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

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]

[a] *M2P2-UMR 7340, CNRS, Aix-Marseille Université, France*
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. By using a uniform Cartesian grid we benefit from the advantage of a new fourth-order direct solver for the solution of the Poisson equation to ensure the incompressibility constraint down to machine zero. For introducing a deformable body in fluid flow, an immersed boundary method is applied to the solution of the Navier-Stokes equations as a forcing term. A Lagrangian structure grid with prescribed motion cover the deformable body interacting with surrounding fluid due to hydrodynamic forces and moment calculated on an Eulerian reference Cartesian 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 control 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. At the same time bio-inspired design of swimming robots are in growth [21]. The costs of experimental studies 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, created by the backbone of the fish which movies from head to tail. The tail beat creates a reverse Kármán street of vortices and will push the fish forward, leaving a momentumless wake back. Efficient simulation of the incompressible flows is also an important problem, because the propagation of the perturbations with the sound speed in all directions in the incompressible media will lead to an elliptic equation. Thus the efficiency of the elliptic solver is crucial in dealing with the incompressible flow solvers. The third bottleneck in numerical simulations of fish-like swimming is the coupling of 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. [24] studied numerically the straight swimming/cruising and sharp turning manoeuvres of a two-dimensional carangiform fish. They showed the carangiform-like swimmer executing a sharp turn through an angle of 70^o from straight coasting within a space of about one body length. Bergman and Iollo [26] performed a numerical simulation of fish rotation control, looking for a food in a prescribed point. They add a radius to backbone of the fish to perform a change in direction by the considered fish which is constructed by a complex transform like Kutta-Joukowski. We will present a simple law for turning of a anguilliform fish. Our rotation control protocol is more or less similar to that presented in [24] and [26] which is based on the angle between the line-of-sight and the direction of advance. But instead of adding a radius to backbone we envisage to work with curvature which seems to be more efficient. We are using the method presented by Boyer et al. [16] 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 reformulate in terms of vorticity and stream-function which is more efficient. The choice of finite difference method (FDM) in this paper is related to the use of immersed boundary method in which a Cartesian grid can be used hence the

use of FDM is straight forward. Among finite difference methods high order compact methods [6] are more advantageous in terms of accuracy and reasonable cost. See [3]-[4] and [8] for a compact high-order discretization of vorticity and stream-function formulation of the Navier-Stokes equations and [14] and [27] for the incompressible Navier-Stokes equations in primitive variables. In the procedure of the solution to the incompressible Navier-Stokes equations an elliptic Poisson equation which is the most time consuming part of the algorithm will encountered frequently. Direct methods like diagonalization or iterative methods (e.g. multi-grid and Krylov subspace methods) can be used. With the use of high-order discretization iterative methods will be less attractive because their rate of convergence is slow. On the other hand in direct methods memory limitation is restrictive for simulations over fine grids. Therefore decoupling of the directions by FFT based methods is very advantageous, however this method will put 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 sine FFT. The main advantages of our method are fourth-order accuracy, efficiency, the possibility to parallelization and convergence down to zero machine. Other advantages and limitations of direct Poisson solver are discussed in the paper. Another bottleneck in numerical simulations of fish swimming is the analysis of fluid/solid interaction, which are divided to strong and loose coupling according to implicit or explicit time advancement, see [30] for a detail discussion. We are using volume penalization method which was proposed by Angot et al. [10] and belongs to the so called immersed boundary methods (IBMs). It consists of modeling solid bodies as porous media, thus getting ride of the Dirichlet boundary conditions by considering both the fluid and the solid as one domain with different permeabilities, so one has a rectangular domain in which the solid is immersed and can move. Penalization method will lead to first-order accuracy near the solid boundary and is one of the most efficient methods in dealing with deformable, moving and rotating bodies in a fluid. See the review of Mitta and Iaccarino [15] for a complete classification and description of the immersed boundary methods. In the present work we will focus on some physical aspects of efficient turning laws which is less studied by other researchers. The method of quaternions is adapted to backbone kinematics description. We combining compact finite differences with vorticity stream-function formulation of the Navier-Stokes equations including penalization term. An efficient direct method is presented to the solution of the Poisson equation. Thus different numerical aspects of algorithm like accuracy in time and space and the error introduced by penalization will be examined. The code is developed in FORTRAN and is accessible for all [31]. The paper is organized as follows. Fist our methodology including governing equations, discretization, kinematics of a fish like-swimming and the algorithm for fluid interaction with forced deformable bodies will be presented. Then validation of the algorithm will be done and errors convergence will be studied. Next the results for rotation control will be 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 the 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), if the Poisson equation (2) properly satisfied. We will encounter with two scalar quantity, i.e., ψ and ω, instead of 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, thus the coding is straightforward. 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:

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

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 being the stream function, satisfying a Poisson equation

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

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

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

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., $[LT^{-2}]$. Penalization parameter η is the porosity (permeability) coefficient of the immersed body with dimension $[T]$. The mask (characteristic) function χ is dimensionless and describes the geometry of the immersed body.

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

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

2.1.1. Spatial discretization

In the present investigation an explicit second-order and an implicit fourth-order compact finite difference methods will use for discretization of the spatial derivatives. The advantage of the compact methods over explicit finite differences is evident in terms of scaled modified wavenumber $w = k\Delta x$ which is demonstrated in Fig. 1. For a given periodic function $f(x) = e^{ikx}, x \in [0, 2\pi]$ with known analytical derivatives $f'(x) = ike^{ikx}$, $f''(x) = -k^2 e^{ikx}$. A numerical approximation of the derivatives at point x_j will 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 wavenumber is a measure of 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 second derivative the scaled modified wavenumbers are compared with analytical values given by Lele [6] for some explicit and implicit methods, a good agreement can be seen. It must be noted that the error in terms of 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 lowest truncation error. For example spectral like five-diagonal finite difference schemes designed by Lele [6] or Kim [19] 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 $(i = 1, ..., N)$ and $h = L_x/(N-1)$. Following Lele [6] a fourth-order approximation of the first and second derivatives are given by the classical Padé scheme:

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

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

for $i = 2, ..., N-1$, near the boundaries a third-order forward/backward stencil must be used, see [6]. A direct solver (LU-decomposition) will applied to the tri/pentadiagonal system of linear equations along each line. The arithmetics of tridiagonal implicit methods are three times of explicit methods and the arithmetics of solution to a pentadiagonal linear system is twice of the tridiagonal system.

2.1.2. Time integration

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

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

$$k_i(\omega) = -\partial_y\psi\ \partial_x\omega + \partial_x\psi\ \partial_y\omega + \nu\nabla^2\omega + \partial_x F_y - \partial_y F_x \tag{8}$$

where $i = 1, 2, 3, 4$. In each time step Eq. (8) must evaluated four times, in which Eq. (2) must be solved for updating of stream-function ($-\nabla^2\psi = \omega^*$). For details and technical discussions of Runge-Kutta methods see [7]. However, Δt is limited by CFL (Courant-Friedrich-Levy) condition which implies that

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

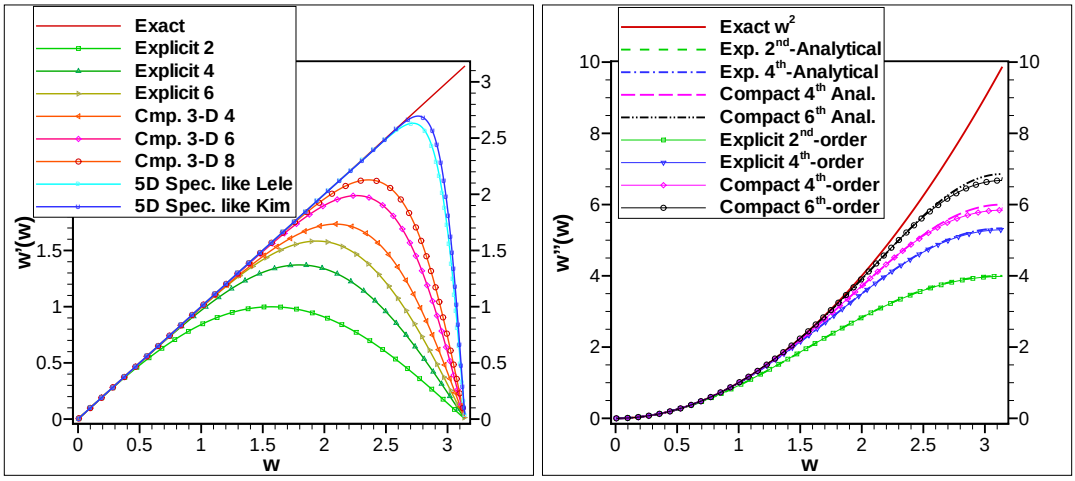
where U is an advection velocity (or a phase speed). In the presence of nonlinearity in space more attention must be payed. Viscous terms put a constraint of the form

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

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 maximum values of the scaled modified wave numbers for the first and second derivatives calculated via fourth-order Padé scheme plotted in Fig. 1. As can be seen $w'_{\max} \in [1, \pi]$ 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-step must be used. In the presence of moving bodies the displacement of the moving body must not exceeds the grid spacing, i.e., $\Delta t \leq \Delta\mathbf{x}/\mathbf{u}_B$, moreover with the use of explicit penalization method another constraint, $\Delta t \leq \eta$, must be respected. Among four above-mentioned constraints, smallest Δt must be chosen.

2.1.3. Fourth-order fast Poisson solver

In the procedure of the solution to the incompressible Navier-Stokes equations an elliptic Poisson equation which is the most time consuming part of the algorithm will encountered frequently. The common case is the pressure Poisson equation frequently used with homogeneous Neumann boundary conditions for the pressure correction in projection methods. Another example is in the vorticity stream-function formulation using equation (2) with Dirichlet boundary condition for vorticity and stream function. Free slip ($\omega = 0$) boundary condition in a close rectangular domain ($\psi = 0$, all around) can cover all the test cases studied in present investigation. In the presence of periodic boundary conditions, FFT based direct solvers can be used to efficiently solve the Poisson equation. Even if the flow is not periodic like most of the practical problems, in accordance with the boundary conditions for elliptic equation (homogeneous Dirichlet/Neumann) sine or cosine FFT can be used. We are presenting a new fourth-order solver for the Poisson equation (2) which is a combination of a compact finite difference with sine FFT. The advantages of our method are higher accuracy, compact tridiagonal stencil, less arithmetics, less memory usage in



(a) First derivative (b) Second derivative

Figure 1: Plots of scaled modified wavenumber for the first and second derivative, $w'(w)$ and $w''(w)$, versus scaled wavenumber $w = k\Delta x$ with the use of different central finite difference methods presented in Lele [6] and Kim [19].

comparison to iterative methods and straightforward parallelization because of decoupling of the operations in different directions. Near linear strong scaling (speedup) and efficiency is reported by Laizet et al. in [23] by introducing dual domain decomposition (or pencil) method, in which information along a line is accessible for a CPU by alternative decomposition of domain in three directions. The limitations (moreover the boundary conditions) will be the use of uniform grid in the direction in which the use of FFT is necessary. Usually when the solver of parabolic part is finite-difference, it is a custom to use a FDM discretization in one direction with out lost of efficiency (via direct tridiagonal solver), the advantage of this approach is the possibility of applying general boundary condition in that direction and using a refined mesh. The second-order version of this solver can be find in [11]. For a compact fourth-order collocated discretization of Poisson equation $-\nabla^2\psi = \omega$, over $N_x \times N_y$ grid points, by using Eq. (11)

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

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

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

because of the presence of Δx^2 factor behind fourth-order derivative, this term cannot be dropped and must be evaluated by second-order accuracy, therefor, the hole approximation scheme yield the fourth-order accuracy. Fourth-order derivative can be evaluated by using the original Poisson equation $-\nabla^2\psi = \omega$, and successive differentiating it 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

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

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

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

Usually the exact wavenumber will replace by modified wavenumber $k_y'^2$ which permits to evaluate the difference between the finite-difference and the spectral approximation of a second derivative [11]. For a fourth-order explicit finite-difference discretization, analytical relations for the scaled modified wavenumber of second derivative from [6], plotted in Fig. 1 (b) is as follow

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

The final tridiagonal system to be solved for the solution in Fourier space for each wavenumber of ψ in y direction is:

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

for $i = 2, ..., N_x - 1$, where $\beta = \Delta x^{-2} - k_y'^2/12$. In summary first a one-dimensional direct-FFT of the forcing function is performed in y direction, then for each line in x direction the above tri-diagonal system must be solved to find the solution ψ in wavenumber space next inverse-FFT of the solution must be performed. For the real data with zero value at the boundaries (homogeneous Dirichlet boundary condition), the natural Fourier transform to use is the *sine* transform, see [7]. The direction of FDM and FFT can be changed to consider no-slip BC in y direction. For taking into account inflow/outflow BC the mean flow must reduce from $\mathbf{u} = \mathbf{U}_\infty - \mathbf{U}$ in vorticity transport equation (1). This

is equivalent to move the grid with $\mathbf{U}_\infty$ and writing the Navier-Stokes equations in moving reference frame instead of Galilean inertial frame.

2.2. The algorithm of fluid interaction with a deformable body

2.2.1. Kinematics of the fish

The geometrically exact theory of nonlinear beams, is developed by Simo [5] and extended for fish vertebral by Boyer et al. [16]. 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 fish backbone in three dimensions for interested readers and future developments, but all the cases in this paper are limited to two-dimensions. Following Boyer et al. the kinematics of the backbone for Eel-like fishes can be determined by integration along arc-length ξ starting with head's situation as boundary condition. The variation of the orientation is obtained by

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

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. $M(\Omega)$ is a anti-symmetric tensor:

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

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

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

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 represent the rate of rotation (twist) of section around backbone along ξ direction. The rotation matrix in terms of the quaternions is given by

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

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

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

where $(.)$ represents time derivative, $(\vee)$ stands for anti-symmetric matrix constructed by a vector, e.g.,

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

where $\Gamma = (\gamma_1, \gamma_2, \gamma_3)^T$ represents local transversal shearing whose first component is the rate of stretching. The accelerations also can deduced from time derivative of Eq. (21). For more details see [16], [21] and [29]. For finding the velocities in the frame attached to the head from velocities V_G in Galilean reference frame,

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

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

2.2.2. Lagrangian structure grid

A symmetric shape is the first choice to start the parameterization of the swimmer's body. The geometry of a two-dimensional swimmer can be characterized by the half with $w(s)$ of the body along its arclength s (midline). Following the work of Kern and Koumoutsakos [17] and Carling et al. [9], the half width $w(s)$ is defined as

$$w(s) = \begin{cases} \sqrt{2 w_h s - s^2} & 0 \le s < s_b \\ w_h - (w_h - w_t)(\frac{s - s_b}{s_t - s_b})^2 & s_b \le s < s_t \\ w_t \frac{L - s}{L - s_t} & s_t \le s \le L \end{cases} \tag{24}$$

where L is the body length, $w_h = s_b = 0.04L$, $s_t = 0.95L$ and $w_t = 0.01L$ (see Fig. 2). In the mid mart of the fish a linear function can be used as in Gazzola et al. [25]. A structure grid will form by normal to backbone lines with thickness given by (24) and will cover the body. The velocity components of each point on Lagrangian grid V_{shape} with (I, J) indexes is given by

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

where V_{BN} and Ω_{BN} are the linear and angular velocities of the backbone, respectively, given by Eq. (21). Radius $(-w < |\mathbf{r}| < w)$ is aligned with the transversal lines of the structure grid normal to the backbone. See Fig. 3 for an example of the grid covering the fish after deformation and the corresponding velocities of each point. The information of the Lagrangian structure grid covering the deformable body must be transfered to Eulerian-Cartesian grid by interpolation for finding $\chi(i, j)$ and $\mathbf{u}_p(i, j)$. We are using a two-dimensional $f(x, y) = axy + bx + cy + d$ linear interpolation which will lead to a 4×4 linear system

to find the coefficients and will solved by direct method of Gauss-Jordan elimination from [7]. For each point of Eulerian grid in which $\chi = 1$ four nearest points of Lagrangian grid will be used. For some points in which $0 < \chi < 1$ due to mollifying by Eq. (26) automatically interpolation becomes extrapolation if that point is at outside of the original Lagrangian shape. All points in the interior of the fish will have $\chi(i,j) = 1$ on Eulerian grid. The mask is mollified by the Shuman [2] filter (26)

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

The interpolated smoothed mask function $\bar{\chi}$ (colored isolines) and the velocity components over the Eulerian grid are demonstrated on Fig. 4. The numbers 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. The resulted velocity field $\mathbf{u}_p$ is not divergence-free, see Gazzola et al. [25] for a complete theoretical, algorithmic and numerical discussion about this subject. We are not considering this issue in the present investigation.

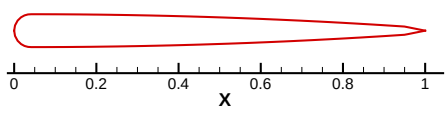


Figure 2: The shape of the fish given by Eq. (24) before deformation.

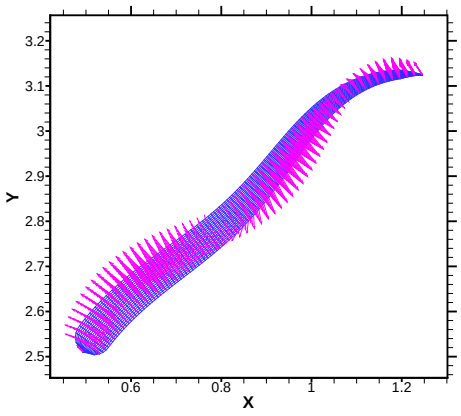


Figure 3: Lagrangian grid covering the fish after deformation and the corresponding velocities of each point.

2.2.3. Hydrodynamic coefficients evaluation

With the use of the penalization method the hydrodynamic forces and the moments acting on the obstacle, 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 computed readily by integrating the penalized velocity over

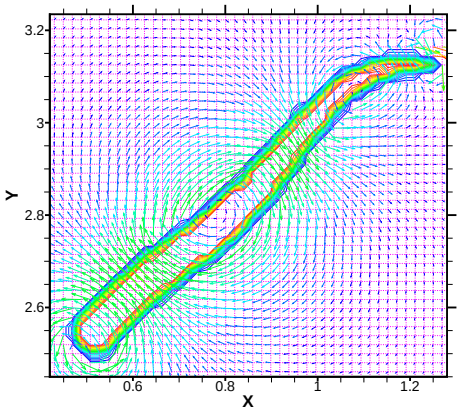


Figure 4: The interpolated smoothed mask function $\bar{\chi}$ (colored isolines) and the velocity components, colored by absolute velocity $\sqrt{u^2+v^2}$, over the Eulerian grid.

the considered volume (surface in two-dimensions), thus we have the force vector in Newton

$$\mathbf{F}^\star = \oint_{\partial\Omega_s} \sigma \cdot \mathbf{n}\,\mathrm{d}l = \lim_{\eta\to 0} \frac{\rho_f}{\eta} \int_{\Omega_s} \chi(\mathbf{u} - \mathbf{u}_B)\,\mathrm{d}s + \rho_f S_{pen} \ddot{\mathbf{X}}_{cg} \quad (27)$$

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

$$\mathbf{F} \approx \frac{1}{\eta S_{pen}} \int_{\Omega_s} \chi(\mathbf{u} - \mathbf{u}_B)\,\mathrm{d}s + \ddot{\mathbf{X}}_{cg} \quad (28)$$

The moment in two-dimensions is evaluated by

$$M_{\mathbf{x}_{cg}} = \oint_{\partial\Omega_s} (\mathbf{x} - \mathbf{x}_{cg}) \times \sigma \cdot \mathbf{n}\,\mathrm{d}l = \lim_{\eta\to 0} \frac{\rho_f}{\eta} \int_{\Omega_s} \chi(\mathbf{x} - \mathbf{x}_{cg}) \times (\mathbf{u} - \mathbf{u}_B)\,\mathrm{d}s + \frac{\rho_f}{\rho_s} J_{cg} \ddot{\theta}_{cg} \quad (29)$$

in $[N.m]$, where $J_{cg} = \int \mathbf{r}^2 dm$ is the moment of inertia taken about 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, dots denote derivation with respect to time and S_{pen} is the surface of penalized area.

2.2.4. Denoising of hydrodynamic coefficients

In dealing with fluid/solid interaction problems, the oscillation of the hydrodynamic forces and moment during successive iterations calculated from Eqs. (28) and (29) causes some trouble in correct prediction of the accelerations. The hydrodynamic forces and moments acting on the body are used in to calculation of the linear and angular accelerations which in turn has an impact on the predicted velocity and trajectory of the solid. The oscillations

are due to the nature of the penalization method, insufficient resolution, the approximative nature of the Eqs. (28) and (29). 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 [17] to denoise the hydrodynamic forces and moment. Simple exponential smoothing does not do well when there is a trend in the data. In such situations, several methods were devised like *second-order (double) exponential smoothing*

$$\hat{F}^n = \alpha F^n + (1-\alpha)(\hat{F}^{n-1} + b^{n-1}) \ , \ n = 3, 4, \dots \quad (30)$$

$$b^n = \beta(\hat{F}^n - \hat{F}^{n-1}) + (1-\beta)b^{n-1} \ , \ (\alpha, \beta) \in [0, 1] \quad (31)$$

where $\hat{F}^1 = F^1$, for $n = 2$ one can use Eqs (30) and (31) 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 $\delta = 10^{-3}$ doing well for denoising of the hydrodynamic forces, moments and centroid change of the deformable bodied. However $\delta = 10^{-3}$ has 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 simulations. A sensitivity analyses must be done for each test case (see also the discussion of the results in Section 3.3).

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 in two dimensions

$$\Sigma(\mathbf{F}_H + \mathbf{F}_G) = m\ddot{\mathbf{X}} \quad (32)$$

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

$$\Delta\mathbf{X}_{cg} = \frac{1}{2}\frac{\mathbf{F}^n}{m}\Delta t^2 + \mathbf{V}^n\Delta t \quad (33)$$

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

$$\Sigma M = \frac{d}{dt}(J_{cg}\dot{\theta}) \quad (34)$$

where M is the applied moments around the reference point. Time integration of Eq. (34) regardless of changes in moment of inertia and M will give the new angle of the object with respect to a given reference

$$\Delta\theta = \frac{1}{2}\ddot{\theta}^n\Delta t^2 + \dot{\theta}^n\Delta t \quad (35)$$

where $\ddot{\theta} = M/J$ and $\dot{\theta}^{n+1} = \dot{\theta}^n + \ddot{\theta}\Delta t$ (dotes denotes derivation with respect to time). Eqs (33) and (35) describing a motion with three degree of freedom (3DOF) for the considered body. In these equations second-order terms can be eliminated as is done in [25] but we are keeping this terms. Eqs. (28) and (29) provide the fluid forces and moment necessary to integrate the system of ODEs formed by Eqs. (32) and Eq. (34). Denoising of force and moment is done by (30). Appropriate initial conditions are necessary. In the present computations we use a first-order scheme for time integration of dynamics equations which seems to be adequate because of error introduced by penalization method which is near first-order. A same time integration method is used in [22] and [25] for dynamics of the body where penalization is used.

2.2.6. Summary of the algorithm

The summary of the algorithm for fluid/structure interaction is Listed in Algorithm 1. The flowchart is demonstrated in Fig. 5.

Algorithm 1 Fluid/structure interaction

1. Start from an initial condition
2. Body kinematics
   (a) Create eel's backbone by Eqs (17), (19) and (21)
   (b) Cover the shape by a Lagrangian structure grid & calcul 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 reference point
   (e) Compute body velocities $u_p(i,j), v_p(i,j)$ in Eulerian grid (Lagrange $\rightarrow$ Euler)
3. Time integration of flow field via RK4
   (a) $\omega_0^* = \omega^n$
   For i=1,2,3
   (b) Calcul $k_i(\omega_{i-1}^*)$ from Eq. (8)
   (c) $\omega_i^* = \omega^n + \alpha_i \ \Delta t \ k_i$
   (d) Solve Eq. (2); $-\nabla^2\psi = \omega_i^*$
   End For
   (e) Calcul $k_4(\omega_3^*)$ 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) Calcul the hydrodynamic coefficients from Eqs. (28) and (29)
   (b) Denoise the coefficients by Eq. (30)
   (c) Calcul displacements from Eq. (33)
   (d) Calcul rotation from Eq. (35)
5. Write necessary data to file
6. If $T < T_{end}$, Go to step 2
7. End

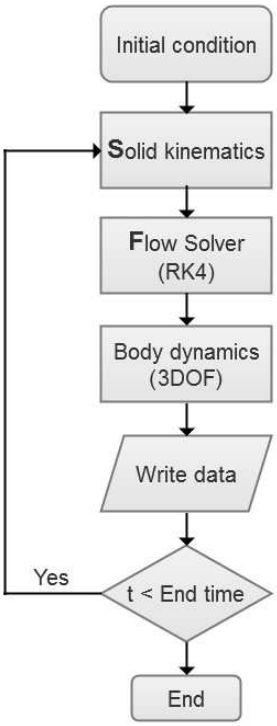


Figure 5: Flowchart of the fluid/solid interaction (FSI) algorithm.

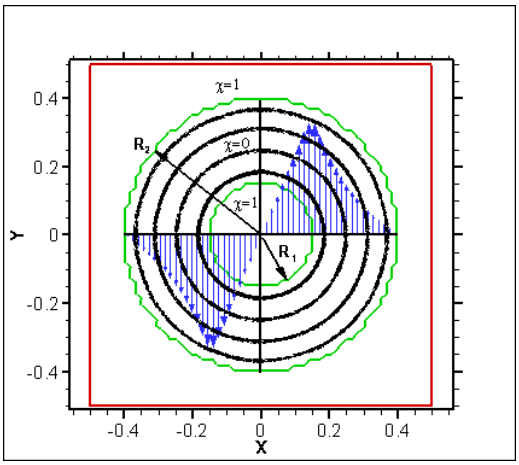


Figure 6: Schematic representation of penalized unit square domain for modeling of Taylor-Couette flow with volume penalization method ($\chi = 0$ represents the fluid domain and $\chi = 1$ the solid domains respectively). The radius of the inner cylinder is $R_1 = 0.15$ and that of outer cylinder is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

3. Validation

Validation of the algorithm is consist of convergence study in space using Taylor-Couette flow. For solid dynamics verification falling of an ellipse in quiescent fluid is considered. Then a test case of fish motion in forward gait will compared with results of Gazzola et al. [25].

3.1. Convergence of spatial error via Taylor-Couette flow

For a rigorous study of the error due to volume penalization term added to the Navier-Stokes equation in vorticity and stream function formulation an exact solution is necessary. 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 represented in Cartesian domains which can coincide with the underlying Cartesian grid used to discretize the governing equations. That is to say although the solver is adapted to a Cartesian domain the mask function which is the representative of penalized area is curved (see Fig. 6) as is in the practical test cases like flow around cylinder or complex geometries which will consider in following. An explicit second-order finite difference method and an implicit fourth-order compact finite difference is used for discretization of the governing equations including the curl of the penalization term $\nabla \times \mathbf{F}$. Taylor-Couette flow [1] consists of a viscous fluid confined between two concentric cylinders with radii $(R1, R2)$ 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 ($u_z = u_r = 0$). This state is known as circular Taylor-Couette flow and for which we have analytical solution. For reason of convenience the solution will represent in cylindrical coordinate, the azimuthal velocity is given by $u_\theta(r) = Ar + B/r$ 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 two cylinder 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. If the use of volume penalization method is desired, the velocity components must enforced in 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. 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. A unit square domain is considered as the solution domain, the time-step of 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 ($\Omega_2 = 0$). Taylor number for this configuration ($Ta = 0.64$) is below the

critical value, thus the flow is purely azimuthal. The runs 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 computed vorticity ω, stream-function ψ and velocity u with exact solutions with $N = 128$ grid point in each direction is plotted in Fig. 7 (a)-(b). The convergence of the L_1-error of u versus the grid resolution are shown for different penalization parameters in Fig. 7 (c). If u_η^{num} denote a numerical solution for 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

$$\|u^{\text{exact}} - u_\eta^{\text{num}}\| \leq \underbrace{\|u^{\text{exact}} - u_\eta\|}_{O(\sqrt{\eta})} + \underbrace{\|u_\eta - u_\eta^{\text{num}}\|}_{O(\Delta x^p)} \qquad (36)$$

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 $\|.\|$ 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 bound for the error $\|u^{\text{exact}} - u_\eta^{\text{num}}\| \leq O(\Delta x)$. The convergence of the L_1-error of u versus different penalization parameters are shown for different grid resolution in Fig. 7 (d), the convergence is shown to be of order $\sqrt{\eta}$. For these calculations the expected formal accuracy $p = 2$ is found and the convergence is between first and second order in space as a function of the resolution N, confirming the analysis by Carbou et al. in [13] and Morales et al. in [28]. We also observe a saturation of the convergence error for large N, corresponding to the penalization error. An optimal resolution can be found for each η. Decreasing η will lead to accuracy enhancement in general but for an explicit integration Δt is limited by η as discussed in Section 2.1.2. A fine grid will also need a small η as can be seen in the Fig. 7.

3.2. Validation of the solid dynamics via falling ellipse

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

$$I^* = 2I/(a^4\rho_f) = (a^2 + b^2)(b/2a^3)(\rho_s/\rho_f)$$

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

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

see [25]. Isolines of the vorticity corresponding to ellipse falling in fluttering regime are illustrated in Fig. 8 at different instants from $t = 0.2$ up to $t = 25$. The corresponding forces and velocity components are illustrated in Fig. 9. 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 and minor diameters of the ellipse. Resolution of the grid is 512×2049, moment of inertia around gravity center $J_{cg} = 0.25\pi ab(a^2+b^2)\rho_s$, initial position $(x_0, y_0) = (0.5l_x, l_y-3a)$ and initial angle of the major diameter with respect to horizon $\theta_0 = \pi/4$, density ratio $\rho_s = 1.538\rho_f$, filter parameter for hydrodynamic coefficients denoising $\delta = 0.001$, viscosity $\nu = 0.01$, gravity in y direction $g = -9.81\text{m/s}^2$, penalization parameter $\eta = 10^{-3}$. A qualitative agreement with the simulations of Kolomenskiy et al. [22] and Gazzoal et al. [25] can be seen.

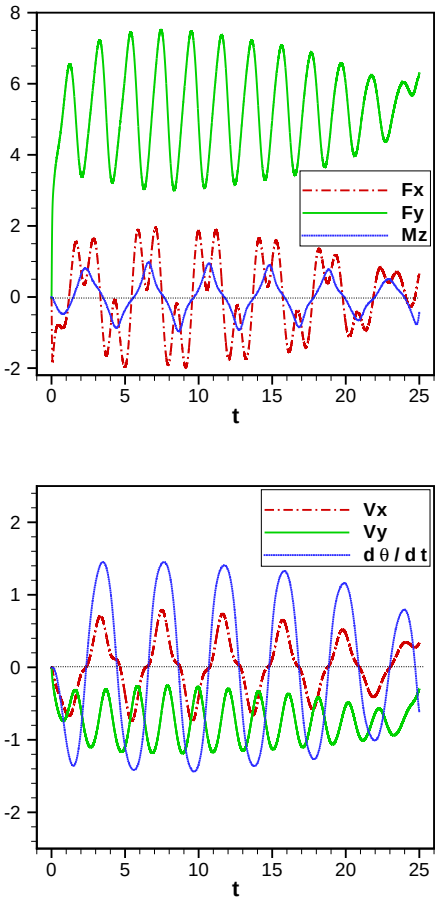
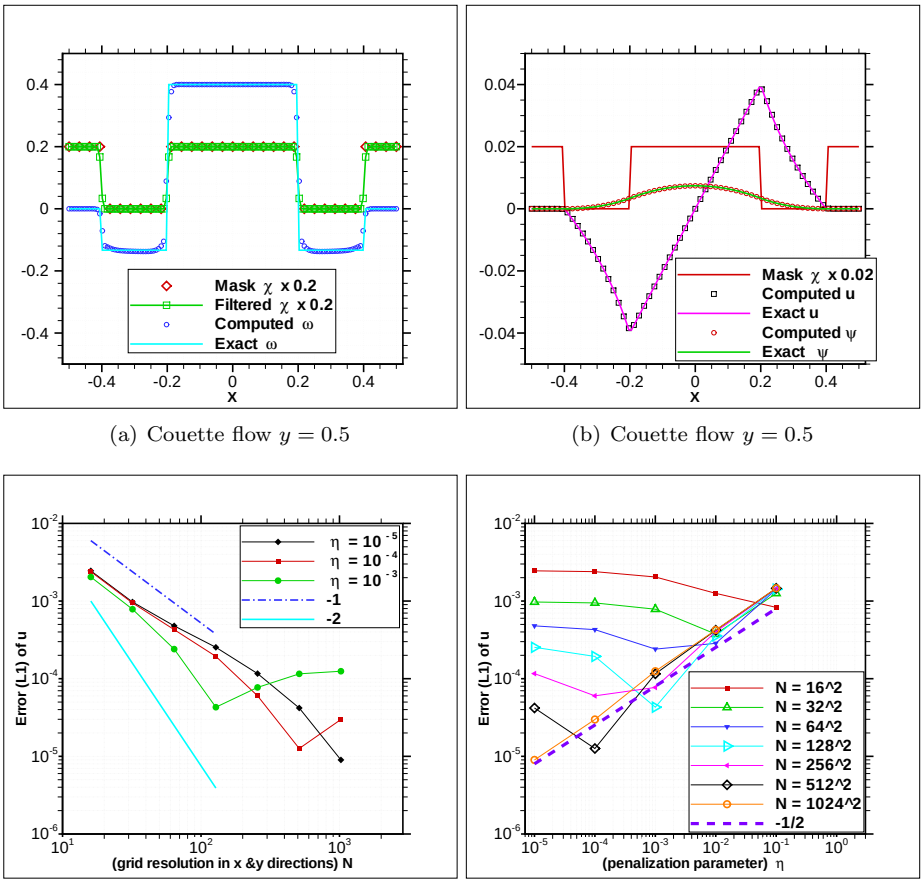


Figure 9: Forces and velocity components of ellipse falling in fluttering regime, $Re \approx 145$.

3.3. Fish in forward gait

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

$$y(x, t) = a(x)\sin(2\pi(x/\lambda + ft)) \qquad (37)$$



(a) Couette flow $y = 0.5$

(b) Couette flow $y = 0.5$

(c) Error convergence by resolution

(d) Error convergence by η

Figure 7: (a) Original and mollified Mask function, comparison of computed vorticity ω with exact solution for $N = 128$ grid point in each direction. (b) Comparison of computed stream-function ψ and velocity u with exact solutions for $N = 128$. (c) Convergence of the L_1-error of u with the spatial resolution (N being the grid resolution in each direction). (d) Convergence of the L_1-error of u with the penalization parameter η.

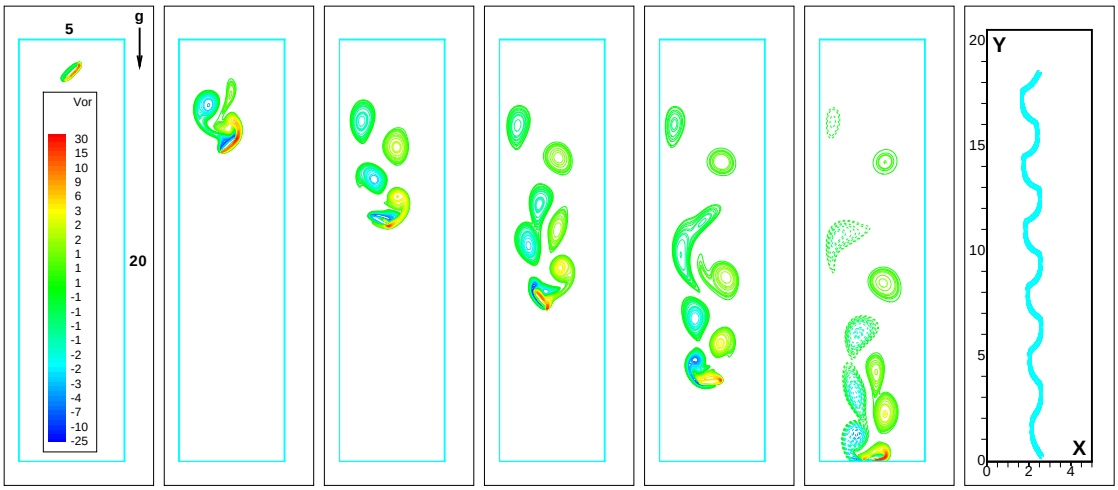
where λ is the wavelength, f is the frequency of the backbone and $a(x)$ is the envelope given by

$$a(x) = a_0 + a_1 x + a_2 x^2 \tag{38}$$

x is defined by inverting the arclength integral, i.e., $\Delta x = \Delta\xi/\sqrt{1 + (\partial y/\partial x)^2}$. Wavelength of the fish is defined in accordance with the geometry of the back bone in Cartesian system. We need the curvature of the backbone to be able to use geometrically exact theory of nonlinear beams. One must switch from the Cartesian system to the curvature, thus second derivative of Eq. (37) will lead to

$$\begin{aligned} k(\xi, t) = &(2a_2 - (2\pi/\lambda)^2 a(\xi)) \sin(2\pi(\xi/\lambda + ft)) \\ &+ (4\pi(a_1 + 2a_2\xi)/\lambda) \cos(2\pi(\xi/\lambda + ft)) \end{aligned} \tag{39}$$

where $a(\xi) = a_0 + a_1\xi + a_2\xi^2$. The parameters used by Kern and Koumoutsakos [17] 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., $\rho_s = \rho_f$. 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. [25] 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 $\eta = 10^{-4}$. We are performing our simulations on a rectangular domain $(x, y) \in [0, 10l_{\text{fish}}] \times [0, 5l_{\text{fish}}]$ by imposing penalization parameter in solid region equal to $\eta = 10^{-3}$ with resolution of 2049×1025 and 1025×513 and $\Delta t = 10^{-3}$. The centroid of the fish is initially positioned at $x_{\text{cg}} = 0.9L_x$ and $y_{\text{cg}} = L_y/2$ in our simulations. The forward velocities of the center of the mass computed with different methods are compared in Fig. 11. We impose two degree of liberty 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 displacement magnitude through a sinusoidal function (see Fig. 10) from zero to its designated value during the first period T in the reference simulations [17] and [25], but we are not considering for this and starting by a sudden movement given by Eq. (37), therefore a deviation from the reference solution can be seen in the the first period. This deviation will continue systematically until the asymptotic velocity



(a) $t = 0.2$ (b) $t = 5$ (c) $t = 10$ (d) $t = 15$ (e) $t = 20$ (f) $t = 25$ (g) Trajectocy

Figure 8: Ellipse falling in fluttering regime, $I^* = 0.16$, $\rho_s/\rho_f = 1.538$, $a/b = 1/5$ and $Re \approx 145$.

is reached at $t = 7$.

The reference simulation of Kern and Koumoutsakos [17] 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 [12]. The simulation of the Gazzola et al. [25] is done by a code based on vortex particle method coupled with Brinkman penalization which handle arbitrarily deforming bodies and specially their corresponding non-divergence free deformation velocity fields. A projection method is used by Gazzola et al. [25], the resulting Poisson equations for rotational (solenoidal) and potential (divergent) components of velocity fields are solved by an unbounded FFT base solver in Cartesian domain. Second and fourth order finite difference discretization is used for all other spatial derivatives. Time step is adapted by a Lagrangian CFL. The difference on the final velocities reported by Gazzola et al. [25] by taking into account the divergence of the velocity field inside the fish due to deformation of the fish is visible. Even though the average divergence over the fish volume is zero (i.e. volume is conserved) locally the deformation is not divergence free. We are not dealing with this issue in this paper because we are not interested in swimmers in which the thickness is high. In our simulations a grid independent simulation is obtained with 2048×1025 grid points. The difference of two simulations with 2049×1025 and 1025×513 grid points can be seen in Fig. 11. The forces are denoised by Eq. (30). 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 smaller amplitude in its oscillations. Filtering of the hydrodynamics coefficients is necessary to prevent the simulation from divergence and non-physical results. We are using second-order exponential filtering instead of first-order filtering used in [17], see Fig. 11. This process is like to adding a damper to the system but a correct value for δ must be chose for obtaining physical results via numerical tests. We propose values in the range of $\delta \in [0.01, 0.001]$ for fluid/solid interaction problems, however this can be depend also to the manner of non-dimensionalization of the forces. In Fig. 12 two snapshots of vorticity isolines obtained during a simulation (at $t = 1$ and $t = 9$) with the aforementioned parameters is demonstrated.

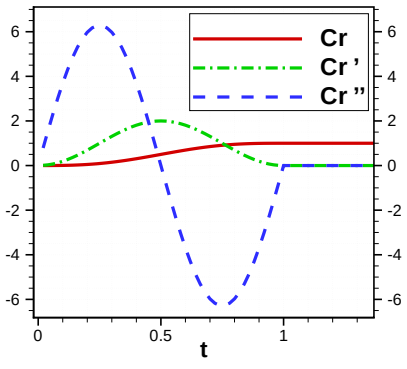


Figure 10: Slop function $Cr(t) = t' - \sin(2\pi t')/(2\pi)$, $t \in [t_i, t_f]$ with $t' = (t - t_i)/(t_f - t_i)$, $t_i = 0$, $t_f = 1$ for gradually starting the motion proposed by Boyer et al. [16]. At $t = 1$ the left-and right-hand limits are equal for the function Cr and its first Cr' and second Cr'' derivatives.

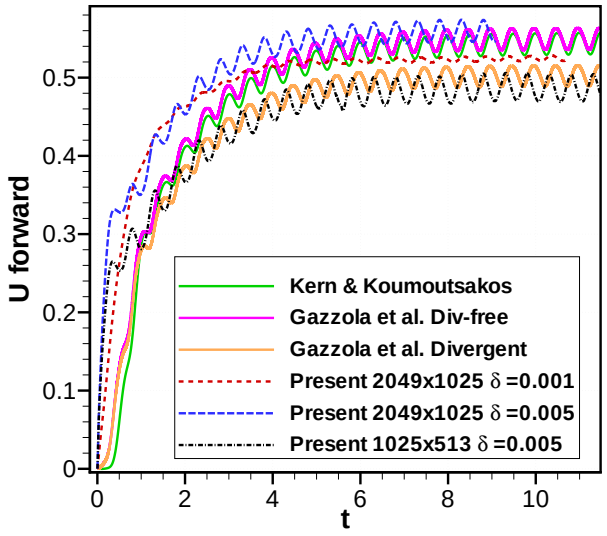
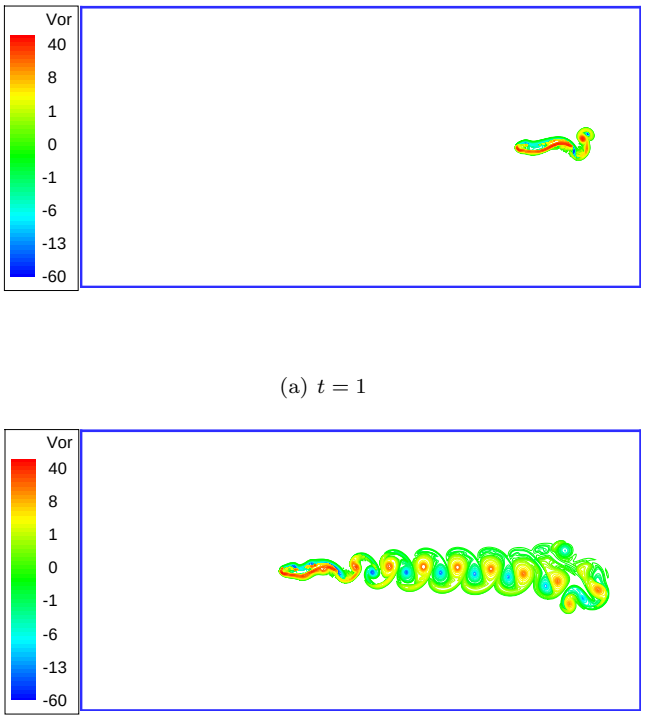


Figure 11: Anguilliform 2D swimmer's ($\lambda = f = 1$) forward velocity U. Solid lines indicate reference simulations performed by Kern and Koumoutsakos (green) [17] and Gazzola et al. (pink and brown) [25]. Dashed lines represent the results with the proposed algorithm.



(a) $t = 1$

(b) $t = 9$

Figure 12: The Snapshots of vorticity isolines obtained during a simulation where $(x, y) \in [0, 10l_{\text{fish}}] \times [0, 5l_{\text{fish}}]$ by imposing penalization parameter in solid region equal to $\eta = 10^{-3}$ with resolution of 2049×1025, Reynolds number approximately $Re \approx 3800$.

4. Application and results

Fish maneuvering law for tracking a fixed goal starting from rest is done by adding a constant curvature $k_{\text{des}}(\theta_{\text{des}}, t)$ all along the fish's backbone, $\xi \in [0, 1]$ to the propulsion mode, i.e., $k_3 = k(\xi) + k_{\text{offset}}$,

$$k_{\text{des}}(\theta_{\text{des}}) = \begin{cases} -\text{sgn}(\theta_{\text{des}})\ k_{\max} & |\theta_{\text{des}}| \geq \theta_{\text{limit}} \\ -\text{sgn}(\theta_{\text{des}})\ k_{\max}\ (\frac{\theta_{\text{des}}}{\theta_{\text{limit}}})^2 & \text{else} \end{cases} \tag{40}$$

where sgn represents the sign function, i.e., $\text{sgn}(\theta_{\text{des}}) = \theta_{\text{des}}/|\theta_{\text{des}}|$. We are using $k_{\max} = \pi$ which is equivalent to turning over a semicircle. For θ_{limit} we are using $\theta_{\text{limit}} = \pi/4$ as in Bergmann and Iollo [26]. See Fig. (13) for a schematic representation of θ_{des}. The change of the curvature must be gradually, i.e. $O(\Delta t)$. Time derivative of the curvature dk/dt is needed in Eq. (21) and can be calculated numerically. In each time step according to the position and direction of the head by considering the target a desired angle θ_{des} will be calculated. From Eq. (40) a desired curvature k_{des} must be calculated.

$$k^{n+1}_{\text{offset}}(k_{\text{des}}) = \begin{cases} k^n_{\text{offset}} + \Delta k & k < k_{\text{desired}} \\ k^n_{\text{offset}} - \Delta k & \text{else} \end{cases} \tag{41}$$

k_{offset} will added to the backbone for rotation control, where $\Delta k = \Delta t\, \pi/T$. To show the performance of the proposed method a test case of food finding with the above rotation protocol is performed. The domain size is $(x, y) \in [0, 5l_{\text{fish}}] \times [0, 5l_{\text{fish}}]$, the resolution is set to 1025×1025, the penalization parameter $\eta = 10^{-3}$, filter parameter $\delta = 0.005$, tail beat frequency $f = 1$, wavelength $\lambda = 1$, kinematic viscosity $\nu = 1.4 \times 10^{-4}$, initial position of the head $(x_0, y_0) = (0.1l_x, 0.5l_y)$ and initial angle of the head $\theta_0 = 0$. In Fig. 14 shows the snapshots of vorticity isolines obtained during a simulation of finding a food located at $(x_f, y_f) = (0.9l_x, 0.5l_y)$ by the quiescent fish. After reaching the vicinity ($r = 0.5l_{\text{fish}}$) of the food the propulsion mode will tends to zero with the mirror of the function presented in Fig. 10, i.e.,

$$C(t) = \frac{t_f - t}{t_f - t_i} + \frac{1}{2\pi} \sin(2\pi \frac{t - t_i}{t_f - t_i})\ ,\ t \in [t_i, t_f] \tag{42}$$

with $t_i = t_{\text{reached}}$, $t_f = t_{\text{reached}} + T$ for gradually stopping the curvature of the backbone during one period.

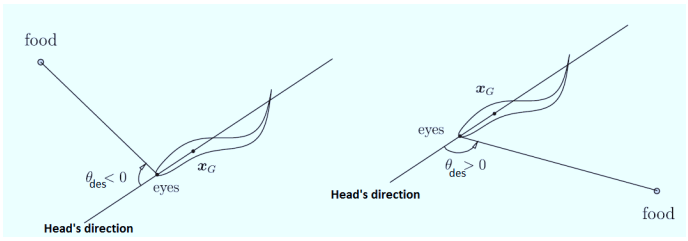


Figure 13: Schematic representation of desired angle for curvature control, $\theta_{\text{des}} = \theta_{\text{food}} - \theta_{\text{Head}}$ is the difference of the angles between head's direction and food's angle ($-\pi < \theta_{\text{des}} < \pi$), picture from Bergmann and Iollo [26] with a slight modification.

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, volume penalization method is applied to the solution of the Navier-Stokes equations as a forcing term. Even if penalization method is first-order in space, a big advantage of this method is that the evaluation of the hydrodynamic coefficients are straightforward. An efficient law for curvature control of an anguilliform swimmer looking for a food, based on geometrically exact theory of nonlinear beam is proposed. 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 guides for future developments is to use a high-order immersed boundary method, adding a multi-resolution analyses to the algorithm for grid adaptation, parallelization and extension to three dimensions. The code is developed in FORTRAN and is accessible for all [31].

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 Smouthing and filtering. Mon. Wea. Rev., vol. 85, 357-361, 1957.

[3] 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, 2, 3, 291-313, 1978.

[4] 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.

[5] J. C. Simo, A fnite strain beam formulation. The three-dimensional dynamic problem. Part I, Computer methods in applied mechanics and enginering, Vol. 49, 55-70, 1985.

[6] S. K. Lele, Compact finite difference schemes with spectral-like resolution. Journal of Computational Physics, Vol. 103, 16-42, 1992.

[7] H. Press, A. Teukolsky, T. Vetterling and P. Flannery, Numerical Recipes. Cambridge University Press, 1992.

[8] W. F. Spotz 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.

[9] 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.

[10] 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.

[11] P. Orlandi, Fluid flow phenomena: A numerical toolkit. Springer, 2000.

[12] 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. 182, 499-516, 2000.

[13] G. Carbou and P. Fabrie, Boundary layer for a penalization method for viscous incompressible flow. Advances in Differential Equations, 8:1453, 2003.

[14] S. Abide and S. Viazzo, A 2D compact fourth-order projection decomposition method. Journal of Computational Physics, Vol. 206, 252-276, 2005.

[15] R. Mittal and G. Iaccarino, Immersed boundary methods. Annual Reviews of Fluid Mechanics, Vol. 37, 239-261, 2005.

[16] 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.

[17] S. Kern and P. Koumoutsakos, Simulations of optimized anguilliform swimming. Journal of Experimental Biology, Vol. 209, 4841-4857, 2006.

[18] J. M. McDonough, Lectures in computational fluid dynamics of incompressible flow: Mathematics, algorithms and implementations. Departments of mechanical engineering and mathematics, University of Kentucky, 2007.

[19] J. W. Kim, Optimised boundary compact finite difference schemes for computational aeroacoustics. Journal of Computational Physics, Vol. 225, 995-1019, 2007.

[20] 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.

[21] M. El Rafei, M. Alamir, N. Marchand, M. Porez and F. Boyer, Multi-variable constrained control approach for a three-dimensional Eel-like robot. Published in IEEE/RSJ 2008 International Conference on Intelligent RObots and Systems, IROS, Nice, France, 2008.

[22] 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.

[23] 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.

[24] 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.

[25] 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.

[26] M. Bergmann and A. Iollo, Modeling and simulation of fish-like swimming. Journal of Computational Physics, Vol. 230, 329-348, 2011.

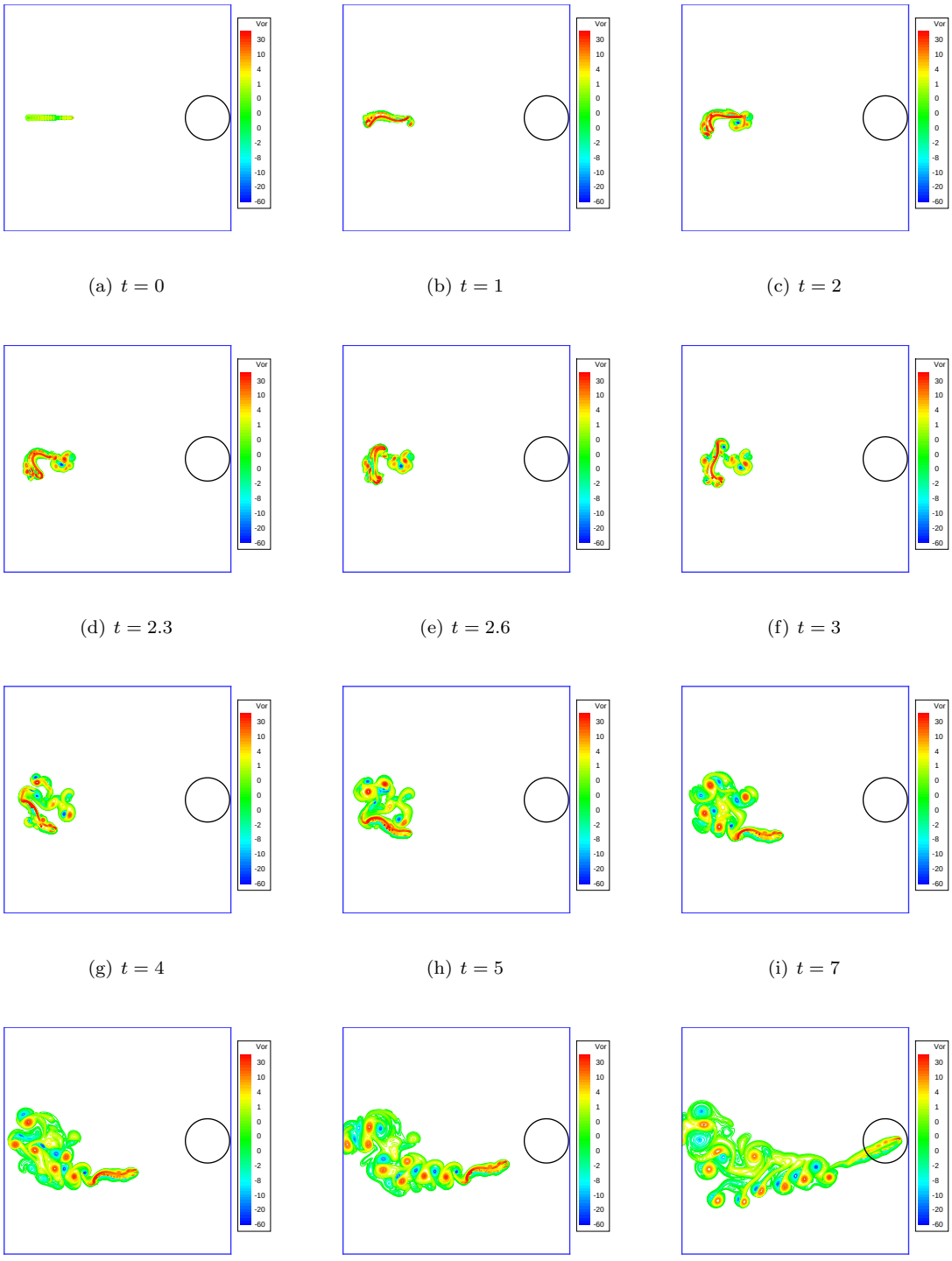
[27] B.J. Boersma, A 6th order staggered compact finite difference method for the incompressible Navier-Stokes and scalar transport equations. Journal of Computational Physics, Vol. 230, Issue 12, 4940-4954, 2011.

[28] 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.

[29] A. Belkhiri, 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.

[30] F. Sotiropoulos and X. Yang, Immersed boundary methods for simulating fluid-structure interaction. Progress in Aerospace Sciences, Vol. 65, 1-21, 2014.

[31] *The code is developped in FORTRAN and is accessible for all by sending a mail to*: ghaffari@L3m.univ-mrs.fr or s.amin.ghaffary@gmail.com



(a) $t = 0$ (b) $t = 1$ (c) $t = 2$

(d) $t = 2.3$ (e) $t = 2.6$ (f) $t = 3$

(g) $t = 4$ (h) $t = 5$ (i) $t = 7$

(j) $t = 9$ (k) $t = 11$ (l) $t = 15$

Figure 14: The Snapshots of vorticity isolines obtained during a simulation of finding a food located at $(x_f, y_f) = (0.9l_x, 0.5l_y)$ by the quiescent fish where $(x, y) \in [0, 5l_{\text{fish}}] \times [0, 5l_{\text{fish}}]$ with resolution of 1025×1025 and $\nu = 1.4 \times 10^{-4}$.