Office of Naval Research Memorandum, Carnegie Institute of Technology*, 52, 1957.
 - [4] A. S. Monin and A. M. Yaglom, *Statistical Fluid Mechanics*, MIT Press, Cambridge, MA, 1971.
 - [5] R. S. Hirsh, High order accurate difference solutions of fluid mechanics problems by a compact differencing technique. *Journal of Computational Physics*, Vol. 19, 90-109, 1975.
 - [6] P. Bontoux, B. Forestier and B. Roux, Analyse et optimisation d'une méthode de haute précision pour la résolution des équations de Navier-Stokes instationnaires. *Journal de Mécanique Appliquée*, Vol. 2, no. 3, 291-316, 1978.
 - [7] B. Roux, P. Bontoux, T. P. Loc and O. Daube, Optimisation of Hermitian methods for Navier-Stokes equations in the vorticity and stream-function formulation. *West Germany, September 9-15, 1979. (A81-36526 16-34) Berlin, Springer-Verlag*, p. 450-468, 1980.
 - [8] J. C. Simo, A finite strain beam formulation. The three-dimensional dynamic problem. Part I, *Computer methods in applied mechanics and engineering*, Vol. 49, 55-70, 1985.
 - [9] J. Kim and P. Moin, Application of a fractional-step method to incompressible Navier-Stokes equations. *Journal of Computational Physics*, Vol. 59, 308-323, 1985.
 - [10] S. K. Lele, Compact finite difference schemes with spectral-like resolution. *Journal of Computational Physics*, Vol. 103, 16-42, 1992.
 - [11] H. Press, A. Teukolsky, T. Vetterling and P. Flannery, *Numerical Recipes*. Cambridge University Press, 1992.
 - [12] W. F. Spitz and G. F. Carey, High-order compact scheme for the steady stream-function vorticity equations. *International Journal for Numerical Methods in Engineering*, Vol. 38, 3497-3512, 1995.
 - [13] J. Carling, T. L. Williams and G. Bowtell, Self-propelled anguilliform swimming: simultaneous solution of the two-dimensional Navier-Stokes equations and Newton's laws of motion. *Journal of experimental biology*, Vol. 201, 3143-3166, 1998.
 - [14] P. Angot, C.-H. Bruneau and P. Fabrie, A penalization method to take into account obstacles in incompressible viscous flows. *Numerische Mathematik*, Vol. 81, no. 4, 497-520, 1999.
 - [15] P. Orlandi, *Fluid flow phenomena: A numerical toolkit*. Springer, 2000.
 - [16] M. Forestier, Etude par méthode spectral de sillages tridimensionnels en fluide stratifié. *Thèse de Doctorat, Université de Nice-Sophia*, 2000.
 - [17] C. Farhat and M. Lesoinne, Two efficient staggered procedures for the serial and parallel solution of three-dimensional nonlinear transient aeroelastic problems. *Comp. Meth. Appl. Mech. Eng.*, Vol. 182, 499-516, 2000.
 - [18] G. Carbou and P. Fabrie, Boundary layer for a penalization method for viscous incompressible flow. *Advances in Differential Equations*, 8:1453, 2003.
 - [19] S. Abide and S. Viazzo, A 2D compact fourth-order projection decomposition method. *Journal of Computational Physics*, Vol. 206, 252-276, 2005.
 - [20] R. Mittal and G. Iaccarino, Immersed boundary methods. *Annual Reviews of Fluid Mechanics*, Vol. 37, 239-261, 2005.
 - [21] F. Boyer, M. Porez and W. Khalil, Macro-continuous computed torque algorithm for a three-dimensional Eel-like robot. *IEEE Transactions on Robotics*, Vol. 22, No. 4, 763-775, August 2006.
 - [22] S. Kern and P. Koumoutsakos, Simulations of optimized anguilliform swimming. *Journal of Experimental Biology*, Vol. 209, 4841-4857, 2006.
 - [23] J. W. Kim, Optimised boundary compact finite difference schemes for computational aeroacoustics. *Journal of Computational Physics*, Vol. 225, 995-1019, 2007.
 - [24] K. Namkoong, J.Y. Yoo, H.G. Choi, Numerical analysis of two-dimensional motion of a freely falling circular cylinder in an infinite fluid, *Journal of Fluid Mechanics*, Vol. 604, 33-53, 2008.
 - [25] M. El Rafei, M. Alamir, N. Marchand, M. Porez and F. Boyer, Multi-variable constrained control approach for a three-dimensional Eel-like robot. *IEEE/RSJ 2008 International Conference on Intelligent Robots and Systems, IROS*, Nice, France, 2008.
 - [26] R. Pasquetti, R. Bwemba and L. Cousin, A pseudo-penalization method for high Reynolds number unsteady flows. *Applied Numerical Mathematics*, Vol. 58, 946-954, 2008.
 - [27] M. Minguez, Large eddy simulation of turbulent flows over complex geometries: Application to a simplified car model. *PhD thesis, Université Paul Cézanne, Aix-Marseille III*, 2008.
 - [28] D. Kolomenskiy and K. Schneider, A Fourier spectral method for the Navier-Stokes equations with volume penalization for moving solid obstacles. *Journal of Computational Physics*, Vol. 228, no. 16, 5687-5709, 2009.
 - [29] F. Sabetghadam, S. A. Ghaffari and M. Dadashi, Implementation of vortex stretching into the two-dimensional Navier-Stokes equations via arbitrary external straining. *Advances in Turbulence XII, Springer Proceedings in Physics*, 132, 2009.
 - [30] S. Laizet, E. Lamballais and J.C. Vassilicos, A numerical strategy to combine high-order schemes, complex geometry and parallel computing for high resolution DNS of fractal generated turbulence. *Computers & Fluids*, Vol. 39-3, pp 471-484, 2010.
 - [31] K.S. Yeo, S.J. Ang and C. Shu, Simulation of fish swimming and manoeuvring by an SVD-GFD method on a hybrid meshfree-Cartesian grid. *Computers & Fluids*, Vol. 39, 403-430, 2010.
 - [32] D. Rossinelli, M. Bergdorf, G. H. Cottet and P. Koumoutsakos, GPU accelerated simulations of bluff body flows using vortex particle methods. *Journal of Computational Physics*, Vol. 229, 3316-3333, 2010.
 - [33] M. Gazzola, P. Chatelain, W. M. van Rees and P. Koumoutsakos, Simulations of single and multiple swimmers with non-divergence free deforming geometries. *Journal of Computational Physics*, Vol. 230, 7093-7114, 2011.
 - [34] M. Bergmann and A. Iollo, Modeling and simulation of fish-like swimming. *Journal of Computational Physics*, Vol. 230, 329-348, 2011.
 - [35] B.J. Boersma, A 6th order staggered compact finite difference method for the incompressible Navier-Stokes and scalar transport equations. *Journal of Computational Physics*, Vol. 230, Issue 12, 4940-4954, 2011.
 - [36] J. A. Morales, M. Leroy, W. J. T. Bos and K. Schneider, Simulation of confined magnetohydrodynamic flows using a pseudo-spectral method with volume penalization. *HAL*, 00719737, 2012.
 - [37] R. Nguyen van yen, D. Kolomenskiy and K. Schneider, Approximation of the Laplace and Stokes operators with Dirichlet boundary conditions through volume penalization: a spectral viewpoint. *arXiv:1206.0002*, 2012.
 - [38] A. Belkhir, Modélisation dynamique de la locomotion compliante : Application au vol battant bio-inspiré de l'insecte. *Thèse de doctorat, École nationale supérieure des mines de Nantes*, 2013.
 - [39] T. Engels, D. Kolomenskiy, K. Schneider and J. Sesterhenn, Two-dimensional simulation of the fluttering instability using a pseudospectral method with volume penalization. *Computers & Structures*, Vol. 122, 101-113, 2013.
 - [40] F. Sotiropoulos and X. Yang, Immersed boundary methods for simulating fluid-structure interaction. *Progress in Aerospace Sciences*, Vol. 65, 1-21, 2014.
 - [41] P. Bontoux, S. Viazzo, K. Schneider and S. A. Ghaffari, An efficient algorithm for simulation of forced deformable bodies interacting with incompressible flows; Application to fish swimming. *11th World Congress on Computational Mechanics*, Barcelona, Spain, 2014.
 - [42] *The code is developed in FORTRAN and is accessible for all by sending a mail to: gghaffari@L3m.univ-mrs.fr or s.amin.gghaffary@gmail.com*

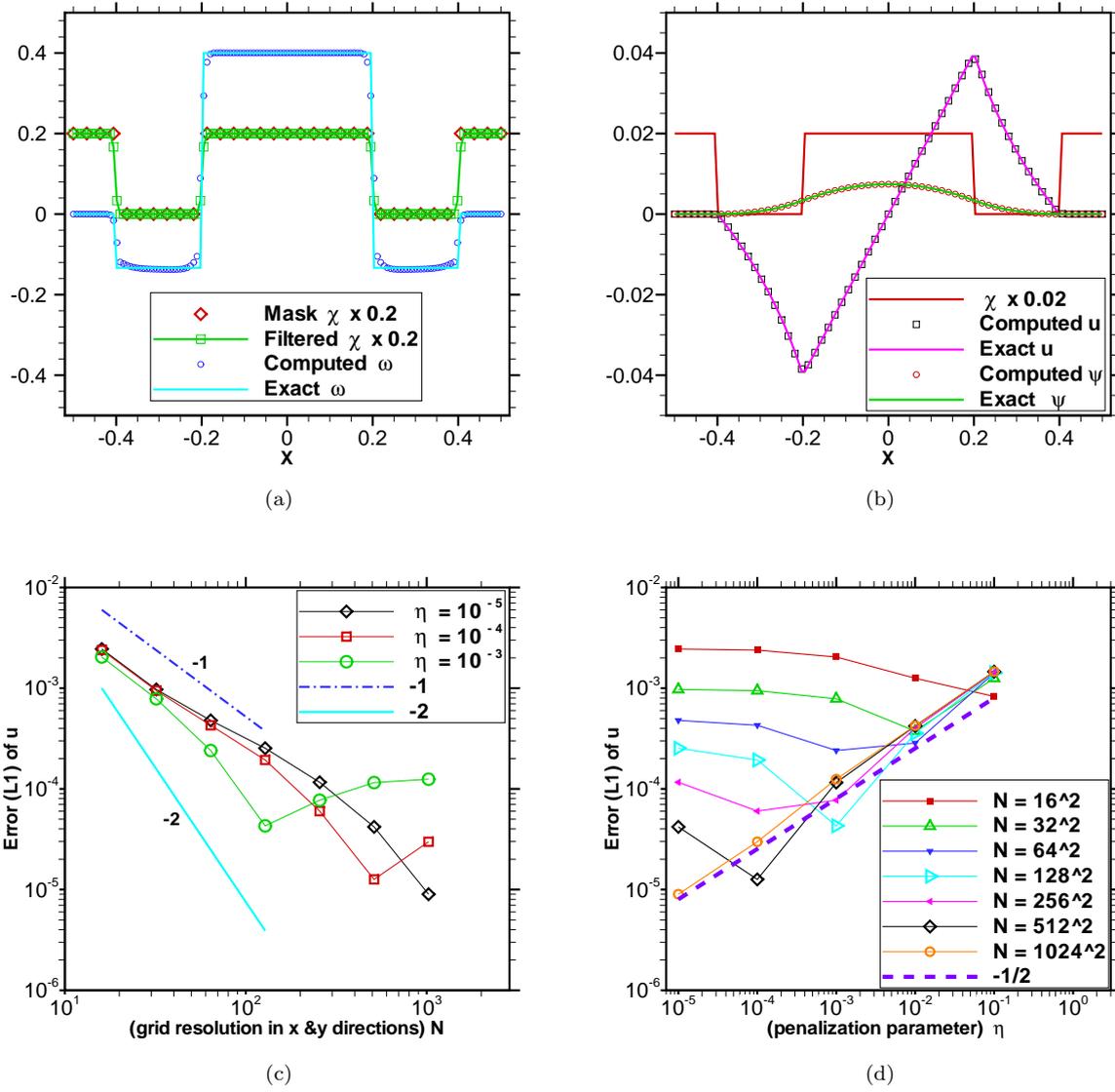


Figure 7: (a) Original and mollified mask function of a Couette flow, comparison of computed vorticity ω with the exact solution at $y = 0.5$ with $N = 128$ grid points in each direction. (b) Comparison of the computed stream-function ψ and the u velocity component with the exact solutions at $y = 0.5$ with $N = 128$. (c) L_1 -error of u versus spatial resolution for different values of η , N represents the grid resolution in each direction. (d) L_1 -error of u versus the penalization parameter η for different resolutions N .

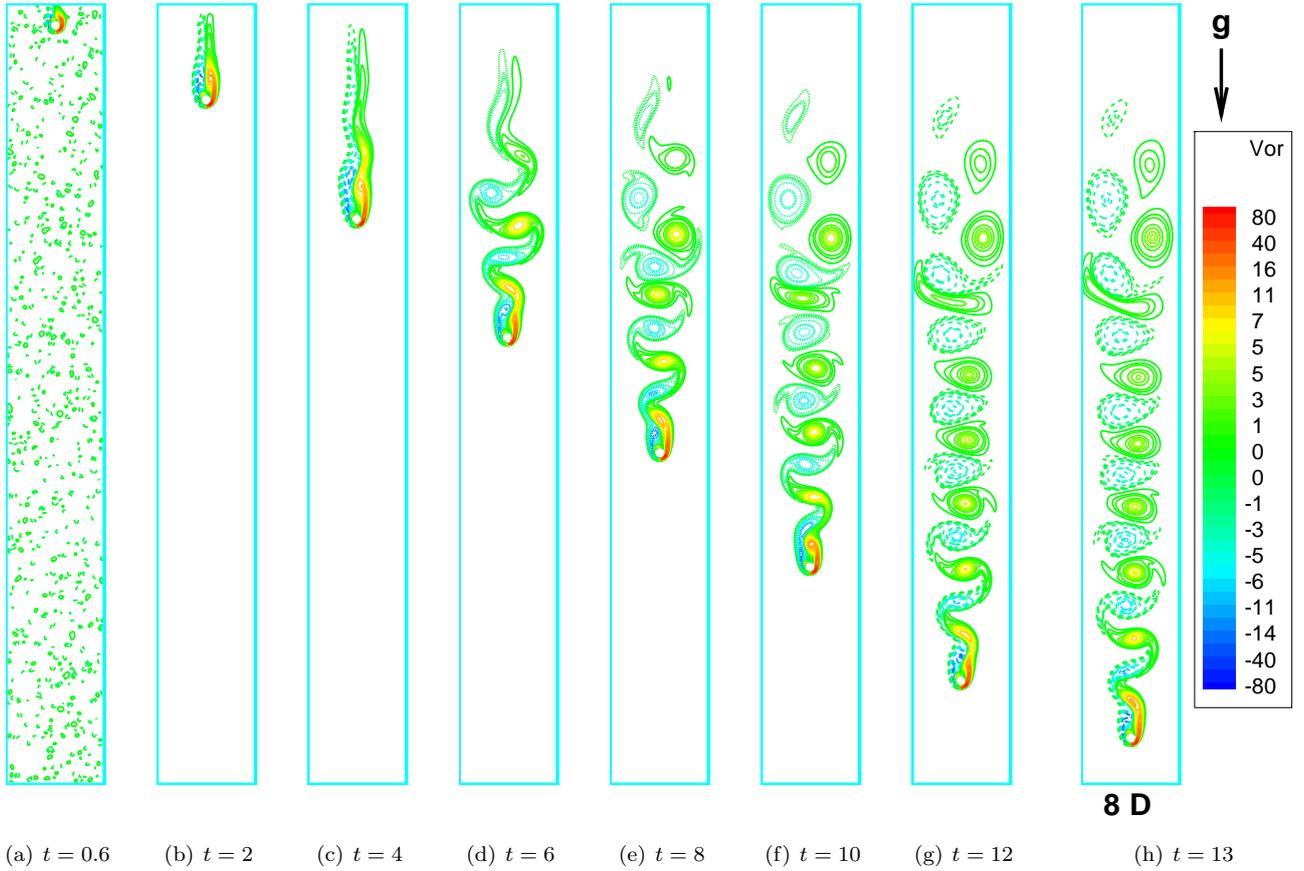


Figure 9: Vorticity isolines (dashed lines are used for negative values) of a falling cylinder in a slightly perturbed fluid, $g = -9.81 \text{ m/s}^2$, $\rho_b/\rho_f = 1.01$, $D = 0.005 \text{ m}$, $(x, y) \in [0, 0.04 \text{ m}] \times [0, 0.32 \text{ m}] = [0, 8D] \times [0, 64D]$, $\nu = 8 \times 10^{-7} \text{ m}^2/\text{s}$ and $Re \approx 144$.

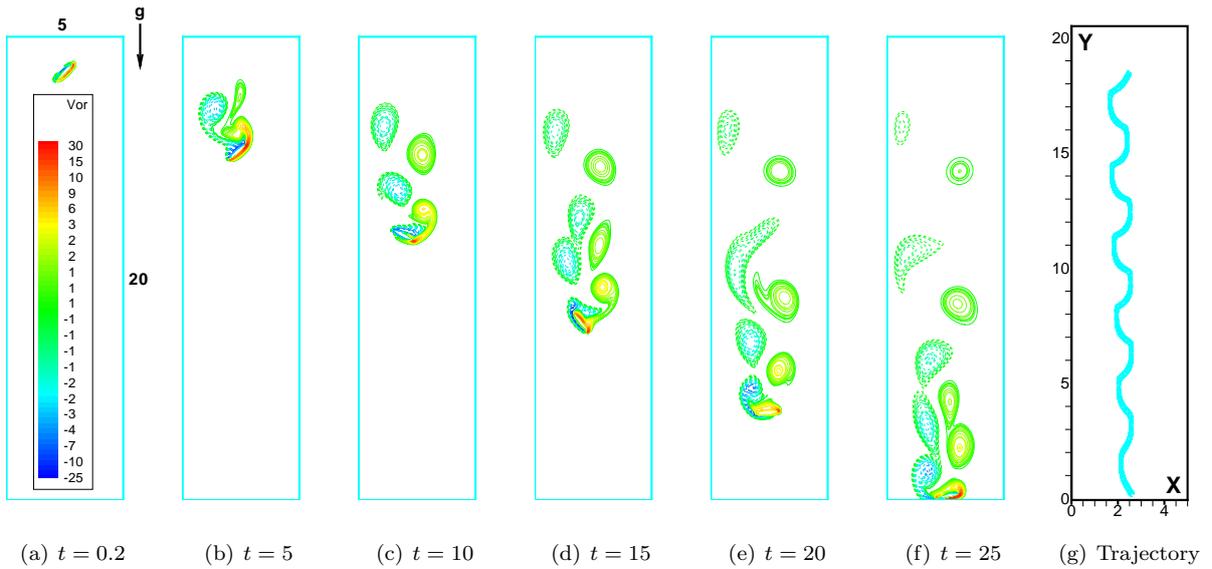


Figure 11: Vorticity isolines (dashed lines are used for negative values) of a falling ellipse in the fluttering regime, $I^* = 0.16$, $\rho_b/\rho_f = 1.538$, $a/b = 1/5$ and $\nu = 0.01$.

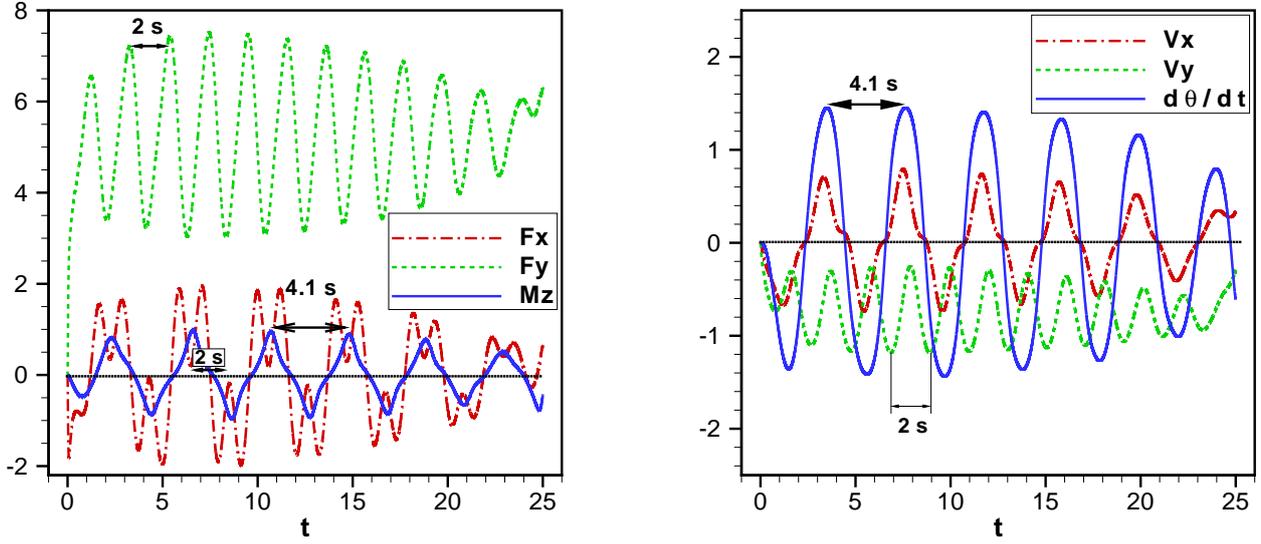


Figure 12: Forces (left) and velocity components (right) of a falling ellipse in the fluttering regime, $I^* = 0.16$, $\rho_b/\rho_f = 1.538$, $a/b = 1/5$ and $\nu = 0.01$.

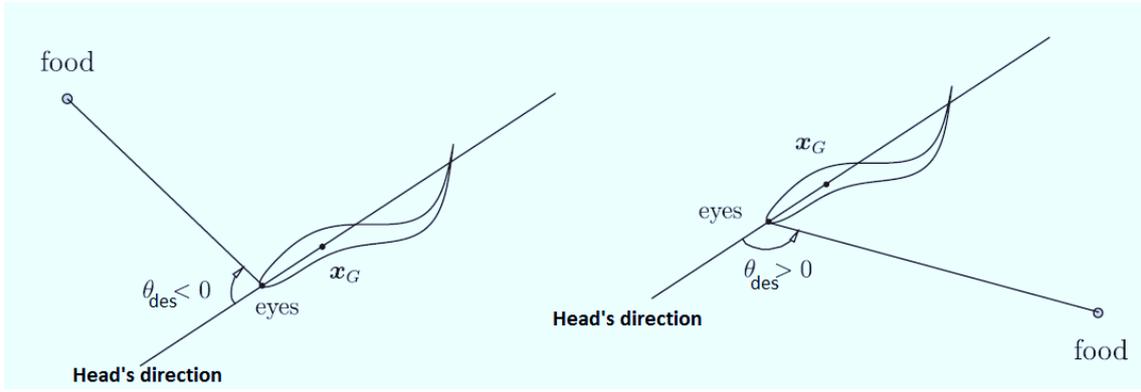


Figure 16: Schematic representation of desired angle for curvature control in rotation, $\theta_{des} = \theta_{food} - \theta_{Head}$ is the difference of the angles between head's direction and food's angle ($-\pi < \theta_{des} < \pi$), picture from Bergmann and Iollo [34] with a slight modification.

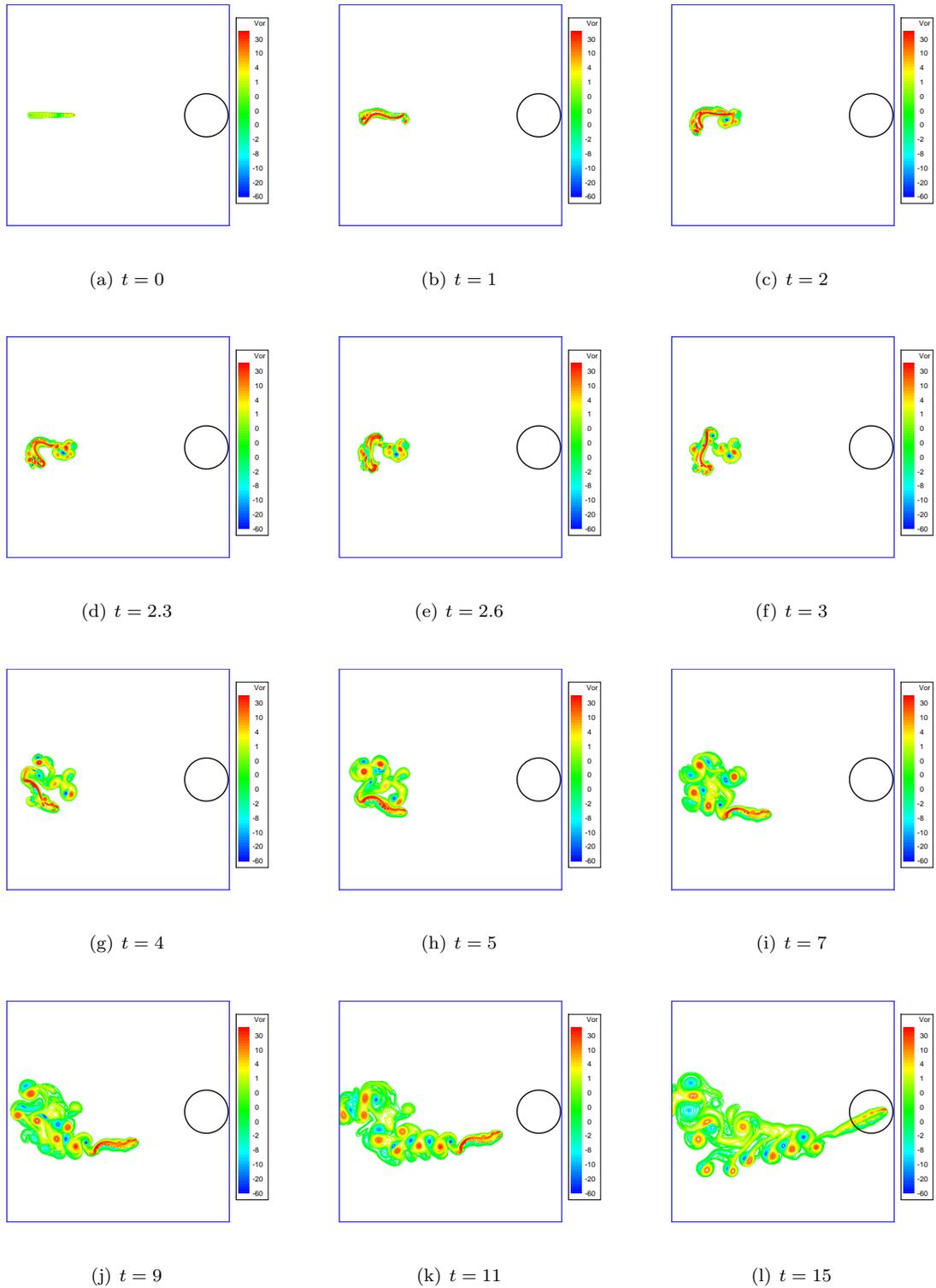


Figure 17: Snapshots of vorticity isolines obtained during a simulation of swimming fish for finding a food which is located at $(x_f, y_f) = (0.9L_x, 0.5L_y)$. At $t = 0$ the fish and the surrounding flow are in rest. After reaching the vicinity ($r = 0.5l_{\text{fish}}$) of the food the curvature of the backbone will tends to zero by Eq. (43). The domain of the solution is $(x, y) \in [0, 5l_{\text{fish}}] \times [0, 5l_{\text{fish}}]$, the resolution of the grid 1024×1024 and kinematic viscosity equal to $\nu = 1.4 \times 10^{-4}$.