Graphical Abstract

Soft, slender and active structures in fluids: embedding Cosserat rods in vortex methods
Arman Tekinalp, Yashraj Bhosale, Songyuan Cui, Fan Kiat Chan, Mattia Gazzola,
Highlights

Soft, slender and active structures in fluids: embedding Cosserat rods in vortex methods
Arman Tekinalp, Yashraj Bhosale, Songyuan Cui, Fan Kiat Chan, Mattia Gazzola,

- Framework for simulating heterogeneous structures made of slender elastic bodies and immersed in viscous flows.
- Two-way coupling between Cosserat rod theory (slender elastic bodies) and vortex methods (incompressible viscous flows).
- Rigorous validation and convergence analysis against a battery of experimental and computational benchmarks.
- Inclusion of a wide range of multi-physics demonstrations from muscular actuation and bio-locomotion to viscous streaming and magnetism.
- Scalability enabled via distributed computing for simulating large-scale problems requiring fine spatial resolutions.
Soft, slender and active structures in fluids: embedding Cosserat rods in vortex methods

Arman Tekinalp\textsuperscript{a,1}, Yashraj Bhosale\textsuperscript{a,1}, Songyuan Cui\textsuperscript{a}, Fan Kiat Chan\textsuperscript{a}, Mattia Gazzola\textsuperscript{a,b,c,}\textsuperscript{*}

\textsuperscript{a}Mechanical Sciences and Engineering, University of Illinois Urbana-Champaign, Urbana, IL 61801, USA
\textsuperscript{b}Carl R. Woese Institute for Genomic Biology, University of Illinois Urbana-Champaign, Urbana, IL 61801, USA
\textsuperscript{c}National Center for Supercomputing Applications, University of Illinois Urbana-Champaign, Urbana, IL 61801, USA

Abstract

We present a hybrid Eulerian-Lagrangian method for the direct simulation of three-dimensional, heterogeneous structures made of soft fibers and immersed in incompressible viscous fluids. Fiber-based organization of matter is pervasive in nature and engineering, from biological architectures made of cilia, hair, muscles or bones to polymers, composite materials or soft robots. In nature, many such structures are adapted to manipulate flows for feeding, locomotion or energy harvesting, through mechanisms that are often not fully understood. While simulations can support the analysis (and subsequent translational engineering) of these systems, extreme fibers’ aspect-ratios, large elastic deformations and two-way coupling with three-dimensional flows, all render the problem numerically challenging. To address this, we couple Cosserat rod theory, which exploits fibers’ slenderness to capture their dynamics in one-dimensional, accurate fashion, with vortex methods via a penalty immersed boundary technique. The favorable properties of the resultant hydroelastic solver are demonstrated against a battery of benchmarks, and further showcased in a range of multi-physics scenarios, involving magnetic actuation, viscous streaming, biomechanics, multi-body interaction, and self-propulsion.

Keywords: vortex methods, immersed boundary method, Cosserat rods, soft body, magnetism, self-propulsion, soft robotics, multiphysics, flow–structure interaction, distributed computing

1. Introduction

This paper presents a hybrid Eulerian-Lagrangian formulation that captures the two-way coupling between multiple heterogeneous, elastic, slender bodies and incompressible viscous fluids. Soft slender structures are ubiquitous and pervasive in natural and artificial systems, in active and passive settings, and across scales, from muscles, tendons and bones that make up full organisms \cite{1, 2, 3}, to polymers, textiles \cite{4, 5, 6, 7}, metamaterials \cite{8, 9, 10} and soft robots \cite{2, 11}. In nature, the interaction between these structures and flow gives rise to a wide array of complex phenomena, ranging from ciliated organisms’ feeding strategies \cite{12}, cephalopods’ stereotypical reaching and grasping motions \cite{13, 14, 3}, hagfish’s predatory behavior and body knotting \cite{15}, aquatic animals’ locomotion \cite{16, 17, 18, 19, 20, 21, 22, 23, 24}, to the interplay between marine vegetation growth and ocean currents \cite{25}. Simulating and analyzing these systems can be valuable to understanding and/or engineering them. However, the presence of extreme aspect ratios in these systems, along with difficulties associated with large elastic deformations and three-dimensional flow coupling, gives rise to significant numerical challenges.

To address these challenges, here we combine Cosserat rod theory and vortex methods. Vortex methods \cite{26, 27}, a class of techniques used to resolve flow-structure interaction, solve the velocity-vorticity formulation of momentum equations (as opposed to the velocity–pressure formulation used in other methods), while discretizing solid and fluid phases in Eulerian, Lagrangian, or hybrid fashions. These methods entail a number of favorable features: guaranteed convergence, accuracy, use of efficient Fourier transforms for solving Poisson equations, compact support of vorticity leading to automatic local (r-)adaptivity, natural treatment of far-field boundary conditions, ability to model complex solid morphologies, computational economy and software scalability rivaling traditional velocity-pressure methods \cite{26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38}. These advantages position vortex method as a versatile option to capture the dynamics of unsteady, complex bodies \cite{33, 37, 38} across scales \cite{36, 39, 40}, to deal with contact physics \cite{32}, multiphase \cite{41} and compressible flows \cite{42, 43}, in 2D as well as in 3D \cite{34, 44, 45}. Despite this versatility, little effort has been made to capitalize on these advantages to resolve two-way coupled dynamics between multiple heterogeneous elastic bodies and the surrounding viscous fluid.

\footnote{*Corresponding author
Email address: mgazzola@illinois.edu (Mattia Gazzola)
1Equal contribution.}

Preprint submitted to Journal of Computational Physics January 19, 2024
Recently, advancements have been made in this direction [46], by integrating inverse map techniques [47] and vortex method to capture the interaction between bulk rigid and elastic bodies and the surrounding fluid. Despite demonstrating versatility, accuracy, and robustness in several multiphysics scenarios, the simulation of fiber-based systems remains hindered by the high spatial resolutions required to resolve geometric scales disparities. One possible way to address this challenge is to utilize Cosserat rod theory [48], which exploits the slenderness of fibers to capture their three-dimensional motions and deformations via a one-dimensional Lagrangian representation. This representation entails a number of attractive features, namely, accuracy [49], linear scaling of computational complexity with rod resolution [49], seamless integration of non-linear constitutive models [2, 3] and external environmental effects such as friction, contact [2, 50], magnetism [51], and hydrodynamic loads [52, 53]. These advantages have resulted in the extensive adoption of elastic rod theory across a wide range of scenarios, including textile manufacturing [4, 5], plant mechanics [54, 55], artificial muscle modeling [56, 57], musculoskeletal and biological tissue modeling [2, 58, 3, 59], bio-hybrid machines [2, 60, 11, 61], 3D printing [62, 63], DNA strands [64, 65], meta-materials [66, 67, 9, 10], model-based [68, 69, 70, 71] and model-free [72, 73] control of soft robots.

Here, the coupling between Cosserat rods and vortex methods is achieved via a mixed Eulerian-Lagrangian approach, and the derived solver’s accuracy, robustness, versatility and scalability is demonstrated through a range of multiphysics scenarios involving magnetically actuated cilia carpets, viscous streaming, biomechanics, multi-body interaction, and self-propulsion. The work is organized as follows: governing equations and modeling techniques are described in Section 2 and Section 3, respectively; proposed algorithm and numerical discretization are detailed in Section 4; rigorous benchmarking and convergence analysis is presented in Section 5; versatility and robustness of the solver is illustrated through a variety of multifaceted cases in Section 6; finally, concluding remarks are provided in Section 7.

2. Governing equations

Here, we provide a description of the governing equations and constitutive laws that dictate the dynamics of multiple rigid bodies and elastic slender structures within a viscous fluid.

2.1. Governing equations for solids and fluids

We start by considering a three-dimensional domain \( \Sigma \) that encompasses an incompressible viscous fluid, as well as rigid and elastic bodies. As a convention, the subscripts \( f \), \( r \) and \( e \) refer to the fluid, rigid body and the elastic body phases, respectively. We use the notations \( \Omega_j \) and \( \partial \Omega_j \) (where \( i = 1, \ldots, N_j \)) to represent the support and boundaries of the elastic solids, and \( \Omega_r \) and \( \partial \Omega_r \) (where \( j = 1, \ldots, N_r \)) for the rigid solids. By defining \( \Omega = \Omega_j \cup \Omega_r \) as the region occupied by solid material, the remaining region \( \Sigma - \Omega \) is occupied by the fluid. The balance of linear and angular momentum in the fluid domain (for infinitesimal Eulerian volumes \( d\mathbf{x} \)) leads to the incompressible Navier–Stokes equations

\[
\nabla \cdot \mathbf{v}_f = 0; \quad \frac{\partial \mathbf{v}_f}{\partial t} + (\mathbf{v}_f \cdot \nabla) \mathbf{v}_f = -\frac{1}{\rho_f} \nabla p_f + \mathbf{v}_f \nabla^2 \mathbf{v}_f + \mathbf{f}_f, \quad \mathbf{x} \in \Sigma - \Omega
\]

(1)

where \( t \in \mathbb{R}^+ \) represents time, \( \mathbf{v}_f : \Sigma \times \mathbb{R}^+ \rightarrow \mathbb{R}^3 \) represents the velocity field, \( \rho_f \) denotes the fluid density, \( \mathbf{v}_f \) denotes the fluid’s kinematic viscosity, \( p_f : \Sigma \times \mathbb{R}^+ \rightarrow \mathbb{R} \) represents the hydrostatic pressure field and \( \mathbf{f}_f : \Sigma \times \mathbb{R}^+ \rightarrow \mathbb{R}^3 \) represents the body force field. The body force field \( \mathbf{f}_f \) can be further decomposed into two parts \( \mathbf{f}_f = \mathbf{f}_{f,v} + \mathbf{f}_{f,\alpha} \), where \( \mathbf{f}_{f,v} \) is a conservative volumetric force (like gravitational acceleration), and \( \mathbf{f}_{f,\alpha} \) is a coupling force used to impose the no-slip condition on the fluid-structure boundary \( \partial \Omega \), as described in Section 3.2. We make the assumption that all the aforementioned fields are sufficiently smooth both in time and space. The interaction between the fluid and elastic solid phases occurs solely through boundary conditions, which enforce velocity continuity (no-slip) and traction forces at all interfaces between the fluid and elastic solid.

\[
\mathbf{v} = \mathbf{v}_f = \mathbf{v}_{e,i}, \quad \mathbf{t}_f \cdot \mathbf{n} = \mathbf{t}_{e,i} \cdot \mathbf{n}, \quad \mathbf{x} \in \partial \Omega_{e,i}
\]

(2)

where \( \mathbf{n} \) denotes the unit outward normal vector at the interface \( \partial \Omega_{e,i} \). The interfacial velocities in the fluid and \( i \)th elastic body are represented by \( \mathbf{v}_f \) and \( \mathbf{v}_{e,i} \) respectively. Similarly, the interfacial Cauchy stress tensors in the fluid and \( i \)th elastic body are denoted as \( \mathbf{t}_f \) and \( \mathbf{t}_{e,i} \) respectively. Within the regions \( \Omega_{e,i} \) occupied by rigid solids \( (j = 1, \ldots, N_r) \), the velocities are kinematically constrained to rigid body modes of pure translation and rotation. Consequently, the interaction between the rigid bodies and the fluid domain is mediated only by the no-slip boundary condition

\[
\mathbf{v} = \mathbf{v}_f = \mathbf{v}_{e,j} = \mathbf{v}_{e,m} + \omega_{e,j} \times (\mathbf{x} - \mathbf{x}_{com,j}), \quad \mathbf{x} \in \partial \Omega_{e,j}
\]

(3)

where \( \mathbf{v}_{e,j} \) is the rigid body velocity field, \( \mathbf{x}_{com,j} \) is the center of mass (COM) position, \( \mathbf{v}_{e,m} \) is the center of mass velocity, and \( \omega_{e,j} \) is the angular velocity about the COM of the \( j \)th rigid body.
2.2. Cosserat rod model for soft, slender heterogeneous structures

To close the above set of equations (Eqs. 1 to 3), it is necessary to specify the governing equations and constitutive laws of the elastic structures. Resolving the dynamics of slender, fiber-like bodies via 3D bulk elasticity frameworks is particularly cumbersome and expensive, given their aspect ratios. We thus adopt the use of Cosserat rod theory.

Cosserat rod theory captures three-dimensional motions and deformations of slender elastic structures via a one-dimensional Lagrangian representation. Figure 1a presents the setup of the Cosserat rod theory model. Each point on the Cosserat rod is characterized by a center-line \( x(s,t) \) ∈ \( \mathbb{R}^3 \) and an orthonormal frame \( \Omega(s,t) = [d_1(s,t)\, d_2(s,t)\, d_3(s,t)] \) (trios of unit vectors \( d_i \in \mathbb{R}^3 \)), which defines the frame transformations between the laboratory and local frames. Here, \( s \in [0, L(t)] \) is the center-line arc-length coordinate, where \( L \) is the rod’s current length, and \( t \) is the time. Any vector \( r \in \mathbb{R}^3 \) defined in the laboratory frame can be transformed into its local frame counterpart \( \bar{r} \in \mathbb{R}^3 \), by \( \bar{r} = \Omega r \) and from local to laboratory frame by \( r = \Omega^T \bar{r} \). For an unsharable and inextensible rod, \( d_3 \) is parallel to the rod tangent \( \bar{x}_s \in \mathbb{R}^3 = \partial_s x \), with \( d_1 \) (normal) and \( d_2 \) (binormal) spanning the cross-section of the rod. Under shear and extension, the rod tangent \( \bar{x}_s \) and \( d_3 \) are no longer parallel and their shift can be quantified via the shear vector \( \bar{\tau} \in \mathbb{R}^3 = \Omega(x_s - d_3) \) in the local frame. The curvature vector \( \kappa(s,t) \in \mathbb{R}^3 \) encodes \( \Omega \)’s rotations along the rod through the relation \( \partial_s d_j = \kappa \times d_j \). The rate of change of \( \Omega \) is given by \( \partial_t d_j = \omega \times d_j \), where \( \omega \in \mathbb{R}^3 \) corresponds to the angular velocity.

![Figure 1: (a) Cosserat rod model. A soft filament (blue) deforming in three-dimensional space is described by a center-line \( x(s,t) \), and a local orthonormal frame \( \Omega(s,t) = [d_1(s,t)\, d_2(s,t)\, d_3(s,t)] \), both functions of centerline’s arc-length coordinate \( s \) and time \( t \). Orange and purple lines around the elastic rod (blue) trace out \( d_1, d_2, d_3 \) respectively. (b) Penalty immersed boundary (pIBM) setup. An elastic body rod (orange region) defined on a Lagrangian domain \( \Omega \) is immersed in a viscous fluid represented on a fixed Eulerian domain \( \Sigma \). The elastic rod body is discretized into Lagrangian elements along the centerline, separated by nodes \( x_{cm}(s,t) \) on the same centline. An immersed boundary \( \partial \Omega \) body-fitted to the elastic body and discretized via forcing points (pink circles), runs parallel to the Lagrangian elements with \( x_{cm} = x_{cm} \) and \( r_{cm} \) is the moment arm of the forcing points. The green area \( \partial \Sigma \) refers to the boundary or interface that separates a Lagrangian element from the viscous fluid that surrounds it.](image)

defining the velocity of the centerline \( v \in \mathbb{R}^3 = \partial_t x \), second area moment of inertia \( I \in \mathbb{R}^3 \), cross-sectional area \( A \), and density \( \rho \), the dynamics [74] of a soft slender body is described by

\[
\frac{\partial^2}{\partial t^2} (\rho A x) = \partial_s (Q^T \tilde{n}) + f
\]  

(4)

\[
\partial_s (\rho \tilde{I} \tilde{\omega}) = \partial_s \tilde{\tau} + \tilde{k} \times \tilde{\tau} + (Q x_s \times \tilde{n}) + (\rho \tilde{I} \tilde{\omega}) \times \tilde{\omega} + Qc
\]  

(5)

where Eqs. 4 and 5 represent linear and angular momentum balance at every cross section, \( \tilde{n} \) and \( \tilde{\tau} \) are the internal forces and torques, \( f \) and \( c \) are the external force and torque densities, respectively. To close the above Eqs. 4 and 5, it is necessary to specify the form of the internal forces (\( \tilde{n} \)) and torques (\( \tilde{\tau} \)) generated in response to shear (\( \tilde{\sigma} \)) and bend/twist (\( \tilde{k} \)) strains. In this study, for simplicity, we assume a perfectly elastic material so that the stress–strain relations are linear. We note that Cosserat rod theory is not limited to modeling linear elasticity, but can be extended to non-linear elastic materials [2, 3]. Within the linear elasticity model, the internal forces \( \tilde{n} \) can be expressed as

\[
\tilde{n} = \tilde{S} (\tilde{\sigma} - \tilde{\sigma}_0)
\]  

(6)

where \( \tilde{\sigma}_0 \) is the intrinsic shear strain and \( \tilde{S} \in \mathbb{R}^{3×3} = \text{diag} (\tilde{S}_1, \tilde{S}_2, \tilde{S}_3) = \text{diag} (\alpha_c \rho A, \alpha_s \rho A, EA) \) is the shear/stretch stiffness matrix. Here, \( \tilde{S}_1, \tilde{S}_2 \) and \( \tilde{S}_3 \) are the shear/stretch rigidities about \( d_1, d_2 \) and \( d_3 \), respectively. Additionally, \( E, G \) and \( \alpha_c \) correspond to the Young’s modulus, the shear modulus, and the shear correction factor, respectively (for details, see [74]). Similarly, the internal torques \( \tilde{\tau} \) can be expressed as

\[
\tilde{\tau} = \tilde{B} (\tilde{k} - \tilde{k}_0)
\]  

(7)
where $\kappa_0$ is the intrinsic rod curvature and $\vec{B} \in \mathbb{R}^{3\times3} = \text{diag}(B_1, B_2, B_3) = \text{diag}(EI_1, EI_2, GI_3)$ is the bend/twist stiffness matrix. Here, $B_1$, $B_2$ and $B_3$ are the bend/twist rigidities about $d_1$, $d_2$ and $d_3$, respectively. Additionally, $I_1$, $I_2$ and $I_3$ correspond to the values of second moment of inertia about $d_1$, $d_2$ and $d_3$, respectively.

The Cosserat rod representation presented above enables a number of favorable features: (1) it captures 3D dynamic effects and all modes of deformation (bending, twist, shear, stretch), particularly relevant in the case of biological or elastomeric materials [74, 3, 55]; (2) non-linear constitutive models can be seamlessly integrated via Eqs. 6 and 7 [2, 3, 55]; (3) complexity scales as $O(n)$ with axial resolution [74]; (4) kinematic or dynamic boundary conditions can be directly enforced, allowing the assembly of multiple rods into complex architectures [2, 3]; (5) continuum and/or distributed actuation and environmental loads can be directly incorporated via $f$ and $c$ into the Cosserat rod dynamics, rendering the inclusion of contact dynamics [9, 10], friction [9, 10], muscular activity [2, 3], magnetism [75] or hydrodynamics [52, 53] straightforward.

3. Methodology

Having established the fundamental governing equations and boundary conditions, we will now introduce the methodologies employed for solving these coupled equations. We employ the Cosserat rod equations to track the dynamics and deformations of elastic slender bodies, and the immersed boundary method to solve the coupling problem in a hybrid Eulerian/Lagrangian vortex method framework.

3.1. Vortex methods

Vortex methods consider the vorticity form of the Navier–Stokes equations, obtained by taking the curl of Eq. 1

$$\frac{\partial \omega_f}{\partial t} + (v_f \cdot \nabla) \omega_f - (\omega_f \cdot \nabla) v_f = v_f \nabla^2 \omega_f + \nabla \times f_f,$$

where $\omega_f : \Sigma \times \mathbb{R}^+ \mapsto \mathbb{R}^3 := \nabla \times v_f$ represents the vorticity field in the fluid. With pressure $p$ eliminated from the governing equations, an incompressible velocity field is then directly recovered from the vorticity by solving a Poisson equation using appropriate boundary conditions on $\Sigma$

$$\nabla^2 \psi_f = -\omega_f; \quad v_f = \nabla \times \psi_f$$

(9)

where $\psi_f : \Sigma \times \mathbb{R}^+ \mapsto \mathbb{R}^3$ (in 2D $\psi_f$ corresponds to the streamfunction). Favorable features of this approach are: guaranteed convergence, accuracy, use of efficient Fourier transforms for solving Poisson equations, compact support of vorticity leading to automatic local (r-)adaptivity, natural treatment of far field boundary conditions, ability to model complex solid morphologies, computational economy and software scalability [26, 27, 28, 29, 30, 31, 32, 33, 17, 34, 35, 36, 37, 38].

3.2. Immersed boundary method

We connect the forcing terms $f$ from the Navier–Stokes equations (Eq. 8) and the Cosserat rod equations (Eqs. 4 and 5) via the immersed boundary method. Immersed boundary methods (IBMs) are widely adopted computational techniques for simulating fluid-structure interaction problems involving complex geometries [76, 77]. In IBMs, the fluid domain is typically discretized using an Eulerian grid, while the immersed object is represented on a Lagrangian grid that can move freely with respect to the underlying Eulerian fluid mesh. The interaction between the fluid and solid phases is modeled using a force distribution function, applied on the Lagrangian and Eulerian grids, to achieve two-way fluid-structure interactions. Due to their versatility, IBMs have been employed in a range applications, from microfluidics [78, 79] and biomedical engineering [80, 76] to the aerospace industry [81].

In this study, we adopt the penalty immersed boundary method (pIBM), a flavor of IBMs that adds penalty forces to the governing equations of both fluid and solid phases, to enforce appropriate boundary conditions (Eqs. 2 and 3) at the fluid-solid interface [82, 83]. Figure 1b presents the setup of the penalty immersed boundary method—an elastic body ($\Omega$) fitted with an immersed boundary at the fluid-solid interface ($\partial \Omega$) is submerged in a viscous fluid $\Sigma$. The steps involved in the pIBM formulation for the two-way coupling can be briefly summarized as follows: (1) interpolate the flow velocity $v_f$ from the Eulerian grid ($\Sigma$) and the body velocity $v_b$ from the Lagrangian domain ($\partial \Omega$) onto the immersed boundary ($\partial \Omega$); (2) compute the coupling force between the two phases based on the deviation between the interpolated velocities; (3) transfer the coupling forces via the forcing terms $f$ to the fluid ($\Sigma$) and the solid ($\partial \Omega$) domains, respectively.

We next present a detailed step-by-step implementation of the pIBM formulation. We begin by computing the velocity of the fluid phase on the Lagrangian boundary $\partial \Omega$ at position $x_{\partial \Omega}$ via interpolation

$$v_{f, \partial \Omega} = \int_{\Sigma} v_f(x) \delta(x_{\partial \Omega} - x) \, dx$$

(10)
where \( v_f(x) \), \( v_f \partial \Omega \) correspond to the fluid velocity at the Eulerian position \( x \) and on the immersed boundary (\( \partial \Omega \)), and \( \delta(\cdot) \) is the Dirac delta function. Simultaneously, we compute the velocity of the \( i \)-th body on \( \partial \Omega \) (\( v_f \partial \Omega \)) as

\[
v_f \partial \Omega = v_{i \text{cm}} + \omega_{i \text{cm}} \times r_{i \partial \Omega}
\]

where \( v_{i \text{cm}} \), \( \omega_{i \text{cm}} \), and \( r_{i \partial \Omega} \) represent the translational and rotational velocity of the center of mass (COM), and the moment arm of the immersed boundary from COM, respectively. With the velocities of both the phases on \( \partial \Omega \) known, the interaction force \( f_{\partial \Omega} \) between the fluid and the \( i \)-th body is then computed as

\[
f_{\partial \Omega} = -\int_0^t \alpha(v_f \partial \Omega - v_f \partial \Omega) \, dt - \beta(v_f \partial \Omega - v_f \partial \Omega)
\]

where \( \alpha \) and \( \beta \) are the coupling stiffness constant and the coupling damping constant, respectively. To enforce the fluid-solid interfacial conditions (Eqs. 2 and 3) accurately, we require \( \alpha \gg 1 \) (although too large values often lead to numerical instabilities [82]). Typically, \( \alpha = O(10^4) \) and \( \beta = O(10) \) are observed to strike a balance between accuracy and stability [84, 85]. The final step in the pIBM formulation involves transferring equal and opposite coupling forces to the fluid and solid phases. The coupling force on the fluid \( f_f \partial \Omega \) at the Eulerian position \( x \) is computed as

\[
f_f \partial \Omega(x) = -\int_{\partial \Omega} f_f \partial \Omega(x) \, \delta(x - x) \, dx_w.
\]

Before computing the coupling forces on the elastic body, note that the elastic rod/body is discretized into Lagrangian elements along the centerline (green zone, Fig. 1b), defined by the nodes \( x_{\text{cm}}(s, t) \) on the same centerline. Then the coupling forces \( f_i \partial \Omega \) and torques \( c_i \partial \Omega \) acting on a single Lagrangian element of the \( i \)-th body \( \Omega_i \) are computed as

\[
f_i \partial \Omega = \int_{\partial \Omega} f_i \partial \Omega \, dx_w \quad c_i \partial \Omega = \int_{\partial \Omega} (r_i \partial \Omega \times f_i \partial \Omega) \, dx_w
\]

where \( \partial \Omega \) refers to the fluid-solid interface of the Lagrangian element. The pIBM formulation presented above is (1) relatively easy to implement and can be integrated into existing numerical codes [86]; (2) decouples the representation of the fluid and solid phases, simplifying mesh generation (no conforming meshes [87]); (3) reduces computational cost by avoiding remeshing in the case of moving boundaries [77]; (4) enables efficiency and parallelization performance typical of structured grids [85]; (5) accommodates different phase representations (fluid or solid), enabling relatively straightforward integration of independently developed methodologies for fluid, solid, or other multi-physics components [88, 89].

In the next section, we proceed to describe the numerical discretization of the elements described above, with a detailed step-by-step explanation of our algorithm.

### 4. Numerical discretization and algorithm

We begin by spatially discretizing the system of equations for the fluid phase (Eqs. 8 and 9) via an Eulerian Cartesian grid of uniform spacing \( h \) to create our computational domain \( \Sigma_h \). The continuous flow fields previously defined are now replaced with their discrete counterparts on \( \Sigma_h \). For the elastic fibers, the Cosserat rod equations (Eqs. 4 and 5) are discretized over a Lagrangian grid, consisting of discrete elements along the centerline with uniform spacing \( ds \). Finally, the fluid-solid interfaces (immersed boundaries) are discretized using Lagrangian grids \( \partial \Omega_h \) body-fitted to the solid domain \( \Omega_h \), with uniform grid spacing \( h_{\partial \Omega} \). The spacing between Lagrangian grid points \( h_{\partial \Omega} \) is calculated based on the spacing of Eulerian grid points \( h \), ensuring that each Lagrangian point spans across 1-2 neighboring Eulerian points. The implementation of a relative grid spacing policy guarantees the synchronized convergence of fluid-solid dynamics, while also preventing artificial gaps between the two interfaces.

Next, we outline Algorithm 1 which describes one complete time step, spanning from \( t^n \) to \( t^{n+1} \), assuming that all relevant quantities are already known up to time \( t^n \).

#### 4.1. Poisson solve and velocity recovery

First, we solve the Poisson equation (Eq. 15) for the variable \( \psi \). Depending on the problem setup, we employ periodic or unbounded boundary conditions. To solve Eq. 15 on the grid, we use a Fourier-series based solver that operates with \( O(n \log(n)) \) complexity. This solver takes advantage of the diagonal nature of the Poisson operator in the case of periodic boundaries [90], allowing for spectral accuracy. For unbounded conditions, we utilize the zero padding technique described by Hockney and Eastwood [90]. Once \( \psi \) is obtained on the grid, we calculate the velocity according to Eq. 16 using the discrete second-order centered finite difference curl operator.
Algorithm 1 General algorithm

\begin{align}
\text{Poisson solve (Section 4.1)} & \quad \nabla^2 \psi_f^j = -\omega_f^p \\
\text{Velocity recovery (Section 4.1)} & \quad v_f^j = \nabla \psi_f^j + V_{f,\infty} \\
\text{Compute flow velocity on fluid-solid interface (Section 4.2)} & \quad v_{f,\partial\Omega}^j = \int_\Omega v_f^j(x) \delta(x_{\partial\Omega}^j - x) \, dx \\
\text{Compute body velocity on fluid-solid interface (Section 4.2)} & \quad v_{b,\partial\Omega}^j = v_{b,\text{cm}}^j + \omega_{b,\text{cm}}^j \times r_{b,\partial\Omega}^j \\
\text{Compute coupling force on fluid-solid interface (Section 4.2)} & \quad f_{f,\partial\Omega}^j = -\int_0^\epsilon \alpha \left( v_{f,\partial\Omega}^j - v_{f,\partial\Omega}^j \right) dt - \beta \left( v_{f,\partial\Omega}^j - v_{f,\partial\Omega}^j \right) \\
\text{Transfer coupling force to fluid (Section 4.2)} & \quad f_{f,\partial\Omega}^j(x) = -\int_{\partial\Omega} f_{f,\partial\Omega}^j(x_{\partial\Omega}^n) \delta(x - x_{\partial\Omega}^n) \, dx_{\partial\Omega} \\
\text{Transfer coupling force to body (Section 4.2)} & \quad f_{b,\partial\Omega}^j = \int_{\partial\Omega} f_{f,\partial\Omega}^j \, dx_{\partial\Omega}; \quad c_{i,\partial\Omega}^j = \int_{\partial\Omega} \left( r_{i,\partial\Omega}^n \times f_{f,\partial\Omega}^j \right) \, dx_{\partial\Omega} \\
\text{Fluid (vorticity) update (Section 4.3)} & \quad \frac{\partial \omega_f^{j+1}}{\partial t} + (v_f^j \cdot \nabla) \omega_f^j = \nabla \left( f_{f,\partial\Omega}^j + f_{f,\partial\Omega}^j \right) + v_f^j \nabla^2 \omega_f^j \\
\text{Vorticity propagation to next time step (Section 4.3)} & \quad \omega_f^{j+1} = \omega_f^j \\
\text{Elastic body update (Section 4.4)} & \quad x_{e,i}^n, v_{e,i}^n, Q_{e,i}^n, \omega_{e,i}^n \rightarrow x_{e,i}^{n+1}, v_{e,i}^{n+1}, Q_{e,i}^{n+1}, \omega_{e,i}^{n+1} \\
\text{Rigid body update (Section 4.4)} & \quad x_{r,i}^{n+1}, v_{r,i}^{n+1}, Q_{r,i}^{n+1}, \omega_{r,i}^{n+1} \\
\end{align}

4.2. Immersed boundary computation

For each solid body, the velocity of the fluid phase on the Lagrangian boundary \( \partial\Omega_b \) at position \( x_{\partial\Omega_b} \) is computed via the discrete form of Eq. 17, which reads

\[
v_{f,\partial\Omega}^j = \sum_{x_{\partial\Omega_b} \in \partial\Omega_b} v_f^j(x_{\partial\Omega_b}) \delta_b \left( x_{\partial\Omega_b} - x_{\partial\Omega_b} \right) h^d
\]

where \( d, \delta_b(\cdot) \) and \( g_b \) correspond to the grid dimension (2 for 2D and 3 for 3D), mollified Delta function, and support of the Delta function, respectively. The mollified Delta function \( \delta_b(\cdot) \) reads

\[
\delta_b(x) = \frac{1}{h^d} \sum_{i=1}^d W \left( \frac{x_i}{h} \right)
\]

where \( i \) symbolizes the grid index and \( W(\cdot) \) is an interpolation kernel. In this study, \( W(\cdot) \) is the four point Delta function introduced by Peskin [91] and widely used in IBM studies. Following interpolation, the velocity of the solid body is then computed on the Lagrangian boundary \( \partial\Omega_b \) via Eq. 18. With the velocities of both the phases on \( \partial\Omega_b \) known, the interaction force \( f_{\partial\Omega_b} \) is then computed via Eq. 19, with the time integral on the RHS of Eq. 19 computed discretely via a left Riemann sum. Following interaction force computations (Eq. 19), the coupling forces are transferred to the fluid phase via the discrete form of Eq. 20.

\[
f_{f,\partial\Omega_b}(x_b) = -\sum_{x_{\partial\Omega_b} \in \partial\Omega_b} f_{\partial\Omega_b}(x_{\partial\Omega_b}) \delta_b \left( x_b - x_{\partial\Omega_b} \right) h^d_{\partial\Omega_b}
\]

Simultaneously, equal and opposite forces and corresponding torques are transferred to the \( j \)th body \( \Omega_i \) via the discrete version of Eq. 21

\[
f_{i,\partial\Omega_b} = \sum_{\partial\Omega_b} f_{\partial\Omega_b} h^d_{\partial\Omega_b} \quad e_{i,\partial\Omega_b} = \sum_{\partial\Omega_b} (r_{i,\partial\Omega_b} \times f_{\partial\Omega_b}) h^d_{\partial\Omega_b}
\]

4.3. Fluid phase (vorticity) update

After the transfer of the coupling forces to the fluid (Eq. 20), we next update the fluid state (vorticity) via Eq. 22. The numerics of this step entails computing three main components that modify the vorticity, namely, the body forces \( \nabla \times (f_{f,\partial\Omega}^j + f_{f,\partial\Omega}^j) \), the diffusion flux \( v_f \nabla^2 \omega_f^j \) and the flow inertia flux, that includes the advection \( (v_f^j \cdot \nabla) \omega_f^j \) and stretching \((\omega_f^j \cdot \nabla) v_f^j \) terms. For the
the advection and stretching terms create spurious divergence in the vorticity field, leading to long-time numerical instabilities. To follow these reasons. In 2D flows, where the vortex stretching term \( \omega \) that since the vorticity \( \nabla \times f \) is defined as the curl of a smooth differentiable velocity field \( v_f = \nabla \times v_f \), theoretically it needs to satisfy the zero divergence constraint \( \nabla \cdot \omega = 0 \). While this is trivially satisfied in 2D, in 3D the numerical errors resulting from the advection and stretching terms create spurious divergence in the vorticity field, leading to long-time numerical instabilities. To mitigate this issue, based on insights from pseudospectral methods [93], we solve Eq. 22 via two back-to-back steps

\[
\frac{d\omega_i^n}{dt} + \nabla \times \left( \omega_i^n \times v_f^n \right) = \nabla \times \left( f_i^n + f_i^n \right) + v_f^n \nabla^2 \omega_i^n \tag{30}
\]

\[
\omega_i^n = L^p(\omega_i^n) \tag{31}
\]

where the first step (Eq. 30) involves reformulating the flow inertia terms into a rotational form \( \nabla \times \left( \omega_i^n \times v_f^n \right) \) (curl of a vector field). The second step (Eq. 31), similar to dealiasing in pseudospectral methods, corresponds to applying a compact discrete (only using neighbouring grid values) filter \( L^p(\cdot) \) on the vorticity outputted from Eq. 30. The superscript \( p \), taking integer values, corresponds to the filter order and qualitatively maps to the strength of the filter; lower values of \( p \) imply stronger filtering action. We set \( p = 5 \) for all 3D simulations to avoid artificial dampening of vorticity while preserving stability of our simulations. Additional details on the numerical implementation and transfer function of the compact discrete filters can be found in Jeanmart and Winckelmans [94].

Following the evaluation of all forcing terms in Eq. 22, we then perform temporal integration using a forward Euler timestep. The vorticity after this step is then carried over to the next timestep (Eq. 23).

4.4. Elastic and rigid body update

In Eq. 24 we resolve the elastic rod dynamics. We discretize the Cosserat rod equations (Eqs. 4 and 5) on a Lagrangian grid of uniform spacing \( ds \) along the centerline (thus forming a set of discrete cylindrical elements of length \( ds \)). For terms on the right-hand side of Eqs. 4 and 5 involving spatial differentiation along the centerline, we use second-order centered finite differences to compute the derivatives. Finally, with the internal and external, forces and torques known, we evolve the kinematic state (positions, orientations and velocities) of all the elastic rods (Eq. 24) using the symplectic, second-order position Verlet time integration scheme. Similarly, rigid bodies are updated in Eq. 25. Numerical details and validation of the Cosserat rod dynamics solver can be found in [49], while the implementation can be found in the open-source software PyElastica [95].

4.5. Restrictions on simulation time step

In our algorithm, we face three main restrictions on the time step size due to the existence of distinct time scales in the coupling problem. The first restriction is a typical CFL (Courant–Friedrich–Lewy) condition, that stems from the advection and stretching of vorticity inside the fluid

\[
\Delta t_1 \leq h \text{ CFL} \frac{\|v_f\|_{\infty}}{C^2}
\]

where \( v_f \) refers to the velocity field inside the fluid. The second restriction is the Fourier condition that ensures stability with regards to explicit time discretization of the viscous stresses inside the fluid

\[
\Delta t_2 \leq k h^2/(2 d v_f)
\]

where \( k \) is a constant usually set to be \( \leq 1 \). Here we set \( k = 0.9 \) throughout. Finally, the third restriction stems from the need to resolve shear waves inside elastic solids. This is a CFL-like restriction dependent on the shear wave speed \( c_{sh} \)

\[
\Delta t_3 \leq d s \text{ CFL} c_{sh}^{-1} = d s \text{ CFL} \sqrt{\rho_v/G}.
\]

Here, \( d s, \rho_v \) and \( G \) correspond to the rod’s Lagrangian grid spacing, solid density and shear modulus, respectively. Combining Eqs. 32 to 34, we obtain the final criterion to adapt the time step during simulation

\[
\Delta t = \min \{\Delta t_1, \Delta t_2, \Delta t_3\}
\]

We observe (a posteriori) that in several cases investigated in Section 5 and Section 6, the time-step restriction from the high elastic shear wave speeds \( \Delta t_3 \), rather than the CFL condition \( \Delta t_1 \) or the diffusion limit \( \Delta t_2 \), limits the global time-step \( \Delta t \). Potential solutions to improve computational efficiency consist in the use of local time stepping techniques [36] or the implicit update of the elastic body dynamics [96].

After providing a comprehensive explanation of our algorithm, we proceed to assess its accuracy and convergence properties through rigorous validations against a variety of analytical and numerical benchmarks.
5. Validation benchmarks

The conducted benchmark tests involve flow past a two-dimensional elastic rod under gravity, a two-dimensional elastic rod flapping in the wake of a rigid cylinder, cross-flow past a three-dimensional elastic rod under gravity, and self-propelled elastic anguilliform swimming in two and three dimensions. For each case, we conduct a convergence analysis by reporting the discrete error $e$ (of relevant physical quantities) as a function of the spatial discretization. We define $e$ as $e = \|p - p_{\text{ref}}\|/p_{\text{ref}}$, where $p$ is a physical quantity obtained from our method and $p_{\text{ref}}$ is the reference solution.

In different problems, the dynamics involved can be influenced by one or more key dimensionless numbers. Here, we provide a compilation of these numbers along with their respective physical interpretations

\[
Re := \frac{\rho_f V L}{\mu_f} \sim \text{inertial forces} / \text{viscous forces}; \quad \text{Cau} := \frac{\rho_f V^2}{E} \sim \text{elastic forces} / \text{inertial forces}; \quad \rho := \frac{\rho_e}{\rho_f} \sim \text{body inertia} / \text{flow inertia} \quad \text{AR} := \frac{L}{D} \sim \text{Slenderness} \quad (36)
\]

where $Re$, $\text{Cau}$, $\rho$, $AR$, $V$, $L$, $\mu_f$, $\rho_f$, $\rho_e$, $D_e$ and $L_e$ correspond to the Reynolds number, Cauchy number [97], density ratio, rod’s aspect ratio, velocity scale, length scale (i.e. the rod length), fluid viscosity, fluid density, elastic rod density, Young’s modulus of the rod and the rod’s cross-sectional diameter, respectively. Besides the numbers above, another dimensionless quantity has been conventionally employed to characterize the dynamics of slender body systems [98, 99]. This number, known as the dimensionless bending stiffness $K_b$, is defined as

\[
K_b := \frac{EI}{\rho_f V^2 L^2} \quad (2D) \quad \text{or} \quad \frac{EI}{\rho_f V^2 L^2 D^2} \quad (3D) \sim \text{bending mode forces} / \text{inertial forces} \quad (37)
\]

where $I$ corresponds to the second area moment of inertia of the rod’s cross-section about the normal/binormal axis (see Section 2.2). The slight difference in the definition of $K_b$ in 2D and 3D stems from the difference in units of $I$, which has units $m^3$ in 2D and $m^4$ in 3D. Lastly, we note that $K_b$ can be expressed as a function of the other non-dimensional parameters mentioned in Eq. 36. For instance, in the case of an elastic rod in 3D with a cross-sectional diameter $D$, the moment of inertia scales as $I \propto D^4$, resulting in the equivalent definition $K_b := (\text{Cau}AR^3)^{-1}$.

5.1. Flapping of a two-dimensional elastic rod in the wake of a rigid cylinder

We first demonstrate the ability of our solver to capture interactions between multiple elastic/rigid bodies immersed in a viscous fluid. Accordingly, we reproduce the case of a two-dimensional elastic rod flapping in the wake of a rigid cylinder, characterized in detail by Turek and Hron [98]. Figure 2a presents the initial physical setup—a fixed rigid cylinder of diameter $D$ is immersed in a channel of width $4D$, with a constant, unbounded, background free stream of velocity $V_\infty\hat{\mathbf{j}}$. An elastic rod of length $L$ is initialized in the same fluid, with its upstream end clamped to the most downstream point of the cylinder. Additional geometric and parametric details can be found in Fig. 2 caption. We simulate this cylinder–rod system long enough after shedding vortices to eventually reach a dynamic, periodic state. This state is visualized via the vorticity contours at a particular time instance in Fig. 2b, where the rod is observed to flap and shed vortices downstream in the channel. We then quantitatively validate our solver by tracking the temporal variation of the rod tip’s horizontal and vertical displacements and compare it against Turek and Hron [98]. As seen from Fig. 2c, our results show close agreement with the benchmark [98].

We then present the grid convergence for this case, by tracking the amplitude of the rod tip’s horizontal and vertical oscillations (Fig. 2c), and computing the error norms with respect to the best resolved case. With CFL = 0.1, we vary the spatial resolution between 128 $\times$ 48 and 1024 $\times$ 384 (with 2048 $\times$ 768 as the best resolved case). As seen from Fig. 2d, we observe grid convergence between first and second order (least squares fit of 1.33 and 1.24 for vertical and horizontal displacements, respectively), consistent with our algorithm. Thus, the results of this section indicate the ability of our approach to successfully capture, in two dimensions, fluid–elastic solid and fluid–rigid solid interactions that are themselves coupled. For an additional, simpler validation of a single, two-dimensional elastic structure immersed in fluid, the reader is referred to the Appendix A.

5.2. Cross-flow past a three-dimensional elastic rod under gravity

Post validation in two dimensions. Here, we consider the case of cross-flow past a three-dimensional elastic rod suspended under gravity, characterized experimentally by Silva-Leon et al. [101]. Figure 3a presents the initial setup— a three-dimensional elastic rod is immersed in a constant, unbounded, background free stream of velocity $V_\infty\hat{\mathbf{j}}$ and is subject to the gravity field $-g\hat{\mathbf{k}}$. The rod is clamped at its top end, allowing it to deform in response to the surrounding cross-flow. Computational setup details can be found in Fig. 3. We simulate the system long enough until the rod deforms and attains a steady deformed equilibrium state, where the drag forces due to cross-flow are balanced by gravity and by the restoring internal stresses. This equilibrium state of the rod is visualized via volume rendered vorticity magnitude, for a particular cross-flow speed, in Fig. 3b. Next, for quantitative validation, we characterize the deformed state via $\theta$, the angle between the axis of the rod and the direction of the free stream flow. We then measure the deformation angle $\theta$ for different flow speeds (which correspond to different Reynolds numbers $Re$) and compare it against the experiments of Silva-Leon et al. [101]. As seen from Fig. 3c, our results agree with the benchmark [101].
Figure 2: Flapping of a two-dimensional elastic rod in the wake of a rigid cylinder. (a) Setup. The 2D rigid cylinder of diameter $D$ is initialized inside a 2D flow domain of size $(17.5D, 6.5625D)$, with its center at $(3D, 3.28125D)$. The 2D elastic rod with length $L = 3.5D$, $AR = 17.5$ and mass per unit length $m_e$ is initialized with one end fixed at $(3D, 3.28125D)$ and centerline at $y = 3.28125D$. Both these bodies are immersed in a viscous fluid with a background free-stream velocity $V_\infty$. Additionally, two no-slip walls are initialized at $y = 1.28125D$ and $y = 5.28125D$, respectively. The key non-dimensional parameters in this benchmark are $t V_\infty / D$, $Re := V_\infty D / \nu = 100$, $K_e = 0.02177$ and non-dimensional mass ratio $M_e = m_e/\rho L = 0.5714$. Other computational parameters are CFL = 0.1, $\alpha = -5 \times 10^3$ and $\beta = -20$. (b) Snapshot of the vorticity field (orange and blue indicate positive and negative vorticity respectively) for $t V_\infty / D = 77$, depicting the rod flapping motion and vortex shedding. (c) Comparison of temporal variation of the rod tip’s horizontal and vertical displacements against previous studies [98]. (d) Grid convergence. Error norms for the amplitudes of the rod tip’s vertical and horizontal displacements, plotted against grid spacing $h$. Further details can be found under the examples/2d_examples/VortexInducedFlagFlappingCase folder of [100].

Next, we present the grid convergence for this case, by tracking the deformation angle $\theta$ for a particular free stream speed ($Re = 29.139$), and computing the error norms against the best resolved case. With CFL = 0.1, we vary the spatial resolution between $96 \times 96 \times 24$ and $384 \times 384 \times 96$ (with $512 \times 128 \times 128$ as the best resolved case). As seen from Fig. 3d, our method presents convergence between first and second order (least square fit of 1.81), consistent with the proposed algorithm.

5.3. Self-propelled soft anguilliform swimmers

Following the successful validation of our technique against a three-dimensional static benchmark, we now demonstrate its ability to capture untethered, three-dimensional coupled dynamics, via the case of a self-propelled anguilliform swimmer immersed in a viscous fluid. We point out that this benchmark is the most demanding of those illustrated so far. This is because the validation depends on precisely capturing the dynamics of active, unconstrained slender bodies with varying cross-sections, while simultaneously resolving the interplay between body inertia, elasticity, viscous drag and flow inertia.

In Figure 4a we present the physical setup of the anguilliform swimmer, as described by Kern and Koumoutsakos [102]. The swimmer has length $L$, a cross-section that varies along the centerline coordinate $s \in [0, L]$, and is immersed in an unbounded fluid of the same density. The body geometry of the swimmer is described by the half-width $w(s)$ in two dimensions, and by the half-width $w(s)$ and half-height $h(s)$ in three dimensions (see Appendix B). In Kern and Koumoutsakos [102], the swimmer’s deformation (gait) is kinematically imposed, i.e. the influence of the fluid on the fish only affects rigid body translational and rotational velocity components. Here instead, the body of the fish is an elastic Cosserat rod, fully coupled to the flow and actuated using a traveling wave of torques and forces, modeling muscular actuation [74]. Here, torques and forces are back-calculated so as to reproduce the body lateral undulations of Kern and Koumoutsakos [102], in turn based on Carling et al. [16]. For this gait, the lateral displacement of the mid-line $\gamma(s, t)$, is defined as

\[
\gamma(s, t) = 0.125 s \left( \frac{1}{1.03125} \sin \left[ 2 \pi \left( \frac{s - L}{F} \right) \right] \right)
\]

where $t$ is the time and $T$ is the period. Additional computational details can be found in the caption of Fig. 4, while the procedure to determine the actuation parameters are reported in Appendix C.
Figure 3: Cross-flow past a three-dimensional elastic rod under gravity. (a) Setup. A 3D elastic rod with length $L$, $AR = 11$, $\rho = 826.45$ is initialized inside a 3D flow domain of size $(1.8L, 0.45L, 1.8L)$, with one end fixed at $(0.36L, 0.225L, 1.35L)$ and center-line along $x = 0.36L$. The rod is immersed in a viscous flow with a free-stream velocity $V_\infty$ and a gravitational field $-g\hat{z}$. The rod deflection angle $\theta$ is measured at equilibrium, where a balance between elastic, flow and gravitational forces is present. The key non-dimensional parameters in this benchmark are $K_\theta$, Froude number $Fr := gD/V_\infty^2$, stretch to bending ratio $K_s/K_b := EAL^2/(EI) = 62500$ and $Re := V_\infty D/V_f$. Other computational parameters are CFL=0.1, $\alpha = -2 \times 10^6$, and $\beta = -10^5$. (b) Snapshot of the steady state deformation of the elastic rod with volume rendering of vorticity magnitude (lighter and darker blue indicate lower and higher vorticity magnitude, respectively). The physical parameters for this simulation are set to $Re = 29.139$, $K_\theta = 0.0309$, and $Fr = 3.24 \times 10^{-3}$. The spatial resolution of the domain is set to $256 \times 256 \times 64$. (c) Comparison of rod deflection $\theta$ vs $Re$ against empirical (black line) [101] and experimental (red circles) [101] results. In order to be consistent with experiments, we vary $K_b$ as $K_b = 534.47/Re^2$ and $Fr$ as $Fr = 2.75/Re^2$. (d) Grid convergence. Error norm of rod deflection $\theta$ for $Re = 29.139$ plotted against grid spacing $h$. Further details can be found under the examples/3d_examples/FlowPastRodCase folder of [100].

In all simulations, the elastic swimmer starts from a straight, rest configuration and actuation forces and torques are gradually increased from zero to their designated values during the first cycle. We simulate the swimmers long enough until they reach a steady swimming average terminal speed. For a two-dimensional swimmer, the vorticity contours at a particular time instance are reported in Fig. 4h. In three dimensions, vortex rings are observed in the swimmer’s wake, visualized via the volume rendered Q-criterion at a particular time instance (Fig. 4c). The wakes of both 2D and 3D swimmers are found to be qualitatively consistent with benchmark literature [33, 102]. Next, we proceed with quantitatively validating our simulations by characterizing the swimmers’ motion via the velocity components of the center of mass (normalized by the velocity scale $L/T$) that are parallel ($V_{\parallel}$) and perpendicular ($V_{\perp}$) to the swimming direction. We track the temporal variation of the velocity components $V_{\parallel}$ and $V_{\perp}$ and compare them against previous studies [33, 102]. As seen from Fig. 4d for the three-dimensional case, our results show close agreement with the benchmarks. The two-dimensional quantitative comparison is instead reported in the Appendix D for brevity, albeit we note similar good agreement as for the 3D case. Finally, we present the grid convergence by tracking the terminal velocity components $V_{\parallel}$ and $V_{\perp}$ and computing the error norms against the best resolved case. For three-dimensional swimming, with CFL = 0.1, we vary the spatial resolution between $256 \times 128 \times 32$ and $512 \times 256 \times 64$ (with $768 \times 384 \times 96$ as the best resolved case). As seen from Fig. 4e, our method presents convergence between first and second order (least squares fit of 1.16 for $V_{\perp}$ and 2.1 for $V_{\parallel}$), again consistent with the spatial discretization employed in our algorithm. Overall the results of this Section validate our algorithm against a range of benchmarks, showing the accuracy and robustness of our numerical scheme and its implementation. These results are complemented by a convergence analysis which is found to be consistent with the employed discrete operators and across physical scenarios.

6. Multi-physics illustrations

Here, we further highlight the versatility of our solver by demonstrating a range of multi-physics applications. These include flow past a three-dimensional soft rod in an inertia dominated flow environment, viscous streaming induced by magnetic soft assemblies of filaments, and the locomotion of a self-propelling soft octopus. Before proceeding with the illustrations, we recall the key dimensionless parameters Eqs. 36 and 37 that govern the dynamics of these systems, along with their physical interpretations—Re (Reynolds number) defined as the ratio of inertial to viscous forces, Ca (Cauchy number) as the ratio of inertial to elastic forces, $\rho$ (density ratio) as the ratio of body inertia to flow inertia, $AR$ (aspect ratio) as the ratio of body length to diameter, and $K_b$ (bending stiffness) as the ratio of bending forces to inertial forces.

6.1. Three-dimensional flapping of an elastic rod

For this demonstration we consider the three-dimensional flapping of an elastic rod due to an inertia dominated flow characterized by $Re \sim O(10^5)$. Such inertia dominated flow regimes typically present fine spatio-temporal flow features, which necessitate
Figure 4: Self-propelled elastic anguilliform swimmers in two and three dimensions. (a) Setup. An elastic swimmer with length $L$, arc-length $s \in [0, L]$, half width $w(s)$ and half height $h(s)$ is immersed in an unbounded flow domain and is actuated using forces and torques that generate the periodic mid-line kinematics (with time-period $T$) described in Eq. 38. Here, we consider two- and three-dimensional variants of the swimmer, where $w(s)$ defines the geometry in 2D, while $w(s)$ and $h(s)$ define the geometry of the body in 3D (see Appendix B). In order to capture the effect of varying cross-sections on the swimmer dynamics, we modify $I_i$ (the second moment of the area around the $i$-director) accordingly along the rod. In 2D, the elastic swimmer is initialized with its head ($s = 0$) at $(5.45L, 0.68L)$ in a flow domain of size $(7L, 1.38L)$. In 3D, the body head is initialized at $(2.9 L, L, 0.25 L)$ in a flow domain of size $(4L, 2.0L, 0.5L)$. The elastic swimmer’s motion is characterized via the velocity components of the center of mass, that are parallel ($V_\parallel$) and perpendicular ($V_\perp$) to the swimming direction. These velocity components ($V_\parallel$ and $V_\perp$) are normalized by the velocity scale $L/T$. Key non-dimensional parameters in this benchmark are $t/T$, $\rho = 1$, actuation Reynolds number $Re := L^2 / (T \nu f) = 7142$, $K_b$ (set as 0.7044 in 2D and 0.1724 in 3D). Other computational parameters are CFL = 0.1, $\alpha$ (set as $-8 \times 10^4$ in 2D and $-3.2 \times 10^4$ in 3D) and $\beta$ (set as $-10$ in 2D and $-32$ in 3D). (b) Snapshot of the vorticity field (orange and blue indicate positive and negative vorticity, respectively) for $t/T = 10$. (c) Snapshot of the wake generated by the 3D soft swimmer with the volume rendered Q-criterion (orange and blue indicate vorticity and strain dominated areas, respectively) for $t/T = 6$. Inset on the top left corner shows the isometric view of the snapshot. The spatial resolution of the 3D domain for the visualized case is set as $768 \times 384 \times 96$. (d) Validation. Comparison of temporal variation of the elastic swimmer’s velocity components ($V_\parallel$ and $V_\perp$) against previous studies [102, 33] for the 3D swimmer. Validation for the 2D swimmer can be found in Fig. D.9a. (e) Grid convergence. Error norm of $V_\parallel$ plotted against grid spacing $h$ for the 3D swimmer. Convergence study of the 2D swimmer can be found in Fig. D.9b. Further details can be found under the examples/2d_examples/FishSwimmingCase folder for the 2D swimmer and the examples/3d_examples/FishSwimmingCase folder for the 3D swimmer of [100].
high-resolution simulations to accurately resolve the system dynamics. To enable high-resolution simulations, we implement a parallel MPI (Message Passing Interface) based version [103] of our algorithm that maps onto distributed computing architectures (for implementation details see Appendix E). Through this case we then demonstrate how our parallel implementation scales effectively across state-of-the-art distributed computing systems.

We begin, as shown in Fig. 5a (upper left inset), by initializing a three-dimensional elastic rod immersed in a constant, unbounded, background free stream of velocity \( V_\infty \). The rod is clamped at its upstream end and is free to rotate about its axis, thus allowing flapping and rotation in response to the surrounding flow. Additional geometric and parametric details can be found in Fig. 5. We simulate this system long enough until the rod sheds vortices, and eventually reaches either a periodic, quasi-periodic or chaotic state, depending on the rod elasticity \( K_b \). This state is visualized via the volume rendered Q-criterion at a particular time instance and \( K_b \), in Fig. 5a. To characterize the dynamical behavior of the system, we simulate it for varying rod stiffness \( K_b \) and track the coordinate of the rod’s free tip (marked in Fig. 5a, upper left inset) over time (lower right inset of Fig. 5a). In Fig. 5b, the Fourier analysis of the rod tip’s position history is presented for different values of \( K_b \), with the rod’s lateral displacement trajectories shown in the insets. When the rod is stiff (represented by higher values of \( K_b \), shown in the top two plots of Fig. 5b), its dynamics are periodic, characterized by a single peak in the Fourier analysis and a periodic limit cycle in the tip trajectory insets. As the rod becomes softer (i.e. \( K_b \) decreases), additional frequency modes emerge in the Fourier spectrum, represented by additional smaller peaks in the Fourier analysis (bottom left plot), and the system transitions from periodic to quasi-periodic dynamics. This is reflected as a quasi-cycle (an approximate cycle that does not repeat exactly) in the trajectory inset. Further reduction in the rod’s stiffness \( K_b \) results in a transition to chaotic dynamics, demonstrated by a continuous frequency spectrum in the Fourier space (bottom right plot), and chaotic non-repeating curves in the tip trajectory inset. Previous reports have indicated similar dynamical transitions in flapping two-dimensional filaments [104, 105], but no study has investigated three-dimensional counterparts. Although a complete analysis of this system lies beyond the scope of the current paper, this illustration showcases how our algorithm, combined with high-performance computing strategies, can be employed to explore uncharted systems requiring high fidelity simulations, with potential applications in energy harvesting, drag reduction, and related fields.

We conclude this investigation by highlighting how the parallel implementation of our algorithm scales effectively across multiple distributed computing systems. In Fig. 5c, we present the speedup values for both weak (scaling domain size with the number of processes) and strong (keeping the problem size fixed while varying the number of processes) scaling, as a function of the number of employed processes (for details see Fig. 5 caption). The results depicted in Fig. 5c demonstrate that the weak and strong scaling trends align closely with the ideal theoretical trend up to approximately one thousand employed processes, underscoring the efficiency of our algorithm, on top of its accuracy, versatility and relative simplicity.

6.2. Streaming flow generation via magnetized filament assemblies

Here we demonstrate our solver’s ability to capture second-order flow physics and rectification, via the example of viscous streaming generated by magnetized filament assemblies. Viscous streaming, an inertial phenomenon, refers to the steady, time-averaged flow that emerges when a immersed body undergoes small-amplitude oscillations within a viscous fluid. Due to its ability to reconfigure flow topology over short time and length scales, viscous streaming has found applications in multiple aspects of microfluidics, from particle manipulation and drug delivery to chemical mixing [108, 109, 110, 111, 112, 113, 114]. Despite comprehensive understanding of rigid body streaming [115, 116, 114], little is known when body elasticity is involved [117, 118, 119], and even less so when multiple soft streaming bodies are involved. Yet, the rectification of flow through systems comprised of several soft components is pertinent to various biological contexts, including flows produced by ciliated organisms like bacteria and larvae [12]. Cilia, slender hair-like structures located on the outer surface of the organism, drive these flows typically by deforming with a spatial phase difference, termed metachrony, so as to form wave-like motions [120]. Metachronal waves have then been hypothesized to generate, in some cases, streaming flows for feeding or locomotion [12]. Thus, motivated by the potential utility of filament arrays for streaming flow manipulation, we explore here the possible use of synthetic, magnetically driven cilia carpets, as detailed in the study by Gu et al. [8].

Before proceeding, we first describe the computational details involved in the modeling of magnetic filaments. We assume that each filament carries within its local frame a homogeneous and permanent magnetization, characterized by the magnetic dipole moment density \( m \). An external, driving magnetic field, characterized by magnetic flux density \( B \), is assumed to be unidirectional and spatially homogeneous (i.e. gradients are identically zero). Then, at any time instant, the magnetic field generates an external torque \( \tau \) along the rod, which is described by

\[
\tau = Q^T \mathbf{m} \delta V \times \mathbf{B}
\]  

(39)

where \( \delta V \) is the local differential volume and \( Q \) is the local frame of reference. For details concerning the numerical implementation of the magnetic torques, the reader is referred to [75].

Next, we adopt the physical setup shown in Fig. 6a, where a uniform, rectangular arrangement of cilia carpet with 32 × 16 rods of length \( L_c \) is immersed in an unbounded viscous fluid. Similar to Gu et al. [8], the cilia carpet is strongly magnetized with phases varying in a harmonic fashion along the \( x \) direction, with a non-dimensional wavelength \( \hat{\lambda} = \lambda / L_c \), where \( L_c \) is the carpet span along the \( x \) axis. Further details concerning the simulation setup, are reported in the caption of Fig. 6. The carpet is subjected to a
weak, sinusoidally rotating external magnetic field of angular frequency \( \omega \) in the \( x-z \) plane, which creates a metachronal wave that propagates along the \( x \) direction (see Appendix G). As can be seen in Fig. 6, this metachronal wave is indeed found to generate a steady streaming response in the surrounding fluid. Following classical streaming literature [121], we characterize this response via the Womersley number \( M \) defined as \( M := L_\epsilon \sqrt{\omega / \nu_f} \). The streaming flow response for the base case of \( M = 1 \) and \( \lambda = 1 \) is visualized via time-averaged Eulerian streamlines in Fig. 6b–d. This illustrates that a steady, centered vortex appears above the cilia carpet, consistent with the experiments of Gu et al. [8].

Next, we examine the effect of varying Womersley number \( (M) \) and metachronal wavelength \( \lambda \) on the streaming flow topology. We show in Fig. 6e–g) the time-averaged Eulerian streamlines for a fixed \( M = 1 \) and varying non-dimensional wavelength \( \lambda = 1 \), \( \lambda = 2 \), and \( \lambda = 4 \). The wavelength captures the synchronization among the cilia, with \( \lambda \to \infty \) indicating synchronous motion across all cilia. With increasing \( \lambda \), we observe the dissolution of the well-defined, centered vortex, consistent with observations made in
Figure 6: Streaming flow generation via magnetized filament assemblies. (a) Setup. A three-dimensional magnetic cilia carpet, composed of individual elastic rods of length $L_e$, and arranged in a rectangular uniform grid of 32 by 16 rods is initialized in a three-dimensional flow domain. The carpet has a span of $L_{cx}$ and $L_{cy}$ along the $x$ and $y$ directions, respectively, with a rod-to-rod spacing of $L_e/2$ along $x$ and $y$ axes. Each filament, with $AR = 5$ and $\rho = 2390$, is initialized with the lower end fixed, and the centerline pointing along the $z$ direction. The fluid domain is of size $(L_{cx} + 2L_e, L_{cy} + 2L_e, 5L_e)$ with the spatial resolution set as $384 \times 208 \times 110$ throughout. Similar to the experimental setting of Gu et al. [8], the cilia are strongly magnetized with magnetic dipole moment density $m$, where $m = m \sin \theta i + m \cos \theta k$. Here the magnetization angle $\theta$ varies spatially along the $x$ direction following an antiplectic pattern with non-dimensionalized wavelength $\hat{\lambda} = \lambda/L_{cx}$. Following setup, the cilia carpet is subjected to a weak, oscillatory magnetic field of strength $B$ that rotates sinusoidally in the $x-z$ plane given by $B = B \sin(\omega t) i + B \cos(\omega t) k$. The key non-dimensional parameters in this benchmark are the Womersley number $M := L_e \omega/\nu f$, magnetic Bond number $Bo := (mB \hat{A}^2)/(EI) = 3.0$ (ratio of magnetic to elastic forces), and the non-dimensional oscillation frequency of the magnetic field $\omega_0/\omega_0 = 0.16926$, where $\omega_0 = 3.5160/L_e^2 \sqrt{EI/(\rho A)}$ is the natural frequency of the filament’s first bending mode. Other computational parameters are CFL=0.1, $\alpha = -2 \times 10^4$, and $\beta = -10$. The rotating magnetic field creates a metachronal wave that propagates in the $x$-direction along the cilia carpet, ultimately generating streaming flow on top of the carpet. Streaming flow topology visualized via the Eulerian time-averaged streamlines for $M = 1$ and $\hat{\lambda} = 1$ (orange and blue represent positive and negative $x$-velocities, respectively) in (b) an isometric view (c) $y-z$ plane and (d) $x-z$ plane, respectively. (e-g) Flow topology variation with $M = 1$ and increasing wavelength $\hat{\lambda}$, with (e) $\hat{\lambda} = 1$ (f) $\hat{\lambda} = 2$ and (g) $\hat{\lambda} = 4$. Streaming eddies visualized via particle trajectories from equivalent experiments of Gu et al. [8] (with one-to-one $\hat{\lambda}$ values) are included as yellow inset boxes for reference. (h-j) Flow topology variation with $\hat{\lambda} = 1$ and increasing $M$, with (h) $M = 1$ (i) $M = 2$ and (j) $M = 3$. (k-l) Non-dimensional fluid transport flux $\phi/\omega L_e^2$ (Eq. 40) along $x$ axis (measured at the carpet midplane $x = L_e/2$) plotted against $z$ coordinate, for varying ($k$) $\hat{\lambda}$ and (l) $M$. Further details can be found under the examples/3d_examples/MagneticCiliaCarpetCase folder of [100].
developed in [3]. In the virtual muscle model, the contraction of virtual muscles generates forces.

study, we represent each octopus arm as a Cosserat rod, and actuate it using a virtual muscle model [68] based on the understanding of non-linear one-dimensional muscle contraction into complex three-dimensional motions, as observed in actual octopuses. In this section, we describe the physical setup of the octopus swimming experiment, as shown in Fig. 7a. This computational experiment involves an eight armed octopus immersed in a three-dimensional unbounded viscous flow. Each soft arm is modeled as a tapered Cosserat rod of length $L$. The arms and the rigid spherical head are connected to the elastic body through spring-damper boundary conditions. The arms are actuated by contracting their (virtual) longitudinal muscles, following the activation profile specified in Eq. 42. Additional details about the simulations are provided in the caption of Fig. 7. Two octopuses are considered in the experiment: an adult and a juvenile. For the adult octopus, the key non-dimensional parameters $K_v$, actuation Reynolds number $R_{e_v} := LD/(T
u_f)$, $AR$, and the taper ratio are based on measurements from experimental studies [71, 126, 1]. The juvenile octopus is a geometrically scaled down version of the adult octopus, roughly 10$\times$ smaller in size [127]. We then assume that the juvenile octopus has the same activation function $a(s, t)$, $E$, $AR$, and taper ratio as that of the adult, which results in the adult and the juvenile

$\phi(z) = \int_{0}^{L_{cv}} v_y(y, z) \, dy$ (40)

where $L_{cv}$ is the carpet span along the $y$ axis, and $v_y(y, z)$ is the flow speed along $x$ axis at $x = L_{cv}/2$. We then track $\phi$ as a function of distance above the carpet ($z$) with varying wavelength $\lambda$ (Fig. 6k) and Womersley number $M$ (Fig. 6l). Consistent with findings of Gu et al. [8], the flow strength decreases with increasing $\lambda$, implying diminishing streaming effects. In contrast, varying $M$ (keeping $\lambda = 1$ fixed) produces stronger non-linearities, and a local maximum of the transport flux near $M = 2$. Overall, the above-demonstrated generation and tunability of streaming flows via artificial filament arrays hints at their potential in microfluidic applications, from particle manipulation and transport to chemical mixing [122], while underscoring the ability of our solver to capture such complex second order effects, in agreement with experiments.

6.3. Octopus swimming

Here we demonstrate the versatility of our proposed algorithm by incorporating features of bio-inspired, soft robotic locomotion, namely endogenous biomechanical muscle models, muscular actuations, and soft body assembly, via the case of a self-propelling octopus. Muscular hydrostats, such as octopus arms, are unique in that they lack bones altogether, relying solely on intricate weavings of muscle fibers within their architecture. These structures confer unparalleled coordination and reconfiguration abilities, endowing these animals with remarkable manipulation skills. These distinctive features have drawn the attention of biologists and soft roboticists to model and characterize the octopus’ underwater manipulation and locomotion strategies [123, 124]. However, these studies employ simplistic drag models or neglect elasticity and the arm’s internal structure, and thus fail short of resolving the two-way coupling with the flow environment, as well as the effect of the muscular structure, on the resultant dynamics.

Recently, bio-realistic models based on Cosserat rods have been developed to replicate the heterogeneous organization and the bio-mechanics of the octopus arms [3]. In particular, these models capture the role of the muscular architecture in translating non-linear one-dimensional muscle contraction into complex three-dimensional motions, as observed in actual octopuses. In this study, we represent each octopus arm as a Cosserat rod, and actuate it using a virtual muscle model [68] based on the understanding developed in [3]. In the virtual muscle model, the contraction of virtual muscles generates forces $f$ and torques $Qe$ on the arm center-line [68]. These contractile forces are modeled as

$f_m := a\sigma_{\max}A_mh(l_m)$ (41)

where $a$ denotes the muscle activation level, $\sigma_{\max} := 10E$ represents the maximum muscle stress, $E$ is Young’s modulus of the arm, $A_m := \pi r_m^2$ denotes the muscle cross-sectional area, and $h(l_m) \in [0, 1]$ is the normalized force-length relationship obtained by fitting experimental octopus data [125], and $l_m$ is the length of the muscle [68]. This framework allows us to integrate experimentally obtained biomechanical data through the term $h(l_m)$, while investigating different neuro-muscular activations via the term $a$. Next, in order to generate the swimming stroke motion, we use a travelling wave activation profile similar to the one presented in Kazakidi et al. [123].

$$a(s, t) = a_{\max} \left( \frac{\cos(\pi s) + 1}{2} \right)^4 \sin^2 \left(2\pi w \frac{s}{L} - \pi \frac{t}{T} \right)$$ (42)

where $a_{\max} = 0.2$ is the maximum activation magnitude, $T$ is the period, and $w = 0.05$ is the wave number [123]. In all our simulations, the octopus starts from a rest configuration and the muscle activation $a_{\max}$ is gradually increased from zero to the designated value during the first period.

In the following, we describe the physical setup of the octopus swimming experiment, as shown in Fig. 7a. This computational experiment involves an eight armed octopus immersed in a three-dimensional unbounded viscous flow. Each soft arm is modeled as a tapered Cosserat rod of length $L$. The arms and the rigid spherical head are connected to the elastic body through spring-damper boundary conditions. The arms are actuated by contracting their (virtual) longitudinal muscles, following the activation profile specified in Eq. 42. Additional details about the simulations are provided in the caption of Fig. 7. Two octopuses are considered in the experiment: an adult and a juvenile. For the adult octopus, the key non-dimensional parameters $K_v$, actuation Reynolds number $R_{e_v} := LD/(T
u_f)$, $AR$, and the taper ratio are based on measurements from experimental studies [71, 126, 1]. The juvenile octopus is a geometrically scaled down version of the adult octopus, roughly 10$\times$ smaller in size [127]. We then assume that the juvenile octopus has the same activation function $a(s, t)$, $E$, $AR$, and taper ratio as that of the adult, which results in the adult and the juvenile...
Figure 7: Octopus swimming. (a) Setup. An eight armed octopus with soft, tapered arms is immersed in a three-dimensional, unbounded, viscous flow. Each soft arm is modeled as a Cosserat rod with length $L$, arc-length $s \in [0, L]$ at base $r(0) = r_b$ and at tip $r(L) = r_t$), taper ratio $r_b/r_t = 12$, base AR: $L/(2r_b) = 4.16$, and is actuated via muscular forces and torques. These forces and torques are computed using the virtual muscle model, where two virtual longitudinal muscles (depicted as pink rods) upon contraction generate external forces $f$ and external torques $Qe$ on the arm centerline $x(s, t)$ [68]. The longitudinal muscles of radius $r_m(s, t) := 0.25 r(s, t)$ are attached at $x_m(s, t) = x(s, t) \pm 0.625 r(s, t) d_1(s, t)$. All 8 arms are actuated by contracting the inside longitudinal muscle, marked in (a), via a traveling wave activation profile of time-period $T$ described in Eq. 42. The octopus head is modeled as a rigid sphere with diameter $D = 0.12 L$ and center initialized at $(3.15 L, 1.25 L, 1.25 L)$. The eight arms and the head of the octopus are connected via the octopus body, which is modeled as a soft tapered Cosserat rod. The fluid domain is of size $(5L, 2.5L, 2.5L)$ with the spatial resolution of the domain set as $392 \times 196 \times 196$ throughout. We consider two different flow regimes in this study, specifically regimes in which juvenile and adult octopus are observed. Key non-dimensional parameters in this benchmark are $t/T$, $\rho = 1.02$, actuation Reynolds number $Re_a := LD/(\nu f)$ (set as 48 for juvenile and 4800 for adult), $K_b$ (set as 8.3 for juvenile and 0.083 for adult), and the swimming Reynolds number $Re_{sw} := v_f D/\nu f$, which emerges as $O(10)$ for juvenile and $O(10^2)$ for adult octopus, based on the observed swimming speeds shown in (f). Other computational parameters are CFL = 0.1, $\alpha = -2 \times 10^4$ and $\beta = -10$. Snapshots of the volume rendered vorticity magnitude field (white and blue indicates lower and higher vorticity, respectively) around the juvenile octopus for (b) power stroke at $t/T \approx 2.4$ and (c) recovery stroke at $t/T \approx 2.9$. Snapshots of the wake generated by the adult octopus, with the volume rendered vorticity magnitude field for (d) power stroke at $t/T \approx 4.6$ and (e) recovery stroke at $t/T \approx 5.1$. Temporal variation of the octopus head’s swimming (f) velocity and (g) position in the adult and juvenile flow regimes. Further details can be found under the examples/3d_examples/OctopusSwimming folder of [100].

Octopus operating in different flow regimes, characterized by distinct values of the arm stiffness $K_b$ and the actuation Reynolds number $Re_a$.

We begin with visualizing the flow generated by the adult and juvenile octopuses while swimming across the flow domain. Figure 7b presents the power stroke of the juvenile octopus swimming, (visualized via the volume-rendered vorticity magnitude field), where the eight arms generate an octuplet of vortex rings, which are then shed during the recovery stroke shown in Fig. 7c, resulting in forward propulsion of the octopus. For the adult case instead, an increase in the actuation Reynolds number $Re_a$ causes the vortex rings to break down into a jet, as seen from the power and recovery strokes shown in Fig. 7d and e, respectively. Next, we
present the temporal variation of the octopus’ swimming velocity and its head’s position, in both adult and juvenile flow regimes, in Fig. 7f and g. The non-dimensional swimming speed of the juvenile octopus is found to be larger than that of the adult, which is consistent with previous observations of octopuses of different sizes [24]. Overall, this demonstration provides valuable insights into the swimming behavior of octopuses and can be leveraged in the future to develop efficient soft robotic systems inspired by the locomotion strategies of these animals.

7. Conclusion

In summary, we have presented a framework that integrates vortex methods and Cosserat rod theory to capture the two-way flow-structure interaction between multiple heterogeneous, slender soft bodies immersed in a viscous fluid. Our approach employs the penalty immersed boundary method to couple the fluid and the immersed soft structure, while accurately accounting for the mechanics of soft bodies via Cosserat rod theory. The algorithm we have developed is straightforward and concise, while still being accurate and robust, as demonstrated by rigorous benchmarking and convergence analysis against a wide range of experimental and numerical tests. Through various multifaceted illustrations (which themselves may serve as detailed benchmarks for future studies), we further demonstrate our solver’s versatility, applicability and robustness across multiphysics scenarios, boundary conditions, constitutive and actuation models. Particularly, the accurate resolution of a wide range of physics encompassing muscle actuation, multi-body connection, self-propulsion, streaming and magnetism in a scalable fashion, demonstrates the utility of our method in a wide range of applications, from bio-locomotion and soft robotics to microfluidics. While our algorithm has shown to scale effectively across distributed computing systems, the use of convenient grid-based operators for solving the flow equations renders the solver portable, in a scalable fashion, to advanced parallel architectures such as GPUs. However, beyond porting the algorithm to GPUs, opportunities for algorithmic improvement also exist. First, instead of the current mollification approach, modified higher-order stencils can be utilized to resolve the physics near the interface [128, 129, 130] to improve grid convergence now limited between first and second order. Second, an unconditionally stable temporal integration algorithm can be applied to solve for the diffusion step [131] to achieve faster time-to-solutions in viscous flow regimes. Finally, a time step restriction based on the shear wave speed inside the elastic solid can be bypassed through the use of an implicit time stepper [96] or a local time-stepping technique [36]. All the above directions are avenues of future work.

8. CRediT authorship contribution statement

Arman Tekinalp: Conceptualization, Data curation, Formal Analysis, Investigation, Methodology, Software, Validation, Visualization, Writing – original draft, Writing – review & editing. Yashraj Bhosale: Conceptualization, Data curation, Formal Analysis, Investigation, Methodology, Software, Validation, Visualization, Writing – original draft, Writing – review & editing. Songyuan Cui: Conceptualization, Data curation, Formal Analysis, Investigation, Methodology, Software, Validation, Visualization, Writing – original draft, Writing – review & editing. Fan Kiat Chan: Conceptualization, Data curation, Formal Analysis, Investigation, Methodology, Software, Validation, Visualization, Writing – original draft, Writing – review & editing. Mattia Gazzola: Conceptualization, Data curation, Formal Analysis, Funding acquisition, Investigation, Methodology, Project administration, Resources, Software, Supervision, Validation, Visualization, Writing – original draft, Writing – review & editing.

9. Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

10. Acknowledgements

This study was jointly funded by ONR MURI N00014-19-1-2373 (M.G.), ONR N00014-22-1-2569 (M.G.), NSF EFRI C3 SoRo #1830881 (M.G.), NSF CAREER #1846752 (M.G.), and with computational support provided by the Bridges2 supercomputer at the Pittsburgh Supercomputing Center through allocation TG-MCB190004 from the Extreme Science and Engineering Discovery Environment (XSEDE; NSF grant ACI-1548562).

Appendix A. Validation of flow past a two-dimensional elastic rod under gravity

We test our method for the ability to capture the dynamics of two-dimensional elastic structures immersed in viscous fluids. To do so we adopt the benchmark of flow past a two-dimensional elastic rod suspended under gravity, commonly used for validation in flag flapping studies [84, 132]. Figure A.8a presents the initial setup—a two-dimensional elastic rod of length \( L \) is immersed in a constant, unbounded, background free stream of velocity \( V_\infty \hat{i} \) and under the gravitational field \( g \hat{j} \). The rod is clamped at the
Figure A.8: Flow past a two-dimensional elastic rod under gravity. (a) Setup. The 2D elastic rod with length $L$, $AR = 50$ and mass per unit length $m_e$, is initialized inside a 2D flow domain of size $(6L, 3L)$, with one end fixed at $(L, 1.503L)$ and centerline at $y = 1.503L$. The rod is immersed in a viscous fluid, under a gravity field $g$ and a background free-stream velocity $V_\infty \hat{\mathbf{i}}$. The key non-dimensional parameters in this benchmark are convective time unit $t_{V_\infty}/L$, $Re = V_\infty L/\nu_f = 200$, $K_b = 0.0015$, non-dimensional mass ratio $M_e = m_e/\rho_f L = 1.5$ and Froude number $Fr = gL/V_\infty^2 = 0.5$. Other computational parameters are CFL = 0.1, $\alpha = -8 \times 10^4$ and $\beta = 30$. (b) Snapshot of the vorticity field (orange and blue indicate positive and negative vorticity respectively) for $t_{V_\infty}/L \approx 70$, depicting the rod flapping motion and vortex shedding. (c) Comparison of temporal variation of rod tip’s vertical displacement against previous studies [132]. (d) Grid convergence. Error norm of rod tip’s vertical displacement amplitude plotted against grid spacing $h$. Further details can be found under examples/2d_examples/FlowPastRodCase folder of [100].

upstream end, allowing the rod to flap freely in response to the surrounding flow and gravity. Computational setup details can be found in Fig. A.8.

We simulate the system long enough to observe von Kármán vortices, and eventually the system reaches a dynamically quasi-steady, periodic state. This state is visualized via the vorticity contours at a particular time instance, as shown in Fig. A.8b. Next, to validate our method, we track the vertical coordinate of the rod’s free tip and compare it against the periodic tip response from previous studies [132]. As seen from Fig. A.8c, our results show close quantitative agreement with the previous study [132].

We then present the grid convergence for this benchmark, by tracking the amplitude of the rod tip’s cross-stream oscillations (Fig. A.8c), and computing the error norms with respect to the best resolved case. We fix CFL = 0.1 and vary the spatial resolution between $256 \times 128$ and $1024 \times 512$ (with $2048 \times 1024$ as the best resolved case). As seen from Fig. A.8d, our method presents grid convergence between first and second order (least squares fit of 2.1), consistent with the spatial discretization of the operators.

Appendix B. Geometrical details of anguilliform swimmers

Here we describe the geometry used for the anguilliform swimmers shown in Section 5.3 of the main text. The swimmer is modeled as an elastic rod with varying cross-section dependent on centerline coordinate $s$. The two-dimensional swimmer cross-section is characterized using the half width $w(s)$, while the three-dimensional swimmer cross-section is an ellipse with half width $w(s)$ and half height $h(s)$. Following Gazzola et al. [33], the half width $w(s)$ is defined as

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

(B.1)

where $L$ is the body length, $w_h = 0.04L$, $s_t = 0.95L$ and $w_t = 0.01L$. In Gazzola et al. [33], the thickness reduction from head to tail ($s_h \leq s < s_t$) is linear instead of quadratic for the 2D case, therefore we implemented the same modification here. For 3D swimmers, we define the half height $h(s)$ following Gazzola et al. [33]

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

(B.2)
where \( a = 0.51L \) and \( b = 0.08L \).

**Appendix C. Muscular actuation of anguilliform swimmer**

We start by creating a virtual rod that is identical to the elastic swimmer, with the former following precisely the kinematics Eq. 38 suggested by Carling et al. [16]. Here, we update the directors of the virtual rod under the zero-shear assumption (\( d_3 \equiv t \)). Next, based on the updated positions and directors we compute the corresponding internal torques \( \tau_v \) within the virtual rod via Eq. 5. The internal torques of the virtual rod are in turn translated into muscular torques \( \tau_m \) acting upon the elastic swimmer via frame transformation

\[
\tau_m(s, t) = Q^T_v(s, t)\tau_v(s, t)
\]

(C.1)

where \( Q^T_v \) is the transpose of the virtual rod directors. Finally, since the elastic swimmer is free to shear (and therefore may deform slightly different from the virtual rod), local correcting forces are computed, proportionally to the deviation between the elastic fish and the virtual rod node positions

\[
f_m(s, t) = k (y_v(s, t) - y_e(s, t))
\]

(C.2)

where \( k := EA \) is the spring constant, \( y_v \) and \( y_e \) are lateral displacements of the virtual rod and the elastic fish, respectively.

**Appendix D. Two-dimensional anguilliform swimmer validation**

Here we present the validation and convergence analysis for the two-dimensional anguilliform swimmer shown in Section 5.3. For validation, we track the temporal variation of the velocity components \( V_\parallel \) and \( V_\perp \) of the 2D swimmer and compare them against previous studies [33, 102]. As seen from Fig. D.9a, our results show close agreement with the benchmarks in two dimensions. We next present the grid convergence for this case by tracking \( V_\parallel \) and \( V_\perp \) and computing the error norms against the best resolved case. For two-dimensional swimming, with CFL = 0.1, we vary the spatial resolution between 1024 × 204 and 2048 × 409 (with 3072 × 614 being the best resolved case). As shown in Fig. D.9b, our technique presents close to second order convergence (least squares fit of 1.82 for \( V_\perp \) and 2.2 for \( V_\parallel \), respectively).

**Appendix E. Algorithm mapping on distributed computing architectures for large-scale simulations**

In this section, we provide additional details surrounding the mapping of our algorithm onto distributed computing architectures. In the following, we describe how the flow and immersed body domains are decomposed and communicated across different computing processes.

**Appendix E.1. Mapping flow domain**

In the employed numerical method (Algorithm 1), the fluid domain, and hence the flow fields, are defined on structured or uniform Cartesian meshes. This in turn results in rectangular (2D) or cuboidal (3D) physical and computational domains.
Appendix E.1.1. Domain decomposition

In order to ensure balanced workloads between processors, we employ the Cartesian topology available via the Message Passing Interface (MPI) \[133, 134, 135, 136\] for decomposing the fluid domain into equally-sized subdomains that are mapped onto different processors (Fig. E.10a). Given the assigned subdomain, each processor then performs differential operations with mesh-based kernels (i.e. numerical differential stencils). These local operations typically result in computational costs that scale linearly with the number of grid points. However, in order to obtain the correct solution on the boundaries of these subdomains, communication with the adjacent subdomain becomes necessary to ensure the local operations are carried out with the correct boundary data.

Figure E.10: (a) Domain decomposition for a representative 2D problem. We focus our example here on the purple central subdomain and use a nine-point stencil to motivate the need for communication across subdomains residing on different processes. We introduce ghost cells for storing data copied from neighboring subdomains in order to ensure the correct computation of differential operators. (b) Ghost cell communication pattern for complete data exchange: receiving (left) incoming and sending (right) outgoing data.

Appendix E.1.2. Communication

In order to exchange the boundary data between neighboring subdomains, we extend the subdomain by \(n_s\) mesh points (grey regions of subdomains in Fig. E.10), where \(n_s\) is chosen based on the maximum kernel support required by the numerical stencils employed in Algorithm 1. For example, a second-order central finite difference stencil requires one adjacent mesh point for its
kernel operation and thus has a kernel support of $n_s = 1$. This extended layer of mesh points is commonly known as ghost/halo points (Fig. E.10a), and stores updated copies of the data from the corresponding neighboring subdomains for consistent kernel operation on the boundaries.

As an illustrative example, we show in Fig. E.10 the communication pattern involved for a 2D problem decomposed and distributed equally into 9 different processors. We start by drawing our attention to the communication patterns involved for the representative interior (purple) subdomain in Fig. E.10, and note that the communication pattern generically holds for other subdomains as well. The orange and blue regions indicate the subdomains directly and diagonally adjacent to the interior subdomain, respectively. We then consider a representative nine-point stencil kernel (pink box in Fig. E.10a) that operates on the structured mesh. As can be seen, in the extreme case of kernel operation at corner cells of our focal subdomain, the stencil involves information from a total of three neighboring processes. This means that for the operator to be computed correctly on that corner cell, communication needs to happen between the purple subdomain, two directly (orange) and one diagonally (blue) adjacent subdomains. Figure E.10b illustrates the data flow for updating the ghost region that ensures correct computation of kernels with $n_s > 0$ across all subdomains.

Appendix E.2. Mapping immersed body domain

The immersed structures in the employed numerical method (Algorithm 1) are defined on unstructured meshes. In order to resolve the structural dynamics of the slender immersed bodies, we leverage pyelastica [95], a simple and convenient library for simulating assemblies of slender, one-dimensional structures using Cosserat Rod theory. However, the package currently does not support distributed computation. While it is reasonable to assign the Cosserat rod simulation on a single master process, information from the updated structure will eventually be needed in corresponding flow subdomains during the flow–structure interaction step in Algorithm 1. For the interaction to take place correctly, our algorithm requires the position, velocity and forces defined on the unstructured mesh native to the immersed body. In the following, we describe how the immersed structure is distributed to and retrieved from different processes throughout the interaction step in our method.

Appendix E.2.1. Domain decomposition

Given the dynamic, unstructured nature of the moving immersed body, the master process will need to decompose the body’s discretization mesh based on the position of the body’s grid points relative to the fluid subdomain, and remap them to the corresponding fluid subdomain at every time step. This is done in a straightforward fashion based on the Cartesian topology established by MPI [135, 136, 134, 137] for domain decomposition of the structured fluid domain (Appendix E.1).

Appendix E.2.2. Communication

Once the decomposition map is computed, relevant fields residing on the unstructured mesh can be communicated to and from their corresponding subdomain for consistent resolution of flow–structure interaction. We summarize the communication pattern for a simple unstructured body mesh in Fig. E.11. As described above, the master process first identifies the corresponding flow subdomain to which every single point of the body grid resides in. This information is then stored in a mapping containing the address of destination processes for each body grid point. Based on this mapping, the mesh points are communicated to the

Figure E.11: Communication pattern involved for connecting structured and unstructured grids, thus allowing extension of flow simulator to interact with external structural dynamics packages such as pyelastica [95] for Cosserat rod simulations.
corresponding flow subdomain, followed by the flow–structure interaction step of Algorithm 1. Once the interaction is complete, relevant updated fields such as forcing on the structure are returned from the subdomains back to the master process accordingly based on the mapping established earlier. With the forcing information updated on the master process, pyelastica [95] then performs temporal integration to advance the rod to the next time step, resulting in an updated position and velocity of the rod’s unstructured mesh. This process is then repeated at every time step, thus ensuring a continuous and consistent workflow for the fluid–structure interaction.

Appendix F. Validation for flow past a rigid sphere

Here we test our algorithm for its ability to capture flow–structure interaction physics for immersed rigid bodies, by considering the classical case of flow past a fixed rigid sphere. Figure F.12a presents the initial setup—a fixed rigid sphere of diameter $D$ is immersed in a constant, unbounded, background free stream of velocity $V_i$ (details in Fig. F.12 caption). We simulate the system long enough until it reaches either a steady state (observed for $Re < 200$) or a quasi-steady periodic state (observed for $200 < Re < 1000$). For low $Re$ ($< 200$), a steady recirculation zone (eddy) attaches to the downstream end of the sphere, while for moderate $Re$ ($200 < Re < 1000$), periodic shedding of vortices is observed in the wake of the sphere. The periodic vortex shedding state is visualized for a particular $Re$ via the volume rendered Q-criterion in Fig. F.12b. As seen from Fig. F.12b, hairpin vortices are observed in the wake of the sphere, consistent with previous studies [138]. Next, for validation, we vary $Re$ and track two commonly used non-dimensional diagnostics: the length of the attached recirculation eddy $s/D$ (only for $Re < 200$) and the mean drag coefficient $C_D$ for the sphere. Figure F.12c-d present the variations of recirculation eddy lengths and mean drag coefficients with $Re$, with comparison against previous studies [138, 139, 140, 141]. As seen from Fig. F.12e, our results show close agreement with previous studies.

Next, we present the grid convergence for this case, by tracking the mean drag coefficient $C_D$ for $Re = 100$, and computing the error norms against the best resolved case. With $CFL = 0.1$, we vary the spatial resolution between $32 \times 16 \times 16$ and $256 \times 128 \times 128$ (with $512 \times 256 \times 256$ as the best resolved case). As seen from Fig. F.12e, our method exhibits spatial convergence between first and second order (least square fit of 1.53), consistent with the numerical discretization of our solver.

Appendix G. Animations of simulated cases

We have included animated videos of all cases simulated in the paper, which can be accessed through the following link.

Appendix H. Software

The implementation of the algorithm in this study and corresponding cases can be found here. For the MPI based implementation the reader is referred to this link.

References

Figure F.12: Flow past a rigid sphere. (a) Setup. The rigid sphere with diameter $D$ is initialized inside a 3D flow domain of size $(5D, 2.5D, 2.5D)$, and with its center fixed at $(1.25D, 1.25D, 1.25D)$. The sphere is immersed in a viscous flow with a free-stream velocity $V_\infty$. The key non-dimensional parameters in this benchmark is $Re = V_\infty D/\nu$. Other computational parameters are $CFL = 0.1$, $\alpha = -1.5 \times 10^3$ and $\beta = -87.5$. (b) Snapshot of flow past sphere at $Re = 400$ with volumetric rendering of Q-criterion (orange and blue indicate vorticity and strain dominated areas, respectively). To capture the far-field flow features, we run this particular simulation with a flow domain of size $10D \times 2.1D \times 2.1D$, spatial resolution $384 \times 160 \times 160$, and sphere center fixed at $(2.15D, 2.1D, 2.1D)$. Formation of hairpin vortex in the wake is captured, consistent with previous studies [138]. (c) Comparison of non-dimensional attached eddy length ($s/\delta$) vs $Re$ against experiments (red circles) [139] and previous simulations (black line with circle markers) [138]. (d) Comparison of mean drag coefficient ($C_D$) vs $Re$ against empirical (black line) [141] and experiments (red circles) [140]. (e) Convergence: Error norms of sphere under examples/3d_examples/FlowPastSphereCase folder of [100].


