Published in final edited form as:

Comput Fluids. 2013 April 1; 77: 76–96. doi:10.1016/j.compfluid.2013.02.017.

A parallel overset-curvilinear-immersed boundary framework for simulating complex 3D incompressible flows

Iman Borazjani¹, Liang Ge², Trung Le³, and Fotis Sotiropoulos³

¹Department of Mechanical and Aerospace Engineering, SUNY University at Buffalo, NY, USA

²Department of Surgery, University of California, San Francisco, CA, USA

³St. Anthony Falls Laboratory, University of Minnesota, 2 Third Avenue SE, Minneapolis, MN, USA

Abstract

We develop an overset-curvilinear immersed boundary (overset-CURVIB) method in a general non-inertial frame of reference to simulate a wide range of challenging biological flow problems. The method incorporates overset-curvilinear grids to efficiently handle multi-connected geometries and increase the resolution locally near immersed boundaries. Complex bodies undergoing arbitrarily large deformations may be embedded within the overset-curvilinear background grid and treated as sharp interfaces using the curvilinear immersed boundary (CURVIB) method (Ge and Sotiropoulos, *Journal of Computational Physics*, 2007). The incompressible flow equations are formulated in a general non-inertial frame of reference to enhance the overall versatility and efficiency of the numerical approach. Efficient search algorithms to identify areas requiring blanking, donor cells, and interpolation coefficients for constructing the boundary conditions at grid interfaces of the overset grid are developed and implemented using efficient parallel computing communication strategies to transfer information among sub-domains. The governing equations are discretized using a second-order accurate finite-volume approach and integrated in time via an efficient fractional-step method. Various strategies for ensuring globally conservative interpolation at grid interfaces suitable for incompressible flow fractional step methods are implemented and evaluated. The method is verified and validated against experimental data, and its capabilities are demonstrated by simulating the flow past multiple aquatic swimmers and the systolic flow in an anatomic left ventricle with a mechanical heart valve implanted in the aortic position.

1 Introduction

Handling arbitrarily complex geometries and/or moving boundaries is a major challenge in simulations of real-life biological flows, which requires creative approaches for mesh generation and boundary condition implementation. Many methods, including Chimera overset grid, [1, 2], immersed boundary methods [3, 4], level set methods [5, 6], vortex methods [7, 8], and penalization methods [9, 10, 11] have been developed and successfully applied to specifically address this challenge. In particular, major advances in immersed boundary methods, which are of interest in this work, have made it possible to efficiently

© 2013 Elsevier Ltd. All rights reserved.

iman@buffalo.edu, tel: 1-716-645-1468.

Publisher's Disclaimer: This is a PDF file of an unedited manuscript that has been accepted for publication. As a service to our customers we are providing this early version of the manuscript. The manuscript will undergo copyediting, typesetting, and review of the resulting proof before it is published in its final citable form. Please note that during the production process errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

study flow problems that are far more complicated than those that traditional computational fluid dynamics (CFD) methods (simple structured or unstructured grids) could handle in the past. Recent successful applications of immersed boundary methods include, among others, simulations of prosthetic heart valves [3, 12, 13, 14], biofilming processes [15], flexible fibers [16], flapping filaments [17], aquatic swimming [18, 19, 20], vortex-induced vibrations [21], etc.

The most attractive feature of immersed boundary methods is the inherent separation of grid generation from the underlying geometry. The computational domain, which contains both the fluid and embedded solid regions, is discretized with a single, fixed, non-boundary conforming mesh system, most commonly a Cartesian grid. The effect of immersed boundaries is accounted for by adding forcing terms, either explicitly or implicitly, to the governing equations of fluid motion such that the presence of the appropriate boundary conditions at the location of the immersed boundaries are satisfied [3, 4]. Depending on the specific approach employed to enforce the boundary conditions at immersed bodies, immersed boundary methods are typically classified as diffused [3, 17, 22] or sharp-interface methods [23, 18, 24, 25]—see [4] for a recent review of various approaches. Despite many attractive features of the immersed boundary methods, a number of important limitations makes their application rather challenging for certain types of flows.

The first such limitation arises when applying an immersed boundary method to a flow problem involving moving bodies undergoing large displacements and/or complex multi-connected geometries. For example, when the immersed body is displaced within the background grid over large distances, such as in the case of a falling sphere or a swimming fish, the entire background grid region within which the body is displaced should be discretized with high numerical resolution, which could increase the computational costs considerably. This problem can be solved either by adaptively refining the grid following the motion of the swimmer [26, 22, 27, 28], or, when appropriate, by choosing a non-inertial frame of reference with respect to which the displacement of the body is reduced or completely eliminated [29, 30, 31]. Immersed boundary methods in conjunction with the non-inertial reference frame formulation have indeed been used to reduce the computational cost in simulations of such problems [32, 29, 30]. Note that, however, it is obviously not possible to apply the non-inertial formulation to problems involving multiple moving bodies. Nevertheless, for problems involving a single moving body the non-inertial frame formulation is preferred over the adaptive grid method because of its overall simplicity and lower computational costs. Furthermore, efficient parallelization of an adaptive grid solver is not straight forward and continues to be the subject to active research [27]. The standard non-inertial frame formulation of the Navier-Stokes equations contains source terms for the translational, Coriolis, and centrifugal accelerations. Such source terms reduce the stability of the numerical algorithm. For that a conservative formulation has been proposed, which does not have source terms and has improved stability properties [31, 33]. This formulation, however, has thus far been applied in conjunction with immersed boundary methods only on Cartesian grids [30].

The second limitation arises when the Cartesian-grid based immersed boundary methods are applied to internal flow problems, especially multi-branching configurations typically encountered in cardiovascular anatomies, pulmonary airways, etc (Fig. 1). Although in such problems one could use the brute-force approach, namely discretizing the entire computational domain with a single structured background grid and embedding the entire complex geometry of the solid walls as an immersed body, such treatment results in a large number of wasted computational nodes located outside the regions of interest and is thus very impractical [34]. To overcome such difficulties different strategies have been proposed. One approach is to discretize the multi-connected domain using body-fitted unstructured

grids [35, 36] while another is to recast the structured Cartesian formulation into an unstructured Cartesian grid layout [34]. These methods, however, change the structured layout of the grid making it difficult to use efficient parallel structured solvers. A more versatile modeling paradigm, which we propose in this work, is to integrate the overset grid approach with structured curvilinear grids, and a sharp-interface immersed boundary method. For example, with reference to Fig. 1, the left ventricle, which undergoes large deformation during the cardiac cycle, is embedded in a background curvilinear grid and treated as an immersed boundary, the aorta may be discretized with a separate boundary-fitted curvilinear grid or also treated as an immersed boundary in a background curvilinear domain approximating the shape of the aorta, and the mechanical valve prosthesis or any other native or prosthetic aortic valve, which typically undergoes large displacement and/or deformation, is also treated as an immersed boundary. The background left ventricle and aorta grids overlap in the left-ventricle outflow track region and the solutions in these two sub-domains communicate using the overset grid approach. Such hybrid approach could be extended to include other blood vessels in the simulations, such as the subclavian and carotid arteries shown in Fig. 1, and used to model efficiently a broad range of complex internal flows with embedded immersed boundaries undergoing large displacement and/or deformation. It is the main objective of this paper to develop the computational infrastructure required to achieve such general and versatile modeling paradigm.

In traditional overset grid formulations, pioneered by [37, 38, 39, 40, 1], a complex flow domain is decomposed into a set of simpler, overlapping subdomains such that it can be discretized easily with a set of simple, boundary conforming, curvilinear coordinates [41]. The governing equations are solved independently on each subdomain and information from one subdomain is transferred to another subdomain by specifying the boundary conditions at their interfaces [42, 41]. This is similar to the flux-exchange method [43, 44, 45, 46] (not the state-exchange method based on domain decomposition [47, 48, 49]) employed in coupling continuum and molecular dynamics domains in multi-scale simulations [50]. More specifically, the flux-exchange method is based on the direct exchange of flux information in the overlap domain between the particle region and the continuum region, and relies on the matching of fluxes of mass, momentum and energy [48]. In the state-exchange method the state information between the particle simulation and the Navier-Stokes equations is transferred through an overlap region where the particles' dynamics is constrained; the constrained dynamics is often imposed via a dynamic relaxation technique [48].

The simplest approach to specify the boundary conditions at the interface is to interpolate all the variables from one subdomain to the other [1, 51, 2, 52]. Such interpolation, however, does not necessarily satisfy global conservation, and for that a critical aspect of the overset grid method is the development of conservative interpolation schemes [53, 42, 41, 54, 55, 56]. However, if a fractional step method is used to advance the incompressible flow governing equations in time the intermediate velocities are not conservative quantities—do not satisfy mass conservation. Therefore, any type of conservative interpolation from such non-conservative velocities cannot satisfy global mass conservation. Consequently, an explicit mass-imbalance correction needs to be added to the interpolated velocity field to ensure global mass conservation on each overlapping grid. Zang and Street [57] add such an explicit correction, which is proportional to the local flux at each cell, and Burton and Eaton [58] add an explicit correction, which is proportional to the local area of each cell. As shown by Zang and Street such corrections retain the second order accuracy of the solution [57].

In this paper we build on the computational algorithms developed by our group to develop a new hybrid computational framework, which generalizes the sharp-interface CURVIB method [59, 60] to a non-inertial frame of reference and integrates it for the first time with overset grids to enable high-resolution, fluid-structure interaction simulations of real-life

biological flows, which could not be tackled with the algorithms we have developed in the past. To advance toward this modeling paradigm a number of major algorithmic advances are reported in this paper. These include: 1) extending the conservative non-inertial formulation of [31, 33] to generalized curvilinear co-ordinates; 2) developing conservative interpolation algorithms by explicit mass correction for overset grids suitable for the fractional step method used in the CURVIB method; and 3) developing communication strategies for transferring data among sub-domains to enable efficient parallel implementation of the resulting overset-CURVIB flow solver. The versatility of the numerical method is demonstrated by applying it to simulate challenging biological fluid mechanics problems, including fish swimming in a school arrangement and the systolic flow patterns in an anatomic left ventricle with a mechanical bileaflet valve implanted in the aortic position.

The paper is organized as follows: In the numerical methods section 2, we present the governing equations in the non-inertial frame and their transformation into curvilinear coordinates. We briefly describe the CURVIB solver in the non-inertial frame, then focus on the conservative interpolation for the fractional step methods and the communication strategies for parallel implementation of the overset grids. In the numerical results section 3, we present several cases to show the accuracy and capabilities of our method. The rotationally oscillating cylinder shows the accuracy of the non-inertial frame, while the steady and pulsatile flow through the 90° bend validates the overset grid approach. The 2nd order accuracy of our overset grid approach is verified by comparing it against a benchmark solution of the lid-driven cavity flow on a single grid. The overset-CURVIB approach is compared in terms of flow field, computational time, and parallelization with the single grid CURVIB approach for a self-propelled mackerel in section 3.4. The capabilities of the method are demonstrated by the first 3D simulations of multiple swimmers as well as the first simulation of a mechanical heart valve driven by the motion of the left-ventricle. Finally, the new findings of this work are summarized and the needs for future work are discussed.

2 Numerical method

For the sake of clarity we adopt the following notation in this paper. *Italic variables* are scalar while the **Boldface variables** are vectors. **Boldface with underline** denotes a matrix.

2.1 Governing equations in a general non-inertial frame of reference

The governing equations for the fluid domain are the three-dimensional, unsteady incompressible continuity and Navier-Stokes equations. To reduce the displacement region of the immersed boundary relative to the grid, as discussed in the introduction, we formulate the governing equations in a general arbitrarily moving non-inertial frame of reference (Fig. 2). The non-inertial formulation with respect to relative velocities produces source terms (e.g., Coriolis) that reduce the stability of the numerical algorithms. Therefore, a conservative form of the equations in the non-inertial reference frame, developed by Vinokur [33] and Beddhu et al. [31] and used by Kim and Choi [30], is used in this work. The conservative formulation in tensor notation reads as follows:

$$\frac{\partial u_r}{\partial x_r} = 0 \quad (1)$$

$$\left(\frac{\partial u_q}{\partial t} \right) + \frac{\partial}{\partial x_r} \left[(u_r - v_r) u_q + u_r w_q \right] = - \frac{\partial p}{\partial x_q} + \frac{1}{Re} \frac{\partial^2 u_q}{\partial x_r \partial x_r} \quad (2)$$

where,

$$u_q = u_q^{non-int} + v_q = Q_{qr} u_r^{int} \quad (3)$$

$$v_q = w_q + u_q^{ctr} \quad (4)$$

$$w_q = \epsilon_{qlm} \Omega_l x_m^{int} \quad (5)$$

where, u_q^{ctr} and Ω_q are the translational and rotational velocity of the non-inertial frame, respectively, relative to the inertial frame as shown in Fig. 2. $u_q^{non-int}$ and u_q^{int} are the Cartesian components of the velocity vector as observed by a viewer in the non-inertial and inertial frame, respectively. Note that u_q is *not* equal to the absolute (inertial) velocity (u_q^{int}) if the non-inertial frame is rotating. In fact, it has the same magnitude but has been rotated similar to the rotation of the non-inertial frame relative to the inertial frame. v_q can be considered as the components of the grid velocity vector due to the motion of the non-inertial reference frame while w_q is component of the grid velocity vector due to the rotation of the non-inertial reference frame, and ϵ_{qlm} is the alternating tensor indicating the cross product of the vectors Ω and $\mathbf{x}^{int}$. The Q_{qr} is the orthogonal rotation tensor that rotates the non-inertial frame to the inertial frame orientation—i.e. $x_q^{non-int} = Q_{qr} (x_r^{int} - x_q^{ctr})$ ($q, r=1$ to 3) (see Fig. 2). The $x_q^{non-int}$ and x_q^{int} are the components of the position vectors in the non-inertial and inertial reference frames, respectively, and the origin of the non-inertial reference frame is positioned at $\mathbf{x}^{ctr}$ from the inertial reference frame (see Fig. 2). The equations in the inertial reference frame can easily be recovered by setting the translational and angular velocities of the non-inertial frame to zero.

2.2 Curvilinear coordinate transformation of the governing equations in a general non-inertial frame

In order to use curvilinear grids, we transform the Navier-Stokes equations from the (x_1, x_2, x_3) Cartesian coordinates, Eqns. (2), to (ξ^1, ξ^2, ξ^3) curvilinear coordinates by means of a generalized curvilinear coordinate transformation defined as:

$$\xi^r = \xi^r(x_1, x_2, x_3) \quad r=1, 2, 3 \quad (6)$$

The fully transformed equations in tensor notation (repeated indices imply summation) read as follows ($r, q = 1, 2, 3$) [60]:

$$J \frac{\partial}{\partial \xi^r} (U^r) = 0 \quad (7)$$

$$\frac{\partial U^r}{\partial t} = \frac{\xi_q^r}{J} \left(\frac{\partial u_q}{\partial t} \right) = \frac{\xi_q^r}{J} \left(-C_1(u_q) - C_2(w_q) - G_q(p) + \frac{1}{Re} D(u_q) \right) \quad (8)$$

where C_1 , C_2 , G , and D are the two convective, gradient, and viscous operators defined in curvilinear coordinates as follows:

$$C_1(\cdot) = J \frac{\partial}{\partial \xi^r} [(U^r - V^r) \cdot] \quad (9)$$

$$C_2(\cdot) = J \frac{\partial}{\partial \xi^r} [U^r \cdot] \quad (10)$$

$$G_q(\cdot) = J \frac{\partial}{\partial \xi^r} \left(\frac{\xi_q^r}{J} \cdot \right) \quad (11)$$

$$D(\cdot) = J \frac{\partial}{\partial \xi^r} \left(\frac{g^{rm}}{J} \frac{\partial}{\partial \xi^m} \cdot \right) \quad (12)$$

and J is the determinant of the Jacobian of the transformation

$J = |\partial(\xi^1, \xi^2, \xi^3) / \partial(x_1, x_2, x_3)|$, $\xi_q^r = \frac{\partial \xi^r}{\partial x_q}$, g^{rm} , is the contravariant metric of the transformation, $g^{rm} = \xi_q^r \xi_q^m$, and U^r and V^r are the contravariant volume flow rates (volume fluxes hereafter) due to fluid and grid velocity, respectively, which are related to the Cartesian velocity components as follows:

$$U^r = u_m \xi_m^r / J, \quad V^r = v_m \xi_m^r / J \quad (13)$$

$$u_m = U^r J \frac{\partial x_m}{\partial \xi^r}, \quad v_m = V^r J \frac{\partial x_m}{\partial \xi^r} \quad (14)$$

In the above equations u_m , v_m and w_m are the components of the fluid and grid velocity vectors defined by Eqns. (3) to (5), respectively.

2.3 The CURVIB solver in the non-inertial frame

In this section we extend the curvilinear/immersed boundary (CURVIB) solver in the inertial frame [60] to a general non-inertial frame. Similar to the inertial frame version [60], a fully-curvilinear staggered grid discretization approach is employed, which does not require either the explicit evaluation of the Christoffel symbols or the discretization of all three momentum equations at cell interfaces. The flow Eqn. 7 and 8 are advanced in time using a fractional step method proposed by Ge and Sotiropoulos [60]. In the first step, the momentum equations (Eqn. 8) are discretized in a fully implicit manner using second-order backward differencing in time:

$$\frac{3U^{(*)} - 4U^{(n)} + U^{(n-1)}}{2\Delta t} = RHS(U^{(*)}, u^{(*)}, p^{(n)}) \quad (15)$$

where n denotes the time level and RHS is the right hand side of Eqn. 8. The convective term $C_2(w_q)$ Eqn. 10 and the viscous term $D(u_q)$ Eqn. 12 in the RHS are discretized using QUICK and the three-point central finite differencing schemes, respectively. The details of the convective term $C_1(u_q)$ Eqn. 9 discretization in the non-inertial frame can be found in the appendix.

Eqn. 15 is solved implicitly using a Matrix-Free Newton-Krylov method [60] to obtain the intermediate fluxes $U^{(*)}$. The intermediate fluxes $U^{(*)}$ are not divergence-free and need to be corrected to satisfy the continuity Eqn. 7. This is accomplished by solving the following Poisson equation for the pressure correction $\phi = p^{(n+1)} - p^{(n)}$:

$$-\frac{\partial}{\partial \xi^r} \left(\frac{\xi_q^r}{J} G^q(\phi) \right) = \frac{3}{2\Delta t} \frac{\partial}{\partial \xi^r} (U^{r(*)}) \quad (16)$$

The Poisson Eqn. 16 is solved using GMRES with multigrid as a preconditioner [60]. Following the solution of the Poisson equation, the pressure and contravariant volume fluxes are updated as follows:

$$p^{(n+1)} = p^{(n)} + \phi \quad (17)$$

$$U^{r(n+1)} = U^{r(*)} + \frac{2\Delta t}{3} \frac{\xi_q^r}{J} G^q(\phi) \quad (18)$$

To solve the governing equations, boundary conditions at the domain's outer boundaries as well as at the inner moving walls should be supplied. Only the fluxes (U^r) are required at the boundaries, i.e., no pressure boundary conditions are required for the solution of the momentum Eqns. 15 due to the fully staggered formulation of the CURVIB [18, 60]. To have a well-posed pressure-correction scheme, proper application of boundary conditions is essential [61]. The proper boundary conditions for the pressure-correction Eqn. 16 are the Neumann conditions derived from the momentum Eqn. 18 [61] (see also section 9.3.2 of [62]):

$$\frac{\partial \phi}{\partial n} = \frac{3}{2\Delta t} \frac{J}{\xi_q^r} (U^{r(n+1)} - U^{r(*)}) \cdot n_q \quad (19)$$

where n_q is normal to the boundary. However, in the above equation $U^{(n+1)}$ might not be known a priori at some boundaries, e.g., outlet, or overset grid interfaces. $U^{(n+1)}$ at these boundaries cannot be calculated because this is basically a chicken and egg problem: (1) to obtain the corrected flux $U^{(n+1)}$ from Eqn. 18 on the boundaries, ϕ is required; and (2) to obtain ϕ , the corrected flux $U^{(n+1)}$ at the boundary is required for the pressure-correction boundary condition Eqn. 19. To solve this issue, therefore, it is customary to assume $U^{(n+1)} = U^{(*)}$, which is $\mathcal{O}(\Delta t^2)$ approximation. This reduces the boundary condition for the pressure-correction Eqn. 19 to:

$$\frac{\partial \phi}{\partial n} = 0 \quad (20)$$

With this Neumann boundary condition, the pressure at all of the cell centers are obtained by solving the Poisson Eqn. 16 to satisfy the continuity equation at each cell to machine zero [18, 60].

The 3D, arbitrarily complex moving boundaries inside the domain are handled with a sharp-interface immersed boundary method [18]. The method blanks out the nodes inside the immersed bodies (solid nodes), i.e., these nodes do not affect the solution. The boundary conditions are reconstructed on the fluid nodes in the immediate vicinity of the immersed boundary (IB nodes) along the normal to the boundary [18]. The equations are solved on the rest of the grid nodes (fluid nodes) with the boundary conditions supplied at the IB nodes. The method has been shown to be 2nd order accurate for a variety of flows involving moving boundaries [18, 20]. At each time step the background grid nodes are first classified into fluid, solid, and IB nodes using an efficient ray-tracing algorithm [59].

Note that the method presented here does not require dealing with a non-divergence free deformation velocity field such as that encountered in the formulation of Gazzola et al. [10]. This is because the sharp-interface CURVIB requires only the fluid velocities at the surface of the deforming body and not a deformation velocity field extended inside and outside of the immersed body as required by the Gazzola et al. [10] formulation.

2.4 Overset grids

In an overset grid formulation, several grids (denoted as sub-grids hereafter) are arbitrarily overlapped to discretize a complex, multi-connected flow domain. Fig. 3 shows the schematic of an arbitrary overset grid with three sub-grids Ψ_1 to Ψ_3 . Sub-grids Ψ_1 and Ψ_3 only overlap with each other, whereas sub-grid Ψ_2 is completely enclosed by sub-grid Ψ_1 . To solve the governing equations on each sub-grid, boundary conditions should be provided at the interfaces, e.g., with reference to Fig. 3 the boundary conditions for boundary Γ_1^3 of sub-grid Ψ_1 are obtained from sub-grid Ψ_3 , whereas those on boundary Γ_3^1 of sub-grid Ψ_3 are obtained from sub-grid Ψ_1 . If one sub-grid is enclosed by another sub-grid, [such as sub-grid Ψ_2 , which is fully contained within sub-grid Ψ_1 as shown in Fig. 3] we blank some nodes of the outer sub-grid in the overlapping region to transfer the information from the inner sub-grid to the outer sub-grid. The blanking region is shown as the white area in Fig. 3. We reconstruct the fluxes for a buffer layer, shown as the area between the two dashed lines in the white area of Fig. 3, by interpolation similar to the interface fluxes. The buffer layer is incorporated to maintain similar discretization stencil on the grid node in the immediate vicinity of the blanked region. In what follows we discuss the search algorithm for identifying the buffer region, donor nodes and interpolation coefficients, and parallel reconstruction of the fluxes at the boundaries.

2.4.1 Search algorithm and interpolation coefficients for overset grids—As

discussed in the previous section if one sub-grid is completely enclosed by another sub-grid, a buffer layer and a blanking region is required. To identify these regions and reconstruct the boundary conditions at the interfaces, we extend the ideas we developed in the context of the CURVIB method to overset grids. For example, the blanked nodes are identified using a ray-tracing algorithm similar to the one used for identifying grid nodes located within an immersed boundary in the CURVIB method [59]. The ray-tracing algorithm has been used in identifying the blanking regions and hole cutting for overset grids in several algorithms for generating overset grids [63, 64, 65]. In the ray-tracing algorithm a half-ray is shot from the given point. If this ray intersects the closed surface mesh odd number of times, then the point under consideration is located inside the closed surface while if it intersects the closed surface an even number of times it is located outside. The closed surface mesh is unstructured and triangular similar to the mesh used to discretize immersed boundaries are in the CURVIB method [59]. In other words, the interface between the enclosed sub-grid and the surrounding sub-grid is treated as a virtual immersed boundary. The unstructured mesh is constructed either using a preprocessing mesh generation software or by triangulating the inner sub-grid on the surface of a box $[i_s, i_e] \times [j_s, j_e] \times [k_s, k_e]$, where $i_s, i_e, j_s, j_e, k_s, k_e$ are start and end nodes in the i-, j-, and k-direction of the inner sub-grid, respectively. This idea, i.e. treating the interface between two sub-grids as an immersed boundary, enables us to have blanking regions with arbitrarily complex shapes.

After the interface and buffer nodes are identified, their donor cells in other sub-grids, from which the velocities are interpolated, and the corresponding interpolation coefficients should be identified by another search algorithm. The central module of this search algorithm is the *point-in-the-box* test. For this test the points are the cell corners of each cell surface at the interface or in the buffer layer of the recipient sub-grid while the boxes are the grid cells from the cell centers of the donor sub-grid (Fig. 4). A point is inside the box if the vector

formed from the center of each six sides of the box to the point is in the same direction as that side's inward unit normal vector (see Fig. 4), i.e., if for all box surfaces $\kappa=1$ to 6:

$$d^\kappa = (p - p_{mid}^\kappa) \cdot \frac{r_1^\kappa \times r_2^\kappa}{r_1^\kappa \cdot r_2^\kappa} > 0 \quad (21)$$

where p_{mid}^κ is the surface center of the κ th side of the box and r_1^κ and r_2^κ are vectors formed by opposite surface corners of this surface such that their cross product is in the direction of the inward normal. d^κ provides the distance between the κ th side of the box and the point. If the sides are numbered $\kappa=1$ to 6 for $i, i+1, j, j+1, k$, and $k+1$ sides, respectively, as in Fig. 4, then the trilinear interpolation coefficients are obtained from the distances to the sides as follows:

$$a_1 = \frac{d^1}{d^1 + d^2} \quad (22)$$

$$a_2 = \frac{d^3}{d^3 + d^4} \quad (23)$$

$$a_3 = \frac{d^5}{d^5 + d^6} \quad (24)$$

Note that the trilinear interpolation preserves the second order accuracy only if the overlap width remains the same as the mesh is refined [66]. To preserve the second order accuracy quadratic interpolation is needed [66].

To find the donor cell for a given point, we can loop over all grid cells of the donor domain and perform the point-in-the-box test until the cell is found. Such brute force approach is expensive and time consuming. To reduce the cost and accelerate the search, we use the CURVIB bounding box and control cell strategies [59] according to the subsequently outlined Algorithm 1. In the control cell strategy, the donor domain is divided into several Cartesian boxes i.e. control cells. First, we find the control cell that a recipient node p belongs to. Then we perform the point-in-the-box test only for the donor grid cells that are inside the control cell. The bounding box checks before performing the point-in-the-box test if the grid cell is outside a bounding box of the recipient domain since if a grid cell is outside the bounding box, then it cannot be the donor cell. By leveraging these two strategies we can drastically reduce the number of point-in-the-box tests and the computational cost of this search. Algorithm 1 demonstrates the enhanced point-in-the-box search with the control cell and bounding box strategies.

Algorithm 1. Enhanced point-in-the-box test

Locate the control cell (i_x, i_y, i_z) where point p is located with:

$$i_x = \text{INT}((x_p - x_{min})/dx) + 1$$

$$i_y = \text{INT}((y_p - y_{min})/dy) + 1$$

$$i_z = \text{INT}((z_p - z_{min})/dz) + 1$$

```

for
   $i = I_{min}(i_x, i_y, i_z)$  to  $I_{max}(i_x, i_y, i_z)$  do

  for
     $j = J_{min}(i_x, i_y, i_z)$  to  $J_{max}(i_x, i_y, i_z)$  do

    for
       $k = K_{min}(i_x, i_y, i_z)$  to  $K_{max}(i_x, i_y, i_z)$  do

      if cell ( $i, j, k$ ) is in the bounding box then

        count = 0

        for
           $\kappa = 0$  to 6 do

            if
               $p$  is inside the side  $\kappa$  according to eqn. 21 then

                count = count + 1

              else

                Exit the  $\kappa$  loop

              end if

            end for

            if
              count is 6 then

                 $p$  is in the cell  $i, j, k$ 

                Exit all the loops and return ( $i, j, k$ ) and  $d^k$ 

              end if

            end if

          end for

        end for

      end for

```

Calculate the interpolation coefficient based on d^k according to Eqn. 24

2.4.2 Parallel reconstruction of fluxes at the interface and buffer layer—As

discussed in section 2.3 the only boundary conditions required by the CURVIB at the interfaces are the fluxes due to the fully staggered discretization of the method. In the

previous section, we described the technique to identify the donor cells and the interpolation coefficients for all the interface and buffer layer nodes (called recipient nodes, hereafter). Consequently, interpolating the velocities from the donor cells to recipient nodes would be a relatively straightforward task if all the computations were performed on a single CPU. However, all the sub-grids are divided into N_{proc} processes in a general parallel distribution of the overset grids, i.e., the donor cell information can be on a different distributed memory than the recipient node. Therefore, we need to communicate the information between the processes to transfer the information of the donor cell to the recipient node.

The simplest solution is to broadcast the information of all the cells to all the processes, i.e., the information of all the overset grid nodes is available on all the processes. In such a case, the interpolation can be easily performed similar to the single memory/process system. However, the communication cost of this method is quite high and reduces the scalability of the code as the number of processes is increased. Another approach is to only broadcast the information of the donor cells and not the entire sub-grid. Therefore, we set up a communication scatter such that it gathers the donor cell information and broadcasts it. By this method the donor cell information is available for all recipient cells to reconstruct the fluxes. Despite this communication strategy being more efficient relative to the full broadcast, it still increases the communication cost of the overset grid relative to a single grid. However, using overset grids drastically reduces the number of grid points required to achieve the same level of resolution relative to a single grid, thus, reducing the overall computational cost of the simulation. As will be shown in section 3.4 the reduction in computational cost due to the lower number of grid points overcomes the higher communication costs associated with the transfer of information between the processes in the overset grid.

To reconstruct the flux at recipient cells, we reconstruct the velocity at the cell corners of an interface point p of a sub-grid by trilinear interpolation from the Cartesian velocities at the cell centers of the donor cell (Fig. 4). Then we calculate the cell surface center velocity of the recipient cells by averaging the interpolated velocities of the cell corners. Finally, from the cell surface center velocity, the flux U^r for each recipient cell surface is obtained using Eqn. 14. Algorithm 2 summarizes the parallel reconstruction of the fluxes at the interfaces.

2.4.3 Overset-CURVIB solver—There are different methods to solve the governing equations in overset grids. For example, the easiest is the additive Schwartz algorithm that solves each domain sequentially. This algorithm, however, is not that efficient for parallel programming. Therefore, we use a parallel method such that all the domains are solved together during the momentum step of the fractional step method (Eqn. 15). In Ge and Sotiropoulos [60] the momentum Eqn. 15 were solved using a Jacobian-free, Newton–Krylov solver that solves the non-linear function

$$F(U^{(*)}) = \frac{3U^{(*)} - 4U^{(n)} + U^{(n-1)}}{2\Delta t} - RSH(U^{(*)}, u^{(*)}, p^{(n)}) = 0 \quad (26)$$

Algorithm 2. Parallel interpolation

Read in the donor and interpolation coefficient information

Gather the donor cell information from different processes

Broadcast the donor information to all processes

for all recipient nodes **do**

interpolate the corner nodes using trilinear interpolation:

$$\begin{aligned}
 u_p = & a_1 a_2 a_3 u_{i,j,k} + (a_1 - 1) a_2 a_3 u_{i+1,j,k} + \\
 & a_1 (a_2 - 1) a_3 u_{i,j+1,k} + (a_1 - 1) (a_2 - 1) a_3 u_{i+1,j+1,k} + \\
 & a_1 a_2 (a_3 - 1) u_{i,j,k+1} + (a_1 - 1) a_2 (a_3 - 1) u_{i+1,j,k+1} + \\
 & a_1 (a_2 - 1) (a_3 - 1) u_{i,j+1,k+1} + (a_1 - 1) (a_2 - 1) (a_3 - 1) u_{i+1,j+1,k+1}
 \end{aligned} \quad (25)$$

end for

for all recipient cell surface centers **do**

Average the recipient cell corner velocities

Calculate the flux from cell surface center velocity

end for

or a single grid. In an overset grid system we have multiple non-linear functions for each sub-grid. We unite the non-linear functions of each sub-grid into a single composite non-linear function using Data Management Multi Grid (DMMG) tool in PETSc [67] in the overset solver. This composite function is solved using a Matrix-free, Newton–Krylov DMMG solver from PETSc libraries [68]. The boundary conditions at the interfaces and the buffer layer are reconstructed using the most recent solution of the donor domain (in the manner explained in section 2.4) before each evaluation of function ($F(\mathbf{U}^{(*)})$) of the Jacobian-free, Newton–Krylov solver.

After the momentum step the intermediate fluxes and velocities $\mathbf{U}^{(*)}$ are available on each sub-grid. Since the intermediate velocities do not satisfy continuity, the global conservation of mass is not satisfied at the overset grid interfaces. Therefore, we correct the fluxes at the interfaces to satisfy the global conservation of mass as will be explained in section 2.4.4. After this correction we can solve the Poisson Eqn. 16 on each sub-grid separately. As explained in section 2.3, we implicitly have assumed that the fluxes at the boundaries at $(n+1)$ time level are equal to their respective intermediate fluxes, i.e. $\mathbf{U}^{(n+1)} = \mathbf{U}^{(*)}$, which is a $\mathcal{O}(\Delta t^2)$ approximation. This is essential to have consistent boundary conditions for the pressure-correction Eqn 16 for each sub-grid similar to a single grid [61, 57]. The use of pressure value interpolated from other sub-grids as boundary condition for Eqn 16 is inconsistent with Eqn. 19, i.e., produces inconsistent solutions [61, 57].

2.4.4 Conservation laws and solution consistency—In order to satisfy the continuity and drive the divergence of velocity to machine zero, it is essential to satisfy the following conditions:

1. Geometric conservation (free stream preservation)
2. Global mass conservation

The geometric conservation law requires that the grid cells be closed:

$$\oint_S \mathbf{n} dS = 0 \quad (27)$$

where $\oint_S$ is the integral over the surface of the grid cell, dS is the cell surface and $\mathbf{n}$ is the normal to the surface. Furthermore, the rigid body motion of the grid, attached to the non-inertial reference frame requires that:

$$\oint_S \mathbf{n} \cdot \mathbf{v} dS = 0 \quad (28)$$

where $\mathbf{v}$ (see Eqn. 3) is the grid velocity due to the non-inertial reference frame motion. Using Eqns. 3 and 27 Eqn. 28 can be replaced by:

$$\oint_S \mathbf{x} \times \mathbf{n} dS = 0 \quad (29)$$

where, $\mathbf{x}$ is the position vector from the center of the non-inertial frame. The discrete equivalent of the above geometric conservation equations 27 and 29 read as follows:

$$\mathfrak{D}_{i,j,k}^q (\xi_r^q / J) = 0 \quad (30)$$

$$\mathfrak{D}_{i,j,k}^q (\mathbf{M}_r^q) = 0 \quad (31)$$

where, $\mathbf{M}^q$ is the moment of area vector in the ξ^q direction and $\mathfrak{D}_{i,j,k}^q$ is the discrete divergence operator defined as:

$$\begin{aligned} \mathfrak{D}_{i,j,k}^q (U^q) = & J \frac{1}{\Delta \xi^1} \left[(U^1)_{i+1/2,j,k} - (U^1)_{i-1/2,j,k} \right] + \\ & J \frac{1}{\Delta \xi^2} \left[(U^2)_{i,j+1/2,k} - (U^2)_{i,j-1/2,k} \right] + \\ & J \frac{1}{\Delta \xi^3} \left[(U^3)_{i,j,k+1/2} - (U^3)_{i,j,k-1/2} \right] \end{aligned} \quad (32)$$

To satisfy the above equations we calculate ξ_r^q / J at each grid surface as discussed in [33] such that it is exactly equal to the surface area of the grid cell defined by four corners, e.g., with reference to Fig. 5 ξ_r^1 / J . If we calculate the $\mathbf{M}^q$ at each grid surface by calculating the cross product of the center of each surface with its surface area, i.e., $\mathbf{M}^q = \mathbf{x} \times \xi^q / J$, then Eqn. 31 is satisfied to machine zero for Cartesian grids but not for arbitrary curvilinear grids. Therefore, we calculate $\mathbf{M}^q$ in a conservative manner first by dividing surface of the cell into two triangles and calculating the moment of area for each triangle (see Fig. 5 for specific index definitions):

$$\mathbf{M}_{123} = \mathbf{x}_{123} \times \mathbf{S}_{123} \quad (33)$$

where, $\mathbf{x}_{123} = (\mathbf{x}_1 + \mathbf{x}_2 + \mathbf{x}_3)/3$ is the center of the triangle with vertices $\mathbf{x}_1$, $\mathbf{x}_2$, and $\mathbf{x}_3$ and the area of the triangle $\mathbf{S}_{123}$ given by:

$$\mathbf{S}_{123} = \frac{1}{2} (\mathbf{x}_2 - \mathbf{x}_1) \times (\mathbf{x}_3 - \mathbf{x}_1) \quad (34)$$

Finally, we add the moment of area of the two triangles to get the surface moment of area. For example, in the ξ^1 direction we have:

$$\mathbf{M}^1 = \mathbf{M}_{123} + \mathbf{M}_{134} \quad (35)$$

The surface and moment of area vectors can be calculated once at the start of the computations and stored in the memory. We use the stored moment of area values, to decrease the computational cost by calculating the grid velocity as follows instead of directly using Eqn. 13:

$$V^r = u_{ctr} \cdot \xi^r / J + \Omega \cdot M^r \quad (36)$$

Apart from the geometric conservation considerations the global conservation of mass should be satisfied as well. If the global mass conservation is not satisfied the convergence of the numerical method and especially the Poisson equation is hampered due to the global mass imbalance. In case of mass imbalance the Poisson equation converges to a constant (not machine zero) determined by the value the global mass imbalance [62]. Therefore, significant work has been dedicated to developing conservative interpolation schemes [54, 41, 53, 42]. However, such interpolation schemes are not suitable to use with fractional step methods for incompressible flows since the intermediate velocity does not conserve mass. Therefore, the mass flux at the interfaces should be corrected to satisfy the global mass conservation.

To clarify, the boundary condition Eqn. 20 for the Poisson Eqn. 16 should satisfy the following compatibility condition obtained by integrating Eqn. 16 over the domain (Ω volume) and using divergence theorem to turn the volume integral into a surface integral (Γ surface):

$$\int \frac{\partial}{\partial \xi^r} \left(\frac{\xi_q^r}{J} G^q(\phi) \right) d\Omega = \int \frac{\partial \phi}{\partial n} d\Gamma \quad (37)$$

The above condition is satisfied automatically on staggered grids. On non-staggered grids the possible inconsistency from Eqn. 37 can be circumvented by subtracting the right-hand-side of the Poisson equation by the same fixed amount required to satisfy the compatibility

constraint Eqn. 37 [69, 70]. Substituting $\frac{\partial \phi}{\partial n} = 0$ Eqn. 20 into the above condition:

$$\int \frac{3}{2\Delta t} \left(U^{r(*)} \right) n_r d\Gamma = \int \frac{3}{2\Delta t} \frac{\partial}{\partial \xi^r} \left(U^{r(*)} \right) d\Omega = \int \frac{\partial}{\partial \xi^r} \left(\frac{\xi_q^r}{J} G^q(\phi) \right) d\Omega = \int \frac{\partial \phi}{\partial n} d\Gamma = 0 \quad (38)$$

The compatibility Eqn. 38 indicates that the intermediate fluxes $U^{r(*)}$ need to satisfy the global conservation of mass to have a well-posed Poisson Eqn. 16. Therefore, the intermediate fluxes obtained from interpolation at interfaces need to be corrected to satisfy the global conservation of mass.

There are different methods to correct interpolated fluxes to satisfy the global conservation of mass. Zang and Street [57] have suggested a global mass imbalance correction proportional to the flux to satisfy the global mass conservation for their fractional step method. Burton and Eaton [58] also use an explicit correction but proportional to the local area for their fractional step method. Such local flux correction has been shown to be second order accurate by Zang and Street [57].

We have implemented several correction methods to examine the effect of such correction on the accuracy of the computed solution. The magnitude of the local flux correction can be proportional to a quantity Q , which can be the:

1. $\text{area } A = \sqrt{\left(\xi_1^r / J \right)^2 + \left(\xi_2^r / J \right)^2 + \left(\xi_3^r / J \right)^2}$
2. flux U^r
3. normal velocity $u_n = U^r / A$

The local flux U^r can be corrected to provide the global mass conserving local flux $\bar{U}^r$ according to:

$$\bar{U}^r = U^r - \frac{\epsilon_v |Q|}{Q_{sum}} \frac{\mathbf{n} \cdot \xi_r}{|\mathbf{n} \cdot \xi_r|} \quad (39)$$

where ϵ_v is the global mass imbalance defined as

$$\epsilon_v = \int (U^{r(*)}) n_r d\Gamma \quad (40)$$

and Q_{sum} is sum of the absolute value of quantity Q at all grid interfaces. $\mathbf{n}$ is the outward unit normal vector to the overset grid boundary, which can either be in or opposite to the

coordinate ξ^r direction. In fact, the term $\frac{\mathbf{n} \cdot \xi_r}{|\mathbf{n} \cdot \xi_r|} = \pm 1$ determines the sign (addition vs. subtraction) of the correction from the flux $\bar{U}^r$. We have systematically tested the different correction methods based on different definitions of Q and found that the solution did not depend on the correction method as long as the boundary conditions were corrected to satisfy the mass conservation and the Poisson equation for the pressure could converge to machine zero.

3 Numerical results

In this section we apply the numerical method to simulate several different flows including flow past a rotationally oscillating cylinder, through a 90 degree bend, past a self-propelled swimmer and a school of fish, and in an anatomic left-heart system. These cases are selected here to: (1) validate and verify our solver; (2) illustrate the capability of the solver in handling complicated flow problems; and (3) investigate the consistency, stability, and robustness of the solver. In what follows we discuss each test case and summarize our findings at the end.

3.1 Rotationally oscillating cylinder

This test case is chosen to verify our solver in the non-inertial frame. We simulate a rotationally oscillating cylinder in an initially stagnant fluid. We perform simulations in both inertial and non-inertial reference frame and compare the results with numerical results of [30]. All the simulation parameters, domain size, and boundary conditions are chosen similar to [30]. A rotational motion of the cylinder is prescribed by a harmonic oscillation:

$$\omega_c(t) = A_m \sin(2\pi f t) \quad (41)$$

where ω_c is the angular velocity of the cylinder, and A_m and f are the amplitude and frequency of the oscillation, respectively. We can define the Reynolds number as $Re = U_m D / \nu$, where $U_m = A_m D / 2$, D is the diameter of the cylinder, and ν is the kinematic viscosity. For this test case the $Re = 300$ and $f = 0.1$ as in [30]. To simulate this test case in the inertial frame, the grid is fixed and the cylinder rotates with respect to the grid. However, in the non-inertial frame simulation the grid is fixed to the center of the cylinder and rotates with its motion. The size of the computational domain is chosen $-50D < x_r < 50D$ and $-50D < y_r < 50D$ [30]. Dirichlet boundary conditions are used for all outer boundaries as follows for the inertial reference frame:

$$u_{abs} = 0; \quad (42)$$

and for the non-inertial reference frame according to Eqn. 3:

$$\underline{u} = \underline{Q} \underline{u}_{abs} = 0; \quad (43)$$

Fig. 6 shows the time histories of the torque coefficient defined as $C_T = T / (0.5 \rho U_m^2 D^2 / 2)$, where T is the torque and ρ is the fluid density, obtained in different reference frames. It is evident that the time histories in the non-inertial and inertial reference frames are practically identical. Our results are also in agreement but slightly lower than the results of [30] who reported C_T to oscillate between -0.34 to 0.34 . This small discrepancy can be related to the different numerical discretization and immersed boundary reconstruction between present study and the Kim and Choi [30].

3.2 Cavity flow

To validate and verify the overset grid aspects of our method and its order of accuracy, we simulate the lid-driven cavity flow on a single grid and several overset grid with different mesh sizes. The overset grid layout of the square cavity is shown in Fig. 7, which consists of two sub-grids: a $L \times L$ outer grid and a $0.5L \times 0.5L$ inner grid. The inner grid is rotated 45° relative to the center of the cavity. A blanking region of $0.25L \times 0.25L$ is placed at the center of the outer grid for the information from the inner grid to be transferred to the outer grid. The top lid is moving with constant speed U in horizontal direction and the Reynolds number (Re) based on the lid velocity, the length of the square cavity L , and the fluid kinematic viscosity is 100. Three overset grids with grid spacings h of $1/64$, $1/128$, and $1/256$ are reconstructed and compared against the benchmark solution of a single grid with grid spacing $1/512$ to determine the accuracy of the overset method.

The contours of the velocity components on the overset grids are compared against the single grid solution in Fig. 8. It can be observed that the velocity values of the two sub-grids in the overlap region are consistent and match each other in all cases. Furthermore, The velocity components of the overset grids follow closely the single-grid solution. In fact, in the coarsest mesh with spacing $h=1/64$ there is slight difference between the contour lines (Fig. 8a), which is reduced as the mesh is refined (Fig. 8b,c). To quantitatively demonstrate the order of accuracy of the method, the solution of the single grid is considered as benchmark solution and the error in the solution of the overset grids is calculated relative to this solution:

$$\text{Error} = \sum_{b=1}^2 \frac{1}{N^2} \sum_{i=1}^N \sum_{j=1}^N \left((u_{i,j}^b - u_{i,j}^{sg})^2 + (v_{i,j}^b - v_{i,j}^{sg})^2 \right) \quad (44)$$

where N is the number of grid points in the overset grids in each i and j direction; $u_{i,j}^b$ and $v_{i,j}^b$ are the overset grid solutions on the (i, j) grid node for the sub-grid b of the overset grid; and $u_{i,j}^{sg}$ and $v_{i,j}^{sg}$ are the benchmark single grid solutions interpolated onto the (i, j) grid node of the overset sub-grid b . It is expected that the overset grid solution approaches the benchmark solution as the grid is refined, i.e., the error to be reduced quadratically with grid refinement. Fig. 9 plots the error against the grid spacing in log-log scale and demonstrates that the error reduces with higher than second order accuracy with grid refinement.

3.3 Steady and Pulsatile flow through a 90° bend

We simulate the 3D steady and pulsatile flow through a strongly curved 90° pipe bend to demonstrate the accuracy of the overset method for a 3D steady and pulsatile flow. The test case with steady inflow was studied experimentally by Bovendeerd et al. [71] while the pulsatile case was studied by Rindt et al. [72]. The geometry of this test case is shown in Fig. 10. The radius of curvature of the bend is $3D$, where D is the diameter of the pipe. The

inlet is placed $5D$ before the bend while the outlet is placed $7D$ after the bend. To test the overset method we use three overlapping grids as shown in Fig. 10 with $41 \times 41 \times 21$, $41 \times 41 \times 64$, and $41 \times 41 \times 33$ nodes for first, second, and third grid blocks (called grid CH1 hereafter), respectively. To test the performance of the overset method with overlapping grids with different resolutions, we test two other cases for the pulsatile simulations. One case with a finer mesh (called grid CH2) with $81 \times 81 \times 64$ nodes for the middle (bend) grid block while the resolution of the other grid blocks remained the same. Another case (called grid CH3) with the resolution of the middle block unchanged and two coarser mesh $21 \times 21 \times 21$ and $21 \times 21 \times 33$ for the first and last grid blocks, respectively.

The Reynold number (based on the bulk velocity U and diameter D of the pipe) for the steady experiment was $Re = 700$ [71]. In the pulsatile experiments [72] a gear pump providing a steady flow of $Re = 500$ in conjunction with a piston pump generating a sinusoidal flow waveform with Reynolds number ranging from -300 to 300 were used to generate the pulsatile flow waveform. The resulting Womersley number of the experimental flow is 7.8 . The time scale based on the length (D) and velocity (median U) is non-dimensional time period of inflow oscillations, which is equal to $T=12.3$. We use a non-dimensional time step of $\Delta t=0.02$ for the steady and 0.0123 for the pulsatile simulations.

We use Neumann boundary conditions at the outlet while a fully developed (parabolic) velocity profile at the inlet of the steady case and a profile from the Womersley solution of a fully developed pulsatile flow in a circular pipe at the inlet of the pulsatile case as follows [60]:

$$u_{\text{inlet}} = 2 \left[1 - \left(\frac{r}{R} \right)^2 \right] - i \frac{K}{\omega} e^{i\omega t} \left[1 - \frac{J_0 \left(r \sqrt{\frac{-i\omega}{\nu}} \right)}{J_0 \left(R \sqrt{\frac{-i\omega}{\nu}} \right)} \right] \quad (45)$$

where J_0 is the Bessel function of the first kind and order zero, r is the radial distance from the center of the pipe, R is the radius of the pipe, ω represents the angular frequency of the flow oscillation, which is 13.31 rad/s, and ν is the fluid viscosity. As in [60] to generate the sinusoidal flow waveform that varies in accordance with the experiments from $Re = 200$ to $Re = 800$, the constant K is selected to be 0.375 . The above equation is solved using MATLAB and the resulting solutions are stored and fed into the flow solver to specify the time-varying inlet flow. The so computed inflow waveform is in good overall agreement with the experimental inflow condition as shown in Fig. 11.

The simulated flow in the bend with the steady inlet is compared against the experimental measurements [71] in Fig. 12. The streamwise velocity profiles at the plane of symmetry for both numerical and experimental results are plotted at different angles ($\theta = 0^\circ, 23.4^\circ, 58.5^\circ$ and 90°) in the bend in Fig. 12. As seen, the calculated velocity profiles are in excellent agreement with the measurements.

The simulated flow in the bend with the pulsatile inlet is compared against the experimental results [72] and single grid computations of Ge and Sotiropoulos [60] in Fig. 13. The streamwise velocity profiles at the plane of symmetry are plotted at five different location i.e. $\theta = 0^\circ, 22.5^\circ, 55^\circ, 67.5^\circ$, and 90° at four different time instants during the cycle ($0, 0.25T, 0.5T$ and $0.75T$). The solutions on different overset grids are in excellent agreement with the computations on the single grid of Ge and Sotiropoulos [60]. The computations are in good overall agreement with the experimental results of Rindt et al [72] with the maximum discrepancy at $t=0.75T$. As discussed in [60] this discrepancy can largely be attributed to the inlet boundary conditions, which exhibits the largest deviation from the experimental wave form near $t=0.75T$ as seen in Fig. 11. The results on different overset

grids (CH1, CH2, CH3) with different overlapping resolutions are in excellent agreement with each other and the results of Ge and Sotiropoulos [60] on a single grid, which demonstrates the consistency and accuracy of our overset method.

3.4 Aquatic swimming

In this section we simulate two cases to demonstrate our capabilities: A self-propelled mackerel and multiple tethered mackerels swimming in the diamond arrangement. The self-propelled mackerel case is to demonstrate the important roles of both the reference frame and the overset grid: 1) Choosing the frame of reference to be attached to the mackerel's center of mass reduces the displacement region of the mackerel, i.e., the required domain size, considerably. If the simulations were performed in the inertial frame a long high-resolution grid would be required in the path of the swimmer. 2) The overset grid can help us to increase the resolution of the grid locally near the immersed boundaries efficiently without considerably increasing the total number of grid points. The four mackerels swimming case is to demonstrate how the overset grid reduces the required total number of grid points without decreasing the resolution near the swimmers. To the best of the authors' knowledge, this is the first 3D simulation of multiple swimmers. Previous simulations of multiple swimmers were only carried out in 2D, e.g., see [10]. The 3D simulations of multiple swimmers were not possible without the overset grids due to the high number of grid points required (high computational cost) if a single grid was used.

3.4.1 A self-propelled mackerel—The CURVIB method in the non-inertial moving frame was successfully applied to simulate steady straight-line self-propelled aquatic swimming over a single grid [32, 29]. The reference frame was attached to the center of mass of the fish. The body motion of fish relative to the center of mass was prescribed and the fluid equations in the non-inertial frame Eqn. 8 were solved to find the flow field around the fish. The velocity of the frame (center of mass) was calculated based on the fluid forces on the body of the fish using a strongly-coupled fluid-structure interaction algorithm as in [59]. Here we carry out similar simulations in the non-inertial frame but on an overset grid instead of a single grid. The results of the overset and single grid simulations are compared against each other to: 1) verify the results of the overset grid; and 2) show the benefits of the overset grid in terms of computational cost.

The computational domain is a cuboid with dimensions $2L \times L \times 7L$, where L is the fish length, similar to our previous simulations [29, 20]. The domain width $2L$ and height L are more than ten times the mackerel width $0.2L$ and height $0.1L$, respectively. As in our previous simulations, the fish is placed $1.5L$ from the inlet plane in the axial direction and centered in the transverse and the vertical directions. The computational domain is discretized with two different grids: An overset grid (Fig. 14) and a single grid with the same resolution near the fish. The results of the overset grid are verified against the single grid results.

The overset grid consists of two sub-grids. The outer sub-grid discretizes the complete domain while the inner sub-grid with dimensions $1L \times 0.5L \times 1.5L$ provides higher resolution near the fish body. The inner sub-grid is discretized with a uniform mesh with spacing $h = 0.005L$ in all directions with $201 \times 101 \times 301 \approx 6$ million grid nodes. The outer sub-grid is discretized with 5.5 million grid nodes. In the outer sub-grid, a uniform mesh with constant spacing $h = 0.008L$ is used in the smaller cuboid enclosing the fish and the mesh is stretched from the faces of this smaller cuboid to the boundaries of the computational domain using a hyperbolic tangent stretching function. Inside the outer sub-grid a region with dimensions $0.29L \times 0.27L \times 1.14L$ is blanked, whose solution is interpolated from the inner sub-grid. The total number grid points in both sub-grids is 11.5

million. A single grid of about 20 million nodes provides the same grid spacing of $h = 0.005L$ in the cuboid enclosing the fish. The single grid mesh is stretched using a hyperbolic tangent stretching function to the boundaries of the domain.

The body of the mackerel used in this study are exactly the same as those used in our previous simulations, which was modeled after the actual anatomy of a mackerel [29, 20]. Only the caudal fin was retained for the mackerel due to lack of detailed experimental data for other fins [29, 20]. The mackerel body is meshed with triangular elements as needed by the immersed boundary search algorithm (Fig. 14). The kinematics for steady swimming is generally in the form of a backward traveling wave with the largest wave amplitude at the fish tail, which can be described by the following equations for the lateral undulations of the fish body (all lengths are non-dimensionalized with the fish length L):

$$h(z, t) = a(z) \sin(2\pi z/\lambda - 2\pi ft) \quad (46)$$

In the above equation: z is the axial (swimming) direction measured along the fish axis from the tip of the fish head; $h(z, t)$ is the lateral excursion of the body at time t ; $a(z)$ is the amplitude envelope of lateral motion as a function of z ; λ is the wavelength and f is the frequency of the backward traveling wave. For a typical mackerel the amplitude envelop can be approximated by a quadratic curve of the form [20]:

$$a(z) = a_0 + a_1 z + a_2 z^2 \quad (47)$$

with coefficients $a_0 = 0.02$, $a_1 = -0.08$, and $a_2 = 0.16$ to match the experimental curve of Videler and Hess [73] obtained for a mackerel. The maximum displacement occurs at the tail $a_{max} = 0.1$, i.e. $h_{max} = 0.1L$. The non-dimensional wavelength is $\lambda/L = 0.95$ for a typical mackerel [20, 73]. We divide the tail beat period into 240 time steps, which corresponds to a non-dimensional time step of $\Delta t = 0.00139$. As described in [29] in detail, the non-dimensional frequency (Strouhal number) fL/U is chosen based on tethered simulations [20]. From our previous tethered simulations [20] we know that at Strouhal number about 0.6 the swimmer can produce enough thrust to cancel drag (i.e., zero average net force) at tether velocity U_o corresponding to Reynolds number $Re = 4000$. Therefore, in a self-propelled simulation if we choose the non-dimensional frequency to be 0.6 the final average velocity during steady state should be close to the tether velocity U_o resulting in a Reynolds number close to 4000.

The virtual swimmer starts to undulate in an initially stagnant fluid and the boundary conditions on the domain outer boundaries are far-field (Neumann) boundary conditions. We prescribe the above mackerel kinematics and based on the fluid forces generated by the fish motion the velocity and the position of the fish center of mass is obtained through the our strongly-coupled fluid-structure interaction algorithm [59]. To stabilize the strong-coupling we use under-relaxation and Aitken acceleration technique [59]. For more information about self-propelled simulations the reader is referred to [29].

The simulations are carried out until the fish reaches the quasi steady-state, i.e., the fish average velocity stays constant. It takes more than 20 tail beat cycles to reach quasi steady-state (Fig. 15). Fig. 15 shows the swimming speed of the fish in the overset and single grid. It can be observed that the results on overset and single grid are almost identical and their difference is less than 0.2% as shown in the inset of the figure. We visualize the flow field in the midplane of the fish using contours of axial and lateral velocity and vorticity in the Fig. 16. The boundary of the inner sub-grid is demonstrated by the thick black rectangle. The outer domain blanked region is demonstrated by the thick red rectangle. The two sub-grids overlap each other in the region between these two rectangles. As can be observed in the

figure the flow quantities on each sub-grid in the overlapping region are quite similar and the contour lines overlap each other and the single grid contour lines, i.e., the solution is consistent over the two sub-grids and is in excellent agreement with the single grid results. Furthermore, it can be observed that the vortices from one sub-grid are advected and pass across the interface to other sub-grid without any visible distortion. Fig. 17 shows the 3D flow field generated by the self-propelled mackerel. It can be observed that the wake bifurcates into two rows of vortices (double row structure) similar to the single grid solution, which is expected for the current Strouhal number of about 0.6 [20].

Finally, it should be noted that the overset grid almost cut the required grid nodes in half while maintaining the same resolution near the fish body. This drastically reduces the computational time of the overset grid relative to the single grid as shown in Fig. 18. In Fig 18 the CPU time computed by taking the average of computation time of the first ten time steps, which is plotted against the number of CPUs for both overset and single grid simulations in log-log scale. The CPU times are obtained on the Nami cluster with 448 computing cores at the University at Buffalo. This cluster contains 28 computing nodes, each node containing 2×8 core (Magny-Cours) AMD 2.0 GHz with 2GB RAM per core and QDR infiniband. Typically, the momentum Eqn. 15 converges within 10 iterations while the Poisson Eqn. 16 converges within 30 iterations. At low number of CPUs the Poisson equation can take as much as twice the CPU time of the momentum equation. However, as the number of CPUs is increased the momentum and Poisson consume similar amount of CPU time. It can be observed that the CPU time of the overset grid is consistently lower than the single grid for all CPU numbers. Even at the highest number of CPUs (highest communication time) the overset grid is about 40% faster (compare CPU time of 11.1 sec for a single grid versus 8.1 sec for the overset grid when running on 400 cores). Also note that for the single grid the minimum number of cores required is 32 (due to memory requirements) while the overset grid can be run on as low as 8 cores due to lower number of grid points. These advantages make the overset grid significantly more attractive than the single grid approach.

3.4.2 Multiple mackerels in diamond arrangement—The 3D simulations of a single swimmer have been performed by several groups recently [29, 19, 20, 10, 74, 75]. However, the 3D simulations of multiple swimmers have not been carried out yet probably due to the high computational cost of such simulations, and most recent simulations has been two dimensional [10]. To demonstrate the power of our hybrid overset-CURVIB method in simulating multiple swimmers in 3D, we simulate four tethered swimmers in diamond arrangement as observed in Fig. 19. The overset grid drastically reduces the total number of grid points required for such simulation while maintaining the grid resolution near each swimmer. The side swimmers are placed $1.1L$ and $1.5L$ laterally and posteriorly, respectively, relative to the front swimmer (L is the fish length). The last swimmer is placed $3L$ behind the front swimmer. The background computational sub-grid is a cuboid with dimensions $2L \times L \times 10L$. The inner sub-grid with dimensions $1L \times 0.5L \times 1.5L$, similar to our inner sub-grid in the previous section for the single swimmer, is placed on each swimmer over the outer sub-grid to provide a higher resolution near the fish bodies (19). Inside the outer sub-grid regions with dimensions $0.29L \times 0.27L \times 1.14L$ are blanked over each inner sub-grid, whose solution is interpolated from the inner sub-grids. The outer sub-grid is discretized with 7 million grid nodes similar to our previous simulations. A uniform mesh with constant spacing $h = 0.02L$ is used to discretize a smaller cuboid enclosing all the fish in the outer sub-grid. The mesh is stretched from the faces of this smaller cuboid to the boundaries of the computational domain using a hyperbolic tangent stretching function. Each inner sub-grid is discretized with a uniform mesh with spacing $h = 0.005L$ in all directions with $201 \times 101 \times 301 \approx 6$ million grid nodes similar to the inner sub-grid in the previous section for the single swimmer. Therefore, the total number of grid points of the

overset grid is about 31 million grid nodes. A single grid of similar resolution would require at least 90 million grid points!

The mackerel geometry and kinematics are exactly the same as what was described in the previous section for a single swimmer. We prescribe the steady swimming kinematics in the form of backward traveling wave Eqn. (46) for all of the swimmers. There is no phase difference between the backward traveling waves of different swimmers. We divide the tail beat period into 240 time steps, which corresponds to a non-dimensional time step of $\Delta t = 0.00139$. We carry out simulations with all of the swimmers hold in place with a virtual tether. We use a uniform flow at the inflow, Neumann boundary condition at the outlet, and slip wall on the side boundaries. We carry out the simulations at Reynolds number of 4000 and Strouhal number of 0.6.

We visualize the flow field in the midplane of the fish using vorticity contours in Fig. 19. The boundary of the inner sub-grid is demonstrated by the thick black rectangle. As can be observed in the figure the flow quantities on each sub-grid in the overlapping region are quite similar and the contour lines overlap each other, i.e., the solution is consistent over the sub-grids. Furthermore, it can be observed that the vortices from one sub-grid are advected and passed correctly to the other sub-grid. Fig. 20 shows the 3D flow field generated by the tethered swimmers. It can be observed that the wake of each swimmer bifurcates into two rows of vortices (double row structure), which is expected for the current Strouhal number of 0.6 [20], and interact with downstream fish wakes. In this test case the swimmers were tethered and did not move relative to each other. If they were to move, they might move out of the inner domain that provides the high-resolution near them. This problem will be tackled in our future work and can be handled by moving the overset grids with the center of mass of the swimmers.

3.5 Flow through a mechanical heart valve driven by the contraction of the left-ventricle

3.5.1 Background—More than 50% of the implanted prosthesis for replacing diseased natural heart valves is of BMHV type mainly due to its very long durability [76]. All current BMHV designs, however, are prone to a number of complications, including among others, increased risk of blood clotting and possible hemorrhage [77]. Such complications are believed to be related to the complex, non-physiologic blood flow patterns induced by the valve. Therefore, it is critical to characterize the flow physics created by these valves.

A typical BMHV consists of two leaflets, which can rotate around a hinge attached to a circular housing. The leaflets rotate around their hinges passively in response to the torque imparted by the periodic pulsatile flow giving rise to a very complex FSI problem. The periodic pulsatile flow is created by the periodic contraction and expansion of left ventricle (LV). For a typical adult patient the Reynolds number of the flow (based on the aorta diameter and the bulk flow at the aortic root) varies from zero, during the diastolic phase when the valve is closed, to nearly 6000, at peak systole when the valve leaflets are fully open. The complex motion of the LV and dynamic interaction and response of the leaflets to the flow generated by such motion in addition to the complex geometry of the LV, anatomic aorta, and BMHV creates a challenging environment for numerical simulations.

We have recently developed a fluid-structure interaction (FSI) algorithm capable of performing high-resolution three-dimensional FSI simulation of BMHV under physiologic conditions (peak $Re = 6000$) [59]. As we have extensively discussed in [59], obtaining converged FSI iterations for the BMHV problem is a particularly challenging task because of the combined effects of the leaflet properties (very low reduced inertia) and local flow effects. The low reduced inertia of the leaflets causes the motion of the leaflets to be dominated by the added-mass effect, which in general has a detrimental effect on the

stability of FSI iterations. We found that the under-relaxed strong-coupling of fluid and structure domains combined with the Aitken acceleration technique could yield converged FSI iterations [59]. The computed results reported in [59] were in excellent agreement with the experiments in a straight axisymmetric aorta [78], both in terms of the instantaneous flow patterns and the calculated leaflet kinematics. We have applied the above method to simulate a BMHV implanted in a realistic anatomical aorta geometry [12] and at different orientations [13].

In all the above simulations and recent simulations of other groups focused on heart valves [79, 80, 14, 81], however, the left-ventricle (LV) is replaced by a straight circular pipe. The leaflets open and close due to the pulsatile inflow waveform prescribed at the inlet of the pipe and not the actual motion of the LV. On the other hand, the simulations focused on left-ventricle have not resolved the heart valves and replaced the valves with simple boundary conditions [82, 83, 84, 85]. However, the valves are shown to affect the vortex dynamics inside the left-ventricle [86]. Furthermore, the asymmetric flow created by the LV can drastically affect the performance of the valves and their hemodynamics, which cannot be captured by the simulations with symmetric pipe inflow. Here we carry out the first simulation of a heart valve driven by the motion of the left-ventricle during systole using our overset-CURVIB solver. Our overset grid consists of two grids: one grid that covers the LV and a body-fitted grid for the anatomic aorta geometry. The LV and BMHV are handled as immersed boundaries. The LV motion is prescribed while the BMHV leaflets motion is calculated based on the FSI algorithm explained in detail in [59]. In this work we have focused on the systolic phase, in which the LV is contracting, the aortic valve opens, and the mitral valve remains closed. It should be noted that the emphasis of our work is not on the physiologic realism of the computed results but rather on demonstrating the capabilities of our numerical approach. For that, a very simple model is used to approximate the LV wall motion during systole as described below.

3.5.2 Computational details and results—The left ventricle and aorta geometries are reconstructed from MRI data from a healthy volunteer at Georgia Institute of Technology with a method similar to [87]. The resulted surface mesh is post-processed using commercial software Mimics 12.0. The left atrium, the carotid and coronary arteries are removed from the original data. The final anatomy includes a left ventricle and the aorta branch as shown in Figure 22.

We assume that the deformation of the aorta is small, i.e., the aorta is assumed rigid, while the left ventricle chamber undergoes substantial deformation. We track the surface of the left ventricular chamber by a set of material points, represented by the triangular surface mesh of the LV geometry (Fig. 22). We prescribe the kinematics of each of these material points by a simple model due to the absence of exact experimental measurements. The model uses the radial and longitudinal locations of the material points to determine the motion of that point. The LV longitudinal axis is defined as the line pointing from the center of the atrium to the apex of the heart. The motion of a material point is constrained into radial motion which is orthogonal to the LV longitudinal axis. We assume that the velocity of a point $\mathbf{u}(t)$ is a function of these three components: i) time (t) ii) radial position ($\mathbf{r}$) iii) longitudinal position ($\mathbf{l}$), whose effects are independent of each other as follows:

$$\mathbf{u}(t) = \Psi(t) \phi(\mathbf{r}) \kappa(\mathbf{l}) \mathbf{e}_r \quad (48)$$

Ψ , ϕ and κ are the temporal, radial and longitudinal variations of the velocity field. With $\mathbf{e}_r$ is the unit vector of the radial direction at the current point. If the spatial distribution is considered to be linearly axis-symmetrical and the temporal variation is approximated by a *sin* wave, it reduces to:

$$\begin{aligned} \mathbf{u}(t) &= \Psi(t) \phi(r) \kappa(l) \mathbf{e}_r \\ \Psi(t) &= \Psi_0 \sin\left(2\pi \frac{t}{T}\right), \quad \Psi_0 = 0.002 \\ \phi(r) &= |r| \\ \kappa(l) &= |l| \end{aligned} \quad (49)$$

The period T is set to be a typical healthy subjects at rest, i.e., $T = 1$ s (60bpm). A few snapshots of the LV shape at several time instants are shown in Fig. 22.

Similar to our previous simulations [12, 59, 13] we use a St. Jude Regent 23mm implanted in the aortic position. The BMHV is meshed with triangular unstructured mesh as required by the CURVIB solver as shown in Fig. 21. As in the straight aorta simulations of [59], we have neglected the hinge mechanism connecting the leaflets and the valve housing. In addition, the damping due to the friction of the hinge was neglected as well since it is very small relative to the flow forces and no experimental data are currently available on the actual value of the friction coefficient. Therefore, the governing equation of the leaflets based on angular momentum, assuming the x axis as the rotation axis of the valve, will be:

$$I_x \frac{\partial^2 \theta}{\partial t^2} = M_x \quad (50)$$

where, θ is the opening angle of the leaflet. I_x is the non-dimensional moment of inertia of the leaflet equal to 0.001 based on leaflet density of 1750 kg/m^3 and blood density of 1030 kg/m^3 similar to our previous work [12, 59, 13]. The peak Reynolds number created by the above LV kinematic based on the maximum bulk velocity at the aortic root and the aortic root diameter is 5800.

The grid comprises of two blocks: the left ventricle and aorta block. The left ventricle block is a rectangular Cartesian mesh of size $101 \times 101 \times 101$ (1 million) while the aorta block is a body-fitted grid of size $181 \times 181 \times 201$ (7 million). The computational grid is shown in Fig. 23 (every 4 grid node is shown for clarity) along with the LV and BMHV, which are placed as immersed boundaries over the background overset grid. The non-dimensional time Δt is 0.007.

The motion of the LV was prescribed as discussed above while the motion of the leaflets were calculated using under-relaxed strong-coupling algorithm combined with the Aiken acceleration technique [59]. This FSI method could reduce the residual (i.e., the difference in the angular velocity and position of each leaflet between two successive iterations) by about 4 orders of magnitude to 10^{-6} within 4 to 5 iterations.

Figure 24 shows the out-of-plane vorticity contour on the plane passing through the symmetry plane of the leaflets. Similar to the work of [78], the shear layers from the leading and trailing edges of the leaflet start to roll up (Figure 24a) and become unstable at peak systole and shed von Karman like structures (Figure 24b). The shear layers in the sinus area also roll up and extracts an opposite sign vorticity at the wall. In contrast to the symmetric characteristic of the shear layer roll-up in simulations and experiments with symmetric inflow [78, 59, 12, 13], the rolling up here is asymmetrical and it immediately splits into several portions prior to interacting with the distal sinus wall. Note that the Von Karman vortex sheet is directed to align with the ascending aortic orientation (point to the left) indicating the dominance of geometrical factors on the flow structure similar to the work of [12]. At late systole, the above-mentioned organized structures burst into turbulent-like small structures (Figure 24c) as observed in the simulations and experiments with symmetric aorta [78, 59, 12].

The three-dimensional flow structure is visualized in Figure 25 using the iso-surfaces of q -criteria. At early systole, the ring-like structure at aortic wall intertwine with the central structure emanating from the leaflets. The merging of the wall ring and the central structure is stronger at non-coronary cusp plane (ZX plane). At peak systole, the vortical structures occupy all the aortic root space and the separation of the shear layer from the sinus wall is visible. The performance of the BMHV strongly depends on the incoming inflow from the left-ventricle. The methods developed here can be used to simulate the BMHV implanted in different orientation in the aortic and mitral positions on a patient-specific basis, thereby enabling surgical planning, i.e., the placement of the heart valve can be optimized before the surgery through virtual surgeries. This type of surgical planning is impossible with previous simulation of the valves with a straight pipe inflow.

4 Summary and Conclusions

We have extended the CURVIB method [60, 59] to develop a new hybrid computational framework, which generalizes the CURVIB to a non-inertial frame of reference and integrates it for the first time with overset grids. The new hybrid computational framework has enabled high-resolution fluid-structure interaction simulations of real-life biological flows, which could not be tackled with our previously developed methods. The overset grid allows us to efficiently discretize complicated multi-connected geometries and/or increase the resolution of the grid locally without incurring a large computational time penalty. The large motions of the boundaries on the overset grids are handled by treating them as sharp-interface immersed boundaries using ideas developed in the context of the CURVIB method to handle solid immersed bodies. The non-inertial frame of reference allows us to choose the most suitable frame to decrease the displacement region of a moving body such as a swimming fish, a falling sphere, or flow in the cardiovascular system when the patient is moving, etc. Major algorithmic advancements were developed and implemented in this work to make this hybrid computational framework possible. These include: 1) extending the conservative formulation of the non-inertial frame of [33, 31, 30] to generalized curvilinear coordinates; 2) developing conservative interpolation algorithm based on explicit mass correction for the overset grids suitable for the fractional step method used in the CURVIB approach; and 3) developing communication strategies for transferring data among subdomains to enable efficient parallel implementation of the overset-CURVIB solver.

A major obstacle in using overset-grids with fractional step method is the issue of mass conservation. Since in fractional step method the intermediate velocities do not conserve mass, any type of interpolation from such velocities will not conserve mass. Therefore, an explicit correction should be added to the interpolated values at the interfaces to satisfy the global conservation of mass. We have tested different types of mass correction proportional to the area, velocity, or flux at each interface cell and did not find any significant difference between them. In fact, as long as the mass was conserved by any type of correction the results were practically identical. This is due to the fact that such corrections do not affect the second order accuracy of the solution [57].

Another challenge in using overset grids is in efficient parallel communication strategies to transfer information between subdomains. We have implemented a communication strategy that broadcasts the velocities of the donor nodes, which are located on a different process than the recipient nodes, in order to carry out the interpolation of the interface and the buffer layer boundary conditions effectively. Nevertheless, no matter how efficient the communication is in overset grids, inherently it still has overhead communication costs relative to a single grid. As shown in this work the additional communication cost of overset grids is offset by the reduction in computational cost due to lower number of grid points. In fact, the use of overset grids reduced the total number grid points from 20 (for a single grid)

to 11 million while preserving the same resolution in the self-propelled fish simulations. In the multiple swimmer simulations, the total number of grid points in the overset grid was around 31 million, whereas at least 90 million grid points would be required for a single grid to maintain the same resolution near each swimmer.

The above advancements have enabled us to tackle two challenging biological problems for the first time. We have simulated multiple 3D swimmers as well as a mechanical heart valve driven by the motion of the left-ventricle for the first time. To the best of the authors' knowledge the 3D simulations of aquatic swimmers have been restricted to a single swimmer and only 2D simulations of multiple swimmers have been performed to date [10]. The previous simulations of mechanical heart valves had ignored the left-ventricle geometry and had replaced it with a straight pipe with a pulsatile inflow [79, 80, 14, 81]. The previous simulations of left-ventricle, on the other hand, had ignored the valves and replaced them with simple inflow/outflow conditions [82, 83, 84, 85].

Finally, it is important to emphasize that simulations such as those reported herein were not possible without the overset grid, general non-inertial reference frame formulation and their integration with the CURVIB method. Our future work will focus on extending our method to moving overset methods, i.e., the sub-grids in the overset grid can move relative to each other to follow an immersed boundary. In the moving sub-grids the equations will be solved in the non-inertial frame, whereas in the fixed background the equations will be solved in the inertial frame. This will enable us to simulate multiple self-propelled swimmers that move relative to each other.

Acknowledgments

This work was supported by NIH Grant RO1-HL-07262, NSF Grant 0625976, Minnesota Supercomputing Institute, and the Center for Computational Research (CCR) of University at Buffalo. We are grateful to Ajit Yoganathan and the members of Georgia Tech's Cardiovascular Fluid Mechanics Laboratory for providing us with the geometry of the mechanical valve and the left ventricle.

A Discretization of convective terms in the non-inertial frame

The discretization of convective term only in the ξ^1 direction is provided. The discretization in the other directions are similar. The convective term $C_1(u_q)$ Eqn. 9 at grid position i, j, k is as follows:

$$C_1(u_q)_{i,j,k} = J \frac{1}{\Delta \xi^1} \left[F_1(u_q)_{i+1/2,j,k} - F_1(u_q)_{i-1/2,j,k} \right] + J \frac{1}{\Delta \xi^2} \left[F_2(u_q)_{i,j+1/2,k} - F_2(u_q)_{i,j-1/2,k} \right] + J \frac{1}{\Delta \xi^3} \left[F_3(u_q)_{i,j,k+1/2} - F_3(u_q)_{i,j,k-1/2} \right] \quad (\text{A.1})$$

for,

$$F_1(u_q)_{i+1/2,j,k} = \left[\left(\frac{U_+^1}{J} - \frac{V_-^1}{J} \right) u_q^- + \left(\frac{U_-^1}{J} - \frac{V_+^1}{J} \right) u_q^+ \right]_{i+1/2,j,k} \quad (\text{A.2})$$

where,

$$U_+ = \frac{1}{2} (U_{i+1/2,j,k} + |U_{i+1/2,j,k}|) \quad (\text{A.3})$$

$$U_- = \frac{1}{2} (U_{i+1/2,j,k} - |U_{i+1/2,j,k}|) \quad (\text{A.4})$$

and $|\cdot|$ denotes the absolute value. $u^{+/-}$ are defined as follows:

$$u_{i+1/2,j,k}^+ = \frac{1}{8} (6u_{i+1,j,k} - u_{i,j,k} + 3u_{i+2,j,k}) \quad (\text{A.5})$$

$$u_{i+1/2,j,k}^- = \frac{1}{8} (6u_{i,j,k} - u_{i-1,j,k} + 3u_{i+1,j,k}) \quad (\text{A.6})$$

References

- [1]. Benek, J.; Steger, J.; Dougherty, F. A flexible grid embedding technique with application to the Euler equations. Computational Fluid Dynamics Conference; Danvers, MA. 1983. p. 373-382.
- [2]. Henshaw W. A fourth-order accurate method for the incompressible Navier-Stokes equations on overlapping grids. Journal of Computational Physics. 1994; 113(1):13-25.
- [3]. Peskin CS. Flow patterns around heart valves: A numerical method. Journal of Computational Physics. 1972; 10:252-271.
- [4]. Mittal R, Iaccarino G. Immersed boundary methods. Annual Review of Fluid Mechanics. 2005; 37:239-261.
- [5]. Sethian J. A fast marching level set method for monotonically advancing fronts. Proceedings of the National Academy of Sciences. 1996; 93(4):1591.
- [6]. Osher S, Fedkiw R. Level set methods: an overview and some recent results. Journal of Computational physics. 2001; 169(2):463-502.
- [7]. Cottet G, Koumoutsakos P. Vortex methods: theory and practice. Cambridge Univ Pr. 2000
- [8]. Leonard A. Vortex methods for flow simulation. Journal of Computational Physics. 1980; 37(3): 289-335.
- [9]. Coquerelle M, Cottet G. A vortex level set method for the two-way coupling of an incompressible fluid with colliding rigid bodies. Journal of Computational Physics. 2008; 227(21):9121-9137.
- [10]. Gazzola M, Vasilyev O, Koumoutsakos P. Shape optimization for drag reduction in linked bodies using evolution strategies. Computers & Structures. 2011; 89(11):1224-1231.
- [11]. Liu Q, Vasilyev O. A brinkman penalization method for compressible flows in complex geometries. Journal of Computational Physics. 2007; 227(2):946-966.
- [12]. Borazjani I, Ge L, Sotiropoulos F. High-Resolution Fluid-Structure Interaction Simulations of Flow Through a Bi-Leaflet Mechanical Heart Valve in an Anatomic Aorta. Annals of Biomedical Engineering. 2010:1-19. [PubMed: 20676773]
- [13]. Borazjani I, Sotiropoulos F. The effect of implantation orientation of a bi-leaflet mechanical heart valve on kinematics and hemodynamics in an anatomic aorta. ASME Journal of Biomechanical Engineering. 2010; 132(11):111005-8.
- [14]. De Tullio M, Cristallo A, Balaras E, Verzicco R. Direct numerical simulation of the pulsatile flow through an aortic bileaflet mechanical heart valve. Journal of Fluid Mechanics. 2009; 622:259-290.
- [15]. Dillon R, Fauci L, Fogelson A, Gaver D III. Modeling Biofilm Processes Using the Immersed Boundary Method. Journal of Computational Physics. 1996; 129(1):57-73.
- [16]. Stockie J, Green S. Simulating the motion of flexible pulp fibres using the immersed boundary method. J. Comput. Phys. 1998; 147:147-165.
- [17]. Zhu L, Peskin C. Simulation of a flapping flexible filament in a flowing soap film by the immersed boundary method. Journal of Computational Physics. 2002; 179(2):452-468.

- [18]. Gilmanov A, Sotiropoulos F. A hybrid cartesian/immersed boundary method for simulating flows with 3d, geometrically complex, moving bodies. *Journal of Computational Physics*. 2005; 207(2):457.
- [19]. Borazjani I, Sotiropoulos F. Numerical investigation of the hydrodynamics of anguilliform swimming in the transitional and inertial flow regimes. *Journal of Experimental Biology*. 2009; 212(4):576–592. [PubMed: 19181905]
- [20]. Borazjani I, Sotiropoulos F. Numerical investigation of the hydrodynamics of carangiform swimming in the transitional and inertial flow regimes. *Journal of Experimental Biology*. 2008; 211:1541–1558. [PubMed: 18456881]
- [21]. Borazjani I, Sotiropoulos F. Vortex-induced vibrations of two cylinders in tandem arrangement in the proximity–wake interference region. *Journal of fluid mechanics*. 2009; 621:321. [PubMed: 19693281]
- [22]. Gri th B, Hornung R, McQueen D, Peskin C. An adaptive, formally second order accurate version of the immersed boundary method. *Journal of Computational Physics*. 2007; 223(1):10–49.
- [23]. Leveque R, Li Z. The Immersed Interface Method for Elliptic Equations with Discontinuous Coefficients and Singular Sources. *SIAM Journal on Numerical Analysis*. 1994; 31(4):1019–1044.
- [24]. Ye T, Mittal R, Udaykumar H, Shyy W. An accurate cartesian grid method for viscous incompressible flows with complex immersed boundaries. *Journal of Computational Physics*. 1999; 156(2):209–240.
- [25]. Fadlun E, Verzicco R, Orlandi P, Mohd-Yusof J. Combined immersed-boundary finite-difference methods for three-dimensional complex flow simulations. *Journal of Computational Physics*. 2000; 161(1):35–60.
- [26]. Roma A, Peskin C, Berger M. An adaptive version of the immersed boundary method. *Journal of computational physics*. 1999; 153(2):509–534.
- [27]. Rossinelli D, Hejazialhosseini B, Spampinato D, Koumoutsakos P. Multicore/multi-gpu accelerated simulations of multiphase compressible flows using wavelet adapted grids. *SIAM Journal on Scientific Computing*. 2011; 33(2):512–540. URL <http://epubs.siam.org/doi/abs/10.1137/100795930>.
- [28]. Agresar G, Linderman J, Tryggvason G, Powell K. An adaptive, Cartesian, front-tracking method for the motion, deformation and adhesion of circulating cells. *Journal of Computational Physics*. 1998; 143(2):346–380.
- [29]. Borazjani I, Sotiropoulos F. On the role of form and kinematics on the hydrodynamics of self-propelled body/caudal fin swimming. *J Exp Biol*. 2010; 213(1):89–107. URL <http://jeb.biologists.org/cgi/content/abstract/213/1/89>. [PubMed: 20008366]
- [30]. Kim D, Choi H. Immersed boundary method for flow around an arbitrarily moving body. *Journal of Computational Physics*. 2006; 212(2):662.
- [31]. Beddhu M, Taylor LK, Whitfield DL. Strong conservative form of the incompressible navier-stokes equations in a rotating frame with a solution procedure. *J. Comput. Phys*. 1996; 128(2):427–437.
- [32]. Borazjani, I. Ph.D. thesis. University of Minnesota; Jun. 2008 Numerical simulations of fluid/structure interaction problems in biological flows.
- [33]. Vinokur M. An analysis of finite-difference and finite-volume formulations of conversion laws. *Journal of computational physics(Print)*. 1989; 81(1):1–52.
- [34]. Zélicourt D, Ge L, Wang C, Sotiropoulos F, Gilmanov A, Yoganathan A. Flow simulations in arbitrarily complex cardiovascular anatomies–An unstructured Cartesian grid approach. *Computers and Fluids*. 2009; 38(9):1749–1762.
- [35]. Appanaboyina S, Mut F, Löhner R, Putman C, Cebal J. Computational fluid dynamics of stented intracranial aneurysms using adaptive embedded unstructured grids. *International journal for numerical methods in fluids*. 2008; 57(5):475–493.
- [36]. Löhner R, Cebal J, Camelli F, Appanaboyina S, Baum J, Mestreau E, Soto O. Adaptive embedded and immersed unstructured grid techniques. *Computer Methods in Applied Mechanics and Engineering*. 2008; 197(25-28):2173–2197.

- [37]. Starius G. On composite mesh difference methods for hyperbolic differential equations. *Numer. Math.* 1980; 35:241–255.
- [38]. Starius G. Composite mesh difference methods for elliptic and boundary value problems. *Numer. Math.* 1977; 28:243–258.
- [39]. Volkov EA. The method of composite meshes for finite and infinite regions with piecewise smooth boundaries. *Proc. Steklov Inst. Math.* 1968; 96:145–185.
- [40]. Volkov EA. A finite difference method for finite and infinite regions with piecewise smooth boundaries. *Doklady.* 1966; 168(5):744–747.
- [41]. Tang H, Casey Jones S, Sotiropoulos F. An overset-grid method for 3D unsteady incompressible flows. *Journal of Computational Physics.* 2003; 191(2):567–600.
- [42]. Chesshire G, Henshaw W. A scheme for conservative interpolation on overlapping grids. *SIAM Journal on Scientific Computing.* 1994; 15:819.
- [43]. O'Connell S, Thompson P. Molecular dynamics-continuum hybrid computations: a tool for studying complex fluid flows. *Physical Review E.* 1995; 52(6):5792–5795.
- [44]. Nie X, Chen S, Robbins M, Robbins M. A continuum and molecular dynamics hybrid method for micro-and nano-fluid flow. *Journal of Fluid Mechanics.* 2004; 500:55–64.
- [45]. Flekkoy E, Wagner G, Feder J. Hybrid model for combined particle and continuum dynamics. *EPL (Europhysics Letters).* 2000; 52:271.
- [46]. Delgado-Buscalioni R, Coveney P. Continuum-particle hybrid coupling for mass, momentum, and energy transfers in unsteady fluid flow. *Physical Review E.* 2003; 67(4):046704.
- [47]. Wang Y, He G. A dynamic coupling model for hybrid atomistic-continuum computations. *Chemical engineering science.* 2007; 62(13):3574–3579.
- [48]. Fedosov DA, Karniadakis GE. Triple-decker: Interfacing atomistic-mesosopic-continuum flow regimes. *Journal of Computational physics.* 2009; 228(4):1157–1171.
- [49]. Dupuis A, Kotsalis E, Koumoutsakos P. Coupling lattice boltzmann and molecular dynamics models for dense fluids. *Physical Review E.* 2007; 75(4):046704.
- [50]. Koumoutsakos P. Multiscale flow simulations using particles. *Annual Review of Fluid Mechanics.* 2005; 37(1):457–487.
- [51]. STEGER J, BENEK J. On the use of composite grid schemes in computational aerodynamics. *Computer Methods in Applied Mechanics and Engineering.* 1987; 64:301–320.
- [52]. Paik J, Escauriaza C, Sotiropoulos F. On the bimodal dynamics of the turbulent horseshoe vortex system in a wing-body junction. *Physics of Fluids.* 2007; 19:045107.
- [53]. Berger M. On conservation at grid interfaces. *SIAM Journal on Numerical Analysis.* 1987; 24(5):967–984.
- [54]. Wang Z. A fully conservative interface algorithm for overlapped grids. *Journal of Computational Physics.* 1995; 122(1):96–106.
- [55]. Wright J, Shyy W. A pressure-based composite grid method for the Navier-Stokes equations. *Journal of computational physics(Print).* 1993; 107(2):225–238.
- [56]. Rai MM. A conservative treatment of zonal boundaries for Euler equation calculations. *Journal of Computational Physics.* 1986; 62(2):472–503.
- [57]. Zang Y, Street R. A composite multigrid method for calculating unsteady incompressible flows in geometrically complex domains. *International Journal for Numerical Methods in Fluids.* 1995; 20(5):341–362.
- [58]. Burton T, Eaton J. Analysis of a fractional-step method on overset grids. *Journal of Computational Physics.* 2002; 177(2):336–364.
- [59]. Borazjani I, Ge L, Sotiropoulos F. Curvilinear immersed boundary method for simulating fluid structure interaction with complex 3d rigid bodies. *Journal of Computational Physics.* 2008; 227(16):7587–7620. URL <http://www.sciencedirect.com/science/article/B6WHY-4SFS0TD-1/1/00d6c87a245>. [PubMed: 20981246]
- [60]. Ge L, Sotiropoulos F. A numerical method for solving the 3d unsteady incompressible navierstokes equations in curvilinear domains with complex immersed boundaries. *Journal of Computational Physics.* 2007; 225(2):1782–1809. [PubMed: 19194533]

- [61]. Gresho P, Sani R. On pressure boundary conditions for the incompressible navier-stokes equations. *International Journal for Numerical Methods in Fluids*. 1987; 7(10):1111–1145.
- [62]. Pletcher, R.; Tannehill, J.; Anderson, D. *Computational fluid mechanics and heat transfer*. 3rd Edition. CRC Press (Taylor & Francis Group); 2013.
- [63]. Henshaw WD. Ogen: An overlapping grid generator for Overture, Research Report UCRL-MA-132237. Lawrence Livermore National Laboratory. 1998
- [64]. Petersson NA. Hole-cutting for three-dimensional overlapping grids. *SIAM J. Sci. Comp.* 1999; 21:646–665.
- [65]. Suhs, NE.; Dietz, WE.; Rogers, SE.; Nash, SM.; Onufer, JT. Tech. Rep. NASA Ames Research Center; 2001. Pegasus user's guide. www.nas.nasa.gov/fogers/cgt/doc/man.html
- [66]. Chesshire GS, Henshaw WD. Composite overlapping meshes for the solution of partial differential equations. *Journal of Computational Physics*. 1990; 90(1):1–64.
- [67]. Balay, S.; Buschelman, K.; Gropp, WD.; Kaushik, D.; Knepley, MG.; McInnes, LC.; Smith, BF.; Zhang, H. PETSc Web page. 2009. <http://www.mcs.anl.gov/petsc>
- [68]. Balay, S.; Buschelman, K.; Eijkhout, V.; Gropp, WD.; Kaushik, D.; Knepley, MG.; McInnes, LC.; Smith, BF.; Zhang, H. Tech. Rep. ANL-95/11 - Revision 3.0.0. Argonne National Laboratory; 2008. PETSc users manual.
- [69]. Briley RW. Numerical method for predicting three-dimensional steady viscous flow in ducts. *Journal of Computational Physics*. 1974; 14(1):8–28.
- [70]. Ghia U, Ghia K, Shin C. High-Re solutions for incompressible flow using the Navier-Stokes equations and a multigrid method. *Journal of Computational Physics*. 1982; 48(3):387–411.
- [71]. Bovendeerd P, Vosse F, Steenhoven A. Steady entry flow in a curved pipe. *Journal of Fluid Mechanics*. 1987; 177:233–46.
- [72]. Rindt C, Van Steenhoven A, Janssen J, Vossers G. Unsteady entrance flow in a 90 curved tube. *J. Fluid Mech*. 1991; 226:445–474.
- [73]. Videler J, Hess F. Fast continuous swimming of two pelagic predators, saithe (*Pollachius virens*) and mackerel (*Scomber scombrus*): a kinematic analysis. *Journal of experimental biology*. 1984; 109:209–228.
- [74]. Kern S, Koumoutsakos P. Simulations of optimized anguilliform swimming. *Journal of Experimental Biology*. 2006; 209(24):4841–4857. 10.1242/jeb.02526. [PubMed: 17142673]
- [75]. Liu H, Wassersug R, Kawachi K. The three-dimensional hydrodynamics of tadpole locomotion. *Journal of Experimental Biology*. 1997; 200(22):2807–2819. 0022-0949 Peer Reviewed. [PubMed: 9344964]
- [76]. Yoganathan A, Chandran K, Sotiropoulos F. Flow in Prosthetic Heart Valves: State-of-the-Art and Future Directions. *Annals of Biomedical Engineering*. 2005; 33(12):1689–1694. [PubMed: 16389514]
- [77]. Yoganathan AP, He Z, Jones SC. Fluid mechanics of heart valves. *Annual Review of Biomedical Engineering*. 2004; 6:331–362.
- [78]. Dasi LP, Ge L, Simon HA, Sotiropoulos F, Yoganathan AP. Vorticity dynamics of a bileaflet mechanical heart valve in an axisymmetric aorta. *Physics of Fluids*. 2007; 19:067105. aIP.
- [79]. Grigioni M, Daniele C, Del Gaudio C, Morbiducci U, Balducci A, D'Avenio G, Barbaro V. Three-dimensional numeric simulation of flow through an aortic bileaflet valve in a realistic model of aortic root. *ASAIO Journal*. 2005; 51(3):176. [PubMed: 15968945]
- [80]. Nobili M, Morbiducci U, Ponzini R, Del Gaudio C, Balducci A, Grigioni M, Maria Montecchi F, Redaelli A. Numerical simulation of the dynamics of a bileaflet prosthetic heart valve using a fluid-structure interaction approach. *Journal of biomechanics*. 2008; 41(11):2539–2550. [PubMed: 18579146]
- [81]. Tai C, Liew K, Zhao Y. Numerical simulation of 3D fluid–structure interaction flow using an immersed object method with overlapping grids. *Computers and Structures*. 2007; 85(11-14): 749–762.
- [82]. Saber N, Gosman A, Wood N, Kilner P, Charrier C, Firmin D. Computational flow modeling of the left ventricle based on in vivo mri data: initial experience. *Annals of biomedical engineering*. 2001; 29(4):275–283. [PubMed: 11339325]

- [83]. Pedrizzetti G, Domenichini F. Nature optimizes the swirling flow in the human left ventricle. *Physical Review Letters*. 95(10)
- [84]. Domenichini F, Pedrizzetti G, Baccani B. Three-dimensional filling flow into a model left ventricle. *Journal of Fluid Mechanics*. 2005; 539:179–198.
- [85]. Domenichini F, Pedrizzetti G. Intraventricular vortex flow changes in the infarcted left ventricle: numerical results in an idealised 3d shape. *Computer Methods in Biomechanics and Biomedical Engineering*. 2011; 14(01):95–101. [PubMed: 21271416]
- [86]. Pedrizzetti G, Domenichini F, Tonti G. On the left ventricular vortex reversal after mitral valve replacement. *Annals of biomedical engineering*. 2010; 38(3):769–773. [PubMed: 20094914]
- [87]. de Zélicourt D, Pekkan K, Kitajima H, Frakes D, Yoganathan A. Single-Step Stereolithography of Complex Anatomical Models for Optical Flow Measurements. *Journal of Biomechanical Engineering*. 2005; 127:204. [PubMed: 15868804]

- An overset-curvilinear immersed boundary method in a general moving reference frame
- Efficient communication for parallel implementation of overset grids
- Different flux corrections at the interfaces of overset grids required by the fractional step method
- Self-propelled fish-like swimming with overset grids in a general moving frame
- A mechanical heart valve opened by the contraction of the left-ventricle

Fig. 1. Anatomical geometry of the left heart system consisting of multiple branching and large moving boundaries.

Fig. 2. Schematic of the inertial and non-inertial reference frames. The $x^{non-int}$ and x^{int} are the position vectors in the non-inertial and inertial reference frames, respectively, and the origin of the non-inertial reference frame is positioned at x^{ctr} from the inertial reference frame. Ω is the instantaneous angular velocity of the non-inertial frame.

Fig. 3.

2D schematic of a typical chimera grid layout with three sub-grids Ψ_1 to Ψ_3 . The boundary condition at the interfaces Γ_1^3, Γ_2^1 , etc are interpolated from the other sub-grid. For Domain Ψ_1 inside Γ_1^2 is blanked (blanking region). We reconstruct the velocities in the buffer layer between the two dashed lines inside the blanked region from domain Ψ_2 .

Fig. 4.

Schematic of search and trilinear interpolation for a point p of the recipient domain in donor cell. d^k for $k = 1, 6$ shows the distance to the k th surface of the donor cell. p_{mid}^1 , r_1^1 , and r_2^1 depict the midpoint and vectors for constructing the inward normal on the $k = 1$ surface of the donor cell for the point-in-the-box test.

Fig. 5. Schematic of a grid cell surface with four corners numbered 1 to 4. $\mathbf{n}$ is the normal vector; $\mathbf{x}_{1234}$ is the center of the cell surface; $\mathbf{x}_{123}$ and $\mathbf{x}_{134}$ are the center of the triangles 123 and 134, respectively; S_{1234} is the area of the cell surface; S_{123} and S_{134} are the areas of the triangles 123 and 134, respectively.

Fig. 6.
Time histories of the torque coefficient for the flow around a rotationally oscillating cylinder at $Re = 300$.

Fig. 7. Overset grid layout for the lid-driven cavity flow. The inner domain and blanking region boundaries are shown by thick red and blue lines, respectively.

Fig. 8. Contours of horizontal (left) and vertical (right) velocity components for the lid-driven cavity flow ($Re = 100$) on the overset grid with grid spacing h of (a) $1/64$, (b) $1/128$, and (c) $1/256$. The blue dashed lines represent the contours on the benchmark single grid ($h=1/512$). The black solid and the red dotted lines are the contours on the outer and inner blocks of the overset grid, respectively.

Fig. 9. Overset grid solution error relative to the benchmark single grid solution as a function of grid size in log-log scale. The error shows more than 2^{nd} order reduction with mesh refinement.

Fig. 10. The geometry and grid for the 90° pipe bend. Three curvilinear overlapping grids are used to discretize the geometry.

Fig. 11.

Incoming flow waveform. Solid line: analytical flow waveform prescribed at the inlet of the flow domain obtained by assuming that the inlet flow follows the Womersley solution; Points: the measurements of [72]. Taken from [60].

Fig. 12.

Steady flow through a 90° pipe bend. The calculated streamwise velocity profile at the plane of symmetry of the chimera grid (lines) are compared with experimental results (points). r_i and r_o are the inner and outer bend radius, respectively.

Fig. 13.

Pulsatile flow through a 90° pipe bend. The calculated streamwise velocity profile at the plane of symmetry of the chimera grid CH1 (solid black lines), CH2 (dotted black lines), CH3 (dashed black lines) are compared with numerical results of Ge and Sotiropoulos [60] (red lines) and experimental results (points) at time (a) $t = 0$ (b) $t = 0.25T$ (c) $t = 0.5T$, and (d) $t = 0.75T$. r_i and r_o are the inner and outer bend radius, respectively.

Fig. 14.

The overset grid layout near the mackerel and the blanking region is shown. The mackerel is meshed with triangular elements required by the CURVIB method. The overset grid is shown only on the midplane of the fish. For clarity only every third grid point is plotted.

Fig. 15.
Comparison of the self-propelled fish swimming speed on single and overset grid.

Fig. 16.

Contours of axial velocity (top), lateral velocity (middle), and vorticity (bottom) in the midplane of the fish for the single grid (blue), and the overset grid with outer sub-grid (red) and inner sub-grid (black). Dashed lines represent negative values. The contour lines of each sub-grid on the overlapping region are consistent with the single grid solution and vortices are advected from one sub-grid to the other.

Fig. 17.
The 3D vortical structures visualized by the iso-surfaces of q -criteria. The wake structure is bifurcated into two rows of vortices.

Fig. 18.

The CPU time versus the number of CPUs for the overset grid (2 blocks) and the single grid (1 block) simulations in log-log scale. It can be observed that the overset simulations are faster for all the tested number of CPUs.

Fig. 19.

Contours of vorticity in the midplane of the four fish swimming in the diamond arrangements. The contour lines of each sub-grid on the overlapping region are consistent and vortices are advected from one domain to the other. Red color denotes contours on the background grid while black on the inner sub-grids. Dashed lines represent negative values. Thick black lines represent the boundaries of the inner sub-grids

Fig. 20. The 3D vortical structures visualized by the iso-surfaces of q-criteria for four fish swimming in diamond arrangement.

Fig. 21.
The bi-leaflet mechanical heart valve is meshed with triangular elements as required by the CURVIB method.

Fig. 22.

The left ventricle is meshed with triangular elements as required by the CURVIB method. The kinematics of the left ventricle is prescribed based on the radial, r , and longitudinal (from based to apex) locations of each grid node. The silhouettes demonstrate the shape of the left ventricle at several time instants.

Fig. 23.

(a) The computational setup of the overset-grid and the valve and left ventricle as immersed bodies. The left-ventricle motion of is inside a Cartesian grid while the valve is implanted at the aortic root inside the body-fitted grid of the aorta. (b) A cross-section of the aorta body-fitted grid. For clarity only every four grid nodes are plotted in (a) and (b).

Fig. 24. Out-of-plane vorticity contour at the midplane of the valve indicating the formation of sinuses and leaflets shear layers (left), Von Karman shedding of vortices from the leaflets (middle) and the breakdown of vortical structures after the peak systole (right).

Fig. 25.
The 3D iso-surfaces of Q-criteria at three different time instants in the cycle.