



Contents lists available at ScienceDirect

# Communications in Nonlinear Science and Numerical Simulation

journal homepage: [www.elsevier.com/locate/cnsns](http://www.elsevier.com/locate/cnsns)

Research paper

## A finite element method for simulating soft active non-shearable rods immersed in generalized Newtonian fluids



Roberto Federico Ausas<sup>a</sup>, Cristian Guillermo Gebhardt<sup>b,\*</sup>,  
Gustavo Carlos Buscaglia<sup>a</sup>

<sup>a</sup>Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, Av. Trabalhador São-carlense, 400, 13566-590, São Carlos, SP, Brazil

<sup>b</sup>Geophysical Institute (GFI) and Bergen Offshore Wind Centre (BOW), University of Bergen, Allégaten 70, 5007 Bergen, Norway

### ARTICLE INFO

#### Article history:

Received 17 August 2021

Received in revised form 15 December 2021

Accepted 19 December 2021

Available online 27 December 2021

#### Keywords:

One-dimensional solids with finite

thickness and finite length

Generalized Newtonian fluids

Fluid–structure interaction

Finite element method

Soft active bio-matter realm

Freely available Firedrake implementation

### ABSTRACT

We propose a finite element method for simulating one-dimensional solid models with finite thickness and finite length that move and experience large deformations while immersed in generalized Newtonian fluids. The method is oriented towards applications involving microscopic devices or organisms in the soft-bio-matter realm. By considering that the strain energy of the solid may explicitly depend on time, we incorporate a mechanism for active response. The solids are modeled as Cosserat rods, a detailed formulation being provided for the planar non-shearable case. The discretization adopts one-dimensional Hermite elements for the rod and two-dimensional low-order Lagrange elements for the fluid's velocity and pressure. The fluid mesh is boundary-fitted, with remeshing at each time step. Several time marching schemes are studied, of which a semi-implicit scheme emerges as most effective. The method is demonstrated in very challenging examples: the roll-up of a rod to circular shape and later sudden release, the interaction of a soft rod with a fluid jet and the active self-locomotion of a sperm-like rod. The article includes a detailed description of a code that implements the method in the Firedrake library.

© 2021 The Author(s). Published by Elsevier B.V. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

One-dimensional solid models (strings, cables, trusses, bars, beams, filaments, rods, etc.) are widely used in macroscopic structural analysis. Well established procedures exist, together with academic and commercial codes, that approximate these models and provide meaningful solutions to the analyst. They are also useful in modeling microscopic phenomena, both in the solid realm (e.g., fiber-reinforced composites [1,2]) and in the soft-bio-matter realm (e.g., human hair [3,4], climbing plants [5], catheters and stents [6], DNA [7,8], cilia and flagella [9] and further biological applications [10]) that is the focus of this contribution.

Simple elongated organisms, or the appendices (flagella, cilia) of more complex ones, are frequently described as one-dimensional entities. This is very convenient, since the model's unknowns are reduced, for each time instant, to a set of functions defined on some real interval  $[0, S]$ . Initial conditions and control actions are in this way much simpler to build and analyze. Though much of the theory and methods of macroscopic solid mechanics apply straightforwardly, some challenges are raised, or emphasized, by the microscopic realm.

\* Corresponding author.

E-mail address: [cristian.gebhardt@uib.no](mailto:cristian.gebhardt@uib.no) (C.G. Gebhardt).

Firstly, the usual boundary conditions that attach or clamp the solid body to a rigid immovable boundary are in general absent. The body is often floating in the surrounding fluid, and if the action of the fluid is neglected, *or if the forces that the fluid exerts on the solid are frozen in time* (as done in staggered FSI algorithms), the physical/mathematical problem for the solid becomes undetermined (only computable up to an arbitrary translation and rotation). This difficulty is typical of sedimenting and swimming bodies, even if they are zero-dimensional (i.e., a set of rigid particles, connected or not). Adequate treatments of the translation and rotation degrees of freedom (dofs) of bodies immersed in fluid have been presented in [11–13] among others.

A second challenging aspect of modeling one-dimensional bodies in soft-bio matter is the very large displacements and deformations that are usually involved. The strong nonlinearities that appear in the differential geometry of curves comes into play and cannot be simplified/linearized. This makes that only quite sophisticated formulations of Computational Solid Mechanics (CSM) can be adopted. Later on we describe one of such formulations, which corresponds to a planar nonlinear non-shearable rod, but it is certainly not the only possible one. The reader is referred to [14,15] for further details in the context of similar formulations in the three-dimensional setting.

Another peculiarity of soft-bio-matter problems is that, as happens also in aerodynamics, the effect of the surrounding fluid on the solid bodies is much more consequential than simply energy dissipation. Swimmers, in particular, rely on the complex response of the fluid for their propulsion and guidance [16–20]. A quite accurate fluid model is thus necessary to approximate the motion of microscopic organisms. In many cases, this means solving the full Navier–Stokes equations or variants thereof (in the case, for example, of non-Newtonian fluid’s rheology [21,22]), since simplified models such as Resistive Force Theory may yield misleading conclusions [23].

In CSM it is customary to denote as Fluid–Structure Interaction (FSI) problems those that require the coupled solution of the fluid’s and solid’s partial differential equations. Thus, one of the challenges of soft-bio problems is that they are often strongly coupled FSI problems. The literature on FSI is quite vast and many previous works introduce concepts and methods that justify and motivate some of the procedures proposed in this article [24–28].

We would also like to mention that microscopic organisms are usually *active*, which implies moreover, the need of considering them as controllable or controlled systems [29–31]. They produce mechanical energy through molecular motors that consume chemical compounds such as ATP (*adenosine triphosphate*). The incorporation of active elements in the model, which is not standard in CSM, is quite mandatory in the soft-bio realm.

In the following we propose finite element algorithms for one-dimensional solid models with finite thickness and finite length moving and deforming within a fluid. The algorithm does not rely on the regularizing effect of inertia of either the solid or the fluid, and will be presented neglecting all inertial terms to simplify the exposition and notation. The formulation adopts moving boundary-following meshes, which are in general more accurate than immersed-boundary methods à la Peskin [32] at the expense of requiring a robust meshing software. This turns to be a main innovation of the current manuscript. It should be mentioned that the complexity of micro-organisms’ shape and movement is simpler to address using boundary element and other singularity-based methods. They have been developed and exploited in a quite extensive body of previous work (see [33–41] and references therein), but they have limitations that make a finite element method *for the fluid* to be interesting. The most evident limitation is that singularity-based methods cannot be applied to fluids with non-Newtonian rheology, which are quite abundant. Beyond affecting the dynamics of locomotion in itself, this would have as well a non-negligible influence on the power expenditures [42,43]. Some works available in the literature consider nonlinear flows exhibiting shear-thinning [44–48], however, these do not solve the exact flow equations and/or rely on some problem-specific hypotheses that strongly simplify the problem. Therefore, the ability of the model proposed to consider generalized Newtonian rheologies represents as well another main contribution of this work. The second limitation arises when the body motion is coupled to the transport of some species (e.g., nutrients) by the surrounding fluid. Singularity-based methods do not compute a sparse representation of the bulk velocity of the fluid, which thus requires a (costly) additional reconstruction step before the transport calculations. Although the transport problem is not considered in the present contribution, the general setting proposed here can be extended in that direction. Such an extension will be then considered in the sequel.

## 2. Formulating the problem

### 2.1. The surface motion of a Cosserat rod

A quite general model of rod-like solids is provided by the Cosserat theory of rods [49,50]. In this theory, a rod is a one-dimensional object which can thus be parameterized by a single **body coordinate**  $s \in [0, S]$ . A **configuration** of the rod is defined by two elements: (a) a continuously differentiable mapping from  $[0, S]$  into  $\mathbb{R}^d$ , where  $d$  is the dimension of the ambient space, satisfying  $m'(s) \neq 0$  for all  $s$ ; and (b) a set of  $d$  vector-valued functions defined in  $[0, S]$ , called **directors**. The first director at each point can be chosen very conveniently to be the tangent vector, and the other (or others) must be *linearly independent* so as to complete a basis of  $\mathbb{R}^d$ .

The reference configuration  $(\mathbf{M}, \mathbb{D})$  of the rod is denoted by the mapping  $\mathbf{M}(s)$  and the directors  $\mathbb{D}(s) = \{\mathbf{D}_i(s)\}$ , with  $i = 1, \dots, d$  and  $s \in [0, S]$ . Let  $\mathbf{m}$  and  $\mathfrak{d} = \{\mathbf{d}_i\}$ , with  $\mathbf{d}_1(s) = \mathbf{m}'(s)$ , be the analogous functions that characterize another arbitrary configuration (i.e., the “actual” configuration). The basic idea is to adopt kinematical assumptions such that the deformation of the body’s wet surface is totally encoded in the functions  $\mathbf{m}$  and  $\{\mathbf{d}_i\}$ . Let  $\hat{S}$  be the wet surface in the

reference configuration, assumed to be closed and orientable. The material point that occupies the position  $\mathbf{Y} \in \widehat{\mathcal{S}}$  in the reference configuration is assigned a body coordinate  $\sigma(\mathbf{Y}) \in [0, S]$ . The function  $\sigma : \widehat{\mathcal{S}} \rightarrow [0, S]$  is assumed continuous and differentiable. If  $\widehat{\mathcal{C}} = \{\mathbf{M}(s), s \in [0, S]\}$  is the **centroidal line** of the body in the reference configuration,  $\mathbf{M}(\sigma(\mathbf{Y}))$  is the position of the centroidal point to which  $\mathbf{Y}$  is considered to be “attached”. We can thus write, for all  $\mathbf{Y} \in \widehat{\mathcal{S}}$ ,

$$\mathbf{Y} = \mathbf{M}(\sigma(\mathbf{Y})) + \sum_{k=1}^d \theta_k(\mathbf{Y}) \mathbf{D}_k(\sigma(\mathbf{Y})) , \quad (1)$$

which uniquely defines  $d$  real functions  $\{\theta_i\}$  on  $\widehat{\mathcal{S}}$ . In matrix notation

$$\boldsymbol{\theta}(\mathbf{Y}) = [\mathbb{D}(\sigma(\mathbf{Y}))]^{-1} (\mathbf{Y} - \mathbf{M}(\sigma(\mathbf{Y}))) , \quad (2)$$

where  $\mathbb{D}$  is the matrix that has as columns the components of the directors  $\{\mathbf{D}_i\}$  in some (Cartesian for simplicity) basis  $\{\mathbf{E}_i\}$ ,  $i = 1, \dots, d$ . Also, it is always possible (upon reparameterizing the body so that  $s$  is the arc length along  $\widehat{\mathcal{C}}$ ) to choose the reference directors forming an orthonormal basis, in which case  $\mathbb{D}$  is an orthogonal matrix with determinant  $+1$ , i.e., an element of the Lie Group  $SO(d)$ . In this way,  $\mathbf{Y}$  can be rewritten in matrix notation as

$$\mathbf{Y} = \mathbf{M}(\sigma(\mathbf{Y})) + \mathbb{D}(\sigma(\mathbf{Y})) \boldsymbol{\theta}(\mathbf{Y}) . \quad (3)$$

The kinematical assumption is that in the actual configuration, the position  $\mathbf{y}$  of the same material point is given by

$$\mathbf{y} = \boldsymbol{\psi}(\mathbf{Y}) = \mathbf{m}(\sigma(\mathbf{Y})) + \sum_{k=1}^d \theta_k(\mathbf{Y}) \mathbf{d}_k(\sigma(\mathbf{Y})) . \quad (4)$$

This can again be rewritten in matrix notation as

$$\mathbf{y} = \mathbf{m}(\sigma(\mathbf{Y})) + \mathbb{d}(\sigma(\mathbf{Y})) \boldsymbol{\theta}(\mathbf{Y}) , \quad (5)$$

where  $\mathbb{d}$  is the matrix that has as columns the components of the directors  $\{\mathbf{d}_k\}$  in the basis  $\{\mathbf{E}_i\}$ ,  $i = 1, \dots, d$ .

Notice that for each  $s \in [0, S]$ , the set  $\widehat{\mathcal{R}}(s) = \sigma^{-1}(s)$  consists of the points in the reference wet surface  $\widehat{\mathcal{S}}$  that have body coordinate  $s$ . From (4) we see that the actual position of this set of *material points* is totally dictated by  $\mathbf{m}(s)$  and  $\mathbb{d}(s)$ . Further, let  $\mathbf{A}$  and  $\mathbf{B}$  be two points belonging to  $\widehat{\mathcal{R}}(s)$ . Then

$$\mathbf{b} - \mathbf{a} = \mathbb{d}(s) [\boldsymbol{\theta}(\mathbf{B}) - \boldsymbol{\theta}(\mathbf{A})] = \mathbb{d}(s) [\mathbb{D}(s)]^{-1} (\mathbf{B} - \mathbf{A}) , \quad (6)$$

so that, restricted to  $\widehat{\mathcal{R}}(s)$ , the deformation is *linear*. Later on we consider extensions for bodies with a head attached to  $s = S$ , in the sense that  $\widehat{\mathcal{R}}(S)$  is a region of positive measure of  $\widehat{\mathcal{S}}$  (see Fig. 1). Such a head is thus subject to the translation  $\mathbf{m}(S) - \mathbf{M}(S)$  plus the linear deformation of gradient  $\mathbb{d}(S)[\mathbb{D}(S)]^{-1}$ . A similar appendage can be attached to  $s = 0$ .

Assuming that  $\mathbf{m}$ ,  $\{\mathbf{d}_k\}$ ,  $\sigma$  and  $\{\theta_k\}$  are regular enough, the mapping  $\boldsymbol{\psi}$  is continuous and differentiable. Under the additional restriction of  $\boldsymbol{\psi}$  being injective (no self intersections) we have that the actual wet surface  $S = \boldsymbol{\psi}(\widehat{\mathcal{S}})$  is also closed and orientable because  $\boldsymbol{\psi} : \widehat{\mathcal{S}} \rightarrow S$  is structure-preserving, continuous and invertible, i.e., a homeomorphism.

In this work we do not attempt a rigorous characterization of the configurational manifold  $\mathcal{Q}$  of a Cosserat rod. It is certainly a subset of the product space  $V \times W$ , where  $V$  is the space of differentiable vector-valued functions from  $[0, S]$  to  $\mathbb{R}^d$  and  $W$  is the space of matrix-valued functions from  $[0, S]$  to  $\mathbb{R}^{d \times d}$ . The differentiability of  $\mathbf{m}$  and  $\mathbb{d}$  must ensure not only that  $\boldsymbol{\psi}$  is a homeomorphism, but also that the mechanical energy with which the rod is endowed is finite<sup>1</sup>. The latter is frequently the most restrictive condition.

The tangent space  $T_{\mathbf{q}}\mathcal{Q}$  of configurational variations  $\delta\mathbf{q} = (\delta\mathbf{m}, \delta\mathbb{d})$  with respect to the actual configuration  $\mathbf{q} = (\mathbf{m}, \mathbb{d})$ , can thus be identified to a *subspace* of  $V \times W$ , subject to the admissibility constraint  $\delta\mathbf{d}_1 = \delta\mathbf{m}'$ .

Each configurational variation  $\delta\mathbf{q}$  induces, through (4), a vector field  $\mathbf{w}_{\delta\mathbf{q}}$  over the actual wet surface  $S$  given by

$$\mathbf{w}_{\delta\mathbf{q}}(\mathbf{y}) = \delta\mathbf{y} = \delta\mathbf{m} + \delta\mathbb{d} \boldsymbol{\theta} . \quad (7)$$

It represents the infinitesimal movement of the material points of the surface produced by  $\delta\mathbf{q}$ . The operator defined by  $H_{\mathbf{q}} : \delta\mathbf{q} \mapsto \mathbf{w}_{\delta\mathbf{q}}(\cdot)$  is *linear* from  $T_{\mathbf{q}}\mathcal{Q}$  into  $U(S)$ , the space of vector fields on  $S$ .

Letting now the actual configuration to evolve in time, dependence on the additional time variable  $t$  appears in the expressions above. For functions of both  $s$  and  $t$  we keep the notation  $f'$  for  $\partial_s f$  and introduce  $\dot{f}$  for  $\partial_t f$ .

It is clear that the configuration  $\mathbf{q}(t)$  and the generalized velocity  $\dot{\mathbf{q}}(t)$  of the rod at a given instant  $t$  must belong to  $\mathcal{Q} \times T_{\mathbf{q}(t)}\mathcal{Q} \subset \mathcal{Q} \times T\mathcal{Q}$ . From the definitions above, if  $\mathbf{u}_b(\mathbf{x}, t)$  is the velocity field of the body at the wet surface, then it is computed from  $\dot{\mathbf{q}}$  as

$$\mathbf{u}_b(\mathbf{y}, t) = \mathbf{w}_{\dot{\mathbf{q}}(t)}(\mathbf{y}, t) = [H_{\mathbf{q}(t)} \dot{\mathbf{q}}(t)](\mathbf{y}, t) , \quad \forall \mathbf{y} \in S(t) . \quad (8)$$

<sup>1</sup> To characterize the boundedness of the mechanical energy, it is necessary first to introduce deformation measures and then a constitutive model.

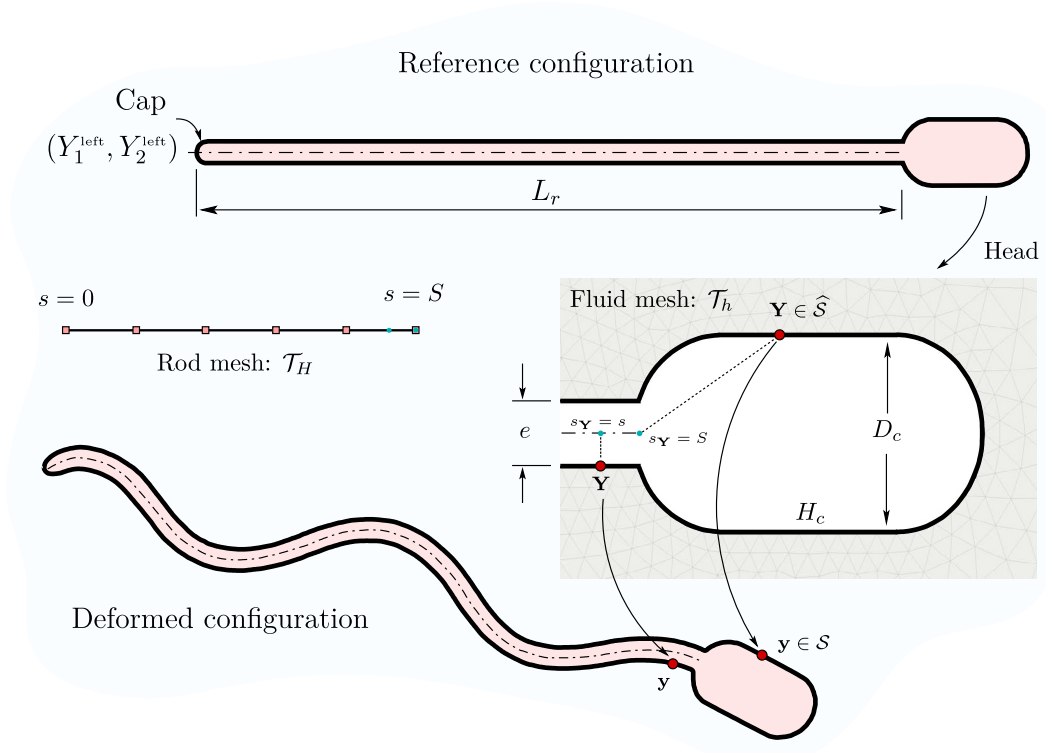


Fig. 1. Schematic of the computational domain showing the reference and a deformed configuration of the rod swimmer. The 2D fluid mesh and 1D rod mesh are also shown.

2.2. The case of a planar non-shearable rod

Further kinematical assumptions can be introduced, leading to more specific models. They will result in further linear admissibility constraints in  $T_q\mathcal{Q}$ . Let us consider a planar ( $d = 2$ ) non-shearable rod, which is adopted hereafter as application example. This rod model assumes that, in all configurations,  $\mathbf{d}_2$  is always normal to  $\mathbf{d}_1$ , so that

$$\mathbf{d}_2(s) = a(s)\check{\mathbf{n}}(s) = a(s)\mathbf{J}\check{\mathbf{t}}(s) = \frac{a(s)}{\|\mathbf{m}'(s)\|} \mathbf{J}\mathbf{m}'(s), \tag{9}$$

where  $\check{\mathbf{t}}$  and  $\check{\mathbf{n}}$  are tangent and normal unit vectors, respectively, and

$$\mathbf{J} := \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}. \tag{10}$$

We will further assume that the coefficient  $a(s)$  is totally determined by the curve  $\mathbf{m}$ . An example of this situation is when  $a(s)$  is a function of  $\mathbf{m}'(s)$ ,  $\mathbf{m}''(s)$ , etc., but the dependence could as well be nonlocal. To fix ideas, let us consider the particular case

$$a = a(\|\mathbf{m}'(s)\|). \tag{11}$$

With these kinematical assumptions the function  $\mathbf{m}(\cdot)$  totally determines both directors and thus also the whole deformation of the wet surface, as given by

$$\mathbf{y} = \mathbf{m}(s) + \theta_1(\mathbf{Y})\mathbf{m}'(s) + \theta_2(\mathbf{Y})\frac{a(\|\mathbf{m}'(s)\|)}{\|\mathbf{m}'(s)\|}\mathbf{J}\mathbf{m}'(s) \tag{12}$$

with  $s = \sigma(\mathbf{Y})$  and  $\mathbf{y} = \Psi(\mathbf{Y})$ . The additional constraint on  $T_q\mathcal{Q}$  is, defining  $b(s) = \|\mathbf{m}'(s)\|$ ,

$$\delta\mathbf{d}_2(s) = \frac{a'(b(s))}{b(s)}\mathbf{m}'(s) \cdot \delta\mathbf{m}'(s)\check{\mathbf{n}}(s) + a(b(s))\delta\check{\mathbf{n}}(s). \tag{13}$$

Since

$$\delta\check{\mathbf{n}}(s) = \mathbf{J}\delta\check{\mathbf{t}}(s) \quad \text{and} \quad \delta\check{\mathbf{t}}(s) = \frac{1}{b(s)}\mathbf{P}^\perp \delta\mathbf{m}', \tag{14}$$

where  $\mathbf{P}^\perp = \mathbf{I} - \check{\mathbf{t}} \otimes \check{\mathbf{t}}$ , we can also write the constraint as

$$\delta \mathbf{d}_2(s) = \left[ \frac{a'(b(s))}{b(s)} \check{\mathbf{n}}(s) \otimes \mathbf{m}'(s) + \frac{a(b(s))}{b(s)} \mathbf{J} \mathbf{P}^\perp(s) \right] \delta \mathbf{m}'(s). \tag{15}$$

The previous calculations show that (7) in this case takes the form

$$\begin{aligned} \mathbf{w}_{\delta \mathbf{q}}(\mathbf{y}) &= \delta \mathbf{m}(s) + [\theta_1(\mathbf{Y}) \mathbf{I} + \theta_2(\mathbf{Y}) \mathbf{G}(s)] \delta \mathbf{m}'(s) \\ &= \{\mathbf{I} + [\theta_1(\mathbf{Y}) \mathbf{I} + \theta_2(\mathbf{Y}) \mathbf{G}(s)] \partial_s\} \delta \mathbf{m}(s), \end{aligned} \tag{16}$$

where  $\mathbf{G}(s)$  is the  $2 \times 2$  matrix that appears between brackets on the right-hand side of (15).

Without loss of generality, we assume that  $\|\mathbf{D}_1(s)\| = \|\mathbf{D}_2(s)\| = 1$ , so that the reference directors coincide with tangent and normal unit vectors, namely

$$\mathbf{D}_1(s) = \check{\mathbf{T}}(s) \quad \text{and} \quad \mathbf{D}_2(s) = \check{\mathbf{N}}(s), \quad \forall s. \tag{17}$$

In such a case, the coefficient  $a(s)$  measures the local contraction (if  $< 1$ ) or expansion (if  $> 1$ ) of imaginary fibers along the normal direction, while  $b(s) = \|\mathbf{m}'(s)\|$  is the local contraction (if  $< 1$ ) or expansion (if  $> 1$ ) of imaginary fibers along the tangent direction.

The configuration of this model of rod, as becomes clear from above, is totally determined by  $\mathbf{m}(\cdot)$ . One can thus adopt a *reduced* configuration manifold  $\mathcal{Q}$  consisting just of the centroid mappings  $\mathbf{m} : [0, S] \rightarrow \mathbb{R}^d$ , in which case  $\mathbf{q} = \mathbf{m}$ ,  $\delta \mathbf{q} = \delta \mathbf{m}$  (unrestricted), and  $T_{\mathbf{q}} \mathcal{Q}$  is a subspace of  $V$ . In this reduced formulation, the operator  $H_{\mathbf{m}} : T_{\mathbf{q}} \mathcal{Q} \rightarrow U(S)$  (the space of vector fields on  $S$ ) is given by

$$H_{\mathbf{m}} = \mathbf{I} + [\theta_1(\mathbf{Y}) \mathbf{I} + \theta_2(\mathbf{Y}) \mathbf{G}(s)] \partial_s. \tag{18}$$

Furthermore, expression (8) for the velocity of the body at the wetted surface particularizes to

$$\mathbf{u}_b(\mathbf{y}, t) = [H_{\mathbf{m}(\cdot, t)} \dot{\mathbf{m}}(\cdot, t)](\mathbf{y}, t) = \dot{\mathbf{m}}(s, t) + [\theta_1(\mathbf{Y}) \mathbf{I} + \theta_2(\mathbf{Y}) \mathbf{G}(s)] \dot{\mathbf{m}}'(s, t). \tag{19}$$

### 2.3. The ambient fluid and the fluid–solid kinematics

Let us denote by  $\Omega_f(\mathbf{q})$  the domain of the ambient fluid when the rod is in the configuration  $\mathbf{q}$ . It has  $S(\mathbf{q})$  as internal boundary and let  $\Gamma$  be its (fixed) external boundary of which a portion  $\Gamma_{\mathbf{u}}$  is a wall at which the fluid velocity  $\mathbf{u}$  is known (and equal to zero, for simplicity). Considering just the fluid part of the system, the space of admissible velocity fields is then

$$X_{\mathbf{q}} = \left\{ \mathbf{w} \in [H^1(\Omega_f(\mathbf{q}))]^d \mid \mathbf{w}(\mathbf{x}) = \mathbf{0} \text{ a.e. in } \Gamma_{\mathbf{u}} \right\}. \tag{20}$$

**Remark.** The space  $H^1(\Omega_f)$  in the definition of  $X_{\mathbf{q}}$  corresponds to a Newtonian ambient fluid. Other rheological behaviors may require its substitution by a suitable Sobolev space  $W^{m,p}(\Omega_f)$ .

The most frequent kinematical conditions at the fluid–solid interface are *no-slip conditions*, i.e., the fluid velocity  $\mathbf{u}$  must satisfy

$$\mathbf{u}(\mathbf{y}) = \mathbf{u}_b(\mathbf{y}) = [H_{\mathbf{q}} \dot{\mathbf{q}}](\mathbf{y}), \quad \forall \mathbf{y} \in S. \tag{21}$$

We assume that the kinematics of the solid at configuration  $\mathbf{q}$  is sufficiently smooth so that for any  $\mathbf{p} \in T_{\mathbf{q}} \mathcal{Q}$  the induced surface field  $\mathbf{w} = H_{\mathbf{q}} \mathbf{p}$  can be extended to  $\Omega_f$  as an element of  $X_{\mathbf{q}}$ , satisfying  $\|\mathbf{w}\|_{X_{\mathbf{q}}} \leq c \|\mathbf{p}\|_{T_{\mathbf{q}} \mathcal{Q}}$  for some  $c \geq 0$ . This requires, in particular, that  $H_{\mathbf{q}} \mathbf{p}$  belongs to the trace space of  $X_{\mathbf{q}}$  on  $S(\mathbf{q})$  for all  $\mathbf{p}$ . At this point it is convenient to define

$$\widehat{H}_{\mathbf{q}} = L_{\mathbf{q}} H_{\mathbf{q}} \tag{22}$$

as a linear continuous operator from  $T_{\mathbf{q}} \mathcal{Q}$  into  $X_{\mathbf{q}}$ , where  $L_{\mathbf{q}}$  is an (arbitrary) continuous linear extension operator from functions defined on  $S(\mathbf{q})$  to functions in  $X_{\mathbf{q}}$ .

As a consequence, at any instant  $t$  the pair  $(\dot{\mathbf{q}}, \mathbf{u}) \in T_{\mathbf{q}(t)} \mathcal{Q} \times X_{\mathbf{q}(t)}$ , which we call the *generalized velocity of the fluid–solid system*, must belong to the *space of kinematically admissible fluid–solid motions* defined by

$$Z_{\mathbf{q}(t)} = \left\{ (\mathbf{p}, \mathbf{w}) \in T_{\mathbf{q}(t)} \mathcal{Q} \times X_{\mathbf{q}(t)} \mid \mathbf{w}|_{S(\mathbf{q}(t))} = H_{\mathbf{q}(t)} \mathbf{p} \right\}, \tag{23}$$

where the notation  $f|_S$  denotes the restriction of the function to  $S$  (in the sense of traces). The variational formulation of the instantaneous FSI problem at time  $t$  is posed in the space  $Z_{\mathbf{q}(t)}$ .

2.4. The principle of virtual work and the evolution problem

In the previous sections we have defined and described the kinematics of the fluid–solid systems of which the solid’s motion is parameterized as a Cosserat rod. Now we consider the dynamics of such systems, as dictated by the principle of virtual work.

We assume that the rod is endowed with a differentiable energy  $E(t, \mathbf{q})$ , dependent on both the time and the configuration, and that all the inertial effects are neglected. Then, at each instant  $t$ , the instantaneous configuration  $\mathbf{q} \in \mathcal{Q}$  and generalized velocity  $(\dot{\mathbf{q}}, \mathbf{u}) \in Z_{\mathbf{q}}$  must satisfy

$$\langle D_{\mathbf{q}}E(t, \mathbf{q}), \delta \mathbf{q} \rangle + \int_{\Omega_f} \boldsymbol{\sigma} : \boldsymbol{\varepsilon}(\delta \mathbf{w}) \, d\Omega = \int_{\Omega_f} \mathbf{f} \cdot \delta \mathbf{w} \, d\Omega + \langle \mathbf{F}(t), \delta \mathbf{q} \rangle, \tag{24}$$

for all  $(\delta \mathbf{q}, \delta \mathbf{w}) \in Z_{\mathbf{q}}$ , where  $D_{\mathbf{q}}E(\mathbf{q})$  is the energy differential (with respect to  $\mathbf{q}$ ),  $\boldsymbol{\sigma}$  is the Cauchy stress tensor of the fluid,  $\boldsymbol{\varepsilon}$  is the symmetric gradient operator,  $\mathbf{f}$  is the body force acting on the fluid,  $\mathbf{F}$  is the generalized (non-conservative) force acting on the rod not coming from the ambient fluid and  $\langle \cdot, \cdot \rangle$  is the duality pairing between  $T_{\mathbf{q}}\mathcal{Q}$  and its dual. In what follows we will assume that  $\mathbf{f} = 0$  (notice that the Dirichlet boundary conditions are also zero). It is not at all essential, but it simplifies the presentation considerably.

From (24) one obtains the evolution problem in variational form. Let us consider an incompressible Stokesian fluid model defined by

$$\boldsymbol{\sigma} = -p\mathbf{I} + 2\mu \boldsymbol{\varepsilon}(\mathbf{u}) \quad \text{and} \quad \nabla \cdot \mathbf{u} = 0 \tag{25}$$

(incompressible Newtonian fluid, or quasi-Newtonian if  $\mu$  depends on the deformation rate), where  $p$  stands for the pressure field. Then the evolution problem reads

**Problem P1.** “Find  $\mathbf{q} : (0, T) \rightarrow \mathcal{Q}$ , such that for a.e.  $t \in (0, T)$  there exists  $\mathbf{u}(\cdot, t) \in X_{\mathbf{q}(t)}$  and  $p(\cdot, t) \in \Pi_{\mathbf{q}(t)} = L^2(\Omega_f(\mathbf{q}(t)))$  satisfying  $\mathbf{q}(0) = \mathbf{q}_0 \in \mathcal{Q}$ , together with

$$(\mathbf{q}'(t), \mathbf{u}(\cdot, t)) \in Z_{\mathbf{q}(t)} \quad (\text{i.e., kinematic admissibility}), \tag{26}$$

and

$$\langle D_{\mathbf{q}}E(t, \mathbf{q}(t)), \delta \mathbf{q} \rangle + \int_{\Omega_f} 2\mu \boldsymbol{\varepsilon}(\mathbf{u}) : \boldsymbol{\varepsilon}(\delta \mathbf{w}) \, d\Omega - \int_{\Omega_f} p \nabla \cdot \delta \mathbf{w} \, d\Omega = \langle \mathbf{F}, \delta \mathbf{q} \rangle, \tag{27}$$

$$\int_{\Omega_f} r \nabla \cdot \mathbf{u} \, d\Omega = 0, \tag{28}$$

for all  $(\delta \mathbf{q}, \delta \mathbf{w}) \in Z_{\mathbf{q}(t)}$  and all  $r \in \Pi_{\mathbf{q}(t)}$ ”.

Under realistic hypotheses on the data, problem P1 is expected to admit a unique solution. These hypotheses involve the energy function, requiring it to be smooth enough and exhibiting lower semicontinuity or even better some kind of suitable convexity, with the kernel of  $D_{\mathbf{q}}E$  (typically rigid-body motions) of finite dimension. They also involve the forcing, requiring it to be bounded in some suitable norm, for example

$$\sup_{t \in (0, T)} \|\mathbf{F}(t)\|_{T_{\mathbf{q}(t)}\mathcal{Q}'} < +\infty, \tag{29}$$

and the geometry of the system, requiring the operator  $\widehat{H}_{\mathbf{q}}$  to be **injective** and **compact** from  $T_{\mathbf{q}}\mathcal{Q}$  into  $X_{\mathbf{q}}$ . In particular, any non-zero infinitesimal deformation  $\delta \mathbf{q}$  of the solid must produce a non-zero surface field  $H_{\mathbf{q}}\delta \mathbf{q}$  on  $S$ . In particular, if the rod is not attached to any solid boundary, the rigid motions are controlled by the viscous operator.

The numerical approximation proposed in this work is best understood if we introduce a new problem, equivalent to P1. For this purpose, let us consider an arbitrary instant  $t$  at which the solid configuration is  $\mathbf{q}$ . Then, for each  $\mathbf{p} \in T_{\mathbf{q}}\mathcal{Q}$  there exists a unique velocity–pressure pair  $(\mathbf{u}, p)$  that satisfies the boundary condition

$$\mathbf{u}|_{S(\mathbf{q})} = H_{\mathbf{q}}\mathbf{p} \tag{30}$$

and the fluid’s differential equations in  $\Omega_f(\mathbf{q})$ . In variational terms, it is the unique solution to (27)–(28) when  $\delta \mathbf{q} = 0$ , that is,

$$\int_{\Omega_f(\mathbf{q})} 2\mu \boldsymbol{\varepsilon}(\mathbf{u}) : \boldsymbol{\varepsilon}(\delta \mathbf{w}) \, d\Omega - \int_{\Omega_f(\mathbf{q})} p \nabla \cdot \delta \mathbf{w} \, d\Omega = 0, \tag{31}$$

$$\int_{\Omega_f(\mathbf{q})} r \nabla \cdot \mathbf{u} \, d\Omega = 0, \tag{32}$$

for all  $r \in \Pi_{\mathbf{q}}$  and all  $\delta \mathbf{w}$  in

$$X_{\mathbf{q}}^0 = \{\mathbf{v} \in X_{\mathbf{q}} \mid \mathbf{v} = 0 \text{ on } S(\mathbf{q})\}. \tag{33}$$

We denote this velocity field by  $\mathcal{U}_{\mathbf{q},\mathbf{p}}$  and the corresponding pressure field by  $\mathcal{P}_{\mathbf{q},\mathbf{p}}$ . By construction, given  $\mathbf{q}$  and  $\mathbf{p}$  it holds that

$$(\mathbf{p}, \mathcal{U}_{\mathbf{q},\mathbf{p}}) \in Z_{\mathbf{q}} \tag{34}$$

and that

$$-\nabla \cdot [2\mu \boldsymbol{\varepsilon}(\mathcal{U}_{\mathbf{q},\mathbf{p}})] + \nabla \mathcal{P}_{\mathbf{q},\mathbf{p}} = 0 \quad \text{and} \quad \nabla \cdot \mathcal{U}_{\mathbf{q},\mathbf{p}} = 0, \quad \text{a.e. in } \Omega_f(\mathbf{q}). \tag{35}$$

From the assumed continuity and injectivity of  $\widehat{H}_{\mathbf{q}}$ , we have that  $\mathcal{U}_{\mathbf{q},\mathbf{p}}$  and  $\mathcal{P}_{\mathbf{q},\mathbf{p}}$  are continuous in  $\mathbf{p}$ . If the fluid is Newtonian, they are linear in  $\mathbf{p}$  and there exists  $C_S \geq 0$  (dependent on  $\mathbf{q}$ ) such that, for all  $\mathbf{p} \neq 0$ ,

$$0 < \|(\mathcal{U}_{\mathbf{q},\mathbf{p}}, \mathcal{P}_{\mathbf{q},\mathbf{p}})\|_{X_{\mathbf{q}} \times \Pi_{\mathbf{q}}} \leq C_S \|\mathbf{p}\|_{T_{\mathbf{q}}\mathcal{Q}}. \tag{36}$$

The formulation of problem P1 that is considered for approximation is then as follows:

**Problem P2.** “Find  $\mathbf{q} : (0, T) \rightarrow \mathcal{Q}$  such that  $\mathbf{q}(0) = \mathbf{q}_0$  and

$$b(\mathbf{q}(t); \dot{\mathbf{q}}(t), \delta\mathbf{q}) + \langle D_{\mathbf{q}}E(t, \mathbf{q}(t)), \delta\mathbf{q} \rangle = \langle \mathbf{F}(t), \delta\mathbf{q} \rangle, \tag{37}$$

for a.e.  $t \in (0, T)$  and all  $\delta\mathbf{q}$  such that  $\delta\mathbf{q}(t) \in T_{\mathbf{q}(t)}\mathcal{Q}$ , where  $b(\mathbf{q}; \cdot, \cdot) : T_{\mathbf{q}}\mathcal{Q} \times T_{\mathbf{q}}\mathcal{Q} \rightarrow \mathbb{R}$  is given by

$$b(\mathbf{q}; \mathbf{p}, \delta\mathbf{q}) = \int_{\Omega_f(\mathbf{q})} [2\mu \boldsymbol{\varepsilon}(\mathcal{U}_{\mathbf{q},\mathbf{p}}) : \boldsymbol{\varepsilon}(\widehat{H}_{\mathbf{q}}\delta\mathbf{q}) - \mathcal{P}_{\mathbf{q},\mathbf{p}} \nabla \cdot (\widehat{H}_{\mathbf{q}}\delta\mathbf{q})] d\Omega. \tag{38}$$

Notice that  $b(\mathbf{q}; \mathbf{p}, \delta\mathbf{q})$  may be nonlinear in  $\mathbf{p}$  and is always linear in  $\delta\mathbf{q}$ .

**Remark.** The fluid dissipation is given by

$$\mathcal{D}(\mathbf{q}; \mathbf{p}) = \int_{\Omega_f(\mathbf{q})} 2\mu \boldsymbol{\varepsilon}(\mathcal{U}_{\mathbf{q},\mathbf{p}}) : \boldsymbol{\varepsilon}(\mathcal{U}_{\mathbf{q},\mathbf{p}}) d\Omega, \tag{39}$$

which using (31)–(32) is seen to satisfy, for  $\mathbf{p} \neq 0$ ,

$$\mathcal{D}(\mathbf{q}; \mathbf{p}) = b(\mathbf{q}; \mathbf{p}, \mathbf{p}) > 0. \tag{40}$$

**Remark.** In the linear (Newtonian) case  $b(\mathbf{q}; \cdot, \cdot)$  is a scalar product in  $T_{\mathbf{q}}\mathcal{Q}$ . In fact, the extension operator  $L_{\mathbf{q}}$  (see (22)) can then be chosen as the Stokes solution; i.e.  $L_{\mathbf{q}}H_{\mathbf{q}}\mathbf{p} = \mathcal{U}_{\mathbf{q},\mathbf{p}}$ . Then the form  $b(\mathbf{q}; \cdot, \cdot)$  simplifies to

$$b(\mathbf{q}; \mathbf{p}, \delta\mathbf{q}) = \int_{\Omega_f(\mathbf{q})} 2\mu \boldsymbol{\varepsilon}(\mathcal{U}_{\mathbf{q},\mathbf{p}}) : \boldsymbol{\varepsilon}(\mathcal{U}_{\mathbf{q},\delta\mathbf{q}}) d\Omega, \tag{41}$$

which is indeed bilinear, symmetric and positive definite. This however will not be further exploited because our interest is in formulations that remain valid when the fluid is nonlinear.

### 2.5. Full model of a planar non-shearable rod

Let us now fully define problem P2 for the planar non-shearable rod of Section 2.2, so that the whole procedure becomes clear. We adopt the reduced configuration in which  $\mathbf{q} = \mathbf{m}$ , so that a configuration is a function  $\mathbf{m} : [0, S] \rightarrow \mathbb{R}^2$ . Similarly, the reference configuration is  $\mathbf{M} : [0, S] \rightarrow \mathbb{R}^2$ . It only remains to specify the energy function  $E$ . Assuming both  $\mathbf{m}$  and  $\mathbf{M}$  to be smooth enough, we define the rod’s energy<sup>2</sup> as

$$E = \frac{1}{2} \int_0^S [C_{\epsilon} \epsilon^2 + C_{\kappa} \kappa^2] ds, \tag{42}$$

where  $\epsilon$  is the local measure of elongation change

$$\epsilon = \check{\mathbf{t}} \cdot \mathbf{m}' - \check{\mathbf{T}} \cdot \mathbf{M}' - \epsilon_0, \tag{43}$$

$\kappa$  measures the curvature change

$$\kappa = \check{\mathbf{t}} \cdot \check{\mathbf{n}}' - \check{\mathbf{T}} \cdot \check{\mathbf{N}}' - \kappa_0 \tag{44}$$

and  $C_{\epsilon}$  and  $C_{\kappa}$  are positive constants (they could also depend on  $s$ ). The functions  $\epsilon_0(t, s)$  and  $\kappa_0(t, s)$ , assumed known, correspond to spontaneous values of elongation and curvature. The energy depends explicitly on time through them, thus providing a mechanism for the injection of mechanical energy into the system even if the external forces  $\mathbf{F}$  are zero (for self-locomotion, for example). Broadly speaking, for  $E(t, \mathbf{q}(t))$  to be finite, and thus  $\mathbf{q}(t) \in \mathcal{Q}$ , the tangential deformation

<sup>2</sup> This is borrowed from the small-strain rod theories.



must belong to  $H^1(0, S)$  and the normal deformation to  $H^2(0, S)$ . This is sufficient regularity to expect that the induced vector fields  $H_{\mathbf{q}(t)}\delta\mathbf{q}$  belong to  $H^{\frac{1}{2}}(S)$  and that  $\widehat{H}_{\mathbf{q}}$  is injective and compact as required. In physical terms, this means that the smoothness of  $\mathbf{m}(t, \cdot) : [0, S] \rightarrow \mathbb{R}^2$  is dictated by the rod and not by the ambient fluid.

The first variation of the elastic energy reads

$$\delta E = \langle D_{\mathbf{m}}E, \delta\mathbf{m} \rangle = \int_0^S (\epsilon C_\epsilon \delta\epsilon + \kappa C_\kappa \delta\kappa) ds, \quad (45)$$

where the variations of  $\epsilon$  and  $\kappa$  are

$$\delta\epsilon = \check{\mathbf{t}} \cdot \delta\mathbf{m}, \quad (46)$$

$$\delta\kappa = |\mathbf{m}'|^{-2} \mathbf{m}'' \cdot (2\check{\mathbf{n}} \otimes \check{\mathbf{t}} - \mathbf{J}) \delta\mathbf{m} - |\mathbf{m}'|^{-1} \check{\mathbf{n}} \cdot \delta\mathbf{m}, \quad (47)$$

which can be written in compact form as

$$\delta E = \int_0^S (\mathcal{B}(\mathbf{m}', \mathbf{m}'') \delta\mathbf{m}) \cdot \mathbf{s}_{\text{elastic}} ds \quad (48)$$

with the stress resultants defined as

$$\mathbf{s}_{\text{elastic}} = \mathbf{C} \mathbf{e}_{\text{elastic}}, \quad (49)$$

where

$$\mathbf{C} := \begin{pmatrix} C_\epsilon & 0 \\ 0 & C_\kappa \end{pmatrix} \quad \text{and} \quad \mathbf{e}_{\text{elastic}} = \begin{bmatrix} \epsilon \\ \kappa \end{bmatrix}. \quad (50)$$

The first component of  $\mathbf{s}_{\text{elastic}}$  is to be assigned to the elastic axial force and the second one to the elastic bending moment. The linearized strain operator is given by

$$\mathcal{B}(\mathbf{m}', \mathbf{m}'') := \begin{pmatrix} \check{\mathbf{t}} \cdot \mathbf{I} & 0 \\ |\mathbf{m}'|^{-2} \mathbf{m}'' \cdot (2\check{\mathbf{n}} \otimes \check{\mathbf{t}} - \mathbf{J}) & |\mathbf{m}'|^{-1} \check{\mathbf{t}} \cdot \mathbf{J} \end{pmatrix} \begin{pmatrix} \partial_s^1 \\ \partial_s^2 \end{pmatrix} \quad (51)$$

with  $\partial_s^1 : \mathbf{m} \mapsto \mathbf{m}'$  and  $\partial_s^2 : \mathbf{m} \mapsto \mathbf{m}''$  denoting differentiation operators.

Lastly, the second variation of the potential energy reads

$$\Delta\delta E = \int_0^S \delta\mathbf{m} \cdot [\mathcal{H}_{\text{material}}(\mathbf{m}', \mathbf{m}'') + \mathcal{H}_{\text{geometric}}(\mathbf{m}', \mathbf{m}'')] \Delta\mathbf{m} ds, \quad (52)$$

in which the operator associated to the material stiffness is given by

$$\mathcal{H}_{\text{material}}(\mathbf{m}', \mathbf{m}'') = \mathcal{B}(\mathbf{m}', \mathbf{m}'')^T \mathbf{C} \mathcal{B}(\mathbf{m}', \mathbf{m}'') \quad (53)$$

and the operator associated to the geometrical stiffness is given by

$$\mathcal{H}_{\text{geometric}}(\mathbf{m}', \mathbf{m}'') = \partial_{\mathbf{m}'}(\mathcal{B}(\mathbf{m}', \mathbf{m}'')^T \mathbf{s}_{\text{elastic}}) \partial_s^1 + \partial_{\mathbf{m}''}(\mathcal{B}(\mathbf{m}', \mathbf{m}'')^T \mathbf{s}_{\text{elastic}}) \partial_s^2. \quad (54)$$

**Remark.** The formulation above is *objective*, i.e., invariant under translations and rotations, as well as *path-independent*, i.e., for conservative loading and time-independent energy function the work produced through any arbitrary closed path is identically zero. The rigid body motions span the null space of the stiffness operator.

### 3. Numerical method

#### 3.1. Fully discrete formulation

We make a strong hypothesis here, namely that finite element spaces  $\mathcal{Q}_H \subset \mathcal{Q}$  exist such that any  $\mathbf{q} \in \mathcal{Q}$  can be approximated by some  $\mathbf{q}_H \in \mathcal{Q}_H$ . Since  $\mathcal{Q}_H$  is a vector space, it is identical to its tangent space at any point so that we will define, for simplicity,

$$\mathcal{Q}_H := \mathcal{Q}_H \equiv T_{\mathbf{q}_H} \mathcal{Q}_H, \quad \forall \mathbf{q}_H \in \mathcal{Q}_H. \quad (55)$$

This assumption holds for the planar non-shearable rod that has been adopted as example, for which  $\mathcal{Q}_H$  is taken as the space of  $C^1$ -conforming piecewise cubic polynomials (for both components of  $\mathbf{q} = \mathbf{m}$ ).

We also assume that, for each configuration  $\mathbf{q}_H \in \mathcal{Q}_H$  there exist finite element spaces  $V_h(\mathbf{q}_H) \subset X_{\mathbf{q}_H}$  and  $\Pi_h(\mathbf{q}_H) \subset \Pi_{\mathbf{q}_H}$ . The parameters  $H$  and  $h$  represent the sizes of the meshes of the rod (one-dimensional) and of the fluid ( $d$ -dimensional), respectively. The construction of  $V_h(\mathbf{q}_H)$  and  $\Pi_h(\mathbf{q}_H)$  requires a mesh over  $\Omega_f(\mathbf{q}_H)$ . Notice that meshes corresponding to



different times, which will in general have different  $\mathbf{q}_H$ 's, are totally independent. To each  $\mathbf{r}_H \in Q_H$  we associate the fields  $\mathbf{u}_h = \mathcal{U}_{h,\mathbf{q}_H,\mathbf{r}_H}$  and  $p_h = \mathcal{P}_{h,\mathbf{q}_H,\mathbf{r}_H}$  unique solutions of

$$\left\{ \begin{array}{l} (\mathbf{u}_h, p_h) \in V_h(\mathbf{q}_H) \times \Pi_h(\mathbf{q}_H), \\ \mathbf{u}_h = \mathcal{I}_h H_{\mathbf{q}_H} \mathbf{r}_H \quad \text{on } S(\mathbf{q}_H), \\ \int_{\Omega_f(\mathbf{q}_H)} 2\mu \boldsymbol{\varepsilon}(\mathbf{u}_h) : \boldsymbol{\varepsilon}(\mathbf{w}_h) \, d\Omega - \int_{\Omega_f(\mathbf{q}_H)} p_h \nabla \cdot \mathbf{w}_h \, d\Omega = 0, \\ \int_{\Omega_f(\mathbf{q}_H)} m_h \nabla \cdot \mathbf{u}_h \, d\Omega + \int_{\Omega_f(\mathbf{q}_H)} \tau_h \nabla p_h \cdot \nabla m_h \, d\Omega = 0, \end{array} \right. \tag{56}$$

for all  $m_h \in \Pi_h(\mathbf{q}_H)$  and all  $\mathbf{w}_h$  in

$$V_{h0}(\mathbf{q}_H) = \{\mathbf{w}_h \in V_h(\mathbf{q}_H) \mid \mathbf{w}_h = 0 \text{ on } S(\mathbf{q}_H)\} . \tag{57}$$

Notice that if  $\mu$  depends on the local strain rate then problem (56) is nonlinear. The term containing  $\tau_h$  is a stabilization term so that equal-order elements are rendered convergent by adopting  $\tau_h = h^2/4\mu$ . The operator  $\mathcal{I}_h$  interpolates a vector function defined on  $S(\mathbf{q}_H)$  onto the trace space of  $V_h(\mathbf{q}_H)$ . We assume that the spaces  $V_h$  and  $\Pi_h$  are built from the Taylor-Hood  $P_2 - P_1$  element (in which case  $\tau_h = 0$ ) or from the equal order  $P_1 - P_1$  element, so that it holds that [51,52]

$$\|\mathbf{u}_h - \mathcal{U}_{h,\mathbf{q}_H,\mathbf{r}_H}\|_{H^1(\Omega_f)} + \|p_h - \mathcal{P}_{h,\mathbf{q}_H,\mathbf{r}_H}\|_{L^2(\Omega_f)} \leq c h^k \tag{58}$$

with  $c$  not depending on  $h$  and  $k = 2$  for the  $P_2 - P_1$  case and  $k = 1$  for the  $P_1 - P_1$  one.

In our implementation we adopt Lagrange finite elements, so that  $\mathcal{I}_h$  is the nodal interpolation at the mesh nodes that lie on  $S(\mathbf{q}_H)$ . The extension operator  $L_{\mathbf{q}_H}$  is the simplest possible, namely all nodal values at nodes not belonging to  $S(\mathbf{q}_H)$  are set to zero. We define a vector field  $\mathbf{w}_h = \widehat{H}_{h,\mathbf{q}_H} \delta \mathbf{q}_H$  as taking the values  $H_{\mathbf{q}_H} \delta \mathbf{q}_H$  at nodes that lie on  $S(\mathbf{q}_H)$ , and zero at all others, i.e.,

$$\mathbf{w}_h = \widehat{H}_{h,\mathbf{q}_H} \delta \mathbf{q}_H \in V_h, \quad \mathbf{w}_h = \begin{cases} \mathcal{I}_h H_{\mathbf{q}_H} \delta \mathbf{q}_H & \text{on } S(\mathbf{q}_H), \\ \mathbf{0} & \text{at all interior nodes of } V_h. \end{cases} \tag{59}$$

With the previous definitions, and using superscript  $n$  to denote the time step, we will consider the following numerical methods:

**Fully discrete formulations of problem P2:** “Find  $\mathbf{q}_H^n \in Q_H$ , for  $n = 0, 1, 2, \dots$ , such that  $\mathbf{q}_H^0 = \mathcal{I}_H \mathbf{q}_0$  and

$$b_h \left( \mathbf{q}_H^n; \frac{\mathbf{q}_H^{n+1} - \mathbf{q}_H^n}{\delta t}, \delta \mathbf{q}_H \right) + \langle D_{\mathbf{q}} E(t_*, \mathbf{q}_*), \delta \mathbf{q}_H \rangle + S(\mathbf{q}_H^n; \mathbf{q}_H^{n+1} - \mathbf{q}_H^n, \delta \mathbf{q}_H) - \langle \mathbf{F}(t_*), \delta \mathbf{q}_H \rangle = 0 \tag{60}$$

for all  $\delta \mathbf{q}_H \in Q_H$ , where  $b_h(\mathbf{q}_H; \cdot, \cdot) : Q_H \times Q_H \rightarrow \mathbb{R}$  is given by

$$b_h(\mathbf{q}_H; \mathbf{p}_H, \delta \mathbf{q}_H) = \int_{\Omega_f(\mathbf{q}_H)} [2\mu \boldsymbol{\varepsilon}(\mathcal{U}_{h,\mathbf{q}_H,\mathbf{p}_H}) : \boldsymbol{\varepsilon}(\widehat{H}_{h,\mathbf{q}_H} \delta \mathbf{q}_H) - p_{h,\mathbf{q}_H,\mathbf{p}_H} \nabla \cdot (\widehat{H}_{h,\mathbf{q}_H} \delta \mathbf{q}_H)] \, d\Omega , \tag{61}$$

$\delta t = t_{n+1} - t_n$  is the time step and  $S(\mathbf{q}_H; \cdot, \cdot) : Q_H \times Q_H \rightarrow \mathbb{R}$  is a stabilizing bilinear form which will in general depend on the specific rod model.

The following choices for  $t_*$  and  $\mathbf{q}_*$  define the different variants considered in our study:

- **Explicit method:**  $t_* = t_n, \mathbf{q}_* = \mathbf{q}_H^n, S = 0$ .
- **Semi-implicit method:**  $t_* = t_n, \mathbf{q}_* = \mathbf{q}_H^n, S = S_{\text{material}}^n \neq 0$ .
- **Pseudo-implicit method:**  $t_* = t_{n+1}, \mathbf{q}_* = \mathbf{q}_H^{n+1}, S = 0$ ”.

Notice that in all three methods the bilinear (or nonlinear) form  $b_h$  is evaluated at  $\mathbf{q}_H^n$ , which means that the fluid’s geometry and mesh correspond to the previous time,  $t_n$ . The explicit method is a simple forward Euler of problem P2. The pseudo-implicit method, on the other hand, considers the elasticity of the rod **implicitly** and thus requires nonlinear iterations to compute  $\mathbf{q}_H^{n+1}$ . It is however not an implicit method since, as just said, the fluid’s treatment is explicit. Finally the semi-implicit method leads, when the fluid is Newtonian, to a linear algebraic system of equations, just as the explicit method. However, increased temporal stability is achieved by selecting  $S$  equal to  $S_{\text{material}}^n$  (the material stiffness, which depends on the rod model) evaluated at configuration  $\mathbf{q}_H^n$ , which is a positive semidefinite bilinear form. Such a particular stabilization is basically intended to improves the relative coercivity of the resulting bilinear form or nonlinear form, respectively.

Assuming  $\mathbf{q}_H^n$  known, the left-hand side of (60) can be viewed as a residual  $\mathcal{R} : \mathbb{R}^N \rightarrow \mathbb{R}^N$ ,  $N$  being the dimension of  $Q_H$ , i.e.,

$$\mathcal{R}(\mathbf{q}_H^{n+1}) = 0 . \tag{62}$$

The evaluation of the residual  $\mathcal{R}$  is crucial for any iterative strategy to solve (62). Each component of  $\mathcal{R}$  corresponds to taking  $\delta \mathbf{q}_H$  equal to one of the basis functions of  $Q_H$ . The computation is quite standard for all the terms in  $\mathcal{R}$  except for the first one, which materializes the fluid–rod interaction through the bilinear (or nonlinear) form  $b_h(\mathbf{q}_H^n; \cdot, \cdot)$ .

For each candidate solution  $\mathbf{q}_H^*$ , the computation of  $b_h(\mathbf{q}_H^n; (\mathbf{q}_H^* - \mathbf{q}_H^n)/\delta t, \delta \mathbf{q}_H)$  involves the following steps:

1. Build a mesh  $\mathcal{T}_h$  for the fluid domain  $\Omega_f^n = \Omega_f(\mathbf{q}_H^n)$  with the geometry defined by  $\mathbf{q}_H^n$ .
2. Solve the fluid problem (56) with  $\mathbf{r}_H = (\mathbf{q}_H^* - \mathbf{q}_H^n)/\delta t$ . Let  $(\mathbf{u}_h^*, p_h^*)$  be the corresponding velocity–pressure pair.
3. For each  $i = 1, \dots, n_{\text{dof}}$ , if  $\mathbf{N}^i$  is the  $i$ th basis function of  $Q_H$ , compute  $H_{\mathbf{q}_H^n} \mathbf{N}^i$  at the **fluid nodes** on the rod's boundary. Setting it to zero for all the other fluid nodes, one obtains  $\mathbf{w}_h^i = \widehat{H}_{h, \mathbf{q}_H^n} \mathbf{N}^i$ .
4. The first term in the  $i$ th component of  $\mathcal{R}(\mathbf{q}_H^*)$  is then obtained from the integral

$$\mathcal{R}^i(\mathbf{q}_H^*) = b_h \left( \mathbf{q}_H^n, \frac{\mathbf{q}_H^* - \mathbf{q}_H^n}{\delta t}, \mathbf{N}^i \right) + \text{o.t.} = \int_{\Omega_f(\mathbf{q}_H^n)} [2\mu \boldsymbol{\varepsilon}(\mathbf{u}_h^*) : \boldsymbol{\varepsilon}(\mathbf{w}_h^i) - p_h^* \nabla \cdot \mathbf{w}_h^i] d\Omega + \text{o.t.}, \quad (63)$$

where “o.t.” stands for the “other terms” in the left-hand side of (60), which all come from the solid model.

Notice that steps 1 and 3 do not depend on  $\mathbf{q}_H^*$  and can be computed just once per time step. It remains to provide the discretized equations of the rod so as to compute the remaining terms in  $\mathcal{R}^i(\mathbf{q}_H^*)$ . These equations, which depend on the rod model, are provided in detail for the case of the planar non-shearable rod in the following section.

**Remark.** For any  $\mathbf{q} \in \mathcal{Q}$ , any  $\mathbf{r}_H \in Q_H$  and any basis function  $\mathbf{N}^i \in Q_H$ , the difference between  $b(\mathbf{q}; \mathbf{r}_H, \mathbf{N}^i)$  and  $b_h(\mathbf{q}; \mathbf{r}_H, \mathbf{N}^i)$  is that in the latter the exact Stokes solution  $(\mathcal{U}_{\mathbf{q}, \mathbf{r}_H}, \mathcal{P}_{\mathbf{q}, \mathbf{r}_H})$  is approximated on a mesh of  $\Omega_f(\mathbf{q})$  of size  $h$ . A first formal estimate of the consistency error is

$$\begin{aligned} & b(\mathbf{q}; \mathbf{r}_H, \mathbf{N}^i) - b_h(\mathbf{q}; \mathbf{r}_H, \mathbf{N}^i) = \\ &= \int_{\Omega_f(\mathbf{q})} [2\mu \boldsymbol{\varepsilon}(\mathcal{U}_{\mathbf{q}, \mathbf{r}_H}) : \boldsymbol{\varepsilon}(\widehat{H}_{\mathbf{q}} \mathbf{N}^i) - \mathcal{P}_{\mathbf{q}, \mathbf{r}_H} \nabla \cdot (\widehat{H}_{\mathbf{q}} \mathbf{N}^i)] d\Omega - \\ & - \int_{\Omega_f(\mathbf{q})} [2\mu \boldsymbol{\varepsilon}(\mathcal{U}_{h, \mathbf{q}, \mathbf{r}_H}) : \boldsymbol{\varepsilon}(\widehat{H}_{h, \mathbf{q}} \mathbf{N}^i) - \mathcal{P}_{h, \mathbf{q}, \mathbf{r}_H} \nabla \cdot (\widehat{H}_{h, \mathbf{q}} \mathbf{N}^i)] d\Omega \leq \\ & \leq c_1 \left( \|\mathcal{U}_{\mathbf{q}, \mathbf{r}_H} - \mathcal{U}_{h, \mathbf{q}, \mathbf{r}_H}\|_{H^1(\Omega_f)} + \|\mathcal{P}_{\mathbf{q}, \mathbf{r}_H} - \mathcal{P}_{h, \mathbf{q}, \mathbf{r}_H}\|_{L^2(\Omega_f)} \right) + c_2 \|H_{\mathbf{q}} \mathbf{N}^i - \mathcal{I}_h H_{\mathbf{q}} \mathbf{N}^i\|_{H^{1/2}(S)} \leq \\ & \leq c_1 h^k + c_2' h^{3/2}, \end{aligned}$$

where  $c_1$ ,  $c_2$  and  $c_2'$  depend on  $\mathbf{q}$ ,  $\mathbf{r}_H$  and  $\mathbf{N}^i$ , but not on  $h$ , and we have used that  $\mathbf{N}^i$  is in  $H^2(0, S)$  in formally estimating the third term (interpolation error). By applying a duality argument the term  $h^k$  can probably be improved to  $h^{2k}$ . Recall that the total error involves the consistency error just discussed together with the interpolation error coming from the (Hermite) solid model,  $\|\mathbf{q} - \mathcal{I}_H \mathbf{q}\|_{H^2(0, S)} \leq c_3 H^2$ , where  $\mathcal{I}_H \mathbf{q} \in V_H$  is the Hermite interpolant of  $\mathbf{q}$ .

### 3.2. Software: Firedrake implementation for a planar non-shearable rod

Let us discuss the ingredients involved in the finite element implementation of the fluid structure interaction problem for the particular case of a planar non-shearable rod. The finite element library Firedrake [53] has been adopted. This platform allows expressive specification of PDE's using the Unified Form Language (UFL) from the FEniCS Project [54] through a python interface.

#### 3.2.1. Kinematics, mesh generation and remeshing

For the implementation we first need to collect some results previously introduced related to the kinematics of the planar non-shearable rod. Let us recall that the rod configuration at time  $t_n$  is encoded in the  $\mathbf{q}_H^n$  function, so that the fluid domain  $\Omega_f(\mathbf{q}_H^n)$  can be built from  $\mathbf{q}_H^n$  and the parameters  $\{\theta_1, \theta_2\}$  computed at the reference configuration. To fix ideas, let us consider the rod shown in Fig. 1 and suppose we have constructed an initial mesh around it. This will be the reference configuration. For each node over the wet boundary  $\widehat{S}$  (the red dots in the figure) we record its position  $\mathbf{Y}$  in this reference state. From this, the parameter  $s_{\mathbf{Y}} = \sigma(\mathbf{Y}) \in [0, S]$  to which each material point is attached can be found. Along its evolution, in any configuration of the rod, the actual position  $\mathbf{y}$  of the same material node is computed according to

$$\mathbf{y} = \mathbf{q}_H(s_{\mathbf{Y}}) + \theta_1(\mathbf{Y}) \mathbf{q}'_H(s_{\mathbf{Y}}) + \theta_2(\mathbf{Y}) a(s_{\mathbf{Y}}) \mathbf{J} \underbrace{\frac{\mathbf{q}'_H(s_{\mathbf{Y}})}{\|\mathbf{q}'_H(s_{\mathbf{Y}})\|}}_{\hat{\mathbf{n}}_H} \quad (64)$$

In the numerical experiments to be shown later on we consider two particular cases

$$a(s) = 1, \quad \text{and} \quad a(s) = \frac{1}{\|\mathbf{q}'_H(s)\|} \quad (65)$$

Consider the rod shown in Fig. 1 whose length in the reference configuration is denoted by  $L_r$  (Lref in the code). This rod has a cap with rounded shape at the left end, a head at the right end and thickness  $e \ll L_r$ . The left cap is described

by half of a circle of diameter  $e$ . The right head has two circular caps of diameter  $D_c = \beta_d e$  and a central region of height  $H_c = \beta_h e$ , being  $\beta_d, \beta_h > 1$  user defined parameters such that its size can be easily altered. These geometrical features are all indicated in the figure. For illustration purposes the left and right parts are dealt differently in this example, namely, the left rounded tip is part of the rod's domain, i.e., all the material points have  $0 \leq s_Y < S$ , while the material points belonging to the head are all attached to  $s_Y = S$ . This amounts to define

$$s_Y = \begin{cases} \frac{S}{L_r}(Y_1 - Y_1^{\text{left}}) & \text{if } Y_1 - Y_1^{\text{left}} \in [0, S) \\ S & \text{if } Y_1 - Y_1^{\text{left}} \geq S \end{cases} \quad (66)$$

where  $(Y_1^{\text{left}}, Y_2^{\text{left}})$  is the initial position of the left end. Since  $\mathbb{D} = \mathbf{I}$  and  $\mathbf{M}(s) = (sL_r/S + Y_1^{\text{left}}, Y_2^{\text{left}})$  by Eq. (2) the parameters  $\theta_i$  are thus given by

$$(\theta_1(\mathbf{Y}), \theta_2(\mathbf{Y})) = \begin{cases} (0, Y_2 - Y_2^{\text{left}}) & \text{if } s_Y \in [0, S) \\ (Y_1 - Y_1^{\text{left}} - L_r, Y_2 - Y_2^{\text{left}}) & \text{otherwise} \end{cases} \quad (67)$$

To update the position of nodes over the wet boundary, Eq. (64) is implemented by the python function `UpdateWetBoundary()`. This in turn calls some user defined functions, namely, `SigmaofY()` (Eq. (66)), `ComputeThetas()` (Eq. (67)) and `aofs()` (Eq. (65)) shown next. Additionally, the function `aofsp()` that computes  $a'(s)$ , to be used later on, is also shown.

```
# Program these functions according to your problem
def SigmaofY(xnodref): # Eq. (66)
    sY = S*(xnodref[0] - Y1left)/Lref
    if(sY > S):
        sY = S
    return sY
def ComputeThetas(xnodref): # Eq. (67)
    sY = SigmaofY(xnodref)
    if(sY < S):
        theta1 = 0
    else:
        theta1 = xnodref[0] - Y1left - Lref
    theta2 = xnodref[1] - Y2left
    return theta1, theta2

def aofs(modqp): # Eq. (65)
    return 1/modqp
def aofsp(qp, qpp): # Derivative of (65)
    return -np.dot(qp, qpp) / np.linalg.norm(qp)**3

def UpdateWetBoundary(xref, qH): # Eq. (64)
    new_y = np.zeros(shape=(nnod_wet,2))
    for j in range(nnod_wet): # Swept over wet nodes
        sY = SigmaofY(xref[j,:])
        el_inside = int(one_minus_eps * sY / h_r)
        xi = 2*sY/h_r - (2*el_inside+1)
        q = EvalqH(qH, el_inside, xi)
        qp = EvalqHp(qH, el_inside, xi)
        theta1, theta2 = ComputeThetas(xref[j,:])
        modqp = np.linalg.norm(qp)
        ny = q + theta1*qp + theta2*aofs(modqp)/modqp*np.dot(Jota, qp)
        new_y[j,:] = ny[:]
```

Functions `EvalqH()` and `EvalqHp()` that specify the rod configuration are explained later on when discussing the discretization of the rod problem. Table 1 lists several important objects and variables that appear in the code above and others to be presented afterwards. Notice that some of these variables may have global scope in the code.

The user defined functions can be adapted to consider other cases. For instance, for the simpler case of a rod with both flats ends the parameters  $\theta_i$  are just given by

$$(\theta_1(\mathbf{Y}), \theta_2(\mathbf{Y})) = (0, Y_2 - Y_2^{\text{left}}) \quad \forall s_Y \in [0, S]$$

For the sake of simplicity we have omitted additional arguments, such as  $\mathbb{D}$  and  $\mathbf{M}$ , that can be passed to these functions if necessary to deal with different or more complex reference configurations.

**Table 1**

List of important symbols and variables that appear in the Firedrake-python implementation made available alongside the article.

Symbol	Python	Description
$L_r$	Lref	Length of rod in the reference configuration ( $\mathbf{M}, \mathbb{D}$ )
$(Y_1^{\text{left}}, Y_2^{\text{left}})$	Y1left, Y2left	Position coordinates of the left end in the reference configuration
$(C_\epsilon, C_\kappa)$	Ce, Ck	Rod's cross-sectional elastic properties
$\mathbf{I}$	iden, I	$\text{np.array}([[1,0],[0,1]])$ , Identity(2) – Identity matrix
$\mathbf{J}$	Jota, J	$\text{np.array}([[0,-1],[1,0]])$ , as_matrix([[0, -1], [1, 0]]) – Unitary skew-symmetric matrix
$\mathcal{T}_H$	mesh_r	Firedrake object – 1D rod mesh
$N_{\text{el}}$	Nelrod	Number of elements in rod mesh
$N_{\text{nod}}$	Nnodrod	=Nelrod+1, Number of nodes in rod mesh
$N_{\text{dofs}}$	Ndofs	=4*Nnodrod, Total number of rod's dofs
$Q_H$	QH	VectorFunctionSpace(mesh_r, 'HER', degree=3, dim=2) – Hermite vector function space for rod discretization
$\mathbf{N}^i$	deltaqH[i-1]	=Function(QH) – Nodal basis functions for the space $Q_H$
$H$	h_r	=Lref/Nelrod, Fixed elemental length in the rod mesh
$s_Y = \sigma(\mathbf{Y})$	sY	=SigmaofY(xnodref) – Rod coordinate to which point $\mathbf{Y}$ is attached
$\mathbf{q}_H^{n+1}$	qH	=Function(QH) – Rod configuration at current time as function belonging to the space $Q_H$
$\mathbf{q}_H^n$	qHn	=Function(QH) – Rod configuration at previous time as function belonging to the space $Q_H$
$\dot{\mathbf{q}}_H^{n+1}$	qHdot	=Function(QH) – Rod velocity at current time as function belonging to the space $Q_H$
$\alpha = \{\alpha_i\}_{i=1}^{N_{\text{dofs}}}$	alpha	=np.zeros(Ndofs) – Rod velocity dofs at current time
$\mathbf{q}_H(s)$	q	Rod configuration evaluated at arbitrary $s$
$\mathbf{q}'_H(s)$	qp	First derivative of $\mathbf{q}_H$ evaluated at arbitrary $s$
$\mathbf{q}''_H(s)$	qpp	Second derivative of $\mathbf{q}_H$ evaluated at arbitrary $s$
$\delta t$	dt	Time step
–	EM	=MixedElement([VectorElement('Lagrange', 'triangle', 1), FiniteElement('Lagrange', 'triangle', 1)])
$V_{h0} \times \Pi_h$	W_f	=FunctionSpace(mesh_f, EM) – Mixed Lagrangian function space for fluid discretization
$\mathcal{T}_h$	mesh_f	Firedrake object – 2D fluid mesh
$\mathbf{w}_h^i$	wh[i-1]	=Function(W_f.sub(0)), ith induced velocity basis function
$(\hat{\mathbf{u}}_h, p_h)$	uo, p	=split(sol_f), sol_f = Function(W_f) – velocity–pressure pair
$\mathbf{u}_h$	u	Final velocity – $\mathbf{u}_h = \hat{\mathbf{u}}_h + \sum_{i=1}^{N_{\text{dof}}} \alpha_i^{n+1} \mathbf{w}_h^i$
$(\mathbf{v}_h, q_h)$	v, q	=TestFunctions(W_f)
$\tau_h$	tauh	=hk*hk/(4*mu) (hk = CellDiameter(mesh_f)) – Stabilization parameter in equal-order formulation
–	refinement	Typical element size close to the wet boundary
–	nnod_wet	Number of nodes over the wet boundary
–	scaling	=np.array([1., h_r]) – Scaling factors of Hermite shape functions
–	one_minus_eps	1-1e-12, geometrical tolerance

Once the new position of nodes over the wet boundary has been found, a finite element mesh is built. Linear triangles are used for discretization of the fluid domain. The mesh generator gmsh [55] (version 3.0.6) through the python library pygmsh [56] (version 6.1.0) is used for such a task. The advancing front algorithm from the gmsh suite was chosen in the numerical experiments below (gmsh\_args=['-algo', 'front2d']). Two functions are used to handle the mesh generation in our code, namely, MeshFluidRegion() and RemeshFluidRegion(). The former can be adapted so as to define different initial rod configurations.

```

...
mesh_f, ... = MeshFluidRegion( ..., refinement, gmsh_args, ... )
.
.
.
if (need_remesh):
    mesh_f, ... = RemeshFluidRegion( mesh_f, x_f, xref, . , list_ord, ... )
...

```

Many of the arguments in those functions have been omitted to avoid non-essential details. They nevertheless appear in the source code made available alongside the article. The characteristic size  $h$  of elements close to the rod's boundary  $\mathcal{S}(\mathbf{q}_H^0)$  in the initial configuration is controlled by the parameter refinement. A key issue during remeshing is to preserve the position of nodes located over the wet boundary. To that end, the mesh coordinates  $x_f$  of the original mesh are passed as an additional argument to the remeshing function along an ordered list of nodes list\_ord to be connected during

remeshing to form the new wet boundary. Since we expect to use relatively large time steps, remeshing will be applied at each time in the numerical experiments to be shown later on. Fig. 1 shows a detail of a typical initial mesh used in computations.

### 3.2.2. Discrete rod problem

Consider the rod domain  $[0, S]$  discretized into  $N_{el}$  elements of uniform length  $H = S/N_{el}$ . These parameters are identified in the implementation by `Nelrod` and `h_r`, respectively. Define the unknown discrete rod velocity  $\mathbf{q}_H \in Q_H$  between  $t_n$  and  $t_{n+1}$  as

$$\dot{\mathbf{q}}_H^{n+1} = \frac{\mathbf{q}_H^{n+1} - \mathbf{q}_H^n}{\delta t} \quad (68)$$

which is written as a linear combination of the nodal basis functions  $\{\mathbf{N}^i\}_{i=1}^{N_{dof}}$  of  $Q_H$ , i.e.,

$$\dot{\mathbf{q}}_H^{n+1} = \sum_{i=1}^{N_{dof}} \alpha_i \mathbf{N}^i \quad (69)$$

For discretization of the rod we use, as anticipated, one dimensional  $C^1$ -conforming Hermite elements and a regular grid with elements of length  $H$ . Being  $\dot{\mathbf{q}}_H^{n+1}$  a vector field with two components in 2D, the total number of dofs in the rod is  $N_{dof} = 4 N_{nod}$  with  $N_{nod} = N_{el} + 1$ . On each element  $e = 1, \dots, N_{el}$  of  $\mathcal{T}_H$  we approximate any function of  $Q_H$  as

$$\mathbf{q}_H(s)|_e = \mathbf{N}^e(\xi_s) \bar{\mathbf{q}}_H^e, \quad s = s_e \frac{1 - \xi_s}{2} + s_{e+1} \frac{1 + \xi_s}{2}, \quad s_e = eH \quad (70)$$

with the nodal values being  $\bar{\mathbf{q}}_H^e$  and the matrix of shape functions defined by

$$\mathbf{N}^e(\xi) = \left( \mathbf{N}_m^-(\xi) \quad \frac{H}{2} \mathbf{N}_t^-(\xi) \quad \mathbf{N}_m^+(\xi) \quad \frac{H}{2} \mathbf{N}_t^+(\xi) \right), \quad (71)$$

with the standard Hermite's polynomials

$$N_m^-(\xi) = \frac{1}{4}(2 + \xi)(1 - \xi)^2, \quad (72a)$$

$$N_t^-(\xi) = \frac{1}{4}(1 + \xi)(1 - \xi)^2, \quad (72b)$$

$$N_m^+(\xi) = \frac{1}{4}(2 - \xi)(1 + \xi)^2, \quad (72c)$$

$$N_t^+(\xi) = -\frac{1}{4}(1 - \xi)(1 + \xi)^2. \quad (72d)$$

These shape functions are defined for  $\xi \in [-1, +1]$ . Their evaluation at arbitrary points, as done for instance in `EvalqH()`, needs to be programmed manually since no Firedrake function that works for Hermite elements is available for such a task.

The following code block shows the definition of the vector space  $Q_H$  as a Firedrake object. Also, in our implementation we explicitly build the basis functions  $\mathbf{N}^i$  of the space  $Q_H$ . These functions are used later on as test functions in (63). In the code we also illustrate how any initial condition (a planar rod of length  $L_r$ ) is defined by using the Firedrake function `project()`.

```

mesh_r = IntervalMesh(Nelrod, S)
QH = VectorFunctionSpace(mesh_r, 'HER', degree=3, dim=2)
count, deltaqH = 0, []
for i in range(Nnodrod):
    [deltaqH.append(Function(QH)) for d in range(4)]
    deltaqH[count+0].dat.data[2*i][0] = 1
    deltaqH[count+1].dat.data[2*i][1] = 1
    deltaqH[count+2].dat.data[2*i+1][0] = h_r
    deltaqH[count+3].dat.data[2*i+1][1] = h_r
    count = count + 4
# Initial condition
x_r = SpatialCoordinate(mesh_r)
qH = project(as_vector([x_r[0]*Lref/S + Y1left, Y2left]), QH)

```

Let us now see how the terms arising from the rod model in Eq. (60) are computed for the case of a planar non-shearable rod. So, the terms we are considering here are

$$\text{Terms from rod model} = \langle D_q E(t_*, \mathbf{q}_*), \delta \mathbf{q}_H \rangle + S(\mathbf{q}_H^n; \mathbf{q}_H^{n+1} - \mathbf{q}_H^n, \delta \mathbf{q}_H) - \langle \mathbf{F}(t_*), \delta \mathbf{q}_H \rangle, \quad (73)$$

where  $t_* = t_n$ ,  $\mathbf{q}_* = \mathbf{q}_H^n$  correspond to the explicit method, for which  $S = 0$ , and to the semi-implicit method, for which  $S$  is given by

$$\begin{aligned} S(\mathbf{q}_H^n; \mathbf{q}_H^{n+1} - \mathbf{q}_H^n, \delta \mathbf{q}_H) &= \int_0^S \delta \mathbf{q}_H \cdot \mathcal{K}_{\text{material}}^n(\mathbf{q}_H^{n+1} - \mathbf{q}_H^n) ds \\ &= \delta t \int_0^S (\mathcal{B}^n \delta \mathbf{q}_H) \cdot \mathbf{C} (\mathcal{B}^n \dot{\mathbf{q}}_H^{n+1}) ds \\ &= \delta t \int_0^S \delta \mathbf{e}_{\text{elastic}} \cdot \mathbf{C} \Delta \mathbf{e}_{\text{elastic}} ds. \end{aligned} \quad (74)$$

Above,  $\delta \mathbf{e}_{\text{elastic}} = \mathcal{B}^n \delta \mathbf{q}_H$  and  $\Delta \mathbf{e}_{\text{elastic}} = \mathcal{B}^n \dot{\mathbf{q}}_H^{n+1}$ . Notice that in the notation we have used a supraindex  $n$  to denote the operators  $\mathcal{B}$  and  $\mathcal{K}_{\text{material}}$  evaluated at  $(\mathbf{q}_H^n, \dot{\mathbf{q}}_H^n)$ . Also notice for later use that

$$S(\mathbf{q}_H^n; \mathbf{q}_H^{n+1} - \mathbf{q}_H^n, \delta \mathbf{q}_H) = \delta t S(\mathbf{q}_H^n; \dot{\mathbf{q}}_H^{n+1}, \delta \mathbf{q}_H) \quad (75)$$

In turn, the pseudo-implicit method considers  $t_* = t_{n+1}$ ,  $\mathbf{q}_* = \mathbf{q}_H^{n+1}$ , and  $S = 0$ . The first term in (73) can also be written down explicitly, i.e.,

$$\langle D_{\mathbf{q}} E(t_*, \mathbf{q}_*), \delta \mathbf{q}_H \rangle = \int_0^S (\mathcal{B}^n \delta \mathbf{q}_H) \cdot \mathbf{C} \mathbf{e}_{\text{elastic}}^* ds = \int_0^S \delta \mathbf{e}_{\text{elastic}} \cdot \mathbf{C} \mathbf{e}_{\text{elastic}}^* ds, \quad (76)$$

where  $\mathbf{e}_{\text{elastic}}^*$  is the elastic strain corresponding to time  $t_*$  and configuration  $\mathbf{q}_*$ .

The implementation shown below is flexible and allows one to choose among the three different methods just proposed. Care was taken that there is a close correspondence between the presented formulation and the code of its Firedrake implementation. Functions `epsilon()` and `kappa()` compute the strain measures given in (43) and (44), and `depsilon()` and `dkappa()` their first variations with respect to  $d\mathbf{q}$  given by Eqs. (46) and (47). Lastly, `RodTerms()` computes all three terms of (73). In particular, `qs` corresponds to  $\mathbf{q}_*$ , the variables `eps` and `kap` are the components of  $\mathbf{e}_{\text{elastic}}^*$ , the variables `deps` and `dkap` are the components of  $\delta \mathbf{e}_{\text{elastic}}$  and the variables `depsd` and `dkapd` are those of  $\Delta \mathbf{e}_{\text{elastic}}$ .

Notice that the generalized force considered in the implementation is of the form

$$\langle \mathbf{F}(t), \delta \mathbf{q} \rangle = \int_0^S (\mathbf{f}_0(s, t) \cdot \delta \mathbf{q}(s) + \mathbf{f}_1(s, t) \cdot \delta \dot{\mathbf{q}}(s)) ds \quad (77)$$

where  $\mathbf{f}_0$  and  $\mathbf{f}_1$  are user defined functions.

```
def epsilon(t, qp, L0): # Eq. (43)
    return (dot(t, qp) - L0)
def kappa(t, np, k0): # Eq. (44)
    return (dot(t, np) - k0)
def depsilon(t, dq): # Eq. (46)
    return dot(t, dq.dx(0))
def dkappa(t, n, qp, qpp, dq): # Eq. (47)
    dnttmJ = 2*outer(n, t) - J
    dqp = dq.dx(0)
    dqpp = dq.dx(0).dx(0)
    mqp2 = dot(qp, qp)
    aux = dot(dnttmJ, dqp)
    return dot(qpp, aux)/mqp2 - dot(n, dqpp)/sqrt(mqp2)

def RodTerms(dt, eref, espont, fe, qHn, qHdot, dq): # Eq. (73)
    if(scheme == 'explicit' or scheme == 'semi_implicit'):
        qs = qHn
    else:
        qs = qHn + Constant(dt)*qHdot
    qp = qs.dx(0)
    qpp = qs.dx(0).dx(0)
    mqp = sqrt(dot(qp, qp))
    t = qp/mqp
    n = dot(J, t)
    Bper = I - outer(t, t)
    tp = dot(Bper, qpp)/mqp
    np = dot(J, tp)
    deps = depsilon(t, dq)
    eps = epsilon(t, qp, eref[0]+espont[0])
    dkap = dkappa(t, n, qp, qpp, dq)
```

```

kap = kappa(t,np,eref[1]+espont[1])
Fgrod = deps*Ce*eps + dkap*Ck*kap
if(scheme == 'semi_implicit'): # Stabilization: Eq. (74)
    depsd = depsilon(t,qHdot)
    dkapd = dkappa(t,n,qp,qpp,qHdot)
    Fgrod += Constant(dt)*(depsd*Ce*deps + dkapd*Ck*dkap)
Fgrod -= dot(fe[0],dq) + dot(fe[1],dq.dx(0))
return assemble(Fgrod * dx(degree=5))

```

### 3.2.3. Induced velocity basis functions

Construction of the velocity basis functions  $\mathbf{w}_h^i$  deserves special attention since it is a key ingredient in the proposed implementation and is related to the construction of the space of kinematically admissible velocity fields in the discrete case. Consider the finite element partition  $\mathcal{T}_h$  of  $\Omega_f(\mathbf{q}_h^n)$ . The  $i$ th basis function corresponds to taking  $\mathbf{N}^i$ ,  $i = 1, \dots, N_{\text{dof}}$  in Eq. (19). For a given node over the wet surface  $\mathcal{S}(\mathbf{q}_h^n)$ , with coordinate  $\mathbf{y}_j$ , corresponding to the material point  $\mathbf{Y}$  in the reference configuration and attached to the rod coordinate  $s_Y$ , we have

$$\mathbf{w}_h^i(\mathbf{y}_j) = \left( H_{\mathbf{q}_h^n} \right) (\mathbf{y}_j) = \mathbf{N}^i(s_Y) + [\theta_1(\mathbf{Y})\mathbf{I} + \theta_2(\mathbf{Y})\mathbf{G}(s_Y)] [\mathbf{N}^i(s_Y)]'. \quad (78)$$

and is defined as zero at all the nodes of  $\Omega_f(\mathbf{q}_h^n)$  that are not on the wet surface. These nodal values define  $\mathbf{w}_h^i$ , which in the discrete formulation appears as  $\mathbf{w}_h^i = H_{h,\mathbf{q}_h^n} \mathbf{N}^i$ . The relevant code to compute these functions is shown in the block below. For the  $i$ th dof we compute the function values of `w_h[i-1]` defined as a function belonging to the velocity space `W_f.sub(0)`. The python function `ApplyHhatqN()` executes a sweep over all rod's dofs, through four nested loops (over `i,d,c`), thus totalizing  $2*2*N_{\text{nodrod}}$  functions. The velocity is computed at each node on the wet surface by sweeping over the `nnod_wet` nodal coordinates `xref[j,:]` and calling function `VelWet()`, which is also shown. The `SupportInfo()` function provides auxiliary information to verify if a given point `j` is in the support of the corresponding basis function `i`.

```

def VelWet(xnodref,qp,qpp,Ndot,Ndotp): # Eq. (78)
    modqp = np.linalg.norm(qp)
    tan = qp / modqp
    nn = np.dot(Jota,tan)
    nntensorqp = np.tensordot(nn,qp,0)
    Bper = iden - np.tensordot(tan,tan,0)
    theta1,theta2 = ComputeThetas(xnodref)
    vel_wetnod = Ndot + theta1*Ndotp + \
        theta2*aofs(modqp)/modqp*np.dot(Jota*Bper,Ndotp) + \
        theta2*aofsp(qp,qpp)/modqp*np.dot(nntensorqp,Ndotp)
    return vel_wetnod

def ApplyHhatqN(W_f, xref, list_nodes_wet, qH): # Eq. (22)
    scaling = np.array([1.,h_r])
    Ndot, Ndotp = np.zeros(2), np.zeros(2)
    count, wh = 0, []
    for i in range(Nnodrod): # Swept over rod nodes
        [wh.append(Function(W_f.sub(0), name='wh')) for d in range(4)]
        for j in range(nnod_wet): # Swept over wet fluid nodes
            k = list_nodes_wet[j]
            el_inside, xi, idn = SupportInfo(i, xref[j,:])
            if(el_inside == i-1 or el_inside == i):
                qp = EvalqHp(qH, el_inside, xi)
                qpp = EvalqHpp(qH, el_inside, xi)
                HermBase, DerHermBase = HermB(xi), DHermB(xi)
                for d in range(2): # Swept over dofs per rod node
                    for c in range(2):
                        Ndot.fill(0.), Ndotp.fill(0.)
                        Ndot[c] = scaling[d]*HermBase[idn[d]]
                        Ndotp[c] = scaling[d]*(2/h_r)*DerHermBase[idn[d]]
                        vel = VelWet(xref[j,:],qp,qpp,Ndot,Ndotp)
                        wh[count+c+2*d].dat.data[k][:] = vel[:]
            count = count + 4
    return wh

```



### 3.2.4. Discrete fluid problem

Corresponding to any velocity field in the rod the induced velocity field in the fluid domain admits the additive decomposition

$$\mathbf{u}_h = \hat{\mathbf{u}}_h + \sum_{i=1}^{N_{\text{dof}}} \alpha_i \mathbf{w}_h^i = \hat{\mathbf{u}}_h + \sum_{i=1}^{N_{\text{dof}}} \alpha_i \hat{H}_{h, \mathbf{q}_H^n} \mathbf{N}^i \quad (79)$$

where  $\hat{\mathbf{u}}_h$  vanishes on  $S(\mathbf{q}_H^n)$ . Eq. (79) translates into the python code shown below

```
# Eq. (79)
uo, p = split(sol_f)
u = uo
for i in range(ndofs):
    u += Constant(alpha[i])*wh[i]
```

The implementation presented here is restricted to the case of  $P_1$  linear elements both for velocity and pressure, rendered stable by taking  $\tau_h = h_K^2/4\mu$  where  $h_K$  stands for the cell diameter. In the experiments below no body forces are considered, although their inclusion is straightforward. The following function determines  $\hat{\mathbf{u}}_h$  and the pressure field  $p_h$  by solving the fluid problem (56):

```
def SolveFluid(u, p, v, q, hk, sol, bcs): # Eq. (56)
    mu = viscosity(u)
    tauh = hk*hk/(4*mu) # Stabilization parameter
    F = ( inner(2*mu*sym(grad(u)), grad(v)) - div(v)*p ) * dx
    F += ( q*div(u) + tauh*inner(grad(q), grad(p)) ) * dx
    Jac = derivative(F, sol)
    solve(F == 0, sol, bcs=bcs, J=Jac, solver_parameters=sp)
    uo, p = split(sol)
    return uo, p
```

Recall that the fluid problem can be nonlinear if the viscosity  $\mu$  depends on  $\mathbf{u}$ . This nonlinearity is dealt with by means of a Newton–Raphson method with line search. Otherwise, the fluid solver converges in one single iteration. Firedrake is coupled to the PETSc library [57], from which the SNES functions for the non-linear part and several solvers for the linear part, can be invoked. This is defined in the `solver_parameters` object given next. Notice, the `mumps` [58,59] direct solver has been chosen for the linear solves.

```
# Fluid-solver options
sp = {
    'snes_type': 'newtonls',
    'snes_monitor': None,
    'snes_converged_reason': None,
    'snes_linesearch_type': 'basic',
    'snes_max_it': 20,
    'snes_divergence_tolerance': 1e20,
    'snes_rtol': 1e-7,
    'snes_atol': 1e-10,
    'mat_type': 'aij',
    'ksp_type': 'preonly',
    'pc_type': 'lu',
    'pc_factor_mat_solver_type': 'mumps'
}
```

### 3.2.5. FSI nonlinear coupling solver

Consider Eq. (63) viewed as a function of  $\hat{\mathbf{q}}_H^{n+1}$ , or equivalently as a function of the unknown parameters  $\underline{\alpha} = (\alpha_1, \dots, \alpha_{N_{\text{dof}}})^T$ . For each  $i = 1, \dots, N_{\text{dof}}$

$$\begin{aligned} \mathcal{R}^i(\underline{\alpha}) = & \int_{\Omega_f(\mathbf{q}_H^n)} [2\mu \boldsymbol{\varepsilon}(\mathbf{u}_h) : \boldsymbol{\varepsilon}(\mathbf{w}_h^i) - p_h \nabla \cdot \mathbf{w}_h^i] d\Omega + \langle D\mathbf{q}E(t_*, \mathbf{q}_*), \mathbf{N}^i \rangle + \\ & + \delta t S(\mathbf{q}_H^n; \hat{\mathbf{q}}_H^{n+1}, \mathbf{N}^i) - \langle \mathbf{F}(t_*), \mathbf{N}^i \rangle, \end{aligned} \quad (80)$$

where now the implementation of all the terms has been made precise. The residual function `residualFSI()` has to be provided to the nonlinear solver. To find  $\underline{\alpha}$  we use the function `root()` from the `scipy.optimize` library (see [60]) within the loop over time steps as follows

```
def residualFSI(alpha,
                mesh, hk, v, q, sol, wh, Ndofs, dt,
```

```

        eref, espont, fe,
        qHn, qHdot, deltaqH, bcs): # Eq. (80)
u,p = split(sol)
for i in range(Ndofs):
    u += Constant(alpha[i])*wh[i]

u,p = SolveFluid(u, p, v, q, hk, sol, bcs)
for i in range(Ndofs):
    u += Constant(alpha[i])*wh[i]

for i in range(Nnodrod): # Build rod velocity based on alpha's
    qHdot.dat.data[2*i][0] = alpha[4*i+0]
    qHdot.dat.data[2*i][1] = alpha[4*i+1]
    qHdot.dat.data[2*i+1][0] = alpha[4*i+2]*h_r
    qHdot.dat.data[2*i+1][1] = alpha[4*i+3]*h_r

mu = viscosity(u)
for i in range(Ndofs): # Testing against the wh[:]
    res_fsi[i] = assemble(inner(2*mu*sym(grad(u)),grad(wh[i]))*dx)
    res_fsi[i]-= assemble(div(wh[i])*p*dx)
    res_fsi[i]+= RodTerms(dt, eref,espont,fe, qHn,qHdot,deltaqH[i])
return res_fsi

...
# Begin loop over time steps
for n in range(ntimes):
    qHn.assign(qH)
    espont = SpontaneousStrain(tt, dt, x_r)
    fe = ExternalForce(tt, dt, x_r)
    wh = ApplyHhatqN(W_f, xref, list_nodes_wet, qHn)
    solnl = root(residualFSI,
        alpha,
        args = ((mesh_f, hk, v, q, sol_f, wh, Ndofs, dt,
            eref, espont, fe, qHn, qHdot, deltaqH, bcs_f)),
        method = 'krylov',
        options = {'disp': True, 'fatol': 1e-6,
            'maxiter': 5, 'line_search': 'armijo',
            'jac_options': {'inner_tol': 1e-12,
                'inner_maxiter': 100000,
                'method': 'gmres',
                'rdiff': 1e-4}})

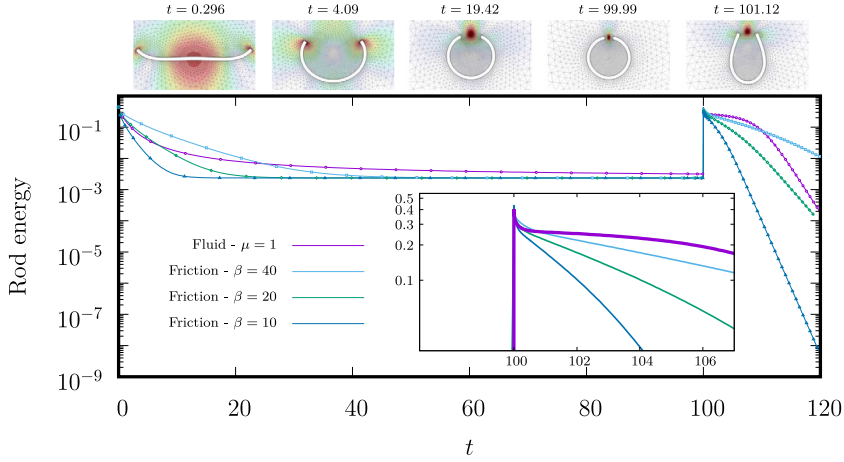
    alpha = solnl.x
    ...
    qH.assign(qHn + Constant(dt)*qHdot)
    new_wet = UpdateWetBoundary(xref, qH)
    for i in range(nnod_wet): # Move the wet boundary
        x_f[list_nodes_wet[i],:] = new_wet[i,:]

    if(need_remesh):
        new_mesh_f, ... = RemeshFluidRegion ( mesh_f, ...)
    ...

```

In the snippet above, the code has been slightly simplified so as to point out to the important features of the implementation. The solution is sought by using a Newton–Krylov method [61]. Being an inexact Newton method, convergence speed may depend on the choice of algorithmic parameters. In the numerical experiments below, the set of parameters shown in the code have worked well in general and can be recommended. Finally, results can be visualized with Paraview [62] as the simulation runs simply by opening some pvd files created to output the velocity and pressure fields. In the case of a nonlinear fluid an additional file to output the viscosity is also saved. Additional information, such as the rod elastic energy, the fluid dissipation and the geometric position of the left end of the rod as a function of time are written to an ascii file called `evol.txt`.

The source code that solves the last numerical experiment to be presented in the next section, namely, the swimming of a flexible rod immersed in a viscous fluid is provided. The code can be found in <https://gitlab.com/rfausas/microswimmers>. `git` along with details on the exact firedrake version, its dependencies used in this work and some useful information to run the script (see README). All experiments to be presented hereafter were run into a desktop computer with Core i7-7700K CPU @ 4.2 GHz with Linux Ubuntu 18.04.3 LTS.



**Fig. 2.** Energy as a function of time for the roll-up and release of a pre-stressed rod. The purple curve corresponds to the fluid–structure-interaction model proposed in this article, with the fluid being Newtonian with  $\mu = 1$ . The other curves consider a simpler model, in which the fluid is replaced by a friction force  $-\beta \mathbf{u}$  with  $\beta = 10, 20$  and  $40$ . The inserts show the geometry and fluid mesh at several instants, colored with the velocity magnitude.

### 4. Numerical experiments

#### 4.1. Roll-up and roll-out of a flexible rod

The first experiment we consider consists of the roll-up and the subsequent roll-out of an initially straight rod of unit length. This is accomplished by setting the spontaneous curvature  $\kappa_0$  to

$$\kappa_0(s, t) = \begin{cases} -0.999 \times 2\pi & \text{if } t < 100 \\ 0 & \text{otherwise} \end{cases} \tag{81}$$

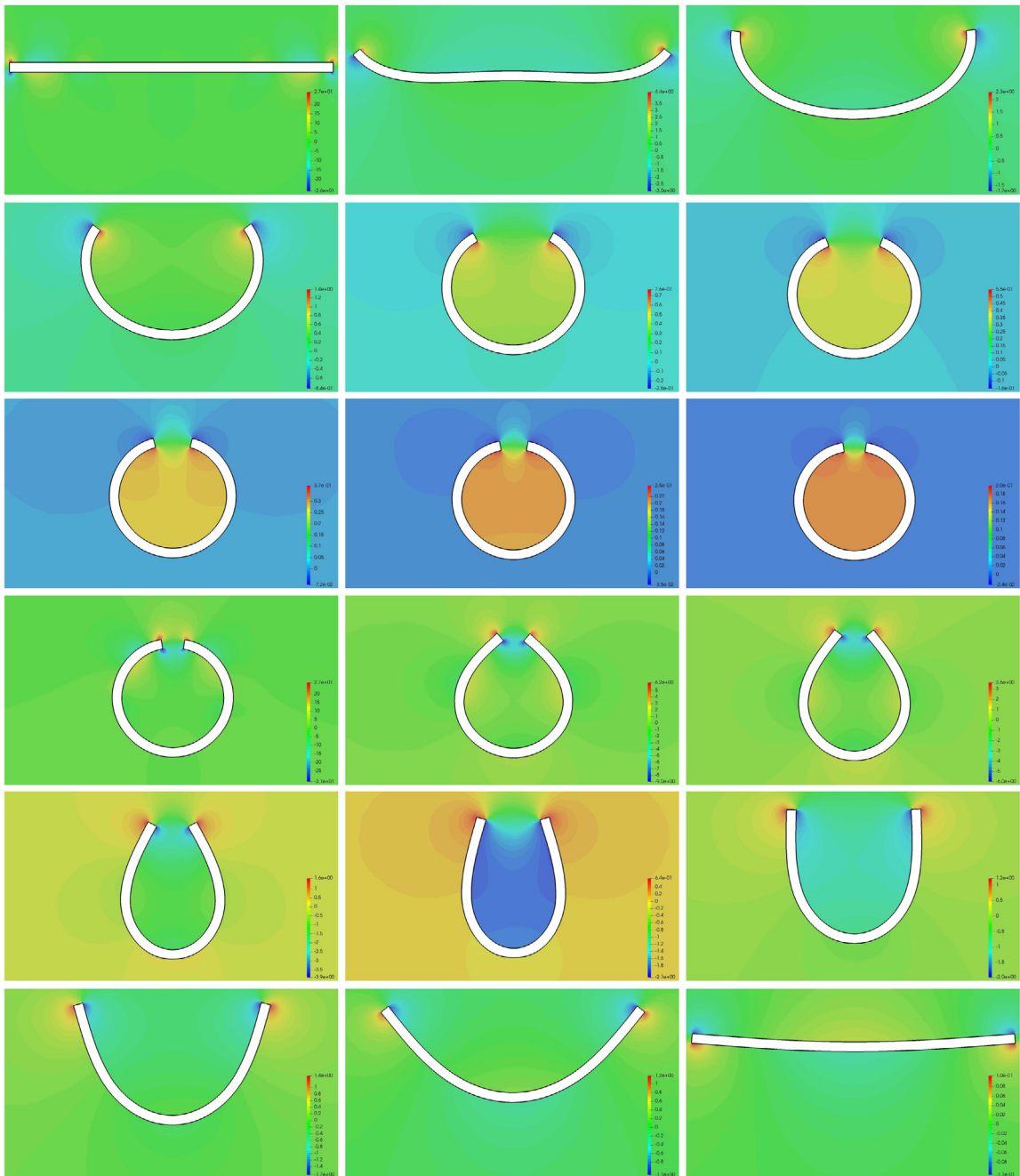
i.e., the initially straight rod rolls towards a circular rod of radius  $R = \frac{1}{0.999 \cdot 2\pi}$  (i.e., almost to closure)<sup>3</sup> and at  $t = 100$  the rod is released. We take  $C_\epsilon = 90$  and  $C_\kappa = 0.0225$  and  $a(s) = \frac{1}{\|\mathbf{q}_t^i(s)\|}$ . Both ends of the rod are flat. The computational domain is the square region  $\Omega = [0, 3]^2$ . Both lateral walls are open (traction free) and the bottom and top walls are subject to zero-velocity boundary conditions. The rod is discretized with  $N_{el} = 4$  Hermite elements and the fluid mesh has elements of characteristic size  $h = 1/100$  close to the rod’s boundary. Both stages of the process are shown in Fig. 2 that displays the rod energy as a function of time. Initially, the rod relaxes to a minimum energy state ( $\mathcal{E} \sim 2.3 \times 10^{-3}$ ). This minimum is not zero because the circular shape cannot be represented exactly by cubic polynomials. At  $t = 100$  the pre-stress is released and a second relaxation process takes place until the initial straight shape is recovered, exactly. For this problem the time step is adaptively chosen so as to better capture the different time scales observed along the simulation, specially in the initial stages ( $t < 1$ ) and immediately after release  $t \geq 100$ . Inserts with plots of the rod shape and the triangular fluid mesh generated are shown at different times, including the rod just prior to release and just after it. In Fig. 2 we have also included the behavior of a simpler model of rod, in which the ambient fluid is substituted by a friction force  $-\beta \mathbf{u}_h$  proportional to the rod’s surface velocity. To be precise, the variational residual of this simpler model reads

$$\mathcal{R}^i(\underline{\alpha}) = \int_{S(\mathbf{q}_h^n)} \beta \mathbf{u}_h \cdot \mathbf{w}_h^i dS + \langle D_{\mathbf{q}} E(t_*, \mathbf{q}_*), \mathbf{N}^i \rangle + \delta t S(\mathbf{q}_H^n, \dot{\mathbf{q}}_H^{n+1}, \mathbf{N}^i) - \langle \mathbf{F}(t_*), \mathbf{N}^i \rangle = 0 . \tag{82}$$

The plot shows results corresponding to  $\beta = 10, 20$  and  $40$ . One thing to notice is that the dynamical behavior of the simpler model differs from that of the true fluid structure interaction problem, specially in the released phase. This shows the importance of the complete FSI treatment being proposed in this work. Snapshots of the complete rod evolution together with contours of pressure are shown in Fig. 3, the first three rows of frames corresponding to the pre-stressed phase and the last three to the released phase. The dynamics and intermediate shapes of these two phases are appreciably different.

Let us now assess the convergence properties of the scheme by pre-stressing the rod with  $\kappa_0 = -\pi$ , such that equilibrium shape corresponds to half of a circle of radius  $1/\pi$ . We compute the curvature and the positional errors

<sup>3</sup> The closure of the rod is avoided to prevent the change of mesh’s topology.



**Fig. 3.** Shape evolution and pressure field for the roll-up and release of a pre-stressed rod immersed in a viscous fluid. The instants of the frames above are  $t = 0, 0.296, 1.698, 4.089, 11.42, 19.42, 37.42, 67.42, 99.99, 100, 100.07, 100.2, 100.5, 103.0, 107.2, 109.0, 111.0, 119.0$ , respectively.

as

$$e_\kappa = \left( \int_0^1 \kappa^2 ds \right)^2 \quad \text{and} \quad e_p = \int_0^1 \left| (\tilde{\mathbf{q}}_H \cdot \tilde{\mathbf{q}}_H)^{\frac{1}{2}} - \pi^{-1} \right| ds \tag{83}$$

with  $\tilde{\mathbf{q}}_H = \mathbf{q}_H - \mathbf{r}_c$  being  $\mathbf{r}_c = \frac{1}{2} (\mathbf{q}_H(s=0) + \mathbf{q}_H(s=1))$ . Fig. 4 shows these errors as a function of the mesh parameter  $H$  at the final time of the relaxation. The convergence rate for the curvature error  $e_\kappa$  is  $\mathcal{O}(H^2)$  and for the positional error  $e_p$  is  $\mathcal{O}(H^4)$ , as expected for Hermite cubic polynomials [15]. To conclude this subsection, the convergence properties of the

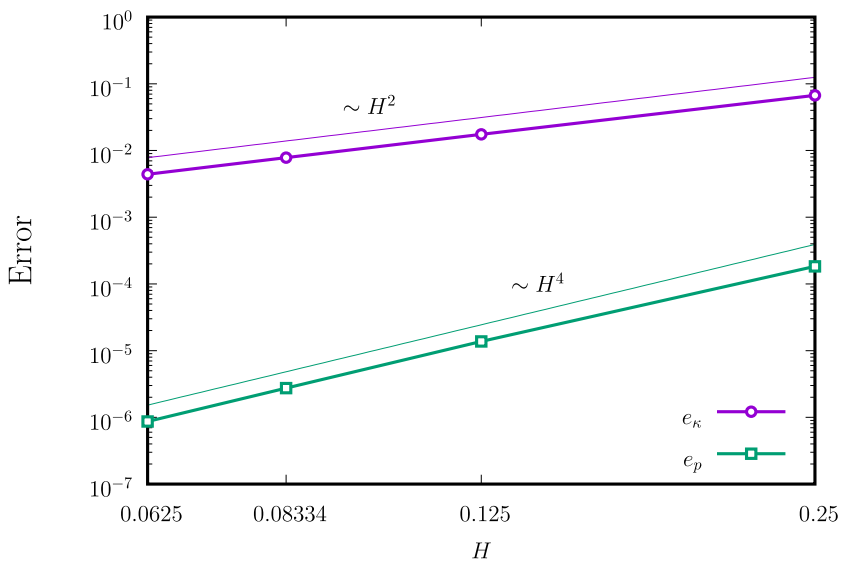


Fig. 4. Curvature and positional error for the roll-up of a pre-stressed rod immerse in a viscous fluid.

Table 2

Contributions of the different terms in the discrete energy balance.

$\delta t$	$\int_{t_i}^{t_f} \int_{\Omega} 2\mu \nabla^S \mathbf{u}_h : \nabla^S \mathbf{u}_h \, d\Omega \, dt$	$-\int_{t_i}^{t_f} \int_{\Omega} p_h \nabla \cdot \mathbf{u}_h \, d\Omega \, dt$	$\int_{t_i}^{t_f} \delta t S(\cdot, \cdot, \cdot) \, dt$	$\mathcal{E}(t_f) - \mathcal{E}(t_i)$
0.04	0.037543	0.00103	0.003307	-0.042951
0.02	0.038663	0.00105	0.001369	-0.041734
0.01	0.039334	0.00107	0.000702	-0.041462
0.005	0.039705	0.00108	0.000359	-0.041335
0.0025	0.039905	0.00109	0.000182	-0.041273

semi-implicit scheme with respect to time step refinement is also assessed. To that end, we compute a discrete energy balance error as

$$e_{\mathcal{E}} = \left| \int_{t_i}^{t_f} R_{\mathcal{E}}(t) dt \right|, \tag{84}$$

in which

$$R_{\mathcal{E}}(t) = \frac{d\mathcal{E}}{dt} + \int_{\Omega(\mathbf{q}_H)} (2\mu \nabla^S \mathbf{u}_h : \nabla^S \mathbf{u}_h - p_h \nabla \cdot \mathbf{u}_h) \, d\Omega + \delta t S(\mathbf{q}_H; \dot{\mathbf{q}}_H, \dot{\mathbf{q}}_H) \tag{85}$$

is the discrete power imbalance, namely the rate of energy change (first term of  $R_{\mathcal{E}}$ ) plus fluid dissipation (second term) plus the dissipation produced by the stabilization (third term). Notice that the  $p_h \nabla \cdot \mathbf{u}_h$  term in the instantaneous energy balance (85) is zero in the exact problem and also in the discrete problem if  $\tau_h = 0$ , but not in our  $P_1/P_1$  implementation. In computing (85),  $\frac{d\mathcal{E}}{dt}$  is taken as  $\frac{\mathcal{E}(t_n) - \mathcal{E}(t_{n-1})}{\delta t}$  and all other quantities are taken at instant  $n$ . Also,  $t_i = 0.04$  and  $t_f = 0.8$ . The error  $e_{\mathcal{E}}$  for different time steps  $\delta t$ , namely, 0.04, 0.02, 0.01, 0.005 and 0.0025 is displayed in Fig. 5, showing the expected convergence rate  $\mathcal{O}(\delta t)$ . To quantify the relative importance of the different contributions to the energy balance (85) we show in Table 2 the integral of each term in the same time interval,  $(t_i, t_f)$ . The main contributions are the physical ones, namely the viscous dissipation and the time derivative of  $\mathcal{E}$  (second and fifth columns) which approach each other as the time step is refined. The semi-implicit stabilization term (fourth column) is smaller and  $\mathcal{O}(\delta t)$ . The pressure term (third column) is also rather small but independent of  $\delta t$ , since in fact it is a consequence of the discretization of the fluid problem and thus depends on  $h$ .

#### 4.2. Large deflection of a flexible beam in a channel

In the second experiment we consider a cantilever beam of initial length  $S = 1$  and thickness  $e = 0.03$  whose left end is clamped at  $(x_1, x_2) = (0, \frac{3}{2})$  in the computational domain  $[0, 3]^2$ . The boundary conditions at the left end of the rod are thus  $\mathbf{q} = (0, \frac{3}{2})$  and  $\frac{dq_2}{ds} = 0$ . The right end of the beam is rounded and free. The material parameters are  $C_{\epsilon} = 90$  and  $C_{\kappa} = 0.0225$  as in the previous experiment. The fluid is assumed to be Newtonian with viscosity  $\mu = 1$ . Also, we consider the function  $a(s) = 1$  (see Eq. (65)). The beam is discretized with  $N_{el} = 8$  elements and the fluid mesh with elements

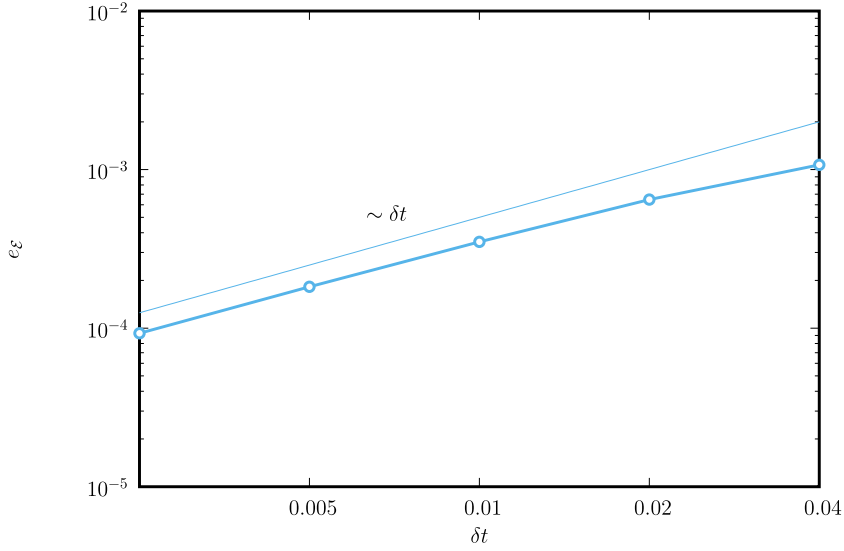


Fig. 5. Energy balance error for the roll-up of a pre-stressed rod immersed in a viscous fluid.

having characteristic size  $h = 1/100$  near the beam boundary, which gives meshes with varying number of elements, that oscillates between  $\sim 5K$  to  $\sim 6.5K$ . The time step  $\delta t$  is set to the constant value 0.05 throughout the simulation. We impose a non-uniform (constant in time) spontaneous curvature given by

$$\kappa_0(s) = 2\pi (1 + \tanh(4s)) . \tag{86}$$

As a result of this, the beam starts to deform into a scroll shape from  $t = 0$  until  $t = 12$  when a fluid inlet located on the lower part of the left boundary is suddenly opened. The fluid velocity profile at the inlet is set to be the function

$$\mathbf{u}(x_1 = 0, x_2) = ((3 - e)x_2^3 - 2x_2^4, 0) , \quad 0 \leq x_2 \leq \frac{3 - e}{2} . \tag{87}$$

The influx of the fluid, which leaves through the upper boundary, unrolls the beam and pushes it upwards, resulting in a complex process with large displacements and rotations and strong fluid–beam interaction. The bottom wall ( $x_2 = 0$ ), the right wall ( $x_1 = 3$ ) and the part of the left wall above the beam ( $x_1 = 0, x_2 > \frac{3+e}{2}$ ), are permanently subject to zero-velocity boundary conditions. The top wall ( $x_2 = 3$ ) is an outlet boundary through which the fluid is free to leave the domain.

The complete deformation process is plotted in Fig. 6 that shows the pressure field and arrows of the fluid velocity. Due to the large difference in magnitude, the velocity arrows in the roll-up phase (first two rows of frames) are scaled by the factor 0.15, while in the roll-out phase (the last two rows of frames) this factor is 0.035. The beam attains a stationary equilibrium shape at the final time of the simulation as the result of the balance between the internal beam forces and the interaction with the ambient fluid. Also, Fig. 7 displays details with LIC (Line Convolution Integral) streamlines produced by Paraview 5.4.1, colored by the velocity magnitude at some selected instants.

### 4.3. Swimming of a finite-length flexible rod

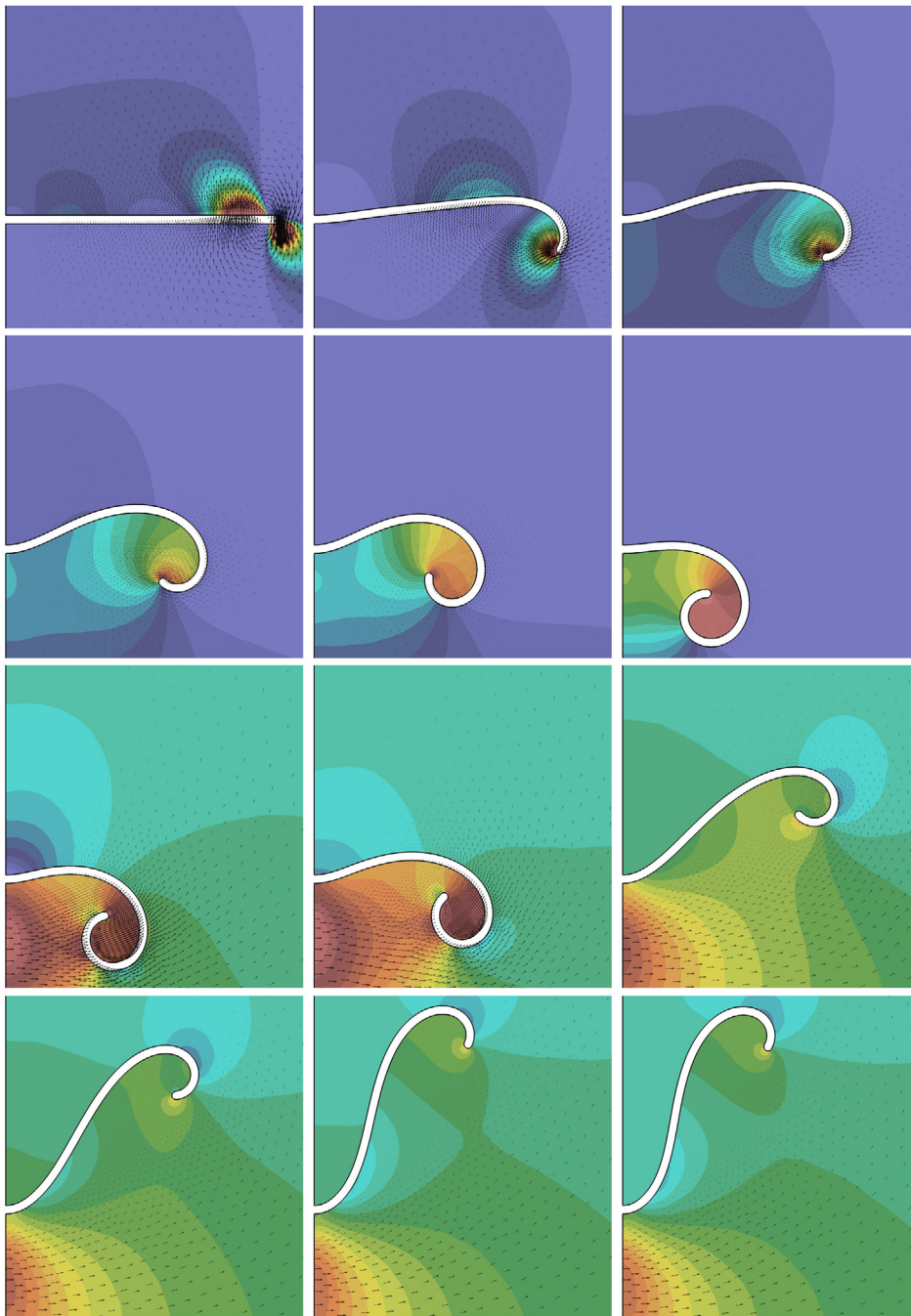
In this last experiment a swimming flexible rod immersed in a viscous fluid is studied. The computational domain is the square region  $\Omega = [0, 3]^2$ . Both lateral walls are open and the bottom and top wall are subject to zero-velocity boundary conditions. The rod shown in Fig. 1 is considered, with  $e = 0.03$ ,  $D_c = 3e$ ,  $H_c = 3e$ , and unit initial length. The elastic parameters are taken to be  $C_\epsilon = 90$  and  $C_\kappa = 0.0225$ .

First, the fluid is assumed to be Newtonian with viscosity  $\mu = 1$ . By taking the spontaneous curvature in Eq. (44) to be the wave

$$\kappa_0(s, t) = 20 \sin(4\pi(s - 2t)) , \tag{88}$$

the rod swims as seen in Fig. 8, which shows the rod’s shape together with contours of pressure and velocity magnitude at different times, namely,  $t = 0.0, 1.0, 2.4, 3.2$ . Note that by setting  $\kappa_0(s, t)$  we are not imposing the kinematics of the rod. Instead, the rod’s configurations result from the interaction with the ambient fluid as the rod tends to relax to its instantaneous minimum energy state. The time step adopted for all cases in this experiment is  $\delta t = 0.01$ , constant throughout the simulation. The rod is discretized by 8 elements and the fluid by a triangular mesh with characteristic

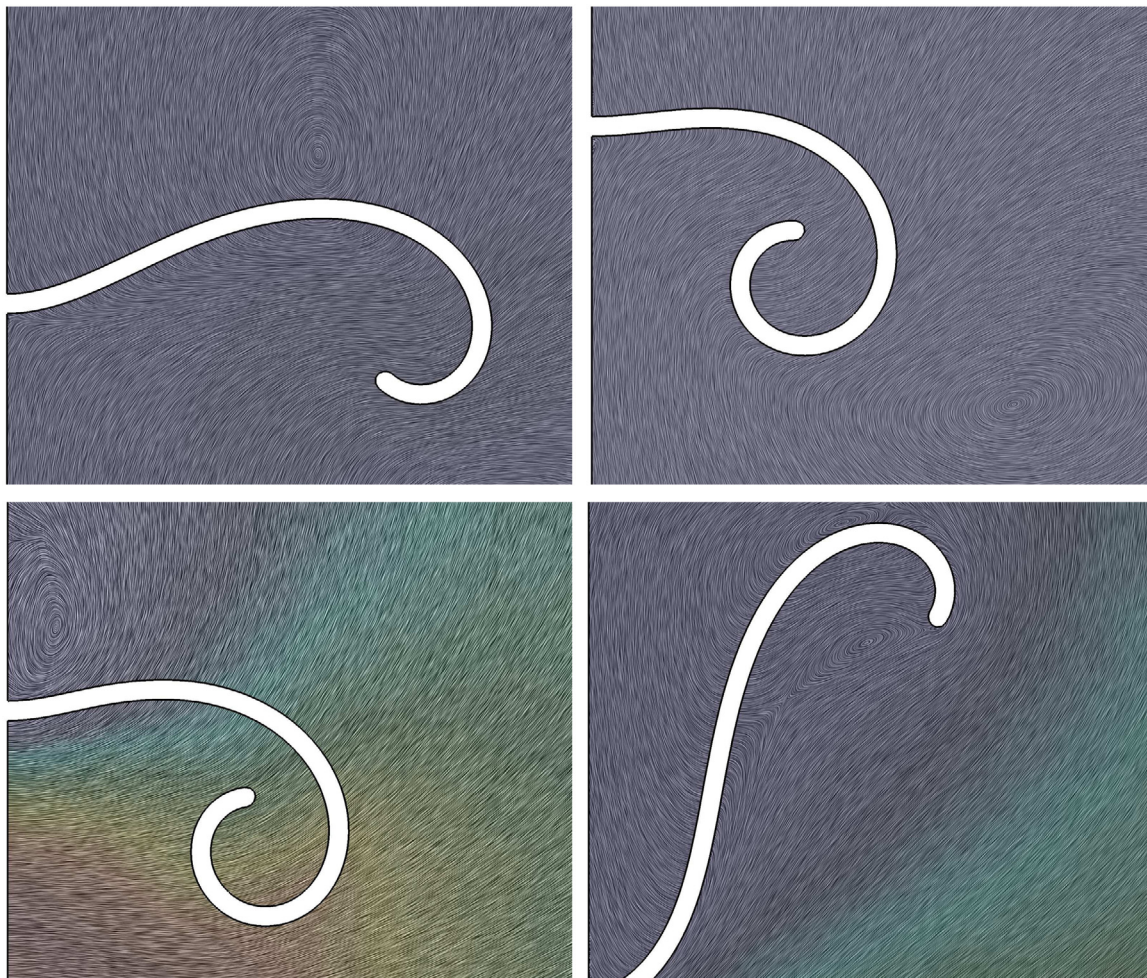




**Fig. 6.** Shape, contours of pressure field and velocity vectors for the cantilever rod subject to a pre-stress and a velocity field switched on from time  $t = 12$ . Pressure scale: 4.1 (red),  $-0.11$  (blue) for  $t < 12$  (first two rows) and 9.7 (red),  $-4.7$  (blue) for  $t \geq 12$  (last two rows). Velocity vector scale: 0.15 ( $t < 12$ ) and 0.035 ( $t \geq 12$ ). Shown are the instants  $t = 0, 0.5, 1.25, 2.5, 5, 11.95, 12.1, 12.4, 14, 16, 25, 30$ .

element size  $h = 0.01$  close to the swimmer's boundary. Notice that far away from the swimmer the mesh is coarser. Remeshing of the fluid domain is being executed at each time step, the total number of elements in the triangulation oscillates about 5K. This is almost mandatory due to the large time step being used in the simulation. We observe that as a result of this periodic deformation with large amplitude the rod exhibits a net displacement. As previously stated we consider  $a(s) = \frac{1}{\|\mathbf{q}_t(s)\|}$  (see Eq. (65)), which penalizes to some degree extensional deformations of the swimmer. Recall





**Fig. 7.** LIC Streamlines colored by the velocity magnitude for the cantilever rod subject to a pre-stress and a velocity field switched on at time  $t = 12$ . Velocity scale: 1.2 (red), 0.3 (cyan) and 0 (dark blue). Shown are details corresponding to instants  $t = 2.5, 11.95, 12.1, 30$ .

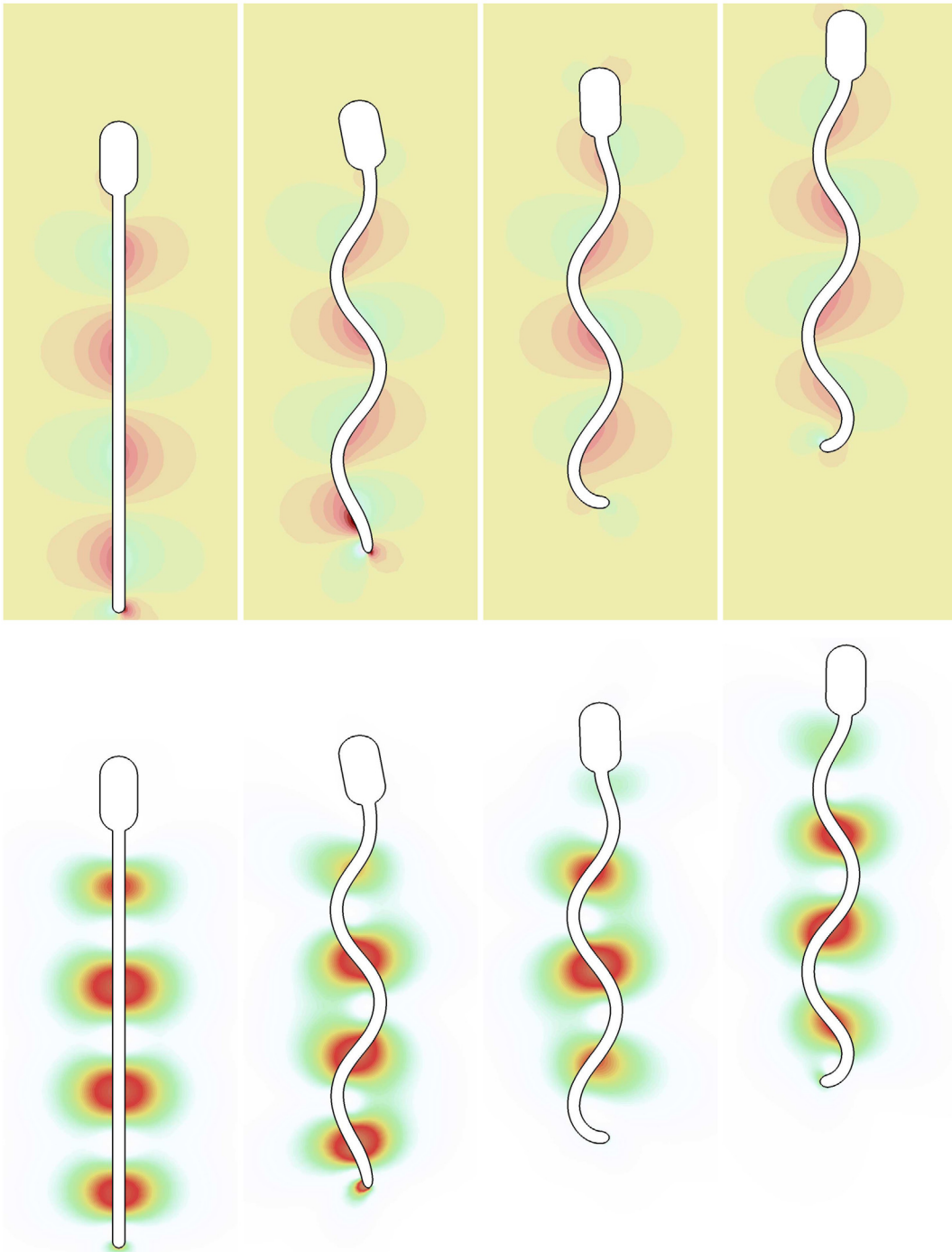
that, according to Eq. (66) all points belonging to the swimmer’s head are attached to the end point of the rod making the head to be aligned with the rod axis at all times.

To assess the impact of the discretization parameters on the rod’s swimming performance, Fig. 9 shows the swimmer and the fluid mesh at times  $t = 1.0, 1.8, 2.4, 3.2$  (from bottom to top), and rod resolutions  $H = 1/4, 1/6, 1/8, 1/12$  (from left to right). It is interesting to note that the swimmer’s head experiences some stretching, specially for the coarsest level  $H = 1/4$ . The level of discretization also affects the net displacement of the swimmer as shown in Fig. 10. The main conclusion that can be drawn from these results is that the deformation of the rod is surely well represented by a mesh consisting of at least 8 elements, the results being quite similar for finer discretizations.

Next, Fig. 11 shows results for different fluid mesh refinements close to the wet boundary of the deformable rod, namely, from left to right,  $h = 1/50, 1/100, 1/200$  (corresponding to fluid meshes with about 3K, 5K and 10K elements, respectively). For the rod we consider a fixed resolution  $H = 1/8$ . Contours of pressure and the fluid mesh are shown. The net displacement of the rod for this case is plotted in Fig. 12. As one can notice the results are almost independent of the fluid discretization for meshes finer than  $h = 1/100$ .

Let us take advantage of this test case to compare the different options for time discretization that have been introduced in the previous section, namely, the semi-implicit, the explicit and the pseudo-implicit schemes. In all previous tests we have only shown the results of the semi-implicit method, which is in fact the one with the best performance, but it is instructive to report at least one quantitative comparison.

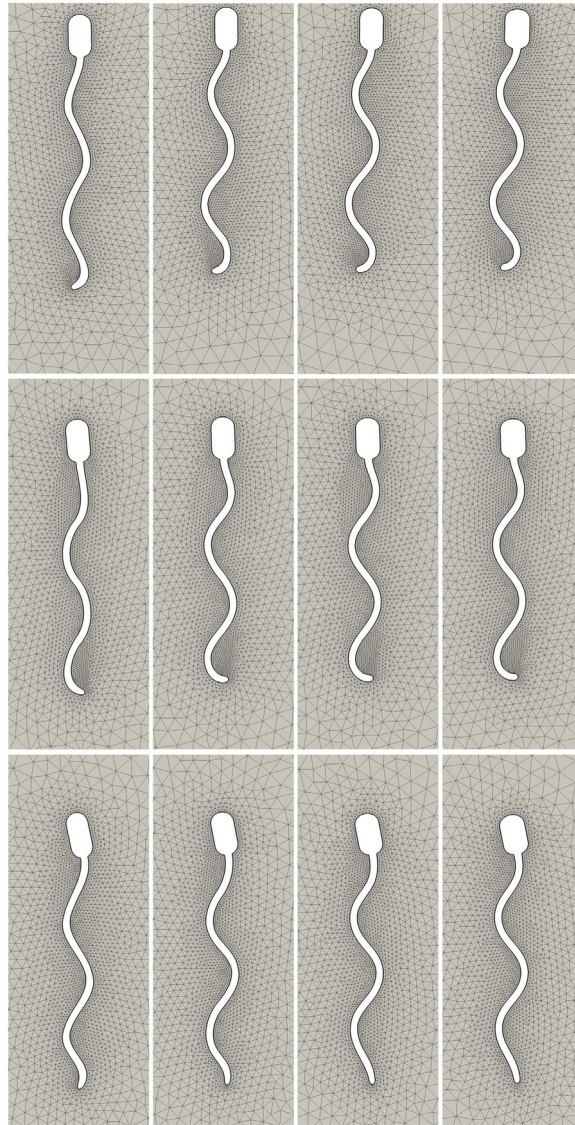
A sensitive variable to compare is the viscous dissipation  $\int_{\Omega_f} 2\mu \nabla^S \mathbf{u}_h : \nabla^S \mathbf{u}_h$ , which is shown in Fig. 13 as a function of time for all methods and different time steps. The explicit method is unstable for  $\delta t > 10^{-3}$ , while with the others we can take  $\delta t$  much larger without algorithmic crash. The results with all methods seem to show convergence to the same solution as  $\delta t \rightarrow 0$ . The semi-implicit and pseudo-implicit methods provide good approximations for  $\delta t = 10^{-2}$ . However,



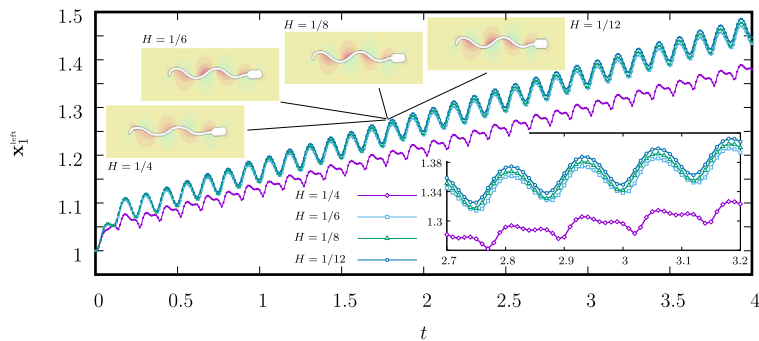
**Fig. 8.** Contours of velocity magnitude (top) and pressure (bottom) for a swimmer in a Newtonian fluid at different times (from left to right,  $t = 0.0, 1.0, 2.4, 3.2$ ). The rod is discretized with  $H = 1/8$  and the fluid with typical refinement  $h = 1/100$ .

notice that the pseudo-implicit method requires several non-linear iterations (typically 3 to 5) to achieve convergence, which the semi-implicit, being a linear scheme, does not require. Results are quite poor for both, the semi-implicit and pseudo-implicit methods for  $\delta t = 2.5 \times 10^{-2}$ . The comparison is complemented with Fig. 14, which assesses the impact of the temporal scheme and time step on the swimmer's advance. The explicit and semi-implicit schemes for  $\delta t = 10^{-3}$  deliver very similar results (green squares and light blue diamonds, respectively). Finer time steps produce results that are undistinguishable from those. The semi-implicit and the pseudo-implicit schemes for  $\delta t = 10^{-2}$  (yellow triangles

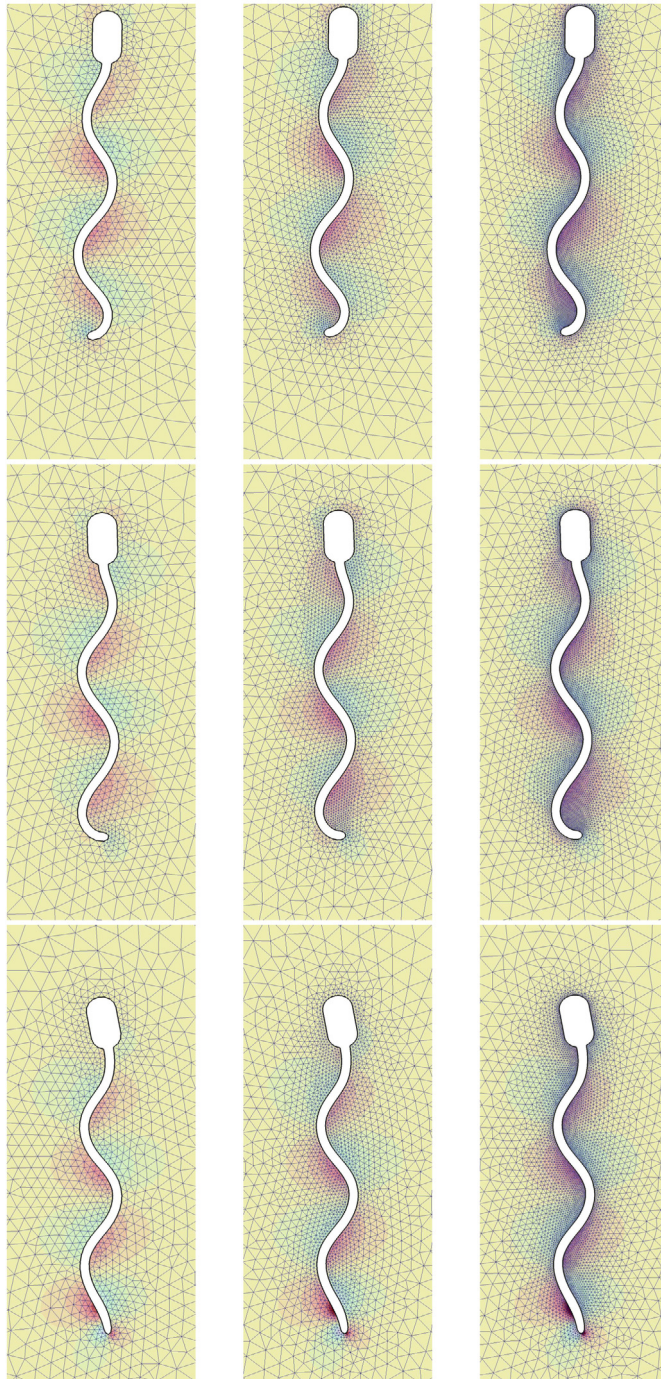




**Fig. 9.** Fluid mesh for a swimmer in a Newtonian fluid with typical refinement  $h = 1/100$  at different times (from bottom to top,  $t = 1.0, 1.8, 3.2$ ). The rod is discretized with different number of elements (from left to right,  $H = 1/4, 1/6, 1/8, 1/12$ ).

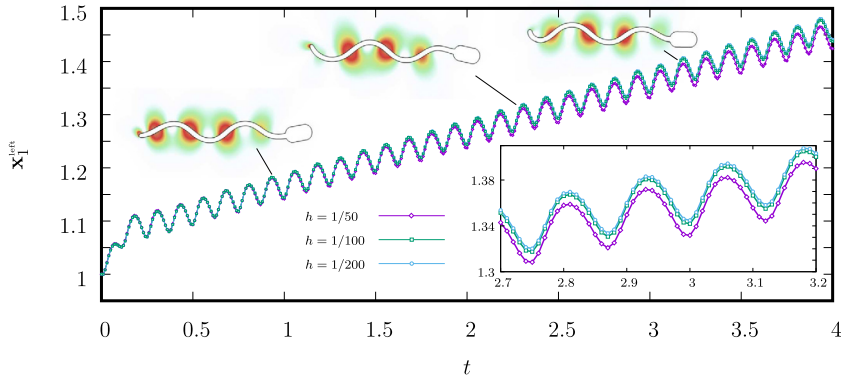


**Fig. 10.** Swimmer position as a function of time. The fluid is discretized with mesh refinement  $1/100$ . The rod is discretized with different number of elements  $H = 1/4$  (purple),  $H = 1/6$  (light blue),  $H = 1/8$  (green),  $H = 1/12$  (dark blue).

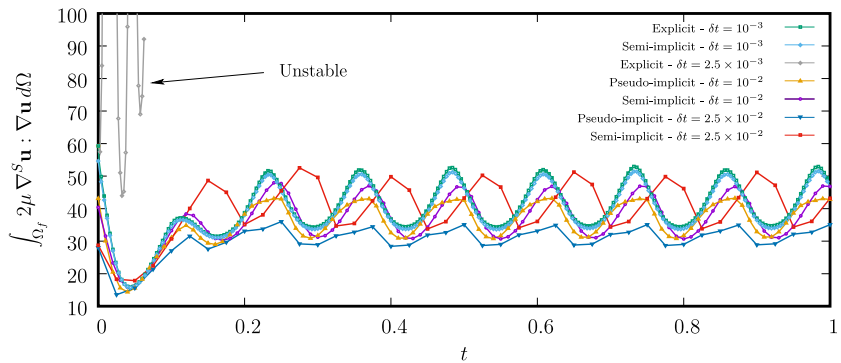


**Fig. 11.** Contours of pressure and mesh for a swimmer in a Newtonian fluid at different times (from bottom to top,  $t = 1.0, 1.8, 3.2$ ). The rod is discretized with  $H = 1/8$  and the fluid with different mesh refinements (from left to right,  $h = 1/50, 1/100, 1/200$ ).

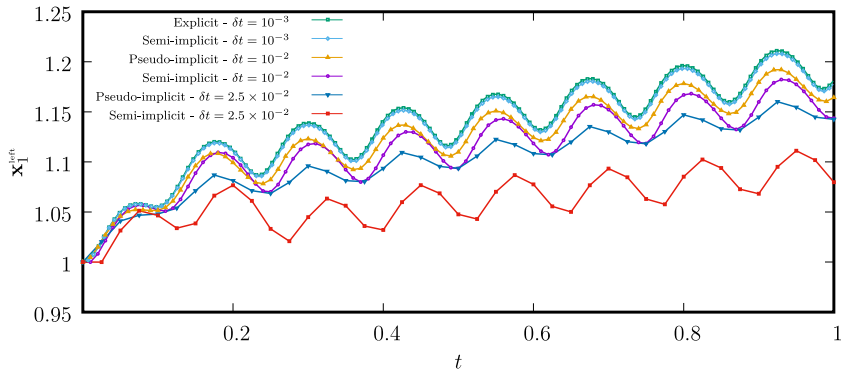
and magenta circles, respectively) produce reasonably accurate results, although some differences can be noticed. There appears a phase error in the advance of the swimmer while the average velocity after the initial transient is essentially correct. For  $\delta t = 2.5 \times 10^{-2}$  the error worsens, but still the main features of the swimmer's movement persist, however, the semi-implicit method (red solid squares) clearly exhibits a more dissipative behavior than the pseudo-implicit one (blue solid triangles).



**Fig. 12.** Swimmer position as a function of time. The rod is discretized with  $H = 1/8$  and the fluid is discretized with different mesh refinements  $h = 1/50$  (purple),  $1/100$  (green) and  $1/200$  (blue).



**Fig. 13.** Fluid dissipation as a function of time for the semi-implicit, explicit and pseudo-implicit schemes and different time steps.



**Fig. 14.** Swimmer position as a function of time for the explicit, pseudo-implicit and semi-implicit and different time steps.

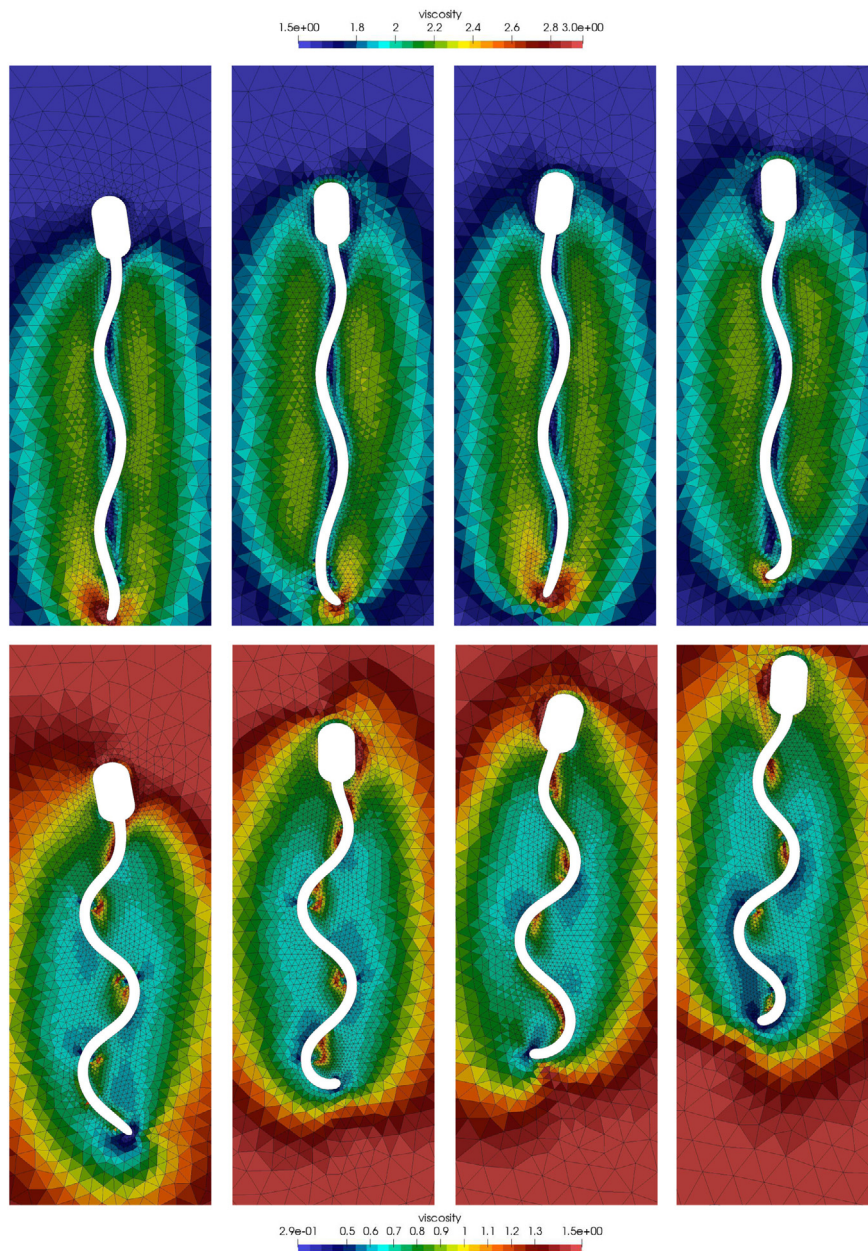
Summarizing, the semi-implicit method with intermediate time steps is a suitable and non-expensive choice that captures the swimming of finite length flexible rods in viscous fluids.

To conclude, we consider the case of a non-Newtonian fluid behavior. To that end, a Carreau–Yasuda [63,64] viscosity law is considered, namely,

$$\mu(\dot{\gamma}) = \eta_{\infty} + (\eta_0 - \eta_{\infty})(1 + 2\lambda \dot{\gamma}^2)^{\frac{r-1}{2}} \tag{89}$$

where  $\dot{\gamma} = \sqrt{\nabla^S \mathbf{u} : \nabla^S \mathbf{u}}$ ,  $\eta_0 = 1.5$ ,  $\eta_{\infty} = 10^{-3}$ ,  $\lambda = 1$  and  $r$  is a power index. A shear thinning or shear thickening behavior is acquired depending on whether  $r < 1$  or  $r > 1$ , respectively. The swimmer is shown at different times in Fig. 15. The bottom part corresponds to the shear thinning case with  $r = 0.7$ , whilst the top part corresponds to the shear thickening case with  $r = 1.15$ . The rod is discretized with 8 elements and the fluid mesh with elements of





**Fig. 15.** Contours of viscosity and mesh for a swimmer in a Carreau–Yasuda fluid at different times (from left to right,  $t = 1.0, 1.8, 2.4, 3.2$ ). The top row corresponds to shear thickening (power index  $r = 1.15$ ) and the bottom row to shear thinning (power index  $r = 0.7$ ). The rod is discretized with  $H = 1/8$  and the fluid with typical refinement  $h = 1/100$ .

characteristic size  $h = 1/100$  close to the wet boundary. Fig. 15 also shows the fluid viscosity, which is an elementwise constant field, since linear elements are being used for discretization of the Stokes problem. Note that the swimmer exhibits a significantly larger deformation and bigger net displacement in the shear thinning fluid as compared to the shear thickening one. Nevertheless, the aim at this point is not to perform a fair comparison of the swimmers' performance attained in each case, since the effective viscosity can be quite different, but to show the complex viscosity patterns that appear as the swimmers deform, which are plainly handled by the proposed formulation. At this point, let us recall that the fluid nonlinearity is solved at each evaluation of the residual function by means of a Newton–Raphson procedure with line search available in the SNES PETSc through the Firedrake interface.

## 5. Concluding remarks

This work proposes a fluid–structure interaction framework for Cosserat rods immersed in flows exhibiting complex rheologies. The inertia of the rod and of the fluid are assumed negligible, as is typical in microsystems. Such an approach is suitable to study soft-bio-matter problems, e.g., elongated microorganism or appendices such as flagella and/or cilia. To showcase the promising features of the proposed approach, we specialize the new framework to the case of a planar non-shearable rod surrounded by an incompressible fluid with generalized Newtonian rheology.

As the chosen mechanical model relies on a configuration space that avoids shearing by construction, the deformation measures as well as the position of the wet surface depend only on the centroidal curve of the rod. Although we employed the simplest possible elastic energy for the rod, it is also possible to consider more general hyperelastic constitutive models. Due to that, a discrete version of the model (in the sense of the finite element method) can be easily built and handled within the Firedrake platform, taking full advantage of its capabilities for modeling complex systems. In addition, the source code, which is made available alongside this manuscript, is lean and quite intuitive. The simplicity of the proposed framework for modeling active distributed devices/organisms, as needed to investigate locomotion, are noteworthy.

The flexibility of Firedrake for modeling complex fluids allows to investigate fluid–structure interactions involving shear-thinning or shear-thickening Newtonian models. Nonlinear constitutive behaviors are rarely addressed in the literature and cannot be treated with other methods such as boundary elements. The fluid's rheology, however, is sometimes essential to model elongated microorganisms immersed in biological fluids [9].

Though the reported work is rather extensive, there are many possible directions for further developments. The interaction among several elongated appendices, the coupling with the transport equation to consider the absorption of nutrients or the release of substance such as toxins or similar, more general constitutive models for the swimmer's elastic behavior as well as the optimization or self-learning of swimming trajectories are just a very few examples among a plethora of interesting applications that can be considered in future works. Another possible research direction is related to the derivation and implementation of time integration schemes of higher order. This is indeed ongoing work. Finally, the code is made freely available for interested colleagues to download and adapt to their own goals.

## CRedit authorship contribution statement

**Roberto Federico Ausas:** Conceptualization, Methodology, Software, Investigation, Writing – original draft, Writing – review & editing. **Cristian Guillermo Gebhardt:** Conceptualization, Methodology, Formal analysis, Investigation, Writing – original draft, Writing – review & editing. **Gustavo Carlos Buscaglia:** Conceptualization, Methodology, Formal analysis, Investigation, Writing – original draft, Writing – review & editing.

## 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.

## Acknowledgments

Roberto Federico Ausas and Gustavo Carlos Buscaglia gratefully acknowledge financial support from the São Paulo Research Foundation FAPESP and from the Conselho Nacional de Desenvolvimento Científico e Tecnológico (grants CEPID-CeMEAI 2013/07375-0 and INCT-MACC).

The authors acknowledge as well the University of Bergen for the open access funding.

Finally, the authors thank to D. Ham and K. Sagiyama from the Firedrake project and P. Farrell for giving some guidelines in the initial stages of the development.

## References

- [1] Lé B, Legrain G, Moës N. Mixed dimensional modeling of reinforced structures. *Finite Elem Anal Des* 2017;128:1–18.
- [2] Steinbrecher I, Mayr M, Grill M, Kremheller J, Meier C, Popp A. A mortar-type finite element approach for embedding 1D beams into 3D solid volumes. *Comput Mech* 2020;66:1377–98.
- [3] Bertails F, Audoly B, Cani M-P, Querleux B, Leroy F, Lévêque J-L. Super-helices for predicting the dynamics of natural hair. *ACM Trans Graph* 2006;25:1180–7.
- [4] Knoch P, Bonanni U, Magnenat-Thalmann N. Hair simulation model for real-time environments. In: *Computer Graphics International Conference*. Victoria, British Columbia, Canada: ACM; 2009, p. 5–12.
- [5] McMillen T, Goriely A. Tendril perversion in intrinsically curved rods. *J. Nonlinear Sci* 2002;12(3):241–81.
- [6] Wang Z, Fratarcangeli M, Ruimi A, Srinivasa AR. Real time simulation of inextensible surgical thread using a Kirchhoff rod model with force output for haptic feedback applications. *Int J Solids Struct* 2017;113–114:192–208.
- [7] Benham C. An elastic model of the large-scale structure of duplex DNA. *Biopolymers* 1979;18(3):609–23.
- [8] Schlick T. Modeling superhelical DNA: recent analytical and dynamic approaches. *Curr Opin Struct Biol* 1995;5(2):245–62.
- [9] Rodrigo Vélez-Cordero J, Lauga E. Waving transport and propulsion in a generalized Newtonian fluid. *J Non-Newton Fluid Mech* 2012;199:37–50.
- [10] Fauci L, Gueron S, editors. Computational modeling in biological fluid dynamics. *The IMA Volumes in Mathematics and its Applications*, vol. 124, Springer; 2001.



- [11] Maniyeri R, Kan S. Numerical study on the rotation of an elastic rod in a viscous fluid using an immersed boundary method. *J Mech Sci Technol* 2012;26:1515–22.
- [12] Ishimoto K, Lauga E. The N-flagella problem: elasto-hydrodynamic motility transition of multi-flagellated bacteria. *Proc R Soc Lond Ser A Math Phys Eng Sci* 2019;475:20180690.
- [13] Paz S, Buscaglia G. Simulating squirmers with volumetric solvers. *J Braz Comput Soc Mech Sci Eng* 2020;42:547.
- [14] Romero I, Gebhardt C. Variational principles for nonlinear Kirchhoff rods. *Acta Mech* 2020;231:625–47.
- [15] Gebhardt C, Romero I. On a nonlinear rod exhibiting only axial and bending deformations: mathematical modeling and numerical implementation. *Acta Mech* 2021;232:3825–47.
- [16] Childress S, Hosoi A, Schultz W, Wang Z, editors. Natural locomotion in fluids and on surfaces. The IMA Volumes in Mathematics and its Applications, vol. 155, Springer; 2012.
- [17] Park Y, Kim Y, Lim S. Locomotion of a single-flagellated bacterium. *J Fluid Mech* 2019;859:586–612.
- [18] Danis U, Rasooli R, Chen C-Y, Dur O, Sitti M, Pekkan K. Thrust and hydrodynamic efficiency of the bundled flagella. *Micromachines* 2019;10:449.
- [19] Liu Z, Qin F, Zhu L, Yang R, Luo X. Effects of the intrinsic curvature of elastic filaments on the propulsion of a flagellated microrobot. *Phys Fluids* 2020;32:041902.
- [20] Ranner T. A stable finite element method for low inertia undulatory locomotion in three dimensions. *Appl Numer Math* 2020;156:422–45.
- [21] Shen X, Arratia P. Undulatory swimming in viscoelastic fluids. *Phys Rev Lett* 2011;106:208101.
- [22] Montenegro-Johnson T, Smith D, Lohin D. Physics of rheologically enhanced propulsion: Different strokes in generalized Stokes. *Phys Fluids* 2013;25:081903.
- [23] Rorai C, Zaitsev M, Karabasov S. On the limitations of some popular numerical models of flagellated microswimmers: importance of long-range forces and flagellum waveform. *R Soc Open Sci* 2018;6:180745.
- [24] Farhat C, Lesoinne M, Le Tallec P. Load and motion transfer algorithms for fluid/structure interaction problems with non-matching discrete interfaces: Momentum and energy conservation, optimal discretization and application to aeroelasticity. *Comput Methods Appl Mech Engrg* 1998;157:95–114.
- [25] Sarrate J, Huerta A, Donea J. Arbitrary Lagrangian–Eulerian formulation for fluid–rigid body interaction. *Comput Methods Appl Mech Engrg* 2001;190:3171–88.
- [26] Quaini A, Quarteroni A. A semi-implicit approach for fluid-structure interaction based on an algebraic fractional step method. *Math Models Methods Appl Sci* 2007;17:957–83.
- [27] Boffi D, Gastaldi L. A fictitious domain approach with Lagrange multiplier for fluid-structure interactions. *Numer Math* 2017;135:711–32.
- [28] Annese M, Fernández M, Gastaldi L. Splitting schemes for a Lagrange multiplier formulation of FSI with immersed thin-walled structure: stability and convergence analysis. 2020, arXiv:2007.04699.
- [29] Alouges F. Low reynolds number swimming and controlability. *ESAIM: Proceedings* 2013;41:1–14.
- [30] Alouges F, DeSimone A, Heltai L, Lefebvre-Lepot A, Merlet B. Optimally swimming stokesian robots. *Discr Contin Dyn Syst - B* 2013;18:1189–215.
- [31] Faris Y, Pomet J-B, Régner S, Giraldo L. Comparison of optimal actuation patterns for flagellar magnetic micro-swimmers. *IFAC-PapersOnLine* 2020;53:9125–30.
- [32] Peskin C. The immersed boundary method. *Acta Numer* 2002;11:479–517.
- [33] Phan-Thien N, Trag-Cong T, Ramia M. A boundary-element analysis of flagellar propulsion. *J Fluid Mech* 1987;184:533–49.
- [34] Trachtenberg S, Fishelov D, Ben-Artzi M. Bacterial flagellar microhydrodynamics: Laminar flow over complex flagellar filaments, analog archimedean screws and cylinders, and its perturbations. *Biophys J* 2003;85:1345–57.
- [35] Goto T, Nakata K, Baba K, Nishimura M, Magariyama Y. A fluid-dynamic interpretation of the asymmetric motion of singly flagellated bacteria swimming close to a boundary. *Biophys J* 2005;89:3771–9.
- [36] Smith D. A boundary element regularized Stokeslet method applied to cilia- and flagella-driven flow. *Proc R Soc Lond Ser A Math Phys Eng Sci* 2009;465:3605–26.
- [37] Shum H, Gaffney E. Hydrodynamic analysis of flagellated bacteria swimming near one and between two no-slip plane boundaries. *Phys Rev E* 2015;91:033012.
- [38] Duprat C, Stone H, editors. Fluid-Structure Interactions in Low-Reynolds-Number Flows. *Soft Matter Series*, The Royal Society of Chemistry; 2016, p. P001–477.
- [39] Shum H. Microswimmer propulsion by two steadily rotating helical flagella. *Micromachines* 2019;10:65.
- [40] Carichino L, Olson S. Emergent three-dimensional sperm motility: coupling calcium dynamics and preferred curvature in a Kirchhoff rod model. *Math Med Biol* 2019;36:439–69.
- [41] Giuliani N, Rossi M, Noselli G, DeSimone A. How euglena gracilis swims: Flow field reconstruction and analysis. *Phys Rev E* 2021;103:023102.
- [42] Gagnon D, Arratia P. The cost of swimming in generalized Newtonian fluids: experiments with *c. elegans*. *J Fluid Mech* 2016;800:753–65.
- [43] Gagnon D, Montenegro-Johnson T. Thrifty swimming with shear-thinning; a note on out-of-plane effects for undulatory locomotion through shear-thinning fluids. *ANZIAM J* 2018;59:443–54.
- [44] Riley E, Lauga E. Empirical resistive-force theory for slender biological filaments in shear-thinning fluids. *Phys Rev E* 2017;95:062416.
- [45] Qu Z, Breuer K. Effects of shear-thinning viscosity and viscoelastic stresses on flagellated bacteria motility. *Phys Rev Fluids* 2020;5:073103.
- [46] Qin K, Peng Z, Chen Y, Nganguia H, Zhu L, Pak O. Propulsion of an elastic filament in a shear-thinning fluid. *Soft Matter* 2021;17:3829–39.
- [47] Nganguia H, Pietrzyk K, Pak O. Swimming efficiency in a shear-thinning fluid. *Phys Rev E* 2017;96:062606.
- [48] Li G, Lauga E, Ardekani A. Microswimming in viscoelastic fluids. *J Non-Newton Fluid Mech* 2021;297:104655.
- [49] Antman SS. *Nonlinear Problems of Elasticity*. Springer; 1995.
- [50] Rubin MB. *Cosserat Theories: Shells, Rods and Points*. Springer; 2000.
- [51] Taylor C, Hood P. A numerical solution of the Navier-Stokes equation using the finite element technique. *Comput Fluids* 1973;1:73–100.
- [52] Franca L, Frey S. Stabilized finite element methods: II. The incompressible Navier-Stokes equations. *Comput Methods Appl Mech Engrg* 1992;99:209–33.
- [53] Rathgeber F, Ham D, Mitchell L, Lange M, Luporini F, Mcrae A, Bercea G-T, Markall G, Kelly P. Firedrake: Automating the finite element method by composing abstractions. *ACM Trans Math Software* 2016;43:1–27.
- [54] Alnæs M, Logg A, Ølgaard K, Rognes M, Wells G. Unified form language: A domain-specific language for weak formulations of partial differential equations. *ACM Trans Math Software* 2014;40:1–37.
- [55] Geuzaine C, Remacle J-F. Gmsh: A 3-D finite element mesh generator with built-in pre- and post-processing facilities. *Internat J Numer Methods Engrg* 2009;79:1309–31.
- [56] Schlömer N. Pygmsh 6.1.0. 2020, <https://pypi.org/project/pygmsh/6.1.0/>.
- [57] Balay S, Abhyankar S, Adams M, Brown J, Brune P, Buschelman K, Dalcin L, Dener A, Eijkhout V, Gropp W, Karpeyev D, Kaushik D, Knepley M, May D, Curfman McInnes L, Tran Mills R, Munson T, Rupp K, Sanan P, Smith B, Zampini S, Zhang H. PETSc web page. 2021, <https://www.mcs.anl.gov/petsc>.

- [58] Amestoy P, Duff I, Koster J, L'Excellent J-Y. A fully asynchronous multifrontal solver using distributed dynamic scheduling. *SIAM J Matrix Anal Appl* 2001;23:15–41.
- [59] Amestoy P, Buttari A, L'Excellent J-Y, Mary T. Performance and scalability of the block low-rank multifrontal factorization on multicore architectures. *ACM Trans Math Software* 2019;45:2:1–26.
- [60] Virtanen P, Gommers R, Oliphant T, Haberland M, Reddy T, Cournapeau D, Burovski E, Peterson P, Weckesser W, Bright J, van der Walt S, Brett M, Wilson J, Millman K, Mayorov N, Nelson A, Jones E, Kern R, Larson E, Carey C, Polat I, Feng Y, Moore E, Van der Plas J, Laxalde D, Perktold J, Cimrman R, Henriksen I, Quintero E, Harris C, Archibald A, Ribeiro A, Pedregosa F, van Mulbregt P, SciPy 10 Contributors. *Scipy 1.0: Fundamental algorithms for scientific computing in python. Nature Methods* 2020;17:261–72. <http://dx.doi.org/10.1038/s41592-019-0686-2>.
- [61] Kelley C. *Iterative Methods for Linear and Nonlinear Equations*. Siam; 1995.
- [62] Ahrens J, Geveci B, Law C. *ParaView: An End-User Tool for Large Data Visualization, Visualization Handbook*. Elsevier; 2005.
- [63] Montenegro-Johnson T, Smith A, Smith D, Loghin D, Blake J. Modelling the fluid mechanics of cilia and flagella in reproduction and development. *The Eur J Phys E* 2012;35:111.
- [64] Kennedy P, Zheng R. *Flow Analysis of Injection Molds*. Carl Hanser Verlag GmbH Co KG; 2013.