Journal of Computational Physics 466 (2022) 111375

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

A finite volume algorithm for the dynamics of filaments, rods, and beams

Paul M. Ryan [a], Charles W. Wolgemuth [a,b,c,*]

[a] Department of Physics, University of Arizona, Tucson, AZ 85721, United States of America
[b] Department of Molecular and Cellular Biology, University of Arizona, Tucson, AZ 85721, United States of America
[c] Johns Hopkins Physical Sciences-Onocology Center, Johns Hopkins University, Baltimore, MD 21218, United States of America

ARTICLE INFO

Article history:
Received 5 August 2021
Received in revised form 8 April 2022
Accepted 6 June 2022
Available online 10 June 2022

Keywords:
Filament dynamics
Finite volume
Fluid-structure interactions
Cellular motility

ABSTRACT

Filaments, rods, and beams are ubiquitous in biology and in many man-made products and structures. While a substantial amount of research has been done to understand the statics and dynamics of these long, thin objects, there remain many unanswered and unstudied problems related to the dynamics of bending and twisting filamentary objects. Simulating the general dynamics of these structures in 3D remains challenging. For example, the net force and torque on a free filament immersed in fluid at low Reynolds number must be zero. However, standard finite difference approaches will often fail to preserve the zero force and torque conditions. These numerical artifacts cause spurious rotations and translations that prohibit, or at least limit, their accuracy in simulating the dynamics of filaments, rods, and beams in these contexts (such as the free-swimming motion of a filamentary microorganism). Here we develop a finite volume discretization based on the Kirchoff equations that naturally guarantees the correct total integral of the forces and torques on filaments, rods, or beams. We then couple this discretization to resistive force theory to develop a stable, accurate dynamic algorithm of filament motion at low Reynolds number. We use a range of sample problems to highlight the utility, stability, and accuracy of this method. While our sample problems focus on low Reynolds number dynamics in the context of resistive force theory (RFT), our discretized finite volume algorithm is general and can be applied to inertial dynamics, immersed boundary methods, and boundary integral methods, as well.

1. Introduction

Semi-flexible objects that are significantly longer in one dimension than in the other two dimensions are common in nature and engineering; from microfilaments defining cell structure to the beams that support buildings. Long chain molecules, bacteria, plant tendrils, clothing fibers, hair, eels, snakes, medical tubing, and bridge cables comprise just a small subset of other relevant examples. It is hardly surprising that the physical behavior of these objects has received significant attention for well over a century and a half, beginning with the seminal work of Kirchoff [1]. In more recent years, a substantial amount of work has gone into understanding the role that filamentary objects play in biology (e.g., [2–5]). For example, it has been found that the molecular motors that replicate our DNA generate and detect twist in the genomic strands [6]. Many bacteria and microorganisms propel themselves using dynamic filamentary structures, or are themselves

* Corresponding author.
E-mail address: wolg@email.arizona.edu (C.W. Wolgemuth).

https://doi.org/10.1016/j.jcp.2022.111375
0021-9991/

dynamic filamentary or rod-shaped objects [7]. Therefore, it is not surprising that a substantial amount of work in recent years has investigated various methods for handling the dynamics of filamentary objects at low Reynolds number [8–16].

In the context of microorganism motility, consider a swimming sperm. The sperm cell is composed of two parts, a head and a tail. In this very simple description, we can assume the head is an inanimate, roughly spheroidal blob. The tail on the other hand is a long, thin rod with internal motors that cause it to flex and bend [17]. The sperm cell is small, on order of tens of microns, and swims at speeds of around 100 μm/s [18]. The Reynolds number, a measure of the relative importance of inertia to resistive forces, is less than 0.1, which implies that inertia is relatively unimportant and the sum of the forces that act on the sperm cell is approximately equal to zero. The sperm cell's motility is then exactly determined by the balance of the net thrust from the tail's undulation and fluidic drag from the moving cell body. To accurately simulate sperm motions, which has recently shown how the tail's asymmetric beating pattern leads to symmetric swimming trajectories [17], requires that the numerical algorithm preserve the net force and torque applied to the cell. Otherwise, if these quantities are not preserved, the simulation will produce artificial swimming speeds and rotations.

To highlight another aspect that our algorithm handles, it is interesting to note that many microorganism swimmers cannot be described as simple elastic rods or filaments. For instance, the force producing motor in the tail of the sperm cell, called an axoneme, consists of 11 microtubule filaments in a 9+2 arrangement that is linked together by walking motor proteins [19]. Spirochete bacteria, such as the bacterium that causes Lyme disease, are long, thin cells with tens of rotating helical filaments that wrap around the body producing traveling waves along the cell body subsequently driving motility [20]. Modeling these types of coupled filaments numerically requires a general approach that can go beyond linear elastic constitutive relationships [21]. In addition, for biopolymers and filaments that are even smaller in size than the sperm cell, random thermal forces can drastically influence the motion of the filament. Accurate simulation of a filament in the presence of these thermal forces is then important in a number of biological problems, and has become an area of active research [22–24].

To address these issues, we start from the Kirchoff rod equations, a general physical description of the restoring forces and torques generated inside long, thin objects when they are deformed. Discretizing the equations using a finite volume algorithm allows us to handle how forces and torques act between neighboring segments of the simulated rod, filament, or beam in order to exactly prescribe the net force and torque. We show here that this algorithm is accurate and stable, even in the presence of random forces. Numerous sample problems are used to exemplify the utility of this algorithm for a range of problems.

2. The dynamics of filaments at low Reynolds number

Because rods, beams, and filaments are much longer in length than they are along their cross-sectional dimensions, the shape of these thin objects can be primarily defined by the centerline position, which can be described by the vector $\mathbf{r}(s,t)$, where s is the arclength and t is time. We can then define a material frame by affixing an orthonormal triad $(\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3)$ to each point along the centerline, where $\mathbf{e}_3 = \partial\mathbf{r}/\partial s$ is the tangent vector, $\mathbf{e}_1$ points to a line along the surface of the filament, and $\mathbf{e}_2 = \mathbf{e}_3 \times \mathbf{e}_1$ (Fig. 1). As one moves along the centerline, this orthonormal frame rotates with the rotational rate given by the strain vector $\boldsymbol{\Omega} = \Omega_1\mathbf{e}_1 + \Omega_2\mathbf{e}_2 + \Omega_3\mathbf{e}_3$, such that

$$\frac{\partial \mathbf{e}_i}{\partial s} = \boldsymbol{\Omega} \times \mathbf{e}_i \,. \tag{1}$$

The rotation rate of the orthonormal triad at fixed arclength is given by

$$\frac{\partial \mathbf{e}_i}{\partial t} = \boldsymbol{\omega} \times \mathbf{e}_i \,, \tag{2}$$

where $\boldsymbol{\omega} = \omega_1\mathbf{e}_1 + \omega_2\mathbf{e}_2 + \omega_3\mathbf{e}_3$ is the spin vector [25].

When a rod is bent and twisted, strain in the material results in restoring forces and torques, which we denote by $\mathbf{F}$ and $\mathbf{M}$, respectively. For any infinitesimal length of the rod, the restoring moment is related to the force as [26] (Fig. 1)

$$\frac{\partial \mathbf{M}}{\partial s} = -\mathbf{e}_3 \times \mathbf{F} + \rho I \left(\mathbf{e}_1 \times \frac{\partial^2 \mathbf{e}_1}{\partial t^2} + \mathbf{e}_2 \times \frac{\partial^2 \mathbf{e}_2}{\partial t^2} \right) + \mathbf{m}_{\text{ext}} \,, \tag{3}$$

where ρ is the mass density of the rod, I is the moment of inertia about the tangent vector, and $\mathbf{m}_{\text{ext}}$ is the net external moment per length applied to the infinitesimal length. Newton's 2nd law applied to this segment gives [26]

$$\rho A \frac{\partial^2 \mathbf{r}}{\partial t^2} = \frac{\partial \mathbf{F}}{\partial s} + \mathbf{f}_{\text{ext}} \,, \tag{4}$$

with A the cross-sectional area and $\mathbf{f}_{\text{ext}}$ the net external applied force per unit length (i.e., the body forces). Eqs. (3)-(4) are the general form of the dynamic Kirchoff equations. For a rod or filament that is immersed in a fluid at low Reynolds number, the inertial terms can be ignored and the body forces and torques come from the fluid's resistive forces and torques [27]. In this case, the dynamic rod equations become

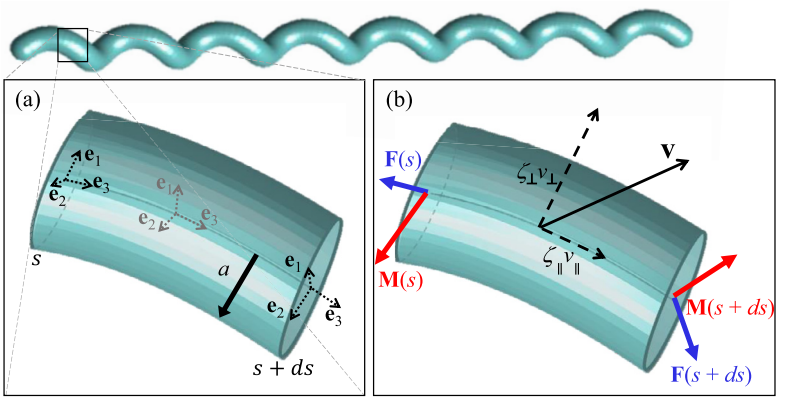

Fig. 1. A long, thin helical rod is decomposed into short segments; for example, one that spans from arclength position s to $s+ds$. (a) At each point along the centerline of the rod, an orthonormal triad $(\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3)$ is defined, where $\mathbf{e}_3$ is the tangent direction. The orthonormal triad rotates as one moves from s to $s+ds$. (b) Forces $\mathbf{F}$ and moments $\mathbf{M}$ act on the cross-sections. Torque balance on the segment then leads to a relationship between the derivative of the moment and the force, given by Eq. (3). When the filament moves in a low Reynolds fluid at velocity $\mathbf{v}$, the body resistive force is proportional to that velocity. This diagram assumes the ratio of perpendicular to parallel drag coefficients $\zeta_\perp/\zeta_\parallel = 2$.

$$\frac{\partial \mathbf{M}}{\partial s} = -\mathbf{e}_3 \times \mathbf{F} + \zeta_r \omega_3 \mathbf{e}_3 + \mathbf{m}\,, \tag{5}$$

$$\zeta_\perp \frac{\partial \mathbf{r}}{\partial t} + (\zeta_\parallel - \zeta_\perp)\left(\frac{\partial \mathbf{r}}{\partial t}\cdot \mathbf{e}_3\right)\mathbf{e}_3 = \frac{\partial \mathbf{F}}{\partial s} + \mathbf{f}\,, \tag{6}$$

where the drag from the fluid of viscosity η is defined from resistive force theory (RFT) [28], with $\zeta_r \sim 4\pi\eta a^2$ the rotational drag coefficient and $\zeta_\perp \sim 4\pi\eta$ and $\zeta_\parallel \sim 2\pi\eta$ the translational drag coefficients for motion of the segment perpendicular or parallel to $\mathbf{e}_3$, respectively. The external torque and force, $\mathbf{m}$ and $\mathbf{f}$, now do not include the resistive drag from the fluid.

Note that Eq. (6) is in the form of a conservation equation, similar to the continuity equation, which relates the changes in density, ρ, to the divergence of a flux $\mathbf{J}$ and a source term S as

$$\frac{\partial \rho}{\partial t} = -\nabla \cdot \mathbf{J} + S\,. \tag{7}$$

It is well known that the solution of the continuity equation with finite difference schemes does not, in general, conserve the total mass of the system in cases where $S = 0$ [29]. Finite volume schemes, on the hand, explicitly enforce conservation of mass by directly balancing fluxes between adjacent control volumes [29]. Therefore, the form of the dynamic filament equations suggests that finite volume schemes may be appropriate for properly balancing the forces and moments in simulations. Although the finite volume algorithm that we develop here focuses on the discretization of moment and torque components of Eqs. (5)-(6), it is straightforward to extend this method to handle inertial dynamics as well. In what follows, we use these two equations to define the force, torque, displacement, and rotation of any rod, filament or beam in a low Reynolds environment.

Eq. (5) can be used to define the force in terms of the moment by taking the cross product of this equation with the tangent vector:

$$\mathbf{F} = \frac{\partial}{\partial s}(\mathbf{e}_3 \times \mathbf{M}) - \frac{\partial \mathbf{e}_3}{\partial s} \times \mathbf{M} + \mathbf{e}_3 \times \mathbf{m} + F_3 \mathbf{e}_3\,, \tag{8}$$

where F_3 is the tangential component of the force, which is defined by a constitutive relationship governing the extensibility of the rod. In many situations, thin rods and filaments can be considered to be inextensible. This is because the forces required to compress or extend a rod by a certain amount scale like the cross-sectional area, while bending forces scale like the square of the cross-sectional area [30].

To close the dynamic equations, a constitutive relationship is needed to relate the restoring moment to the deformation of the rod. Assuming linear elasticity leads to [26]

$$\mathbf{M} = A_1(\Omega_1 - \kappa_1)\mathbf{e}_1 + A_2(\Omega_2 - \kappa_2)\mathbf{e}_2 + C(\Omega_3 - \tau_0)\mathbf{e}_3\,, \tag{9}$$

where A_1 and A_2 are bending moduli and C is the twist modulus. The equilibrium shape of the filament is defined by the preferred curvatures κ_1 and κ_2 and the preferred torsion τ_0. It is useful to note that

$$\Omega_1 \mathbf{e}_1 + \Omega_2 \mathbf{e}_2 = \mathbf{e}_3 \times \frac{\partial \mathbf{e}_3}{\partial s} = \mathbf{e}_3 \times \frac{\partial^2 \mathbf{r}}{\partial s^2}\,, \tag{10}$$

which allows the cross product of the tangent vector and the moment to be written as

$$\mathbf{e}_3 \times \mathbf{M} = -A_+ \frac{\partial^2 \mathbf{r}}{\partial s^2} + A_- \left(\mathbf{e}_1\mathbf{e}_1 - \mathbf{e}_2\mathbf{e}_2\right) \cdot \frac{\partial^2 \mathbf{r}}{\partial s^2} - A_1\kappa_1\mathbf{e}_2 + A_2\kappa_2\mathbf{e}_1 \,, \tag{11}$$

where $A_+ = (A_1 + A_2)/2$ and $A_- = (A_1 - A_2)/2$. In addition,

$$\begin{aligned}\frac{\partial \mathbf{e}_3}{\partial s} \times \mathbf{M} &= -M_3\left(\Omega_1\mathbf{e}_1 + \Omega_2\mathbf{e}_2\right) + \left(M_1\Omega_1 + M_2\Omega_2\right)\mathbf{e}_3 \\ &= -C\left(\Omega_3 - \tau_0\right)\left(\mathbf{e}_3 \times \frac{\partial \mathbf{e}_3}{\partial s}\right) + \left(A_1\left(\Omega_1 - \kappa_1\right)\Omega_1 + A_2\left(\Omega_2 - \kappa_2\right)\Omega_2\right)\mathbf{e}_3 \,,\end{aligned} \tag{12}$$

which allows us to write the force as

$$\mathbf{F} = \frac{\partial}{\partial s}\left(\mathbf{e}_3 \times \mathbf{M}\right) + C\left(\Omega_3 - \tau_0\right)\left(\mathbf{e}_3 \times \frac{\partial \mathbf{e}_3}{\partial s}\right) + F_3'\mathbf{e}_3 + \mathbf{e}_3 \times \mathbf{m} \,, \tag{13}$$

where

$$F_3' = F_3 - A_1\left(\Omega_1 - \kappa_1\right)\Omega_1 - A_2\left(\Omega_2 - \kappa_2\right)\Omega_2 \,. \tag{14}$$

3. A finite volume discretization of the forces and torques along a filament

We begin the development of our algorithm by discretizing the arclength coordinate along the centerline of the filament. Because we are interested in simulating incompressible filaments, we assume that the nodes are evenly spaced with grid spacing Δs. We will use a soft constraint to enforce the incompressibility condition, which will be described later. We define the vector position of the filament, the orthonormal triad, the strain vector, and the restoring moment at the node locations, while the force is defined half way between the individual nodes (Fig. 2). We use the notation that the vector position of the ith node at time t is $\mathbf{r}_i^t$. For all of the terms in the dynamic equations, except the tangential force, we define $\mathbf{e}_3$ at the nodal positions. This can be done with a standard second-order finite difference derivative,

$$\left.\frac{\partial \mathbf{r}}{\partial s}\right|_i = \begin{cases} \dfrac{\left(-3\mathbf{r}_1 + 4\mathbf{r}_2 - \mathbf{r}_3\right)}{2\Delta s} & (i = 1) \,, \\ \dfrac{\left(\mathbf{r}_{i+1} - \mathbf{r}_{i-1}\right)}{2\Delta s} & (2 \le i \le N-1) \,, \\ \dfrac{\left(\mathbf{r}_{N-2} - 4\mathbf{r}_{N-1} + 3\mathbf{r}_N\right)}{2\Delta s} & (i = N) \,, \end{cases}$$

where N is the number of nodes. The tangent vector is then

$$\mathbf{e}_3 = \frac{1}{\left|\frac{\partial \mathbf{r}}{\partial s}\right|}\frac{\partial \mathbf{r}}{\partial s} \,. \tag{15}$$

The spatial second derivative of $\mathbf{r}$ can also be computed using a second-order finite difference derivative. However, we often compute the first and second derivatives of $\mathbf{r}$ using the derivatives of the sixth-order Lagrange interpolating polynomial defined on a seven-node stencil about the point of interest. This choice doesn't affect the convergence of the method but does improve accuracy (see Sec. 7.2). The 1 and 2 components of the strain vector are computed as

$$\begin{aligned}\Omega_{1,i} &= -\,\mathbf{e}_2 \cdot \left.\frac{\partial^2 \mathbf{r}}{\partial s^2}\right|_i \,, \\ \Omega_{2,i} &= \mathbf{e}_1 \cdot \left.\frac{\partial^2 \mathbf{r}}{\partial s^2}\right|_i \,.\end{aligned}$$

To construct the finite volume discretization of the equations of motion, we integrate Eq. (6) over a control volume centered about each node. For the ith node, we integrate from $s_{i-1/2}$ to $s_{i+1/2}$, leading to

$$\frac{\zeta_\perp L_c}{\Delta t}\left[\hat{\mathcal{I}} + \left(\frac{\zeta_\parallel}{\zeta_\perp} - 1\right)\mathbf{e}_{3,i}^t\mathbf{e}_{3,i}^t\right] \cdot \left(\mathbf{r}_i^{t+\Delta t} - \mathbf{r}_i^t\right) = \mathbf{F}_{i+1/2} - \mathbf{F}_{i-1/2} + \Delta s\mathbf{f}_i \,, \tag{16}$$

where $\hat{\mathcal{I}}$ is the identity matrix, and L_c is the control volume about each node, which is equal to Δs for all interior nodes and $\Delta s/2$ for the end nodes. We also define the drag matrix

$$\hat{\mathcal{D}}_i = \frac{\zeta_\perp L_c}{\Delta t}\left[\hat{\mathcal{I}} + \left(\frac{\zeta_\parallel}{\zeta_\perp} - 1\right)\mathbf{e}_{3,i}^t\mathbf{e}_{3,i}^t\right] \,. \tag{17}$$

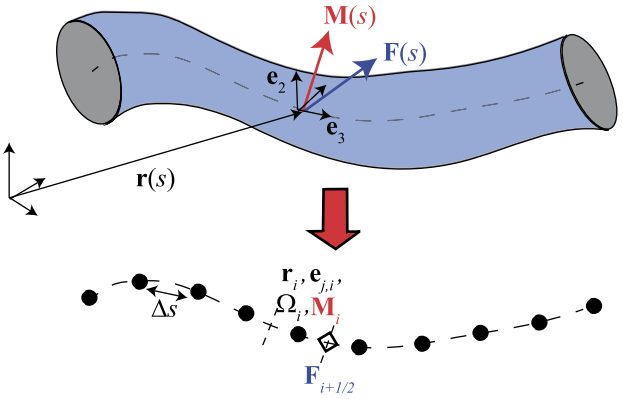

Fig. 2. Discretization of filament geometry, forces, and torques. We discretize the filament using a number of evenly spaced nodes with grid spacing Δs. The vector position of the centerline, the orthonormal triad vectors, $\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3$, the strain vector $\boldsymbol{\Omega}$, and the moment are defined at the grid points (circles), while the force is defined at the half grid points (diamonds). For our finite volume algorithm, we integrate over control volumes centered on the grid nodes, denoted by the dashed lines.

Here we have not defined at what time the force $\mathbf{F}$ is evaluated, as this term will be handled in a semi-implicit form that is described in Sec. 4. Likewise, how the force per length $\mathbf{f}$ should be handled depends on the functional form of that term. In many applications, we will use that $\mathbf{f} = 0$; however, we will also show how to handle this term when it represents random thermal forces (see Sec. 7.4). For now, in order to simplify the exposition, we will treat $\mathbf{f}$ as zero.

The crux of our method lies in how we handle the forces at the half grid point in Eq. (16), especially near the boundary nodes. We use Eq. (8) to discretize the forces at the half-nodes:

$$\mathbf{F}_{i+1/2} = \frac{1}{\Delta s}\left(\mathbf{e}_{3,i+1} \times \mathbf{M}_{i+1} - \mathbf{e}_{3,i} \times \mathbf{M}_i\right) - M_3 \left(\frac{\partial \mathbf{r}}{\partial s} \times \frac{\partial^2 \mathbf{r}}{\partial s^2}\right)_{i+1/2} + F'_{3,i+1/2} \frac{(\mathbf{r}_{i+1} - \mathbf{r}_i)}{\Delta s}, \tag{18}$$

where we have used $\mathbf{m} = 0$. Note that the last term on the right-hand side is discretized using the second-order approximation to the derivative of $\mathbf{r}$ at the half grid point to define the tangent vector. The first term on the right-hand side of this equation only depends on terms defined at the neighboring nodes. The second term is discretized as

$$M_3 \left(\frac{\partial \mathbf{r}}{\partial s} \times \frac{\partial^2 \mathbf{r}}{\partial s^2}\right)_{i+1/2} = \frac{1}{2}\left(M_{3,i+1}\left(\frac{\partial \mathbf{r}}{\partial s} \times \frac{\partial^2 \mathbf{r}}{\partial s^2}\right)_{i+1} + M_{3,i}\left(\frac{\partial \mathbf{r}}{\partial s} \times \frac{\partial^2 \mathbf{r}}{\partial s^2}\right)_{i}\right). \tag{19}$$

Finally, to handle the modified tangential force F'_3, we break it into two pieces. The first piece, $-A_1(\Omega_1 - \kappa_1)\Omega_1 - A_2(\Omega_2 - \kappa_2)\Omega_2$, is handled in the same way as in Eq. (19), where we use the average of the values at nodes i and $i+1$ to approximate the value at $i + 1/2$. The second term is F_3, which is used to enforce the incompressibility of the filament via a soft constraint:

$$F_{3,i+1/2} = \sigma_0 \Delta s \left(\frac{\Delta s}{|\mathbf{r}_{i+1} - \mathbf{r}_i|} - 1\right), \tag{20}$$

where σ_0 is a force per unit length which is related to the extensibility of the filament. In our algorithm, we set the σ_0 parameter in order to maintain the internodal spacing within a small range of Δs. The factor of Δs that multiplies σ_0 gives the proper scaling of the compressional modulus when the grid spacing is changed. For the simulations we describe here, $2,000 \le \sigma_0 \le 20,000$ was sufficient to maintain the internodal spacing within 1% of Δs.

Combining all of these terms, we can rewrite the discretized dynamic equation for the filament as

$$\begin{aligned}\hat{\mathcal{D}}_i \cdot \left(\mathbf{r}_i^{t+\Delta t} - \mathbf{r}_i^t\right) = \frac{1}{\Delta s}&\left(\mathbf{e}_{3,i+1} \times \mathbf{M}_{i+1} - 2\mathbf{e}_{3,i} \times \mathbf{M}_i + \mathbf{e}_{3,i-1} \times \mathbf{M}_{i-1}\right) \\ &- \frac{1}{2}\left(M_{3,i+1}\left(\frac{\partial \mathbf{r}}{\partial s} \times \frac{\partial^2 \mathbf{r}}{\partial s^2}\right)_{i+1} - M_{3,i-1}\left(\frac{\partial \mathbf{r}}{\partial s} \times \frac{\partial^2 \mathbf{r}}{\partial s^2}\right)_{i-1}\right) \\ &+ F'_{3,i+1/2}\frac{(\mathbf{r}_{i+1} - \mathbf{r}_i)}{\Delta s} - F'_{3,i-1/2}\frac{(\mathbf{r}_i - \mathbf{r}_{i-1})}{\Delta s}.\end{aligned} \tag{21}$$

For a free filament, the moment is zero at the ends. Therefore, at nodes 1 and 2 we have

$$\hat{\mathcal{D}}_1 \cdot \left(\mathbf{r}_1^{t+\Delta t} - \mathbf{r}_1^t\right) = \frac{1}{\Delta s}\left(\mathbf{e}_{3,2} \times \mathbf{M}_2\right) - \frac{1}{2}\left(M_{3,2}\left(\frac{\partial \mathbf{r}}{\partial s} \times \frac{\partial^2 \mathbf{r}}{\partial s^2}\right)_2\right) + F'_{3,3/2}\frac{(\mathbf{r}_2 - \mathbf{r}_1)}{\Delta s}, \tag{22}$$

$$\hat{\mathcal{D}}_2 \cdot \left(\mathbf{r}_2^{t+\Delta t} - \mathbf{r}_2^t\right) = \frac{1}{\Delta s}\left(\mathbf{e}_{3,3} \times \mathbf{M}_3 - 2\mathbf{e}_{3,2} \times \mathbf{M}_2\right) - \frac{1}{2}\left(M_{3,3}\left(\frac{\partial \mathbf{r}}{\partial s} \times \frac{\partial^2 \mathbf{r}}{\partial s^2}\right)_3\right) + F'_{3,5/2}\frac{(\mathbf{r}_3 - \mathbf{r}_2)}{\Delta s} - F'_{3,3/2}\frac{(\mathbf{r}_2 - \mathbf{r}_1)}{\Delta s}\,, \tag{23}$$

with similar expressions at the $N-1$ and Nth nodes.

For a filament with a prescribed moment at the ends, the boundary condition can be directly implemented into these equations. It is convenient to write these equations in the compact form

$$\hat{\mathcal{D}} \cdot \left(\mathbf{r}^{t+\Delta t} - \mathbf{r}^t\right) = \hat{\mathcal{O}}_2 \cdot (\mathbf{e}_3 \times \mathbf{M}) - \hat{\mathcal{O}}_1 \cdot \left(M_3\left(\frac{\partial \mathbf{r}}{\partial s} \times \frac{\partial^2 \mathbf{r}}{\partial s^2}\right)\right) + \hat{\mathcal{F}}_3 \cdot \mathbf{r}\,, \tag{24}$$

by defining a modified second derivative operator,

$$\hat{\mathcal{O}}_2 = \frac{1}{\Delta s}\begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & \cdots \\ 0 & -2 & 1 & 0 & 0 & 0 & 0 & \cdots \\ 0 & 1 & -2 & 1 & 0 & 0 & 0 & \cdots \\ 0 & 0 & 1 & -2 & 1 & 0 & 0 & \cdots \\ & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \\ \cdots & 0 & 0 & 0 & 1 & -2 & 1 & 0 \\ \cdots & 0 & 0 & 0 & 0 & 1 & -2 & 0 \\ \cdots & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}, \tag{25}$$

a modified first derivative operator,

$$\hat{\mathcal{O}}_1 = \frac{1}{2}\begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & \cdots \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & \cdots \\ 0 & -1 & 0 & 1 & 0 & 0 & 0 & \cdots \\ 0 & 0 & -1 & 0 & 1 & 0 & 0 & \cdots \\ & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \\ \cdots & 0 & 0 & 0 & -1 & 0 & 1 & 0 \\ \cdots & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ \cdots & 0 & 0 & 0 & 0 & 0 & -1 & 0 \end{bmatrix}, \tag{26}$$

and a tangential forcing operator,

$$\hat{\mathcal{F}}_3 = \frac{1}{\Delta s}\begin{bmatrix} -F'_{3,3/2} & F'_{3,3/2} & 0 & 0 & \cdots \\ F'_{3,3/2} & -(F'_{3,3/2} + F'_{3,5/2}) & F'_{3,5/2} & 0 & \cdots \\ & \ddots & \ddots & \ddots & \\ \cdots & 0 & F'_{3,N-\frac{3}{2}} & -(F'_{3,N-\frac{3}{2}} + F'_{3,N-\frac{1}{2}}) & F'_{3,N-\frac{1}{2}} \\ \cdots & 0 & 0 & F'_{3,N-\frac{1}{2}} & -F'_{3,N-\frac{1}{2}} \end{bmatrix}. \tag{27}$$

These matrices are not only convenient for writing out the equations in a compact form, they are also directly used in the algorithm. Indeed, the $\hat{\mathcal{O}}_1$ and $\hat{\mathcal{O}}_2$ matrices are the most relevant and important aspect of our method: directly accounting for the physics of how the restoring forces and moments are distributed along the filament while automatically imposing the free-end boundary conditions. As an example of how these can be used, consider a linearly elastic filament with $A_1 = A_2 = A$ and $\kappa_2 = 0$, which then has

$$\mathbf{e}_3 \times \mathbf{M} = -A\frac{\partial^2 \mathbf{r}}{\partial s^2} - A\kappa_1 \mathbf{e}_2 = -A\hat{\mathcal{L}}_2 \cdot \mathbf{r} - A\kappa_1 \mathbf{e}_2\,, \tag{28}$$

where we have defined a general second derivative matrix operator $\hat{\mathcal{L}}_2$ that computes an arbitrary discretization of the second derivative of its argument. The equation of motion can then be written as

$$\hat{\mathcal{D}} \cdot \left(\mathbf{r}^{t+\Delta t} - \mathbf{r}^t\right) = -A\hat{\mathcal{O}}_2 \cdot \left(\hat{\mathcal{L}}_2 \cdot \mathbf{r} + \kappa_1 \mathbf{e}_2\right) - \hat{\mathcal{O}}_1 \cdot \left(C\left(\Omega_3 - \tau_0\right)\left(\frac{\partial \mathbf{r}}{\partial s} \times \frac{\partial^2 \mathbf{r}}{\partial s^2}\right)\right) + \hat{\mathcal{F}}_3 \cdot \mathbf{r}\,. \tag{29}$$

As can be seen from this, any general constitutive relationship for the moment, in principle, can be used, and the operators $\hat{\mathcal{O}}_1$ and $\hat{\mathcal{O}}_2$ then construct the appropriate discretization of the Kirchoff equations for the moments and forces along the filament.

It should be noted that while we have laid out the algorithm for an approximately inextensible filament with roughly constant Δs, it is straightforward to modify the algorithm to handle extensible filaments. In order to do this, the unevenly-spaced finite difference approximations to the first and second derivatives need to be used to define the tangent vector (Eq. (15)) and the second derivatives of the shape that define Ω_1 and Ω_2. In addition, the control volume lengths L_c will be variable as well as the spacing defined in the operator $\mathcal{O}_2$. With these alterations, extensible filament dynamics can be handled using our method.

4. Advancing the backbone equation in time

In describing the method that we use to time-step the equation of motion (Eq. (16)), we will assume that the filament is a linearly elastic rod with moment given by Eq. (9). Using Eqs. (11) and (24), we can then write the general discrete form of the linear elastic rod equations using a semi-implicit, backward Euler method:

$$\begin{aligned}\hat{\mathcal{D}}\cdot\left(\mathbf{r}^{t+\Delta t}-\mathbf{r}^{t}\right) = &-\left[\hat{\mathcal{O}}_2\cdot\left(A_+\hat{\mathcal{I}}-A_-\left(\mathbf{e}_1^t\mathbf{e}_1^t-\mathbf{e}_2^t\mathbf{e}_2^t\right)\right)\cdot\hat{\mathcal{L}}_2\right]\cdot\mathbf{r}^{t+\Delta t}\\ &-\hat{\mathcal{O}}_2\cdot\left(A_1\kappa_1\mathbf{e}_2^t-A_2\kappa_2\mathbf{e}_1^t\right)\\ &-\hat{\mathcal{O}}_1\cdot\left(C\left(\Omega_3^t-\tau_0\right)\left(\frac{\partial\mathbf{r}^t}{\partial s}\times\frac{\partial^2\mathbf{r}^t}{\partial s^2}\right)\right)+\hat{\mathcal{F}}_3^t\cdot\mathbf{r}^{t+\Delta t}\,,\end{aligned}\tag{30}$$

where $\hat{\mathcal{F}}_3^t$ is the tangential force operator with the forcing terms evaluated at time t. Rewriting the time-stepping algorithm as a linear system of equations,

$$\hat{\mathcal{M}}\cdot\mathbf{r}^{t+\Delta t}=\hat{\mathcal{D}}\cdot\mathbf{r}^t+\mathbf{R}^t\,,\tag{31}$$

with

$$\hat{\mathcal{M}}=\hat{\mathcal{D}}+\left[\hat{\mathcal{O}}_2\cdot\left(A_+\hat{\mathcal{I}}-A_-\left(\mathbf{e}_1^t\mathbf{e}_1^t-\mathbf{e}_2^t\mathbf{e}_2^t\right)\right)\cdot\hat{\mathcal{L}}_2\right]-\hat{\mathcal{F}}_3^t\tag{32}$$

$$\mathbf{R}^t=-\hat{\mathcal{O}}_2\cdot\left(A_1\kappa_1\mathbf{e}_2^t-A_2\kappa_2\mathbf{e}_1^t\right)-\hat{\mathcal{O}}_1\cdot\left(C\left(\Omega_3^t-\tau_0\right)\left(\frac{\partial\mathbf{r}^t}{\partial s}\times\frac{\partial^2\mathbf{r}^t}{\partial s^2}\right)\right)\tag{33}$$

5. The dynamic algorithm for rotations of the orthonormal frame

The second important aspect of our algorithm is how we handle the rotation of the orthonormal frame. If not handled properly, the dynamics can produce drift in the equilibrium configuration of the filament due to a loss in correspondence between the twist, Ω_3, and the orientations of $\mathbf{e}_1$ and $\mathbf{e}_2$ (this drift is especially apparent when thermal fluctuations of chiral filaments are considered). To solve for the dynamics of the orthonormal triad, we only need to timestep $\mathbf{e}_1$, as the orientation of $\mathbf{e}_3$ is determined from the backbone dynamics, $\mathbf{e}_3=\frac{1}{|\partial\mathbf{r}/\partial s|}\frac{\partial\mathbf{r}}{\partial s}$, and $\mathbf{e}_2=\mathbf{e}_3\times\mathbf{e}_1$. The dynamic equation for $\mathbf{e}_1$ comes from Eq. (2):

$$\frac{\partial\mathbf{e}_1}{\partial t}=\omega_3\mathbf{e}_2-\omega_2\mathbf{e}_3\,.\tag{34}$$

Using Eqs. (1),(2), and (5) it is straightforward to show that

$$\omega_2=\mathbf{e}_1\cdot\frac{\partial\mathbf{e}_3}{\partial t}\,,\tag{35}$$

$$\zeta_r\omega_3=C\left(\frac{\partial\Omega_3}{\partial s}-\frac{\partial\tau_0}{\partial s}\right)+A_1\kappa_1\Omega_2-A_2\kappa_2\Omega_1\,.\tag{36}$$

Therefore, because $\Omega_3=\mathbf{e}_2\cdot(\partial\mathbf{e}_1/\partial s)$, we can rewrite the dynamic equation for $\mathbf{e}_1$ (Eq. (34)) as

$$\frac{\partial\mathbf{e}_1}{\partial t}=\frac{1}{\zeta_r}\left(C\frac{\partial}{\partial s}\left(\mathbf{e}_2\cdot\frac{\partial\mathbf{e}_1}{\partial s}-\tau_0\right)+A_1\kappa_1\Omega_2-A_2\kappa_2\Omega_1\right)\mathbf{e}_2-\left(\mathbf{e}_1\cdot\frac{\partial\mathbf{e}_3}{\partial t}\right)\mathbf{e}_3\,.\tag{37}$$

To timestep this equation, we decompose it into its two components and handle each one separately. First, we consider the $\mathbf{e}_2$ component and step to an intermediate value of $\mathbf{e}_1=\mathbf{e}'$ by integrating over the control volumes surrounding each node and using a semi-implicit backward Euler method:

$$\frac{L_c}{\Delta t}\left(\mathbf{e}_i'-\mathbf{e}_{1,i}^t\right)=\frac{C}{\zeta_r}\left(\left(\mathbf{e}_2\cdot\frac{\partial\mathbf{e}'}{\partial s}-\tau_0\right)_{i+1/2}-\left(\mathbf{e}_2\cdot\frac{\partial\mathbf{e}'}{\partial s}-\tau_0\right)_{i-1/2}\right)\mathbf{e}_{2,i}+\frac{L_c}{\zeta_r}\left(A_1\kappa_1\Omega_{2,i}-A_2\kappa_2\Omega_{1,i}\right)\mathbf{e}_{2,i}\,.\tag{38}$$

The terms at the half-grid points are evaluated as

$$\left(\mathbf{e}_2 \cdot \frac{\partial \mathbf{e}'}{\partial s}\right)_{i+1/2} \approx \frac{1}{2\Delta s}\left(\mathbf{e}_{2,i+1} + \mathbf{e}_{2,i}\right) \cdot \left(\mathbf{e}'_{i+1} - \mathbf{e}'_i\right) \tag{39}$$

and Eq. (38) is then solved for $\mathbf{e}'$. To guarantee that $\mathbf{e}'$ is a unit vector, it is renormalized after each time step by dividing by the magnitude of the vector at each grid point.

For the second part of the time stepping, we then consider the equation,

$$\frac{\partial \mathbf{e}'}{\partial t} = -\left(\mathbf{e}' \cdot \frac{\partial \mathbf{e}_3}{\partial t}\right)\mathbf{e}_3 \,. \tag{40}$$

The integral of this equation from time t to time $t + \Delta t$ represents the rotation of $\mathbf{e}'$ that occurs when $\mathbf{e}_3^t$ rotates to $\mathbf{e}_3^{t+\Delta t}$. Therefore, we use a rotation matrix approach to determine the value of $\mathbf{e}_1^{t+\Delta t}$, which is the new value of $\mathbf{e}'$ after the integration of Eq. (40):

$$\mathbf{e}_1^{t+\Delta t} = \mathbf{e}' - \frac{\left(\mathbf{e}' \cdot \mathbf{e}_3^{t+\Delta t}\right)\left(\mathbf{e}_3^t + \mathbf{e}_3^{t+\Delta t}\right)}{1 + \mathbf{e}_3^t \cdot \mathbf{e}_3^{t+\Delta t}} \,. \tag{41}$$

This equation guarantees that the magnitude of $\mathbf{e}_1^{t+\Delta t}$ is the same as the magnitude of $\mathbf{e}'$, and also that $\mathbf{e}_1^{t+\Delta t}$ is perpendicular to $\mathbf{e}_3^{t+\Delta t}$.

6. The full algorithm

Our full finite volume algorithm for the dynamics of filaments, rods, or beams at low Reynolds number is then as follows:

1. Initialize the backbone position of the filament $\mathbf{r}$ and the orientation of $\mathbf{e}_1$ at all grid nodes in a manner consistent with any boundary conditions and with $\mathbf{e}_1$ perpendicular to the tangent direction of the filament.
2. Repeat the following until either the maximum time or maximum number of iterations is reached:
   i. Set κ_1, κ_2, and τ_0 to their values at the current time step.
   ii. Compute $\mathbf{e}_3 = \frac{\partial \mathbf{r}/\partial s}{|\partial \mathbf{r}/\partial s|}$ using an appropriate discretization of the first derivative. Then compute $\mathbf{e}_2 = \mathbf{e}_3 \times \mathbf{e}_1$.
   iii. Compute the components of the strain vector using Eq. (1) such that $\Omega_1 = -\mathbf{e}_2 \cdot \left(\partial^2 \mathbf{r}/\partial s^2\right)$, $\Omega_2 = \mathbf{e}_1 \cdot \left(\partial^2 \mathbf{r}/\partial s^2\right)$, and $\Omega_3 = \mathbf{e}_2 \cdot (\partial \mathbf{e}_1/\partial s)$. The second derivative of $\mathbf{r}$ should be computed with the corresponding approximation to the derivative that was used for the first derivative in the preceding step.
   iv. Compute the tangential force F_3 at the half grid nodes using Eq. (20).
   v. Time step $\mathbf{r}$ using Eq. (31).
   vi. Time step $\mathbf{e}_1$ by first solving for the intermediate value $\mathbf{e}'$, which represents the rotation of $\mathbf{e}_1$ into $\mathbf{e}_2$ using Eq. (38). Then rotate $\mathbf{e}'$ into $\mathbf{e}_3$ using Eq. (41).

7. Results

We now describe the application of this algorithm to a number of test problems. For all of the test problems, our algorithm was implemented in MATLAB® and we ran our simulations on standard desktop or laptop computers, running Windows 10. Solution of the linear systems of equations was carried out with the backslash operator (which uses UMFPack to do a direct solution of the linear system). Most simulations took on order of tens of minutes to a couple hours to complete when using an Intel® Core i7-4790 3.6 GHz with 4 cores. Unless otherwise noted, all simulations assume motion in water, $\eta = 10^{-3}$ Pa·s, with resistive drag coefficients $\zeta_\perp = 2\zeta_\parallel = 4\pi\eta$.

7.1. The shortcomings of a finite difference approach

Before we begin describing the results of our test cases, we first highlight the downfalls of a straightforward implementation of a finite difference approach using a simple model of an undulating, filamentary swimmer: a free filament undergoing planar traveling wave undulations, as was originally examined theoretically by Taylor [31] and has been used as a test case for computational methods (e.g., [32]). We consider a filament of length $L = 1$ with circular cross-section, such that $A_1 = A_2 = 1$, $C = 1$, and preferred curvatures and torsion $\kappa_1 = \epsilon \sin(ks - \omega t)$ and $\kappa_2 = \tau_0 = 0$, with $\epsilon = 8$, $k = 4\pi$, and $\omega = 10$. Because we are considering an integer number of wavelengths, at low Reynolds number the filament should translate along a straight line at constant velocity. This scenario leads to the following force per length

$$\frac{\partial \mathbf{F}}{\partial s} = -\frac{\partial^4 \mathbf{r}}{\partial s^4} - \frac{\partial^2 (\kappa_1 \mathbf{e}_2)}{\partial s^2} + \frac{\partial}{\partial s}\left(\Omega_3 \left(\mathbf{e}_3 \times \frac{\partial^2 \mathbf{r}}{\partial s^2}\right)\right) + \frac{\partial}{\partial s}\left(\left(F_3 - \left(\left(\Omega_1 - \kappa_1\right)\Omega_1 + \Omega_2^2\right)\right)\mathbf{e}_3\right) \,. \tag{42}$$

Rather than using the discretization described in Sec. 3, we instead compute the derivatives in the top line of Eq. (42) using sixth-order accurate, finite-difference approximations similar to the method described in [16]:

$$\frac{\partial \mathbf{F}}{\partial s} = -\hat{\mathcal{L}}_4 \mathbf{r} - \hat{\mathcal{L}}_2 (\kappa_1 \mathbf{e}_2) + \hat{\mathcal{L}}_1 \left(\Omega_3 \left(\mathbf{e}_3 \times \frac{\partial^2 \mathbf{r}}{\partial s^2} \right) \right) + \frac{\partial F_3'}{\partial s} \mathbf{e}_3 \,, \tag{43}$$

where $\hat{\mathcal{L}}_m$ is the sixth-order accurate approximation to the mth derivative, with these derivatives given by

$$\begin{aligned}
\left.\frac{\partial f}{\partial s}\right|_i &= \frac{1}{60\Delta s} \left(f_{i+3} - 9 f_{i+2} + 45 f_{i+1} - 45 f_{i-1} + 9 f_{i-2} - f_{i-3} \right) \,, \\
\left.\frac{\partial^2 f}{\partial s^2}\right|_i &= \frac{1}{2\Delta s^2} \left(\frac{1}{45} f_{i+3} - \frac{3}{10} f_{i+2} + 3 f_{i+1} - \frac{49}{9} f_i + 3 f_{i-1} - \frac{3}{10} f_{i-2} + \frac{1}{45} f_{i-3} \right) \,, \\
\left.\frac{\partial^3 f}{\partial s^3}\right|_i &= \frac{1}{8\Delta s^3} \left(-f_{i+3} + 8 f_{i+2} - 13 f_{i+1} + 13 f_{i-1} - 8 f_{i-2} + f_{i-3} \right) \,, \\
\left.\frac{\partial^4 f}{\partial s^4}\right|_i &= \frac{1}{6\Delta s^4} \left(-f_{i+3} + 12 f_{i+2} - 39 f_{i+1} + 56 f_i - 39 f_{i-1} + 12 f_{i-2} - f_{i-3} \right) \,,
\end{aligned}$$

and we handle the discretization of the derivative of F_3' as described in Sec. 3. We also timestep $\mathbf{e}_1$ using the procedure described in Sec. 5. (How $\mathbf{e}_1$ is integrated in time is not overly important because the preferred curvature drives a planar waveform that does not induce any twist in the filament, and the orthonormal frame does not rotate about the tangent vector during the simulations.)

Because the finite difference derivatives do not directly impose the free boundary conditions at the ends, we impose these by setting the following conditions at the ends ($s = 0, 1$):

$$\begin{aligned}
&\hat{\mathcal{L}}_2 \mathbf{r} = -\kappa_1 \mathbf{e}_2 \\
\mathbf{e}_2 \cdot &\frac{\partial \mathbf{e}_1}{\partial s} = 0 \\
&\hat{\mathcal{L}}_3 \mathbf{r} = -\hat{\mathcal{L}}_1 (\kappa_1 \mathbf{e}_2)
\end{aligned}$$

These boundary conditions define the values of the $i = 1, 2, N-1$ and N nodes and are imposed by evaluating the right-hand side of each equation at the previous time step and using those values to set the value of the left-hand side at the current time step.

We ran simulations for a total time of 4.0 using 128 nodes, a time step $\Delta t = 10^{-5}$, and $\sigma_0 = 1.6 \times 10^4$. As is shown in Fig. 3a, over the course of the simulation the swimmer maintains a sinusoidal shape, but there are non-physical rotations and translations that occur. These translations and rotations are owed to the finite difference discretization not preserving the zero net force and torque conditions for the free swimmer. Indeed, the undulating filament ends up translating in the positive x direction (Fig. 3b), even though a filament undergoing sinusoidal traveling wave oscillations propagating in the positive x direction should translate in the negative x direction.

As a comparison, we used the same parameters in our finite volume algorithm and found that the filament undergoes traveling wave oscillations and swims at roughly constant speed along the negative x direction (Fig. 3c,d). In agreement with theoretical predictions, we found a swimming speed of ~ 0.106. The resistive force calculation using an amplitude $2\epsilon/k^2$ predicts a speed of 0.114, when $\zeta_\perp/\zeta_\parallel = 2$, $\epsilon = 8$, and $\omega = 10$ [33]. Therefore, our finite volume algorithm provides good agreement with the behavior that is predicted theoretically, while the finite difference implementation is prone to fictitious translations and rotations, even in this simple test case.

As we have mentioned previously, the aspect of our algorithm that enforces proper balance of the forces and moments between segments of the filament is completely encoded within our $\mathcal{O}_1$ and $\mathcal{O}_2$ operators, with the integral of the forces per length along any segment (i.e., control volume) of the filament given by

$$\int \mathbf{f} \mathrm{d}s = \hat{\mathcal{O}}_2 \cdot (\mathbf{e}_3 \times \mathbf{M}) - \hat{\mathcal{O}}_1 \cdot \left(M_3 \left(\frac{\partial \mathbf{r}}{\partial s} \times \frac{\partial^2 \mathbf{r}}{\partial s^2} \right) \right) + \hat{\mathcal{F}}_3 \cdot \mathbf{r} \,, \tag{44}$$

where the moment can be defined and discretized as appropriate to the problem. Any filament dynamics algorithm that works with the force per length (or the integral of the force per length), can then employ our methodology. As an example, the immersed boundary (IB) method simulates fluid-structure interactions by smearing the integral of the force per length due to a filamentary object out over a nearby set of Eulerian nodes on which the fluid velocity is to be computed [34]. This procedure is often represented mathematically by defining the force density that the filament exerts on the fluid:

$$\mathbf{f}_f = \int -\mathbf{f}(s,t)\, \delta_c (\mathbf{x} - \mathbf{r}(s,t)) \mathrm{d}s \,, \tag{45}$$

where the integral represents the discrete smearing of the force applied from the Lagrangian filament node located at $\mathbf{r}(s,t)$ onto the Eulerian grid nodes at $\mathbf{x}$, and the delta function δ_c is a discrete representation of the Dirac delta function [34]. This smeared-out force density $\mathbf{f}_f$ drives the fluid motion, and the fluid velocity is then interpolated back onto the nodes

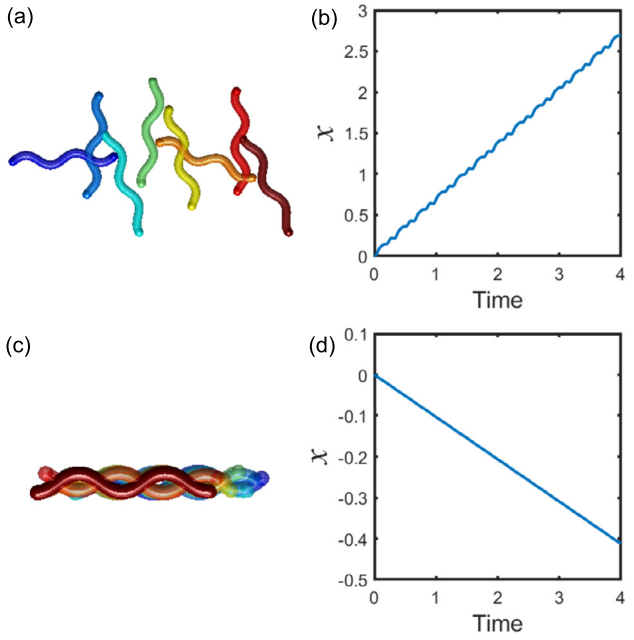

Fig. 3. Comparison of the finite difference algorithm (a) to the results of our finite volume algorithm (c) for a filamentary, undulating swimmer driven by a sinusoidal traveling wave. The images show a stroboscopic montage where the coloring of the filament denotes time. The initial shape is shown in dark blue, and transitions to red for the last frame. The x-coordinate of the center of mass for the finite difference (b) and finite volume methods (d) are also shown. Both simulations use the same parameters as given in the text. (a,b) The finite difference algorithm produces fictitious forces and torques that cause the filament to rotate and translate in a non-physical manner, with the center of mass translating in the positive x direction, opposite of what should occur for filament undulations that travel in the positive direction. (c,d) The finite volume algorithm gives a steadily propagating traveling wave along the filament that produces a nearly constant translation in the negative x direction, as is expected. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

that define the swimmer in order to define the local velocity of the swimmer (Fig. 4). To implement our method into the IB method, the force in Eq. (44) can be used to define the force from a specific node in Eq. (45). To confirm that our method is easily implemented in the IB method, we modified the IB code provided in [35] to simulate the undulating swimmer defined above using our force algorithm (Eq. (44)). We ran simulations on a $64 \times 64 \times 64$ eulerian grid where each volume of fluid is $(2\Delta s)^3$ using a time step $\Delta t = 10^{-4}$, and 41 Lagrangian nodes for the immersed swimmer. We found stable swimming of the undulating filament with the expected velocity field in the fluid as shown in Fig. 4 and consistent with what has been seen in previous IB simulations [32].

7.2. Undulating swimmer algorithm convergence rate

The free filament undulating as a traveling wave with spatiotemporally-varying preferred curvature $\kappa_1 = \epsilon \sin(ks - \omega t)$ also provides a means for evaluating the accuracy and stability of our finite volume algorithm. Using this scenario and the same parameters given in Sec. 7.1, we ran simulations for a total time $t_f = 1.0$, varying either the grid spacing Δs or the time step Δt. We then compared simulations with grid spacings Δs and $\Delta s/2$ run with the same time step, by computing the L^2-norm difference of the net displacements of the nodes between the two simulations, which was calculated as

$$L^2_{\Delta s} = \sqrt{\sum_{i=1}^{N} \left| \left(\mathbf{r}^{\Delta s}_{i,t_f} - \mathbf{r}^{\Delta s}_{i,0} \right) - \left(\mathbf{r}^{\Delta s/2}_{2i-1,t_f} - \mathbf{r}^{\Delta s/2}_{2i-1,0} \right) \right|^2}, \tag{46}$$

where N is the total number of nodes and $\mathbf{r}^{\Delta s}_{i,t}$ is the position of the ith node at time t for the simulation with grid spacing Δs. This definition compares every node in the Δs simulation with its corresponding node in the $\Delta s/2$ simulation. For simulations where the time step was varied, we computed the L^2-norm differences between the net displacements from simulations run at Δt with those at $\Delta t/2$. In addition to varying the grid spacing and time step, we also examined how changing the accuracy of the derivatives that are used in computing the moments affected the L^2-norms. We compared using standard 2nd-order central difference derivatives to the 6th-order accurate derivatives described above. The L^2-norm errors are given in Table 1.

We then computed the rate of convergence, m, by assuming that numerical error scaled as a power of the grid spacing like Δs^m (or as a power of the time step). Then, the spatial rate of convergence is given by

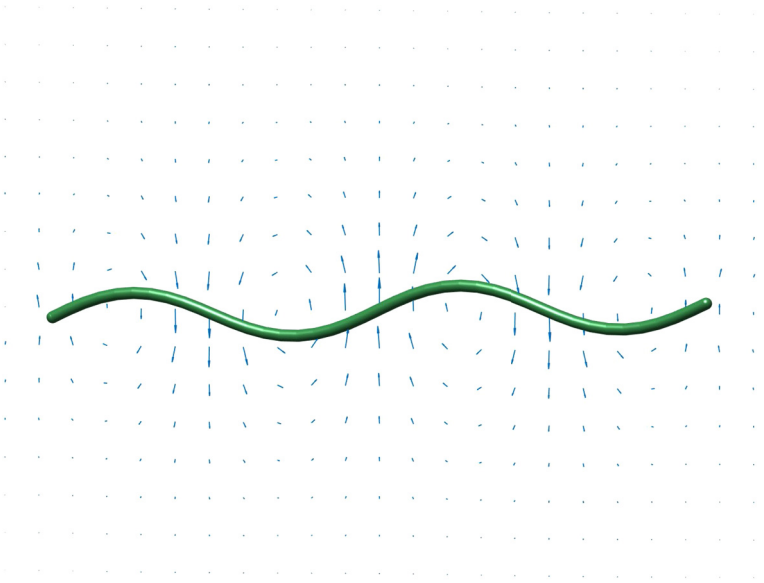

Fig. 4. Snapshot from an immersed boundary simulation that uses the finite volume algorithm to define the force per length along an undulating filamentary swimmer immersed in a fluid, as described in the text. The simulation is carried out in 3 dimensions, with the sinusoidal swimmer moving only in the x-y plane. Arrows represent the fluid flow on the Eulerian grid.

Table 1
The spatial and temporal L^2-norm differences and rates of convergence for the undulating swimmer using 2nd and 6th order accurate spatial derivatives. For testing the spatial convergence, we used a fixed $\Delta t = 10^{-6}$ and varied Δs between 0.004 and 0.1. For the temporal convergence, we used a fixed $\Delta s = 0.016$ and varied Δt between 10^{-6} and 4×10^{-6}. The listed $L^2_{\Delta s}$ has $\Delta s = 0.016$.

Derivative order	Varying Δs		Varying Δt	
	$L^2_{\Delta s}$ error	m	$L^2_{\Delta t}$	m
2nd	8.93×10^{-3}	1.98	7.63×10^{-5}	1.00
6th	1.21×10^{-3}	2.13	7.64×10^{-5}	1.00

$$m \ln 2 = \ln \left(\frac{L^2_{\Delta s} - L^2_{\Delta s/2}}{L^2_{\Delta s/2} - L^2_{\Delta s/4}} \right) , \tag{47}$$

with an analogous expression for the temporal rate of convergence. The results for the rates of convergence are given in Table 1. As expected from our use of a Backward Euler scheme, we find that the temporal convergence is linear in the time step. In addition, we find that the spatial rate of convergence is quadratic in the grid spacing, regardless of whether we use 2nd-order central difference derivatives or 6th-order accurate derivatives in defining the moment. However, using 6th-order accurate derivatives does improve the accuracy of the method by about a factor of 7. The 6th-order accurate derivatives probably do not alter the rate of convergence due to the fact that our discretization of the tangential forces is only second-order accurate.

The example above is a simple case of internal force actuation, where the spatiotemporally varying preferred curvature that we use could equally well be thought of as an active moment created by off-axis internal forces, similar to models that have been suggested for eukaryotic cilia and flagella (e.g., [36,37]). Any active moments can be implemented in our algorithm in the same way as is described for the term $A_1\kappa_1\mathbf{e}_2 - A_2\kappa_2\mathbf{e}_1$ in Eq. (30). Active forces per length can be added by integrating the force per length over a control volume and adding the resultant force to the right-hand side of this equation (similar to how we handle thermal forces, see Sec. 7.4).

7.3. Helical extension in an external flow

The previous test problem shows that our algorithm can handle the case of a free, moving rod under the action of internal forces. To explore a case where one end of the filament is anchored and external forces are applied from motion of the surrounding fluid, we consider a helix, clamped at one end with the other end free. A linear fluid flow is then applied in the direction parallel to the long axis of the helix, causing the helix to undergo axial extension. This test case was chosen as it has been previously examined experimentally [38], thereby providing data with which to further validate our algorithm.

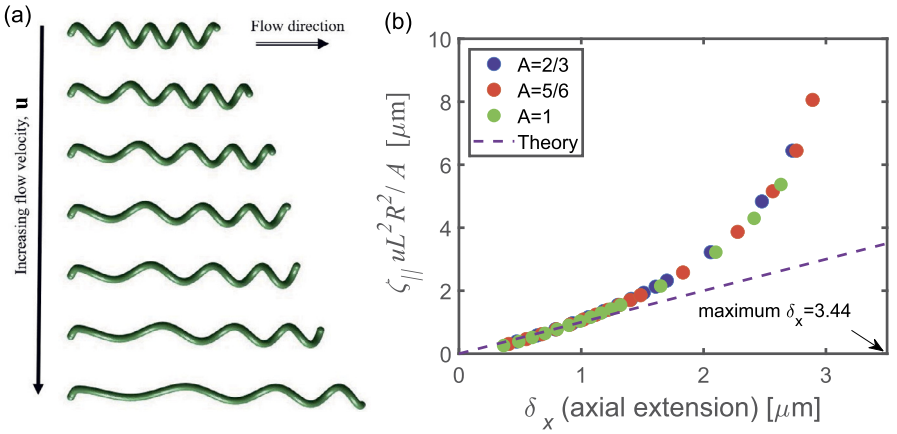

Fig. 5. Axial extension of anchored elastic helices in external fluid flows. (a) Steady state shapes of the deformed helices as the fluid flow magnitude is increased, with the magnitude of flow velocity increasing from top to bottom. The helix is clamped at the left end, while the right end is free. The parameters are as described in the text with $A = 1$ pN·μm and choosing a range of representative values for linear flow velocity $\mathbf{u}$. (b) Axial extension versus the dimensionless velocity $\zeta_{\parallel} u R^2 L^2 / A$. The relationship between the axial extension and the dimensionless velocity agrees well with the theoretical results Eq. (49) for small extensions, and diverges in a similar fashion as was observed experimentally in [38]. Each color represents a different bending modulus as represented in the figure legend (color online).

We consider a helix of length $L = 5.0$ μm and cross-sectional radius $a = 0.1$ μm, with an equilibrium radius of curvature $R = 0.3$ μm, torsion $\tau = 1$ μm^{-1}, curvature $\kappa = 3$ μm^{-1}, and the helical axis initially aligned along the x axis. The left end of the helix is clamped in place, with an equilibrium end-to-end distance $L_e \approx 1.56$ μm. A fluid flow $\mathbf{u} = u\hat{\mathbf{x}}$ is applied, and the simulation is run until the helix comes to a steady state configuration (determined by the point when the velocity of the right-end node of the helix is less than 0.01 $\frac{\mu m}{s}$). To account for the applied fluid velocity $\mathbf{u}$, we redefine the resistive force on the filament using the relative velocity of the filament with respect to the external fluid. Eq. (6) then becomes,

$$\zeta_{\perp}\left[\hat{\mathcal{I}} + \left(\frac{\zeta_{\parallel}}{\zeta_{\perp}} - 1\right)\mathbf{e}_3\mathbf{e}_3\right] \cdot \left(\frac{\partial \mathbf{r}}{\partial t} - \mathbf{u}\right) = \frac{\partial \mathbf{F}}{\partial s}\,. \tag{48}$$

We ran a number of simulations using fluid speeds u ranging from 0.3 $\frac{\mu m}{s}$ to 1.8 $\frac{\mu m}{s}$ in increments of 0.15 $\frac{\mu m}{s}$, with a fluid viscosity $\eta = 0.1$ Pa·s, and bending moduli $A = \frac{2}{3}$, $\frac{5}{6}$, and 1. For each simulation, we measured the steady state axial extension δ_x. As expected, increasing the fluid velocity at constant bending modulus stretches out the helix more, with the helix stretching most near the anchor point, while remaining helical near the free end (Fig. 5a). The shapes that we find are in good agreement with the shapes that were observed experimentally [38]. In addition, previous work suggests that the axial extension of an elastic helix in linear flow should be given by [38,39]

$$\delta_x = \frac{\zeta_{\parallel} u R^2 L^2}{A}\left[1 + \mathcal{O}\left(\frac{R}{L}\right)\right], \tag{49}$$

where the last term is small, since it is of the order of the ratio of the helix radius to the total length of the filament. This equation suggests that the axial extension should scale linearly with the length parameter, $\zeta_{\parallel} u R^2 L^2 / A$. Consistent with these prior results, we find that for small extensions, δ_x does scale linearly with the length parameter, as predicted by Eq. (49), and at high extensions (above 1.2 μm in our simulations) the length parameter diverges as the extension approaches the maximal extension, $L - L_e$, which agrees with the experimental results [38] (Fig. 5b). Likewise, we find that our axial extensions collapse to a single curve when plotted against the dimensionless velocity, even when the bending modulus is varied, as predicted theoretically.

7.4. Thermal diffusion of a helical filament

Simulation of inextensible elastic filaments undergoing random thermal motion remains an open area of research [22–24]. For filamentary objects at the scale of microns and smaller, as are commonly encountered in biology (e.g., DNA, actin, microtubules, rod-shaped bacteria), random thermal forces can play a dominant role in the dynamics. Adding thermal forces to our algorithm is straightforward, as these forces come in as an external body force per length that act at each point along the filament. We can add these forces in by defining the external force per length in Eq. (6) as being a random, uncorrelated vector field $\mathbf{f} = \boldsymbol{\xi}$, with correlation function $\langle \xi_i(s,t)\,\xi_j(s',t')\rangle = \lambda^2 \delta(s-s')\,\delta(t-t')\,\delta_{ij}$, where $\delta(s)$ is the Dirac delta function and δ_{ij} is the Kronecker delta. The value of λ^2 is determined from the Stokes-Einstein relationship. Specifically, we decompose the thermal force per unit length into components along the orthonormal triad directions, $\boldsymbol{\xi} = \xi_1\mathbf{e}_1 + \xi_2\mathbf{e}_2 + \xi_3\mathbf{e}_3$, and consider the autocorrelation of one of these components integrated over a length Δs and a small time Δt:

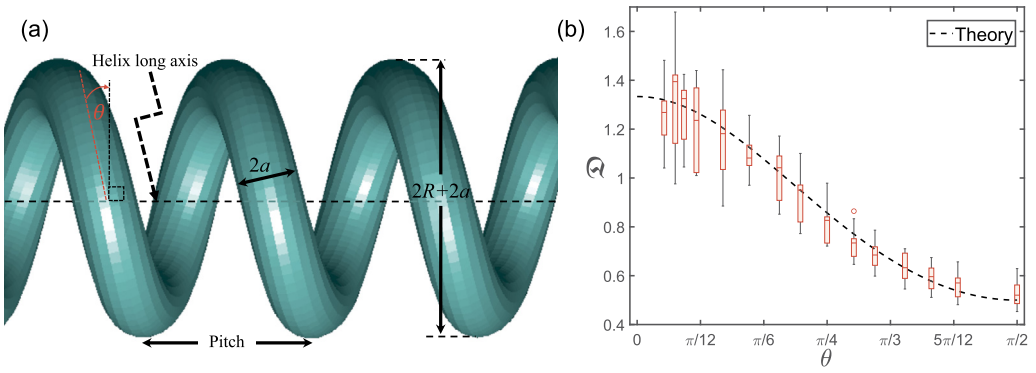

Fig. 6. (a). Helical filament showing the relationship between the tangent vector (shown as red dashed line), helical long axis, helix angle (θ), filament radius (a), and the helix radius (R). (b) Ratio of perpendicular and parallel diffusion with respect to the long axis of the helix. A toroidal filament (pitch angle $\theta = 0$) has a diffusion ratio around 4/3. As θ is increased, the diffusion ratio decreases, eventually going to 1/2 as $\theta \to \pi/2$. These results match well with the theoretical prediction given in Eq. (57) (dashed line). For each value of θ, we used $\tau = 1$ and varied the curvature as $\kappa = \cot\theta$. The 1st quartile, median, and 3rd quartiles are shown as horizontal red lines. Ten simulations were done for each value of θ.

$$\begin{aligned}\iiiint \xi_i(s,t)\,\xi_i(s',t')\,ds'dsdtdt' &= \lambda^2 \iiiint \delta(s-s')\,(t-t')\,ds'dsdtdt' \\ &= \lambda^2 \Delta s \Delta t\,. \end{aligned} \tag{50}$$

Then, from the Stokes-Einstein relationship we have

$$\lambda^2 = \frac{2\zeta_{\perp,\parallel} k_B T}{\Delta s \Delta t}\,, \tag{51}$$

where $\zeta_{\perp(\parallel)}$ is used for forces perpendicular (parallel) to the tangent direction. Note that this implies that we can write the thermal force per unit length as

$$\boldsymbol{\xi} = \sqrt{\frac{2k_B T}{\Delta s \Delta t}}\left[\sqrt{\zeta_\perp}\boldsymbol{\eta} + \left(\sqrt{\zeta_\parallel} - \sqrt{\zeta_\perp}\right)(\boldsymbol{\eta}\cdot\mathbf{e}_3)\,\mathbf{e}_3\right]\,, \tag{52}$$

where $\boldsymbol{\eta}$ is a normally distributed random vector with unit variance.

In our finite volume representation (Eq. (16)), we need to consider the integral of the random force over each control volume, L_c, over a time step Δt. Therefore, the integrated force term that goes into our algorithm is

$$L_c \Delta t \boldsymbol{\xi} = \sqrt{2k_B T L_c \Delta t}\left[\sqrt{\zeta_\perp}\boldsymbol{\eta} + \left(\sqrt{\zeta_\parallel} - \sqrt{\zeta_\perp}\right)(\boldsymbol{\eta}\cdot\mathbf{e}_3)\,\mathbf{e}_3\right]\,, \tag{53}$$

which is well-behaved when Δs and/or Δt go to zero.

As a test of the applicability of our algorithm to problems involving this type of random forcing, we consider the diffusion of free helical filaments. A helix provides a good test case as it has a fixed angle between the tangent and the helical long axis (Fig. 6a). We can then explore how diffusion is affected as the pitch angle of the helix θ is changed, which tests the validity of our implementation of Eq. (53). We consider helices with pitch angles that range from $\frac{\pi}{2}$ (straight cylindrical filament) to zero (a circular filament) and quantify the ratio of the diffusion coefficient for movements perpendicular to the long axis of the helix relative to diffusion parallel to it. The expected value of this ratio can be derived from the n-dimensional mean squared displacement (MSD), $\text{MSD} = 2ntD$ where D is the Brownian diffusion coefficient and t is time. The tangent vector for a helix aligned with the z axis is

$$\mathbf{e}_3 = -\cos\theta\left(\sin\left(\frac{s\sin\theta}{R}\right)\hat{\mathbf{x}} - \cos\left(\frac{s\sin\theta}{R}\right)\hat{\mathbf{y}}\right) + \sin\theta\hat{\mathbf{z}}\,, \tag{54}$$

where the pitch angle is $\theta = \tan^{-1}(\frac{\tau}{\kappa})$, and the curvature and torsion of the helix are $\kappa = \cos^2\theta/R$ and $\tau = \sin\theta\cos\theta/R$, respectively. The effective drag coefficients for motions parallel and perpendicular to the z axis are then

$$\zeta_z = \zeta_\parallel \sin^2\theta + \zeta_\perp \cos^2\theta\,, \tag{55}$$

$$\zeta'_\perp = \frac{1}{2}\left[\zeta_\perp\left(1+\sin^2\theta\right) + \zeta_\parallel\cos^2\theta\right]\,. \tag{56}$$

The Stokes-Einstein relation then predicts that the ratio of the diffusion coefficients for motion perpendicular to the helical axis relative to motion parallel to it is $\mathfrak{D} = \frac{D'_\perp}{D_\parallel} = \frac{\zeta_z}{\zeta'_\perp}$, which, using $\frac{\zeta_\parallel}{\zeta_\perp} = \frac{1}{2}$, is

$$\mathfrak{D} = \frac{2\cos^2\theta + \sin^2\theta}{1 + \sin^2\theta + 0.5\cos^2\theta}\,. \tag{57}$$

To test this prediction, we ran simulations using a helix of total length $L = 10.0$ μm, with radius $R = \sin\theta\cos\theta$, bending modulus $A = 1.0$ pN·μm, compressibility coefficient $\sigma_0 = 2000$, fluid viscosity $\eta = 0.1$ Pa·s, and thermal energy $k_BT = 0.004114$ pN·μm. Simulations were run for a total time 1.5 s, using 500 spatial nodes, and a time step $\Delta t = 10$ μs. The normally distributed random forces, $\boldsymbol{\eta}$, were generated using random numbers produced by the MATLAB 2021a Mersenne Twister generator using seed numbers 1→10. We varied the pitch angle from θ = 0 to $\frac{\pi}{2}$ and computed the displacement of the center of mass for motions perpendicular and parallel to the current long axis of the helix. We used displacements over time intervals of $d\tau = 1$ ms when computing the MSD. These simulations show good agreement with the expected result from Eq. (57) (Fig. 6b).

This method for calculating the diffusion of a helix is accurate to our predicted diffusion Eq. (57), however, if the pitch of the helix becomes small, the diffusive behavior of the helix changes. In these cases, using a method which allows for hydrodynamic interactions, such as the IB method, is more accurate. In the case where the pitch becomes very small, the helix will diffuse similar to a large cylinder of radius $r_{cyl} \approx R$. Although we did not implement a stochastic IB method here, we have included how to implement a simple case of a 2D swimmer with the IB method and our algorithm (see Fig. 4).

7.5. Buckling of an elastic filament immersed in a thermal fluid at a hyperbolic flow stagnation point

Complex fluid flows can occur in engineering and biology at the microscopic level. For instance, in a cell's cytoplasm, molecular motors translating along elastic actin filaments entrain the surrounding cytoplasmic fluid. The resulting fluid flow can then further bend, stretch, twist, or buckle these slender cytoskeletal fibers inside the cell [40]. Recent experimental work sought to understand the dynamics of actin filaments in shearing flows by examining the deformations that occurred when actin was introduced into a hyperbolic, low Reynolds number flow [41]. In this work, it was found that thermal fluctuations produce a small pre-bend in the actin filaments, which then allowed the fluid flow to fully buckle the actin. In addition, it was found that the fluid-induced bending deformations were crucially dependent on a single parameter Σ that depended on the rate of strain of the fluid γ [40,41]:

$$\Sigma = \frac{2\eta\gamma L^4}{\pi^3 A \ln\left(\frac{L^2}{4a^2 e}\right)}\,, \tag{58}$$

where η is viscosity of the fluid, L is the length of the filament, a is the filament body radius, and e is Euler's number.

Systems such as these provide an excellent test case of the functionality of our algorithm for situations where external flows and random thermal fluctuations are both at play. To test our algorithm against the experimental results, we used experimental results from 2 dimensional [41] and 3 dimensional studies [42,43]. To begin, we examined the 2 dimensional case in which we simulated a straight filament centered at a hyperbolic Stokesian flow stagnation point where the velocity profile is $\mathbf{u} = \gamma\,(-x, y, 0)$ (Fig. 7), and aligned along the positive x direction, which, when γ is positive, indicates compressional flow (whereas a negative value would indicate stabilizing, extensional flow). This external fluid velocity term is handled the same way as in Eq. (48). The filament will undergo an Euler buckling transition under these conditions as long as there is a transverse force also present (such as that found from stochastic thermal forces). Although the stochastic forces allow for the buckling to occur, our persistence lengths $\ell_p = \frac{A}{k_BT}$ used in this experiment are on the order of, or larger than, the length of the filament. Therefore, in our simulations, any large-scale buckling dynamics observed can be attributed to the Stokesian flow mentioned above, not directly to thermal stochastic forces. Using this experimental setup, we observed the same major bending modes which follow from the eigenvalues from the Euler-Lagrange equation [41]. In these simulations, we include thermal forces in the x and y dimensions only in order to reduce any motion in the z dimension. Our results (Fig. 7) compare very well with the published results [41,44], wherein a non-dimensional tip-to-tip length of the filament was defined $P = 1 - \frac{\delta}{L}$. The minimum tip-to-tip length, δ, of the filament is reduced as Σ increases. Also, the major buckling modes which we observe in our simulations also correlate to buckling modes theoretically and experimentally shown at the previously reported Σ values [41].

Next, we examined the three dimensional case where the hyperbolic flow is still $\mathbf{u} = \gamma\,(-x, y, 0)$ for all z dimension points. In this case, we include the full three dimensional thermal force, $\boldsymbol{\xi}$, and add in an additional stochastic thermal moment. This stochastic thermal moment per unit length, $\boldsymbol{\xi}_m$ can be derived in the same fashion as Eq. (53), in which our resulting thermal moment is $L_c\Delta t\boldsymbol{\xi}_m = \sqrt{2k_BTL_c\Delta t}\left[\sqrt{\zeta_r}\mathbf{e}_3\right]$. Adding this to our algorithm is easiest when including it directly into Eq. (36) on the right hand side. We then defined an effective radius of helicity about the center of the filament, as in [42,43], $R_{eff} = \sqrt{y(s)^2 + z(s)^2}$, which is evaluated between $\frac{L}{4} < s < \frac{3L}{4}$ where the buckling is most pronounced. In these cases, we expect the radius of buckling to follow $R_{eff} \propto \left(\frac{\eta\gamma}{A_p}\right)^{\frac{-1}{4}}$. Our effective non-dimensionalized radius, $\frac{R_{eff}}{L}$, versus $\bar{\mu}$ is defined as

$$\bar{\mu} = \frac{8\pi\eta\gamma L^4}{cA_p} \tag{59}$$

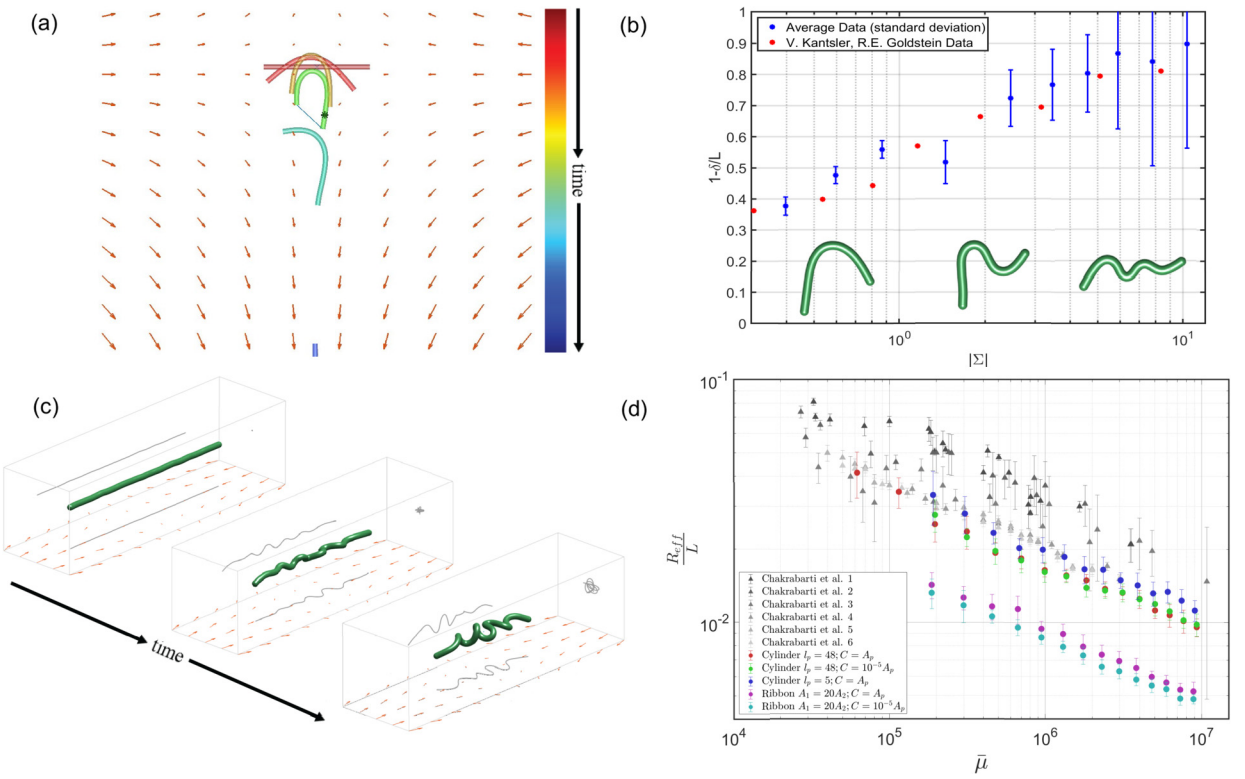

Fig. 7. 2D and 3D filament buckling caused by Stokesian flow under thermal fluctuations. a) Stroboscopic montage of a single filament undergoing buckling over time due to compressional flow. Time indicated by color of the filament (color online). '*' indicates the time point at which filament's opposite ends are closest together (representative of filament buckling under conditions such that $0 < \Sigma < 1$. b) Non-dimensionalized tip to tip length versus Σ Eq. (58). Our data (blue) show correlation to data from [41]. Inlays of buckled filaments show the dominant buckling mode for the Σ ranges indicated below. In 2D simulations, Σ was varied by adjusting the following parameters within the given ranges: $A_1 = A_2 = 0.1\text{-}3$ pN·μm, $\gamma = 30\text{-}60$ s^{-1}, $\eta = 1\text{-}15$ cP, $L = 3\text{-}4$ μm, and $\sigma = (1\text{-}5) \cdot 10^3$ pN·μm. c) Time sequential images of 3D filament buckling correlating to results of [42,43] which indicated helix formation in a hyperbolic flow. d) Combination of Chakrabarti et al.'s data and our simulation data plotting largest effective radius of the buckled helix versus non-dimensionalized viscosity [43]. All data show the correct correlation to Eq. (59).

where c is a constant that is defined by the geometry of the filament, $c = -\ln(-\alpha^2 e)$, α is the aspect ratio, and e is euler's number. Our simulations of 3D buckling under stochastic forces and hyperbolic flows (Fig. 7d) correlate well to the previous experimental and simulation data from Chakrabarti et al. and also match the expected scaling result, $R_{eff} \propto \left(\frac{A_p}{\eta\gamma}\right)^{\frac{1}{4}}$. In the previous papers [42,43], a twist modulus was not utilized in the energy functional of the filament. This being the case, in (Fig. 7d) we also included comparative data in which our twist modulus was taken to be negligibly low in comparison to $\boldsymbol{\xi}_m$. Our results for modifying twist modulus on a cylindrical filament which uses $A_1 = A_2 = A_p$ show only very small changes to our slope and the absolute value for effective radius. One interesting feature is when $A_1 = 20A_2$, which describes a ribbon-like filament which is much more susceptible to buckling along the $\mathbf{e}_2$ dimension than that of $\mathbf{e}_1$. In this case, the dominant effect driving a change in effective radius will be the weaker dimension's bend modulus. Substituting A_2 for A_p in this case is warranted, however, for ease of plotting, we did not shift the x-axis of our ribbon data. Reducing the twist modulus on a ribbon further reduces the effective radius of the buckling filament since there are 2 degrees of freedom which each have a very small modulus. The small twist and bend moduli lead to an increase propensity for our simulation to stop prematurely due to the filament coming into contact with itself. More regarding our finite volume algorithm and ribbons is included in a later section.

7.6. The twirling to whirling transition

The preceding test cases included twist of the filament, but did not directly highlight the applicability of our algorithm for handling the dynamics of twist. Therefore, we chose to examine a twist-driven instability to examine this functionality. We consider the low Reynolds number dynamics of an elastic filament that is rotated about its tangent vector from an end that is clamped in place. The other end of the filament is free. This problem was originally examined analytically and using a weakly nonlinear theory in [45], where it was found that for low rotational speeds, the filament remained straight. Then, at a critical rotational frequency $\omega_c \simeq 8.9\left(A/\zeta_r L^2\right)$, the filament buckles and begins to *whirl*, where the bent backbone of the filament rotates about the original filament direction at a speed $\chi \ll \omega$, while still rotating about its tangent vector

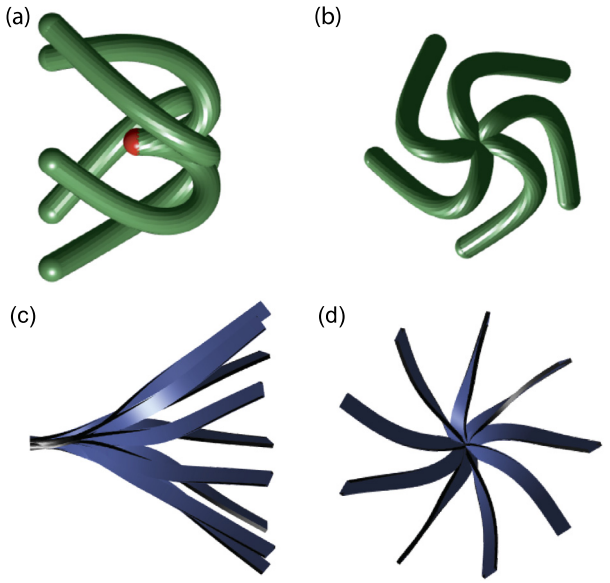

Fig. 8. Twirling to large amplitude whirling. Stroboscopic montage of a side (a) and end-on (b) view of the steady-state shape of the whirling filament when the angular twirling speed is just above onset of the instability, $\omega = 9A/\zeta_r L^2$. As seen in previous simulations and experiments, the filament takes on a large amplitude whirling shape after onset of the whirling instability. The red coloring denotes the clamped end of the filament. The side (c) and end-on (d) views of the whirling state for a filament with a non-circular cross-section ($A_1 = 1$, $A_2 = 0.05$) when $\omega = 5A_1/\zeta_r L^2 = 9.5A_+/\zeta_r L^2$. When the bending moduli are substantially different than each other, the amplitude of the steady state shape grows with the distance from onset of the instability. Consequently, the ribbon shape shown here does not buckle back on itself, as it does for the filament in (a,b).

at a rotational speed comparable to the turning frequency ω [45]. Simulating a weakly nonlinear theory for the filament dynamics, these authors found that the transition to whirling was a Hopf bifurcation and the amplitude of this whirling state grew like the square root of the distance from onset [45]. However, a later analysis using the immersed boundary method instead found that immediately above the critical turning frequency, the filament took on a large amplitude whirling shape, where the free end of the filament buckled back past the clamped end [46]. This large amplitude deformation was subsequently seen in other simulations [47] and confirmed experimentally in [48].

To simulate this system, we consider a filament of length $L = 1$ and radius $a = 0.1$ with bending and twist moduli $A = C = 1$, initially aligned with the x-axis. The filament is clamped at the left end ($x = 0$) with the other end free. At the clamped end, the filament is rotated with angular speed ω about the tangent vector, which is implemented by setting a boundary condition, $\mathbf{e}_1(0,t) = \cos\omega t\hat{\mathbf{x}} + \sin\omega t\hat{\mathbf{y}}$. We used 200 nodes, a time step $\Delta t = 10^{-5}$, and $\sigma_0 = 1.6 \times 10^4$. The simulations are initialized using a shape that is slightly perturbed from straight, and the simulations were run until steady state (typically for a total time of 5 - 10).

As predicted by the previous theory [45], we find that for values of $\omega \leq 8.8/\zeta_r$ initial perturbations die away exponentially in time. For values of $\omega \geq 9.0/\zeta_r$, the steady state shape of the filament is a large amplitude whirling shape, where the filament buckles back on itself and rotates at a constant angular speed about the x axis (Fig. 8a,b).

We can also use this system to explore whether our algorithm is stable when the bending moduli are not equal, $A_1 \neq A_2$, in which case $A_- \neq 0$. For a ribbon-like filament, the bending modulus along the longer dimension is significantly larger than the modulus for the thin dimension. In addition, it has recently been found that there are also modest changes to the RFT drag coefficients [49]. We therefore set $A_1 = C = 1$ and varied A_2 from 0.05 to 1. Since the alterations to the RFT coefficients are small (an aspect ratio of 100 between the long-to-short ribbon dimensions leads to approximately a 17% variation in the RFT coefficients), we ignored the differences in drag coefficients in these simulations. We then determined the critical value of the turning frequency at which the straight filament became unstable. We found that the algorithm worked over the full range of A_2 values and that the critical turning frequency is determined by the average bending modulus; that is, for values of $\omega_c \simeq 8.9A_+/\zeta_r L^2$, an initially straight state filament would transition to a steady state whirling shape. Interestingly, when the difference between A_1 and A_2 was large enough, we found that rather than the instability leading to a large amplitude whirling state, that the amplitude of the deformed steady state shape depended on the distance from the onset of the instability, $\omega - \omega_c$, similar to what had been predicted by the weakly nonlinear theory for filaments with circular cross-sections [45] (Fig. 8c,d). This result suggests that the small amplitude Hopf bifurcation solution is stable when one of the bending moduli is greater than the other one (which also involves slower turning frequencies than when A_1 and A_2 are comparable), but that this solution is unstable to a secondary instability when $A_1 \sim A_2$. If this is correct, then this secondary instability must involve higher than quadratic order terms, which would explain why the original work did not capture the large amplitude whirling state [45]. More work will need to be done to investigate the full details of this instability.

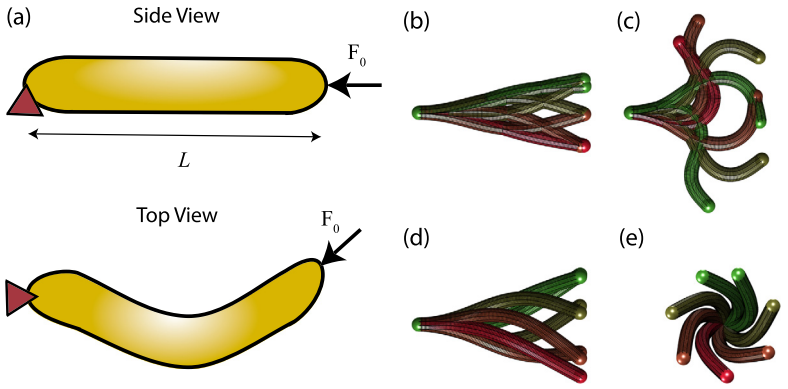

Fig. 9. (a) Schematic of the basic model for flailing as viewed from the side and the top. A thin rod of length L is anchored at the left end and a force of magnitude F_0 is applied at the other end in the direction of the tangent vector. (b-e) Results of the simulations of flailing filaments. (b) Stroboscopic montage of the flailing motions for a force $F_0 = 38$. The filament undergoes periodic, planar flailing motions. (c) Stroboscopic montage for the flailing motions when $F_0 = 200$, showing larger amplitude motions that include bends where the filament has two bends in it. A time step of $\Delta t = 10^{-7}$ was used for this simulation. Stroboscopic montage showing the side (d) and end-on (e) views of the filament for the case when $F_0 = 43.6$ and a small initial twist perturbation has been imposed. Instead of planar flailing, the filament steadily rotates with a 3-dimensional shape. (b-e) The colors are drawn such that the first and last frames of the montage are colored in green, the middle time point is shown in red, and the intermediate states are yellow.

7.7. Flailing filaments

As a final test of our method, we examine the case of a filament that is clamped at one end, while a tangential force of magnitude F is applied at the free end of the filament (Fig. 9a). This model was originally used to examine the buckling that occasionally occurs in the gliding, rod-shaped bacterium *Myxococcus xanthus* [50,51]. The application of a compressive force applied at one end of the filament causes the filament to buckle when the magnitude of the force is $F \approx 37.5A/L^2$ [51]. Beyond the buckling point, the filament *flails*, undulating back and forth with an amplitude that depends on the magnitude of the force. We selected this scenario as a test case because the original work that was done to model this system could not stably simulate the dynamics of the vector position of the cell body using a finite difference algorithm [51]. Instead, the previous work simulated the curvature dynamics for 2D motions and integrated the curvature to find the bacterial cell shape. Therefore, this scenario provides an additional check on the stability of our algorithm for a situation that has been shown to be problematic.

Here we considered a rod-shaped cell of length $L = 1$ with bending and twist moduli $A = C = 1$. The rod is clamped at the left end with $\mathbf{r}(s = 0) = 0$ and $\mathbf{e}_3(s = 0) = \hat{\mathbf{x}}$. The filamentary cell body is initially aligned primarily along the x axis, with a small perturbation added so that the filament is not exactly straight,

$$\mathbf{r}(s, t = 0) = s\hat{\mathbf{x}} + 0.02\mathcal{W}_k(s)\,\hat{\mathbf{z}}\,, \tag{60}$$

with

$$\mathcal{W}_k(s) = \left(\sin ks - \sinh ks + \frac{(\cos k + \cosh k)(\cos ks - \cosh ks)}{\sin k + \sinh k}\right)\,, \tag{61}$$

the kth mode solution to the biharmonic equation with a clamped boundary at $s = 0$ and free boundary at $s = 1$. We perturb the first order mode, $k = 1.875$, in our simulations. A compressional force $\mathbf{F} = F_0\mathbf{e}_3$ is applied at $s = L$, with the other components of the force and torque equal to zero. We used 128 nodes, a time step $\Delta t = 10^{-6}$ or 10^{-7} (for the largest values of F_0), and $\sigma_0 = 2 \times 10^4$.

Consistent with the previous work, we find that for values of $F_0L^2/A \leq 37$ that the initial perturbation dies away, whereas for values of $F_0L^2/A \geq 38$ the initial perturbation grows and the filament flails (Fig. 9b, c). The shape and dynamics of the flailing filament are comparable to what was previously observed.

Our simulations are fully three dimensional. When we start the simulations with a planar perturbation and zero twist, the dynamics maintains the planar motions that were seen with the 2D calculations reported in [51]. However, in 3D it is possible that the tangentially applied force at the filament end can also produce purely rotational motions that include twisting of the filament. To examine whether there are regions of phase space where the tangentially applied force can produce these types of motions, we start the filament with the same initial perturbation to the backbone shape, but also include a small perturbation in the twist of the filament, Ω_3. We find that for a range of forces $40 \lesssim F_0 \lesssim 100$ that there is a stably rotating solution that is also possible. Indeed, as shown in Fig. 9d, e when a force $F_0 = 43.6$ is used and a small initial twist is applied to the filament, the planar flailing solution (identified by the vertical lines in the trajectory shown in Fig. 9e) transition to a persistent steady rotation, reminiscent of the whirling motions described in [45].

8. Conclusions

Here we have described a new finite volume algorithm for handling the low Reynolds number dynamics of elastic filaments. The major benefit of this new method is that it handles force and torque balance between subvolumes of the filament in an exact way, thereby guaranteeing that the total force and moment that act on the filament are correct and do not include numerical artifacts of the discretization (similar to how finite volume methods preserve mass conservation in conservative systems). The force and torque balance are handled by two matrix operators, $\hat{\mathcal{O}}_1$ and $\hat{\mathcal{O}}_2$. These operators act directly on the discretization of the moment, therefore the local force and torque balance conditions are unaffected by the form and method used to discretize the moment. In addition, our method for discretizing the force and moment is not dependent on the type of dynamics (low Reynolds number or inertial) and can be implemented in other dynamic filament algorithms. As we have shown above, it is straightforward to implement our operator method for defining the forces and moments in the immersed boundary method.

As previously mentioned, simulating filament dynamics is a highly active area of current research and many different algorithms to handle these types of problems have been developed in recent years [8–16]. Many of these algorithms (e.g., [11,13–15]) use a method similar to that proposed in [52], where the forces and moments are defined at the half-grid point. In these algorithms, the elastic force is determined by defining springs that act between nodes, which likely enforce proper force balance between neighboring segments. However, the springs that define the elastic forces act as a soft constraint for relating the forces to the moments, whereas our algorithm directly defines this relationship from the Kirchoff equations. Furthermore, these other algorithms use auxiliary orthonormal frames that are not necessarily aligned with the local tangent vector, and another soft constraint is used to keep the auxiliary frames roughly aligned with the tangent. The algorithm presented here works directly with the material orthonormal triad, which not only preserves the material frame orientation explicitly, but also reduces the number of variables that need to be solved for each time step. Indeed, another benefit of our algorithm is how we handle the twist dynamics, which are given by the motion of $\mathbf{e}_1$. We track the motion of this unit vector by decomposing the dynamics into a component along each perpendicular direction, $\mathbf{e}_2$ and $\mathbf{e}_3$. We use a semi-implicit backward Euler method to time step the motion along $\mathbf{e}_2$, and then compute the rotation of this new vector into the tangent direction using a rotation matrix approach. This combined approach provides a stable method that maintains the appropriate relationship between $\mathbf{e}_1$ and the twist Ω_3.

Using the simple test case of a free-swimmer that moves by propagating a sinusoidal traveling wave, we validated our algorithm and showed that the convergence is second order in space and first order in time. The temporal convergence could likely be improved by using a Crank-Nicolson scheme in place of the backward Euler time stepping method that we have employed.

We then showed the utility of our algorithm by examining an array of test cases. These test cases explored the ability of our method to handle free and constrained filaments, helical filaments, filaments in external flow fields, and the actions of thermal forces and twisting torques, along with combinations of these scenarios. We also showed that the algorithm is stable, even when the two bending moduli are vastly different. In all cases, our algorithm agreed well with previous simulations and experiments. In addition, we were able to show two new behaviors, the small amplitude, steady-state whirling of a rod with $A_1 \gg A_2$, and the 3D rotation of a filament acted on by a tangential force applied at its end. Therefore, this algorithm is broadly useful for studying a wide range of three-dimensional, low Reynolds number filament dynamic problems. As already mentioned, our operator approach to discretizing the forces and moments makes it straightforward to apply this method in an inertial context; however, this remains for future research. In addition, here we have only examined the application of this method to linearly elastic constitutive laws. More complex relationships between the moment and shape should be easily implemented, but we have not considered those situations here.

CRediT authorship contribution statement

Paul M. Ryan: Formal analysis, Investigation, Methodology, Software, Validation, Visualization, Writing – original draft, Writing – review & editing. **Charles W. Wolgemuth:** Conceptualization, Formal analysis, Investigation, Methodology, Supervision, Validation, Visualization, 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.

Acknowledgements

This research was partially funded (PMR) by the National Institute of Health under grant T32 GM132008.

References

[1] G. Kirchoff, Über des gleichgewicht und die bewegung eines unedlich dünnen elatichen stabes, Crelles J. 56 (1859) 285–313.

[2] J.F. Marko, E.D. Siggia, Bending and twisting elasticity of dna, Macromolecules 27 (1994) 981–988.
[3] E. Lauga, T.R. Powers, The hydrodynamics of swimming microorganisms, Rep. Prog. Phys. 72 (2009) 096601.
[4] S.A. Stevens, Methods for modeling cytoskeletal and dna filaments, Phys. Biol. 11 (2014) 011001.
[5] C.K. Park, N.C. Horton, Structures, functions, and mechanisms of filament forming enzymes: a renaissance of enzyme filamentation, Biophys. Rev. 11 (2019) 927–994.
[6] S.S. Teves, S. Henikoff, Dna torsion as a feedback mediator of transcription dynamics, Nucleus 5 (2014) 211–218.
[7] C. Wolgemuth, Biomechanics of Cell Motility, vol. 7, Elsevier Inc., 2012, pp. 168–193.
[8] C. Moreau, L. Giraldi, H. Gadelha, The asymptotic coarse-graining formulation of slender-rods, bio-filaments and flagella, J. R. Soc. Interface 15 (0 2018), https://doi.org/10.1098/rsif.2018.0235.
[9] B.J. Walker, K. Ishimoto, H. Gadêlha, E.A. Gaffney, Filament mechanics in a half-space via regularised stokeslet segments, J. Fluid Mech. 879 (2019) 808–833, https://doi.org/10.1017/jfm.2019.723.
[10] S.F. Schoeller, A.K. Townsend, T.A. Westwood, E.E. Keaveny, Methods for suspensions of passive and active filaments, J. Comput. Phys. 424 (2021) 109846, https://doi.org/10.1016/j.jcp.2020.109846.
[11] M. Jabbarzadeh, H. Fu, A numerical method for inextensible elastic filaments in viscous fluids, J. Comput. Phys. 418 (2020) 109643, https://doi.org/10.1016/j.jcp.2020.109643.
[12] H. Guo, Y. Man, K. Wan, E. Kanso, Intracellular coupling modulates biflagellar synchrony, J. R. Soc. Interface 18 (2021) 20200660.
[13] M. Gazzola, L. Dudte, A. McCormick, L. Mahadevan, Forwardand inverse problems in the mechanics of soft filaments, R. Soc. Open Sci. 5 (2018) 171628, https://doi.org/10.1098/rsos.1716.
[14] S.D. Olson, S. Lim, R. Cortez, Modeling the dynamics of an elastic rod with intrinsic curvature and twist using a regularized Stokes formulation, J. Comput. Phys. 238 (2013) 169–187, https://doi.org/10.1016/j.jcp.2012.12.026.
[15] S. Lim, Dynamics of an open elastic rod with intrinsic curvature and twist in a viscous fluid, Phys. Fluids 22 (2) (2010) 024104, https://doi.org/10.1063/1.3326075.
[16] A.-K. Tornberg, M.J. Shelley, Simulating the dynamics and interactions of flexible fibers in Stokes flows, J. Comput. Phys. 196 (1) (2004) 8–40, https://doi.org/10.1016/j.jcp.2003.10.017.
[17] H. Gadêlha, P. Hernández-Herrera, F. Monotoya, A. Darszon, G. Corkidi, Human sperm uses asymmetric and anisotropic flagellar controls to regulate swimming symmetry and cell steering, Sci. Adv. 6 (2020) eaba5168.
[18] F.-B. Tian, L. Wang, Numerical modeling of sperm swimming, Fluids 6 (2021) 73.
[19] T. Ishikawa, Axoneme structure from motile cilia, Cold Spring Harb. Perspect. Biol. 9 (2017), a028076.
[20] D.K. Vig, C.W. Wolgemuth, Swimming dynamics of the Lyme disease bacterium, Phys. Rev. Lett. 109 (2012) 218104.
[21] W. Kan, C.W. Wolgemuth, The shape and dynamics of the *leptospiraceae*, Biophys. J. 93 (2007) 54–61.
[22] M. Deng, L. Grinberg, B. Caswell, G.E. Karniadakis, Effects of thermal noise on the transitional dynamics of an inextensible elastic filament in stagnation flow, Soft Matter 11 (2015) 4962.
[23] K. Liu, B. Chakrabarti, D. Saintillan, A. Lindner, O. du Roure, Morphological transitions of elastic filaments in shear flow, Proc. Natl. Acad. Sci. USA 115 (2018) 9438–9443.
[24] K. Liu, J. Lowengrub, J. Allard, Efficient simulation of thermally fluctuating biopolymers immersed in fluids on 1-micron, 1-second scales, J. Comput. Phys. 386 (2019) 248–263.
[25] A. Goriely, M. Tabor, The nonlinear dynamics of filaments, Nonlinear Dyn. 21 (2000) 101–133.
[26] B.D. Coleman, E.H. Dill, M. Lembo, Z. Lu, I. Tobias, On the dynamics of rods in the theory of Kirchoff and Clebsch, Arch. Ration. Mech. Anal. 121 (1993) 339–359.
[27] E.M. Strawbridge, C.W. Wolgemuth, Surface traction and the dynamics of elastic rods at low Reynolds number, Phys. Rev. E 86 (2012) 031904.
[28] R.E. Johnson, C.J. Brokaw, Flagellar hydrodynamics. a comparison between resistive-force theory and slender-body theory, Biophys. J. 25 (1979) 113–127.
[29] O. Kolditz, Computational Methods in Environmental Fluid Mechanics, 1st edition, Springer, Berlin, Heidelberg, 2002.
[30] L.D. Landau, E.M. Lifshitz, Theory of Elasticity, vol. 7, 3rd ed., Course of Theoretical Physics, Butterworth-Heinemann, Oxford, 1986.
[31] G.I. Taylor, The action of waving cylindrical tails in propelling microscopic organisms, Proc. R. Soc. A 211 (1952) 225–239.
[32] L.J. Fauci, C.S. Peskin, A computational model of aquatic animal locomotion, J. Comput. Phys. 77 (1988) 85–108.
[33] M.W. Harman, M.J. Dunham-Ems, S.M Caimano, A.A. Belperron, L.K. Bockenstedt, H.C. Fu, J.D. Radolf, C.W. Wolgemuth, The heterogeneous motility of the Lyme disease spirochete in gelatin mimics dissemination through tissue, Proc. Natl. Acad. Sci. USA 109 (2012) 3059–3064.
[34] C.S. Peskin, The immersed boundary method, Acta Numer. 11 (2002) 479–517, https://doi.org/10.1017/S0962492902000077.
[35] G. Sun, T. Goodwill, C. Peskin, Immersed boundary ib-matlab, https://github.com/ModelingSimulation/IB-MATLAB, 2019.
[36] D. Oriola, H. Gadelha, J. Casademunt, Nonlinear amplitude dynamics in flagellar beating, R. Soc. Open Sci. 4 (2017) 160698, https://doi.org/10.1098/rsos.160698.
[37] B. Chakrabarti, D. Saintillan, Spontaneous oscillations, beating patterns, and hydrodynamics of active microfilaments, Phys. Rev. Fluids 4 (2019) 043102, https://doi.org/10.1103/PhysRevFluids.4.043102.
[38] J.T. Pham, A. Morozov, A.J. Crosby, A. Lindner, O. du Roure, Deformation and shape of flexible, microscale helices in viscous flow, Phys. Rev. E 92 (2015) 011004, https://doi.org/10.1103/PhysRevE.92.011004.
[39] M. Kim, T.R. Powers, Deformation of a helical filament by flow and electric or magnetic fields, Phys. Rev. E 71 (2005) 021914, https://doi.org/10.1103/PhysRevE.71.021914.
[40] Y.-N. Young, M.J. Shelley, Stretch-coil transition and transport of fibers in cellular flows, Phys. Rev. Lett. 99 (2007) 058303, https://doi.org/10.1103/PhysRevLett.99.058303.
[41] V. Kantsler, R.E. Goldstein, Fluctuations, dynamics, and the stretch-coil transition of single actin filaments in extensional flows, Phys. Rev. Lett. 108 (2012) 038103, https://doi.org/10.1103/PhysRevLett.108.038103.
[42] R. Chelakkot, R.G. Winkler, G. Gompper, Flow-induced helical coiling of semiflexible polymers in structured microchannels, Phys. Rev. Lett. 109 (2012) 178101, https://doi.org/10.1103/PhysRevLett.109.178101.
[43] B. Chakrabarti, Y. Liu, J. LaGrone, R. Cortez, L. Fauci, O. du Roure, D. Saintillan, A. Lindner, Flexible filaments buckle into helicoidal shapes in strong compressional flows, Nat. Phys. 16 (2020) 689, https://doi.org/10.1038/s41567-020-0843-7.
[44] H. Manikantan, D. Saintillan, Buckling transition of a semiflexible filament in extensional flow, Phys. Rev. E 92 (2015) 041002, https://doi.org/10.1103/PhysRevE.92.041002.
[45] C.W. Wolgemuth, T. Powers, R. Goldstein, Twirling and whirling: viscous dynamics of rotating elastic filaments, SIAM J. Sci. Comput. 25 (2004) 2066–2083.
[46] S. Lim, C.S. Peskin, Simulations of the whirling instability by the immersed boundary method, J. Bateriol. 177 (1995) 5846–5852.
[47] H. Wada, R.R. Netz, Nonequilibrium hydodynamics of a rotating filament, Europhys. Lett. 75 (2007) 645–651.
[48] I.R. Bruss, K. Mutha, H.K Stoll, V. Collins, B. Nguyen, D.J.D. Carter, M.P. Brenner, K.J. Russell, Twirling, whirling, and tensioning: plectoneme formation and suppression in flexible filaments, Phys. Rev. Res. 1 (2019) 032020.
[49] L. Koens, E. Lauga, Analytical solutions to slender-ribbon theory, Phys. Rev. Fluids 2 (2017) 084101.

[50] A.M. Spormann, A.D. Kaiser, Gliding movements in *Myxococcus xanthus*, J. Bateriol. 177 (1995) 5846–5852.
[51] C.W. Wolgemuth, Force and flexibility of flailing myxobacteria, Biophys. J. 89 (2005) 945–950.
[52] A. Lim, S. Ferent, X.S. Wang, C.S. Peskin, Dynamics of a closed rod with twist and bend in fluid, SIAM J. Sci. Comput. 31 (2008) 273.