Journal of Computational Physics 228 (2009) 2366–2390

Contents lists available at ScienceDirect

Journal of Computational Physics

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

A new mathematical formulation and fast algorithm for fully resolved simulation of self-propulsion

Anup A. Shirgaonkar [a,1], Malcolm A. MacIver [a,b], Neelesh A. Patankar [a,*]

[a] Department of Mechanical Engineering, R.R. McCormick School of Engineering and Applied Science, Northwestern University, Evanston, IL 60208, USA
[b] Department of Biomedical Engineering, R.R. McCormick School of Engineering and Applied Science, Northwestern University, Evanston, IL 60208, USA

ARTICLE INFO

Article history:
Received 21 April 2008
Received in revised form 22 October 2008
Accepted 2 December 2008
Available online 24 December 2008

Keywords:
Self-propulsion
Biolocomotion
DNS
Fully resolved simulation (FRS)
Immersed boundary method
Distributed Lagrange multiplier method

ABSTRACT

We present a computational algorithm for fully resolved numerical simulation (FRS) of rigid and deforming bodies moving in fluids. Given the deformation of the body in its own reference frame, the method solves for the swimming velocity of the body together with the surrounding flow field, and the hydrodynamic forces on the body. We provide the mathematical foundation of the algorithm based on distributed Lagrange multipliers, and show that it naturally connects with vortex methods through a vorticity source at the interface. We demonstrate applications to rigid and flexible bodies, membranes, and bodies with a propelling membrane attached to them. In contrast to some existing methods, the swimming velocity of the body is not prescribed but is computed along with the forces, without requiring a body-fitted grid. The algorithm is designed to be fast, efficient, and easy to implement in existing fluid dynamics codes for practical solid–fluid problems in engineering and biology.

1. Introduction

Swimming fish are a biological model system for understanding the interconnections between mechanics, physiology and neuronal activity during locomotion. Musculoskeletal, neuronal, and sensory systems interact closely with each other resulting in the complex process of locomotion [1–3]. To gain insights into biolocomotion, a desirable approach is to use reduced order models for such multiple systems in an integrative setting to allow the study of how each function performs coherently along with the others. The empirical data or numerical simulations required for development and validation of reduced order models can become complex and expensive due to the detailed mechanics and neural function of the organism. In particular, the hydrodynamics of aquatic locomotion involves multiple phenomena such as viscous effects, boundary layer separation and vortex shedding which require elaborate measurements or computations to characterize sufficiently. Perhaps due to this complexity, many unanswered questions still remain, such as (i) how can the swimming velocity of an organism be determined from the movement of its propulsive surfaces? (ii) What is the efficiency of various modes of swimming? (iii) What are the wake signatures of the various modes of swimming, and are they indicative of the swimming efficiency?

A standard reduced order model approach relies on equating *a priori* estimates of drag and thrust on a fish at its swimming velocity. This has led to many conflicting results. For instance, for a given swimming velocity, the drag on swimming fish is often estimated to be higher than that found in towing experiments [4,5]. However, there are also measurements on an actively swimming robotic vehicle which show that the power needed to self-propel the robot is reduced by half compared to the power needed to tow the robot with the body straight and rigid [6]. Regarding the thrust at a given swimming speed,

[*] Corresponding author. Tel.: +1 847 491 3021; fax: +1 847 491 3915.
E-mail address: n-patankar@northwestern.edu (N.A. Patankar).
[1] Present address: Department of Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, MA 02139, USA.

0021-9991/$ - see front matter © 2008 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2008.12.006

Lighthill's theory [5,7,8] is found to over predict thrust by a factor of up to 3. One of the root causes of the conflicting results on the drag and thrust on swimming fish is that in the case of self-propulsion, drag and thrust cannot be properly separated [4,9]. There is no net time-averaged hydrodynamic force acting on the surface of the fish during steady swimming. One may artificially split the net hydrodynamic force into two parts – thrust and drag. However, locations where thrust and drag are generated on the fish body can be intermingled [10]. Thus, there is no general way to *a priori* separate and predict both thrust and drag. This implies that the approach of balancing thrust and drag to obtain the swimming velocity is unsuitable. Fully resolved simulation (FRS) of self-propulsion is useful in this case to obtain the swimming velocity and at the same time obtain the force distribution on the surface of the fish. In our discussion, fully resolved simulation implies that the fluid–solid coupling is not modeled (e.g. using drag models) but instead the flow around the swimming body is fully resolved.

The separation of thrust and drag also produces difficulties in coming up with a consistent definition of swimming efficiency. Swimming efficiency is often given in terms of the Froude efficiency, which is defined as the ratio of work done by the thrust force to the total power input. Once again defining thrust is not straightforward when thrust and drag are intermingled as discussed above. As a result, there is no clear understanding of the comparative efficiency of the different swimming modes [4,10]. FRS of self-propulsion can provide the net mechanical energy cost of swimming and lead to useful efficiency measures such as those used by Kern and Koumoutsakos [11] to compare swimming efficiency across various swimming styles.

Usually, strong downstream flow in the wake is interpreted to signify thrust as well as energy lost in the wake [4,12–15]. However, the wake of steadily swimming eels is found to lack substantial downstream flow [10]. This raises a fundamental question: what is the signature of thrust in wakes of various swimming modes, and how is it correlated with swimming efficiency? Tytell and Lauder [10] speculate that wakes probably show a gradation from those of mackerel, for example, which primarily flap their caudal fins, to those of eels, which deform their entire body. The wake could thus be indicative of the propulsion mechanism which can be studied by FRS of self-propulsion. Such simulations will also make it possible to find swimming gaits that minimize wakes but are also efficient.

Each of the above issues could be addressed by sophisticated experimental techniques or through FRS schemes. Often experimental techniques are intrusive, making measurements in the natural state of the system difficult, especially when force measurements are involved [16]. Hence high-fidelity numerical simulations can greatly contribute to the study of the fluid dynamics of biolocomotion, leading to better predictive models for the motion of a variety of organisms.

Modeling the hydrodynamics of moving organisms can be challenging due to the strong coupling between the body and the surrounding fluid, which are governed by widely separate material characteristics such as stiffness and viscosity. To model this, there are two approaches of interest. In the first approach, referred to as the elastohydrodynamic approach, the muscle activation would be given. Then, the elastic equations for the body would be solved together with the equations of motion of the body and the fluid. This approach will require detailed knowledge of muscle anatomy, physiology, and muscle activation patterns, as well as the elastic properties of the fish body and fins. This requirement is far beyond what is understood for the best characterized systems, and is an open area of research. Currently, such approaches rely on simplifications of the elastic and hydrodynamic equations [17,18].

The second approach, referred to as the hydrodynamic approach, is to find the swimming velocity resulting from a given deforming motion (kinematics) of the body and/or fins. This is a purely hydrodynamic problem that excludes the need to solve elasticity equations. Typical theories in aquatic locomotion fall into this category. These theories assume that rhythmic or non-rhythmic deformation kinematics are given based on experimental motion capture data. The hydrodynamic approach does not ignore the net elastic effect. It is embedded in the body deformation kinematics, within the accuracy with which the 3D kinematic data was obtained. As more high-precision 3D kinematics data of aquatic organisms becomes available (e.g. [19,20]), the hydrodynamic approach can use it to obtain high-accuracy predictions of their flow field and swimming velocities. Here, we provide an algorithm for simulating free-swimming organisms using the hydrodynamic approach.

Some of the prior FRS studies involve simulating the flow around deforming organisms at a *specified* constant swimming velocity (e.g. [21–25]). These simulations provide valuable insight into the fluid mechanics of aquatic locomotion, but are limited by the assumption of constant *specified* swimming velocity. Computations based on this assumption miss two crucial phenomena occurring in real free-swimming organisms. First, the swimming velocity is seldom constant, but almost always oscillatory with a constant mean value in the steady swimming state. Second, these temporal changes in the swimming velocity are likely to affect the flow characteristics around the body both qualitatively and quantitatively. To include these details, FRS should also solve for the swimming velocity instead of taking it is as an input. This will account for the two-way coupling between the flow field around the body and its swimming velocity.

There are FRS studies that present simulations of self-propelling organisms, i.e. they do not specify the swimming velocity but obtain it as a solution. These studies include a self-propelling eel [27,11], a flagellar structure with a head [28], and a copepod [23]. In these approaches, typically, the flow field is advanced first and then additional equations of motion for the body are solved based on the hydrodynamic force on the body. This explicit coupling can cause numerical instability due to the non-linear nature of the fluid dynamics equations and the angular momentum equation for the body [29].

Kern and Koumoutsakos [11] recently performed FRS of self-propelling eels with detailed body geometry and reproduced flow features observed in experiments of Tytell and Lauder [10]. In their work the geometric detail was made possible using a body-fitted grid. A body-fitted grid has to be regenerated periodically in order to conform to the changing configuration of the eel body. Body-fitted grid methods are difficult to apply to problems with arbitrarily shaped complex objects and multibody configurations. This is because in these cases, grid regeneration can significantly add to the computational cost. Additionally, fluid solvers are based on either unstructured or curvilinear structured grids, both of which can be computationally expensive compared to fast solvers for fixed structured rectilinear grids. Good scalability is also challenging in parallel implementations that are often essential to solve high Reynolds number problems. This is known from prior work on fully resolved simulation of rigid particulate flows where body-fitted grid methods were found to be computationally expensive compared to immersed boundary method-based fixed grid techniques [29,30].

Vortex particle methods provide an alternative to curvilinear grids, and simulation of geometrically simple self-propelling objects have been carried out using these methods [26,31]. For complex configurations, however, the vortex elements need remeshing.

Motivated by the requirement of being able to efficiently handle arbitrarily shaped swimming bodies, we present a constraint-based formulation for the problem of self-propulsion. Then we present an efficient algorithm to solve the new set of governing equations. The new formulation and the algorithm provide a rigorous basis for an immersed boundary implementation of the self-propulsion problem. It allows implicit coupling between the fluid and the fish body without the need to solve additional equations of motion for the fish body. Thus, the new approach addresses several of the computational challenges discussed above. The proposed technique can also easily handle multiple bodies and deformation patterns. The algorithm extends the approach for rigid particulate flows by Patankar et al. [32] and Patankar [33] to self-propulsion of deforming bodies [34,35].

In the following, Section 2 gives the mathematical formulation. Section 3 presents the algorithm and its implementation. Section 4 demonstrates the application of the algorithm for rigid and flexible membranes, rigid and flexible bodies, and a self-propelling black ghost knifefish as an example of a rigid body with an attached flexible fin. Section 5 gives concluding remarks.

2. Mathematical formulation

We present a mathematical formulation for an algorithm applicable to a class of problems which can have a solid body (flexible or rigid) moving in a fluid either by the fluid motion (passive) or by self-propulsion (active), or some combination of the two. In the case of a flexible body, it is assumed that its deformation is specified. The objective is to obtain the flow field around the body, the swimming velocity of the body, and the forces on the body. For self-propulsion, the body is either flexible so that it can generate an undulatory motion or has a flexible membrane attached to it to generate such motion. An example of the former is an eel, while knifefish—which swim by undulating a long midine fin while keeping the body straight—is an example of the latter. All undulatory motion of the body and the membrane are assumed to be given, such as through 3D kinematics measurements. In the hydrodynamic approach, the kinematics can also be obtained separately by carrying out an *a priori*, independent calculation of velocity using an elastic response model of the body when the muscular forces are given, without considering the fluid mechanics. The hydrodynamic approach will always take the kinematics of deformation as the input. Thus, the source of this kinematic data can be direct velocity measurements or a separate elastic model, depending on whether the forces or the velocity of the organism are available from measurements.

We consider a generic "organism" consisting of a "body" domain Ω_s. The total computational domain, Ω, includes Ω_s and the fluid domain $\Omega_f = (\Omega - \Omega_s)$. The boundary of Ω is $\partial\Omega$, and that of Ω_s is $\partial\Omega_s$. Although we consider a single immersed body for convenience, the formulation and the algorithm are easily extendable to multiple bodies, i.e. when Ω_s is not simply connected. The gravitational body force will be ignored in the following for simplicity of exposition. However, if gravity is present then a body force term can be added to the momentum equation. The fluid domain is governed by the momentum equation,

$$\rho_f\left(\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u}\cdot\nabla\boldsymbol{u}\right) = \nabla\cdot\boldsymbol{\sigma} \quad \text{in } \Omega_f, \tag{2.1}$$

the continuity equation,

$$\nabla\cdot\boldsymbol{u} = 0 \quad \text{in } \Omega_f, \tag{2.2}$$

the boundary condition,

$$\boldsymbol{u} = \boldsymbol{u}_{\partial\Omega} \quad \text{on } \partial\Omega, \tag{2.3}$$

the interface conditions on the fluid–solid boundary,

$$\boldsymbol{u} = \boldsymbol{u}_i \quad \text{and} \quad \boldsymbol{\sigma}\cdot\hat{\boldsymbol{n}} = \boldsymbol{t}_{\partial\Omega_s} \quad \text{on } \partial\Omega_s, \tag{2.4}$$

and the initial condition,

$$\boldsymbol{u}(\boldsymbol{x}, t=0) = \boldsymbol{u}_o(\boldsymbol{x}) \quad \text{in } \Omega_f(t=0). \tag{2.5}$$

Here ρ_f and $\boldsymbol{u}$ are the fluid density and velocity, $\hat{\boldsymbol{n}}$ is the outward normal unit vector to $\partial\Omega_s$, outward being away from the body, and $\boldsymbol{\sigma}$ is the stress tensor inside the fluid. $\boldsymbol{t}_{\partial\Omega_s}$ is the traction vector acting from the fluid on the solid body surface. $\boldsymbol{u}_i$ is the (unknown) velocity of the solid–fluid interface. The initial velocity $\boldsymbol{u}_o$ is required to satisfy Eq. (2.2) and the boundary velocity in Eq. (2.3) should satisfy the compatibility condition due to Eq. (2.2) at all times. The Dirichlet boundary condition (2.3) is used on the external boundary, but the formulation can be easily extended to consider other boundary conditions. The fluid equations (2.1)–(2.5) should be solved coupled with the equations of motion of the solid. The fluid and solid equations are coupled through the interface conditions in Eq. (2.4).

For an incompressible flow, $\boldsymbol{\sigma} = -p\boldsymbol{I} + \boldsymbol{\tau}$, where p is the mechanical pressure and $\boldsymbol{\tau}$ is the deviatoric stress tensor. p can be regarded as a Lagrange multiplier associated with the incompressibility constraint [36, p. 472]. For an incompressible Newtonian fluid with viscosity μ, $\boldsymbol{\tau} = 2\mu\boldsymbol{D}(\boldsymbol{u})$, where $\boldsymbol{D}(\boldsymbol{u}) = \frac{1}{2}(\nabla\boldsymbol{u} + \nabla\boldsymbol{u}^T)$ is the deformation rate tensor. For a viscoelastic fluid, $\boldsymbol{\tau}$ is constituted by both the viscous and elastic stresses in the fluid.

The motion of the material inside the solid domain is governed by momentum balance. For an elastic solid, the momentum equation is expressed as the Navier equation [37, p. 73]. This is called the elastohydrodynamic approach. When the solid is stiff relative to the fluid, i.e. its elastic and/or viscoelastic moduli are high, the coupled solution becomes stiff. This may cause instability, inaccuracy, or excessive inefficiency in numerical solutions of the equations. However, when the deformation velocity of the body is specified, which we referred to as the hydrodynamic approach earlier, a different problem can be formulated. In this case it is more desirable to consider the solid as a fluid with an additional constraint associated with the specified deformation velocity field within the body. We will formulate this approach below. It is noted that when the body has no deformations, i.e. it is rigid, the approach below reduces to that developed previously for rigid particulate flows [30,32].

Considering the solid as a fluid, the governing equations at each point within the body are the momentum equation,

$$\rho_s\left(\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u}\cdot\nabla\boldsymbol{u}\right) = \nabla\cdot\boldsymbol{\sigma}_s \quad \text{in } \Omega_s, \tag{2.6}$$

the continuity equation,

$$\nabla\cdot\boldsymbol{u} = 0 \quad \text{in } \Omega_s, \tag{2.7}$$

a constraint for the rigid component of motion,

$$\boldsymbol{D}(\boldsymbol{u}-\boldsymbol{u}_f) = \frac{1}{2}[\nabla(\boldsymbol{u}-\boldsymbol{u}_f) + \nabla(\boldsymbol{u}-\boldsymbol{u}_f)^T] = \boldsymbol{0} \quad \text{in } \Omega_s, \tag{2.8}$$

the interface conditions on the fluid–solid boundary,

$$\boldsymbol{u} = \boldsymbol{u}_i \quad \text{and} \quad \boldsymbol{\sigma}_s\cdot\hat{\boldsymbol{n}} = \boldsymbol{t}_{\partial\Omega_s} \quad \text{on } \partial\Omega_s, \tag{2.9}$$

and the initial condition,

$$\boldsymbol{u}(\boldsymbol{x}, t=0) = \boldsymbol{u}_o(\boldsymbol{x}) \quad \text{in } \Omega_s(t=0), \tag{2.10}$$

where ρ_s is the solid density, $\boldsymbol{\sigma}_s$ is the stress tensor, and $\boldsymbol{u}_f$ is the given deformation velocity field of the body in its own frame of reference. We have not yet committed to any specific form for $\boldsymbol{\sigma}_s$ within the solid. We will subsequently see that it will be composed of the pressure due to the continuity constraint and a Lagrange multiplier stress field due to the constraint (2.8) on the rigid component of motion.

Constraint (2.8) arises as follows. We decompose the solid velocity as $\boldsymbol{u} = \boldsymbol{u}_r + \boldsymbol{u}_f$ where $\boldsymbol{u}_r$ is the rigid motion of the body, and $\boldsymbol{u}_f$ is the deforming motion of the body in its own reference frame. $\boldsymbol{u}_f$ can represent undulations of a membrane (e.g. pectoral fins), or the organism's body (e.g. eel). It can also be a combination of motion of the body and the appendages. An example of this is found in sharks, where the piston-like motion of the muscles running along the spinal cord produces body waves, and the fins provide additional maneuverability. The specified deformation kinematics $\boldsymbol{u}_f$ are the only motion input required by this formulation. The translational and rotational swimming velocities of the body, which form its rigid motion $\boldsymbol{u}_r$, are obtained as a solution. The constraint (2.8) follows from the requirement that the deformation rate tensor associated with the rigid motion component $\boldsymbol{u}_r$ is zero in Ω_s.

Before proceeding, some remarks are in order regarding $\boldsymbol{u}_f$. The organism's own frame of reference is arbitrary in the sense that it depends on the exact form of $\boldsymbol{u}_f$. For cyclic motion, this means that the reference frame is chosen so that the deformation kinematics closely approximate the animal's body deformation in a frame moving with the animal's average swimming velocity over one propulsive cycle. However, all the mathematical details generalize to non-cyclic motions, such as those used for rapid maneuvers.

To illustrate this clearly, we take the example of the gymnotiform mode of swimming. The gymnotiform mode refers to swimming by means of a long (approximately 3/4ths of the body length) baseline fin that is attached along the ventral midline of the body, while keeping the body straight [38]. An example is the cruising motion of the South American weakly electric black ghost knifefish Apteronotus albifrons (Fig. 1). A reasonable approximation for the cruising mode is that the body moves rigidly with all deformations restricted to the fin. In this case the organism's frame of reference is fixed to three fixed, non-collinear points on its body. The fin undulates in this frame. In general the organism's reference frame is different from the reference frame attached to its center of mass. The former changes in time consistent with the definition of $\boldsymbol{u}_f$.

We note that Eq. (2.8) can be cast into the form [32]:

$$\begin{aligned} \nabla\cdot\boldsymbol{D}(\boldsymbol{u}-\boldsymbol{u}_f) &= \boldsymbol{0} \quad \text{in } \Omega_s, \\ \boldsymbol{D}(\boldsymbol{u}-\boldsymbol{u}_f)\cdot\hat{\boldsymbol{n}} &= \boldsymbol{0} \quad \text{on } \partial\Omega_s. \end{aligned} \tag{2.11}$$

This constraint gives rise to a distributed Lagrange multiplier stress field in the momentum equation (2.6). This is similar to how there is mechanical pressure in the momentum equation of an incompressible fluid due to the divergence-free con-

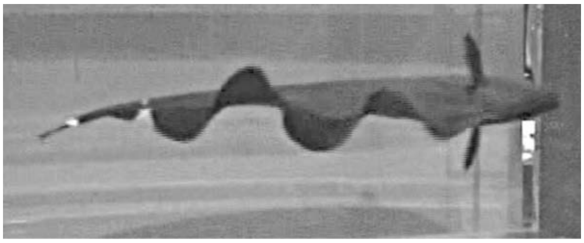


Fig. 1. Bottom view of a black ghost knifefish, *Apteronotus albifrons*, during steady swimming. Body deformations are small compared to fin deformations during steady swimming.

straint on velocity. Next, we present the formal variational formulation that leads to the distributed Lagrange multiplier corresponding to the motion constraints.

2.1. Variational formulation and distributed Lagrange multipliers

In the momentum transport equations for multi-component systems, all materials can be treated as a fluid with their respective material properties such as moduli of compressibility, viscosity, and thermal properties such as heat conductivity and speed of sound. This idea has been utilized to consolidate the compressible and incompressible equations for shallow geophysical flows into a single formulation [39]. However, in the limit of incompressibility, due to the infinite speed of sound in the material, the compressible equations become stiff. In such a case it is desirable to introduce mechanical pressure as a Lagrange multiplier corresponding to the incompressibility constraint. Much in the same spirit, perfectly rigid solids can be considered as a limit of fluids with infinite viscosity and compressibility, and a prior formulation [32,33,40] successfully addresses the resultant stiffness by introducing the rigidity constraint on the velocity inside the solid using a stress field. Since the viscosity is effectively infinite, the deformation rate tensor must be zero for the stress field to remain well behaved, as ensured by the Lagrange multiplier identified in the weak formulation of Patankar et al. [32].

To extend these ideas, we follow Patankar et al. [32], but now with specified body deformations. First we will review how mechanical pressure is introduced as a Lagrange multiplier corresponding to the divergence-free velocity constraint. Then, we will analogously identify the Lagrange multiplier term corresponding to the constraint in Eq. (2.8).

For the fluid domain, first we impose the divergence-free constraint on the solution and the variation spaces of velocity. These spaces also satisfy the boundary conditions on velocity. These are specified to be subsets of the Sobolev space $H^1(\Omega_f)^3$, so that both the functions and their first derivatives have finite L^2-norms:

$$V_{u,d}(t) = \{\boldsymbol{u} | \boldsymbol{u} \in H^1(\Omega_f)^3,\ \nabla\cdot\boldsymbol{u} = 0 \text{ in } \Omega_f,\ \boldsymbol{u} = \boldsymbol{u}_{\partial\Omega} \text{ on } \partial\Omega\}, \tag{2.12}$$

$$V_{o,d}(t) = \{\delta\boldsymbol{u} | \delta\boldsymbol{u} \in H^1(\Omega_f)^3,\ \nabla\cdot\delta\boldsymbol{u} = 0 \text{ in } \Omega_f,\ \delta\boldsymbol{u} = \boldsymbol{0} \text{ on } \partial\Omega\}, \tag{2.13}$$

where $\delta\boldsymbol{u}$ are the virtual variations of velocity and the subscript "*d*" denotes the divergence-free constraint. The weak form of Eqs. (2.1)–(2.5) in the fluid domain Ω_f can be written by using the principal of virtual work:

$$\int_{\Omega_f}\rho_f\left[\frac{\partial\boldsymbol{u}}{\partial t}+\boldsymbol{u}\cdot\nabla\boldsymbol{u}\right]\cdot\delta\boldsymbol{u}\,d\boldsymbol{x} - \int_{\Omega_f}\nabla\cdot(\boldsymbol{\sigma}\cdot\delta\boldsymbol{u})d\boldsymbol{x} + \int_{\Omega_f}\boldsymbol{\tau}:\boldsymbol{D}(\delta\boldsymbol{u})d\boldsymbol{x} = 0 \quad \forall\delta\boldsymbol{u}\in V_{o,d}. \tag{2.14}$$

Recall that $\boldsymbol{\sigma}$ and $\boldsymbol{\tau}$ are the total and deviatoric stress tensors in the fluid, related by $\boldsymbol{\sigma} = -p\boldsymbol{I}+\boldsymbol{\tau}$. The second term on the left hand side is the total work done on the fluid which is the same as the work done on the entire domain by the boundary traction, as becomes clear from the Gauss theorem:

$$\int_{\Omega_f}\nabla\cdot(\boldsymbol{\sigma}\cdot\delta\boldsymbol{u})d\boldsymbol{x} = \int_{\partial\Omega}\boldsymbol{t}_{\partial\Omega}\cdot\delta\boldsymbol{u}\,d\boldsymbol{x} - \int_{\partial\Omega_s}\boldsymbol{t}_{\partial\Omega_s}\cdot\delta\boldsymbol{u}\,d\boldsymbol{x}, \tag{2.15}$$

where $\boldsymbol{t}_{\partial\Omega_s}$ is the traction force vector acting *on the solid* from the fluid at the interface $\partial\Omega_s$, and $\boldsymbol{t}_{\partial\Omega}$ is the traction acting on the fluid at $\partial\Omega$. The first term on the right hand side of Eq. (2.15) is zero because $\delta\boldsymbol{u}$ is zero on $\partial\Omega$. The third term on the left hand side of Eq. (2.14) is the dissipation term. Now we relax the divergence-free constraint on the velocity spaces and impose it through a Lagrange multiplier p. The constraint term in the functional space (i.e. the contribution from the constraint to the Lagrangian of motion) is the integral of $p\nabla\cdot\boldsymbol{u}$ over the fluid domain. The weak form (2.14) is modified by adding the variation of this term as

$$\int_{\Omega_f}\rho_f\left[\frac{\partial\boldsymbol{u}}{\partial t}+\boldsymbol{u}\cdot\nabla\boldsymbol{u}\right]\cdot\delta\boldsymbol{u}\,d\boldsymbol{x} + \int_{\partial\Omega_s}\boldsymbol{t}_{\partial\Omega_s}\cdot\delta\boldsymbol{u}\,d\boldsymbol{x} + \int_{\Omega_f}\boldsymbol{\tau}:\boldsymbol{D}(\delta\boldsymbol{u})d\boldsymbol{x} - \int_{\Omega_f}p\nabla\cdot\delta\boldsymbol{u}\,d\boldsymbol{x} - \int_{\Omega_f}\delta p\nabla\cdot\boldsymbol{u}\,d\boldsymbol{x} = 0$$
$$\forall\delta\boldsymbol{u}\in V_o,\ \forall\delta p\in L^2(\Omega_f), \tag{2.16}$$

where δp is the variation of pressure. Here we have used Eq. (2.15), and the negative signs on the last two integrals are customary. The velocity space, the variation space for velocity, and the pressure space are now:

$$V_u(t) = \{\boldsymbol{u} | \boldsymbol{u} \in H^1(\Omega_f)^3,\ \boldsymbol{u} = \boldsymbol{u}_{\partial\Omega} \text{ on } \partial\Omega\}, \tag{2.17}$$

$$V_o(t) = \{\delta\boldsymbol{u} | \delta\boldsymbol{u} \in H^1(\Omega_f)^3,\ \delta\boldsymbol{u} = \boldsymbol{0} \text{ on } \partial\Omega\}, \tag{2.18}$$

$$L^2_o = \left\{ p | p \in L^2(\Omega_f),\ \int_{\Omega_f} p\, d\boldsymbol{x} = 0 \right\}, \tag{2.19}$$

where $L^2(\Omega_f)$ is the space of square integrable functions in Ω_f, and subscript "o" in the name L^2_o denotes the additional integral constraint on pressure. This constraint is included to remove the undetermined additive constant from the fluid pressure. Using the Gauss theorem, we get

$$\begin{aligned} &\int_{\Omega_f} \rho_f \left[\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u} \cdot \nabla \boldsymbol{u} \right] \cdot \delta\boldsymbol{u}\, d\boldsymbol{x} + \int_{\partial\Omega_s} \boldsymbol{t}_{\partial\Omega_s} \cdot \delta\boldsymbol{u}\, d\boldsymbol{x} - \int_{\Omega_f} \delta\boldsymbol{u} \cdot \nabla \cdot \tau\, d\boldsymbol{x} + \int_{\partial\Omega_s} \delta\boldsymbol{u} \cdot \tau \cdot \hat{\boldsymbol{n}}_{\partial\Omega_s}\, d\boldsymbol{x} + \int_{\Omega_f} \nabla p \cdot \delta\boldsymbol{u}\, d\boldsymbol{x} \\ &- \int_{\partial\Omega_s} p\delta\boldsymbol{u} \cdot \hat{\boldsymbol{n}}\, d\boldsymbol{x} - \int_{\Omega_f} \delta p(\nabla \cdot \boldsymbol{u}) d\boldsymbol{x} = 0 \quad \forall \delta\boldsymbol{u} \in V_o,\ \forall \delta p \in L^2(\Omega_f). \end{aligned} \tag{2.20}$$

Here we have used the fact that velocity variations are zero on $\partial\Omega$. Extracting terms with the variations $\delta\boldsymbol{u}$ over Ω_f produces the momentum equation for the fluid domain,

$$\rho_f \left[\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u} \cdot \nabla \boldsymbol{u} \right] = -\nabla p + \nabla \cdot \tau, \tag{2.21}$$

which is same as Eq. (2.1). Terms involving $\delta\boldsymbol{u}$ over $\partial\Omega_s$ produce the interface stress condition on the fluid–solid boundary (2.4), and the term involving δp produces the incompressibility constraint (2.2).

Now we consider the solid domain. The weak form for the solid domain is similar to Eq. (2.16), but with an additional rigidity constraint. First we impose this constraint on the solution space and the variation space for velocity:

$$V_{u,r}(t) = \{\boldsymbol{u} | \boldsymbol{u} \in H^1(\Omega_s)^3,\ \boldsymbol{D}(\boldsymbol{u} - \boldsymbol{u}_f) = \boldsymbol{0} \text{ in } \Omega_s\}, \tag{2.22}$$

$$V_{o,r}(t) = \{\delta\boldsymbol{u} | \delta\boldsymbol{u} \in H^1(\Omega_s)^3,\ \boldsymbol{D}(\delta\boldsymbol{u}) = \boldsymbol{0} \text{ in } \Omega_s\}, \tag{2.23}$$

where the subscript "r" indicates that the rigidity constraint is imposed on the space. With this, the virtual work statement, i.e. the weak form, for the solid domain is:

$$\int_{\Omega_s} \rho_s \left[\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u} \cdot \nabla \boldsymbol{u} \right] \cdot \delta\boldsymbol{u}\, d\boldsymbol{x} - \int_{\partial\Omega_s} \boldsymbol{t}_{\partial\Omega_s} \cdot \delta\boldsymbol{u}\, d\boldsymbol{x} - \int_{\Omega_s} p\nabla \cdot \delta\boldsymbol{u}\, d\boldsymbol{x} - \int_{\Omega_s} \delta p \nabla \cdot \boldsymbol{u}\, d\boldsymbol{x} = 0 \quad \forall \delta\boldsymbol{u} \in V_{o,r},\ \forall \delta p \in L^2(\Omega_s), \tag{2.24}$$

where the Lagrange multiplier $p \in L^2(\Omega_s)$ has been used to satisfy the continuity constraint. Next we follow Glowinski et al. [30] and Patankar et al. [32] to obtain a weak form that imposes the rigidity constraint by using a Lagrange multiplier field. To that end we first identify the term due to the rigidity constraint that will arise in the functional space, i.e. in the Lagrangian of motion. The constraint equation (2.11) is a vector constraint at each point in space. Hence we consider a vector field $\boldsymbol{\lambda}_r$ defined in Ω_s. We take a dot product of $\boldsymbol{\lambda}_r$ with the first of Eq. (2.11) and integrate over Ω_s. Using the Gauss theorem and the boundary condition in Eq. (2.11) gives

$$\int_{\Omega_s} \boldsymbol{D}(\boldsymbol{\lambda}_r) : \boldsymbol{D}(\boldsymbol{u} - \boldsymbol{u}_f) d\boldsymbol{x} = 0. \tag{2.25}$$

This is the term due to the rigidity constraint that will appear in the functional space. Now we relax the rigidity constraint on the velocity solution and variation spaces, and apply it in a weak form by adding a variation of Eq. (2.25) to Eq. (2.24). The variation of the Lagrangian form $\boldsymbol{D}(\boldsymbol{\lambda}_r) : \boldsymbol{D}(\boldsymbol{u} - \boldsymbol{u}_f)$ for the constraint is

$$\boldsymbol{D}(\boldsymbol{\lambda}_r) : \boldsymbol{D}(\delta\boldsymbol{u}) + \boldsymbol{D}(\delta\boldsymbol{\lambda}_r) : \boldsymbol{D}(\boldsymbol{u} - \boldsymbol{u}_f). \tag{2.26}$$

Adding the integral of these terms to Eq. (2.24), the weak form of the equations of motion for the solid domain becomes

$$\begin{aligned} &\int_{\Omega_s} \rho_s \left[\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u} \cdot \nabla \boldsymbol{u} \right] \cdot \delta\boldsymbol{u}\, d\boldsymbol{x} - \int_{\partial\Omega_s} \boldsymbol{t}_{\partial\Omega_s} \cdot \delta\boldsymbol{u}\, d\boldsymbol{x} + \int_{\Omega_s} \nabla p \cdot \delta\boldsymbol{u}\, d\boldsymbol{x} - \int_{\partial\Omega_s} p\delta\boldsymbol{u} \cdot \hat{\boldsymbol{n}}\, d\boldsymbol{x} - \int_{\Omega_s} \delta p(\nabla \cdot \boldsymbol{u}) d\boldsymbol{x} - \int_{\Omega_s} \nabla \cdot \boldsymbol{D}(\boldsymbol{\lambda}_r) \\ &\cdot \delta\boldsymbol{u}\, d\boldsymbol{x} - \int_{\Omega_s} \delta\boldsymbol{\lambda}_r \cdot \nabla \cdot \boldsymbol{D}(\boldsymbol{u} - \boldsymbol{u}_f) d\boldsymbol{x} + \int_{\delta\Omega_s} \hat{\boldsymbol{n}} \cdot \boldsymbol{D}(\boldsymbol{\lambda}_r) \cdot \delta\boldsymbol{u}\, d\boldsymbol{x} + \int_{\delta\Omega_s} \hat{\boldsymbol{n}} \cdot \boldsymbol{D}(\boldsymbol{u} - \boldsymbol{u}_f) \cdot \delta\boldsymbol{\lambda}_r\, d\boldsymbol{x} \\ &= 0 \quad \forall \delta\boldsymbol{u} \in H^1(\Omega_s)^3,\ \forall \delta p \in L^2(\Omega_s),\ \forall \delta\boldsymbol{\lambda}_r \in H^1(\Omega_s)^3. \end{aligned} \tag{2.27}$$

Note that $\boldsymbol{u} \in H^1(\Omega_s)^3$, and $\boldsymbol{\lambda}_r \in H^1(\Omega_s)^3$.

Equating the variations with respect to $\delta\boldsymbol{u}$ to zero, we obtain the momentum equation,

$$\rho_s \left[\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u} \cdot \nabla \boldsymbol{u} \right] = -\nabla p + \nabla \cdot \boldsymbol{D}(\boldsymbol{\lambda}_r), \tag{2.28}$$

and the condition at the fluid–solid boundary,

$$[-p\boldsymbol{I} + \boldsymbol{D}(\boldsymbol{\lambda}_r)] \cdot \hat{\boldsymbol{n}} = \boldsymbol{t}_{\Omega_s}. \tag{2.29}$$

Comparing Eqs. (2.28) and (2.29) with Eqs. (2.6) and (2.9), respectively, it is revealed that the stress within the solid is given by $\boldsymbol{\sigma}_s = -p\boldsymbol{I} + \boldsymbol{D}(\boldsymbol{\lambda}_r)$, where $\boldsymbol{\lambda}_r$ is the Lagrange multiplier due to the rigid motion constraint in the solid. Without loss of generality, we recast Eqs. (2.28) and (2.29) by using the constraint Eq. (2.11) to get

$$\rho_s\left[\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u}\cdot\nabla\boldsymbol{u}\right] = -\nabla p + \nabla\cdot[2\mu\boldsymbol{D}(\boldsymbol{u})] + \nabla\cdot\boldsymbol{D}(\boldsymbol{\lambda}), \tag{2.30}$$

$$[-p\boldsymbol{I} + 2\mu\boldsymbol{D}(\boldsymbol{u}) + \boldsymbol{D}(\boldsymbol{\lambda})]\cdot\hat{\boldsymbol{n}} = \boldsymbol{t}_{\Omega_s}, \tag{2.31}$$

where $\boldsymbol{\lambda} = \boldsymbol{\lambda}_r - 2\mu\boldsymbol{u}_f$ is the modified Lagrange multiplier. This form leads to a convenient combined formulation for the fluid–solid domain:

$$\rho\left(\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u}\cdot\nabla\boldsymbol{u}\right) = -\nabla p + \nabla\cdot[2\mu\boldsymbol{D}(\boldsymbol{u})] + \boldsymbol{f} \quad \text{in } \Omega, \tag{2.32}$$

$$\nabla\cdot\boldsymbol{u} = 0 \quad \text{in } \Omega, \tag{2.33}$$

$$\left.\begin{aligned} &\nabla\cdot\boldsymbol{D}(\boldsymbol{u}-\boldsymbol{u}_f) = \boldsymbol{0} \quad \text{in } \Omega_s,\\ &\boldsymbol{D}(\boldsymbol{u}-\boldsymbol{u}_f)\cdot\hat{\boldsymbol{n}} = \boldsymbol{0} \quad \text{on } \partial\Omega_s, \end{aligned}\right. \tag{2.34}$$

$$\boldsymbol{u} = \boldsymbol{u}_{\partial\Omega} \quad \text{on } \partial\Omega, \tag{2.35}$$

$$\boldsymbol{u}(\boldsymbol{x}, t=0) = \boldsymbol{u}_o(\boldsymbol{x}) \quad \text{in } \Omega(t=0). \tag{2.36}$$

$\rho = \rho_f + (\rho_s - \rho_f)H(\Omega_s)$ is the combined density field, where $H(\Omega_s)$ is the Heaviside step function which is 1 in Ω_s and 0 outside. $\boldsymbol{f} = \nabla\cdot\boldsymbol{D}(\boldsymbol{\lambda})$ is the constraint force due to the distributed Lagrange multiplier. From the combined formulation above, it is apparent that the governing equations for the fluid are applicable in the entire domain. To account for the presence of an immersed solid there is an additional forcing term $\boldsymbol{f}$ due to the rigidity constraint (2.34). $\boldsymbol{f}$ is non-zero only in Ω_s. Additionally, the difference in inertia is accounted for through the density field ρ. Note that the interface conditions at the fluid–solid boundary (Eqs. (2.4) and (2.9)) mutually cancel since the boundary is now internal to the combined fluid–solid domain. This formulation can be conveniently, although not necessarily, implemented by an immersed boundary type approach. This will be presented in Section 3.

2.2. Vorticity equation

Vortex methods have been considered by others to simulate self-propulsion [26,31]. Generation and diffusion of vorticity is one aspect of primary interest in these methods. Although we will not use a vortex approach, we show that our formulation can provide some insights into vorticity transport.

The vorticity transport equation in the fluid or solid domain can be obtained from the curl of Eq. (2.32),

$$\rho\left(\frac{\partial\boldsymbol{\omega}}{\partial t} + \boldsymbol{u}\cdot\nabla\boldsymbol{\omega}\right) = \rho\boldsymbol{\omega}\cdot\nabla\boldsymbol{u} + \mu\nabla^2(\boldsymbol{\omega}-\boldsymbol{\omega}_f) + \frac{1}{2}\nabla^2(\nabla\times\boldsymbol{\lambda}_r), \tag{2.37}$$

where $\boldsymbol{\omega} = \nabla\times\boldsymbol{u}$ is the vorticity, and $\boldsymbol{\omega}_f = \nabla\times\boldsymbol{u}_f$ is the vorticity due to the imposed deformation velocity field which is non-zero only in the solid body domain. Here we have used the identity $\nabla\times[\nabla\cdot\boldsymbol{D}(\boldsymbol{\lambda}_r)] = \frac{1}{2}\nabla^2(\nabla\times\boldsymbol{\lambda}_r)$. Eq. (2.37) shows that the rate of change of vorticity of a material point is caused by stretching/tilting of vortex tubes, diffusion of vorticity, and the rigidity constraint.

The diffusion term $\mu\nabla^2(\boldsymbol{\omega}-\boldsymbol{\omega}_f)$ becomes $\mu\nabla^2\boldsymbol{\omega}$ in the fluid domain because $\boldsymbol{\omega}_f$ is non-zero only in the solid domain. The diffusion term is zero in the solid as follows. The rigidity constraint is $\boldsymbol{D}(\boldsymbol{u}_r) = \boldsymbol{0}$, where $\boldsymbol{u}_r = \boldsymbol{u} - \boldsymbol{u}_f$ is the rigid component of the solid motion. If the deformation rate tensor associated with a velocity field is zero in a domain, then the vorticity field corresponding to that velocity field is a constant vector in that domain (see Sokolnikoff [37, p. 27 for proof]). Thus, $\nabla\boldsymbol{\omega}_r = \nabla(\boldsymbol{\omega}-\boldsymbol{\omega}_f) = \boldsymbol{0}$, indicating that the rigid body component of the vorticity inside the solid must not vary with spatial coordinates. This is consistent with the fact that the vorticity inside a region undergoing rigid body motion is twice its angular velocity. Hence there is no diffusion of vorticity in the interior of Ω_s, i.e. $\nabla^2(\boldsymbol{\omega}-\boldsymbol{\omega}_f) = \boldsymbol{0}$.

The third term on the right hand side of Eq. (2.37), $\frac{1}{2}\nabla^2(\nabla\times\boldsymbol{\lambda}_r)$, is non-zero only in the solid domain. Defining $\boldsymbol{\lambda}_\omega = \frac{1}{2}(\nabla\times\boldsymbol{\lambda}_r)$, it becomes $\nabla^2\boldsymbol{\lambda}_\omega$. In fact it can be formally shown, following a derivation similar to that in Section 2.1, that $\boldsymbol{\lambda}_\omega$ is a Lagrange multiplier corresponding to the constraint $\nabla\boldsymbol{\omega}_r = \boldsymbol{0}$ in Ω_s, which is a consequence of the rigidity constraint on $\boldsymbol{u}_r$. $\nabla^2\boldsymbol{\lambda}_\omega$ is the source term due to this constraint.

Inside the solid domain the total vorticity is $\boldsymbol{\omega} = \boldsymbol{\omega}_r + \boldsymbol{\omega}_f$, of which only the rigid component $\boldsymbol{\omega}_r$ is unknown. The angular velocity of the solid body is half of $\boldsymbol{\omega}_r$. Thus, Eq. (2.37) is effectively an equation for the angular velocity of the solid body. It implies that the angular velocity of the solid changes due to the vorticity injected into the domain by the deforming velocity field as well as the source term due to $\boldsymbol{\lambda}_\omega$. The term $\nabla^2\boldsymbol{\lambda}_\omega$ plays a role in the solid domain that is similar to that of the vorticity diffusion term $\mu\nabla^2\boldsymbol{\omega}$ in the fluid domain. It distributes the vorticity in the solid domain.

A key element which shows that our Lagrange multiplier formulation connects to vortex methods in a natural way is the vorticity jump at the solid–liquid interface. Although the velocity is continuous at the interface, its gradient – and hence the vorticity – is discontinuous at the interface. Continuity of velocity at the interface along with the definition of vorticity gives rise to the continuity of the normal component of vorticity [41]:

$$[|\boldsymbol{\omega}|]\cdot\hat{\boldsymbol{n}} = 0, \tag{2.38}$$

and the jump condition on the tangential component of vorticity:

$$\hat{\boldsymbol{n}}\times[|\boldsymbol{\omega}|] = -2[|\boldsymbol{D}(\boldsymbol{u})|]\cdot\hat{\boldsymbol{n}}. \tag{2.39}$$

where $[|\cdot|]$ denotes a jump in a variable across the interface, i.e. the value on the fluid side minus the value on the solid side. Physically, Eq. (2.39) means that the jump in the tangential vorticity is proportional to the jump in the *normal* component of the strain rate of a fluid element on the interface. It is in agreement with a separate derivation presented by Dopazo et al. [41].

Next we consider the vorticity transport at the fluid–solid interface following a procedure similar to that by Dopazo et al. [41]. To do so, we divide the momentum equation (2.32) by the density in the respective domains, take a curl of the resulting equations, and integrate it over a pillbox V that includes the fluid–solid interface. This gives

$$\int_V \frac{D\boldsymbol{\omega}}{Dt}\,d\boldsymbol{x} = \int_V \boldsymbol{\omega}\cdot\nabla\boldsymbol{u}\,d\boldsymbol{x} - \int_V \nabla\times\left(\frac{\nabla p}{\rho}\right)d\boldsymbol{x} + \int_V \nabla\times\left(\frac{\nabla\cdot\boldsymbol{\sigma}}{\rho}\right)d\boldsymbol{x}, \tag{2.40}$$

where $\boldsymbol{\sigma} = 2\mu\boldsymbol{D}(\boldsymbol{u}) + \boldsymbol{D}(\lambda) = 2\mu\boldsymbol{D}(\boldsymbol{u}-\boldsymbol{u}_f) + \boldsymbol{D}(\lambda_r)$. Inside the solid, since $\boldsymbol{D}(\boldsymbol{u}-\boldsymbol{u}_f) = \boldsymbol{0}$, it becomes $\boldsymbol{D}(\lambda_r)$. Inside the fluid, $\boldsymbol{\lambda} = \boldsymbol{0}$, hence $\boldsymbol{\sigma} = 2\mu\boldsymbol{D}(\boldsymbol{u})$. In the limit of zero thickness of the pillbox, the left hand side becomes zero by Reynolds transport theorem and the fact that, when the boundary layer is fully resolved, vorticity does not have any singularity, but has only a discontinuity. The generalized Gauss theorem is then applicable in this pillbox (see Xu and Wang [42, p. 1952, Theorem 3.3]). The first integral on the right hand side approaches zero owing to the fact that both vorticity and velocity gradient have discontinuities but no singularities. Using the generalized Gauss theorem in the limit of the infinitesimally thin pillbox, we obtain

$$-\hat{\boldsymbol{n}}\times\left[\left|\frac{\nabla p}{\rho}\right|\right] + \hat{\boldsymbol{n}}\times\left[\left|\frac{\nabla\cdot\boldsymbol{\sigma}}{\rho}\right|\right] = 0. \tag{2.41}$$

On the fluid side,

$$\hat{\boldsymbol{n}}\times\frac{\nabla\cdot\boldsymbol{\sigma}}{\rho} = \hat{\boldsymbol{n}}\times\frac{\nabla\cdot 2\mu\boldsymbol{D}(\boldsymbol{u})}{\rho_f}. \tag{2.42}$$

Here we use subscripts “*f*” and “*s*” to denote quantities in the fluid and the solid. Upon rearranging using Einstein’s index notation, and using the fact that the flow is incompressible, it can be shown that

$$\hat{\boldsymbol{n}}\times\frac{\nabla\cdot\boldsymbol{\sigma}}{\rho} = \frac{\mu}{\rho_f}[(\hat{\boldsymbol{n}}\cdot\nabla)\boldsymbol{\omega} - \nabla\boldsymbol{\omega}\cdot\hat{\boldsymbol{n}}]. \tag{2.43}$$

Similarly, on the solid side,

$$\hat{\boldsymbol{n}}\times\frac{\nabla\cdot\boldsymbol{\sigma}}{\rho} = \hat{\boldsymbol{n}}\times\frac{\nabla\cdot\boldsymbol{D}(\boldsymbol{\lambda}_r)}{\rho_s} = \frac{1}{\rho_s}[(\hat{\boldsymbol{n}}\cdot\nabla)\boldsymbol{\lambda}_\omega - \nabla\boldsymbol{\lambda}_\omega\cdot\hat{\boldsymbol{n}}] + \hat{\boldsymbol{n}}\times\frac{\nabla(\nabla\cdot\boldsymbol{\lambda}_r)}{\rho_s}. \tag{2.44}$$

where we have used the relation $\boldsymbol{\lambda}_\omega = \frac{1}{2}(\nabla\times\boldsymbol{\lambda}_r)$. Using Eqs. (2.43) and (2.44) in Eq. (2.41), the interface condition can be written as:

$$q^f_\omega + q^s_\omega = \hat{\boldsymbol{n}}\times\left[\left|\frac{-\nabla p}{\rho}\right|\right] - \hat{\boldsymbol{n}}\times\frac{\nabla(\nabla\cdot\boldsymbol{\lambda}_r)}{\rho_s} - \frac{\mu}{\rho_f}\nabla\boldsymbol{\omega}\cdot\hat{\boldsymbol{n}} + \frac{1}{\rho_s}\nabla\boldsymbol{\lambda}_\omega\cdot\hat{\boldsymbol{n}}, \tag{2.45}$$

where

$$q^f_\omega = -\frac{\mu}{\rho_f}\hat{\boldsymbol{n}}\cdot\nabla\boldsymbol{\omega}, \tag{2.46}$$

and

$$q^s_\omega = \frac{1}{\rho_s}\hat{\boldsymbol{n}}\cdot\nabla\boldsymbol{\lambda}_\omega. \tag{2.47}$$

In Eqs. (2.45)–(2.47), all quantities are computed at the fluid–solid interface; those involving fluid variables are computed in a limit from the fluid side and those involving solid variables are computed in a limit from the solid side. q^f_ω and q^s_ω are the fluxes of vorticity from the fluid–solid interface into the fluid and solid domains, respectively. Thus, $q^f_\omega + q^s_\omega$ is the total source of vorticity at the fluid–solid interface. The right hand side of Eq. (2.45) shows that vorticity is generated at the interface due to a pressure jump, the rigidity constraint (with the specified deformation kinematics), and the stretching and tilting of the vortex tubes at the interface.

The vorticity generation equation (2.45) could be specialized to high Reynolds number flows in which vorticity is confined to vortex sheets at the interface and is shed at the trailing edge. This could lead to equations for interfacial vortex sheet dynamics similar to those derived for liquid–gas interfaces by Dopazo et al. [41]. Furthermore, in Section 3.5 we show that the discrete version of our correction step is equivalent to the vortex methods via the generation of a vortex sheet of appropriate strength at the solid–fluid interface.

3. The self-propulsion algorithm

In this section, we present an algorithm to solve Eqs. (2.32)–(2.36) that were formulated for the problem of self-propulsion. The new algorithm is called the Fully Resolved Momentum Redistribution for self-Propulsion (FuRMoRP) algorithm. Recall that the term "fully resolved simulation" (FRS) means that the fluid–solid coupling is not computed using any reduced order models (e.g. drag laws). Instead, the flow around the swimming body is fully resolved, where the fluid could be governed by any constitutive behavior. Here, we have considered only constant viscosity Newtonian fluid. However, the formulation is generalizable to other constitutive forms including turbulence models such as large eddy simulation (LES) or Reynolds averaged Navier–Stokes (RANS) simulations. E.g. if all flow scales including the smallest turbulent scales are resolved, then our method would be a Fully Resolved–Direct Numerical Simulation (FR-DNS) scheme, while an LES based approach would be an FR-LES scheme.

The FuRMoRP algorithm can be implemented by a body-fitted mesh approach as well by an immersed boundary approach. We have implemented the algorithm by an immersed approach in the interest of efficiency when simulating a variety of complex morphologies. In this approach the fluid–solid interface condition is handled implicitly, allowing the use of regular grids even in the case of irregular body configurations. This eliminates the need for grid regeneration. We use a mixed Eulerian–Lagrangian formulation wherein the solid is tracked by a collection of Lagrangian particles distributed within the volume of the body, and the fluid is discretized on an Eulerian background grid. Hence, the location of the fluid–solid interface is captured to within the stencil of the discrete Eulerian–Lagrangian interpolation function (discussed later).

In the algorithm presented below, and in the numerical examples to be presented later, we will assume that the immersed body is neutrally buoyant, i.e. $\rho_f = \rho_s$. This is often reasonable for swimming animals. Note that our formulation, presented in the previous section, is general and can be implemented for non-neutrally buoyant bodies as well.

The mathematical formulation presented above implies that there are three central concepts in the FuRMoRP algorithm:

(1) Any motion of a swimming solid body can be described as the sum of a rigid body motion and a deformation velocity.
(2) The rigidity constraint on the rigid component of the solid velocity field gives rise to a force field just like incompressibility gives rise to a force due to pressure.
(3) The core mechanism of self-propulsion is the redistribution of momentum in the entire domain in such a way as to generate the appropriate swimming velocity of the body. The concept of momentum redistribution will become clear when the method is presented below.

The second statement above has been extensively used in studies of rigid particles in fluids [43,32,33]. The momentum redistribution concept is an extension of that approach to self-propelling flexible bodies. Furthermore, the current method separates the elastic response of the solid from the fluid flow solution. Hence, it leverages the availability of prescribed motion as 3D kinematic data acquisition techniques become more common for biological applications [19,20,44]. This can provide an alternative to relying on force measurements, as velocity measurements are more amenable to non-intrusive recording techniques.

The algorithm has three steps:

Step One: We regard the entire domain as a fluid and solve for an intermediate velocity field $\hat{\boldsymbol{u}}$ using the incompressible Navier–Stokes equations, with the assumption of constant viscosity:

$$\rho\frac{\hat{\boldsymbol{u}} - \boldsymbol{u}^{n-1}}{\Delta t} + \rho(\boldsymbol{u}^{n-1}\cdot\nabla)\boldsymbol{u}^{n-1} = -\nabla\tilde{p} + \mu\nabla^2\boldsymbol{u}^{n-1}, \tag{3.1}$$

where superscript $n-1$ denotes the solution at the end of the previous time step, and superscript n is the solution at the end of the new time step. It should be noted that Δt may be regarded as a sub-step of a higher order time marching scheme, e.g. a fourth order Runge–Kutta scheme. In that case, "$n-1$" and "n" would denote those substeps. $\tilde{p}$ is an intermediate pressure (which will later be corrected when we impose the rigidity constraint), μ is the fluid viscosity, and ρ is the density of the fluid and the organism. The incompressibility constraint $\nabla\cdot\hat{\boldsymbol{u}} = 0$ gives rise to the Poisson equation for pressure,

$$\nabla^2\tilde{p} = -\nabla\cdot[\rho(\boldsymbol{u}^{n-1}\cdot\nabla)\boldsymbol{u}^{n-1}]. \tag{3.2}$$

This work assumes the prescribed deformation to be divergence-free. However, some animals may be changing their body volume dynamically during propulsion [45], and the current formulation can be adapted to that situation by appropriate addition of mass sources in the Poisson equation. We do not consider this in the present work.

Step Two: A second projection step is required. For simplicity of exposition we will assume that there is only one swimming organism in the domain; however, the approach is trivially generalizable to multiple swimming bodies. $\hat{\boldsymbol{u}}$ is not the final velocity field inside the body of the organism because the deforming motion $\boldsymbol{u}_f$ is not yet imposed in the solid body, and the rigid component $\boldsymbol{u}_r$ of the body velocity is yet to be determined. To take into account both of these effects, the projection step corrects the velocity in the body. This correction is equivalent to adding a force $\boldsymbol{f}$ in the semidiscrete Navier–Stokes equations, to obtain a corrected velocity $\tilde{\boldsymbol{u}}$:

$$\rho\frac{\tilde{\boldsymbol{u}}-\hat{\boldsymbol{u}}}{\Delta t}=\boldsymbol{f}, \tag{3.3}$$

where $\boldsymbol{f}$ is an equivalent of the constraint force, $\nabla\cdot\boldsymbol{D}(\boldsymbol{\lambda})$, appearing in Eq. (2.32). An equation for $\boldsymbol{f}$ follows from the constraint in Eq. (2.34). Noting that $\tilde{\boldsymbol{u}}$ must satisfy Eq. (2.34), we get

$$\begin{aligned}&\nabla\cdot\boldsymbol{D}(\tilde{\boldsymbol{u}}-\boldsymbol{u}_f)=\nabla\cdot\boldsymbol{D}(\hat{\boldsymbol{u}}-\boldsymbol{u}_f+\frac{\boldsymbol{f}\Delta t}{\rho})=\boldsymbol{0}\quad\text{in }\Omega_s,\\&\boldsymbol{D}(\tilde{\boldsymbol{u}}-\boldsymbol{u}_f)\cdot\hat{\boldsymbol{n}}=\boldsymbol{D}(\hat{\boldsymbol{u}}-\boldsymbol{u}_f+\frac{\boldsymbol{f}\Delta t}{\rho})\cdot\hat{\boldsymbol{n}}=\boldsymbol{0}\quad\text{on }\partial\Omega_s.\end{aligned}\tag{3.4}$$

This implies that $\hat{\boldsymbol{u}}-\boldsymbol{u}_f+\frac{\boldsymbol{f}\Delta t}{\rho}$ is a rigid motion, which should in fact be equal to the rigid component of the swimmer's velocity. Consequently, $\boldsymbol{f}$ is given by

$$\boldsymbol{f}=\rho\frac{\boldsymbol{u}_b-\hat{\boldsymbol{u}}}{\Delta t}. \tag{3.5}$$

In the above equation, $\boldsymbol{u}_b=\boldsymbol{u}_r+\boldsymbol{u}_f$ is defined only within the solid body, where

$$\boldsymbol{u}_r=\boldsymbol{U}+\omega\times\boldsymbol{r}, \tag{3.6}$$

is yet to be determined, and $\boldsymbol{r}$ is a position vector with respect to the centroid of the solid body.

The rigid velocity $\boldsymbol{u}_r$ is determined as follows. Eqs. (3.3) and (3.5) imply that, in the solid domain, $\hat{\boldsymbol{u}}$ is corrected to $\tilde{\boldsymbol{u}}(=\boldsymbol{u}_b)$. Given that we are modeling a self-propelled body, the total translational and angular momenta in the entire fluid–solid system should be conserved during the projection step. Otherwise, it would lead to an unphysical external force on the system. To conserve the linear and angular momenta during the projection step, we need to ensure that the momenta in the $\hat{\boldsymbol{u}}$ and $\tilde{\boldsymbol{u}}$ fields are the same. Additionally, we note that the projection step (Eqs. (3.3) and (3.5)) is effectively imposed only in the solid domain since $\boldsymbol{f}$ is non-zero only in that domain. The momentum conservation condition thus leads to equations for the unknown translational and angular velocities $\boldsymbol{U}$ and ω:

$$M\boldsymbol{U}=\int_{\Omega_s}\rho_s(\hat{\boldsymbol{u}}-\boldsymbol{u}_f)d\boldsymbol{x}, \tag{3.7}$$

$$\boldsymbol{I}\omega=\int_{\Omega_s}\boldsymbol{r}\times\rho_s(\hat{\boldsymbol{u}}-\boldsymbol{u}_f)d\boldsymbol{x}, \tag{3.8}$$

where M is the mass of the body and $\boldsymbol{I}$ is the instantaneous moment of inertia tensor of the body. The projection step requires no iterations and is computationally inexpensive. It is identical to the approach used by Patankar [33] for freely moving rigid bodies, but extended in this case to flexible bodies with specified deformations.

Step Three: Although the first two steps maintain the divergence-free nature of the velocity field in the interior of the fluid and the solid domains, Step Two above introduces a discontinuity in velocity at the interface, giving rise to a non-zero divergence at the interface (explained later in Section 3.5). Hence in Step Three, $\tilde{\boldsymbol{u}}$ is projected onto a divergence-free velocity space by using a scalar field ϕ to remove this non-zero divergence as:

$$\boldsymbol{u}^n=\tilde{\boldsymbol{u}}-\nabla\phi, \tag{3.9}$$

where ϕ is obtained by solving the Poisson equation

$$\nabla^2\phi=\nabla\cdot\tilde{\boldsymbol{u}}, \tag{3.10}$$

This step is equivalent to adding a pressure force $-\nabla(\frac{\rho\phi}{\Delta t})$ in the Navier–Stokes equations:

$$\rho\frac{\boldsymbol{u}^n-\tilde{\boldsymbol{u}}}{\Delta t}=-\rho\frac{\nabla\phi}{\Delta t}, \tag{3.11}$$

Adding Eqs. (3.1), (3.3), and (3.11), we see that the three steps combined together are equivalent to

$$\rho\frac{\boldsymbol{u}^n-\boldsymbol{u}^{n-1}}{\Delta t}+\rho(\boldsymbol{u}^{n-1}\cdot\nabla)\boldsymbol{u}^{n-1}=-\nabla p^{n-1}+\mu\nabla^2\boldsymbol{u}^{n-1}+\boldsymbol{f}, \tag{3.12}$$

where the corrected pressure is $p^{n-1}=\tilde{p}+\frac{\rho\phi}{\Delta t}$.

To summarize, the FuRMoRP algorithm has the following *three steps*:

(1) Solve the Navier–Stokes equation (3.1) assuming the entire solid and fluid domain is a fluid.
(2) Obtain the rigid motion using Eqs. (3.7) and (3.8). Correct the velocity in the solid body using Eq. (3.3).
(3) Apply the divergence-free condition on the corrected velocity according to Eq. (3.9).

In the algorithm above, the solid body can be fully deforming, partly rigid and partly deforming, or fully rigid. The nature of the body is defined by the deformation velocity field $\boldsymbol{u}_f$. We will consider various body types in our test cases. Additionally, the solid body can be three-dimensional, or it can be an immersed surface or a curve, or any combination. The fact that the algorithm above is valid for immersed surfaces and curves can be rigorously derived. We will show test cases with immersed surfaces that are rigid or flexible, which show that our algorithm accurately predicts their dynamics.

In the above presentation of the FuRMoRP algorithm we have treated the convection and viscous terms explicitly. An implicit approach can be easily devised.

If the three steps in the FuRMoRP algorithm are solved as given above at each time step, it will lead to a scheme that is first order with respect to time. Higher order temporal schemes can be easily constructed by executing the three steps of the FuRMoRP algorithm within the framework of, e.g. a Runge–Kutta-type scheme. In this case, as mentioned before, the three steps would be carried out at each sub-step of the time marching scheme. The spatial order of accuracy will depend on the discretization of the spatial derivatives, and the interpolation scheme used in the Eulerian–Lagrangian coupling between the fluid mesh and the marker particles representing the solid body. The implementation details, including the current choice of spatial and temporal discretization and the interpolation scheme, will be discussed in this section. We will also discuss the algorithmic efficiency and the physical interpretation of the projection step.

3.1. Spatial discretization and time marching

We implemented FuRMoRP using an immersed boundary method for solid bodies. Step One of the algorithm is solved on a uniform structured mesh in the entire domain. Note that the momentum redistribution algorithm is conceptually separate from the flow solver, and hence can be combined equally well with a flow solver that uses a non-uniform mesh. The spatial derivative operators are discretized using 6th order compact finite difference schemes with high spectral resolution [46] and optimized coefficients [47]. The near-spectral resolution of these schemes is useful in resolving the sharp gradients near the solid surface with fewer grid points. An explicit time stepping scheme is used for time advancement. We use a low storage, low dissipation and dispersion Runge–Kutta scheme of fourth order (LDDRK4) [48], which has eleven stages in which the solution proceeds by $2\Delta t$. The three step FuRMoRP algorithm is solved at each of these stages. Since the mesh is structured and uniform, we solve the pressure Poisson equation (Eq. (3.2)) by using Fast Fourier Transforms (FFT). Similarly, Eq. (3.10) in Step Three of the algorithm is also solved using an FFT solver. Implementation of Step Two is done by using Lagrangian particles, and is explained in Section 3.2.

Our current implementation is parallelized and uses the Message Passing Interface (MPI) library for interprocessor communication. The fluid flow solver (i.e. solutions without an immersed solid body) was validated using test cases that allow comparison with analytical solution or experimental data. For more details the reader is referred to Shirgaonkar and Lele [49]. Validation of the FuRMoRP based code will be presented in Section 4.

3.2. Eulerian–Lagrangian coupling and interpolation

We use a mixed Eulerian–Lagrangian approach. In this approach there is a structured mesh for the entire domain, and the solid body is represented by a collection of Lagrangian marker particles or points.

Step Two of the algorithm, where the velocity in the solid domain is corrected to account for the swimming motion, requires Eulerian–Lagrangian interpolation. The deforming velocity field $\boldsymbol{u}_f$ is known at the location of the Lagrangian marker particles within the solid body. Additionally, we define a scalar function ξ which is equal to one at the Lagrangian marker particles. To prepare for the computation of Eqs. (3.7) and (3.8) we project $\boldsymbol{u}_f$ and ξ from the Lagrangian particles to the background Eulerian grid. This is done using top hat interpolation as follows. For each Lagrangian marker particle, we determine which eight Eulerian grid nodes surround it. The value of the variable (i.e. $\boldsymbol{u}_f$ or ξ) at the marker particle is then assigned to those Eulerian nodes. For nodes that have contributions from multiple marker particles, the arithmetic average of the variable values from all contributing marker particles is assigned to that node. For Eulerian nodes in the fluid domain, no particle contribution results, hence the value at these nodes is set to zero.

This projection results in $\boldsymbol{u}_f$ and ξ being defined on the Eulerian grid. It is non-zero only in the region that overlaps with the solid body and zero everywhere else. Specifically, the scalar field $\xi(x,y,z,t)$ is equal to one on Eulerian nodes that fall in the solid domain and zero outside. Thus, $\xi(x,y,z,t)$ identifies the location of the solid body on the Eulerian grid. Since the solid body shape does not conform to the background mesh, there is some distortion of the body shape on the order of the grid size.

The integration operations in Eqs. (3.7) and (3.8) are now performed as:

$$\int_{\Omega_s} \eta(x,y,z,t^*)dx\,dy\,dz = \int_{\Omega} \xi(x,y,z,t^*)\eta(x,y,z,t^*)dx\,dy\,dz \tag{3.13}$$

for any generic variable η at any given time t^*. The integrals are numerically evaluated as summation over Eulerian grid points.

Once Eqs. (3.7) and (3.8) are solved, we correct the velocity at all Eulerian nodes that fall within the solid domain, i.e. at all Eulerian nodes where $\xi = 1$. To do so we replace $\hat{\boldsymbol{u}}$ by $\boldsymbol{u}_b = \boldsymbol{U} + \boldsymbol{\omega} \times \boldsymbol{r} + \boldsymbol{u}_f$. This is equivalent to implementing Eqs. (3.3) and (3.5).

Some remarks regarding this interpolation scheme should be made. Instead of the top hat interpolation, higher order interpolation methods can be used to reduce the error. Lower order interpolation functions such as the delta function

[50] may introduce numerical diffusion at the interface, and higher order polynomial interpolations are known to cause spurious oscillations near the sharp gradients at the boundary. These issues can impact the accuracy of immersed boundary turbulent flow simulations. Better higher order interpolation schemes are available [51,52], or the immersed interface method [53,54] may provide better accuracy at the interface. The effect of a specific choice of interpolation scheme on numerical accuracy has been addressed by other authors [55,56].

For thin membranes, the higher order interpolation functions (e.g. delta function) require costly matrix inversions in order to achieve the specified velocity $\boldsymbol{u}_f$ at the solid location. For the grid points interior to 3D solid objects moving at a prescribed velocity, this issue is relatively benign and matrix inversion can be avoided. Various types of interpolations have been used in such problems [55,57,56]. Also, for membranes where the elastic force is prescribed, delta function interpolation without matrix inversion can be used to spread forces to neighboring grid points. For thin membranes with specified velocity (such as some of the examples illustrated later in this paper), top hat interpolation achieves the desired prescribed velocity at the particle locations in the Eulerian velocity field, without requiring expensive computations or tedious book keeping.

Sensitivity tests for the number of Lagrangian particles per grid cell (N_p) representing the solid body showed that although the velocity field away from the body is relatively insensitive to $N_p,\ N_p = 8$ is needed to obtain accurate fluid–solid surface forces. This is consistent with the observation in prior literature that two particles are needed in each direction per grid cell to prevent "leakage" of fluid through the membrane [50].

3.3. Force computation

The force on the solid body is obtained by using the integration operation (Eq. (3.13)). In particular, the viscous and pressure forces on the solid can be computed by taking $\eta = \mu\nabla^2\boldsymbol{u}^{n-1}$, and $\eta = -\nabla p^{n-1}$, respectively. The computed forces can have some high frequency noise, which is inherent to the immersed boundary method [58] due to movement of the solid–fluid boundary in a fixed background Eulerian mesh. To remove the noise and obtain physical values of the forces up to the grid-timescale, we filter the modes in the forces which vary faster than the grid-timescale. The grid-timescale here is defined as the time required by the point with the maximum velocity on the fin to travel one grid cell. Since the spatial discretization can only capture length scales up to the grid cell size, the above smoothing retains the forces down to the temporal scale consistent with the smallest resolved spatial scale.

3.4. Computational efficiency considerations

One of the strengths of the momentum redistribution algorithm is its efficiency during the correction step. The correction step, which incorporates the solid body into the flow solver, consumes a very small fraction of the total computational time taken by the flow solver. Finite difference temporal and spatial schemes have linear complexity. For incompressible flows the pressure solution is typically the most CPU-intensive task, which gives rise to a slightly superlinear complexity, depending on the pressure solution method used.

The solid projection scheme given by Eqs. (3.7) and (3.8) consists of spatial integration which is a linear-time operation. The pressure component of the projection force, given by Eqs. (3.9) and (3.10), can be included in the pressure solution of the flow solver thereby avoiding extra cost. In the examples presented here, the convective CFL number was more restrictive than the viscous CFL number, and was in the range 0.2–0.5. Table 1 lists the fraction of the total time required for the projection scheme with the *current implementation* for the flexible membrane problem (Section 4.2), the horizontally moving vertical plate in finite domain (Fig. 4), and the black ghost knifefish (Section 4.5) to span 2D and 3D simulations. The vertical plate case tests the extremely load-imbalanced spatial distribution of the Lagrangian particles representing the plate, where all particles are contained in a single processor (a total of eight processors were used). Table 1 shows that for the more typical cases, the solid body projection part takes less than 5% of the total CPU time. This is significantly smaller than the 86% fraction reported for the same problem size in the case of the swimming sheet [59]. Even with the highly imbalanced particle load among processors, the cost is reasonable.

3.5. Physical interpretation of the projection step

3.5.1. Strain rate redistribution within the solid body

Recall that in the projection step, the intermediate velocity field $\hat{\boldsymbol{u}}$ is corrected to the solid body velocity $\boldsymbol{u}_b = \boldsymbol{u}_r + \boldsymbol{u}_f$. Hence the projection step can be represented by the application of a force field, $\boldsymbol{f}$, in the semidiscrete Navier–Stokes equations (Eq. (3.3)). The change in the velocity gradient tensor during this step is

Table 1
CPU time required by the solid body correction part of the algorithm.

Case	Comments	CPU time for correction module (%)
Swimming sheet [Section 4.2]	2D	1.3
Vertical plate of Fig. 4	2D, extreme load imbalance of particles among processors	11.3
Black ghost knifefish [Section 4.5]	3D	3.5

$$\boldsymbol{G}(\boldsymbol{u}_b - \hat{\boldsymbol{u}}) = \boldsymbol{G}(\boldsymbol{u}_b) - \boldsymbol{G}(\hat{\boldsymbol{u}}) = \boldsymbol{G}(\boldsymbol{f})\frac{\Delta t}{\rho}, \tag{3.14}$$

where $\boldsymbol{G}(\boldsymbol{\Phi}) = \partial\Phi_i/\partial x_j$ is the gradient of any vector field $\boldsymbol{\Phi}$. Considering first the symmetric part of this equation,

$$D_{ij}(\boldsymbol{u}_b - \hat{\boldsymbol{u}}) = \frac{1}{2}(u_{bi,j} + u_{bj,i}) - \frac{1}{2}(\hat{u}_{i,j} + \hat{u}_{j,i}) = \frac{1}{2}(f_{i,j} + f_{j,i})\frac{\Delta t}{\rho}. \tag{3.15}$$

The trace of this equation is

$$\nabla \cdot \boldsymbol{u}_b = (\nabla \cdot \boldsymbol{f})\frac{\Delta t}{\rho}, \tag{3.16}$$

since $\hat{\boldsymbol{u}}$ is divergence-free. This term, $\nabla \cdot \boldsymbol{u}_b$ is zero in the interior of the solid, because $\boldsymbol{u}_b = \boldsymbol{u}_r + \boldsymbol{u}_f$, and both $\boldsymbol{u}_r$ and $\boldsymbol{u}_f$ are divergence-free. The projection step (Step Two) creates a non-zero divergence at the interface. Depending on the interpolation scheme, this divergence is spread over a few grid cells near the interface. This divergence is removed using the correction $\nabla\phi$ (Eq. (3.9)), giving an additional mechanical pressure force. The deviatoric part of Eq. (3.15) gives

$$D_{dij}(\boldsymbol{u}_b - \hat{\boldsymbol{u}}) = \left[\frac{1}{2}(u_{bi,j} + u_{bj,i}) - \nabla \cdot \boldsymbol{u}_b\right] - \left[\frac{1}{2}(\hat{u}_{i,j} + \hat{u}_{j,i}) - \nabla \cdot \hat{\boldsymbol{u}}\right] = \left[\frac{1}{2}(f_{i,j} + f_{j,i})\frac{\Delta t}{\rho} - \nabla \cdot \boldsymbol{f}\frac{\Delta t}{\rho}\right], \tag{3.17}$$

where the subscript "d" indicates deviatoric part. But $\boldsymbol{u}_b = \boldsymbol{u}_r + \boldsymbol{u}_f$, and $\boldsymbol{u}_r$ is a rigid body motion, giving us $\frac{1}{2}(u_{bi,j} + u_{bj,i}) - \nabla \cdot \boldsymbol{u}_b = \frac{1}{2}(u_{fi,j} + u_{fj,i}) - \nabla \cdot \boldsymbol{u}_f$. Hence the force construction $\boldsymbol{f}$ is such that it drives the strain rate of the velocity field $\hat{\boldsymbol{u}}$ to that of $\boldsymbol{u}_f$. Patankar [33] presented this efficient force construction for rigid bodies, which we have extended here for flexible bodies with specified kinematics. Now considering the antisymmetric part of Eq. (3.14), we see that the rotation tensor $\boldsymbol{R}$ corresponding to $\hat{\boldsymbol{u}}$ is changed by the vorticity addition, $\nabla \times \boldsymbol{f}$, to the target rotation corresponding to $\boldsymbol{u}_b$:

$$\Delta R_{ij} = \frac{1}{2}(u_{bi,j} - u_{bj,i}) - \frac{1}{2}(\hat{u}_{i,j} - \hat{u}_{j,i}) = \frac{1}{2}(f_{i,j} - f_{j,i})\frac{\Delta t}{\rho}. \tag{3.18}$$

The pressure correction in Eq. (3.9) creates no additional vorticity. In summary, a reinterpretation of the projection step proposed by Patankar [33] in the context of deforming bodies is as follows. The correction step serves the purpose of changing the velocity gradient tensor field inside the solid domain to the target velocity gradient tensor, which is a linear superposition of: (1) the gradient $\boldsymbol{G}(\boldsymbol{u}_f)$ of the specified deformation velocity $\boldsymbol{u}_f$, and (2) a rigid body motion with zero strain rate and rotation $\boldsymbol{G}(\boldsymbol{u}_r)$. Below we show that this process is felt by the fluid domain through velocity discontinuities at the solid–fluid interface, entering the fluid as a vortex sheet and divergence at the interface.

3.5.2. Velocity discontinuities at the interface

Consider a region Ω_ϵ surrounding Ω_s, just outside Ω_s, with a small thickness ϵ normal to $\partial\Omega_s$, with

$$\lim_{\epsilon\to 0}\Omega_\epsilon = \partial\Omega_s.$$

In the projection step, $\hat{\boldsymbol{u}}$ is corrected to $\boldsymbol{u}_b$. This in general creates a velocity discontinuity at the fluid–solid interface. Here we show that the tangential velocity jump causes a vortex sheet to be deposited at the interface, and the normal velocity jump causes a "divergence sheet" at the interface. The change in total vorticity in Ω_s can be expressed as

$$\Delta\boldsymbol{\omega}_s = \iiint_{\Omega_s} \boldsymbol{\omega}_b\, dV - \iiint_{\Omega_s} \hat{\boldsymbol{\omega}}\, dV, \tag{3.19}$$

where $\boldsymbol{\omega}_b = \nabla \times \boldsymbol{u}_b$ and $\hat{\boldsymbol{\omega}} = \nabla \times \hat{\boldsymbol{u}}$ is the vorticity after and before the projection step and $dV \in \mathscr{R}^3$ is a volume element. Using a version of the Gauss theorem on the elements of the tensor $\epsilon_{ijk}u_k$, we obtain

$$\Delta\boldsymbol{\omega}_s = \iint_{\partial\Omega_s} \hat{\boldsymbol{n}} \times \boldsymbol{u}_b\, dA - \iint_{\partial\Omega_s} \hat{\boldsymbol{n}} \times \hat{\boldsymbol{u}}\, dA = \iint_{\partial\Omega_s} \hat{\boldsymbol{n}} \times [|\boldsymbol{u}|] dA, \tag{3.20}$$

where $[|\boldsymbol{u}|] = (\boldsymbol{u}_b - \hat{\boldsymbol{u}})|_{\partial\Omega_s}$ is the velocity jump across the interface, and $dA \in \mathscr{R}^2$ is a surface element in $\mathscr{R}^3$. In the infinite degree of freedom (continuous) system, this discontinuity is equivalent to a vortex sheet of strength $\gamma = \hat{\boldsymbol{n}} \times [|\boldsymbol{u}|]$ according to the definition $\boldsymbol{\omega} = \gamma\delta(\partial\Omega_s)$, where $\hat{\boldsymbol{n}}$ is the outward normal unit vector at the interface and $\delta(\partial\Omega_s)$ is the Dirac delta function at the interface location. We obtain the total vorticity injected into the fluid due to the vortex sheet in Ω_ϵ:

$$\Delta\boldsymbol{\omega}_\epsilon = \iiint_{\Omega_\epsilon} \boldsymbol{\omega}\, dV = \iiint_{\Omega_\epsilon} \gamma\delta(\partial\Omega_s) dn\, dA = \iint_{\partial\Omega_{\epsilon,out}} \hat{\boldsymbol{n}} \times \boldsymbol{u}_{out}\, dA - \iint_{\partial\Omega_{\epsilon,in}} \hat{\boldsymbol{n}} \times \boldsymbol{u}_{in}\, dA, \tag{3.21}$$

where $\partial\Omega_{\epsilon,out}$ and $\partial\Omega_{\epsilon,in}$ are the outward and inward facing boundaries of Ω_ϵ, with outward defined as away from the solid body, and $\hat{\boldsymbol{n}}$ is the local coordinate normal to the body. $dV = dn\, dA$ where $dn \in \mathscr{R}^1$ is a line element normal to the interface in $\mathscr{R}^3$. Since the correction step leaves the velocity in the fluid unchanged (in the continuum limit), $\boldsymbol{u}_{out} = \hat{\boldsymbol{u}}$, and $\boldsymbol{u}_{in} = \boldsymbol{u}_b$ is the solid body velocity at the interface at the end of the projection step. In the limit $\epsilon \to 0$, this gives

$$\Delta\boldsymbol{\omega}_\epsilon = -\iint_{\partial\Omega_s} \hat{\boldsymbol{n}} \times [|\boldsymbol{u}|] dA. \tag{3.22}$$

Eqs. (3.20) and (3.22) say that the total vorticity change in the solid domain during the projection step is exactly canceled by the vortex sheet deposited on the body surface. Thus the projection step conserves the total vorticity in the domain. The vortex sheet diffuses into the fluid over the next time step. This vorticity generation is fundamentally equivalent to the imposition of the no-slip boundary condition at the interface using a vortex sheet in vortex methods [31].

Similarly, it can be shown using the Gauss theorem for the velocity vector that a divergence sheet of strength $\chi = [|\boldsymbol{u}|] \cdot \hat{\boldsymbol{n}}$, where χ is defined by $\nabla \cdot \boldsymbol{u} = \chi\delta(\partial\Omega_s)$, is injected into the fluid at the interface during the projection step. Step Three of the algorithm generates an appropriate pressure force (see Eq. (3.9)) to remove this non-zero divergence.

Finally, we note that in the continuous system these vorticity and divergence sheets are not present. They are purely artifacts of discretization, and are smeared near the interface by the spatial differentiation scheme and the interpolation scheme.

4. Numerical examples

We present the application of FuRMoRP to problems with different geometries. We choose example problems to span one-, two- and three-dimensional bodies, regular and irregular body shapes, rigid and flexible bodies, and their combinations.

4.1. Rigid thin membrane

In simulations of moving body problems, immersed boundary and immersed interface methods are used usually for solid bodies with specified motion [57] or thin membranes with specified stiffness. Few simulations of flexible membranes with specified velocity are available, mainly due to the difficulties in satisfying the exact velocity boundary condition on the thin interface using an inexpensive non-global interpolation. Taira and Colonius [60] recently reported a discrete Lagrange multiplier method and presented the Stokes' problem as one of the test problems. We study the same problem first, and also a thin plate moving perpendicular to the flow which is more challenging for an immersed numerical method to handle because of the possibility of leakage due to the impinging flow.

As a first step we compare numerical and analytical results for an impulsively started plate, known as Stokes' first problem. A thin horizontal plate of infinite horizontal extent is located at $y = 0$ and is impulsively moved at a constant velocity U_o in the horizontal direction. The transient velocity profile for $y \geqslant 0$ is given as [61]

$$u(y,t) = U_o\left[1 - erf\left(\frac{y}{2\sqrt{\nu t}}\right)\right]. \tag{4.1}$$

Without loss of generality we choose the length and time scales as $L = 1,\ U_o = 1$, and $Re = 1/\nu = 500$. Then the drag coefficient C_d is defined as $F/\frac{1}{2}\rho U_o^2 A$, where F and A are the total viscous force acting on one surface of the plate, and the area per unit depth of that surface. It can be evaluated as

$$C_d = \frac{2}{\sqrt{\pi t Re}}, \tag{4.2}$$

where time has been non-dimensionalized by L/U_o. The numerical and theoretical drag coefficients and velocity profiles are compared in Fig. 2. A grid convergence study shows that the numerical method gives first order for the error norms of the velocity field (Fig. 3) due to the first order interpolation used between the Lagrangian and Eulerian grids. It is known that with sufficiently smooth force density at the interface (e.g. a regularized delta function), the immersed boundary method (IBM) can be formally second order accurate [62]. Usually the formal order of accuracy of IBM is studied with such sufficiently smooth forces. In the presence of singular delta function force densities such as in the Stokes' first problem, the formal second order of accuracy is not practically achievable due to the presence of discontinuity in the velocity derivatives across the interface, although the velocity itself is continuous [62,63,54].

Next we consider a vertical thin plate of height b and infinite extent in the third dimension (see Fig. 4) moving at velocity U in the x-direction in an infinite quiescent fluid. Dennis et al. [64] have derived the 2D solution giving an asymptotic value of the drag coefficient, $C_d = 2.09$ for $Re = Ub/\nu = 20$. For our simulations we take the finite domain dimension $H = 22b$ to keep the periodic boundary condition effects small and the number of grid points 704×704. Through the duration of the simulation, the streamline pattern near the periodic boundaries showed that the perturbation resulting from the image vortices is visually undetectable. The thin plate was represented by a spatial distribution of Lagrangian particles over one grid cell thickness in the x direction. Each grid cell denoting the plate contained eight particles distributed randomly within that cell. The time history of $C_d = F/\frac{1}{2}\rho U^2 b$, where F is the total drag force per unit depth acting on both surfaces combined, is compared with the results of Dennis et al. [64] in Fig. 4.

Since the Reynolds number is low, the small difference between the two can be attributed to the finite extent of the domain. The noise in C_D may be attributed to the high correlation of Lagrangian particle positions in the x direction, whereby all particles transition from one grid cell to another simultaneously, within the time window $\Delta x/U$. This coherence results from the impinging nature of the flow. The spurious oscillations have been recognized as an outcome of the movement of solid–fluid boundaries in a background Eulerian mesh [58]. For more generic flows with flexible bodies, this coherence is expected to be substantially diminished due to the body motion, and as we demonstrate later, the noise in the forces is small in all of

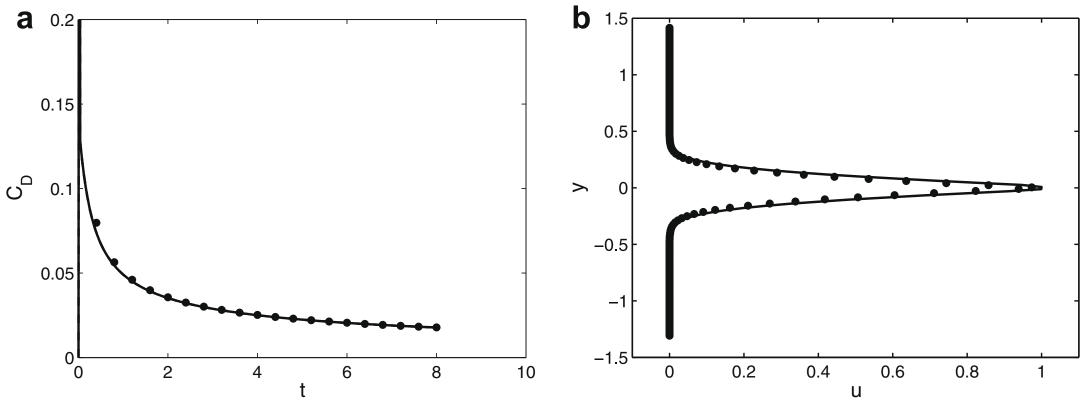


Fig. 2. Comparison of numerical and analytical results for Stokes' first problem. (a) Drag coefficient; (b) velocity profile. —: Numerical; •: analytical.

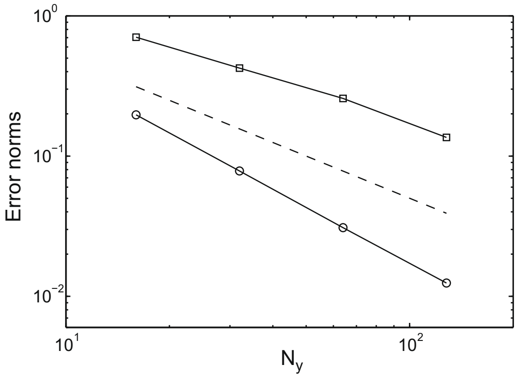


Fig. 3. Behavior of spatial L_2 (— ○ —) and L_∞ (—□—) norms of the error with respect to the number of grid points N_y in the y-direction for Stokes' first problem. Error as a function of the discrete spatial coordinates is defined as the difference between the computed longitudinal (i.e. in the direction of plate velocity) velocity field and its analytical solution at $t = 8$. ---: indicates first order slope.

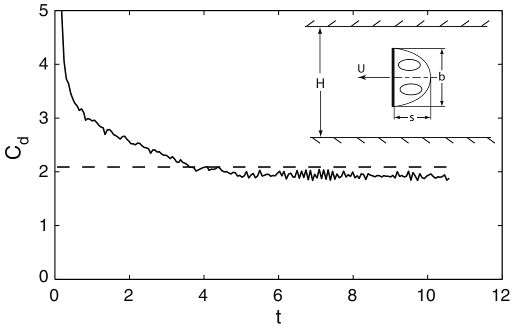


Fig. 4. Drag coefficient for the vertical plate in an unbounded domain at $Re = 20$. Time is non-dimensionalized by b/U. ---: indicates asymptotic value from Dennis et al. [64]. The inset shows the configuration of the problem and the vortices behind the vertical plate.

the numerical tests. The most benign situation is seen in the Stokes' first problem above, where particles do not jump from one grid cell to another in the direction perpendicular to the plate surface. Hence the noise is absent (Fig. 2).

Next, we consider the case when the blocking ratio $\lambda = b/H$ is 0.2 and $Re = 20$ for which Dennis et al. [64], provide flow fields. The upper and lower boundaries are solid walls and in the x direction the boundary conditions are periodic. The solid walls are modeled by frozen Lagrangian immersed particles with zero velocity at all times. The streamline pattern and vorticity field produced by the numerical method (Fig. 5(a)) agree well with those observed experimentally by Taneda and Honji [65] and Dennis et al. [64], and computationally by Bozkurttas et al. [66] and Koumoutsakos and Shiels [67]. Grid spacing of

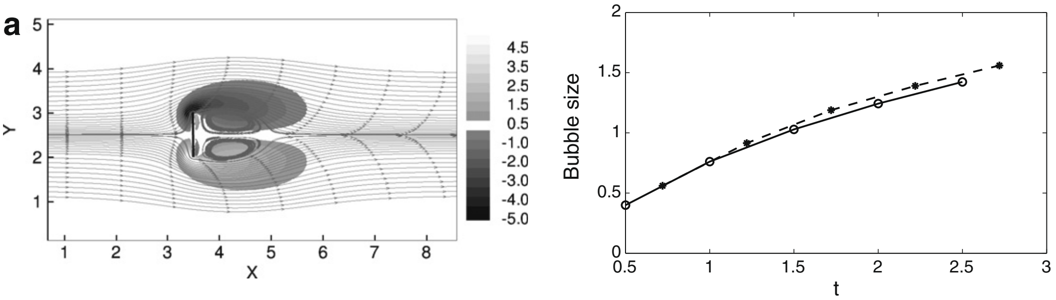


Fig. 5. Results for the vertical flat plate with $\lambda = 0.2$ and $Re = 20$. (a) Vorticity contours and streamlines at $t = 2.5$; (b) bubble size. —: Present study; - - -: Dennis et al. [64].

$b/32$ was used in both directions. Quantitative comparison with experimental results are shown in terms of the bubble size in Fig. 5(b). The bubble size is defined as the horizontal distance of the stagnation point in the wake from the plate. Time is non-dimensionalized by b/U. Since the initial impulse generates a transient which may be captured to different levels of accuracy by experiments and computations, the instance in our results and those of Dennis et al. [64] is matched at the first data point of Dennis et al. [64] to remove the effect of the transient on the comparison. We have also successfully tested our code by calculating the bubble size at a Reynolds number of 126 (see Shirgaonkar et al. [68]).

We compared (see Fig. 6) the time history of the drag coefficient for a higher Reynolds number of 126 to the vortex particle method simulations by Koumoutsakos and Shiels [67]. The immersed boundary formulation causes small spurious oscillations in the drag force as explained above for the case of the plate in an infinite domain. Here we have time-averaged the drag coefficient with a window of four consecutive time steps. Our simulations are with $\lambda = 1/6$ and show excellent agreement with Koumoutsakos and Shiels [67]. The initial impulsive response to the discontinuity in velocity is dependent on the numerics, and cannot be directly compared between the two methods.

4.2. Flexible thin membrane

We now simulate a free-swimming flexible membrane using the momentum redistribution algorithm. A thin, doubly infinite flexible sheet located at $y = 0$ oscillates (in its own frame of reference) with a small amplitude b as

$$y = b\sin(kx - \omega t), \tag{4.3}$$

where $k = 10\pi\ \mathrm{cm}^{-1},\ \omega = 8\pi\ \mathrm{s}^{-1}$ are the wavenumber and angular frequency of oscillations, and x and t are in units of centimeters and seconds, respectively. The density and viscosity of the fluid is $\rho = 1\ \mathrm{g/cm}^3$ and 0.01 g/cm s. The Reynolds number $Re = \rho\omega/\mu\kappa^2$ is 2.5. These choices of parameter values are made to match the same case examined by Fauci and Peskin [59]. The oscillations cause momentum redistribution within the fluid and the sheet acquires a swimming velocity U.

Taylor [69] derived an expression for the swimming velocity using asymptotic expansion for small amplitude Stokes flow:

$$\frac{U}{V} = \frac{1}{2}b^2\kappa^2\left(1 - \frac{19}{16}b^2\kappa^2\right). \tag{4.4}$$

Tuck [70] derived U for finite Reynolds numbers, which for small amplitudes, is given by

$$\frac{U}{V} = \frac{1}{4}b^2\kappa^2\left(1 + \frac{1}{F(R)}\right), \tag{4.5}$$

where

$$F(R) = \left(\frac{1 + \sqrt{1 + Re^2}}{2}\right)^{1/2}. \tag{4.6}$$

Our numerical simulations used a 2D domain size of $\lambda \times \lambda$ where $\lambda = 2\pi/\kappa$ is the wavelength of oscillations. We used a 64×64 uniform grid with periodic boundary conditions in both directions, which is the same resolution used by Fauci and Peskin [59]. A grid sensitivity study, which is shown in Fig. 8, was done to verify that this resolution is sufficient to capture the swimming velocity accurately. This is not a plot for the formal order of accuracy of our approach. Plots for the formal order of accuracy of our scheme, based on L_2 and L_∞ norms, were plotted in Fig. 3 for Stokes' problem. Periodicity in x is equivalent to the infinite extent of the sheet in x. For small amplitudes, the periodicity in y does not affect the results much so that we can compare the velocity to the analytical solutions of Taylor [69] and Tuck [70] which are for infinite domains. We verified this by performing a simulation with domain size $\lambda \times 2\lambda$ for which the results were found to be almost identical to the larger domain results. Even when the sheet amplitude is sub-grid, the numerical method is able to capture the correct

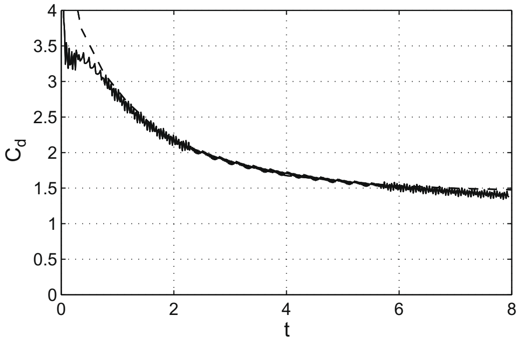


Fig. 6. Time history of drag coefficient for the vertical flat plate at $Re = 126$ with $\lambda = 1/6$. —: Present study; - - -: Koumoutsakos and Shiels [67].

swimming velocity. The only specified motion is the undulating motion of the sheet in its own frame of reference. The swimming velocity is computed as a part of the solution along with the flow field. Fig. 7 shows that our results for U and the flow field match well with Fauci and Peskin [59], Taylor [69] and Tuck [70]. Note that the sheet is free to translate in the x and y directions as dictated by the swimming velocity, and in general it would also experience oscillatory moments with zero mean value resulting from the fluid forces. This would create discontinuities ("holes") in the sheet at the periodic boundaries, which are non-physical and they also give rise to Gibbs' oscillations near the boundaries, manifested as spurious generation of high magnitudes of vorticity. For all simulations we prevented the rotation to avoid this artifact in this test problem. In practical applications, this artifact does not arise because of the finite extent of real organisms.

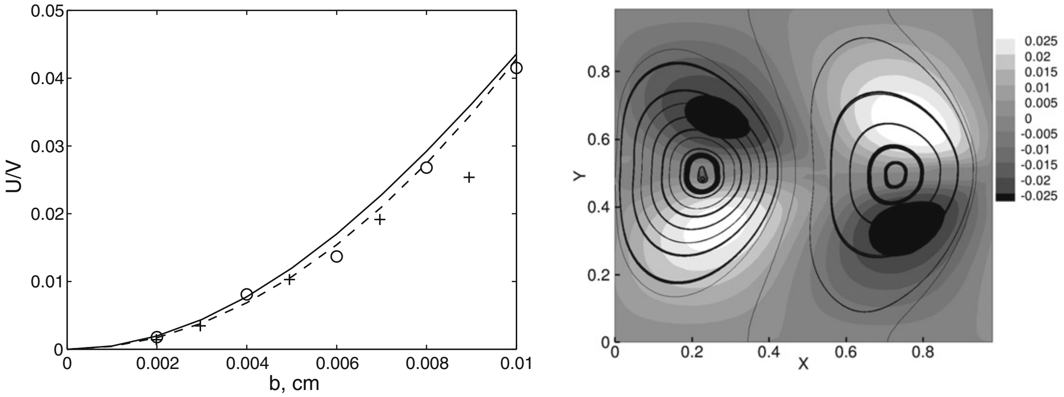


Fig. 7. Swimming velocity from the swimming sheet problem normalized by the phase velocity $V = \omega/\kappa$. —: Taylor [69]; - - -: Tuck [70]; +: Fauci and Peskin [59]; ○: present study. (b) x velocity contours and streamlines. Velocity is normalized by phase velocity.

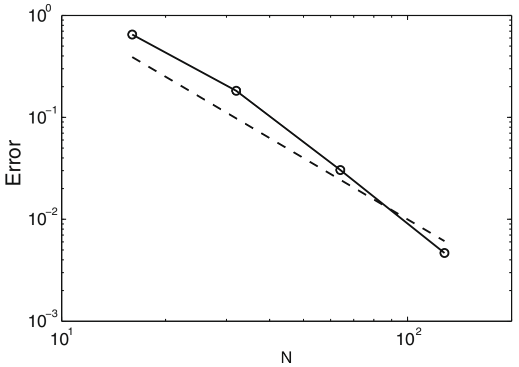


Fig. 8. Error behavior with respect to the number of grid points N in each direction for the swimming sheet problem. Error is defined as $|U - U_A|/U_A$ at steady state, where U_A is the analytical value. - - -: Second order.

Table 2
Drag coefficient and bubble size b for 2D circular cylinder at $Re = 40$.

Study	C_D	b/d
Present	1.52	2.2
Fornberg [71]	1.50	2.24
Marella et al. [72]	1.52	2.30
Kim et al. [55]	1.51	–
Nieuwstadt and Keller [73]	1.55	2.33

4.3. Rigid body

We consider a fixed infinite cylinder of diameter d with an imposed velocity in an infinite domain at $Re = 40$ based on the cylinder diameter. A domain size of $30d \times 30d$ was chosen to keep boundary effects small. The cylinder was initially placed at $(x, y) = (4d, 15d)$ and was moved in the $+x$ direction with speed U. The number of grid points were 750×750, uniform in both directions. The drag coefficient, defined as $C_D = F/\frac{1}{2}\rho U^2 d$, where F is the drag force per unit depth acting on the cylinder, is compared with previous experimental and numerical studies (Table 2), along with the bubble size, defined as the distance of the rear stagnation point from the rear end of the cylinder along the centerline. The value of $C_D = 1.52$ from our simulations compares well with other computational [71,55,72,73] and experimental [74] studies. Our values of C_D lie in the range given in Table 6 in [71] (1.49–1.55) and Table 3 in [72] (1.48–1.52). The time history of the drag coefficient and the steady streamline pattern are shown in Fig. 9.

4.4. Flexible body

As an illustrative example of a flexible body we choose a two-dimensional eel in forward swimming generated by an undulating motion transmitted along its body length. The geometry and the motion specification is taken from Kern and Koumoutsakos [11]. The width of the eel, $w(x)$, grows from zero at the nose to $w_h = 0.04L$ at the head, and then decreases linearly to zero at the tail:

$$w(x) = \begin{cases} \sqrt{2w_h x - x^2} & 0 \leqslant x < x_h, \\ w_h \frac{L-x}{L-x_h} & x_h \leqslant x < L, \end{cases} \tag{4.7}$$

where L is the *projected* length of the eel along its main axis. The main axis is defined as the line about which the displacement of the animal's midline is measured. This displacement is given as [11]

$$y(x,t) = 0.125\frac{x + 0.03125}{1.03125}\sin[2\pi(x - t/T)], \tag{4.8}$$

where T is the period of undulations. The peak undulating velocity (over one cycle) of the tail $(x = L)$ is 0.785. Viscosity is chosen such that the Reynolds number based on the peak undulating velocity and L is 5609, to match Kern and Koumoutsakos [11]. It should be noted that the above coordinates x and y are with respect to the frame attached to the fish body's main axis. A difference between the above geometry and motion pattern and those of Kern and Koumoutsakos [11] is that they specify both quantities as a function of the arc length parameter, s, along the centerline. We choose the projected distance along the main axis, x, so as to keep the body motion divergence-free, which avoids additional mass sources in the domain.

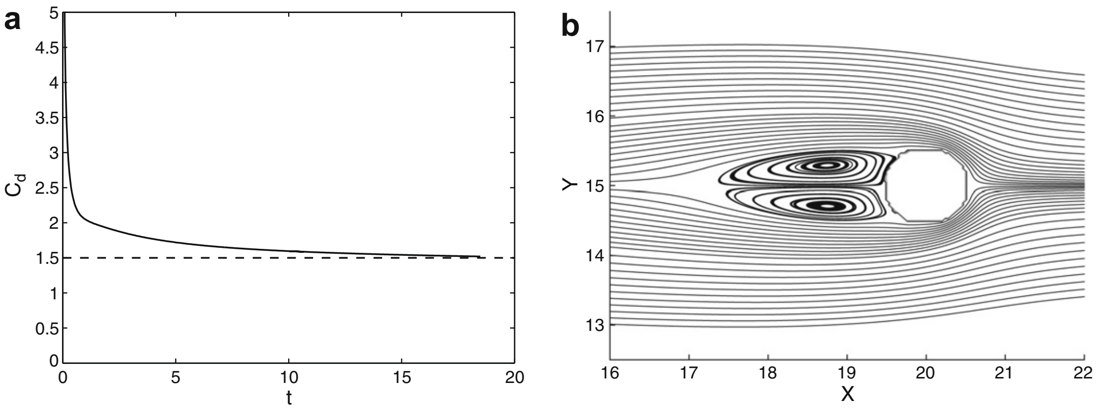
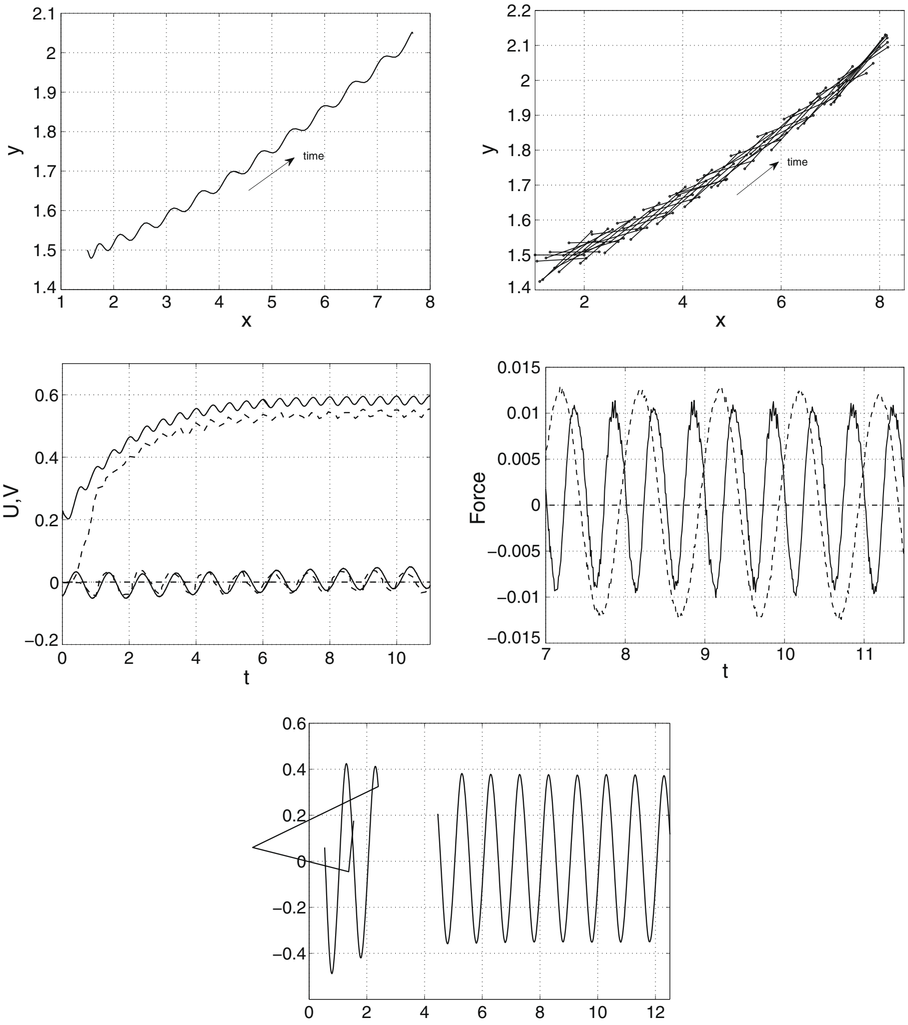


Fig. 9. Results for stationary cylinder at $Re = 40$. (a) Drag coefficient. ---: $C_D = 1.5$. (b) Steady state streamlines.

For the current slender motion pattern, the lateral displacement of the body from the axis is small compared to the body length. The domain size was $8L \times 3L$ with a uniform grid. The initial position of the organism's midline is

$$y(x,0) = 0.125\frac{x+0.03125}{1.03125}\sin(2\pi x), \tag{4.9}$$

which is arbitrarily chosen to have zero phase. Time, distance and velocity are normalized by, respectively, T, L and the velocity scale $U = 1$ which is a consequence of Eq. (4.8). A grid convergence study was done with four different grids: 1280×448, 1600×576, 1920×704, and 2304×832. It was seen that the relative change in mean swimming velocity of the eel between two successive grid refinements fell below 3% for the last stage of grid refinement. We choose the nominal grid with 1920×704 points for simulations reported here. A wider domain in the y direction was tried to study the sensitivity to the periodic boundary conditions, giving almost identical results. At $t = 0$ the flow field is taken as quiescent. Hence there is a transient after which the flow field around the eel reaches its asymptotic structure. During the transient, the moments and forces are not balanced, hence the eel experiences translational and rotational acceleration, manifested as a turn in Fig. 10(a). A quasi-steady state is reached when forces and velocities acquire a constant amplitude with zero mean (Fig. 10(d) and (c)), and the eel's time-averaged trajectory becomes rectilinear. Then the thrust and the drag on the animal balance each other over one complete cycle. The rocking motion pattern (Fig. 10(a)) of the main axis exhibits a quasi-steady



behavior. The angular velocity corresponding to the rigid body component $\boldsymbol{u}_r$ attains a constant amplitude and zero mean (Fig. 10(e)).

The free-swimming velocities show reasonable agreement with the results of Kern and Koumoutsakos [11] (Fig. 10(c)). The slight difference is likely due to the difference between the two deformation patterns. We specify the lateral deformation of the eel in the fixed frame attached to the eel to maintain the overall volume of the body constant, whereas they define it as a function of the body length coordinate. The angle of the asymptotic averaged velocity in this case was found to be 5° with respect to the horizontal direction. As seen from Fig. 11, vorticity is generated over the surface of the body and is then shed at the tail into a reverse Kármán vortex street containing downstream-directed momentum. Each pair of counter-rotating vortices carries with it an impulse in the direction opposite to that of swimming. This demonstrates the central theme of our algorithm – momentum redistribution – where free-swimming is a process in which the body motion imparts a negative momentum to the surrounding fluid, while itself gaining positive total momentum causing propulsion.

The two main features of eel swimming highlighted by Tytell and Lauder [10] are evident in the velocity field in Fig. 12. They are the swirling flow very close to the body on both lateral sides due to the curvature of the body, and the lateral jets generated downstream, each of which lies between a pair of counter-rotating vortices.

We also performed a three-dimensional eel simulation to qualitatively demonstrate that the numerical method reproduces the 3D vorticity structure predicted by the body-fitted finite difference simulation of Kern and Koumoutsakos [11] (Fig. 13). This structure containing alternate vortex rings in a *V*-pattern is also observed in the simulations of low aspect ratio airfoils by Dong et al. [75]. The imposed flow in their case replicates the effect that the self-propulsion velocity has on downstream vortex convection in our case. Hence the two cases produce a similar pattern of vortex rings.

4.5. Free-swimming body with a propelling membrane

A combination of rigid body and flexible membrane occurs in fishes with fins if we neglect the body deformations to the first order. The propulsive force is generated by the undulating motion of one or more fins attached to the fish body, such as pectoral, dorsal, caudal or anal fins. In this section we consider this case to qualitatively demonstrate the capability of the code. As a model problem we consider *Apteronotus albifrons*, the weakly electric black ghost knifefish, which is known for its high maneuverability and swimming without trunk deformations [76]. The simulations are done with a grid size such that the key flow features are resolved. A quantitative investigation is not intended here and will form the basis of a separate study. As a first step, we have recently published a detailed quantitative investigation focused on ribbon fins in knifefish under impulsive conditions (undulating but without translation) in Shirgaonkar et al. [68].

The three-dimensional CAD geometry representative of an adult *Apteronotus albifrons* (Fig. 14(a)) is taken from an accurate cast of the fish made by MacIver and Nelson [20] (Fig. 14(b)). The fish body length L, defined as the distance from the nose to the tail end (shown by circles in Fig. 14(b)), is 10 cm. The anal fin begins at a distance $0.1L$ downstream of the nose, tapers up, and then gradually reduces to zero thickness near the tail. The anal fin is also called a “ribbon fin” because of its geometry.

The fin motion is specified as a sinusoidal wave with frequency $f = 2$ Hz, angular amplitude $\theta_{max} = 30°$, and wavelength $\lambda = L/2$. The angular displacement of any point on the fin is defined with respect to the midsagittal plane of the fish and

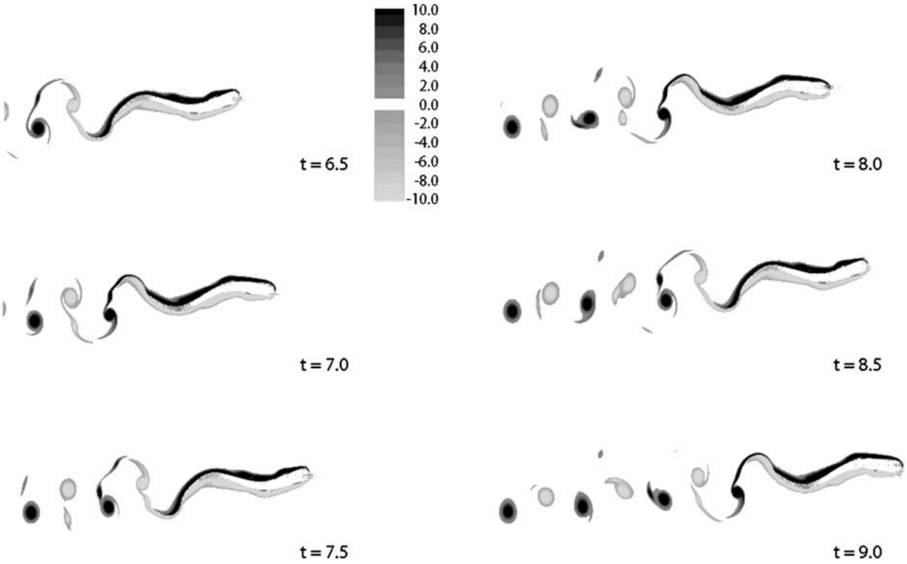


Fig. 11. Vorticity contours for a self-propelling eel. The eel swimming angle is 5° with respect to *x* axis.

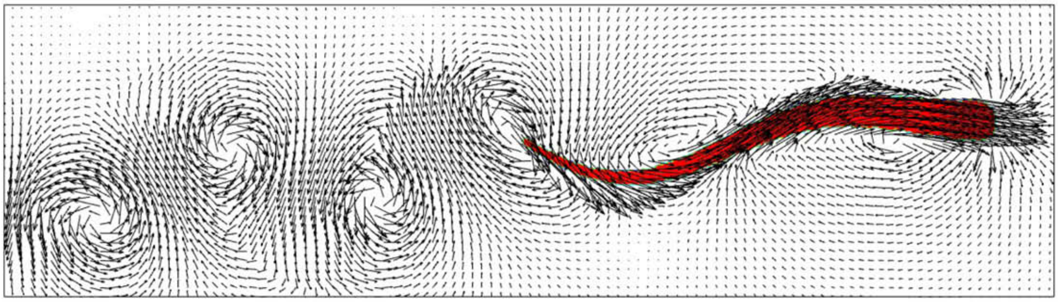


Fig. 12. Instantaneous velocity field for the eel showing vortices close to the body and also the lateral jets downstream of the tail.

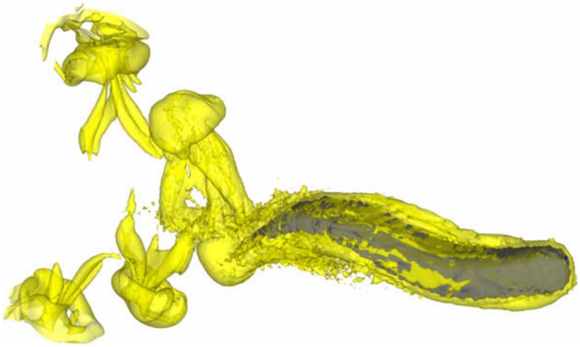


Fig. 13. Isosurfaces of vorticity magnitude 4.5 for a three-dimensional eel. The eel body is shown in dark gray.

about the axis of rotation of the fin. The midsagittal plane passes through the fish nose and tail, and at $t = 0$ it is parallel to the x—y plane (Fig. 14(b)), but subsequently moves according to the rigid body component of the fish motion. The axis of rotation is the line joining the horizontal extrema of the boundary between the body and the fin, and is shown by a dashed line in Fig. 14(b). Note that the tail of the fish (shown by the thick dot in Fig. 14(b)) does not lie on the axis of rotation, because the fin ends at a short distance before the tail. The initial fluid velocity is zero everywhere. The angular displacement is given by

$$\theta = \theta_{max} \sin(kx - \omega t), \tag{4.10}$$

where $k = 2\pi/\lambda$ is the wavenumber and $\omega = 2\pi f$ is the angular frequency. The phase velocity is therefore $U_p = \omega/k = f\lambda$. Thus, the fin excitation is a traveling wave sent along the length of the fin in the downstream direction. A domain size of $3L \times L \times 0.5L$ is used with a $576 \times 192 \times 96$ grid. Here we present the results for swimming velocities and forces, and also the three-dimensional flow field around the fish body. Fig. 15 shows the development of the swimming velocities during the transient behavior of the fish. Time, distance, velocities and forces are non-dimensionalized by L/U_p, L, U_p, and $\rho U_p^2 L^2$, respectively. The swimming velocity rises to about 14% of the phase velocity and begins to stabilize. The pitch angle of the velocity vector with respect to the horizontal is estimated from the projected quasi-steady state mean velocities of $U = 0.15$ and $V = 0.02$ is 7.5°, which lies in the observed range of pitch angles for black ghost knifefish when it is not search-

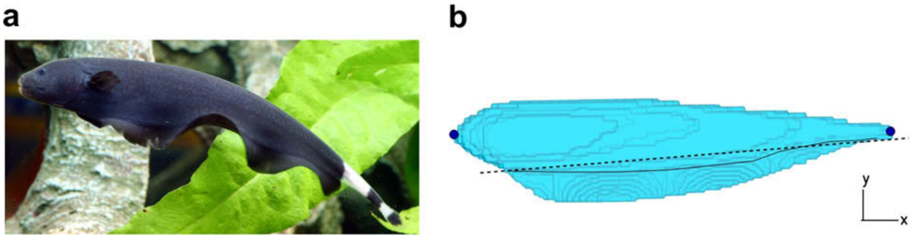
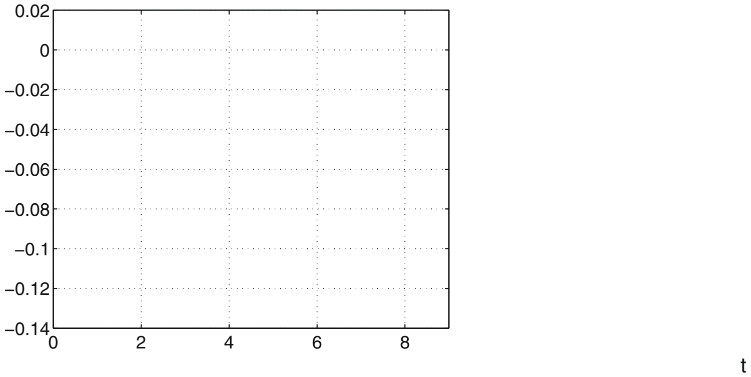


Fig. 14. (a) *Apteronotus albifrons*, the weakly electric black ghost knifefish. Photograph courtesy of Neil Hepworth, *Practical Fishkeeping Magazine*. (b) The 3D geometric model used in the simulations. The thick dots indicate the nose and the tail. The dashed line is the axis of rotation.



ing for prey and is presumably trying to minimize drag. Fig. 16 shows that vorticity is generated on the fin surface, and then shed into the flow along the fin edge, generating propulsive forces. A detailed investigation of this peculiar vortex structure has been reported in [68].

5. Conclusions

We presented a new approach to solve the following problem: given any deforming motion of a swimming body and/or the fins in the frame of reference of the body, compute the swimming velocities and the motion of the surrounding fluid. The approach solves the coupled fluid–body motion with full resolution of the flow field around the swimming body.

The approach presented in this paper is novel for two reasons: first, we have proposed a new constraint-based formulation for the problem of self-propulsion with specified deformation kinematics (Section 2). Second, we have proposed an efficient algorithm, which we call the Fully Resolved Momentum Redistribution for self-Propulsion (FuRMoRP) algorithm, to numerically solve these equations (Section 3).

In the new formulation it is assumed that the entire domain is a fluid. It is ensured that the 'fluid' occupying the domain of the swimming body moves according to the specified deformation kinematics by imposing a constraint based on those kinematics. The unknown pertaining to the body motion is the non-deforming component of its motion which is the swimming velocity. An application of variational principles shows that the constraint leads to a Lagrange multiplier stress field in the body domain similar to how there is pressure in an incompressible fluid. A coupled solution in the fluid–body domains gives the swimming velocity of the body and the velocity field of the fluid. This constraint-based formulation gives a new set of governing equations for the problem of self-propulsion analogous to how distributed Lagrange multiplier methods (DLM) were developed for freely moving rigid particle flows [30,32]. It is known from DLM techniques for rigid particulate flows that such formulations provide a formal basis for the development of fictitious domain or immersed boundary type numerical techniques. Our formulation provides such a basis for the self-propulsion problem.

The new FuRMoRP algorithm shows how the constraint for the self-propulsion problem can be efficiently implemented analogous to how the incompressibility constraint is imposed in Chorin-type pressure correction based fractional time stepping schemes [77]. It is different from prior immersed boundary method-based algorithms for fish swimming [24,25] because FuRMoRP does not need the specification of the entire velocity field in the domain of the swimming body but allows for six degrees of freedom to determine the swimming velocity of that body. There is no need to solve additional equations of motion for the swimming body. The analogy with Chorin-type schemes also implies that FuRMoRP lays the foundation to develop higher order accurate time stepping schemes similar to how the first order pressure correction scheme of Chorin [77] was improved.

The momentum redistribution concept in the FuRMoRP algorithm, which is used to compute the swimming velocity of the body, results from the discrete implementation of the constraint-based formulation of the problem of self-propulsion. This allows for the fluid–body interaction forces to be computed implicitly, rather than explicit computation of forces which is more susceptible to numerical instability. The momentum redistribution approach is also computationally very efficient.

The constraint-based formulation and the FuRMoRP algorithm provide a rigorous basis for an immersed boundary implementation of the problem of self-propulsion. The algorithm can also be implemented by a body-fitted mesh technique. In this work an immersed boundary implementation was used because it results in a powerful technique that can be easily applied to a variety of fish morphologies and swimming styles, and potentially also to flying organisms. The immersed boundary implementation eliminates the need for expensive grid regeneration required by body-fitted meshes. Furthermore, handling arbitrary body shapes and multibody configurations is relatively easy.

We have illustrated the applications of our approach to simple and complex body shapes in one, two, and three dimensions, for rigid and flexible bodies as well as their combinations. We demonstrated that the computational time required for the solid body computation is only a small fraction of the total flow solver.

We also derived vorticity transport equations based on our new constraint-based formulation of the self-propulsion problem. In the discrete system, the formulation was shown to give rise to a vorticity source at the solid–fluid interface which is fundamentally similar to the vorticity source at a boundary in vortex methods.

The formulation does not depend on the constitutive relation of the fluid, and can potentially be applied to turbulent flows using direct numerical simulations (DNS) or large eddy simulations (LES). Finally, the algorithm by itself is independent of the flow solver, is designed to be modular and parallelizable, and can be coupled to various flow solvers to tackle low to high Reynolds number flows. FuRMoRP can be a powerful tool to predict fluid dynamics for biological organisms, underwater vehicle design, physiological fluid dynamics, and integrative neuromechanical modeling.

Acknowledgments

This work was supported by NSF grant IOB-0517683 to M.A.M. Partial support was provided by NSF CAREER grant CTS-0134546 to N.A.P and CBET-0828749 to N.A.P and M.A.M. Computational resources were provided by the San Diego Supercomputer Center through NSF's TeraGrid project grant CTS-070056T to A.A.S., N.A.P. and M.A.M., and in part by a developmental grant from the Argonne National Laboratory to N.A.P.

References

[1] M.H. Dickinson, C.T. Farley, R.J. Full, M.A.R. Koehl, R. Kram, S. Lehman, How animals move: an integrative view, Science 288 (5463) (2000) 100–106.
[2] W.B. Dickson, A.D. Straw, C. Poelma, M.H. Dickinson, An integrative model of insect flight control, in: Proceedings of the 44th AIAA Aerospace Sciences Meeting and Exhibit. PDF AIAA-2006-0034.
[3] Ö. Ekeberg, A combined neuronal and mechanical model of fish swimming, Biol. Cybern. 69 (1993) 363–374.
[4] F.E. Fish, G.V. Lauder, Passive and active flow control by swimming fishes and mammals, Annu. Rev. Fluid Mech. 38 (2006) 193–224.
[5] M.J. Lighthill, Mathematical Biofluiddynamics, SIAM, Philadelphia, 1975.
[6] D.S. Barrett, M.S. Triantafyllou, D.K.P. Yue, M.A. Grosenbaugh, M.J. Wolfgang, Drag-reduction in fish-like locomotion, J. Fluid Mech. 392 (1999) 183–212.
[7] M.J. Lighthill, Aquatic animal propulsion of high hydromechanical efficiency, J. Fluid Mech. 44 (1970) 265–301.
[8] M.J. Lighthill, Large-amplitude elongated-body theory of fish locomotion, Proc. R. Soc. Lond. A 179 (1971) 125–138.

[9] W.W. Schultz, P.W. Webb, Power requirements of swimming: do new methods resolve old questions?, Integr Comp. Biol. 42 (2002) 1018–1025.
[10] E.D. Tytell, G.V. Lauder, The hydrodynamics of eel swimming. I: Wake structure, J. Expt. Biol. 207 (2004) 1825–1841.
[11] S. Kern, P. Koumoutsakos, Simulations of optimized anguilliform swimming, J. Expt. Biol. 209 (2006) 4841–4857.
[12] G.V. Lauder, Function of the caudal fin during locomotion in fishes: kinematics, flow visualization, and evolutionary patterns, Am. Zool. 40 (2000) 101–122.
[13] G.V. Lauder, E.G. Drucker, Morphology and experimental hydrodynamics of fish fin control surfaces, IEEE J. Oceanic Eng. 29 (2004) 556–571.
[14] G.V. Lauder, E.G. Drucker, J. Nauen, C.D. Wilga, Experimental hydrodynamics and evolution: caudal fin locomotion in fishes, Vertebrate Biomechanics and Evolution, Bios Scientific, Oxford, UK, 2003.
[15] G.V. Lauder, J.C. Nauen, E.G. Drucker, Experimental hydrodynamics and evolution: function of median fins in ray-finned fishes, Integr. Comp. Biol. 42 (2002) 1009–1017.
[16] R. Mittal, Computational modeling in bio-hydrodynamics: trends, challenges and recent advances, IEEE J. Oceanic Eng. 29 (3) (2004) 595–604.
[17] Keir Pearson, Orjan Ekeberg, Ansgar Buschges, Assessing sensory function in locomotor systems using neuro-mechanical simulations, Trends Neurosci. 29 (11) (2006) 625–631.
[18] O. Ekeberg, S. Grillner, Simulations of neuromuscular control in lamprey swimming, Philos. Trans. R. Soc. B: Biol. Sci. 354 (1385) (1999) 895–902.
[19] E.M. Standen, G.V. Lauder, Dorsal and anal fin function in bluegill sunfish (*Lepomis macrochirus*): three-dimensional kinematics during propulsion and maneuvering, J. Expt. Biol. 205 (2005) 2753–2763.
[20] M.A. MacIver, M.E. Nelson, Body modeling and model-based tracking for neuroethology, J. Neurosci. Method 95 (2000) 133–143.
[21] I. Akhtar, E. Mittal, G.V. Lauder, E. Drucker, Hydrodynamics of a biologically inspired tandem flapping foil configuration, Theor. Comput. Fluid Dynam. 21 (3) (2007) 155–170.
[22] R. Mittal, H. Dong, G.V. Lauder, P. Madden, Locomotion with flexible propulsors. II: Computational modeling of pectoral fin swimming in sunfish, Bioinsp. Biomim.: Learn. Nat. 1 (2006) S35–S41.
[23] H. Jiang, C. Meneveau, T.R. Osborn, The flow field around a freely swimming copepod in steady motion. Part II: Numerical simulation, J. Plankton Res. 24 (3) (2002) 191–213.
[24] A. Gilmanov, F. Sotiropoulos, A hybrid Cartesian/immersed boundary method for simulating flows with 3D, geometrically complex, moving bodies, J. Comput. Phys. 207 (2005) 457–492.
[25] I. Borazjani, F. Sotiropoulos, Numerical investigation of the hydrodynamics of carangiform swimming in the transitional and inertial flow regimes, J. Expt. Biol. 211 (2008) 1541–1558.
[26] J.D. Eldredge, Numerical simulations of undulatory swimming at moderate Reynolds number, Bioinsp. Biomim. 1 (2006) S19–S24.
[27] 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, J. Expt. Biol. 201 (1998) 3143–3166.
[28] D.J.J. Farnell, T. David, D.C. Barton, Numerical model of self-propulsion in a fluid, J. R. Soc. Interf. 2 (2005) 79–88.
[29] H.H. Hu, N.A. Patankar, M.Y. Zhu, Direct numerical simulations of fluid–solid systems using the arbitrary Lagrangian–Eulerian technique, J. Comput. Phys. 169 (2001) 427–462.
[30] R. Glowinski, T.-W. Pan, T.I. Hesla, D.D. Joseph, A distributed Lagrange multiplier/fictitious domain method for particulate flows, Int. J. Multiphase Flow 25 (1999) 755–794.
[31] J.D. Eldredge, Numerical simulation of the fluid dynamics of 2D rigid body motion with the vortex particle method, J. Comput. Phys. 221 (2) (2007) 626–648.
[32] N.A. Patankar, P. Singh, D. Joseph, R. Glowinski, T. Pan, A new formulation of the distributed Lagrange multiplier/fictitious domain method for particulate flows, Int. J. Multiphase Flow 26 (2000) 1509.
[33] N.A. Patankar, A formulation for fast computations of rigid particulate flows, Center for Turbulence Research Annual Research Briefs, 2001, pp. 185–196. <http://ctr.stanford.edu>.
[34] O.M. Curet, N.A. Patankar, M.A. MacIver, Towards direct numerical simulation of freely swimming fish, in: Amer. Phys. Soc. Div. Fluid Dyn. 59th Annual Meeting, Tampa Bay, FL, USA, November 2006 (Abstract GF. 00005).
[35] A.A. Shirgaonkar, N.A. Patankar, M.A. MacIver, An efficient algorithm for fully resolved simulation of freely swimming bodies, in: Amer. Phys. Soc. Div. Fluid Dyn. 60th Annual Meeting, Salt Lake City, UT, USA, November 2007 (Abstract GF.00005).
[36] L.C. Evans, Partial differential equations, Graduate Studies in Mathematics, Am. Math. Soc., 1998.
[37] I.S. Sokolnikoff, Mathematical Theory of Elasticity, Krieger Publications Co., Malabar, FL, 1983.
[38] C.M. Breder, The locomotion of fishes, Zoologica (N.Y.) 4 (1926) 159–297.
[39] A.A. Shirgaonkar, S.K. Lele, On the extension of the Boussinesq approximation for inertia dominated flows, Phys. Fluid 18 (6) (2006) 066601.
[40] N.A. Patankar, Physical interpretation and mathematical properties of the stress-DLM formulation for rigid particulate flows, Int. J. Comput. Method Eng. Sci. Mech. 6 (2005) 137–143.
[41] C. Dopazo, A. Lozano, F. Barreras, Vorticity constraints on a fluid/fluid interface, Phys. Fluid 12 (8) (2000) 1928–1931.
[42] S. Xu, Z.J. Wang, Systematic derivation of jump conditions for the immersed interface method in three-dimensional flow simulation, SIAM J. Sci. Comput. 27 (6) (2006) 1948–1980.
[43] N. Sharma, N.A. Patankar, A fast computation technique for the direct numerical simulation of rigid particulate flows, J. Comput. Phys. 205 (2005) 439–457.
[44] S.N. Fry, R. Sayaman, M. Dickinson, The aerodynamics of free-flight maneuvers in drosophila, Science 300 (5618) (2003) 495–498.
[45] J.O. Dabiri, M. Gharib, Sensitivity analysis of kinematic approximations in dynamic medusan swimming models, J. Expt. Biol. 206 (2003) 3675–3680.
[46] S.K. Lele, Compact finite difference schemes with spectral-like resolution, J. Comput. Phys. 103 (1992) 16–42.
[47] C. Lui, S.K. Lele, Direct numerical simulation of spatially developing, compressible, turbulent mixing layers, AIAA Paper 2001-0291, Amer. Inst. Aero. Astro., January 2001.
[48] D. Stanescu, W.G. Habashi, 2*N*-storage low dissipation and dispersion Runge–Kutta schemes for computational acoustics, J. Comput. Phys. 143 (1998) 674–681.
[49] A.A. Shirgaonkar, S.K. Lele, Large eddy simulation of early stage aircraft contrails, Technical Report TF-100, Flow Physics and Computation Division, Dept. of Mech. Eng., Stanford University, 2007.
[50] C.S. Peskin, The immersed boundary method, Acta Numer. (2002) 1–39.
[51] G. Iaccarino, R. Verzicco, Immersed boundary technique for turbulent flow simulations, Appl. Mech. Rev. 56 (3) (2003) 331.
[52] R. Mittal, G. Iaccarino, Immersed boundary methods, Annu. Rev. Fluid Mech. 37 (2005) 239–261.
[53] D.V. Le, B.C. Khoo, J. Peraire, An immersed interface method for viscous incompressible flows involving rigid and flexible boundaries, J. Comput. Phys. 220 (2006) 109–138.
[54] S. Xu, Z.J. Wang, An immersed interface method for simulating the interaction of a fluid with moving boundaries, J. Comput. Phys. 216 (2006) 454–493.
[55] J. Kim, D. Kim, H. Choi, An immersed-boundary finite-volume method for simulations of flow in complex geometries, J. Comput. Phys. 171 (2001) 132–150.
[56] E.A. Fadlun, R. Verzicco, P. Orlandi, J. Mohd-Yusof, Combined immersed-boundary finite difference methods for three-dimensional complex flow simulations, J. Comput. Phys. 161 (2000) 35–60.
[57] D. Kim, H. Choi, Immersed boundary method for flow around an arbitrarily moving body, J. Comput. Phys. 212 (2006) 662–680.
[58] M. Uhlmann, First experiments with the simulation of particulate flows, Technical Report 1020, CIEMAT, Madrid, Spain, 2003. ISSN:1135-9420.
[59] L.J. Fauci, C.S. Peskin, A computational model of aquatic animal locomotion, J. Comput. Phys. 77 (1988) 85–108.
[60] K. Taira, T. Colonius, The immersed boundary method: a projection approach, J. Comput. Phys. 225 (2007) 2118–2137.

[61] P.K. Kundu, Fluid Mechanics, Academic Press, 1990.

[62] M.-C. Lai, C.S. Peskin, An immersed boundary method with formal second-order accuracy and reduced numerical viscosity, J. Comput. Phys. 160 (2000) 705–719.

[63] B.E. Griffith, C.S. Peskin, On the order of accuracy of the immersed boundary method: higher order convergence rates for sufficiently smooth problems, J. Comput. Phys. 208 (2005) 75–105.

[64] S.C.R. Dennis, W. Quang, M. Coutanceau, J.-L. Launay, Viscous flow normal to a flat plate at moderate Reynolds numbers, J. Fluid Mech. 248 (1993) 605–635.

[65] S. Taneda, H. Honji, Unsteady flow past a flat plate normal to the direction of motion, J. Phys. Soc. Jpn. 30 (1) (1971) 262.

[66] M. Bozkurttas, H. Dong, V. Seshadri, R. Mittal, F. Najjar, Towards numerical simulation of flapping foils on fixed Cartesian grids, AIAA (2005). 2005-0079.

[67] P. Koumoutsakos, D. Shiels, Simulations of the viscous flow normal to an impulsively started and uniformly accelerated flat plate, J. Fluid Mech. 328 (1996) 177–227.

[68] A.A. Shirgaonkar, O.M. Curet, N.A. Patankar, M.A. MacIver, The hydrodynamics of ribbon-fin propulsion during impulsive motion, J. Expt. Biol. 211 (2008) 3490–3503.

[69] G.I. Taylor, Analysis of the swimming of microscopic organisms, Proc. R. Soc. A 209 (1951) 447.

[70] E.O. Tuck, A note on a swimming problem, J. Fluid Mech. 31 (2) (1968) 305–308.

[71] B. Fornberg, A numerical study of steady viscous-flow past a circular-cylinder, J. Fluid Mech. 98 (1980) 819–855.

[72] S. Marella, S. Krishnan, H. Liu, H.S. Udaykumar, Sharp interface Cartesian grid method. I: An easily implemented technique for 3D moving boundary computations, J. Comput. Phys. 210 (2005) 1–31.

[73] F. Nieuwstadt, H.B. Keller, Viscous flow past circular cylinders, Comp. Fluid 1 (1973) 59.

[74] D.J. Tritton, Experiments on the flow past a circular cylinder at low Reynolds numbers, J. Fluid Mech. 6 (1959) 547–567.

[75] H. Dong, R. Mittal, F.M. Najjar, Wake topology and hydrodynamic performance of lowaspect-ratio flapping foils, J. Fluid Mech. 566 (2006) 309–343.

[76] J.B. Snyder, J.W. Burdick, M.E. Nelson, M.A. MacIver, Omnidirectional sensory and motor volumes in electric fish, PLoS Biol. 5 (11) (2007).

[77] A.J. Chorin, Numerical solution of the Navier–Stokes equations, Math. Comput. 22 (1968) 745–762.