Journal of Computational Physics 230 (2011) 7093–7114

Contents lists available at ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

Simulations of single and multiple swimmers with non-divergence free deforming geometries

Mattia Gazzola [a], Philippe Chatelain [b], Wim M. van Rees [a], Petros Koumoutsakos [a,*]

[a] *Institute of Computational Science, ETH Zurich, CH-8092, Switzerland*
[b] *Institute of Mechanics, Materials and Civil Engineering, Université catholique de Louvain, B-1348, Belgium*

ARTICLE INFO

Article history:
Received 15 October 2010
Received in revised form 8 March 2011
Accepted 20 April 2011
Available online 7 May 2011

Keywords:
Vortex methods
Penalization
Fluid–structure interaction
Deforming body
Swimmer

ABSTRACT

We present a vortex particle method coupled with a penalization technique to simulate single and multiple swimmers in an incompressible, viscous flow in two and three dimensions. The proposed algorithm can handle arbitrarily deforming bodies and their corresponding non-divergence free deformation velocity fields. The method is validated on a number of benchmark problems with stationary and moving boundaries. Results include flows of tumbling objects and single and multiple self-propelled swimmers.

1. Introduction

The simulation and quantification of biological locomotion in fluids has a wide range of engineering applications [1] and presents a series of challenges for the development of suitable numerical methods. Simulations can greatly assist in characterizing aquatic propulsion [2–9], the high energy efficiency observed in many species and the complexity of the mechanisms involved and of the flow structures generated. In recent years the simulation of self-propelled fish-like body immersed in a viscous fluid has been tackled with different methods. Starting from the work of Carling et al. [17], to simulate 2D or 3D swimmers, were used finite differences [17,18], Lagrangian multipliers [4], viscous vortex methods [19], block lower-upper symmetric Gauss-Seidel [20], Singular-Value Decomposition based Generalized Finite Difference (SVD-GFD) [21], finite volumes [22,23] or remeshed Smoothed Particle Hydrodynamics [24]. Swimming organisms have also been investigated via biological observation [10–12] and via robot-replicas [13–15]. A particular type of locomotion considered archetypical is anguilliform swimming, which has been one of the first modes of swimming studied in the legendary experiments of Gray [16]. Anguilliform swimming is achieved through the propagation of curvature waves from the anterior to the posterior of the body and is characterized by high flexibility, resulting in acceleration burst [25], fast or efficient swimming [22]. In the present work we will consider this swimming pattern.

The computational investigation of biological locomotion in fluids entails the accurate capturing of the interaction of the flow with complex and deforming geometries, hence a need for accurate and efficient numerical solvers. Such flow-structure interaction (FSI) problems have been largely handled by means of unstructured grids (ALE – Arbitrary Lagrangian Eulerian). In ALE a moving fitted mesh tracks the solid boundaries and the governing equations for solid and fluid are solved separately, enforcing continuity conditions at the interface [26,23]. Such methods are accurate, but may result in inefficient

* Corresponding author. Tel.: +41 1 632 5258; fax: +41 1 632 1703.
E-mail address: petros@ethz.ch (P. Koumoutsakos).

0021-9991/$ - see front matter © 2011 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2011.04.025

implementation due to the need to adapt the computational elements to unsteady geometries and to resolve accurately the flow structures. Another class of methods considers the system of the body and the fluid simultaneously, where the solid interface is captured implicitly while boundary conditions are enforced by adding forcing terms in the continuous or discrete forms of the equations [27]. This approach has been successfully applied to rigid bodies in fluids, by means of Lagrangian Multipliers [28], immersed interface methods [29,30], immersed boundary methods [31,27,32,33] and Brinkman penalization methods [34,35]. Penalization schemes have been used in one way fluid–solid coupling problems together with spectral and vortex methods [36,37] or adaptive wavelet collocation methods [38]. The two way fluid–solid coupling problem has been tackled in [26,39–42,8] and, in the context of vortex methods, by Coquerelle and Cottet [43], through a Brinkman penalization approach, and by Shiels et al. [44] and Eldredge [45], through direct integration of Newton's equations.

In this work we extend the works of Coquerelle and Cottet [43] and Rossinelli et al. [37], and propose an unbounded vortex particle method for fluid–structure interaction problems with deforming bodies immersed in a viscous incompressible flow.

Coquerelle and Cottet [43] combined vortex methods, Brinkman penalization and projection approach, to deal with full fluid–structure interaction problems limited to rigid 2D and 3D geometries. Their method is inspired by both the fictitious domain approach [28] and projection method [46]. The body velocity is recovered in a self-consistent way from the fluid motion inside the domain captured by a level set function. A numerical analysis of this method in the velocity pressure formulation is given in [41].

Rossinelli et al. [37] presented a 2D vortex method coupled with Brinkman penalization for incompressible, viscous flows, able to capture the interaction from the body to the fluid (one way coupling) and they applied it to study flow past the cylinder at different Reynolds numbers.

Our methodology is a superset of these two works [43,37]. It is a generalization to objects with non-zero divergence deformation velocity fields which relies on the treatment of the dilatational part of the extended flow field and on the use of unbounded solvers.

These components allow several significant advances. They enable the simulation of full fluid–structure interaction problems for arbitrary complex unsteady geometries.

They exploit the localization of the physics, i.e. the compact support of vorticity and dilatation, and provide a natural treatment for far-field boundary conditions [47–49].

Finally, they make the treatment of objects with net volume changes possible, a feat unreachable in [43] because of the boundary conditions imposed by their Poisson solvers and the incompressibility constraint enforced over the entire domain.

The numerical scheme produces a simple algorithm, developed in Section 2. We assess its capabilities through the simulation of benchmark problems such as sedimentation in Section 3. Its application to self-propelled swimmers is presented in Section 4, demonstrating its suitability for the study of biological locomotion. We mention that these investigations are complemented with exhaustive convergence tests.

The exposition of the methodology and the applications mentioned above use a two-dimensional setting. We show that the approach can be readily extended to three dimensions in Section 4.3 where we consider the 3D self-propelled swimmer of [22].

2. Methodology

We present a vortex method to simulate the two-dimensional interaction of a viscous incompressible flow with deformable bodies. The velocity–vorticity form of the Navier Stokes equations is coupled with the Brinkman penalization to enforce the no-slip condition at the body–fluid interface. The exchange of linear and angular momenta between the fluid to the body is achieved through a projection method approach. The deformable body interface is captured by a level set and its time dependent deformation is defined *a priori* in a parametrized form.

2.1. Remeshed vortex methods

We consider a two dimensional incompressible flow governed by the Navier–Stokes equations in their vorticity formulation:

$$\frac{\partial \omega}{\partial t} + \nabla \cdot (\mathbf{u}\omega) = \nu \nabla^2 \omega \tag{1}$$

$$\nabla \cdot \mathbf{u} = 0 \tag{2}$$

where $\omega = \nabla \times \mathbf{u}$ is vorticity and ν is the kinematic viscosity. Vortex methods discretize the vorticity field ω by means of particles, characterized by their position $\mathbf{x}_p$, volume V_p and strength $\Gamma_p = \int_{V_p} \omega d\mathbf{x}$. Particles move according to the velocity field $\mathbf{u}$ and their strengths evolve in accordance with the diffusion term. In order to avoid Lagrangian distortion, a remeshing approach is used: particle strengths and locations are interpolated through a high order moment conserving kernel onto an underlying regular grid at the end of each step [50,47,51]. In this work we used the third order accurate interpolation kernel

M'_4 [52]. The presence of an underlying grid allows the use of fast grid differential operators and efficient grid based Poisson solvers, necessary to recover the velocity field from the Poisson equation

$$\nabla^2\mathbf{u} = -\nabla \times \omega. \tag{3}$$

In the present work, we employ a Fourier-based Poisson solver for unbounded domains as described in [53,54]. Finally, vortex methods are characterized by their relaxed stability properties for convection, accuracy and localization of the computational regions [47–49,55].

2.2. Geometry representation

In this work, the implicit definition of the geometry and its associated grid, is decoupled from the solver. Each shape is embedded in a grid $\mathcal{M}$ that could be stationary or deformable depending on the specific application (Fig. 1). At the node locations $\mathbf{r}_\mathcal{M}$ the deformation velocity $\dot{\mathbf{r}}_\mathcal{M}$, which describes the body motion, and the characteristic function $\chi_\mathcal{M}$ $(0 \leqslant \chi_\mathcal{M} \leqslant 1)$, which localizes the geometry, are computed. Here the mollified characteristic function is evaluated as follows:

$$\chi_\mathcal{M} = \begin{cases} 0 & d < -\epsilon \\ \frac{1}{2}\left[1 + \frac{d}{\epsilon} + \frac{1}{\pi}\sin\left(\pi\frac{d}{\epsilon}\right)\right] & |d| \leqslant \epsilon \\ 1 & d > \epsilon \end{cases}$$

where d is the signed distance from the surface of the body (negative outside, positive inside) and ϵ is the mollification length. As a general rule, for moderate Reynolds numbers (Re < 1e^4), ϵ should be a small fraction (<1%) of the characteristic length of the geometry under study. In effect, it defines the curvature of the finest resolved features of the object. It should also allow the mollified characteristic function to span 4–5 grid points in order to have a numerically stable surface normal.

When required, the separately defined mesh $\mathcal{M}$ is bilinearly interpolated onto the Cartesian computational grid $\mathcal{C}$ (Fig. 1) used by the flow solver, according to

$$\chi_s = \mathcal{P}_l(\mathbf{x}, \mathcal{R}(\theta)\mathbf{r}_\mathcal{M} + \mathbf{x}_{cm}, \chi_\mathcal{M}) \tag{4}$$

$$\mathbf{u}_{DEF} = \mathcal{P}_l(\mathbf{x}, \mathcal{R}(\theta)\mathbf{r}_\mathcal{M} + \mathbf{x}_{cm}, \mathcal{R}(\theta)\dot{\mathbf{r}}_\mathcal{M}) \tag{5}$$

where $\mathcal{P}_l(\mathbf{b}, \mathbf{a}, \boldsymbol{\alpha})$ is the bilinear interpolant of the quantity $\boldsymbol{\alpha}$ stored at grid nodes locations $\mathbf{a}$ onto the grid $\mathbf{b}$, $\mathbf{x}$ are the solver computational grid nodes positions, $\mathbf{x}_{cm}$ and θ are, respectively, the coordinates of the body center of mass and the rotation angle in the simulation frame of reference, while $\mathcal{R}$ is the 2D counterclockwise rotation matrix. For further details, we refer to Appendix A.

2.3. Penalization model and projection approach

The system to be simulated consists of the deforming body and the surrounding viscous, incompressible flow. Let ρ_f and ν denote the fluid density and kinematic viscosity, respectively, and ρ_s the body density, which varies according to the volume so as to conserve mass

$$\rho_s(t) = \rho_{s,0}\frac{V_{s,0}}{V_s(t)} \tag{6}$$

where $\rho_{s,0}$ and $V_{s,0}$ are, respectively, initial density and solid volume. The flow around the body is described by the two-dimensional incompressible Navier–Stokes equations, here written in the velocity ($\mathbf{u}$) and pressure (p) formulation

$$\frac{\partial\mathbf{u}}{\partial t} + (\mathbf{u}\cdot\nabla)\mathbf{u} = -\frac{1}{\rho_f}\nabla p + \nu\nabla^2\mathbf{u} + \mathbf{g}, \quad \mathbf{x} \in \Sigma \setminus \Omega \tag{7}$$

$$\nabla\cdot\mathbf{u} = 0, \quad \mathbf{x} \in \Sigma \setminus \Omega \tag{8}$$

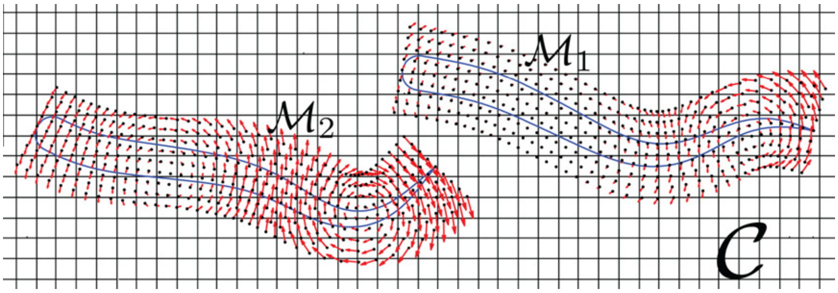

Fig. 1. Schematic of the use of the grids. Here two independents shapes are embedded in their corresponding deformable grids $\mathcal{M}_1$ and $\mathcal{M}_2$, that carry the values of deformation velocity $\dot{\mathbf{r}}_{\mathcal{M}_1}$, $\dot{\mathbf{r}}_{\mathcal{M}_2}$ and characteristic function $\chi_{\mathcal{M}_1}$, $\chi_{\mathcal{M}_2}$ to be bilinearly interpolated onto the Cartesian computational grid $\mathcal{C}$.

where $\mathbf{g}$ is the gravity, Σ is the computational domain and Ω is the support of the solid. The action of the body on the fluid is realized through the no-slip boundary condition at the interface $\partial\Omega$, enforcing the body velocity ($\mathbf{u}_s$) to be the same as the fluid ($\mathbf{u}$) through $\mathbf{u}_s = \mathbf{u}$. The feedback from the fluid to the body is in turn described by Newton's equation of motion

$$M_s\ddot{\mathbf{x}}_s = \mathbf{F}, \quad \frac{d(J_s\boldsymbol{\omega}_s)}{dt} = \tau \tag{9}$$

where M_s is the total mass of the body, J_s its moment of inertia and $\mathbf{F}$ and τ are the hydrodynamic force and momentum, respectively, exerted by the fluid on the body.

We present the coupling of the vortex method (Section 2.1) with the Brinkman penalization technique in order to capture the interaction between the fluid and the body, and to handle flows around complex shapes. Following the approach of [43], we enforce self-consistent rigid motions inside the bodies and extend the method to the case of arbitrary unsteady geometries characterized by non-divergence-free deformation velocity fields.

In the penalization model, the fluid velocity is extended inside the body and the Navier–Stokes equations are solved including an extra term (the penalization term) to model no-slip boundary conditions

$$\frac{\partial\mathbf{u}_\lambda}{\partial t} + (\mathbf{u}_\lambda\cdot\nabla)\mathbf{u}_\lambda = -\frac{1}{\rho}\nabla p_\lambda + \nu\nabla^2\mathbf{u}_\lambda + \mathbf{g} + \lambda\chi_s(\mathbf{u}_s - \mathbf{u}_\lambda), \quad \mathbf{x}\in\Sigma \tag{10}$$

$$\nabla\cdot\mathbf{u}_\lambda = 0, \quad \mathbf{x}\in\Sigma \tag{11}$$

where $\lambda \gg 1$ is the penalization factor, which we can relate to the effective porosity of the body. Importantly, as proved in [56], the error in the penalized solution is bounded by $\|\mathbf{u}-\mathbf{u}_\lambda\| \leqslant C\lambda^{-1/2}\|\mathbf{u}\|$. Furthermore the penalization factor λ is arbitrary and therefore the error can be precisely controlled, enforcing the no-slip conditions to any accuracy degree by setting λ appropriately. We note that throughout the paper, we used a value of lambda $\lambda = 1e^4$, which we found to be a good compromise between accuracy and computational costs (a higher value of λ implies a finer time discretization). We cannot exclude that different problems, characterized by higher Reynolds number, would require a different value of λ. Hereafter in this section we will denote $\mathbf{u}_\lambda$ by $\mathbf{u}$ to simplify the notation. The viscosity ν will be taken constant within the whole domain and the density ρ, which appears in the baroclinic term of Eq. (10), is computed by combining fluid and solid densities through the characteristic function χ_s

$$\rho = (1-\chi_s)\rho_f + \chi_s\rho_s. \tag{12}$$

Taking the curl of the velocity we obtain from (10) the vorticity formulation

$$\frac{\partial\omega}{\partial t} + \nabla\cdot(\mathbf{u}\omega) = (\omega\cdot\nabla)\mathbf{u} + \nu\nabla^2\omega + \frac{1}{\rho^2}\nabla\rho\times\nabla p + \lambda\nabla\times\chi_s(\mathbf{u}_s-\mathbf{u}). \tag{13}$$

Expanding the baroclinic term as a function of the velocity and considering that the stretching term $(\omega\cdot\nabla)\mathbf{u}$ vanishes in two dimensions, we can rewrite the vorticity equation as

$$\frac{\partial\omega}{\partial t} + \nabla\cdot(\mathbf{u}\omega) = \nu\nabla^2\omega - \frac{\nabla\rho}{\rho}\times\left(\frac{\partial\mathbf{u}}{\partial t} + (\mathbf{u}\cdot\nabla)\mathbf{u} - \nu\nabla^2\mathbf{u} - \mathbf{g}\right) + \lambda\nabla\times\chi_s(\mathbf{u}_s-\mathbf{u}). \tag{14}$$

We note a difference in the computation of the pressure gradient when compared to the work of [43]: it includes the viscous flux of vorticity $\nabla^2\mathbf{u} = -\nabla\times\omega$ at the solid boundary. This quantity is important at incepting boundary layers and for unsteady flow conditions.

The velocity of the body $\mathbf{u}_s$ is decomposed into three terms, i.e. its rigid translational velocity $\mathbf{u}_T$, rigid rotational velocity field $\mathbf{u}_R$ and its free deformation velocity field $\mathbf{u}_{DEF}$

$$\mathbf{u}_s = \mathbf{u}_T + \mathbf{u}_R + \mathbf{u}_{DEF} \tag{15}$$

The velocities $\mathbf{u}_T$ and $\mathbf{u}_R$ are the result of the hydrodynamic forces and momentum acting on the body, while $\mathbf{u}_{DEF}$ is prescribed *a priori*, varies in time and is specific to the shape under study. As this extended velocity field can be non-solenoidal inside the object, the continuity equation of Eq. (11) must be modified into

$$\nabla\cdot\mathbf{u} = 0, \quad \mathbf{x}\in\Sigma\setminus\Omega \tag{16}$$

$$\nabla\cdot\mathbf{u} = \nabla\cdot\mathbf{u}_{DEF}, \quad \mathbf{x}\in\Omega \tag{17}$$

An additional consequence is the contribution of this dilatation source to the Helmholtz decomposition of the global velocity field (Section 2.4).

Rigid translation ($\mathbf{u}_T$) and rotational ($\mathbf{u}_R$) velocities are determined via a projection approach. During a part of the time-step, the flow is then allowed to evolve freely over the entire domain, without any forcing, also inside the body as if it was not there.

The resulting extended flow field violates the rigid motion (or prescribed deformation) of the immersed objects. Crucially, this step enables the computation of the transfer of linear and angular momentum over the time step. In effect, the momenta integrals due to this extended flow inside the immersed object correctly capture the fluxes at the boundary over the time step. We refer to [46] for a detailed proof.

Finally, we note that this fluid–structure coupling is achieved without the explicit use of the forces and torques (Eq. (9)).

2.4. Discretization method

Our system is discretized by means of a hybrid particle-mesh method coupled with a level set, within a computational domain Σ that is a Cartesian grid of uniform spacing h.

The grid is used to evaluate differential operators, to compute velocities (Poisson solver [53,54]) and to initialize and remesh vortex particles. In this work all spatial derivatives are computed with second order centered finite difference schemes, in the 2D case, while we used fourth order centered finite difference schemes in the 3D case. The time step is adapted according to a Lagrangian CFL condition (LCFL)

$$\|\nabla \mathbf{u}^{n-1}\|\Delta t^n \leqslant LCFL \tag{18}$$

where the *LCFL* constant controls the allowed distortion of the particles over a time step [51], and $t^n = \sum_{i=1}^{n} \Delta t^i$ for $n \geqslant 0$.

We describe one time step of the proposed algorithm, from t^n to t^{n+1}, assuming that all quantities are known up to t^n. We denote the characteristic function, the velocity, vorticity, divergence, and density fields at time t^n by χ_s^n, $\mathbf{u}^n$, ω^n, σ^n, and ρ^n, respectively. The deformation of the body immersed in the fluid is imposed at all times and is specific to the shape under study, therefore the quantities $\mathbf{r}_{\mathcal{M}}(t),\ \dot{\mathbf{r}}_{\mathcal{M}}(t),\ \chi_{\mathcal{M}}(t)$ are known a priori.

Algorithm 1: General Method

$$\chi_s^n = \mathcal{P}_I\left(\mathbf{x}, \mathcal{R}(\theta^n)\mathbf{r}_{\mathcal{M}}^n + \mathbf{x}_{cm}^n, \chi_{\mathcal{M}}^n\right) \tag{19}$$

$$\mathbf{u}_{DEF}^n = \mathcal{P}_I(\mathbf{x}, \mathcal{R}(\theta^n)\mathbf{r}_{\mathcal{M}}^n + \mathbf{x}_{cm}^n, \mathcal{R}(\theta^n)\dot{\mathbf{r}}_{\mathcal{M}}^n) \tag{20}$$

$$\sigma^n = \chi_s^n\left(\nabla \cdot \mathbf{u}_{DEF}^n\right) \tag{21}$$

$$\rho^n = \rho_s^n \chi_s^n + \rho_f(1 - \chi_s^n) \tag{22}$$

$$\nabla^2 \psi^n = -\omega^n \tag{23}$$

$$\nabla^2 \varphi^n = \sigma^n \tag{24}$$

$$\mathbf{u}^n = \nabla \times \psi^n + \nabla \varphi^n \tag{25}$$

$$\mathbf{u}_T^n = \frac{1}{M_s} \int_\Sigma \rho^n \chi_s^n \mathbf{u}^n d\mathbf{x} \tag{26}$$

$$\dot{\theta}^n = \frac{1}{J_s^n} \int_\Sigma \rho^n \chi_s^n \left(\mathbf{x} - \mathbf{x}_{cm}^n\right) \times \mathbf{u}^n d\mathbf{x} \tag{27}$$

$$\mathbf{u}_R^n = \dot{\theta}^n \times \left(\mathbf{x} - \mathbf{x}_{cm}^n\right) \tag{28}$$

$$\mathbf{u}_\lambda^n = \frac{\mathbf{u}^n + \lambda \Delta t \chi_s^n \left(\mathbf{u}_T^n + \mathbf{u}_R^n + \mathbf{u}_{DEF}^n\right)}{1 + \lambda \Delta t \chi_s^n} \tag{29}$$

$$\omega_\lambda^n = \nabla \times \mathbf{u}_\lambda^n \tag{30}$$

$$\frac{\partial \omega_\lambda^n}{\partial t} = -\frac{\nabla \rho^n}{\rho^n} \times \left(\frac{\partial \mathbf{u}_\lambda^n}{\partial t} + (\mathbf{u}_\lambda^n \cdot \nabla)\mathbf{u}_\lambda^n - \nu \nabla^2 \mathbf{u}_\lambda^n - \mathbf{g}\right) \tag{31}$$

$$\frac{\partial \omega_\lambda^n}{\partial t} = \nu \nabla^2 \omega_\lambda^n \tag{32}$$

$$\frac{\partial \omega_\lambda^n}{\partial t} + \nabla \cdot \left(\mathbf{u}_\lambda^n \omega_\lambda^n\right) = 0 \tag{33}$$

$$\omega^{n+1} = \omega_\lambda^{n+1} \tag{34}$$

$$\mathbf{x}_{cm}^{n+1} = \mathbf{x}_{cm}^n + \mathbf{u}_T^n \Delta t^n \tag{35}$$

$$\theta^{n+1} = \theta^n + \dot{\theta}^n \Delta t^n \tag{36}$$

2.4.1. Geometry and deformations

The methodology is summarized in Algorithm 1. Eqs. (19) and (20) represent the bilinear interpolation from the grid describing the body onto the computational grid. It should be noted that the deformation field $\mathbf{u}_{DEF}$ extends outside the body because the deformable mesh embeds the shape itself. Since the deformations $\mathbf{u}_{DEF}$ may not be locally volume-preserving, the recovery of a consistent velocity field across the extended domain requires us to evaluate the divergence field (Eq. (21)) and add the corresponding dilatational term in the Helmholtz decomposition (Eqs. (23)–(25)).

2.4.2. Divergence

In Eq. (21), the divergence σ is computed as $\chi_s(\nabla \cdot \mathbf{u}_{DEF})$ and not as $\nabla \cdot (\chi_s \mathbf{u}_{DEF})$. The field $\mathbf{u}_{DEF}$ is indeed extended outside the body, resulting in a consistent σ up to the boundary, whereas the second expression will produce a spurious divergence generated by the cut-off of the characteristic function.

We note that the divergence of the deformations is not conserved algebraically through the procedure of Eqs. (19)–(21). It is however captured at a satisfactory accuracy through the use of a bi-linear interpolation and centered finite differences. Additionally, the non-conservation does not lead to any error accumulation as the σ field is recomputed anew at the beginning of every time step.

2.4.3. Density

In step (22) a smooth density field is computed from fluid and solid densities through the characteristic function χ_s. If the shape does not conserve the volume, ρ_s is no longer constant, but becomes a function of time following Eq. (6) to ensure mass conservation.

2.4.4. Extended velocity field

In Eq. (25), we obtain the velocity field $\mathbf{u}$ from its Helmholtz decomposition. Its rotational and potential components, deriving respectively from a stream function ψ and a potential ϕ, are obtained from the Poisson Eqs. (23) and (24). We employ an unbounded Fourier solver [53,54] for the solution of these two scalar Poisson equations.

2.4.5. Projection and penalization

The effect of the fluid on the body is captured through the advection of the shape by the flow itself. The rigid components of the body motion are directly evaluated measuring the average translation ($\mathbf{u}_T$) and angular ($\dot{\theta}$) velocities according to Eqs. (26) and (27) where M_s and J_s are, respectively, the mass and moment of inertia of the body. Note that while M_s is constant, J_s is a function of time (the body is deforming) and therefore must be computed at each time step according to Eq. (56). The rotation velocity field $\mathbf{u}_R$ corresponding to the angular velocity $\dot{\theta}$ is computed in Eq. (28).

Step (29) implements the penalization model in its velocity formulation through a first order implicit Euler time discretization scheme [43,37].

2.4.6. Evolution of vorticity

Eq. (14) is split into the substeps (30)–(33). The first step (Eq. (30)) represents the vorticity correction due to the velocity penalization. The second step (Eq. (31)) is the baroclinic contribution in case body and fluid densities differ. Grid values of $\mathbf{u}^n_\lambda$ and $\mathbf{u}^{n-1}_\lambda$ are used to compute $\partial \mathbf{u}^n_\lambda / \partial t$, with a first order finite difference. Vorticity values are updated given the two previous contributions and then diffused (Eq. (32)) and advected (Eq. (33)) with an explicit second order scheme, as implemented in [37].

2.4.7. Vorticity advection and remeshing

The vorticity carried by the particles is then recomputed at the grid nodes via the interpolation kernel M'_4 [52]. We note that at this point $\mathbf{u}_\lambda$ is not necessarily divergence free all over the domain due to the contribution of $\mathbf{u}_{DEF}$. The advection scheme implemented here, which makes use of particles and remeshing [37], does not need to track particle volume changes caused by non-zero divergence in the velocity field, as showed in [57]. This owes to the fact that we redistribute an integral quantity (Γ_p, Section 2.1), carried by moderately distorted particles, onto a regular grid. This approach guarantees the conservation of vorticity and its accurate interpolation on the mesh.

2.4.8. Structure

The use of a first order implicit Euler operator for velocity penalization (Eq. (14)), and of a Godunov splitting approach (substeps (30)–(33)), which is, in the present form, first order, do not justify the use of a higher order time integration scheme for the advection of the body. Therefore the center of mass location ($\mathbf{x}_{cm}$) and rotation angle (θ) of the body are finally updated with an explicit first order accurate time scheme (Eqs. (35) and (36)).

Finally, we briefly discuss the scheme with respect to the conservation of linear and angular momentum. While the extended domain technique allows the computation of the transfer of momentum between the fluid and the solid, it is the penalization that handles the projection back into a consistent fluid/solid system. The momentum acquired by the solid phase is thus not algebraically enforced in the implicit penalization step.

Moreover, in the case of non-neutrally buoyant bodies, the conservation of momentum will also be conditioned by the properties of the evaluation scheme for the baroclinic term. As a matter of fact, that term demands coupling two formulations (vorticity/velocity and velocity/pressure), and the evaluation of time-derivatives and non-linear terms.

We leave the further investigation of this aspect as a topic of future work and rather focus on the convergence of the scheme in the coming sections.

3. Validation

In this section the proposed method is validated on several test cases. These cases include the flow past an impulsively started cylinder, the sedimentation of a 2D cylinder, a 2D ellipse and a characterization of the handling of added-mass effects by the present technique.

3.1. Flow past an impulsively started cylinder

Flow past an impulsively started cylinder is a degenerate case of flow-structure interaction, in which only the feedback from the solid to the fluid is resolved. We tested our method on such reduced fluid–structure problem, extending the results in [37], and present a systematic error analysis, investigating space and time convergence. Furthermore, as the ultimate goal is the simulation of self-propelled bodies, here the body is displaced at a constant velocity $\mathbf{U}_T$ and immersed in a viscous, incompressible flow, initially at rest. Such a set up greatly simplifies the general method stated above: the body is rigid ($\mathbf{u}_{DEF} = \mathbf{0}$), its velocity is imposed ($\mathbf{u}_T = \mathbf{U}_T$), rotation is not allowed ($\mathbf{u}_R = \mathbf{0}$) and the solid density is assumed to be the same as the fluid ($\rho_s = \rho_f$). Therefore the algorithm reduces to

$$\begin{aligned}
&\chi_s^n = \mathcal{P}_I(\mathbf{x}, \mathbf{r}_{\mathcal{M}} + \mathbf{x}_{cm}^n, \chi_{\mathcal{M}})\\
&\Delta\psi^n = -\omega^n\\
&\mathbf{u}^n = \nabla \times \psi^n\\
&\mathbf{u}_\lambda^n = \frac{\mathbf{u}^n + \lambda\Delta t \chi_s^n \mathbf{U}_T}{1 + \lambda\Delta t\chi_s^n}\\
&\omega_\lambda^n = \nabla \times \mathbf{u}_\lambda^n\\
&\frac{\partial\omega_\lambda^n}{\partial t} = \nu\nabla^2\omega_\lambda^n\\
&\frac{\partial\omega_\lambda^n}{\partial t} + \nabla\cdot(\mathbf{u}_\lambda^n\omega_\lambda^n) = 0\\
&\omega^{n+1} = \omega_\lambda^{n+1}\\
&\mathbf{x}_{cm}^{n+1} = \mathbf{x}_{cm}^n + \mathbf{U}_T\Delta t^n
\end{aligned}$$

To validate the method we compared the drag evolution in time with the reference computations [47,37]. We computed the forces acting on the body $\mathbf{F}$ both by integrating the penalization term (Eq. (37)) after velocity penalization (Eq. (29))

$$\mathbf{F} = \lambda\int_\Sigma \chi_s(\mathbf{u}_\lambda - \mathbf{U}_T)d\mathbf{x} \tag{37}$$

and by means of the expression introduced in [47]

$$\mathbf{F} = \frac{d}{dt}\int_{\Sigma\setminus\Omega}(\boldsymbol{\omega}\times\mathbf{x})d\mathbf{x} \tag{38}$$

where the body force is computed as the time derivative (approximated by a central difference scheme) of the cross product between the vorticity and its location, over the fluid domain. In order to use the latter formulation the computational domain must be big enough to ensure that the generated vorticity is conserved. The discrepancy between the two approaches in the drag curve was found to be at all times less than 1%.

The Reynolds number (Re), nondimensionalized time (T) and drag coefficient (C_d) are defined as follows

$$Re = \frac{|\mathbf{U}_T|D}{\nu}, \quad T = \frac{2|\mathbf{U}_T|t}{D}, \quad C_d = \frac{2F}{|\mathbf{U}_T|^2 D} \tag{39}$$

where D is the cylinder diameter. All simulations of this section were carried out at $Re = 1000$, by setting $U_T = 0.1$, $D = 0.2$ and domain size $[0,1]\times[0,1]$. Fig. 2 shows the evolution of the drag coefficient in time for a 2048 × 2048 resolution, $LCFL = 0.01$ and $\lambda = 1e^4$. It compares well with reference computations [47,37], capturing the transient and drag evolution. Spatial and temporal convergence were also investigated. The convergence order was determined by computing the L^1, L^2, L^∞ norm of the error $e(t)$ (Eq. (40)) of the drag coefficient (Eq. (39)) with respect to the best resolved case

$$e(t) = \|C_{d\text{best resolved}}(t) - C_d(t)\| \tag{40}$$

Space convergence studies were carried out both fixing the model, i.e. by setting the mollification length ϵ proportional to the coarsest resolution ($\epsilon = 2\sqrt{2}h_{\text{coarsest}}$), and fixing the ratio ϵ/h, to investigate convergence to the actual geometry. For both convergence studies, we set $LCFL = 0.01$, $\lambda = 1e^4$ and varied the resolution between 256 × 256 and 4096 × 4096. As can be seen in Fig. 3 the method presents second order convergence (fixing $\epsilon = 2\sqrt{2}h_{256}$) and between first and second order (1.45 for L_{inf}, L^1 and 1.53 for L^2) keeping the ratio $\epsilon/h = 2\sqrt{2}$ fixed. Such space convergence results are in agreement with previous studies for immersed boundary methods [58]. Time convergence study was carried out by setting the resolution

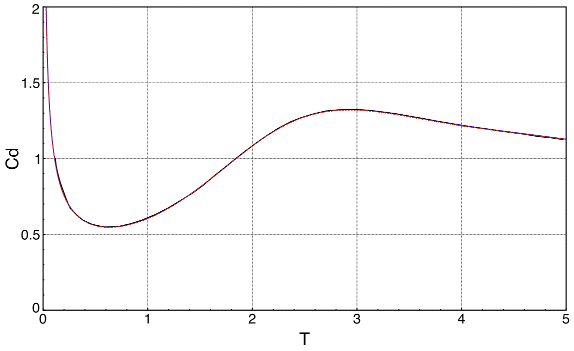

Fig. 2. Drag coefficient (C_d) at Re = 1000 against the nondimesionalized time T. The red line is obtained with a resolution of 2048 × 2048, $LCFL$ = 0.01, $\lambda = 1e^4$ and $\epsilon = 2\sqrt{2}h$. The black and blue lines are, respectively, the reference solutions [47] and [37]. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

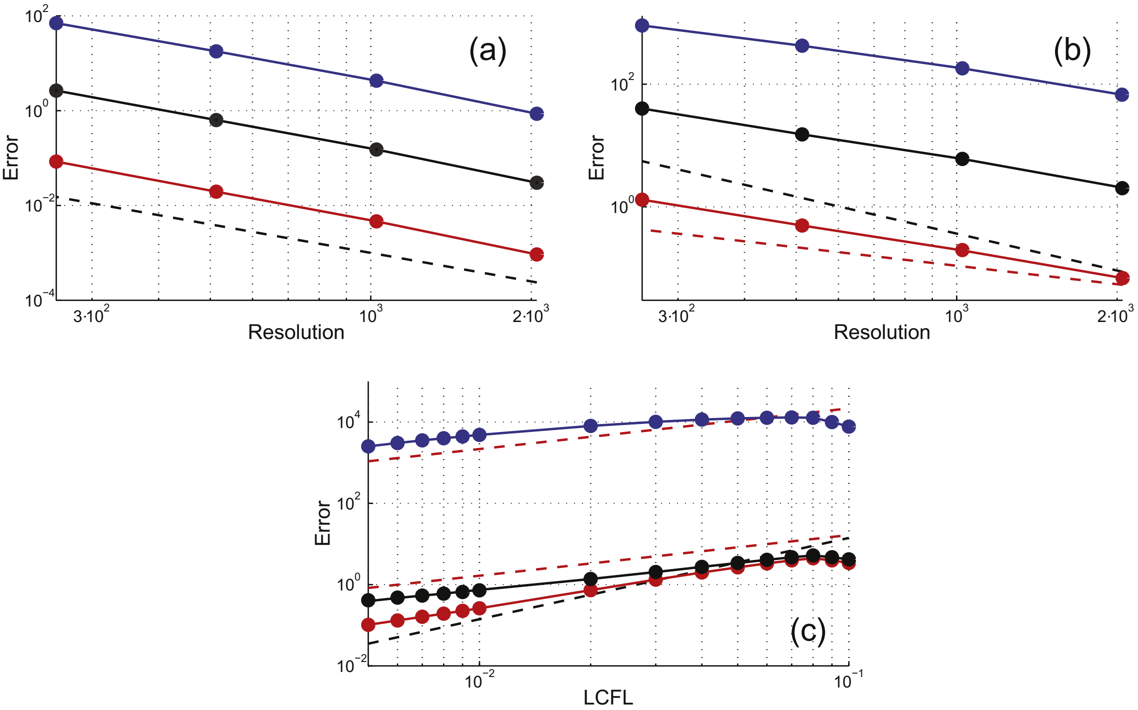

Fig. 3. (a) Space convergence (mollification length fixed based on the coarsest resolution, $\epsilon = 2\sqrt{2}h_{256}$): $L^\infty(e)$ (blue), $L^1(e)$ (black) and $L^2(e)$ (red) are plotted against domain resolution. $LCFL$ was set to 0.01. (b) Space convergence (ratio ϵ/h fixed to $2\sqrt{2}$): $L^\infty(e)$ (blue), $L^1(e)$ (black) and $L^2(e)$ (red) are plotted against domain resolution. $LCFL$ was set to 0.01. (c) Time convergence: $L^\infty(e)$ (blue), $L^1(e)$ (black) and $L^2(e)$ (red) are plotted against $LCFL$. Resolution was set to 1024 × 1024. For all studies we used $\lambda = 1e^4$. Dashed red and black lines represent, respectively, first and second order slopes. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

to 1024 × 1024, $\epsilon = 2\sqrt{2}h_{1024}$, $\lambda = 1e^4$ and varying $LCFL$ between 0.1 and 0.005 (with 0.001 as best resolved case). The convergence order was found to be first order for L_{inf}, L^1 and 1.34 for L^2 (Fig. 3).

3.2. Sedimentation of a 2D cylinder

Here we consider the case of a falling 2D cylinder to test the two-way fluid–solid coupling for a rigid body. We compare our results with Namkoong et al. [59] and investigate time and space convergence of the method. We simulate the same physical system as Namkoong et al.: a rigid 2D cylinder of diameter D = 0.005 m and with ρ_s/ρ_f = 1.01 is released from rest and accelerates due to gravity ($g = -9.81$ m/s^2) until it reaches its asymptotic terminal velocity, corresponding to a Re = 156 in water ($\nu = 8e^{-7}$ m^2/s, ρ_f = 996 kg/m^3). The domain size was set to [0, 0.04] × [0, 0.32] m, resolution 1024 × 8192, $\lambda = 1e^4$,

$LCFL = 0.01$ and $\epsilon = 2\sqrt{2}h$. Fig. 4 shows the time evolution of the normalized streamwise ($\mathbf{u}_{str}/U_t$), where U_t indicates the reference terminal velocity obtained by Namkoong et al. [59], lateral ($\mathbf{u}_{lat}/U_t$) and angular velocity ($\dot{\theta}D/U_t$) obtained with the present method and by Namkoong et al. [59]. As can be seen velocities show the same dynamics as the reference. In particular the streamwise velocity overshoots above the terminal velocity and then slows down when the vortices start shedding. The normalized terminal velocity differs less than 1% from the reference terminal velocity. The lateral and angular velocities in regime also agree both in amplitude and frequency. The initial transient is dominated by the breaking of the symmetry of the flow due to numerical noise. This effect is not controlled (it depends on grid alignment and the implementation) and explains the delayed streamwise velocity overshoot and the different transient profile for the lateral velocity. Fig. 5 shows the vorticity structures generated by the falling cylinder. To investigate space and time convergence, we fixed lateral and angular velocities to zero, since they are triggered by numerical noise. Therefore only one degree of freedom was let free to evolve, namely the streamwise velocity $\mathbf{u}_{str}$. With respect to the previous set up, we set $\rho_s/\rho_f = 1.02$ and $\nu = 1e^{-7}$ m^2/s to achieve a higher Reynolds number. Simulations were stopped when the falling cylinder reached a velocity corresponding to $Re = 1500$, which was reached before the alternating vortex shedding was initiated. The convergence order was determined by computing the L^1, L^2, L^∞ norm of the error $e(t)$ of the streamwise velocity $\mathbf{u}_{str}$ with respect to the best resolved case

$$e(t) = \|\mathbf{u}_{str\text{best resolved}}(t) - \mathbf{u}_{str}(t)\| \tag{41}$$

Space convergence studies were carried out both fixing the model, i.e. by setting the mollification length ϵ proportional to the coarsest resolution ($\epsilon = 2\sqrt{2}h_{\text{coarsest}}$), and fixing the ratio ϵ/h, to investigate convergence to the actual geometry. For both convergence studies, we set $LCFL = 0.001$ (to be $LCFL$-bound at all times), $\lambda = 1e^4$ and varied the resolution between 256 × 256 and 4096 × 4096. As can be seen in Fig. 6 the method presents second order convergence (fixing $\epsilon = 2\sqrt{2}h_{256}$) and between first and second order (1.57 for L_{inf}, L^1 and 1.72 for L^2) keeping the ratio $\epsilon/h = 2\sqrt{2}$ fixed. Time convergence study was carried out by setting the resolution to 2048 × 2048, $\epsilon = 2\sqrt{2}h_{2048}$, $\lambda = 1e^4$ and varying $LCFL$ between 0.1 and 0.005 (with 0.001 as best resolved case). The convergence order was found to be first order for L_{inf}, L^1 and 1.32 for L^2 (see Fig. 6).

Coquerelle and Cottet in [43] carried out a 3D study on a similar problem, to show the capability of vortex methods coupled with penalization for rigid bodies. In [43] the Reynolds number was limited to $Re \sim 20$ and the validation was carried out based on the value of the terminal velocity, while we account for the evolution of the streamwise, lateral and angular velocities in time. Given the low Re, the terminal velocity in [43] is reached after a dimensionless time $T = 2.5$ and this does not allow a complete assessment of the impact of the flow on the lateral and angular velocities. The discretization schemes for the baroclinic term (Eq. (31)) is different too as we included the diffusion contribution, which was neglected in [43]. Furthermore here we provide a rigorous error analysis of the method, investigating space and time convergence.

We also investigate the effect of the interaction mediated by the fluid in the case of multiple falling cylinders. Five horizontally aligned cylinders, separated by a distance $d = 1.5D$, are released form rest from the same height. As can be seen in Fig. 7 the flow structures generated by the single cylinders affect the overall behavior of the system, leading to a complex motion pattern. Trajectories, streamwise, lateral and angular velocities are reported in Fig. 8. The symmetric behavior achieved in such system proves the accuracy of our methodology. The simulation was carried out by setting $D = 0.005$ m, $\nu = 8.0e^{-7}$ m^2/s, $g = 9.81$ m/s^2, $\rho_f = 996$ kg/m^3, $\rho_s/\rho_f = 1.2$, domain size $[0, 0.15] \times [0, 0.15]$ m, resolution 2048 × 2048, $\lambda = 1e^4$, $LCFL = 0.01$ and $\epsilon = 2\sqrt{2}h$.

3.3. Sedimentation of a 2D ellipse

The hydrodynamics of freely falling plates due to gravity is a challenging problem due to the complex interplay between the plates and the induced vortical structures. A wide range of behaviors (steady falling, fluttering, chaotic and tumbling

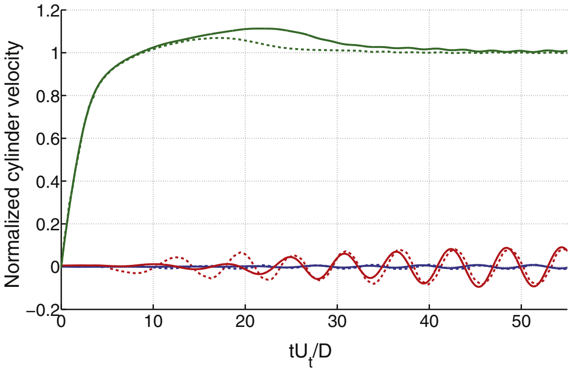

Fig. 4. Time histories of the normalized streamwise (green, $\mathbf{u}_{str}/U_t$), lateral (red, $\mathbf{u}_{lat}/U_t$) and angular (blue, $\dot{\theta}D/U_t$) velocities of the falling disk. Solid lines correspond to the present method, while broken lines the terminal ($U_t = 2.501$ cm/s) and overshooting velocities predicted by Namkoong et al. [59]. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

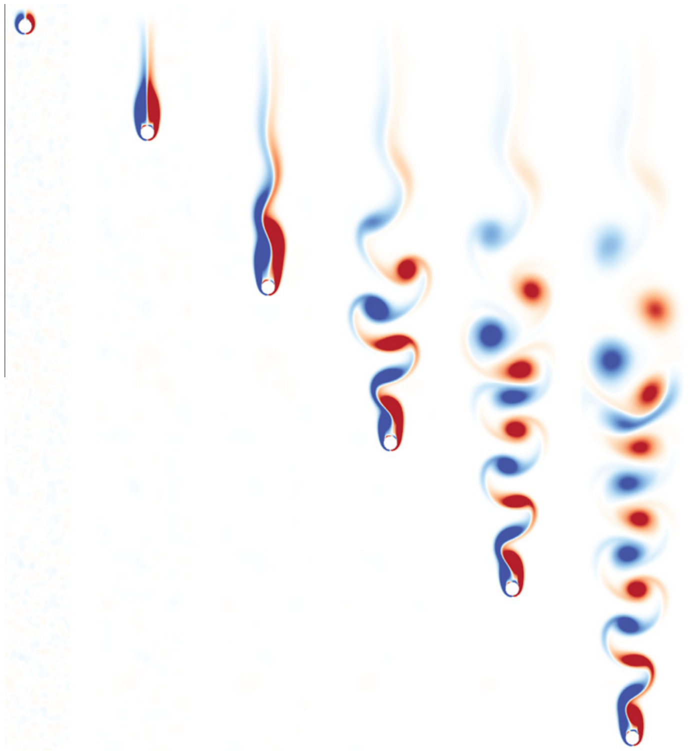

Fig. 5. Sedimentation of a 2D cylinder. Vorticity at different normalized times T: from left to right $T = 2.5$, $T = 11$, $T = 22$, $T = 33$, $T = 44$, $T = 55$.

[60]) can be generated by varying the plate's geometry. In particular, Field et al. [60] showed that the disk's dynamical behavior can be characterized by two parameters, the dimensionless moment of inertia I^* and the Reynolds number. For an ellipse, we use the corresponding 2D definitions: $I^* = 32I/(\pi\rho_f L^4) = (H(L^2 + H^2)\rho_s)/(2L^3\rho_f)$ and $Re = U_t L/\nu$, where I is the 2D ellipse moment of inertia, $(\pi\rho_f L^4)/32$ is the moment of inertia about the axis of symmetry of a cylinder of density ρ_f, L and H are, respectively, the width and thickness of the elliptical plate and U_t the sedimentation average velocity (estimated through $U_t = \sqrt{(}2(\rho_s/\rho_f - 1)Hg)$, g being the gravity [61]). Here we illustrate the ability of the method to capture such complex dynamics, showing simulations for each of the four previously mentioned regimes (Figs. 9 and 10). The simulations reproduce qualitatively the experimental results presented in [61]. In Fig. 9 it is worthwhile to notice how the ellipse is colored when rotating: this is due to the fact that we are plotting the vorticity field of the whole domain and the angular velocity $(\dot{\theta})$ of the ellipse corresponds to a constant vorticity value within the shape itself. All simulations were carried out by setting $H = 1.62e^{-3}$ m, the initial angle to $\alpha = \pi/4$, $\rho_f = 1000\ \mathrm{kg/m^3}$, the domain size to $[0,20L] \times [0,20L]$, the resolution to 2048×2048, $\lambda = 1e^4$, $LCFL = 0.05$ and $\epsilon = \sqrt{2}h$, while H/L, ν and ρ_s/ρ_f were varied to achieve different I^* and Re pairs.

3.4. Added mass

Whenever a body immersed in a fluid is accelerated, additional forces will act on the surface of the solid, and the additional inertia of the system is denoted as added mass. In this section we study the capability of the present method to capture this effect for different geometries accelerated through a viscous, incompressible flow due to gravity (g). In particular, for simple geometries such as the cylinder or the ellipse, the added mass can be computed analytically from potential flow theory

$$M^{\text{cylinder}}_{\text{added}} = \pi\rho_f r^2, \quad M^{\text{ellipse}}_{\text{added}} = \pi\rho_f a^2 \tag{42}$$

where ρ_f is the fluid density, r is the cylinder radius and a is the ellipse semi-axis normal to the acceleration direction. We measured the added mass from

$$(M_{added}(0^+) + M_{solid})\mathbf{a}(0^+) = (M_{solid} - M_{fluid})\mathbf{g} \tag{43}$$

where the acceleration $\mathbf{a}(0^+)$ is computed with a forward Euler finite difference scheme

$$\mathbf{a}(0^+) = \frac{\mathbf{u}(\Delta t) - \mathbf{u}(0)}{\Delta t} \tag{44}$$

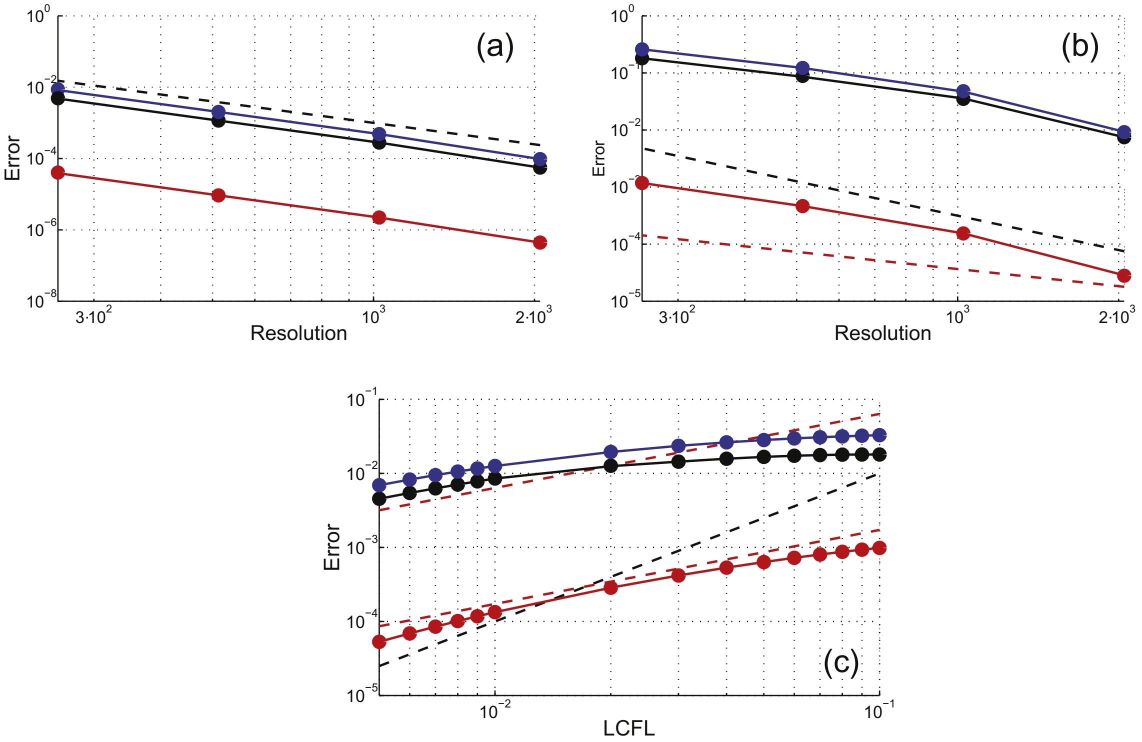

Fig. 6. (a) Space convergence (mollification length fixed based on the coarsest resolution, $\epsilon = 2\sqrt{2}h_{256}$): $L^{\infty}(e)$ (blue), $L^{1}(e)$ (black) and $L^{2}(e)$ (red) are plotted against domain resolution. *LCFL* was set to 0.001. (b) Space convergence (ratio ϵ/h fixed to $2\sqrt{2}$): $L^{\infty}(e)$ (blue), $L^{1}(e)$ (black) and $L^{2}(e)$ (red) are plotted against domain resolution. *LCFL* was set to 0.001. (c) Time convergence: $L^{\infty}(e)$ (blue), $L^{1}(e)$ (black) and $L^{2}(e)$ (red) are plotted against *LCFL*. Resolution was set to 2048 × 2048. For all studies we used $\lambda = 1e^{4}$. Dashed red and black lines represent, respectively, first and second order slopes. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

We carried out this study using the following set up with gravity ($g = -9.81\ \mathrm{m/s^2}$), viscosity $\nu = 1e^{-7}\ \mathrm{m^2/s}$, density $\rho_f = 996\ \mathrm{kg/m^3}$, $\rho_s/\rho_f = 1.02$, domain size $[0, 0.0075] \times [0, 0.0075]$ m, $\lambda = 1e^{4}$ and $\Delta t = 1e^{-7}$. The resolution was varied between 128 × 128 and 1024 × 1024 and the ratio ϵ/h was kept at $5\sqrt{2}$. We investigated the discrepancy between the theoretical added mass and that computed from simulations. The resolution was varied to study cylinder ($r = 0.0025$ m), and ellipses with aspect ratios $a/b = \{1/10, 1/2, 2, 10\}$ (where a and b are, respectively, horizontal and vertical semi-axis and their length is set according to $\max(a, b) = 0.0025$ m).

As it can be seen in Fig. 11, the relative error percentage is shown to converge for all cases. The ellipses with 1/10 and 1/2 aspect ratio show an offset in the error due to the higher ratio ϵ/a. Ellipses with an aspect ratio of 2 and 10 present a slower convergence rate. This effect is expected, as the main vortical structures take place at the lateral extrema of the ellipse, that become less and less resolved as the aspect ratio is increased. Overall the method captures satisfactorily the added mass effect showing relative errors around 1% or less.

4. Swimming applications

In this section we explore the capability of the method in the context of swimming. The method is rigorously validated against reference simulations [17,22] and an error analysis is provided. Simulations involving multiple swimmers are also presented in order to illustrate the interactions mediated by the fluid among school members.

4.1. Self-propelled anguilliform swimmer

We present a comparison between our methodology and the method of Kern and Koumoutsakos [22], which uses finite volume methods and body-conforming grids, for the simulation of 2D self-propelled anguilliform swimmers. The motion pattern used for validation purposes in [22] was first introduced by [17], and consists of a two dimensional deformation of the swimmer mid-line. The lateral displacement of the mid-line, $y_s(s,t)$, in a local frame of reference is defined as

$$y_s(s,t) = 0.125L\frac{0.03125 + \frac{s}{L}}{1.03125}\sin\left[2\pi\left(\frac{s}{L} - \frac{t}{T}\right)\right] \tag{45}$$

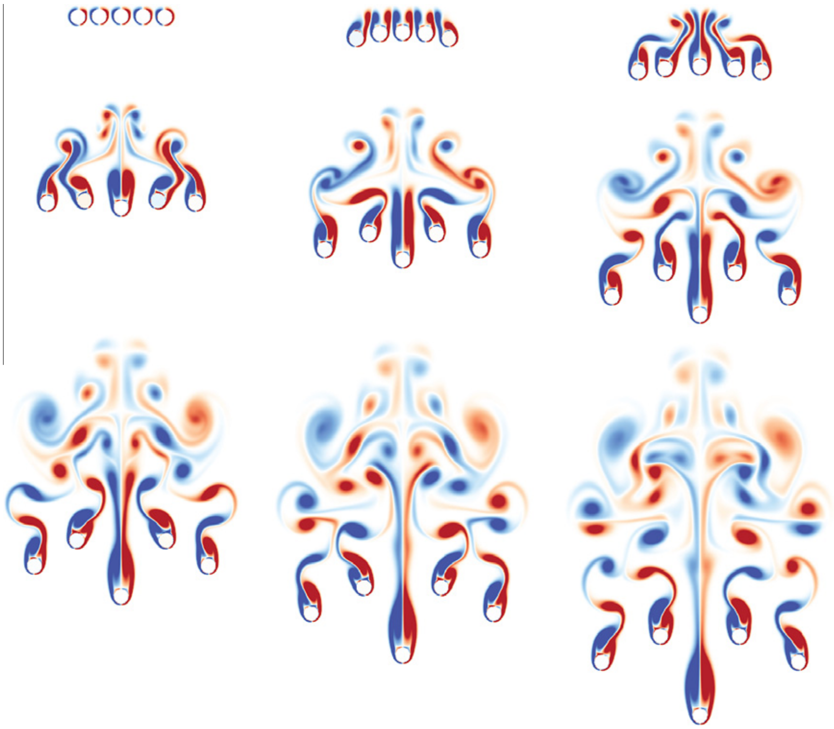

Fig. 7. Vorticity plots of five free falling cylinders. From left to right and top-down: $t = 0.066$ s, $t = 0.33$ s, $t = 0.66$ s, $t = 1.0$ s, $t = 1.33$ s, $t = 1.66$ s, $t = 2.0$ s, $t = 2.33$ s, $t = 2.66$ s.

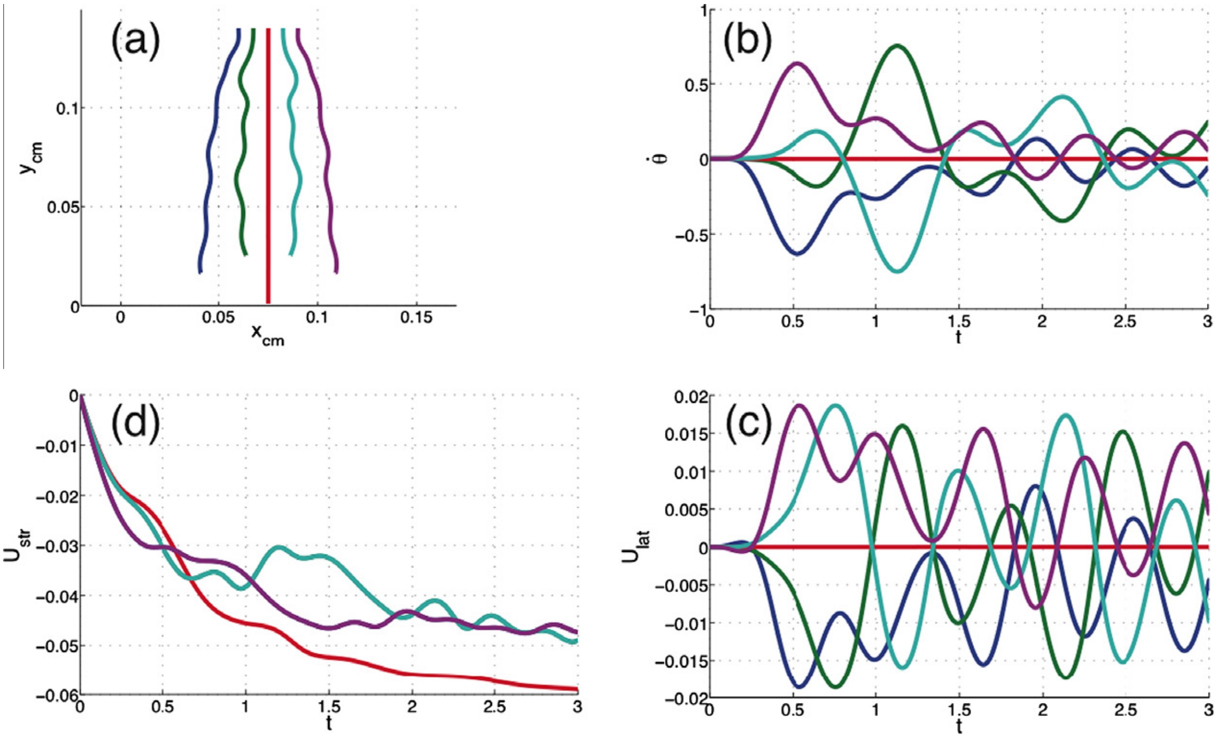

Fig. 8. Five free falling cylinders. (a) Trajectories of the centers of mass, units expressed in m, (b) angular velocities $\dot{\theta}$, (c) streamwise velocities, expressed in m/s, (d) lateral velocities, expressed in m/s. Time t is expressed in seconds.

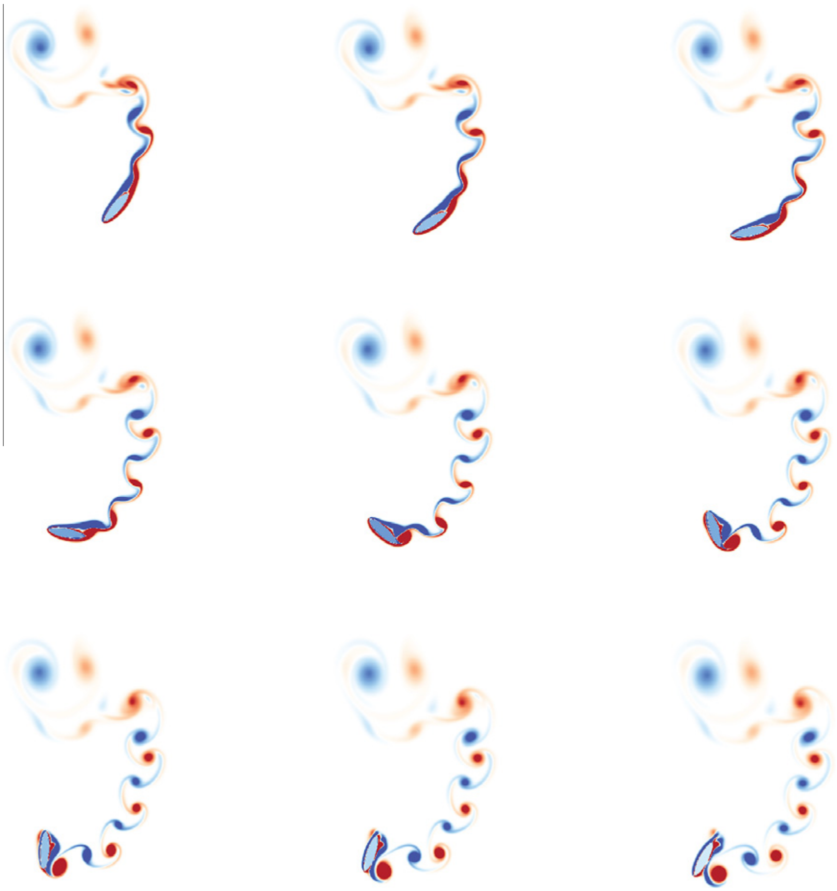

Fig. 9. Vorticity plots of the simulated free falling ellipse for one tumbling revolution ($I^* = 0.146$, $Re = 1000$, $H/L = 1/4$, $\rho_s/\rho_f = 1.1$).

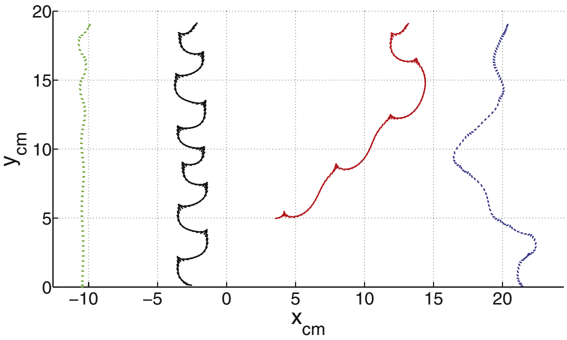

Fig. 10. Free falling ellipses, the segments indicate the major axis of the ellipse (coordinates are reported in cord lengths): (green) steady falling regime ($I^* = 0.146, Re = 100, H/L = 1/4, \rho_s/\rho_f = 1.1$), (black) fluttering regime ($I^* = 0.16, Re = 1147, H/L = 1/5, \rho_s/\rho_f = 1.538$), (red) tumbling regime ($I^* = 0.146, Re = 1000, H/L = 1/4, \rho_s/\rho_f = 1.1$) and (blue) chaotic regime ($I^* = 0.39, Re = 1770, H/L = 1/2, \rho_s/\rho_f = 1.248$). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

where L is the fish length, s is the arc length of the mid-line of the body ($0 \leqslant s \leqslant L$), t is the time, and T the period. The computation of the characteristic function and deformation velocity field on the deformable grid is carried out as detailed in Appendix A. The only difference relies on the parameterization of the mid-line motion. Here, curvature is not given, instead an explicit definition of the mid-line displacement is given. Therefore, the Frenet equations (see Appendix A, Eqs. (51) and (52)) do not have to be solved, rather the normals and velocities are approximated with a finite difference scheme.

All simulations are started with the body at rest and the motion is initialized by gradually increasing the displacement magnitude (through a sinusoidal function) from zero to its designated value during the first cycle. We set the viscosity of

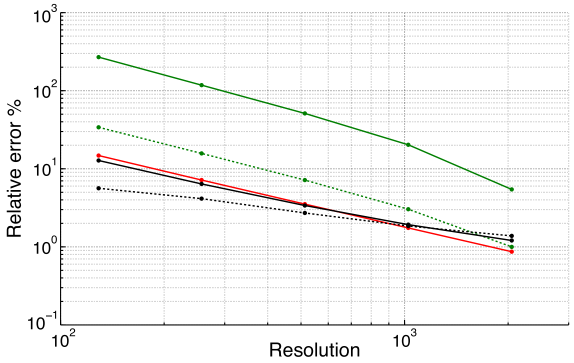

Fig. 11. The relative error percentage between the theoretical added mass and the ones computed through the presented method for several geometries. Simulations were carried out by setting $\Delta t = 1e^{-7}$, varying the resolution between 128 × 128 and 1024 × 1024, and fixing the ratio $\epsilon/h = 5\sqrt{2}$. (red line) cylinder, (green solid line) ellipse with aspect ratio 1/10, (green dashed line) ellipse with aspect ratio 1/2, (black solid line) ellipse with aspect ratio 2, (black dashed line) ellipse with aspect ratio 10. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

the fluid to be $\mu = 1.4e^{-4}$, the body length to be $L = 1$ and the density to be $\rho_{\text{fluid}} = \rho_{\text{body}} = 1$, resulting in a Reynolds number of approximately 3800, given the final swimming velocity. The volume is conserved, therefore fluid and body densities are the same and the baroclinic term (Eq. (31)) drops out. Even though the average divergence over the fish volume is zero (volume is conserved) locally the deformation velocity field is not divergence free and we have to account for that. Fig. 12 depicts the unsteady forward ($U_\parallel$) and lateral ($U_\perp$) velocities of the center of mass of the body, normalized by L. Simulations (Fig. 17) are carried out on a rectangular domain $[0,8L] \times [0,4L]$ with a resolution of 4096 × 2048, $LCFL = 0.01$, $\lambda = 1e^4$, and $\epsilon = 2\sqrt{2}h$.

We compared with [22] the swimming velocity and the forces acting on the swimmer. The swimmer accelerates from rest to an asymptotic mean forward velocity of $\overline{U}_\parallel = 0.54$, while the lateral velocity ($U_\perp$) oscillates around the mean zero value with an amplitude of 0.04. The net force coefficients $C_\parallel$ and $C_\perp$ are evaluated as

$$C_\parallel = \frac{2M_s\dot{U}_\parallel}{\rho_f\overline{U}_\parallel^2 L}, \quad C_\perp = \frac{2M_s\dot{U}_\perp}{\rho_f\overline{U}_\parallel^2 L} \tag{46}$$

and are plotted in Fig. 13. Accelerations are computed using centered finite differences on the present and the reference's [22] velocity time series. The net force coefficients, plotted in Fig. 13, converge to an oscillatory regime with zero mean and a constant amplitude of 0.06 and 0.08, respectively. As can be seen in Figs. 12 and 13, the present method agrees with the reference. It is interesting to note how neglecting the treatment of the local divergence of the fish deformation velocity field ($\mathbf{u}_{DEF}$) results in a considerable reduction of the terminal forward velocity ($U_\parallel$) and in discrepancies in the net force coefficients ($C_\parallel$ and $C_\perp$). This is due to the fact that locally the divergence of the deformation velocity field is non zero and that must be taken into account in the formulation and treatment of the flow model inside the deforming body. We note that divergence is locally non-zero both before and after the interpolation onto the computational grid, and it is not an artifact of the interpolation, furthermore the integral of the divergence of $\mathbf{u}_{DEF}$ over the swimmer's body is zero, ensuring that its total volume is conserved.

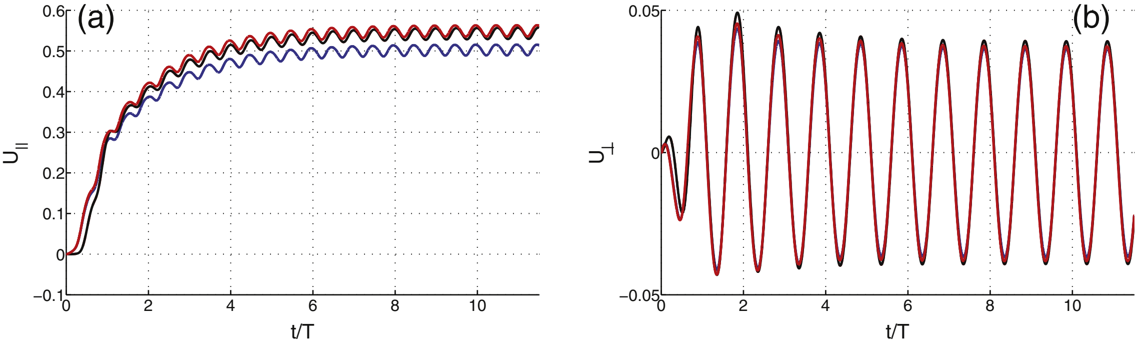

Fig. 12. Anguilliform swimmer: (a) Forward velocity $U_\parallel$, (b) Lateral velocity $U_\perp$. Red lines indicate 2D swimmer simulations with the proposed method. Blue lines indicate simulations with the present method, but without accounting for non-solenoidal deformations (i.e. σ) inside the fish. Black lines indicate the reference finite volume solution [22]. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

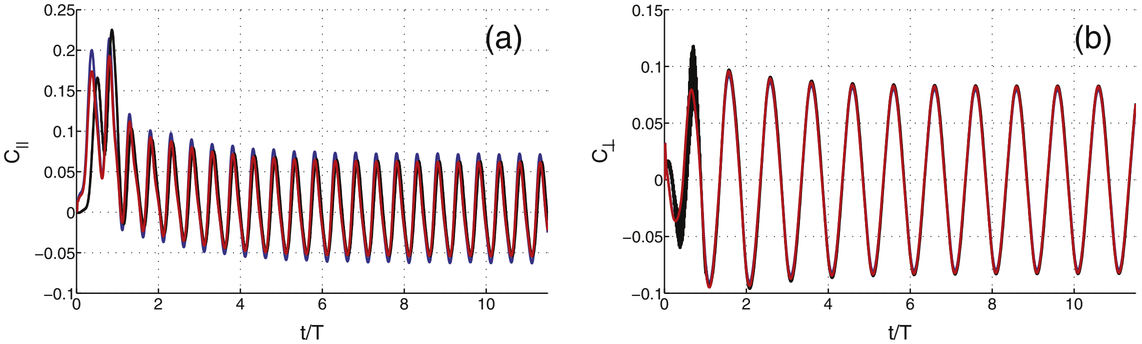

Fig. 13. Anguilliform swimmer: (a) Forward net force coefficient $C_{\|}$, (b) Lateral net force coefficient $C_{\perp}$. Red lines indicate 2D swimmer simulations with the proposed method. Blue lines indicate simulations with the present method, but without accounting for non-solenoidal deformations (i.e. σ) inside the fish. Black lines indicate the reference finite volume solution [22]. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

We investigated convergence for this study case as well. Space convergence studies were carried out both by fixing the model, i.e. by setting the mollification length ϵ proportional to the coarsest resolution ($\epsilon = 2\sqrt{2}h_{\text{coarsest}}$), and by fixing the ratio ϵ/h, to investigate convergence to the actual geometry. Domain size was set to $[0,2L] \times [0,1L]$ and the error was computed as defined by Eq. (41), for forward and lateral velocity during the first two cycles. For both convergence studies, we set $LCFL = 0.001$ (to be $LCFL$-bound at all times), $\lambda = 1e^4$, and varied the resolution between 256 × 256 and 4096 × 4096. As can be seen in Figs. 14 and 15, the method exhibits second order convergence (fixing $\epsilon = 2\sqrt{2}h_{256}$) and between first and second order (1.46 for L_{inf}, L^1 and 1.52 for L^2, in the case of forward velocity, and 1.55 for L_{inf}, L^1 and 1.8 for L^2, in the case of lateral velocity) when keeping the ratio $\epsilon/h = 2\sqrt{2}$ fixed. Time convergence study was carried out by setting the resolution to 2048 × 1024, $\epsilon = 2\sqrt{2}h$, $\lambda = 1e^4$, and varying the $LCFL$ between 0.1 and 0.005 (with 0.001 as best resolved case). The convergence order was found to be first order for L_{inf}, L^1 and 1.37 for L^2 (Fig. 16).

4.2. Multiple swimmers

The capability of the method of capturing the interactions between multiple swimmers is illustrated in this section.

We simulated two fish swimming in tandem [62,18] one next to the other. Fig. 18 shows the vorticity fields generated by two in-phase fish and two anti-phase fish. Simulations are started with bodies at rest and the motion is initialized by gradually increasing the displacement magnitude (through a sinusoidal function) from zero to its designated value during the first cycle. We set the viscosity of the fluid to be $\mu = 1.4e^{-4}$, the body length $L = 1$ and the density $\rho_{\text{fluid}} = \rho_{\text{body}} = 1$. Simulations were carried out on a rectangular domain $[0,8L] \times [0,4L]$ with a resolution of 4096 × 2048, $LCFL = 0.01$, $\lambda = 1e^4$, $\epsilon = 2\sqrt{2}h$, and the separation distance between the fish $d = 0.6L$. As can be noticed the flow structures generated by the in-phase and anti-phase swimming, influence the motion of the partner. In the case of in-phase fish, the flow produced by the couple push them far apart, leading to divergent trajectories (Figs. 18 and 19a), while in the case of anti-phase swimming, they first converge then diverge once they get too close to each other (Figs. 18 and 19b). Furthermore the different

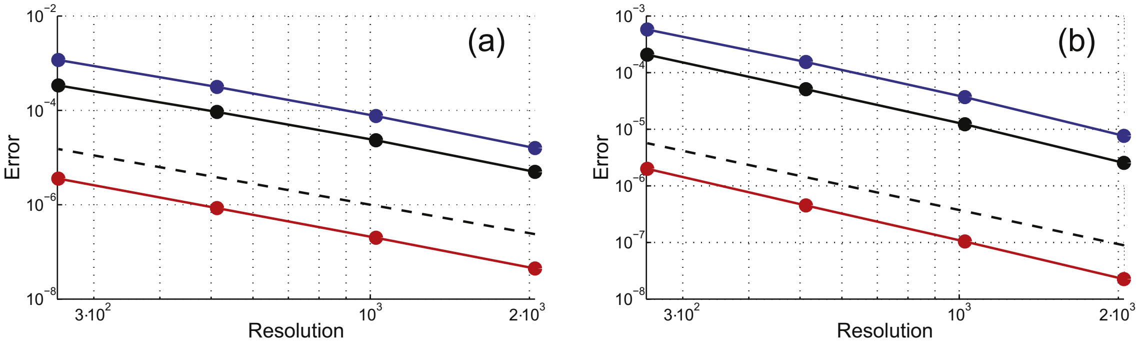

Fig. 14. Anguilliform swimmer: (a) Space convergence, the mollification length fixed based on the coarsest resolution, $\epsilon = 2\sqrt{2}h_{256}$. $L^{\infty}(e)$ (blue), $L^1(e)$ (black) and $L^2(e)$ (red) of the forward velocity $U_{\|}$ are plotted against domain resolution. The $LCFL$ was set to 0.001. (b) Space convergence, the mollification length fixed based on the coarsest resolution, $\epsilon = 2\sqrt{2}h_{256}$. $L^{\infty}(e)$ (blue), $L^1(e)$ (black) and $L^2(e)$ (red) of the lateral velocity $U_{\perp}$ are plotted against domain resolution. The $LCFL$ was set to 0.001. For all studies we used $\lambda = 1e^4$. Dashed black lines represent second order slopes. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

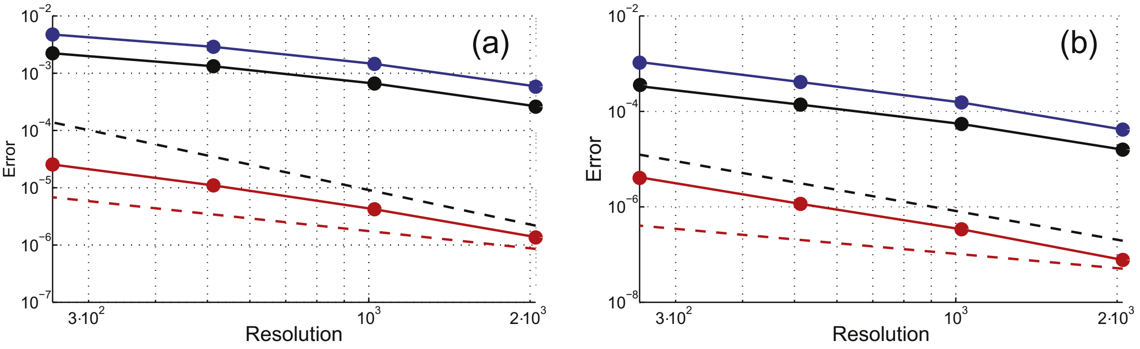

Fig. 15. Anguilliform swimmer: (a) Space convergence (ratio ϵ/h fixed to $2\sqrt{2}$): $L^{\infty}(e)$ (blue), $L^{1}(e)$ (black) and $L^{2}(e)$ (red) of the forward velocity $U_{\parallel}$ are plotted against domain resolution. The *LCFL* was set to 0.001. (b) Space convergence (ratio ϵ/h fixed to $2\sqrt{2}$): $L^{\infty}(e)$ (blue), $L^{1}(e)$ (black) and $L^{2}(e)$ (red) of the lateral velocity $U_{\perp}$ are plotted against domain resolution. The *LCFL* was set to 0.001. For all studies we used $\lambda = 1e^{4}$. Dashed red and black lines represent, respectively, first and second order slopes. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

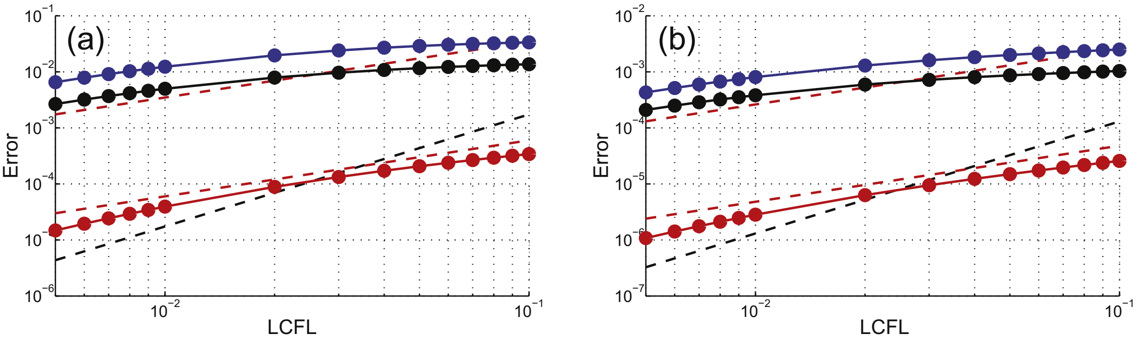

Fig. 16. Anguilliform swimmer: (a) Time convergence: $L^{\infty}(e)$ (blue), $L^{1}(e)$ (black) and $L^{2}(e)$ (red) of the forward velocity $U_{\parallel}$ are plotted against *LCFL*. Resolution was set to 2048 × 2048. (b) Time convergence: $L^{\infty}(e)$ (blue), $L^{1}(e)$ (black) and $L^{2}(e)$ (red) of the forward velocity $U_{\perp}$ are plotted against *LCFL*. The resolution was set to 2048 × 2048. For all studies we used $\lambda = 1e^{4}$. Dashed red and black lines represent, respectively, first and second order slopes. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

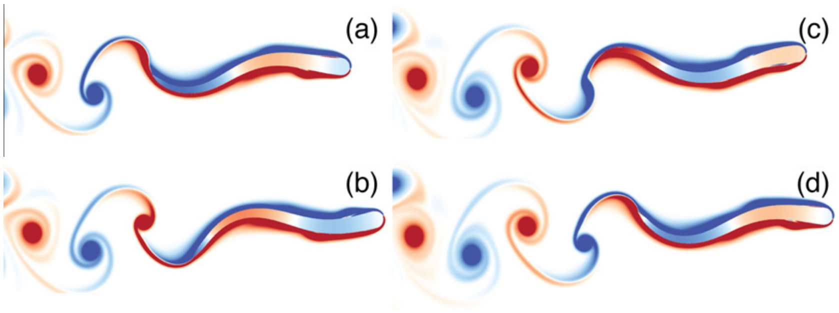

Fig. 17. Self-propelled anguilliform swimmer. Vorticity during one period: (a) $t = t_0$, (b) $t = t_0 + 0.25T$, (c) $t = t_0 + 0.5T$, (d) $t = t_0 + 0.75T$.

phases affect the swimming velocity as can be observed in Fig. 19b. Anti-phase configuration seems to favor the swimmers as they are consistently faster (~3.3%) then the in-phase swimmers.

We also simulated a small school consisting of five individuals, to illustrate the impact of the geometrical school configuration on the fish trajectories and velocities. Fig. 20 illustrates the vortical structures generated by a V-shaped and an aligned school of fish. As can be seen the V-shape favors rectilinear trajectories (Figs. 20 and 21a), while the aligned configuration leads immediately to their divergence (Figs. 20 and 21a). Furthermore the geometrical configuration affects the forward swimming velocity as can be observed in Fig. 21b. V-shape configuration seems to favor swimmers as they are consistently faster (in average ~8.8%) than in the aligned arrangement. Simulations were carried out with the same setting as the tandem fish. The separation distances for the V-shape are $d_y = 0.4L$ and $d_x = 1.5L$, respectively, in the vertical and hor-

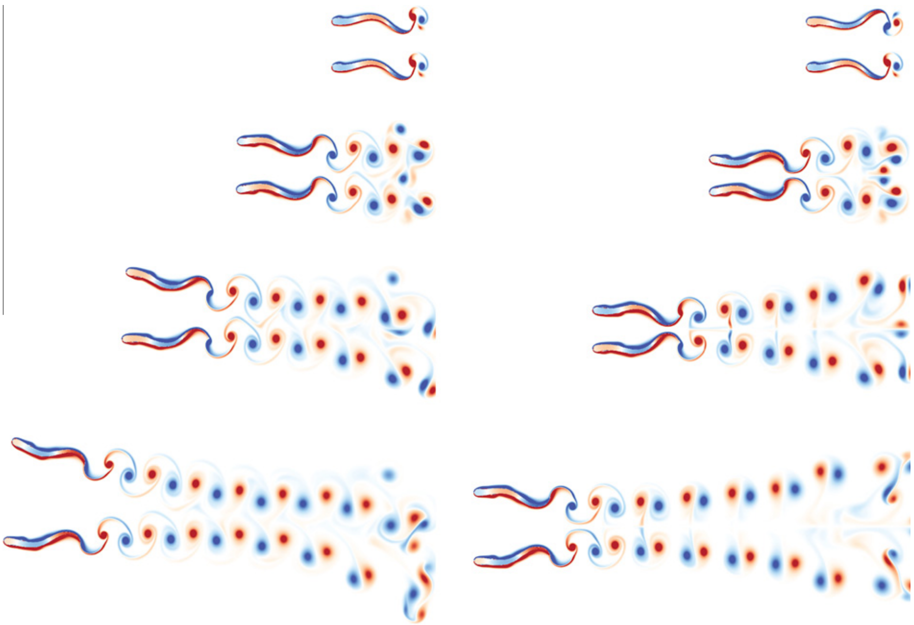

Fig. 18. Tandem fish vorticity plots: (left) phase fish, (right) anti-phase fish. From top to bottom $t/T = 2.0$, $t/T = 3.5$, $t/T = 6.0$, $t/T = 8.5$.

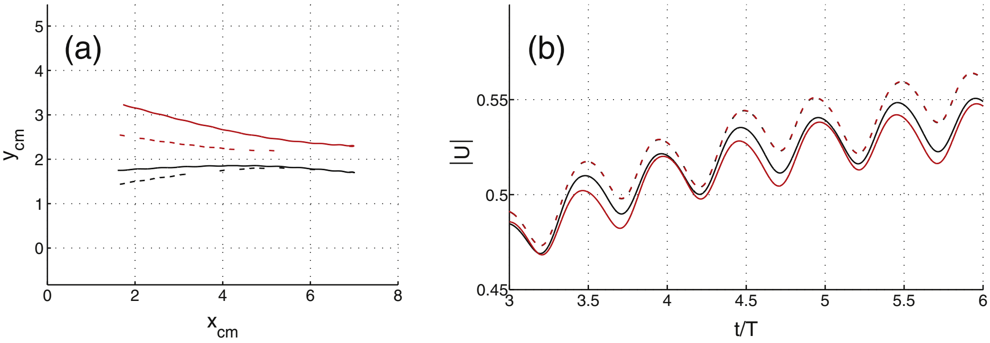

Fig. 19. Tandem fish: (a) trajectories of phase tandem (solid lines) and anti-phase tandem (dashed lines), (b) absolute velocities $|U|$ of phase tandem (solid lines) and anti-phase tandem (dashed lines). Coordinates and velocities are normalized by L.

izontal directions. For the aligned configuration the separation distance is $d_y = 0.4L$. A thorough study of the propulsive efficiencies of the tandem swimmers and five fish schools goes beyond the scope of this paper and is left for future work.

4.3. 3D swimmer

As mentioned before, the method is easily extended to three-dimensions. To validate this case, we present a simulation of a 3D self-propelled anguilliform swimmer and compare the results with the three-dimensional simulations presented by Kern and Koumoutsakos [22]. Details of the three-dimensional geometry are given in the appendix.

As in the 2D case, we set the body density equal to the fluid density, and therefore the baroclinic term drops out of the equations. The penalized Navier–Stokes equations in vorticity form then are given by:

$$\frac{\partial \boldsymbol{\omega}}{\partial t} + \nabla \cdot (\mathbf{u} : \boldsymbol{\omega}) = (\boldsymbol{\omega} \cdot \nabla)\mathbf{u} + \nu \nabla^2 \boldsymbol{\omega} + \lambda \nabla \times \chi_s (\mathbf{u}_s - \mathbf{u}), \tag{47}$$

where $\boldsymbol{\omega}$ is the vorticity vector and $\nabla \cdot (\mathbf{u} : \boldsymbol{\omega})$ denotes the vector with components $\partial/\partial x_i (u_i \omega_j)$.

The algorithm for the 3D case retains all steps of the 2D algorithm, the only addition is the time integration of the stretching term:

$$\frac{\partial \boldsymbol{\omega}^n_\lambda}{\partial t} = (\boldsymbol{\omega}^n_\lambda \cdot \nabla)\mathbf{u}^n_\lambda. \tag{48}$$

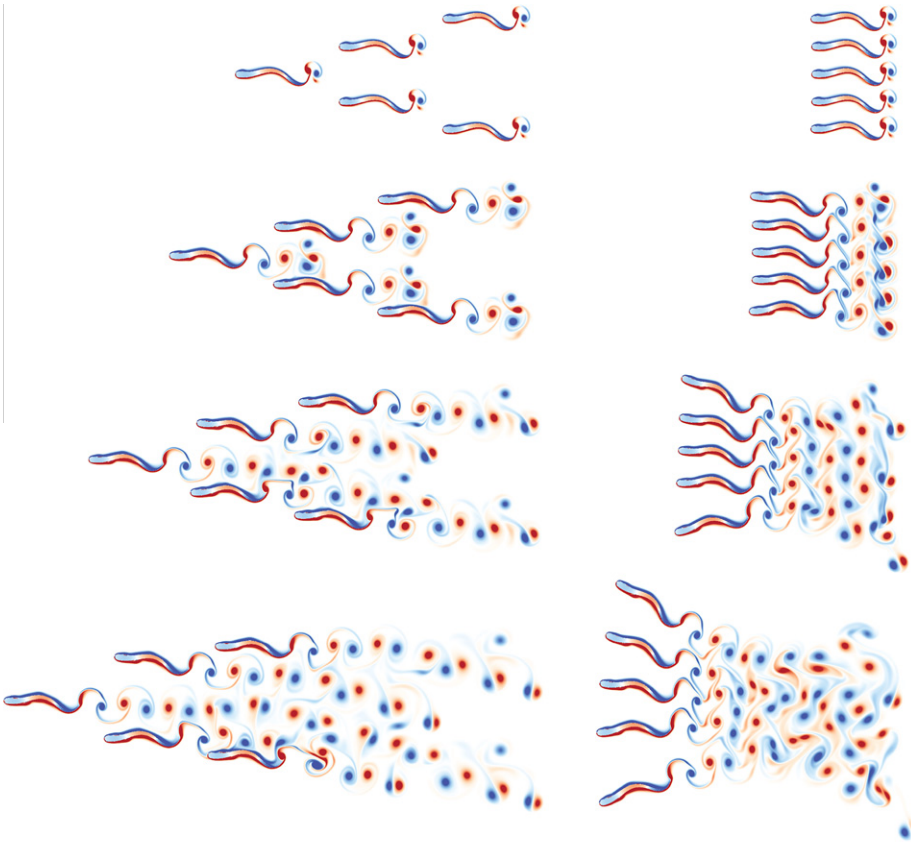

Fig. 20. Vorticity plots of five individuals school. (left) V-shape school, (right) aligned school. From top to bottom $t/T = 1.0$, $t/T = 2.8$, $t/T = 4.6$, $t/T = 6.4$.

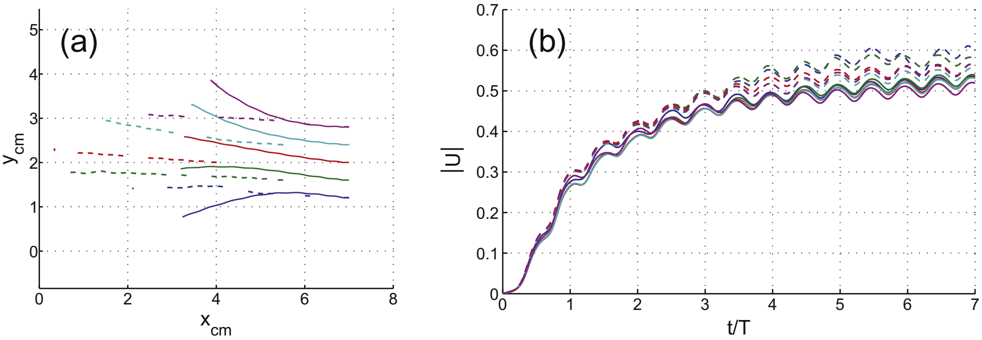

Fig. 21. Five fish school vorticity plots: (a) trajectories of aligned school (solid lines) and V-shape school (dashed lines), (b) absolute velocities $|U|$ of aligned school (solid lines) and V-shape school (dashed lines). Coordinates and velocities are normalized by L.

This equation is solved directly after Eq. (30) in the general algorithm 1. In the 3D case, we solve all spatial derivatives using central fourth order finite differences.

The computational parameters are the same as in the 2D case, in particular $LCFL = 0.01$ and $\lambda = 1e^4$. The grid spacing h is 1/512, with the fish length $L = 1$. To optimize usage of the computational resources, we start with a small domain containing only the fish, and let the domain grow in size during the computation to contain the full vorticity field. The fish is allowed translations only in the two-dimensional plane and rotations are restricted to those around the z-axis, as in the reference computations by Kern and Koumoutsakos [22]. We reproject the vorticity field onto a divergence-free field every 20 timesteps.

Fig. 22 shows the forward and lateral swimming velocity of the anguilliform swimmer compared with the reference results [22]. The current method gives a satisfactory agreement with the reference results. A volume rendering of the vorticity field is given in Fig. 23, showing the vortex rings in the swimmer wake.

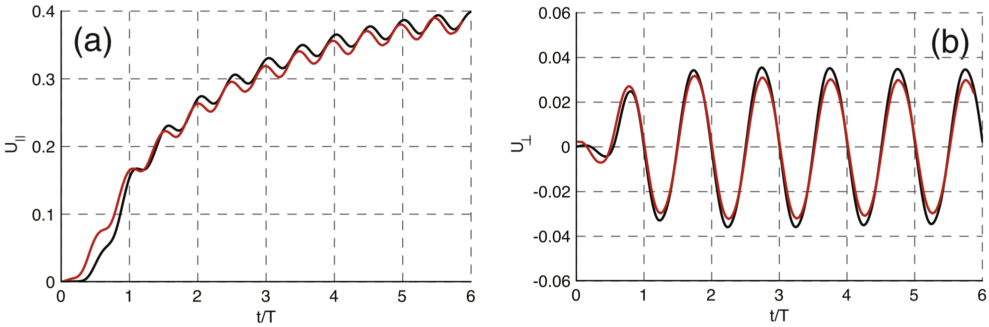

Fig. 22. 3D anguilliform swimmer: (a) Forward velocity $U_{\parallel}$, (b) Lateral velocity $U_{\perp}$. Red lines indicate 3D swimmer simulations with the proposed method. Black lines indicate the reference finite volume solution [22]. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

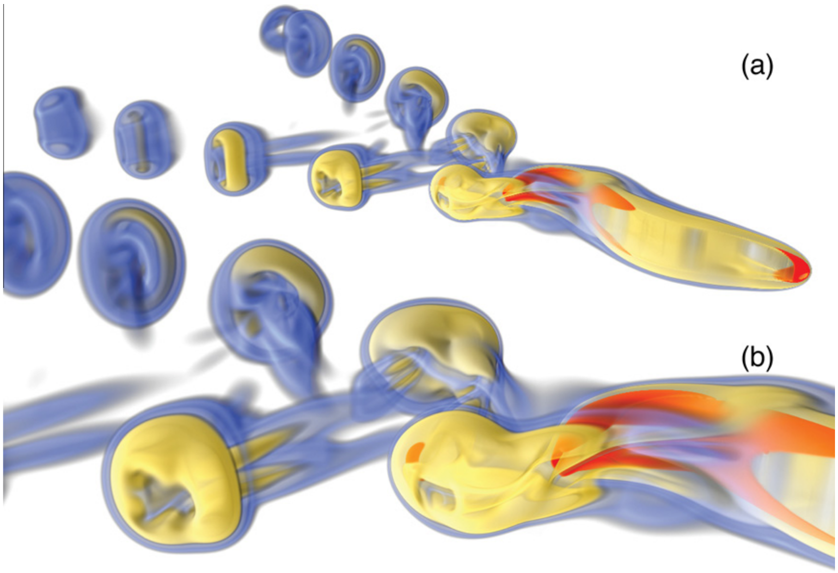

Fig. 23. Volume rendering of the $\boldsymbol{\omega}$-field showing the wake generated by the 3D anguilliform swimmer (a), and inset close to the tail (b).

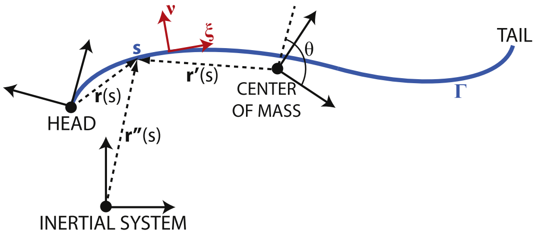

Fig. 24. Coordinate systems.

5. Conclusion

We have presented a vortex particle method for fluid–structure interaction with unsteady deforming geometries in unbounded domains. The method is based on the Brinkman penalization model to capture the feedback from the solid to the fluid and on the projection approach for the interaction from the fluid to the solid. The proposed algorithm can handle deformations characterized by non divergence-free velocity deformation fields, allowing the treatment of arbitrary deforming geometries. It produces an attractively clean algorithm and is shown to accurately reproduce reference simulations for both rigid and deforming solids. In particular, the method is applied to the simulation of self-propelled anguilliform swimmers immersed in a viscous, incompressible flow to demonstrate its suitability for the study of biological locomotion. The simplicity in the treatment of complex unsteady geometries makes it portable to parallel architecture such as GPUs and multicores. Ongoing work involves the coupling of simulations of multiple swimmers with bioinspired optimization algorithms.

Appendix A

The geometrical aspect of the two dimensional swimmer is characterized by the half width $w(s)$ of the body along its arclength s (midline). Following the work of Kern and Koumoutsakos [22] and Carling et al. [17], the half width $w(s)$ is defined as

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

where L is the body length, $w_h = s_b = 0.04L$, $s_t = 0.95L$ and $w_t = 0.01L$. In Kern and Koumoutsakos [22], the thickness reduction from head to tail is linear instead of quadratic for the 2D case, therefore we implemented the same modification here. For the 3D simulations, we use the elliptical height function $h(s)$ from Kern and Koumoutsakos [22]:

$$h(s) = b\sqrt{1 - \left(\frac{s-a}{a}\right)^2}$$

where $a = 0.51L$ and $b = 0.08L$. The three-dimensional shape is then formed by elliptical cross sections with half-lengths $w(s)$ and $h(s)$.

Several analytical descriptions for the motion of anguilliform swimmers have been proposed [63,22,17]. For the sake of generality, we present here a methodology that relies on the parameterization of the instantaneous curvature of the midline of the swimmer. This approach, used by Kern et al. [22], is general and faithful, to some extent, to the fish bio-mechanics. The curvature κ_s along the arc-length s is defined as

$$\kappa_s(s,t) = K(s) \cdot \sin\left[2\pi\left(\frac{t}{T} - \tau(s)\right)\right] \tag{49}$$

where $K(s)$ is the cubic spline interpolation through the m control points of curvature K_i ($i = 1,\ldots,m$), $\tau(s)$ is the phase shift along the body, t is the time, and T is the period. The phase shift $\tau(s)$ is linearly proportional to the arc-length s

$$\tau(s) = \frac{s}{L}\tau_{tail}. \tag{50}$$

Eq. (49) together with $w(s)$ and $h(s)$ determines the characteristic function χ_s, representing the swimmer, and the velocity deformation field of the body $\mathbf{u}_{DEF}$ (Section 2.3) in the Cartesian inertial frame of reference of the solver. This is done by integrating (through a first order explicit Euler scheme) the Frenet equations (Eq. (51)) for a 2D planar curve in the head frame of reference (Fig. 24). The main axis of the head frame of reference coincide with the tangent $\xi(s)$ and normal $\boldsymbol{\nu}(s)$ vectors of the mid-line Γ of the body evaluated at $s = 0$. In such a system of reference the initial conditions for the Frenet equations are $\boldsymbol{\nu}(0) = (0,1)$, $\xi(0) = (1,0)$ and $\mathbf{r}(0) = \mathbf{0}$.

$$\frac{\partial \xi}{\partial s} = \kappa_s \boldsymbol{\nu} \quad \frac{\partial \boldsymbol{\nu}}{\partial s} = -\kappa_s \xi \quad \frac{\partial \mathbf{r}}{\partial s} = \xi \tag{51}$$

In order to obtain an analytical expression of the deformation velocity of the mid-line, we take the time derivative of the Frenet equations (Eq. (52)). We integrate them (Eq. (52)) in the head frame of reference with initial conditions $\dot{\boldsymbol{\nu}}(0) = \mathbf{0}$, $\dot{\xi}(0) = \mathbf{0}$, $\dot{\mathbf{r}}(0) = \mathbf{0}$ and $\xi(s)$ and $\boldsymbol{\nu}(s)$ coming from the integration of (Eq. (51)).

$$\frac{\partial \dot{\xi}}{\partial s} = \dot{\kappa}_s \boldsymbol{\nu} + \kappa_s \dot{\boldsymbol{\nu}} \quad \frac{\partial \dot{\boldsymbol{\nu}}}{\partial s} = -\dot{\kappa}_s \xi - \kappa_s \dot{\xi} \quad \frac{\partial \dot{\mathbf{r}}}{\partial s} = \dot{\xi} \tag{52}$$

The previous steps allow us to calculate the position and the deformation velocity of the mid-line Γ in the head frame of reference. In order to obtain a deformation field for the whole swimming body, it is necessary to extend position and velocities away from the mid-line. This is achieved constructing a deformable mesh whose nodes are uniquely identified by the pair of indices (i,j). Nodes positions off mid-line are computed using the normal vector $\boldsymbol{\nu}(s)$

$$\mathbf{r}_{\mathcal{M}}(i,j) = \mathbf{r}(i \cdot \Delta s, t) + j \cdot \Delta s \cdot \boldsymbol{\nu}(i \cdot \Delta s, t) \tag{53}$$

where $\mathbf{r}_{\mathcal{M}}(i,j)$ is the position of the mesh node identified by the index i (running along the mid-line) and by the index j (running normally away from the mid-line), $\mathbf{r}$ describes the position of the mid-line, and Δs is the spacing of the mid-line discretization (Fig. 25). Successively the local deformation velocity is computed at each grid node location by taking the time derivative of Eq. (53)

$$\dot{\mathbf{r}}_{\mathcal{M}}(i,j) = \dot{\mathbf{r}}(i \cdot \Delta s, t) + j \cdot \Delta s \cdot \dot{\boldsymbol{\nu}}(i \cdot \Delta s, t) \tag{54}$$

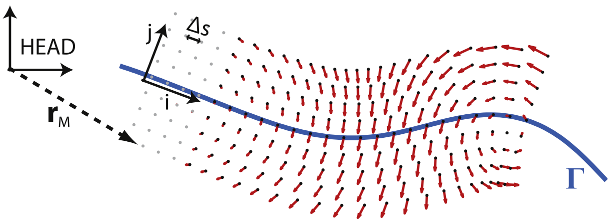

Fig. 25. Deformable mesh and resulting deformation velocity field.

The characteristic function $\chi_{\mathcal{M}}$ $(0 \leqslant \chi_{\mathcal{M}} \leqslant 1)$ is also evaluated at each grid node location by

$$\chi_{\mathcal{M}} = \begin{cases} 0 & d < -\epsilon \\ \frac{1}{2}\left[1 + \frac{d}{\epsilon} + \frac{1}{\pi}\sin\left(\pi\frac{d}{\epsilon}\right)\right] & |d| \leqslant \epsilon \\ 1 & d > \epsilon \end{cases}$$

where d is the signed distance function from the surface of the swimming body (negative outside, positive inside) and ϵ is the mollification width. In the head frame of reference, we calculate center of mass position $\mathbf{r}_{cm}$ and velocity $\dot{\mathbf{r}}_{cm}$

$$\mathbf{r}_{cm} = \frac{\int_\Sigma \rho_s \chi_{\mathcal{M}} \mathbf{r} d\mathbf{x}}{\int_\Sigma \rho_s \chi_{\mathcal{M}} d\mathbf{x}}, \quad \dot{\mathbf{r}}_{cm} = \frac{\int_\Sigma \rho_s \chi_{\mathcal{M}} \dot{\mathbf{r}} d\mathbf{x}}{\int_\Sigma \rho_s \chi_{\mathcal{M}} d\mathbf{x}} \tag{55}$$

where Σ is the entire simulation domain. Center of mass location and velocity are then used to transform the swimmer representation into the center of mass frame of reference $(\mathbf{r}', \dot{\mathbf{r}}^{*\prime})$. Velocities $\dot{\mathbf{r}}^{*\prime}$ are then corrected to account for the vanishing rotational impulse through the angular velocity ω

$$L = \int_\Sigma \mathbf{r}' \times \rho_s \chi_{\mathcal{M}} \dot{\mathbf{r}}^{*\prime} d\mathbf{x}, \quad J_s = \int_\Sigma \mathbf{r}'^2 \rho_s \chi_{\mathcal{M}} d\mathbf{x}, \quad \omega = \frac{L}{J_s}, \quad \dot{\mathbf{r}}' = \dot{\mathbf{r}}^{*\prime} + \omega \times \mathbf{r}' \tag{56}$$

where L and J_s represent respectively the angular momentum and scalar momentum of inertia. An additional coordinate transformation is needed to bring the swimmer representation into the simulation frame of reference $(\mathbf{r}'', \dot{\mathbf{r}}'')$. The above steps ensure that the net force and momentum acting on the body will exclusively come from the hydrodynamics. The deformable mesh is subdivided into triangular patches and we indicate with $\mathcal{P}_I(\mathbf{b}, \mathbf{a}, \boldsymbol{\alpha})$ the bilinear interpolant of the quantity $\boldsymbol{\alpha}$ at the location $\mathbf{b}$ from the vertices of the triangular patches $\mathbf{a}$. Velocities $(\dot{\mathbf{r}}'')$ and characteristic function $(\chi_{\mathcal{M}})$ stored at the node locations $(\mathbf{r}'')$ are then interpolated onto the Cartesian computational grid of the flow solver

$$\chi_s = \mathcal{P}_I(\mathbf{x}, \mathbf{r}'', \chi_{\mathcal{M}}) = \mathcal{P}_I(\mathbf{x}, \mathcal{R}(\theta)\mathbf{r}' + \mathbf{x}_{cm}, \chi_{\mathcal{M}}) \tag{57}$$

$$\mathbf{u}_{DEF} = \mathcal{P}_I(\mathbf{x}, \mathbf{r}'', \dot{\mathbf{r}}'') = \mathcal{P}_I(\mathbf{x}, \mathcal{R}(\theta)\mathbf{r}' + \mathbf{x}_{cm}, \mathcal{R}(\theta)\dot{\mathbf{r}}') \tag{58}$$

where $\mathbf{x}$ are the computational grid nodes positions, $\mathbf{x}_{cm}$ and θ are, respectively, the coordinates of the body center of mass and the rotation angle in the simulation frame of reference, while $\mathcal{R}$ is the 2D counterclockwise rotation matrix. Such construction ensures that $\mathbf{u}_{DEF}$ does not contain any translation nor rotational component.

References

[1] M.S. Triantafyllou, G.S. Triantafyllou, D.K.P. Yue, Hydrodynamics of fishlike swimming, Annual Review Fluid Mechanics 32 (2000). 33.

[2] R. Mittal, Computational modeling in biohydrodynamics: trends, challenges, and recent advances, IEEE Journal of Oceanic Engineering 29 (2004) 95–604.

[3] A. Gilmanov, F. Sotiropoulos, A hybrid cartesian/immersed boundary method for simulating flows with 3d, geometrically complex, moving bodies, Journal of Computational Physics 207 (2005) 457–492.

[4] A.A. Shirgaonkar, M.A. MacIver, N.A. Patankar, A new mathematical formulation and fast algorithm for fully resolved simulation of self-propulsion, Journal of Computational Physics 228 (2009) 2366–2390.

[5] H. Dong, M. Bozkurttas, R. Mittal, P. Madden, G.V. Lauder, Computational modelling and analysis of the hydrodynamics of a highly deformable fish pectoral fin, Journal of Fluid Mechanics 645 (2010) 345–373.

[6] E.D. Tytell, C.Y. Hsu, T.L. Williams, A.H. Cohen, L.J. Fauci, Interactions between internal forces, body stiffness, and fluid environment in a neuromechanical model of lamprey swimming, Proceedings of the National Academy of Sciences of the United States of America 107 (2010) 19832–19837.

[7] E.D. Tytell, I. Borazjani, F. Sotiropoulos, T.V. Baker, E.J. Anderson, G.V. Lauder, Disentangling the functional roles of morphology and motion in the swimming of fish, Integrative and Comparative Biology 50 (2010) 1140–1154.

[8] I. Borazjani, F. Sotiropoulos, Numerical investigation of the hydrodynamics of anguilliform swimming in the transitional and inertial flow regimes, Journal of Experimental Biology 212 (2009) 576–592.

[9] I. Borazjani, F. Sotiropoulos, Numerical investigation of the hydrodynamics of carangiform swimming in the transitional and inertial flow regimes, Journal of Experimental Biology 211 (2008) 1541–1558.

[10] U.K. Muller, J. Smit, E.J. Stamhuis, J.J. Videler, How the body contributes to the wake in undulatory fish swimming: flow fields of a swimming eel (anguilla anguilla), Journal of Experimental Biology 204 (2001) 2751–2762.

[11] E.D. Tytell, The hydrodynamics of eel swimming ii. Effect of swimming speed, Journal of Experimental Biology 207 (2004) 3265–3279.

[12] E.D. Tytell, G.V. Lauder, The hydrodynamics of eel swimming - i. Wake structure, Journal of Experimental Biology 207 (2004) 1825–1841.

[13] M.S. Triantafyllou, G.S. Triantafyllou, An efficient swimming machine, Scientific American 272 (1995) 64–70.

[14] S.D. Kelly, R.J. Mason, C.T. Anhalt, R.M. Murray, J.W. Burdick, Modelling and experimental investigation of carangiform locomotion for control, 1998.

[15] A.J. Ijspeert, A. Crespi, D. Ryczko, J.M. Cabelguen, From swimming to walking with a salamander robot driven by a spinal cord model, Science 315 (2007) 1416–1420.

[16] J. Gray, Studies in animal locomotion i. The movement of fish with special reference to the eel, Journal of Experimental Biology 10 (1933) 88–104.

[17] J. Carling, T.L. Williams, 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 201 (1998) 3143–3166.

[18] M. Bergmann, A. Iollo, Modeling and simulation of fish-like swimming, Journal of Computational Physics 230 (2011) 329–348.

[19] J.D. Eldredge, Numerical simulations of undulatory swimming at moderate Reynolds number, Bioinspiration and Biomimetics 1 (2006) S19–S24.

[20] L.P. Zhang, X.H. Chang, X.P. Duan, Z.Y. Wang, H.X. Zhang, A block lu-sgs implicit unsteady incompressible flow solver on hybrid dynamic grids for 2d external bio-fluid simulations, Computers and Fluids 38 (2008) 290–308.

[21] K.S. Yeo, S.J. Ang, C. Shu, Simulation of fish swimming and manoeuvring by an svd-gfd method on a hybrid meshfree-Cartesian grid, Computers and Fluids 39 (2010) 403–430.

[22] S. Kern, P. Koumoutsakos, Simulations of optimized anguilliform swimming, Journal of Experimental Biology 209 (2006) 4841–4857.

[23] H. Liu, K. Kawachi, A numerical study of undulatory swimming, Journal of Computational Physics 155 (1999) 223–247.

[24] S.E. Hieber, P. Koumoutsakos, An immersed boundary method for smoothed particle hydrodynamics of self-propelled swimmers, Journal of Computational Physics 227 (2008) 8636–8654.

[25] U.K. Muller, J.G.M. van den Boogaart, J.L. van Leeuwen, Flow patterns of larval fish: undulatory swimming in the intermediate flow regime, Journal of Experimental Biology 211 (2008) 196–205.

[26] C. Conca, J. San Martin, M. Tucsnak, Existence of solutions for the equations modelling the motion of a rigid body in a viscous fluid, Communications in Partial Differential Equations 25 (2000) 1019–1042.

[27] R. Mittal, G. Iaccarino, Immersed boundary methods, Annual Review of Fluid Mechanics 37 (2005) 239–261.

[28] R. Glowinski, T.W. Pan, T.I. Hesla, D.D. Joseph, J. Periaux, A fictitious domain approach to the direct numerical simulation of incompressible viscous flow past moving rigid bodies: application to particulate flow, Journal of Computational Physics 169 (2001) 363–426.

[29] L. Lee, R.J. Leveque, An immersed interface method for incompressible Navier–Stokes equations, SIAM Journal on Scientific Computing 25 (2003) 832–856.

[30] M.N. Linnick, H.F. Fasel, A high-order immersed interface method for simulating unsteady incompressible flows on irregular domains, Journal of Computational Physics 204 (2005) 57–192.

[31] C.S. Peskin, Numerical-analysis of blood-flow in heart, Journal of Computational Physics 25 (1977) 220–252.

[32] R. Mittal, H. Dong, M. Bozkurttas, F.M. Najjar, A. Vargas, A. von Loebbecke, A versatile sharp interface immersed boundary method for incompressible flows with complex boundaries, Journal of Computational Physics 227 (2008) 4825–4852.

[33] I. Borazjani, L. Ge, F. Sotiropoulos, Curvilinear immersed boundary method for simulating fluid structure interaction with complex 3d rigid bodies, Journal of Computational Physics 227 (2008) 7587–7620.

[34] E. Arquis, J.P. Caltagirone, On the hydrodynamical boundary-conditions along a fluid layer porous-medium interface – application to the case of free convection, Comptes Rendus de l Academie des Sciences Serie II 299 (1984) 1–4.

[35] P. Angot, C.H. Bruneau, P. Fabrie, A penalization method to take into account obstacles in incompressible viscous flows, Numerische Mathematik 81 (1999) 497–520.

[36] G.H. Keetels, U. D'Ortona, W. Kramer, H.J.H. Clercx, K. Schneider, G.J.F. van Heijst, Fourier spectral and wavelet solvers for the incompressible Navier–Stokes equations with volume-penalization: convergence of a dipole-wall collision, Journal of Computational Physics 227 (2007) 919–945.

[37] D. Rossinelli, M. Bergdorf, G.-H. Cottet, P. Koumoutsakos, Gpu accelerated simulations of bluff body flows using vortex particle methods, Journal of Computational Physics 229 (2010) 3316–3333.

[38] O.V. Vasilyev, N.K.R. Kevlahan, Hybrid wavelet collocation-Brinkman penalization method for complex geometry flows, International Journal for Numerical Methods in Fluids 40 (2002) 531–538.

[39] R.J. Di Perna, P.L. Lions, Ordinary differential equations transport theory and Sobolev spaces, Inventiones Mathematicae 98 (1989) 511–547.

[40] J. Janela, A. Lefebvre, B. Maury, A penalty method for the simulation of fluid-rigid body interaction, ESAIM: Proceedings 14 (2005) 115–123.

[41] C. Bost, G.-H. Cottet, E. Maitre, Convergence analysis of a penalization method for the three-dimensional motion of a rigid body in an incompressible viscous fluid, SIAM Journal of Numerical Analysis 48 (2010) 1313–1337.

[42] I. Borazjani, F. Sotiropoulos, On the role of form and kinematics on the hydrodynamics of self-propelled body/caudal fin swimming, Journal of Experimental Biology 213 (2010) 89–107.

[43] M. Coquerelle, G.H. Cottet, A vortex level set method for the two-way coupling of an incompressible fluid with colliding rigid bodies, Journal of Computational Physics 227 (2008) 9121–9137.

[44] D. Shiels, A. Leonard, A. Roshko, Flow-induced vibration of a circular cylinder at limiting structural parameters, Journal of Fluids and Structures 15 (2001) 3–21.

[45] J.D. Eldredge, Dynamically coupled fluid-body interactions in vorticity-based numerical simulations, Journal of Computational Physics 227 (2008) 9170–9194.

[46] N.A. Patankar, N. Sharma, A fast projection scheme for the direct numerical simulation of rigid particulate flows, Communications in Numerical Methods in Engineering 21 (2005) 419–432.

[47] P. Koumoutsakos, A. Leonard, High-resolution simulations of the flow around an impulsively started cylinder using vortex methods, Journal of Fluid Mechanics 296 (1995) 1–38.

[48] G.H. Cottet, P. Poncet, Advances in direct numerical simulations of 3d wall-bounded flows by vortex-in-cell methods, Journal of Computational Physics 193 (2004) 136–158.

[49] P. Ploumhans, G.S. Winckelmans, Vortex methods for high-resolution simulations of viscous flow past bluff bodies of general geometry, Journal of Computational Physics 165 (2000) 354–406.

[50] P. Koumoutsakos, Inviscid axisymmetrization of an elliptical vortex, Journal of Computational Physics 138 (1997) 821–857.

[51] P. Koumoutsakos, Multiscale flow simulations using particles, Annual Review Fluid Mechanics 37 (2005).

[52] J.J. Monaghan, Extrapolating b splines for interpolation, Journal of Computational Physics 60 (1985) 253–262.

[53] R. Hockney, J. Eastwood, Computer Simulation Using Particles, McGraw-Hill, 1981.

[54] P. Chatelain, P. Koumoutsakos, A Fourier-based elliptic solver for vortical flows with periodic and unbounded directions, Journal of Computational Physics 229 (2010) 2425–2431.

[55] G.-H. Cottet, P. Koumoutsakos, Vortex Methods, Theory and Practice, Cambridge University Press, 2000.

[56] G. Carbou, P. Fabrie, Boundary layer for a penalization method for viscous incompressible flow, Advances in Difference Equations 8 (2003) 1453–1480.

[57] G. Winckelmans, Vortex Methods, vol. 3, John Wiley and Sons, 2004.

[58] A.K. Tornberg, B. Engquist, Regularization techniques for numerical approximation of pdes with singularities, Journal of Scientific Computing 19 (2003) 527–552.

[59] 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 604 (2008) 33–53.

[60] S.B. Field, M. Klaus, M.G. Moore, F. Nori, Chaotic dynamics of falling disks, Nature 388 (1997) 252–254.

[61] A. Andersen, U. Pesavento, Z.J. Wang, Unsteady aerodynamics of fluttering and tumbling plates, Journal of Fluid Mechanics 541 (2005) 65–90.

[62] I. Akhtar, R. Mittal, G.V. Lauder, E. Drucker, Hydrodynamics of a biologically inspired tandem flapping foil configuration, Theoretical and Computational Fluid Dynamics 21 (2007) 155–170.

[63] T. Pedley, S. Hill, Large-amplitude undulatory fish swimming: fluid mechanics coupled to internal mechanics, Journal of Experimental Biology 202 (1999) 3431–3438.