Computer Methods in Applied Mechanics and Engineering 199 (2010) 2417–2424

Contents lists available at ScienceDirect

Computer Methods in Applied Mechanics and Engineering

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

A versatile implicit iterative approach for fully resolved simulation of self-propulsion

Oscar M. Curet [a], Ibrahim K. AlAli [b], Malcolm A. MacIver [a,c], Neelesh A. Patankar [a,*]

[a] Department of Mechanical Engineering, Northwestern University, Evanston, IL 60208, USA
[b] Department of Mechanical Engineering, Stanford University, Stanford, CA 94305, USA
[c] Department of Biomedical Engineering, and Department of Neurobiology and Physiology, Northwestern University, Evanston, IL 60208, USA

ARTICLE INFO

Article history:
Received 16 December 2009
Received in revised form 25 March 2010
Accepted 26 March 2010
Available online 27 April 2010

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

ABSTRACT

We present a computational approach for fully resolved simulation of self-propulsion of organisms through a fluid. A new implicit iterative algorithm is developed that solves for the swimming velocities of the organism with prescribed deformation kinematics. A solution for the surrounding flow field is also obtained. This approach uses a constraint-based formulation of the problem of self-propulsion developed by Shirgaonkar et al. [1]. The approach in this paper is unlike the previous work [1] where a fractional time stepping scheme was used. Fractional time stepping schemes, while efficient for moderate to high Reynolds number problems, are not suitable for zero or low Reynolds number problems where the inertia term in the governing equation is absent or negligible. In such cases the implicit iterative algorithm presented here is more appropriate. We validate the method by simulating self-propulsion of bacterial flagellum, jellyfish (*Aurelia aurita*), and larval zebrafish (*Danio rerio*). Comparison of the computational results with theoretical and experimental results for the test cases is found to be very good.

© 2010 Elsevier B.V. All rights reserved.

1. Introduction

We present a fully resolved simulation (FRS) scheme to compute the self-propulsion of organisms in a fluid. In this work, fully resolved simulation implies that the fluid-solid coupling is not modeled (e.g. by using drag models) but instead the flow around the swimming body is fully computed.

FRS techniques for self-propulsion are relevant in several interdisciplinary areas, e.g., it can be useful to understand the complex interactions between the neuronal, sensory, muscular, and mechanical components of locomotor systems [2]. The maneuverability and efficiency of fish are inspiring new styles of propulsion and maneuvering in underwater vehicles [3]. FRS techniques can serve as a design tool for these applications. FRS can also be an effective tool to investigate the influence of mechanics on the evolution of fish form and function [4].

One FRS approach is to solve the fully coupled elastic body-fluid equations [5,6]. This approach typically takes as input a force field inside the swimming body and outputs the deforming motion and the swimming velocity of the body, and the surrounding flow field. It requires experimental information or an *ad-hoc* estimate on muscle forcing. Muscle forces are typically not available from experiments since it is currently beyond what is understood for best characterized systems.

There are two other classes of FRS approaches: one approach is to simulate the flow around propelling organisms at a *specified* constant swimming velocity [7,8] and specified deforming motion (kinematics). The other approach is to solve for the swimming velocity of the self-propelling organism together with the surrounding flow field [1,9–14] when the deformation kinematics are specified. It is this latter approach that is of interest in this work. Since the deformation kinematics are specified, it excludes the need to solve the elastic response of the body and the fins. Deformation kinematics can be obtained from experimental motion capture data [15].

Motivated by the need to develop an FRS technique that can efficiently handle swimming bodies of a variety of shapes and across different swimming styles, we recently developed a new constraint-based approach [1] for self-propulsion with specified deformation kinematics. In this approach 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. The unknown pertaining to the body motion is the non-deforming component of its motion which is the swimming velocity. A coupled solution in the fluid-body domains gives the swimming velocity of the body and the velocity field of the fluid. Constraint-based approaches for rigid body motion [16,17] or specified velocity fields [18] have been reported in the past.

Shirgaonkar et al. [1] used a fractional time stepping scheme to solve the constraint-based equations for self-propulsion. This algorithm, while efficient for moderate to high Reynolds number problems, cannot be used at zero Reynolds number. This is because the time derivative term in the momentum equation, upon which the fractional time stepping approach is based, is absent in Stokes' flow

* Corresponding author.
E-mail address: n-patankar@northwestern.edu (N.A. Patankar).

0045-7825/$ – see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.cma.2010.03.026

conditions. The approach can also be less efficient for low Reynolds number problems where the inertia term in the governing equation is negligible. In such cases an implicit iterative algorithm is required or preferred [19,20]. This is similar to the how iterative solvers such as SIMPLER [21] are used to obtain steady state or Stokes flow solutions in fluid dynamics instead of using Chorin-type [22] fractional time stepping schemes.

First, we present a new implicit iterative algorithm which we refer to as the Fully Implicit Iterative Self-Propulsion Algorithm (FIISPA) to solve the constraint-based governing equations for self-propulsion. This is a versatile approach that can be easily used to simulate self-propulsion of any fish morphology. FIISPA can perform quasi-Stokes simulation of self-propulsion at zero Reynolds numbers, unlike our previous scheme [1]. However, it is also applicable to problems over a range of Reynolds numbers. Next, we apply our algorithm to the prediction of swimming velocity of live swimming organisms. While there have been comparisons between computed and experimental forces on swimming fish [23], to the best of our knowledge, there has been no direct comparison between swimming velocities computed by *self-propulsion simulations* and experimental data from live swimming organisms during complex motion.

In Section 2 we present the constraint-based formulation for self-propulsion. Section 3 presents the implicit iterative algorithm and its implementation. In Section 4 we present validation of this method by simulating self-propulsion of bacterial flagellum, jellyfish, and larval zebrafish. Section 5 gives concluding remarks.

2. Mathematical formulation

The constraint-based formulation for self-propulsion is summarized below [1]. 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 can be easily extended to multiple bodies. Decompose the velocity field inside the swimming body as $\boldsymbol{u} = \boldsymbol{u}_r + \boldsymbol{u}_f$, where $\boldsymbol{u}_r$ is the rigid motion component of the body velocity and $\boldsymbol{u}_f$ is the deforming motion component of the body velocity 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. 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 key idea behind the constraint-based formulation is to assume that the swimming body is a fluid. However, the "fluid" within the body domain has only six degrees of freedom corresponding to its rigid component of motion. The deformation component $\boldsymbol{u}_f$ of its velocity field is known. Hence, to ensure that the "fluid" within the body domain moves like a swimming body, the following constraint must be imposed on the velocity field $\boldsymbol{u}$ in the body domain,

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

The constraint in Eq. (1) imposes a requirement that the deformation rate tensor associated with the rigid motion component $\boldsymbol{u}_r$ of the body is zero in Ω_s. The governing equations for the combined fluid-body domain are cast as if the entire domain is a fluid and are given by [1],

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

$$\nabla\cdot\boldsymbol{u} = 0 \quad \text{in} \quad \Omega, \tag{3}$$

$$\left.\begin{aligned} \nabla\cdot\boldsymbol{D}(\boldsymbol{u}-\boldsymbol{u}_f) &= 0 \quad \text{in} \quad \Omega_s, \\ \boldsymbol{D}(\boldsymbol{u}-\boldsymbol{u}_f)\cdot\hat{\boldsymbol{n}} &= 0 \quad \text{on} \quad \partial\Omega_s, \end{aligned}\right\} \tag{4}$$

$$\boldsymbol{u} = \boldsymbol{u}_{\partial\Omega} \quad \text{on} \quad \partial\Omega, \tag{5}$$

$$\boldsymbol{u}(\boldsymbol{x}, t=0) = \boldsymbol{u}_o(\boldsymbol{x}) \quad \text{in} \quad \Omega(t=0), \tag{6}$$

where $\boldsymbol{u}$ is the velocity field, p is the pressure, ρ is the density in the fluid-body domain assumed constant and the same everywhere (this is often reasonable for fish swimming, however, extension to unequal densities in the fluid and body domains is straightforward [24]), and μ is the viscosity of the fluid. An incompressible Newtonian fluid is assumed. However, other constitutive forms can be used without any fundamental difficulty. Gravitational body force is not considered since it is balanced by the hydrostatic pressure component in the entire domain. Note that the constraints in Eqs. (1)–(4) are equivalent. Thus, in this formulation the governing equations for the fluid are applicable in the entire domain. To account for the presence of an immersed body there is an additional force density $\boldsymbol{f}$ in the momentum equation due to the rigidity constraint in Eq. (4). $\boldsymbol{f}$ is non-zero only in the body domain. The interface conditions at the fluid-body boundary 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.

3. An implicit iterative algorithm for self-propulsion

In this work, we solve the governing equations for self-propulsion by a new algorithm that generalizes the SIMPLER approach for fluid flow [21] to account for the presence of a swimming body. We use a control-volume formulation with staggered grids. An implicit scheme is used for temporal discretization. Fig. 1 shows the notation for a typical main control volume node P. The domain of our problem is

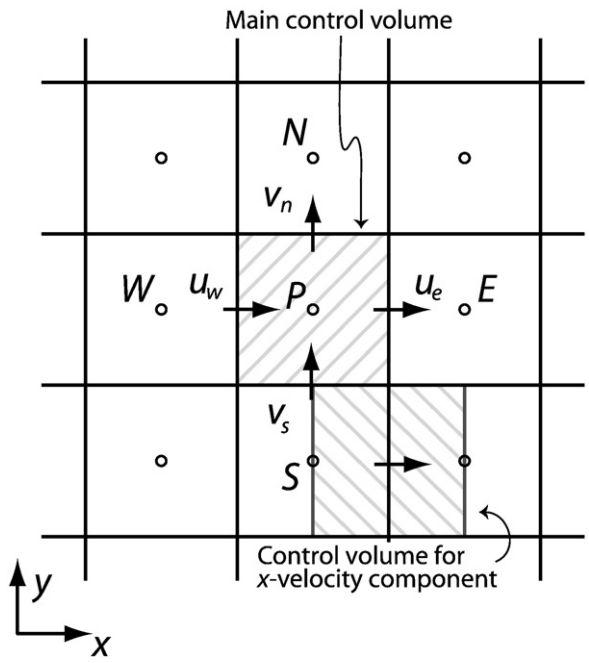

Fig. 1. Notation for the main control volume for pressure and staggered control volumes for velocities. The east, west, north, and south neighbors of node P are denoted by E, W, N, and S, respectively. The top and bottom neighbors on the z-direction (not shown) are T and B. The u velocities are defined at the e and w faces of the main control volume. v_n and v_s are similarly defined. w_t and w_b are defined at the top and bottom faces, respectively, of the main control volume and are not shown in the figure.

divided in control volumes of lengths Δx, Δy, and Δz, in the x, y, and z directions, respectively. The pressure, p, is defined on the main control volume, while the x, y, and z components of the velocity, u, v, and w, respectively, are defined on staggered control volumes. The immersed body is represented by a collection of material points that overlay the background Eulerian grid. The location of a material point is given by the variable $\boldsymbol{X}$. The body force at a staggered control volume location, due to the kinematic constraint, is given by $\boldsymbol{f}$. In our notation, a subscript on a variable represents the node where the variable is defined. For the body forces the direction of the force is indicated as a sub-subscript. For example, $\boldsymbol{f}_{e_x}$ represents the x component of the vector force at node e, and u_e represents u velocity at node e. We also use subscript m to denote a material point. For example, $\boldsymbol{X}_m$ is the location of the material point m.

3.1. Discretization

The continuity Eq. (3) is discretized at the main control volumes. At a typical main control volume, associated with node P, it is given by

$$(u_e - u_w)\Delta y\Delta z + (v_n - v_s)\Delta z\Delta x + (w_t - w_b)\Delta x\Delta y = 0. \tag{7}$$

The momentum Eq. (2) is discretized at the staggered control volumes. The discretized momentum equation at a typical control volume in the x direction is given by

$$a_e u_e = \sum a_{nb}u_{nb} + (p_P - p_E)\Delta y\Delta z + f_{e_x}\Delta V, \tag{8}$$

where subscript nb denotes the respective neighboring nodes, $\sum$ denotes summation over all the neighboring nodes, and $\Delta V = \Delta x\Delta y\Delta z$. The coefficients a are obtained from the spatial and temporal discretizations of the momentum equation (see [21] for details). Momentum equations in the other directions are obtained similarly.

3.2. The FIISPA algorithm

The sequence of steps of FIISPA is shown in Fig. 2. Steps 8 and 9 (dashed box), described below, are the additional steps compared to the SIMPLER algorithm [21] to solve the constraint-based formulation for self-propulsion. At convergence Eq. (4) requires that the velocity field $\tilde{\boldsymbol{u}}$, at the end of Step 7, should be such that $\tilde{\boldsymbol{u}} - \boldsymbol{u}_f$ is a rigid motion in the body domain. In general this condition is not satisfied during intermediate iterations. To eventually get the correct velocity field in the body domain, the force field $\boldsymbol{f}$ in the body domain should be corrected using the following equation:

$$\boldsymbol{f} = \boldsymbol{f}^{**} + \boldsymbol{f}'. \tag{9}$$

The force correction $\boldsymbol{f}'$ is calculated by assuming it to be proportional to the velocity correction $\boldsymbol{u}'$ required in the body domain:

$$\boldsymbol{f}' = \alpha\frac{\rho}{\Delta t}\boldsymbol{u}', \tag{10}$$

where α is a relaxation parameter. We chose to use $\alpha\frac{\rho}{\Delta t}$ as the proportionality factor, however, some other factor could also be chosen. The final converged solution is independent of this choice. The appropriate values for α will be problem dependent; in this work the typical values ranged from $0.5 - 0.8$. A similar approach for force correction in the context of rigid body motion was presented earlier [20]. $\boldsymbol{u}'$ is given by

$$\boldsymbol{u}' = \boldsymbol{u}_b - \bar{\boldsymbol{u}}, \tag{11}$$

where $\boldsymbol{u}_b$ is velocity field in the body domain that satisfies the constraint in Eq. (4). Thus, $\boldsymbol{u}_b$ can be decomposed into rigid ($\boldsymbol{u}_r$) and deformation ($\boldsymbol{u}_f$) velocity fields in the body domain:

$$\boldsymbol{u}_b = \boldsymbol{u}_r + \boldsymbol{u}_f. \tag{12}$$

The rigid component of motion, $\boldsymbol{u}_r$, can be written as

$$\boldsymbol{u}_r = \boldsymbol{U} + \boldsymbol{\omega} \times \boldsymbol{r}, \tag{13}$$

where $\boldsymbol{U}$ is the translational velocity, $\boldsymbol{\omega}$ is the angular velocity, and $\boldsymbol{r}$ is position vector with respect to the center of mass of the body. It follows from Eqs. (11) and (12), that $\boldsymbol{u}'$ should be such that $\tilde{\boldsymbol{u}} - \boldsymbol{u}_f + \boldsymbol{u}'$ is a rigid motion. In addition, the net force and torque in the body domain due to $\boldsymbol{f}'(= \alpha\frac{\rho}{\Delta t}\boldsymbol{u}')$ must be zero, otherwise it would incorrectly lead to a net external force or torque on the self-propelling body. These conditions are satisfied if $\boldsymbol{U}$ and $\boldsymbol{\omega}$ are calculated as follows [1]:

$$M\boldsymbol{U} = \int_{\Omega_s} \rho(\bar{\boldsymbol{u}} - \boldsymbol{u}_f)d\boldsymbol{x}, \tag{14}$$

$$\boldsymbol{I}\boldsymbol{\omega} = \int_{\Omega_s} \boldsymbol{r} \times \rho(\bar{\boldsymbol{u}} - \boldsymbol{u}_f)d\boldsymbol{x} \tag{15}$$

where M, $\boldsymbol{I}$ and ρ are the mass, moment of inertia, and density of the body, respectively.

To summarize, in Steps 8 and 9 of the algorithm, the calculation of $\boldsymbol{U}$ and $\boldsymbol{\omega}$ leads to $\boldsymbol{u}'$ according to Eqs. (11)–(13). The force correction is then computed using Eq. (10) and the force field in the body domain is updated according to Eq. (9). To do these computations the velocity field $\tilde{\boldsymbol{u}}$ obtained at the end of Step 7 of the algorithm is projected on to the material points defining the solid domain by using a delta function interpolation scheme [25]. The calculation of $\boldsymbol{u}'$ (see Eqs. (11)–(13) is performed at the material points and then projected on to the Eulerian grid using delta functions [25]. The force correction $\boldsymbol{f}'$ according to Eq. (10) and the force update according to Eq. (9) is calculated on the Eulerian grid. Note that the velocity field is not corrected in this step. In our simulations, at convergence $\boldsymbol{f}'$ is less then 1% of $\boldsymbol{f}$.

4. Results

In this section we validate the method by focusing on three applications: i) flagellum (Section 4.1), ii) jellyfish (*Aurelia aurita*; Section 4.2), and iii) larval zebrafish (*Danio rerio*; Section 4.3). These cases have been chosen to demonstrate the capability of our approach at Stokes flow conditions (flagellum) and moderate Reynolds numbers (jellyfish, zebrafish), to demonstrate easy application of the method to different organisms, and to quantitatively compare simulation results with experimental data (jellyfish and zebrafish). Additional validations of the method for toroidal swimmers and a moving flat plate normal to the flow are presented in [26].

We used a uniform Cartesian grid with zero velocities at the boundaries of the computational domain for all the test cases. For each case we verify convergence by plotting L_∞, L_1, and L_2 error norms. The error norms are calculated at constant Courant-Friedrichs-Lewy (CFL) number, i.e., for $c\Delta t/\Delta x =$ constant, where c is the velocity scale of the problem. In all cases the lateral velocity that causes the deformation of the swimming organism was used as the velocity scale. The approach is first-order in all cases. This is expected because of the immersed boundary implementation and first-order temporal update of the swimming body. It is noted that our constraint-based formulation can be implemented by body fitted, arbitrary Lagrange-Eulerian, or immersed methods. We chose the immersed boundary implementation due to the ease and applicability of this approach to different morphologies. However, if our approach is implemented by an

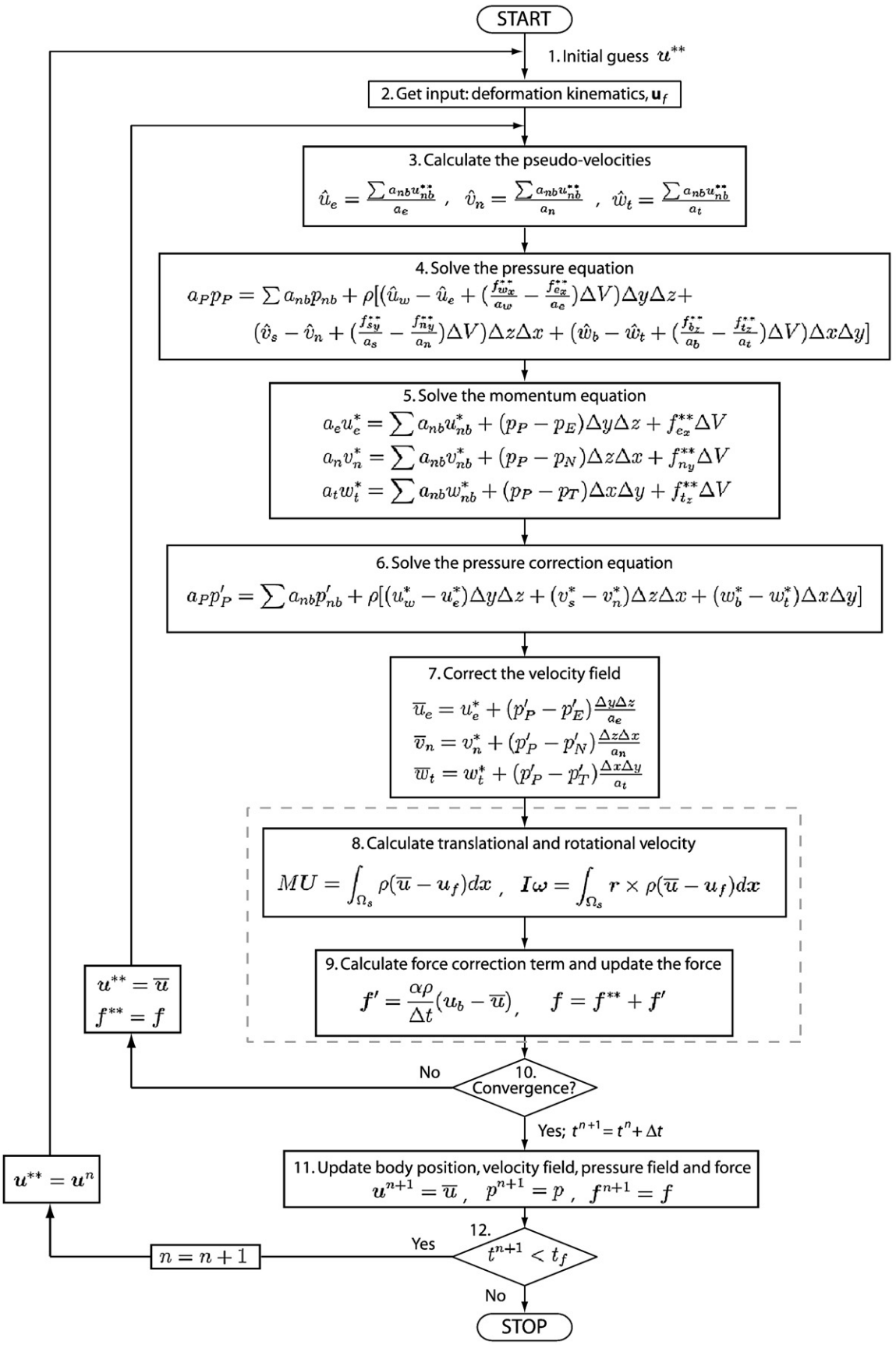

Fig. 2. The FIISPA flowchart. The gray dashed rectangle indicates the steps added to the SIMPLER algorithm to solve the constraint-based formulation of the problem of self-propulsion.

immersed interface type approach together with higher-order update schemes then it could potentially lead to higher-order schemes. The body is represented by material points with spacing approximately equal to that of the Eulerian grid for fluid equations. Since the body is explicitly updated based on the swimming velocity and deformation kinematics, the solution is stable and converges when the *CFL* is *O(1)* or less.

4.1. Flagellum

A flagellum usually propagates plane waves or helical waves. Here, we consider both of these motions and compare our swimming velocity calculations with previous analytic solutions. To demonstrate convergence, we consider a finite thin flagellum with one wavelength, λ, along its length. The flagellum propagates a plane wave given by $y=\eta sin(kx-\omega t)$, where η is the wave amplitude, k is the wave number which is related to the wavelength by $k=2\pi/\lambda$, and ω is the angular frequency. We used a domain size of $(L_x,L_y,L_z)=(2\lambda,0.5\lambda,0.5\lambda)$, the Reynolds number based on wavelength and angular frequency was equal to $Re=\frac{\rho\omega}{\mu k^2}=2.5$, and $\eta k=0.15$. The number of grids were equal in the *x*, *y* and *z* directions for each of the simulations (i.e. $N_x=N_y=N_z$). We considered a constant *CFL* such that $\Delta t/\Delta x=0.2315$ s cm^{-1}. The flagellum was modeled as a one dimensional array of material points with one material point per grid. This choice was made per the recommendation in a prior study [27]. The chosen material point density led to no leakage of flow which we verified from our computations. The average swimming velocities were calculated from one complete wave cycle in steady swimming. Fig. 3A shows first-order convergence according to three norms: L_1, L_2, L_∞ as discussed above.

To demonstrate accuracy we consider the propulsion of flagella that propagate plane or helical waves as a function of wave amplitude. The plane wave motion was given above, and the helical wave motion was given by $y=\eta sin(\theta-\omega t)$ and $z=\eta cos(\theta-\omega t)$, where η is the wave amplitude, θ is a parameter of position, and ω is the angular frequency. For this set of simulations, we consider a finite flagellum of radius $\frac{\lambda}{120}$ with one wavelength along its length, and wavelength λ (θ goes from 0 to 2π in one wavelength). $Re=2.5$ based on wavelength and angular frequency. The fluid domain size was [3λ, 0.5λ, 0.4λ] in the *x*, *y*, and *z* directions, respectively. The grid resolution was$(N_x,N_y,N_z)=(90,60,40)$. The time step was equal to 1.93 ms. The average swimming velocity was calculated from solutions over complete wave cycles. Fig. 4 shows a comparison of the swimming speed, for the flagellum with plane and helical waves, based on both simulation and analytical results. Our computational results are in good agreement with Hancock's analytic solution [28].

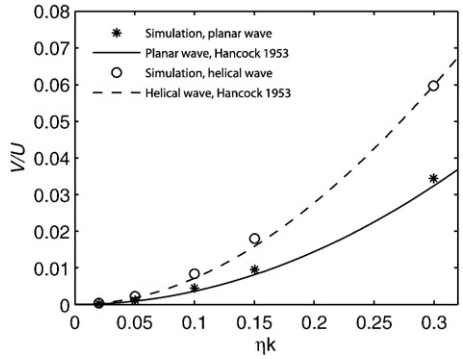

Fig. 4. Comparison of the swimming speed for a flagellum with plane and helical wave as a function of wave amplitude.

4.2. Jellyfish

Self-propulsion simulations of an idealized jellyfish [6] and the crystal jellyfish *Aequorea victoria* [14] have been reported previously. To test our computational method for jet-based propulsion we used experimental kinematics data of the jellyfish *Aurelia aurita* provided by John Dabiri (California Institute of Technology) [29]. The data consisted of the deforming motion of the jellyfish (maximum diameter of approximately 10 cm) with a time resolution of 30 frames per second. The data were given with respect to a reference frame attached to the top end of the jellyfish. Although the swimming velocity of the jellyfish was available from experiments, it was not used in the simulations. Only deformation kinematics were input to the code which solved for the swimming velocity of the jellyfish.

The inset of Fig. 5 depicts the jellyfish outline at the start (black line) and at the end (gray line) of a body contraction. The jellyfish outline was revolved around the axis of symmetry to obtain the three dimensional model. We used a domain size of $(L_x,L_y,L_z)=(20,24,20)$ cm. The density and viscosity of the water were taken as $\rho=1$ gcm^{-3} and $\mu=0.01$ g/(cm s). The Reynolds number based on the calculated swimming velocity and the maximum radius is around 750.

Fig. 3B shows first-order convergence based on jellyfish simulations. In this set of simulations the *CFL* was constant such that $\Delta t/\Delta x=0.0216$ s cm^{-1}. To validate the approach, the numerically solved swimming velocity was compared to the experimental values for time varying complex motion. It can be observed from Fig. 5 that our computational method is able to predict time dependent motion favorably. The main difference between the experimental and computational result is the minimum of the swimming velocity during the relaxation period of the jellyfish. It is possible that the marginal fins on

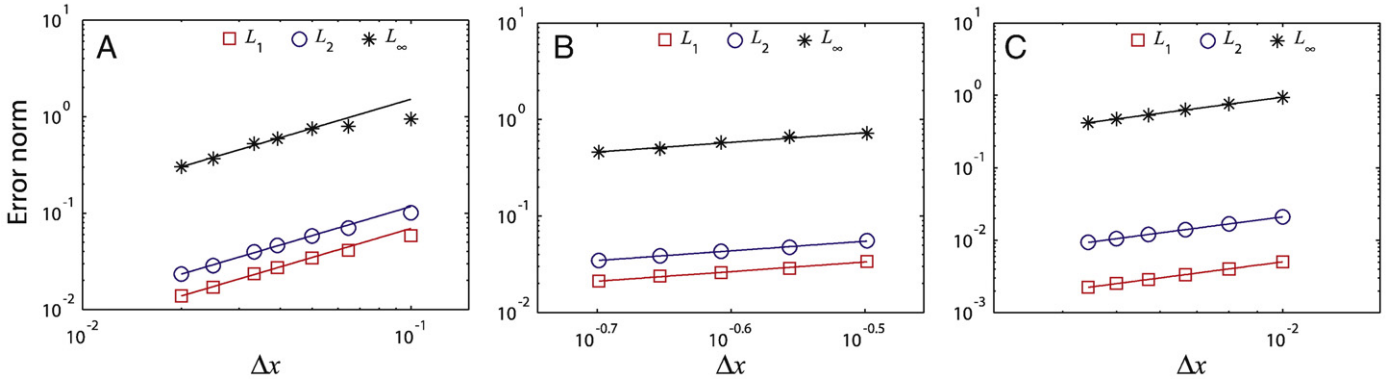

Fig. 3. Norm of the numerical error: □ L_1, ○ L_2, ∗ L_∞. First order slope is indicated with the solid lines. (A) Flagellum convergence ($\Delta t/\Delta x=0.2315$ s cm^{-1}; *CFL* = 0.54). (B) Jellyfish convergence ($\Delta t/\Delta x=0.0216$ s cm^{-1}; *CFL* = 0.22). (C) Zebrafish convergence ($\Delta t/\Delta x=0.08$ s cm^{-1}; *CFL* = 0.55).

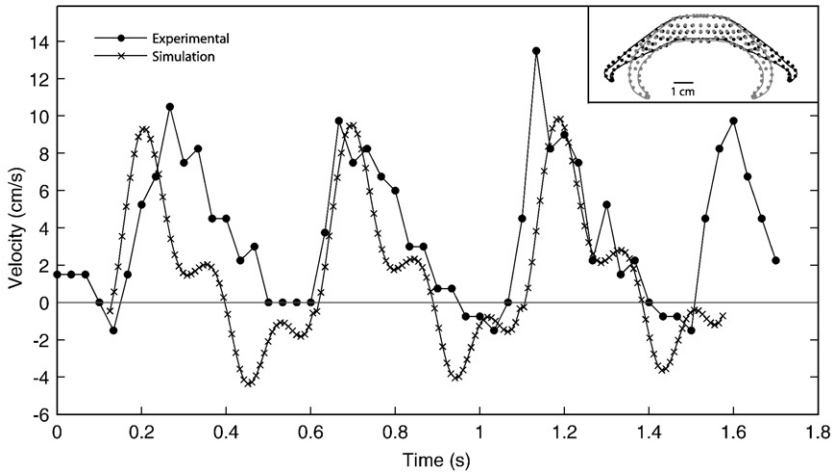

Fig. 5. A comparison between the computed and experimental [29] jellyfish swimming velocities. Inset shows jellyfish outline and material points at the start (black line and black circles) and at the end (gray line and gray circles) of a body contraction.

the jellyfish, which were not included in our simulation, have the effect of reducing the time variation of the forward swimming velocity. Fig. 6 shows a perspective view of the flow field and vortex structure of the jellyfish swimming at the moment of body contraction. The figure clearly illustrates the prominent central jet of the jellyfish swimming during the contraction period. The vortex rings generated by this jet can also be seen. This is in agreement with the experimentally observed flow features [29].

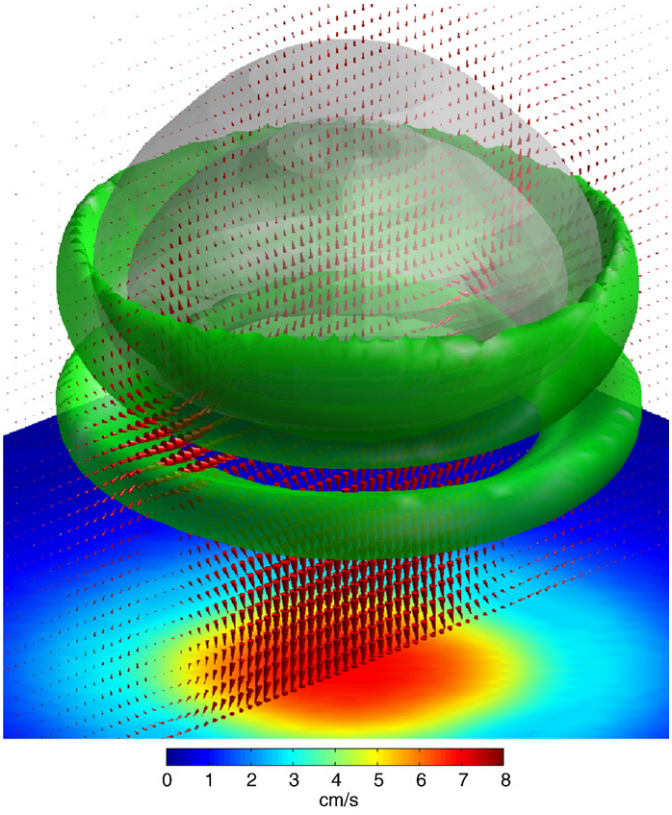

Fig. 6. Perspective view of the flow field and the vortex structure of the jellyfish at the moment of body contraction. The jellyfish is shown in translucent gray color. An isosurface of vorticity magnitude 10 s^{-1} is shown in translucent green. The color map plane shows the magnitude of velocity at a horizontal plane 5.25 cm below the jellyfish. The circular region on this color map illustrates the prominent central jet of the jellyfish swimming during the contraction period.

4.3. Zebrafish

In order to validate our computational code for undulatory swimming we used experimental kinematics data of a larval zebrafish (*Danio rerio*) provided by Melina Hale (The University of Chicago) [30]. The fish was 0.4 cm in length. It had been genetically modified to not develop pectoral fins [30]. The motion of the zebrafish with respect the lab frame were provided with a time resolution of 0.001 s. The kinematics were for approximately four cycles where the zebrafish starts from rest, accelerates while deforming the body, and then gradually decelerates while reducing the wave amplitude. Kinematic data were collected for a total of 0.182 s. Using a lateral view of the zebrafish, a three dimensional surface of the zebrafish was generated. After subtracting the forward swimming velocity of the front end of the zebrafish from the kinematics, the locations and velocities of the internal material points of the zebrafish were input to the code. The domain size considered for this case was $(L_x, L_y, L_z) = (0.4, 1.0, 0.2)$ cm. The density and viscosity of water were taken as $\rho = 1$ g cm^{-3} and $\mu = 0.01$ g/(cm s). The Reynolds number based on the calculated swimming velocity and the body length is around 30.

Fig. 3C shows first-order convergence based on zebrafish simulations keeping $\Delta t/\Delta x = 0.08$ s cm^{-1}. Fig. 7 shows a comparison between the computed and experimental swimming velocities for time varying complex start-up and stopping motion. The differences in the computed and experimental values may be due to two factors, among others: 1) the experimentally measured zebrafish could have experienced a higher drag force due to the proximity of the bottom wall (it was in a petridish), 2) a slight change in the kinematics due to the smoothing and interpolation of the experimental data to obtain a finer time resolution for the simulation. In spite of the these factors, the comparison is very favorable.

Fig. 8 shows a perspective view of the zebrafish and the axial vorticity ($|\omega_y| = 70$ s^{-1}). The isosurfaces of vorticity are shown in red (positive) and blue (negative). The axial vorticity generated by the undulation motion of the zebrafish is similar to that reported previously for eels [12] which also swim using body undulations.

5. Conclusions

We presented a computational approach for fully resolved simulation of self-propulsion of organisms through a fluid. An implicit iterative algorithm, named FIISPA (Fully Implicit Iterative Self-Propulsion Algorithm), is developed that solves for the swimming velocity of the organism for prescribed *deformation* kinematics. A solution for the surrounding flow field is also obtained.

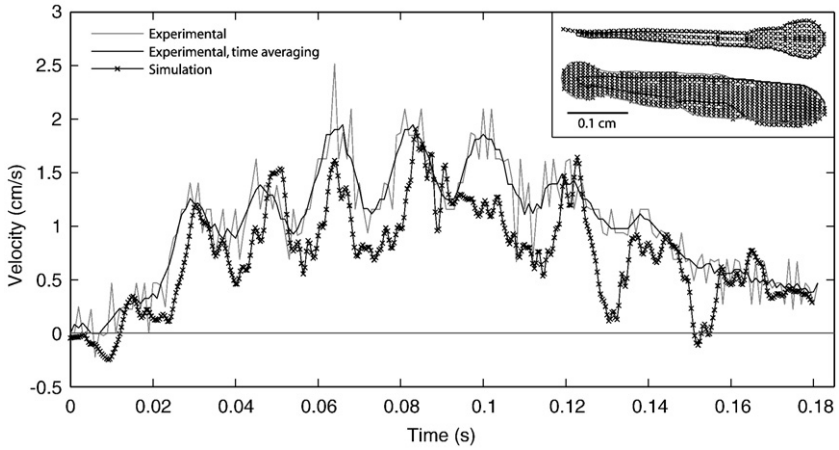

Fig. 7. A comparison between the computed and experimental zebrafish swimming velocities [30]. The time averaging had a span of 5 time steps.

The key novelties of our approach are: First, it is based on a new constraint-based formulation for the problem of self-propulsion with specified deformation kinematics [1]. Second, the FIISPA algorithm is shown to be a simple modification of the SIMPLER algorithm for fluid flow. It is especially effective in zero or low Reynolds number swimming unlike prior fractional time stepping-based schemes [1]. It is also applicable at higher Reynolds numbers.

The FIISPA algorithm (step 8 and 9 of the algorithm) shows how the constraint, based on deformation kinematics, for the self-propulsion problem can be implemented analogous to how the incompressibility constraint is imposed in pressure-based schemes for fluid flow [21]. FIISPA 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. Additional equations of motion for the swimming body are not required.

An immersed body or a body fitted grid approach can be used to implement the FIISPA algorithm. We chose an immersed body implementation since it can be easily applied to different fish morphologies and swimming styles, and potentially also to flying organisms.

We have demonstrated the applications of our approach to a variety of morphologies and swimming styles that span a range of Reynolds numbers. The cases considered were bacterial flagellum, jellyfish, and larval zebrafish. The jellyfish and zebrafish cases were validated against experimental data. We are not aware of prior work where a fully resolved simulation of *self-propulsion* has been validated based on experimental data for live swimming organisms during complex motion.

The FIISPA algorithm is a modular modification of the SIMPLER algorithm. Thus, it can be incorporated without much difficulty into existing fluid flow solvers or any of the commercial packages for fluid flow. Many leading commercial vendors use pressure-based schemes like the SIMPLER algorithm. The approach can also be applied to swimming in non-Newtonian fluids. FIISPA can also be a powerful design tool to develop biomimetic underwater vehicles and for integrative neuromechanical modeling.

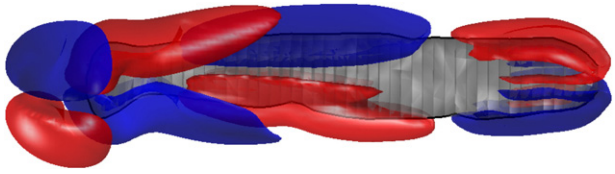

Fig. 8. A perspective view of the zebrafish and the axial vorticity (ω_y). Isosurfaces of vorticity 70 and -70 s^{-1} are shown in red and blue, respectively.

Acknowledgements

We gratefully acknowledge Prof. John Dabiri for providing the jellyfish data and Prof. Melina Hale for providing the zebrafish data. This work was supported by NSF grants IOB-0517683 to M.A.M and CBET-0828749 to N.A.P and M.A.M. Partial support was provided by NSF CAREER grant CTS-0134546 to N.A.P. Support from a DFI fellowship is acknowledged by O.M.C.

References

[1] A. Shirgaonkar, M.A. MacIver, N.A. Patankar, A new mathematical formulation and fast algorithm for fully resolved simulation of self-propulsion, J. Comput. Phys. 228 (2009) 2366–2390, doi:10.1016/j.jcp.2008.12.006.
[2] K. Pearson, O. Ekeberg, A. Buschges, Assessing sensory function in locomotor systems using neuro-mechanical simulations, Trends Neurosci. 29 (11) (2006) 625–631.
[3] M.A. MacIver, E. Fontaine, J.W. Burdick, Designing future underwater vehicles: principles and mechanisms of the weakly electric fish, IEEE J, Ocean. Eng. 29 (3) (2004) 651–659.
[4] P.W. Webb, Form and function in fish swimming, Scient. Am. 251 (1984) 58–68.
[5] L.J. Fauci, C.S. Peskin, A computational model of aquatic animal locomotion, J. Comp. Phys. 77 (1988) 85–108.
[6] H. Zhao, J.B. Freund, R.D. Moser, A fixed-mesh method for incompressible flow–structure systems with finite solid deformations, J. Comp. Phys. 227 (2008) 3114–3140.
[7] 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.
[8] R. Mittal, H. Dong, M. Bozkurttas, F.M. Najjar, A. Vargas, A. von Loebbecke, A versatile sharp interface immersed boundary method for incompressible flows with complex boundaries, J. Comp. Phys. 227 (2008) 4825–4852.
[9] 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. Exp. Biol. 201 (1998) 3143–3166.
[10] 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.
[11] D.J.J. Farnell, T. David, D.C. Barton, Numerical model of self-propulsion in a fluid, J. R. Soc. Interface 2 (2005) 79–88.
[12] S. Kern, P. Koumoutsakos, Simulations of optimized anguilliform swimming, J. Exp. Biol. 209 (2006) 4841–4857.
[13] J.D. Eldredge, Numerical simulations of undulatory swimming at moderate Reynolds number, Bioinsp. Biomim. 1 (2006) S19–S24.
[14] M. Sahin, K. Mohseni, An arbitrary Lagrangian–Eulerian formulation for the numerical simulation of flow patterns generated by the hydromedusa aequorea victoria, J. Comp. Phys. 228 (2009) 4588–4605.
[15] M.A. MacIver, M.E. Nelson, Body modeling and model-based tracking for neuroethology, J. Neurosci. Meth. 95 (2000) 133–143.
[16] 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.
[17] N.A. Patankar, P. Singh, D. Joseph, R. Glowinski, T. Pan, A new formulation of the distributed Lagrange multiplier/fictitious domain method for particulate fows, Int. J. Multiphase Flow 26 (2000) 1509–1524.
[18] K. Taira, T. Colonius, The immersed boundary method: A projection approach, J. Comp. Phys. 225 (2007) 2118–2137.

[19] N. Sharma, N.A. Patankar, A distributed Lagrange multiplier based computational method for the simulation of particulate Stokes flow, Comput. Methods Appl. Mech. Engrg. 194 (2005) 4716–4730.
[20] A. M. Ardekani, S. Dabiri, R. H. Rangel, Collision of multi-particle and general shape objects in a viscous fluid, J. Comput. Phys.
[21] S.V. Patankar, Numerical Heat Transfer and Fluid Flow, Taylor and Francis, 1980.
[22] A.J. Chorin, Numerical solution of the navier-stokes equations, Math. Comput. 22 (1968) 745–762.
[23] M. Bozkurttas, R. Mittal, H. Dong, G.V. Lauder, P. Madden, Low-dimensional models and performance scaling of a highly deformable fish pectoral fin, J. Fluid Mech. 631 (2009) 311–342.
[24] N.A. Patankar, A formulation for fast computations of rigid particulate flows, in: Annual Research Briefs, Stanford University, Center for Turbulence Research, 2001 pp. 185–196.
[25] C.S. Peskin, The immersed boundary method, Acta Numerica (2002) 1–39.
[26] O. M. Curet, An algorithm for fully resolved simulation of self-propulsion through fluid and an analysis of the hydrodynamics of ribbon-fin propulsion, Ph.D. thesis, Northwestern University (2009).
[27] T.T. Bringley, C.S. Peskin, Validation of a simple method for representing spheres and slender bodies in an immersed boundary method for stokes flow on an unbounded domain, J. Comput. Phys. 227 (11) (2008) 5397–5425.
[28] G.J. Hancock, The self-propulsion of microscopic organisms through liquids, Proc. Roy. Soc. A 217 (1953) 96–121.
[29] J. Dabiri, S. Colin, J. Costello, M. Gharib, Flow patterns generated by oblate medusan jellyfish: field measurements and laboratory analyses, J. Exp. Biol. 208 (7) (2005) 1257–1265.
[30] M.H. Green, D.-g. Ahn, R.K. Ho, M.E. Hale, Assessing the function of larval zebrafish pectoral fins during slow swimming, SICB Annual Meeting 2009, Boston, Massachusetts, USA, Abstract. P2.64, 2009.