Methods for suspensions of passive and active filaments

Simon F Schoeller*, Adam K Townsend*, Timothy A Westwood*, Eric E Keaveny†

Department of Mathematics, Imperial College London, London SW7 2AZ, UK

11 February 2020

arXiv:1903.12609v2 [physics.comp-ph] 8 Feb 2020

Abstract

Flexible filaments and fibres are essential components of important complex fluids that appear in many biological and industrial settings. Direct simulations of these systems that capture the motion and deformation of many immersed filaments in suspension remain a formidable computational challenge due to the complex, coupled fluid–structure interactions of all filaments, the numerical stiffness associated with filament bending, and the various constraints that must be maintained as the filaments deform. In this paper, we address these challenges by first describing filament kinematics using quaternions to resolve both bending and twisting, applying implicit time-integration to alleviate numerical stiffness, and using quasi-Newton methods to obtain solutions to the resulting system of nonlinear equations. In particular, we employ geometric time integration to ensure that the quaternions remain unit as the filaments move. We also show that our framework can be used with a variety of models and methods, including matrix-free fast methods, that resolve low Reynolds number hydrodynamic interactions. We provide a series of tests and example simulations to demonstrate the performance and possible applications of our method. Finally, we provide a link to a MATLAB/Octave implementation of our framework that can be used to learn more about our approach and as a tool for filament simulation.

Contents

1 Introduction **2**

2 Filament model **4**
- 2.1 Kinematics . . . 4
- 2.2 Constitutive law and internal moments . . . 5
- 2.3 Kinematic constraint and internal stress . . . 5
- 2.4 Force and moment balances . . . 5
- 2.5 Discrete force and moment balances . . . 6
- 2.6 Steric interactions . . . 7
- 2.7 Hydrodynamics interactions and segment motion . . . 8
  - 2.7.1 RPY mobility matrix . . . 9
  - 2.7.2 Force-coupling method . . . 9
- 2.8 Planar filament deformation and motion . . . 10
  - 2.8.1 Computing the internal moments from the angles . . . 10
  - 2.8.2 Differential-algebraic system and time integration in two dimensions . . . 10
- 2.9 Three-dimensional filament deformation and motion . . . 12
  - 2.9.1 Representing rotations as quaternions . . . 12
  - 2.9.2 Computing the internal moments from the quaternions . . . 13
  - 2.9.3 Differential-algebraic system and time integration in three dimensions . . . 13

*Contributed equally.

†Email addresses: simon.schoeller14@imperial.ac.uk, adam.townsend@imperial.ac.uk, t.westwood16@imperial.ac.uk, e.keaveny@imperial.ac.uk

3 Numerical solvers for the nonlinear systems **15**
3.1 Jacobian-free Newton Krylov (JFNK) . . . 15
3.2 Broyden's method . . . 16
3.2.1 Jacobian-free Broyden's method (JFB) . . . 17
3.2.2 A block diagonal, explicit $\boldsymbol{J}_0$ (EJB) . . . 17
3.3 A comparison of the nonlinear system solvers . . . 17
3.3.1 JFNK versus Broyden's method (JFNK v EJB-b) . . . 18
3.3.2 Choice of Broyden's method (EJB-b v JFB-g) . . . 18
3.3.3 Effects of tolerance and timestep size on performance . . . 20
3.3.4 Effects of segment and filament number on performance . . . 20

4 Accuracy and convergence **21**
4.1 Convergence in space . . . 21
4.2 Convergence in time . . . 22
4.3 Accuracy of the hydrodynamic model . . . 24

5 Simulations **24**
5.1 Implementations . . . 24
5.2 Tethered filaments . . . 25
5.2.1 Rotational dynamics of a single tethered filament . . . 25
5.2.2 Coordination in cilia arrays . . . 27
5.3 Bidisperse suspension of undulatory swimmers . . . 29
5.4 Sedimentation . . . 31
5.4.1 Settling filament . . . 31
5.4.2 Sedimentation of large 2D suspensions . . . 32
5.4.3 Tumbling, settling and buckling of a filament square . . . 34
5.4.4 Sedimentation of flexible filament clouds . . . 35

6 Discussion and conclusions **37**

A Quaternions **37**
A.1 Quaternion initialisation . . . 37
A.2 Internal moments in terms of quaternions . . . 38
A.3 The quaternion ODE . . . 39
A.4 BDF2 in the Lie algebra . . . 39
A.5 Filament initialisation in cloud simulations . . . 40

B Form of the approximate Jacobian **41**
B.1 In 2D . . . 41
B.2 In 3D using Lie algebra elements . . . 42

1 Introduction

Microscopic flexible filaments and rigid fibres suspended in fluid arise in many industrial and biological applications, ranging from the wood pulp fibres found in paper production (Derakhshandeh *et al.*, 2011; Stockie & Green, 1998; Ross & Klingenberg, 1997; Pettersson *et al.*, 2017; du Roure *et al.*, 2017) to the flagella and cilia used by cells to swim, pump, or mix fluid (Lauga & Powers, 2009; Elgeti *et al.*, 2015; Brennen & Winet, 1977; Supatto & Vermot, 2011; Smith *et al.*, 2019; Faubel *et al.*, 2016; Elgeti & Gompper, 2013). In suspension, such as in polymeric fluids, their presence alters the bulk rheological properties of the medium leading to non-Newtonian responses such as shear thinning or viscoelasticity (Larson, 1999). When entangled or connected in networks, filaments and fibres form gels and disordered solids as is the case in important biological materials such as mucus (Hwang *et al.*, 1969; Sheehan & Carlstedt, 1984; Quraishi *et al.*, 1998; Lai *et al.*, 2009) and the extra-cellular matrix (Baker *et al.*, 2009;

Heck *et al.*, 2017). Filaments can also be active biological elements that facilitate transport through self-deformation (Lauga & Powers, 2009; Brennen & Winet, 1977), or through growth and the motion of motor proteins along their lengths (Shelley, 2016).

Simulation of filament or fibre motion requires solving a complex low Reynolds number fluid–structure interaction problem where the motion and deformation of all filaments are coupled through the surrounding fluid. As a result, computations involving filaments use many existing models and methods developed over the years to resolve the hydrodynamic aspects of the problem at various levels of approximation, ranging from drag-based resistive force theory (Moreau *et al.*, 2018; Yamamoto & Matsuoka, 1995) and point and regularised singularity methods (Cortez, 2001; Cortez *et al.*, 2005; Cosentino Lagomarsino *et al.*, 2005; Delmotte *et al.*, 2015; Olson *et al.*, 2013; Smith, 2009; Smith *et al.*, 2007) for single filament problems, to the immersed boundary method (Peskin, 2002; Fauci & Peskin, 1988; Stockie & Green, 1998; Lim *et al.*, 2008; Lim, 2010; Wiens & Stockie, 2015) and discretisations of the integral equations given by slender-body theory coupled with fast-summation techniques (Saintillan *et al.*, 2005; Tornberg & Shelley, 2004; Gustavsson & Tornberg, 2009; Nazockdast *et al.*, 2017; Tornberg & Gustavsson, 2006) for larger collections of flexible filaments or rigid fibres.

While there are a variety of models and methods available to treat filament hydrodynamics, a similar catalogue of methods and algorithms for filament elasticity is less developed. Due to the higher-order derivatives associated with bending and twisting, the equations of filament motion are numerically stiff. In addition, conditions such as inextensibility give rise to constraints that must be satisfied as filaments move and deform. Despite this, explicit time integration is often used (Peskin, 2002; Li *et al.*, 2013; Olson *et al.*, 2013; Simons *et al.*, 2015) to advance the filaments in time, while constraints of inextensibility are mimicked through stiff springs (Chelakkot *et al.*, 2010; Stockie & Green, 1998), both of which tend to limit stable timestep sizes. There have been recent efforts to use implicit time integration schemes (Moreau *et al.*, 2018; Hall-McNair *et al.*, 2019), as well as treat the constraints directly using Lagrange multipliers (Delmotte *et al.*, 2015) or through a suitable choice of the filament degrees of freedom (Moreau *et al.*, 2018; Hall-McNair *et al.*, 2019). These studies, however, are limited in the number of filaments that they can treat as they rely on explicit knowledge of the filament mobility matrix and require dense linear algebra to solve for the unknown Lagrange multipliers, or to implicitly integrate the equations of motion. For specific hydrodynamic approaches intended for larger-scale simulation of many interacting filaments, similar techniques have been used to reduce computational times. For example, predictor–corrector schemes have been successfully used with the immersed boundary method (Wiens & Stockie, 2015), while recent work (Nazockdast *et al.*, 2017), building from Tornberg & Shelley (2004), presents a complete approach utilising Lagrange multipliers and implicit time integration along with matrix-free hydrodynamics using the fast multipole method.

In this study, we provide a comprehensive, computationally-scalable methodology for simulating filament dynamics that is not dependent on the specific method or model used to describe the hydrodynamic forces experienced by the filaments. Our approach only assumes that hydrodynamics forces and torques experienced by the discretised beam segments are linearly related to their translational and rotational velocities, consistent with low Reynolds number hydrodynamics. The methods that we present in this paper expand on those we have already used for planar simulations of thousands of interacting sperm cells (Schoeller & Keaveny, 2018) and the motion of undulatory swimmers through structured (Majmudar *et al.*, 2012) and unstructured (Kamal & Keaveny, 2018) complex environments. This paper presents the fully three-dimensional version of the methodology that takes advantage of unit quaternions to describe filament kinematics and geometric time integration schemes (Iserles *et al.*, 2000; Faltinsen *et al.*, 2001; Park & Chung, 2005) to advance the unit quaternions in time.

Beginning from Kirchhoff rod theory (Kirchhoff, 1859; Lim *et al.*, 2008), we provide a description of the filament model and highlight the differences between deformations restricted to a plane and those that are fully three-dimensional. In particular, for fully three-dimensional deformations, we show how both filament bending and twisting can be captured through the use of unit quaternions to describe how the filament's material frame varies with arclength. We employ Lagrange multipliers to obtain the constraint forces and moments necessary to link the material frame with the positions of points along the filament, and implicit time integration to handle numerical stiffness. By comparing Jacobian-free Newton–Krylov (Knoll & Keyes, 2004) and Broyden's methods (Broyden, 1965; Kvaalen,

1991), we show how the resulting system of nonlinear equations, whose solution provides the updated generalised positions and Lagrange multipliers, can be solved most effectively using Broyden's method with a suitable choice of approximate Jacobian based on a diagonal mobility matrix. Also novel to our approach is our application of geometric time integration to ensure the quaternions remain unit. We find these schemes help to guarantee robustness of the iterative solver for the nonlinear system of equations. We provide a number of tests of our method and explore several examples using both a hydrodynamic approach based on direct summation of the Rotne–Prager–Yamakawa (RPY) tensor (Wajnryb *et al.*, 2013) and matrix-free computations based on the force-coupling method (FCM) (Maxey & Patel, 2001; Lomholt & Maxey, 2003). In doing so, we show that our method provides an effective tool for studying a variety of applications involving filament dynamics including tethered filaments and cilia arrays, interacting undulatory swimmers, and sedimenting suspensions and clouds of filaments. We provide a basic MATLAB/Octave implementation (Schoeller *et al.*, 2019) of the method that can be used to reproduce our numerical results, applied to other problems of interest, or altered to couple with the user's preferred hydrodynamic solver.

2 Filament model

We begin by describing the filament model for a single filament. In this work, a filament refers to a slender (i.e. high-aspect ratio) body, whose shape is given by a curve in space and whose deformations are related to the rotations of a right-handed frame defined along the length of the curve. The filament has circular cross-section with radius a and length L. Its physical size is assumed to be sufficiently small such that its inertia, and that of the surrounding fluid, can be ignored. It can be subject to external forces and moments along its length, such as those due to the surrounding fluid, in addition to the internal moments due to bending and twist, and the internal force that arises due to constraints on filament motion. In addition, we consider the important case where, in the absence of external forces and torques, the filament may relax to a curve with nonzero curvature and twist.

2.1 Kinematics

The positions of the points along the filament's centreline at time t are given by $\boldsymbol{Y}(s,t)$, where $s \in [0, L]$ is the filament arclength. Along with the position, there is a right-handed, orthonormal material frame $\{\widehat{\boldsymbol{t}}(s,t), \widehat{\boldsymbol{\mu}}(s,t), \widehat{\boldsymbol{\nu}}(s,t)\}$ at each s and t, see Fig. 1a. At this stage, the unit vector $\widehat{\boldsymbol{t}}$ and position, $\boldsymbol{Y}$, are independent of each other, but later we will introduce the kinematic constraint, $\partial \boldsymbol{Y}/\partial s = \widehat{\boldsymbol{t}}$, that links these quantities and will lead to forces within the filament.

Filament deformation is determined by how the material frame varies with s. Following Landau & Lifshitz (1986, §18); Powers (2010), this deformation can be described by the vector $\boldsymbol{\Phi}(s,t)$ that describes the spatial-rate of rotation of the local frame with s such that $\partial \boldsymbol{v}/\partial s = \boldsymbol{\Phi} \times \boldsymbol{v}$ for any vector in the material frame, $\boldsymbol{v}$. Using $\widehat{\boldsymbol{t}}$ and $\partial \widehat{\boldsymbol{t}}/\partial s$, $\boldsymbol{\Phi}$ can be expressed as

$$\boldsymbol{\Phi} = \widehat{\boldsymbol{t}} \times \frac{\partial \widehat{\boldsymbol{t}}}{\partial s} + \widehat{\boldsymbol{t}} \left(\boldsymbol{\Phi} \cdot \widehat{\boldsymbol{t}} \right), \tag{1}$$

where the first term is related to bending based on a curve with tangent $\widehat{\boldsymbol{t}}$, while the second term describes twisting about $\widehat{\boldsymbol{t}}$. In terms of the frame vectors, we can write the bending contribution as

$$\widehat{\boldsymbol{t}} \times \frac{\partial \widehat{\boldsymbol{t}}}{\partial s} = \widehat{\boldsymbol{\mu}} \left(\widehat{\boldsymbol{t}} \cdot \frac{\partial \widehat{\boldsymbol{\nu}}}{\partial s} \right) + \widehat{\boldsymbol{\nu}} \left(\widehat{\boldsymbol{\mu}} \cdot \frac{\partial \widehat{\boldsymbol{t}}}{\partial s} \right), \tag{2}$$

while the twist contribution is given by

$$\boldsymbol{\Phi} \cdot \widehat{\boldsymbol{t}} = \widehat{\boldsymbol{\nu}} \cdot \frac{\partial \widehat{\boldsymbol{\mu}}}{\partial s}. \tag{3}$$

2.2 Constitutive law and internal moments

The internal moments, $\boldsymbol{M}$, are linearly related to filament bending, Eq. (2), and twisting, Eq. (3), through the bending, K_B, and twist, K_T, moduli. These moduli are related to the shear modulus, G, and the Young's modulus, E, of the underlying filament material via $K_B = EI_x$, $K_T = GI_z$, where $I_x = \pi a^4/4$ and $I_z = 2I_x$ are the planar and polar moments of inertia for the circular filament cross-section, respectively (Landau & Lifshitz, 1986, §16, 17). Specifically, we have (Landau & Lifshitz, 1986, §18)

$$\begin{aligned}\boldsymbol{M}(s,t) =& K_B\left(\left(\widehat{\boldsymbol{t}}\cdot\frac{\partial\widehat{\boldsymbol{\nu}}}{\partial s}\right)\widehat{\boldsymbol{\mu}}+\left(\widehat{\boldsymbol{\mu}}\cdot\frac{\partial\widehat{\boldsymbol{t}}}{\partial s}\right)\widehat{\boldsymbol{\nu}}\right)\\ &+K_T\left(\widehat{\boldsymbol{\nu}}\cdot\frac{\partial\widehat{\boldsymbol{\mu}}}{\partial s}\right)\widehat{\boldsymbol{t}}.\end{aligned} \tag{4}$$

With this form of $\boldsymbol{M}$, the filament equilibrium configuration is straight and untwisted. Following Lim *et al.* (2008); Lim (2010); Olson *et al.* (2013), to incorporate a nontrivial equilibrium shape into the model, we can introduce the preferred curvatures, $\kappa_\mu(s,t)$ and $\kappa_\nu(s,t)$, and preferred twist, $\gamma_0(s,t)$, such that

$$\begin{aligned}\boldsymbol{M}(s,t) =& K_B\left(\widehat{\boldsymbol{\mu}}\left(\widehat{\boldsymbol{t}}\cdot\frac{\partial\widehat{\boldsymbol{\nu}}}{\partial s}-\kappa_\mu(s,t)\right)+\widehat{\boldsymbol{\nu}}\left(\widehat{\boldsymbol{\mu}}\cdot\frac{\partial\widehat{\boldsymbol{t}}}{\partial s}-\kappa_\nu(s,t)\right)\right)\\ &+K_T\widehat{\boldsymbol{t}}\left(\widehat{\boldsymbol{\nu}}\cdot\frac{\partial\widehat{\boldsymbol{\mu}}}{\partial s}-\gamma_0(s,t)\right).\end{aligned} \tag{5}$$

In the absence of applied forces and torques acting on the filament, its shape is determined completely by κ_μ, κ_ν and γ_0. Additionally, by allowing κ_μ, κ_ν and γ_0 to be time dependent, we can introduce shape changes that allow the filament to propel itself in a surrounding fluid without experiencing a net force or net torque, in accordance with the constraints on swimming at low Reynolds number (Purcell, 1977; Lauga & Powers, 2009).

2.3 Kinematic constraint and internal stress

Along with the internal moments, filament deformation will give rise to an internal force, $\boldsymbol{\Lambda}(s,t)$, or the force on the filament cross-section. In Lim *et al.* (2008); Lim (2010); Olson *et al.* (2013), this force is captured through a constitutive law that relates $\partial\boldsymbol{Y}/\partial s$ and the local frame such that in equilibrium,

$$\frac{\partial\boldsymbol{Y}}{\partial s}=\widehat{\boldsymbol{t}}. \tag{6}$$

We instead impose Eq. (6) as a kinematic constraint, requiring that $\widehat{\boldsymbol{t}}$ be the unit tangent to the curve defined by $\boldsymbol{Y}$. This constraint gives rise to the internal force, $\boldsymbol{\Lambda}(s,t)$ and as such, it can be viewed as the Lagrange multiplier[1] that enforces (6). The internal force provides both the tension in the direction $\widehat{\boldsymbol{t}}$ that keeps the filament inextensible, but also the necessary normal force that couples the force and moment balances as we describe below.

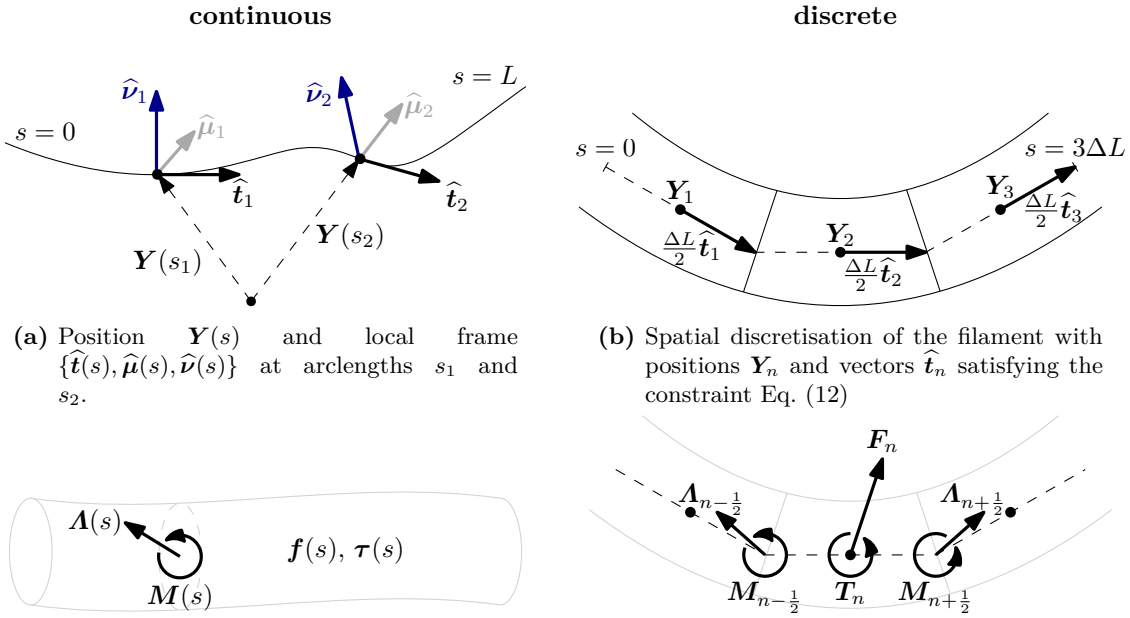
2.4 Force and moment balances

In general, the internal force and moment described above will balance external forces, $\boldsymbol{f}$, and torques, $\boldsymbol{\tau}$, per unit length that act along the filament, see Fig. 1c. The external forces and torques can arise

[1]Rather than considering force and moment balances along the filament, the filament model can alternatively be derived through the introduction of a Lagrangian. The internal force arises from the constraint term,

$$\int_0^L\boldsymbol{\Lambda}\cdot\left(\frac{\partial\boldsymbol{Y}}{\partial s}-\widehat{\boldsymbol{t}}\right)\mathrm{d}s, \tag{7}$$

in the Lagrangian, which will also contain the bending and twist energies, as well as any external potential.



(a) Position $\boldsymbol{Y}(s)$ and local frame $\{\widehat{\boldsymbol{t}}(s), \widehat{\boldsymbol{\mu}}(s), \widehat{\boldsymbol{\nu}}(s)\}$ at arclengths s_1 and s_2.

(b) Spatial discretisation of the filament with positions $\boldsymbol{Y}_n$ and vectors $\widehat{\boldsymbol{t}}_n$ satisfying the constraint Eq. (12)

(c) The internal forces $\boldsymbol{\Lambda}(s)$ and moments $\boldsymbol{M}(s)$. External force and torque densities along the filament are $\boldsymbol{f}(s)$ and $\boldsymbol{\tau}(s)$, respectively.

(d) Forces and torques in the discrete system with $\boldsymbol{F}_n = \Delta L \boldsymbol{f}_n$ and $\boldsymbol{T}_n = \Delta L \boldsymbol{\tau}_n$

Figure 1: Continuous and discrete representations of the filament model

due to, for example, an external field such as gravity or short-ranged filament–filament interactions. In particular, we insist that they include the force and torque per unit length exerted on the filament by the surrounding fluid. We assume that the interaction with the fluid is governed by low Reynolds number hydrodynamics, and hence these forces and torques are linearly related to the filament's motion. As described in Landau & Lifshitz (1986, §19); Powers (2010), the resulting force and moment balances are

$$\frac{\partial \boldsymbol{\Lambda}}{\partial s} + \boldsymbol{f} = \boldsymbol{0}, \tag{8}$$

$$\frac{\partial \boldsymbol{M}}{\partial s} + \widehat{\boldsymbol{t}} \times \boldsymbol{\Lambda} + \boldsymbol{\tau} = \boldsymbol{0}. \tag{9}$$

In addition to these equations, we will have conditions at the filament ends. For example, for filaments with free ends, we have $\boldsymbol{\Lambda}(0,t) = \boldsymbol{\Lambda}(L,t) = \boldsymbol{0}$ and $\boldsymbol{M}(0,t) = \boldsymbol{M}(L,t) = \boldsymbol{0}$. Filaments that are tethered will need to satisfy conditions imposed on $\boldsymbol{Y}$ and $\{\widehat{\boldsymbol{t}}, \widehat{\boldsymbol{\mu}}, \widehat{\boldsymbol{\nu}}\}$ at the tethered end. At this stage, it is important to note that Eqs. (8) and (9) establish the low Reynolds number mobility problem that couples the motion of all filaments through the fluid flows that they generate. This mobility problem, which we describe and solve in the discrete setting (see Section 2.7), yields equations for the translational and angular velocities of the discrete filament segments. The values of these velocities, however, will depend on $\boldsymbol{\Lambda}$ which remains unknown. The values of $\boldsymbol{\Lambda}$ are determined later by imposing the kinematic constraint, Eq. (6), when integrating the differential equations for the filament segment positions and orientations.

2.5 Discrete force and moment balances

To solve the equations numerically, the filament is first discretised into N segments of length ΔL as shown in Fig. 1b. The segments have positions, $\boldsymbol{Y}_n$, and orientations, $\widehat{\boldsymbol{t}}_n$, for $n = 1, \ldots, N$. The segment orientations are the discrete representation of the frame vector, $\widehat{\boldsymbol{t}}$, at the segment position. Applying

central differencing to Eqs. (8) and (9), the force and moment balances become

$$\frac{\boldsymbol{\Lambda}_{n+1/2} - \boldsymbol{\Lambda}_{n-1/2}}{\Delta L} + \boldsymbol{f}_n = \mathbf{0}, \tag{10}$$

$$\frac{\boldsymbol{M}_{n+1/2} - \boldsymbol{M}_{n-1/2}}{\Delta L} + \frac{1}{2}\widehat{\boldsymbol{t}}_n \times \left(\boldsymbol{\Lambda}_{n+1/2} + \boldsymbol{\Lambda}_{n-1/2}\right) + \boldsymbol{\tau}_n = \mathbf{0}, \tag{11}$$

where $\boldsymbol{M}_{n+1/2}$ and $\boldsymbol{\Lambda}_{n+1/2}$ are the internal moment and stress, respectively, between segments n and $n+1$; $\boldsymbol{f}_n$ is the external force per unit length on segment n; and $\boldsymbol{\tau}_n$ is the external torque per unit length on n; see Fig. 1d. Free end conditions are established by requiring that $\boldsymbol{M}_{1/2} = \boldsymbol{M}_{N+1/2} = \mathbf{0}$ and $\boldsymbol{\Lambda}_{1/2} = \boldsymbol{\Lambda}_{N+1/2} = \mathbf{0}$. Similarly, for the kinematic constraint, Eq. (6), we have

$$\boldsymbol{g}_{n+1/2} = \boldsymbol{Y}_{n+1} - \boldsymbol{Y}_n - \frac{\Delta L}{2}\left(\widehat{\boldsymbol{t}}_n + \widehat{\boldsymbol{t}}_{n+1}\right) = \mathbf{0}. \tag{12}$$

Multiplying Eqs. (10) and (11) by ΔL, we convert the force and moment per unit length balances to the force and torque balances such that

$$\boldsymbol{F}_n^C + \boldsymbol{F}_n = \mathbf{0}, \tag{13}$$

$$\boldsymbol{T}_n^E + \boldsymbol{T}_n^C + \boldsymbol{T}_n = \mathbf{0}, \tag{14}$$

for segment n. The elastic torque on n is given by

$$\boldsymbol{T}_n^E = \boldsymbol{M}_{n+1/2} - \boldsymbol{M}_{n-1/2}. \tag{15}$$

Since the internal moment is directly linked to how the local frame changes with arclength, the computation of $\boldsymbol{M}_{n+1/2}$ will depend on whether filament deformation is strictly planar, or fully three-dimensional. We present both cases below in Sections 2.8.1 and 2.9.2.

The constraint force and torque, respectively, on segment n, are given by

$$\boldsymbol{F}_n^C = \boldsymbol{\Lambda}_{n+1/2} - \boldsymbol{\Lambda}_{n-1/2}, \tag{16}$$

$$\boldsymbol{T}_n^C = \frac{\Delta L}{2}\widehat{\boldsymbol{t}}_n \times \left(\boldsymbol{\Lambda}_{n+1/2} + \boldsymbol{\Lambda}_{n-1/2}\right). \tag{17}$$

The values of $\boldsymbol{\Lambda}_{n+1/2}$ are the Lagrange multipliers associated with the discrete version of the kinematic constraint, Eq. (12), that relates the translational and rotational motions of segments n and $n+1$. The Lagrange multipliers are unknown and must be computed as the filament moves and deforms.

Finally, the external forces and torques acting on a segment are defined as $\boldsymbol{F}_n = \Delta L \boldsymbol{f}_n$ and $\boldsymbol{T}_n = \Delta L \boldsymbol{\tau}_n$, respectively. The external forces and torques can arise due to an external field, such as gravity or an external flow, but also through interactions with neighbouring filaments as mediated by the fluid, or through direct collisions and steric interactions. We describe here how we capture these interactions and, in particular, how hydrodynamic interactions lead to segment motion. In the case where the external forces and torques are due only to steric interactions and the surrounding fluid, we have

$$\boldsymbol{F}_n = \boldsymbol{F}_n^B - \boldsymbol{F}_n^H, \tag{18}$$

$$\boldsymbol{T}_n = -\boldsymbol{T}_n^H, \tag{19}$$

where $\boldsymbol{F}_n^B$ are the steric forces experienced by segment n and $-\boldsymbol{F}_n^H$ and $-\boldsymbol{T}_n^H$ are the hydrodynamic force and torque on segment n. We have chosen to introduce the negative sign here so that $\boldsymbol{F}_n^H$ and $\boldsymbol{T}_n^H$ denote, respectively, the force and torque segment n exerts on the fluid.

2.6 Steric interactions

In our simulations, segments from different filaments and non-neighbouring segments from the same filament repel each other via a short-ranged, pairwise barrier force. For simplicity, we take this force to be a function of the centre-to-centre separation of the segments, though a more detailed representation of the segment shape could also be taken.

Specifically, in our simulations, the total barrier force on segment n is

$$\boldsymbol{F}_n^B = \sum_m \boldsymbol{F}_{nm}^B, \tag{20}$$

where the force on segment n due to m is (Dance *et al.*, 2004)

$$\boldsymbol{F}_{nm}^B = F^S \left(\frac{4a^2\chi^2 - r_{nm}^2}{4a^2(\chi^2 - 1)} \right)^4 \frac{\boldsymbol{r}_{nm}}{2a} \quad (\text{if } r_{nm} < 2\chi a), \tag{21}$$

and $\boldsymbol{F}_{nm}^B = \boldsymbol{0}$ if $r_{nm} \geq 2\chi a$. We also have that $\boldsymbol{F}_{nm}^B = \boldsymbol{0}$ for any separation distance if n and m are neighbouring segments from the same filament. In Eq. (21), the displacement between segments is $\boldsymbol{r}_{nm} = \boldsymbol{Y}_n - \boldsymbol{Y}_m$, while $r_{nm} = \|\boldsymbol{r}_{nm}\|$. The parameter F^S controls the strength of repulsion at contact, while χ controls the range over which the barrier force acts. Recall that $2a$ is the filament thickness. In practice, to prevent the segments from overlapping, we choose the reference force F^S to be large compared to the typical bending force and set $\chi = 1.1$ to ensure that the force only acts for particle separations that are only 10% of their contact distance.

2.7 Hydrodynamics interactions and segment motion

The hydrodynamic forces and moments on the segments balance the internal stresses and moments and the inter-segment repulsive forces. In the absence of fluid inertia, this leads to a low Reynolds number mobility problem which provides a linear relationship between the forces and torques on the segments and their velocities and angular velocities,

$$\begin{pmatrix} \boldsymbol{V} \\ \boldsymbol{\Omega} \end{pmatrix} = \mathcal{M} \cdot \begin{pmatrix} \boldsymbol{F}^H \\ \boldsymbol{T}^H \end{pmatrix}, \tag{22}$$

where $\boldsymbol{V}^\top = (\boldsymbol{V}_1^\top, \ldots, \boldsymbol{V}_N^\top)$ is the vector of all segment velocity components. Similarly $\boldsymbol{\Omega}$ contains the angular velocity components for all segments, while $\boldsymbol{F}^H$ and $\boldsymbol{T}^H$ are the vectors that hold the forces and torques, respectively, that all the filament segments exert on the surrounding fluid. The methods presented in this paper can be coupled with a variety of hydrodynamic models and associated computational methodologies, including those that are mobility matrix-free, provided that they retain this linear relationship between the forces that the segments exert on the fluid and segment motion. Such approaches include Stokesian Dynamics (Brady & Bossis, 1988; Sierou & Brady, 2001), boundary integral methods (Pozrikidis, 1992), the method of regularised Stokeslets (Cortez, 2001) and the immersed boundary method applied to the Stokes equations (Peskin, 2002).

In this work, we employ two approaches to perform the mobility matrix multiplication and resolve segment motion: a direct computation based on the Rotne–Prager–Yamakawa (RPY) tensors (Wajnryb *et al.*, 2013), and a matrix-free approach based on the force-coupling method (FCM) (Maxey & Patel, 2001; Lomholt & Maxey, 2003). Both approaches provide a similar resolution of segment hydrodynamic interactions based on the Stokes flows generated by singular and regularised force and torque distributions. In addition, RPY and FCM are closely related to the methods mentioned above and are chosen to demonstrate how our filament model can be used in conjunction with a wide range of methods, models, and solvers for filament motion in viscous fluids. We note that both RPY and FCM link the radius of the filament cross-section to the segment length. Thus, increasing the number of segments changes the aspect ratio of the filament, and hence its hydrodynamic response. While we will see in Section 4.3 that this hydrodynamic model provides accurate values for the hydrodynamic force on the filament over a range of aspect ratios, it could be important to be able to increase (or decrease) the spatial resolution of filament deformation without affecting hydrodynamic response. This decoupling is achieved, for example, with slender body theory (Tornberg & Shelley, 2004), as well as through a modified version of the method of regularised Stokeslets (Cortez, 2018). With this drawback in mind, we designed the overarching computational framework to be flexible and readily accommodate changes to the hydrodynamic model provided it satisfies Eq. (22).

2.7.1 RPY mobility matrix

In one approach, we perform a direct pairwise evaluation of the RPY tensor (Wajnryb *et al.*, 2013) to compute the mobility matrix–vector multiplication. The operation count for this computation scales like the number of segments squared, though more sophisticated, faster techniques (Liang *et al.*, 2013) could also be used. With RPY, and given a collection of N non-overlapping particles of radius a and separations of $\boldsymbol{r}_{nm} = \boldsymbol{Y}_n - \boldsymbol{Y}_m$, $r_{nm} = \|\boldsymbol{r}_{nm}\| > 2a$, in a fluid with viscosity η, the velocities of particle n are given by

$$\begin{pmatrix} \boldsymbol{V}_n \\ \boldsymbol{\Omega}_n \end{pmatrix} = \sum_{m=1}^{N} \begin{pmatrix} \boldsymbol{M}_{nm}^{tt} & \boldsymbol{M}_{nm}^{tr} \\ \boldsymbol{M}_{nm}^{rt} & \boldsymbol{M}_{nm}^{rr} \end{pmatrix} \begin{pmatrix} \boldsymbol{F}_m \\ \boldsymbol{T}_m \end{pmatrix}. \tag{23}$$

For an unbounded fluid, the sub-tensors are defined as

$$\boldsymbol{M}_{nm}^{tt} = \begin{cases} \frac{1}{6\pi\eta a}\boldsymbol{I} & n = m, \\ \frac{1}{8\pi\eta r_{nm}}\left(\left(1 + \frac{2a^2}{3r_{nm}^2}\right)\boldsymbol{I} + \left(1 - \frac{2a^2}{r_{nm}^2}\right)\widehat{\boldsymbol{r}}_{nm}\widehat{\boldsymbol{r}}_{nm}\right) & n \neq m, \end{cases} \tag{24}$$

$$\boldsymbol{M}_{nm}^{rr} = \begin{cases} \frac{1}{8\pi\eta a^3}\boldsymbol{I} & n = m, \\ \frac{1}{16\pi\eta r_{nm}^3}\left(3\widehat{\boldsymbol{r}}_{ij}\widehat{\boldsymbol{r}}_{nm} - \boldsymbol{I}\right) & n \neq m, \end{cases} \tag{25}$$

$$\boldsymbol{M}_{nm}^{tr} = \boldsymbol{M}_{nm}^{rt} = \begin{cases} \boldsymbol{0} & n = m, \\ \frac{1}{8\pi\eta r_{nm}^2}\boldsymbol{\varepsilon}\cdot\widehat{\boldsymbol{r}}_{nm} & n \neq m, \end{cases} \tag{26}$$

where $\widehat{\boldsymbol{r}}_{nm} = \boldsymbol{r}_{nm}/r_{nm}$, $\boldsymbol{I}$ is the identity matrix, and $\boldsymbol{\varepsilon}$ is the three-dimensional Levi-Civita symbol. We note that there are extensions and variants of these tensors which allow for particles of different sizes, overlapping particles, the inclusion of background shear flows, and the presence of no-slip boundaries (Wajnryb *et al.*, 2013; Zuk *et al.*, 2014; Swan & Brady, 2007).

2.7.2 Force-coupling method

We also present computations where the mobility matrix–vector multiplication is performed using the matrix-free FCM (Maxey & Patel, 2001; Lomholt & Maxey, 2003; Liu *et al.*, 2009). With FCM, the forces and torques the segments exert on the fluid are transferred to the fluid through a truncated and regularised force multipole expansion in the Stokes equations,

$$-\boldsymbol{\nabla}p + \eta\nabla^2\boldsymbol{u} + \sum_n \boldsymbol{F}_n^H \Delta_n(\boldsymbol{x}) - \frac{1}{2}\sum_n \boldsymbol{T}_n^H \times \boldsymbol{\nabla}\Theta_n(\boldsymbol{x}) = \boldsymbol{0}, \tag{27}$$

$$\boldsymbol{\nabla}\cdot\boldsymbol{u} = 0, \tag{28}$$

where the sums are over all segments, η is the fluid viscosity, $\boldsymbol{u}$ is the fluid velocity and p is the pressure. The Dirac delta functions in the multipole expansions are replaced by the Gaussians,

$$\Delta_n(\boldsymbol{x}) = \left(2\pi\sigma_\Delta^2\right)^{-3/2}\exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{Y}_n\|^2}{2\sigma_\Delta^2}\right), \tag{29}$$

$$\Theta_n(\boldsymbol{x}) = \left(2\pi\sigma_\Theta^2\right)^{-3/2}\exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{Y}_n\|^2}{2\sigma_\Theta^2}\right). \tag{30}$$

Taking advantage of the ratios established for spherical particles (Maxey & Patel, 2001; Lomholt & Maxey, 2003), we take the Gaussian envelope sizes to be related to the filament thickness through $\sigma_\Delta = a/\sqrt{\pi}$ and $\sigma_\Theta = a/(6\sqrt{\pi})^{1/3}$. In Eq. (27), we have ignored the contribution of the stresslets that would enforce a vanishing rate of strain within the segments (Lomholt & Maxey, 2003). The stresslets can also be included at the cost of several conjugate gradient iterations (Yeo & Maxey, 2010) each time the segment velocities and angular velocities are determined.

After the fluid velocity is found, it is spatially averaged using the same Gaussian functions to obtain the velocity $\boldsymbol{V}_n$ and angular velocity $\boldsymbol{\Omega}_n$ of each segment n. Specifically, we have that

$$\boldsymbol{V}_n = \int \boldsymbol{u}\, \Delta_n(\boldsymbol{x})\, \mathrm{d}^3\boldsymbol{x}, \tag{31}$$

$$\boldsymbol{\Omega}_n = \frac{1}{2}\int (\boldsymbol{\nabla}\times\boldsymbol{u})\, \Theta_n(\boldsymbol{x})\, \mathrm{d}^3\boldsymbol{x}\,. \tag{32}$$

Following Yeo & Maxey (2010); Keaveny (2014), in our computations, the regularised forcing is first evaluated on a regular grid. The Stokes equations subject to periodic boundary conditions are then solved using a Fourier spectral method. Finally, the trapezoidal rule is used to integrate numerically Eqs. (31) and (32) to obtain the translational and angular velocity for each segment.

2.8 Planar filament deformation and motion

In order to keep track of filament deformation and motion, as well as compute the internal moments and forces, we need to describe how the local frame, $\{\widehat{\boldsymbol{t}}(s,t), \widehat{\boldsymbol{\mu}}(s,t), \widehat{\boldsymbol{\nu}}(s,t)\}$, at each point along the filament rotates with time.

In many important problems, such as sperm locomotion through planar flagellar beats (Schoeller & Keaveny, 2018; Yang *et al.*, 2008), filament deformation and motion are restricted to a single plane and only bending occurs. Due to the absence of both elastic and preferred twist, the local frame at each point along the filament will rotate about a single, fixed direction. Without loss of generality, we can take this direction to be aligned with the fixed vector $\widehat{\boldsymbol{e}}_z$, as well as $\widehat{\boldsymbol{\nu}} = \widehat{\boldsymbol{e}}_z$ for all s and t. This direction is always normal to the plane of motion.

2.8.1 Computing the internal moments from the angles

As a result of taking the out-of-plane direction as $\widehat{\boldsymbol{\nu}} = \widehat{\boldsymbol{e}}_z$, we have that $\kappa_\mu = 0$ in Eq. (5). The internal moment is then purely the result of bending and is given by

$$\boldsymbol{M} = K_B \widehat{\boldsymbol{t}} \times \frac{\partial \widehat{\boldsymbol{t}}}{\partial s} - K_B \kappa_\nu \widehat{\boldsymbol{e}}_z, \tag{33}$$

with the understanding that since $\widehat{\boldsymbol{t}}$ lies in the same plane for all s, $\widehat{\boldsymbol{t}} \times \partial\widehat{\boldsymbol{t}}/\partial s$ will be in the direction $\widehat{\boldsymbol{e}}_z$ for all s and t. In the discretised model, the elastic moments are then given by

$$\boldsymbol{M}_{n+1/2} = K_B \left(\widehat{\boldsymbol{t}}_n \times \left(\frac{\widehat{\boldsymbol{t}}_{n+1} - \widehat{\boldsymbol{t}}_n}{\Delta L} \right) - \kappa_\nu(s_{n+1}, t)\widehat{\boldsymbol{e}}_z \right) \tag{34a}$$

$$= K_B \left(\frac{1}{\Delta L} \left(\widehat{\boldsymbol{t}}_n \times \widehat{\boldsymbol{t}}_{n+1} \right) - \kappa_\nu(s_{n+1}, t)\widehat{\boldsymbol{e}}_z \right). \tag{34b}$$

where again it is understood that $\widehat{\boldsymbol{t}}_n \times \widehat{\boldsymbol{t}}_{n+1}$ is in the direction $\widehat{\boldsymbol{e}}_z$ for all n and t. The restriction to a single plane also allows for $\widehat{\boldsymbol{t}}_n$ to be conveniently expressed in terms of an angle $\theta_n(t)$ such that $\widehat{\boldsymbol{t}}_n = (\cos\theta_n, \sin\theta_n, 0)$.

With this expression for $\boldsymbol{M}_{n+1/2}$ found, we now have expressions for all the forces and torques on the segments appearing in the discrete force and torque balances, Eqs. (13) and (14).

2.8.2 Differential-algebraic system and time integration in two dimensions

With expressions for forces $\boldsymbol{F}_n^C$ and $\boldsymbol{F}_n^B$, and torques $\boldsymbol{T}_n^C$ and $\boldsymbol{T}_n^E$, we now move on to integrating the system in time. The mobility problem in Eq. (22) expresses the velocities, $\boldsymbol{V}_n$, and angular velocities, $\boldsymbol{\Omega}_n$, for each segment n, in terms of these aforementioned forces and torques through

$$\begin{pmatrix} \boldsymbol{V} \\ \boldsymbol{\Omega} \end{pmatrix} = \mathcal{M} \cdot \begin{pmatrix} \boldsymbol{F}^C + \boldsymbol{F}^B \\ \boldsymbol{T}^C + \boldsymbol{T}^E \end{pmatrix}. \tag{35}$$

Due to the motion being restricted to the plane, we have $\boldsymbol{V}_n \cdot \widehat{\boldsymbol{e}}_z = 0$ and $\boldsymbol{\Omega}_n = (0, 0, \Omega_n)$ for each n. We emphasise that we cannot simply evaluate Eq. (35) to obtain the segment motion since the values of $\boldsymbol{\Lambda}_{n+1/2}$ appearing in $\boldsymbol{F}_n^C$ (see Eq. (16)) are unknown and are obtained by applying the kinematic constraint, Eq. (12), for each n. Thus, the segment positions and angles can be updated by integrating numerically in time the differential-algebraic system of equations,

$$\frac{\mathrm{d}\boldsymbol{Y}_n}{\mathrm{d}t} = \boldsymbol{V}_n, \tag{36}$$

$$\frac{\mathrm{d}\theta_n}{\mathrm{d}t} = \Omega_n, \tag{37}$$

$$\boldsymbol{Y}_{n+1} - \boldsymbol{Y}_n - \frac{\Delta L}{2}\left(\widehat{\boldsymbol{t}}_n + \widehat{\boldsymbol{t}}_{n+1}\right) = \boldsymbol{0}. \tag{38}$$

Due to the numerical stiffness associated with the overdamped dynamics of elastic beams (Powers, 2010; Nazockdast *et al.*, 2017), as well as the need to ensure that the constraints are satisfied after advancing in time, we discretise in time using the implicit second-order backwards differential formula (BDF) (Ascher & Petzold, 1998) and impose the constraint at the updated positions and orientations (time $j+1$), such that

$$\boldsymbol{Y}_n^{j+1} - \frac{4}{3}\boldsymbol{Y}_n^{j} + \frac{1}{3}\boldsymbol{Y}_n^{j-1} - \frac{2}{3}\Delta t\, \boldsymbol{V}_n^{j+1} = \boldsymbol{0}, \tag{39a}$$

$$\theta_n^{j+1} - \frac{4}{3}\theta_n^{j} + \frac{1}{3}\theta_n^{j-1} - \frac{2}{3}\Delta t\, \Omega_n^{j+1} = 0, \tag{39b}$$

$$\boldsymbol{Y}_{n+1}^{j+1} - \boldsymbol{Y}_n^{j+1} - \frac{\Delta L}{2}\left(\widehat{\boldsymbol{t}}_n^{j+1} + \widehat{\boldsymbol{t}}_{n+1}^{j+1}\right) = \boldsymbol{0}, \tag{39c}$$

where Δt is the timestep size and $\boldsymbol{V}_n^{j+1}$ and Ω_n^{j+1} are given by the mobility problem, Eq. (35), with the right-hand side evaluated at $j+1$ using $\boldsymbol{Y}_n^{j+1}$, θ_n^{j+1}, and $\boldsymbol{\Lambda}_{n+1/2}^{j+1}$. This yields a nonlinear system of equations for the positions, $\boldsymbol{Y}_{n+1}^{j+1}$, and angles, θ_n^{j+1}, as well as the internal force Lagrange multipliers, $\boldsymbol{\Lambda}_{n+1/2}^{j+1}$, associated with each constraint. If M is the number of filaments and N is the number of segments per filament, the position and orientation updates, Eqs. (39a) and (39b), and the constraints, Eq. (39c), constitute a system of $M(5N-2)$ nonlinear equations.

Before solving this system numerically, we first reduce the system size by substituting the constraint, Eq. (39c), into the position updates, Eq. (39a), as to reduce the number of degrees of freedom to those of a so-called robot arm whose motion is completely described by a single position and the orientations of each link comprising the arm. Specifically, in the case of a single filament, we replace $\boldsymbol{Y}_n^{j+1}$ for $n > 1$ by

$$\boldsymbol{Y}_n^{j+1} = \boldsymbol{Y}_1^{j+1} + \frac{\Delta L}{2}\sum_{m=2}^{n}\left(\widehat{\boldsymbol{t}}_{m-1}^{j+1} + \widehat{\boldsymbol{t}}_m^{j+1}\right) \tag{40}$$

to obtain the new, reduced system of equations,

$$\boldsymbol{Y}_1^{j+1} - \frac{4}{3}\boldsymbol{Y}_1^{j} + \frac{1}{3}\boldsymbol{Y}_1^{j-1} - \frac{2}{3}\Delta t\, \boldsymbol{V}_1^{j+1} = \boldsymbol{0}, \tag{41a}$$

$$\theta_n^{j+1} - \frac{4}{3}\theta_n^{j} + \frac{1}{3}\theta_n^{j-1} - \frac{2}{3}\Delta t\, \Omega_n^{j+1} = 0, \tag{41b}$$

$$\boldsymbol{Y}_1^{j+1} + \frac{\Delta L}{2}\sum_{m=2}^{n}\left(\widehat{\boldsymbol{t}}_{m-1}^{j+1} + \widehat{\boldsymbol{t}}_m^{j+1}\right) - \frac{4}{3}\boldsymbol{Y}_n^{j} + \frac{1}{3}\boldsymbol{Y}_n^{j-1} - \frac{2}{3}\Delta t\, \boldsymbol{V}_n^{j+1} = \boldsymbol{0}. \tag{41c}$$

This system may be expressed as $\boldsymbol{f}(\boldsymbol{X}^*) = \boldsymbol{0}$, where the solution $\boldsymbol{X}^*$ contains the updated position for the first segment, all orientation angles and the Lagrange multipliers. For M filaments each with N segments, as a result of this substitution, the dimension of the nonlinear system is reduced from $M(5N-2)$ to $3MN$.

2.9 Three-dimensional filament deformation and motion

In general, there is no restriction about how the filament can bend or twist, and the local frame along the filament can rotate about any axis. Rather than considering the local frame vectors explicitly, we keep track of these rotations using the unit quaternions that map the standard basis to the local frame at each point along the filament. Unit quaternions allow for successive rotations to be computed easily while requiring less storage and fewer floating point operations than rotation matrices. Quaternions also avoid gimbal lock that is typically experienced with Euler angles (Allen & Tildesley, 2017). Before describing how we employ quaternions in our methodology, we provide a brief overview of representing rotations using quaternions.

2.9.1 Representing rotations as quaternions

Quaternions (Allen & Tildesley, 2017; Dunn & Parberry, 2011; Vince, 2011) can be viewed as an extension of the complex numbers in that they have one real part and three imaginary components. As such, the quaternions inherit the notion of conjugation from the complex numbers, $\boldsymbol{q}^* = (q_0, -\boldsymbol{q})$, and the norm of a quaternion is the Euclidean norm $\|\boldsymbol{q}\|^2 = q_0^2 + \|\boldsymbol{q}\|^2 = q_0^2 + q_1^2 + q_2^2 + q_3^2$. Quaternions are also frequently identified with elements of $\mathbb{R}^4$ as

$$\boldsymbol{q} = (q_0, q_1, q_2, q_3) = (q_0, \boldsymbol{q})\,. \tag{42}$$

We use the operation $[(q_0, \boldsymbol{q})]_{\mathbb{R}^3} = \boldsymbol{q}$ to extract the vector in $\mathbb{R}^3$ constructed from the last three entries of a quaternion.

Quaternions are subject to standard element-wise addition, real multiplication by a scalar, and the associative, non-commutative product,

$$\begin{aligned} \boldsymbol{p} \bullet \boldsymbol{q} &= (p_0, \boldsymbol{p}) \bullet (q_0, \boldsymbol{q}) && \text{(43a)} \\ &= (p_0 q_0 - \boldsymbol{p} \cdot \boldsymbol{q},\ p_0 \boldsymbol{q} + q_0 \boldsymbol{p} + \boldsymbol{p} \times \boldsymbol{q})\,. && \text{(43b)} \end{aligned}$$

Under this product, the unit quaternions, i.e. quaternions with $\|\boldsymbol{q}\|^2 = 1$, form a group with identity $\boldsymbol{I}_q = (1, \boldsymbol{0})$ and inverse $\boldsymbol{q}^{-1} = \boldsymbol{q}^*$.

As unit quaternions can be written as

$$\boldsymbol{q} = \left(\cos\left(\frac{\theta}{2}\right), \sin\left(\frac{\theta}{2}\right)\widehat{\boldsymbol{v}}\right), \tag{44}$$

they can be identified with spatial rotations of an angle, θ, anticlockwise about the unit vector, $\widehat{\boldsymbol{v}}$. Specifically, the rotation applied to a vector $\boldsymbol{w}$ is given by

$$(0, \boldsymbol{w}') = \boldsymbol{q} \bullet (0, \boldsymbol{w}) \bullet \boldsymbol{q}^*, \tag{45}$$

where $\boldsymbol{w}'$ is the image of $\boldsymbol{w}$ after its rotation. This rotation can also be written as

$$\boldsymbol{w}' = \boldsymbol{R}(\boldsymbol{q})\,\boldsymbol{w}, \tag{46}$$

where $\boldsymbol{R}(\boldsymbol{q})$ is the rotation matrix whose entries are related to those of the quaternions through

$$\boldsymbol{R}(\boldsymbol{q}) = \begin{pmatrix} 1 - 2q_2^2 - 2q_3^2 & 2(q_1 q_2 - q_3 q_0) & 2(q_1 q_3 + q_2 q_0) \\ 2(q_1 q_2 + q_3 q_0) & 1 - 2q_1^2 - 2q_3^2 & 2(q_3 q_2 - q_1 q_0) \\ 2(q_1 q_3 - q_2 q_0) & 2(q_3 q_2 + q_1 q_0) & 1 - 2q_2^2 - 2q_1^2 \end{pmatrix}. \tag{47}$$

For successive rotations first by $\boldsymbol{q}$, say, followed by $\boldsymbol{p}$, we have that $\boldsymbol{R}(\boldsymbol{p} \bullet \boldsymbol{q}) = \boldsymbol{R}(\boldsymbol{p})\boldsymbol{R}(\boldsymbol{q})$.

2.9.2 Computing the internal moments from the quaternions

As we use quaternions to describe how the local frame rotates as the filament bends and twists, it is convenient to have an expression for the internal moments in terms of the quaternions themselves.

Specifically, the unit quaternions provide the rotation of the standard basis to the local frame vectors, and we have that at each point along the filament and at each time,

$$\boldsymbol{R}\left(\boldsymbol{q}\left(s,t\right)\right) = \left(\widehat{\boldsymbol{t}}\left(s,t\right) \quad \widehat{\boldsymbol{\mu}}\left(s,t\right) \quad \widehat{\boldsymbol{\nu}}\left(s,t\right)\right). \tag{48}$$

In Appendix A.1, we describe how to obtain the quaternion that satisfies this condition at $t = 0$.

Using Eq. (48) and the expression for the internal moments in terms of the frame vectors, Eq. (5), the internal moments can be expressed as

$$\boldsymbol{M} = \boldsymbol{R}\left(\boldsymbol{q}\right)\boldsymbol{D}\left(\left[2\boldsymbol{q}^{*} \bullet \frac{\partial \boldsymbol{q}}{\partial s}\right]_{\mathbb{R}^3} - \begin{pmatrix}\gamma_0 \\ \kappa_\mu \\ \kappa_\nu\end{pmatrix}\right), \tag{49}$$

where $\boldsymbol{D} = \mathrm{diag}(K_T, K_B, K_B)$. The details of this derivation are presented in Appendix A.2. This expression is a special case of more general constitutive laws discussed in Zupan *et al.* (2009).

For the discretised system, the internal moments are then given by

$$\boldsymbol{M}_{n+1/2} = \boldsymbol{R}\left(\boldsymbol{q}_{n+1/2}\right)\boldsymbol{D}\left(2\left[\boldsymbol{q}^{*}_{n+1/2} \bullet \left(\frac{\boldsymbol{q}_{n+1} - \boldsymbol{q}_n}{\Delta L}\right)\right]_{\mathbb{R}^3} - \begin{pmatrix}\gamma_0 \\ \kappa_\mu \\ \kappa_\nu\end{pmatrix}\right), \tag{50}$$

where the interpolated quaternion, $\boldsymbol{q}_{n+1/2}$, is constructed by performing half of the rotation from $\boldsymbol{q}_n$ to $\boldsymbol{q}_{n+1}$. Specifically, we have $\boldsymbol{q}_{n+1/2} = (\boldsymbol{q}_{n+1} \bullet \boldsymbol{q}_n^{*})^{1/2} \bullet \boldsymbol{q}_n$ where the square root is defined in Appendix A.1.

2.9.3 Differential-algebraic system and time integration in three dimensions

With the translational and angular velocities of the segments given by the mobility problem, Eq. (35), but with Eq. (50) used to compute $\boldsymbol{T}^{E}$, the positions and quaternions are updated by integrating the differential-algebraic system,

$$\frac{\mathrm{d}\boldsymbol{Y}_n}{\mathrm{d}t} = \boldsymbol{V}_n, \tag{51}$$

$$\frac{\mathrm{d}\boldsymbol{q}_n}{\mathrm{d}t} = \frac{1}{2}\left(0, \boldsymbol{\Omega}_n\right) \bullet \boldsymbol{q}_n, \tag{52}$$

$$\boldsymbol{Y}_{n+1} - \boldsymbol{Y}_n - \frac{\Delta L}{2}\left(\widehat{\boldsymbol{t}}_n + \widehat{\boldsymbol{t}}_{n+1}\right) = \boldsymbol{0}. \tag{53}$$

The differential equation Eq. (52) for the quaternions describes the time evolution of the entire frame of segment n such that taking $\widehat{\boldsymbol{v}}(t) = \boldsymbol{R}(\boldsymbol{q}_n(t))\widehat{\boldsymbol{v}}(0)$ for a vector $\widehat{\boldsymbol{v}}$ is equivalent to (see Appendix A.3) integrating in time the perhaps more familiar expression

$$\frac{\mathrm{d}\widehat{\boldsymbol{v}}}{\mathrm{d}t} = \boldsymbol{\Omega}_n \times \widehat{\boldsymbol{v}}. \tag{54}$$

As when filament motion and deformation were restricted to a plane, we can update the positions using the second-order BDF scheme. While we would like to update the quaternions using a similar scheme, we must also ensure that after each update they continue to have unit norm as to continue to represent rotations. This could be accomplished by simply applying the second-order BDF scheme to Eq. (52) and subsequently normalising the result. We have found in practice, however, that this approach interfaces poorly with the quasi-Newton methods we use to solve the nonlinear system for the segment positions, orientations and Lagrange multipliers as it introduces timestep restrictions for

numerical stability. Instead, we apply a geometric multi-step method (Faltinsen *et al.*, 2001) to perform the multiplicative update

$$\boldsymbol{q}_n^{j+1} = \boldsymbol{p}_n^{j+1} \bullet \boldsymbol{q}_n^j, \tag{55}$$

for some appropriate unit quaternion $\boldsymbol{p}_n^{j+1}$. This approach preserves the norm of the quaternion to machine precision and, in practice, leads to better numerical stability properties of the larger numerical method.

Applying the general framework set out in Faltinsen *et al.* (2001) to our specific case, we have that for sufficiently small Δt, Eq. (52) has solution

$$\boldsymbol{q}\left(t_0 + \Delta t\right) = \exp\left(\boldsymbol{u}\left(t_0 + \Delta t, \boldsymbol{q}\left(t_0 + \Delta t\right)\right)\right) \bullet \boldsymbol{q}\left(t_0\right), \tag{56}$$

where the vector $\boldsymbol{u}$ is an element of the Lie algebra (in this case $\mathfrak{so}(3)$) and the exponential map (Iserles *et al.*, 2000) is given by

$$\exp\left(\boldsymbol{u}\right) = \left(\cos\left(\frac{\|\boldsymbol{u}\|}{2}\right), \sin\left(\frac{\|\boldsymbol{u}\|}{2}\right)\frac{\boldsymbol{u}}{\|\boldsymbol{u}\|}\right). \tag{57}$$

The Lie algebra element itself satisfies the differential equation

$$\frac{\mathrm{d}\boldsymbol{u}}{\mathrm{d}t} = \mathrm{dexp}_{\boldsymbol{u}}^{-1}\left(\boldsymbol{\Omega}\right), \tag{58}$$

with $\boldsymbol{u}(t_0) = \boldsymbol{0}$. The function $\mathrm{dexp}_{\boldsymbol{u}}^{-1}$ is the inverse of the differential of the exponential mapping and is given by (Iserles *et al.*, 2000)

$$\mathrm{dexp}_{\boldsymbol{u}}^{-1}\left(\boldsymbol{\Omega}\right) = \boldsymbol{\Omega} - \frac{1}{2}\boldsymbol{u}\times\boldsymbol{\Omega} - \frac{1}{2\|\boldsymbol{u}\|^2}\left(\|\boldsymbol{u}\|\cot\left(\frac{\|\boldsymbol{u}\|}{2}\right) - 2\right)\boldsymbol{u}\times\left(\boldsymbol{u}\times\boldsymbol{\Omega}\right). \tag{59}$$

In a general sense, the Lie algebra element can be interpreted as the integral of the angular velocity over time. This holds true if the rotation is about a single axis. For example, if $\boldsymbol{\Omega} \equiv \alpha\widehat{\boldsymbol{e}}_z$ for a constant α, then Eq. (58) with initial condition $\boldsymbol{q}(0) = \boldsymbol{I}_q$ gives $\boldsymbol{u}(t) = \alpha t\widehat{\boldsymbol{e}}_z$ and we obtain $\boldsymbol{q}(t) = (\cos(\alpha t/2), 0, 0, \sin(\alpha t/2))$.

By introducing Eq. (58), the problem of updating the unit quaternions is transferred to one of updating the Lie algebra elements. This can be done using a standard, additive scheme with the only caveat being that since the previously updated Lie algebra elements are in the tangent spaces of the previous quaternions, we must also re-centre our coordinate system at the current quaternion when using a multi-step method (Faltinsen *et al.*, 2001). We show in Appendix A.4 that re-centring this yields the simpler second-order BDF scheme

$$\boldsymbol{u}_n^{j+1} = \frac{1}{3}\boldsymbol{u}_n^j + \frac{2}{3}\Delta t\,\mathrm{dexp}_{\boldsymbol{u}_n^{j+1}}^{-1}\left(\boldsymbol{\Omega}_n^{j+1}\right), \tag{60}$$

for the Lie algebra element of segment n.

Combining the Lie algebra element update with that for the positions, as well as the constraints linking the segment orientations and positions, we obtain the system of equations

$$\boldsymbol{Y}_n^{j+1} - \frac{4}{3}\boldsymbol{Y}_n^j + \frac{1}{3}\boldsymbol{Y}_n^{j-1} - \frac{2\Delta t}{3}\boldsymbol{V}_n^{j+1} = \boldsymbol{0}, \tag{61a}$$

$$\boldsymbol{u}_n^{j+1} - \frac{1}{3}\boldsymbol{u}_n^j - \frac{2}{3}\Delta t\,\mathrm{dexp}_{\boldsymbol{u}_n^{j+1}}^{-1}\left(\boldsymbol{\Omega}_n^{j+1}\right) = \boldsymbol{0}, \tag{61b}$$

$$\boldsymbol{Y}_{n+1}^{j+1} - \boldsymbol{Y}_n^{j+1} - \frac{\Delta L}{2}\left(\widehat{\boldsymbol{t}}_n^{j+1} + \widehat{\boldsymbol{t}}_{n+1}^{j+1}\right) = \boldsymbol{0}, \tag{61c}$$

for the updated segment positions, Lie algebra elements, and Lagrange multipliers. As in 2D, $\boldsymbol{V}_n^{j+1}$ and $\boldsymbol{\Omega}_n^{j+1}$ are given by the mobility problem, Eq. (35), with the right-hand side evaluated at $j+1$. For M filaments, each discretised into N segments, this is an $M(9N-3)$ system of nonlinear equations.

As done in the case of planar filament deformations, we reduce the system size before seeking a numerical solution by substituting

$$\boldsymbol{Y}_n^{j+1} = \boldsymbol{Y}_1^{j+1} + \frac{\Delta L}{2} \sum_{m=2}^{n} \left(\widehat{\boldsymbol{t}}_{m-1}^{j+1} + \widehat{\boldsymbol{t}}_m^{j+1} \right) \tag{62}$$

into the position updates. This yields the new system of equations

$$\boldsymbol{Y}_1^{j+1} - \frac{4}{3}\boldsymbol{Y}_1^j + \frac{1}{3}\boldsymbol{Y}_1^{j-1} - \frac{2}{3}\Delta t\, \boldsymbol{V}_1^{j+1} = \boldsymbol{0}, \tag{63a}$$

$$\boldsymbol{u}_n^{j+1} - \frac{1}{3}\boldsymbol{u}_n^j - \frac{2}{3}\Delta t \exp^{-1}_{\boldsymbol{u}_n^{j+1}} \left(\boldsymbol{\Omega}_n^{j+1} \right) = \boldsymbol{0}, \tag{63b}$$

$$\boldsymbol{Y}_1^{j+1} + \frac{\Delta L}{2} \sum_{m=2}^{n} \left(\widehat{\boldsymbol{t}}_{m-1}^{j+1} + \widehat{\boldsymbol{t}}_m^{j+1} \right) - \frac{4}{3}\boldsymbol{Y}_n^j + \frac{1}{3}\boldsymbol{Y}_n^{j-1} - \frac{2}{3}\Delta t\, \boldsymbol{V}_n^{j+1} = \boldsymbol{0}. \tag{63c}$$

As done in the planar case, we may again express this system as $\boldsymbol{f}(\boldsymbol{X}^*) = \boldsymbol{0}$, where the solution $\boldsymbol{X}^*$ contains the updated position for the first segment, all the Lie algebra elements and the Lagrange multipliers. This substitution reduces the dimension of the nonlinear system from $M(9N-3)$ to $6MN$.

3 Numerical solvers for the nonlinear systems

As described in the previous section, updating the filament positions, orientations, and the Lagrange multipliers requires finding a solution $\boldsymbol{X}^*$ of the nonlinear system of equations of the form $\boldsymbol{f}(\boldsymbol{X}^*) = \boldsymbol{0}$. This system is given by Eq. (41) for planar filament deformations and Eq. (63) in the fully three-dimensional case.

To solve the system, one would typically apply Newton's method. Starting with $\boldsymbol{X}_0$ as an initial guess of the solution, the solution is computed iteratively from

$$\boldsymbol{X}_{k+1} = \boldsymbol{X}_k - \boldsymbol{J}^{-1}(\boldsymbol{X}_k)\boldsymbol{f}(\boldsymbol{X}_k), \tag{64}$$

where $\boldsymbol{J}$ is the Jacobian of the system, $\boldsymbol{J} = \boldsymbol{\nabla}_{\boldsymbol{X}}\boldsymbol{f}$ (i.e. $J_{ij}(\boldsymbol{X}) = \partial f_i(\boldsymbol{X})/\partial X_j$). The process is continued until $\|\boldsymbol{f}(\boldsymbol{X})\|$ is smaller than a given tolerance.

For the systems given by Eqs. (41) and (63), the Jacobian $\boldsymbol{J}$ will contain the mobility matrix, $\mathcal{M}$ (see Eq. (22)), as well as its derivative with respect to segment positions. In general, when one uses fast, matrix-free methods, the mobility matrix is not known explicitly. Even if the mobility matrix was known, the Jacobian would still be very complicated and challenging to compute, especially given the dependence of the mobility matrix on particle positions. Additionally, storing the Jacobian would incur significant memory costs when the number of filament segments is large.

To avoid the costs and complications associated with standard Newton's method, we explore how solutions can be obtained using Jacobian-free Newton–Krylov and Broyden's methods. The remainder of this section provides a description of these methods and, in particular, highlights the advantages and drawbacks associated with these approaches when applied to our specific computation. We also provide results from a series of tests comparing these methods and show that Broyden's method with a suitable approximate Jacobian provides the most effective approach to finding the solution.

3.1 Jacobian-free Newton Krylov (JFNK)

The Jacobian-free Newton–Krylov (JFNK) method (Knoll & Keyes, 2004) uses Newton's method to find the solution of the nonlinear system, but avoids an explicit computation of the Jacobian through judicious evaluations of the function, $\boldsymbol{f}$. With this method, the linear system of the form, $\boldsymbol{J}(\boldsymbol{X}_k)\boldsymbol{x} = \boldsymbol{f}(\boldsymbol{X}_k)$ for unknown $\boldsymbol{x}$, that arises during each Newton iteration is solved using a Krylov subspace method, typically GMRES (Saad & Schultz, 1986). Within GMRES, the Jacobian matrix–vector multiplication required at each iteration is approximated by the finite difference formula,

$$\boldsymbol{J}\boldsymbol{v} = \frac{\boldsymbol{f}(\boldsymbol{X} + \delta\boldsymbol{v}) - \boldsymbol{f}(\boldsymbol{X})}{\delta} + \mathcal{O}(\delta). \tag{65}$$

Since in JFNK multiplication by the Jacobian is replaced by Eq. (65), each GMRES iteration requires evaluating $\boldsymbol{f}(\boldsymbol{X})$.

In our case, this involves evaluating Eq. (41) or Eq. (63) and computing the translational and angular velocities of each segment, which itself requires performing a multiplication by the mobility matrix, the most computational costly aspect of the simulation. Thus, although JFNK does not require explicit knowledge of the Jacobian, it could still incur high computational costs if GMRES is slow to converge and many GMRES iterations are required for each Newton iteration. As a result, and as is often the case with GMRES, preconditioning may be required to limit iteration counts and to ensure good convergence rates.

3.2 Broyden's method

Another approach, and the one that we will ultimately adopt, to obtain the solution of the nonlinear system is Broyden's quasi-Newton method (Broyden, 1965). Like the traditional Newton's method, Broyden's method finds the solution iteratively, but instead uses an initial approximate Jacobian, $\boldsymbol{J}_0$, that is improved at each iteration through rank-one updates based on the secant equation. The choice of update, however, is not unique and the so-called "good" Broyden's method updates the Jacobian itself (see Line 4 in Algorithm 1), while the so-called "bad" Broyden's method updates its inverse (Line 5 in Algorithm 2). In both cases, we can employ a limited-memory version (van de Rotten, 2003) of Broyden's method where the updates are stored as vectors rather than altering and writing to memory the full Jacobian at each iteration. Thus, the only matrix that may require storage is $\boldsymbol{J}_0$, which can be chosen with this memory requirement in mind.

Algorithm 1: Limited-memory, good Broyden's method, to solve $\boldsymbol{f}(\boldsymbol{X}) = \boldsymbol{0}$

1 Take initial guess of solution $\boldsymbol{X}_0$, convergence tolerance $\varepsilon > 0$, estimate $\boldsymbol{J}_0$ for the Jacobian at $\boldsymbol{X}_0$
2 $k = 0$
3 **while** $\|\boldsymbol{f}(\boldsymbol{X}_k)\| > \varepsilon$ **do**
4 $\quad \Delta\boldsymbol{X}_k = -\left[\boldsymbol{J}_0 + \sum_{i=1}^{k} \boldsymbol{c}_i \boldsymbol{d}_i^\top\right]^{-1} \boldsymbol{f}(\boldsymbol{X}_k)$
5 $\quad \boldsymbol{X}_{k+1} = \boldsymbol{X}_k + \Delta\boldsymbol{X}_k$
6 $\quad \boldsymbol{c}_{k+1} = \boldsymbol{f}(\boldsymbol{X}_{k+1})/\|\Delta\boldsymbol{X}_k\|$, $\boldsymbol{d}_{k+1} = \Delta\boldsymbol{X}_k/\|\Delta\boldsymbol{X}_k\|$
7 $\quad k \coloneqq k + 1$
8 **return** $\boldsymbol{X}_k$

Algorithm 2: Limited-memory, bad Broyden's method, to solve $\boldsymbol{f}(\boldsymbol{X}) = \boldsymbol{0}$

1 Take initial guess of solution $\boldsymbol{X}_0$, convergence tolerance $\varepsilon > 0$, estimate $\boldsymbol{J}_0$ for the Jacobian at $\boldsymbol{X}_0$
2 $k = 0$
3 Evaluate $\boldsymbol{J}_0^{-1}$
4 **while** $\|\boldsymbol{f}(\boldsymbol{X}_k)\| > \varepsilon$ **do**
5 $\quad \Delta\boldsymbol{X}_k = -\left[\boldsymbol{J}_0^{-1} + \sum_{i=1}^{k} \boldsymbol{c}_i \boldsymbol{d}_i^\top\right] \boldsymbol{f}(\boldsymbol{X}_k)$
6 $\quad \boldsymbol{X}_{k+1} = \boldsymbol{X}_k + \Delta\boldsymbol{X}_k$
7 $\quad \boldsymbol{c}_{k+1} = -\left[\boldsymbol{J}_0^{-1} + \sum_{i=1}^{k} \boldsymbol{c}_i \boldsymbol{d}_i^\top\right] \boldsymbol{f}(\boldsymbol{X}_{k+1})/\|\Delta\boldsymbol{f}_k\|$, $\boldsymbol{d}_{k+1} = \Delta\boldsymbol{f}_k/\|\Delta\boldsymbol{f}_k\|$, where $\Delta\boldsymbol{f}_k = \boldsymbol{f}(\boldsymbol{X}_{k+1}) - \boldsymbol{f}(\boldsymbol{X}_k)$
8 $\quad k \coloneqq k + 1$
9 **return** $\boldsymbol{X}_k$

For Broyden's method to converge, not only must the convergence conditions for Newton's method be satisfied, but also the initial guess for the Jacobian, $\boldsymbol{J}_0$, must be sufficiently close to $\boldsymbol{J}$. The process of choosing $\boldsymbol{J}_0$ is similar to that of finding a suitable preconditioner in that one seeks $\boldsymbol{J}_0 \approx \boldsymbol{J}$ with the

condition that systems involving $\boldsymbol{J}_0$ should be easily solved. Below we present two approaches that we have explored to obtain approximate Jacobians based on simplified, yet related, physical systems.

3.2.1 Jacobian-free Broyden's method (JFB)

Our first approach involves applying the JFNK-style finite differencing, Eq. (65), to a function, $\boldsymbol{f}_0(\boldsymbol{X}) \approx \boldsymbol{f}(\boldsymbol{X})$, to perform the matrix–vector multiplications by $\boldsymbol{J}_0 = \boldsymbol{\nabla}_{\boldsymbol{X}} \boldsymbol{f}_0$ and solve the systems appearing in Algorithm 1 or Algorithm 2 using GMRES. As this combines a Jacobian-free approach with Broyden's method, we refer to this approach as Jacobian-free Broyden's (JFB) method.

In the tests presented below, the function $\boldsymbol{f}_0$ involves all the steps of the model presented in Section 2, however, we omit both the hydrodynamic interactions between filament segments by replacing the mobility matrix $\mathcal{M}$ by $\mathcal{M}_0 = \mathrm{diag}(\mathcal{M})$ and remove the steric interactions between filament segments. In this way, the computationally expensive mobility matrix multiplication is only performed once per Broyden iteration.

3.2.2 A block diagonal, explicit J_0 (EJB)

Along with JFB, we also construct explicitly the approximate Jacobian based on $\boldsymbol{f}_0(\boldsymbol{X})$ with a diagonal mobility matrix and no steric interactions between filaments. By differentiating $\boldsymbol{f}_0(\boldsymbol{X})$, we obtain expressions for the entries of $\boldsymbol{J}_0$ which we evaluate at each timestep. These expressions are provided in Appendix B.

In general, for simulations involving M filaments, each with N segments, $\boldsymbol{J}_0$ will be $3NM \times 3NM$ for filament motion confined to a plane, and $6NM \times 6NM$ for the fully 3D case. When interactions between the filaments are ignored, $\boldsymbol{J}_0$ will be block diagonal and consist of M blocks of size $3N \times 3N$ for 2D motion and $6N \times 6N$ for fully 3D simulations. As a result, solutions to systems involving $\boldsymbol{J}_0$ can be found by solving the M smaller, independent linear systems corresponding to each block. Additionally, only the M blocks, rather than the entire matrix, need to be stored in memory.

3.3 A comparison of the nonlinear system solvers

Here, we test and compare the different approaches for solving $\boldsymbol{f}(\boldsymbol{X}) = \boldsymbol{0}$. Based on the tests presented below, we find that bad Broyden's method using the explicit expressions for $\boldsymbol{J}_0$ (see Appendix B) provides the most effective approach for solving the system arising from the discretisation of the differential-algebraic system, Eqs. (41) and (63).

In our tests, we consider a monolayer of M filaments, each formed of N segments and subject to a constant force per unit length of magnitude W in the plane of the monolayer. The simulations are performed using the MATLAB with FCM implementation described in the next section. The filaments are initially straight and are distributed uniformly and isotropically in the centre-plane of a periodic domain of size $141\Delta L \times 141\Delta L \times 8.8\Delta L$. Varying N between 5 and 60, we explore a range of filament lengths $L = N\Delta L$. FCM is used to resolve the hydrodynamic interactions between the filaments, and the Stokes equations are solved on a grid of $1024 \times 1024 \times 64$ points. With the domain size fixed, the computational cost of FCM scales linearly with number of filament segments. The filaments are allowed to settle for two characteristic sedimentation times defined as $T = \eta L/W$. The parameter W is set such that the dimensionless elasto-gravitational number (Cosentino Lagomarsino *et al.*, 2005) is $B = L^3 W/K_B = 1000$ and filament deformation is significant. The timestep size is set to be $\Delta t = T/300$.

In the following tests, the GMRES residual tolerance for both JFNK and JFB is 10^{-4} and the JFNK finite difference parameter δ is set to 10^{-7}. The Newton and Broyden iteration tolerances for the infinity norm of the residual relative to a are both $\varepsilon = 10^{-4}$. These parameters have been selected to minimise wall times while still ensuring sufficient accuracy. To quantify solver performance, we measure average wall times, the average number of mobility matrix multiplications per timestep, and the average iteration count per timestep. These averages of are computed between simulation times $t = T$ and $t = 2T$ once the suspension has evolved away from its initial configuration. All simulations were performed on a 16-core 2.5GHz AMD Opteron 6380 processor.

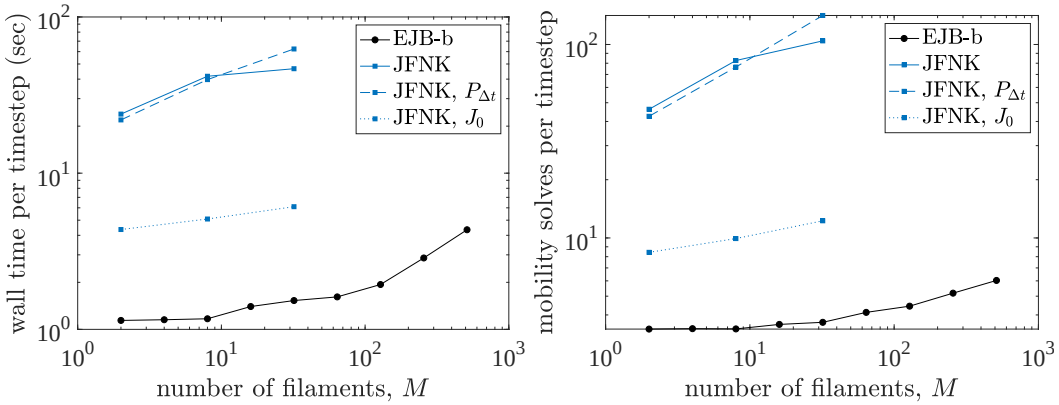


Figure 2: Comparison between bad Broyden's method with $\boldsymbol{J}_0$ (EJB-b) and JFNK with three different preconditioner options: (i) unpreconditioned, (ii) $\boldsymbol{P}_{\Delta t}$, and (iii) $\boldsymbol{J}_0$. **Left:** Average wall time per timestep for M filaments formed of $N = 15$ segments. **Right:** Average number of mobility solves required per timestep.

3.3.1 JFNK versus Broyden's method (JFNK v EJB-b)

We first compare wall times per timestep for the simulation run with JFNK with those using "bad" Broyden's method with the explicit approximate Jacobian (EJB-b), Algorithm 2. A key aspect of using JFNK effectively is the choosing the correct preconditioner to accelerate GMRES convergence. Here, we explore two preconditioners. In the first instance, employ a diagonal right preconditioner that mitigates ill-conditioning due to the factor of Δt appearing in entries of the Jacobian associated with the system Eqs. (41) and (63). Since other entries in the Jacobian are $\mathcal{O}(1)$, small timesteps can increase the Jacobian condition number, resulting in slow GMRES convergence. As demonstrated explicitly in Appendix B for a drag-only mobility matrix, this factor of Δt appears in the Jacobian as a result of variations with respect to the Lagrange multipliers.

Accordingly, the relevant diagonal entries of the right preconditioner are set to $\Delta t/(6\pi\eta a)$. Specifically, for a single, N segment filament in 2D, the right preconditioner is

$$\boldsymbol{P}_{\Delta t} = \begin{pmatrix} \boldsymbol{I}_{N+2} & \boldsymbol{0} \\ \boldsymbol{0} & \frac{\Delta t}{6\pi\eta a}\boldsymbol{I}_{2N-2} \end{pmatrix}, \tag{66}$$

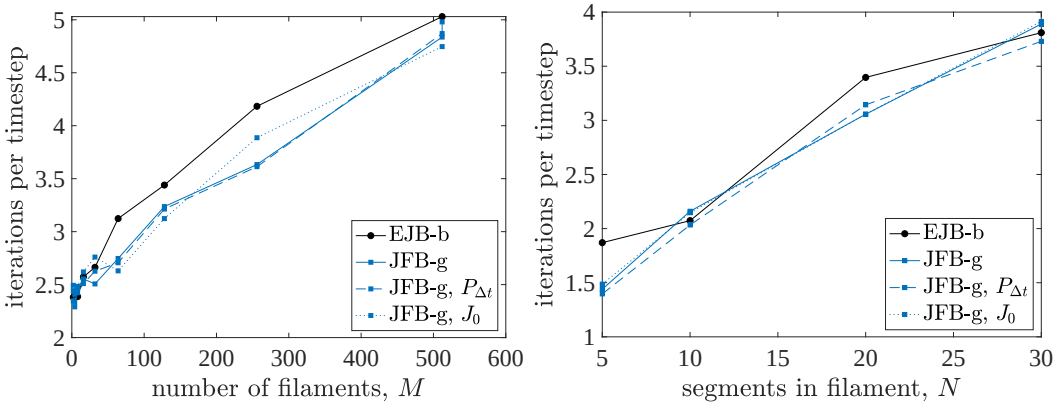
with the system ordered as in Eq. (41).

To provide a balanced comparison with Broyden's method, we also use as a left preconditioner the block-diagonal, explicit Jacobian described in Section 3.2.2 and derived in Appendix B. Based on the filament model with a diagonal mobility matrix and no steric iteractions, this preconditioner captures the local physics of the problem and its block structure allows for rapid system solves and reduced storage costs.

In Fig. 2, we compare the performance of JFNK with these preconditioners against that of EJB-b. The left panel of Fig. 2 shows the average wall time per timestep, while the right panel indicates the average number of mobility solves required per timestep. By "mobility solve", we mean multiplying by the mobility matrix using the force-coupling method (FCM) as described in Section 2.7.2. We see that left preconditioning with the approximate Jacobian improves the performance of JFNK much more so than $\boldsymbol{P}_{\Delta t}$. Based on our wall time measurements, however, EJB-b is still approximately five times faster than JFNK. The additional costs per timestep associated with JFNK are due to the higher number of mobility solves that it requires. While EJB-b needs only one mobility solve per Broyden iteration, each Newton iteration in JFNK requires multiple GMRES iterations that each cost one mobility solve. Due to these additional costs, JFNK is not competitive for solving the nonlinear system.

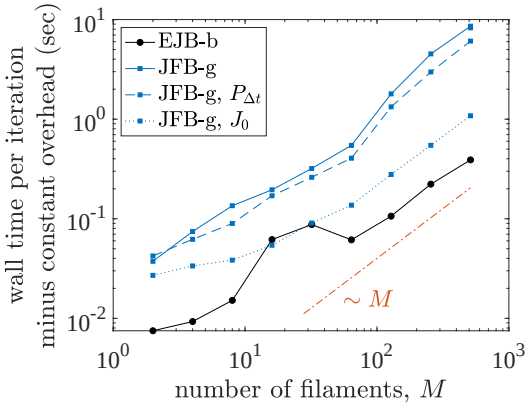
3.3.2 Choice of Broyden's method (EJB-b v JFB-g)

In this section, we compare the performance of a JFB implementation of good Broyden's method (JFB-g) with EJB-b. Implementing *good* Broyden's method with the explicit approximate Jacobian is

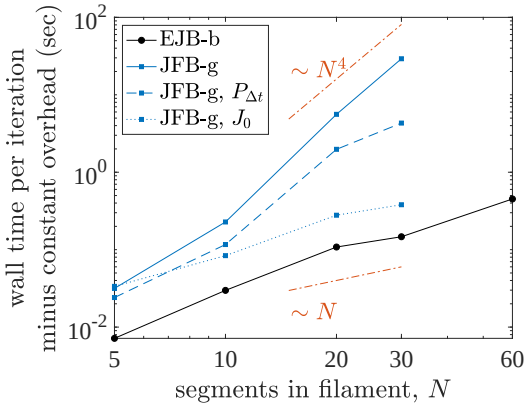


(a) Average number of Broyden iterations per timestep for M filaments, formed of $N = 15$ segments. Filaments are initially straight and randomly oriented.

(b) Average number of Broyden iterations per timestep for $M = 128$ filaments, formed of N segments. Filaments are initially straight and aligned horizontally.



(c) Average wall time per iteration for M filaments, formed of $N = 15$ segments. Filaments are initially straight and randomly oriented.



(d) Average wall time per iteration for $M = 128$ filaments, formed of N segments. Filaments are initially straight and aligned horizontally.

Figure 3: Comparison between bad Broyden's method with $\boldsymbol{J}_0$ (EJB-b) and JFNK with three different preconditioner options: (i) unpreconditioned, (ii) $\boldsymbol{P}_{\Delta t}$, and (iii) $\boldsymbol{J}_0$. For the wall time measurements, we have subtracted the initial wall time per iteration of approximately 0.3 s that is independent of M and N and is associated with setting up the fluid solver.

computationally expensive since it would require solving a dense linear system that changes with each Broyden iteration (see Line 4 in Algorithm 1). In contrast, for bad Broyden's method, we avoid the JFB implementation since we update directly the inverse of the explicit approximate Jacobian. As a result, the system we need to solve at each Broyden iteration remains the same.

As discussed in Section 3.3.1 for JFNK, preconditioning is also required to accelerate GMRES convergence in JFB-g. In the tests presented below, we again trial the right preconditioner $\boldsymbol{P}_{\Delta t}$, as well as the approximate Jacobian, $\boldsymbol{J}_0$, as a left preconditioner.

The performance measurements for JFB-g with and without preconditioners and EJB-b simulations are presented in Fig. 3. For all approaches, the number of Broyden iterations per timestep increases similarly with both the number of segments per filament (Fig. 3(b)) and the number of filaments (Fig. 3(a)). This increase is expected, as increasing the number of filaments or segments increases the concentration of the suspension. This in turn increases filament interactions and the number of iterations required.

When we examine the wall time per Broyden iteration, however, differences between the approaches begin to emerge. Fig. 3(c) shows the wall time per iteration as a function of the number of filaments. We find that for all methods, the wall time per iteration scales linearly with the number of filaments M. This is a result of the FCM mobility solve being the most expensive part of the iteration and its

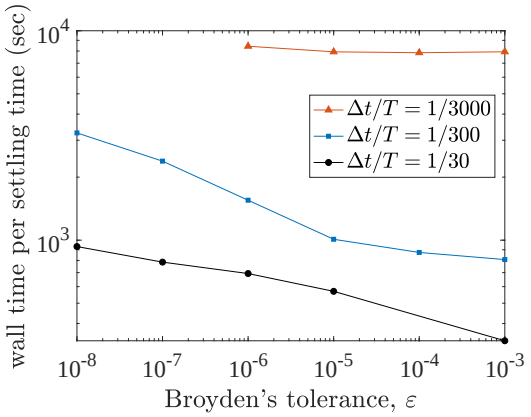


Figure 4: Average wall time per sedimentation time for EJB-b as the Broyden tolerance and timestep size are changed. The simulations are performed with $M = 256$ filaments, formed of $N = 15$ segments and $B = 1000$. The filaments are initially straight and randomly oriented. The simulation with $\varepsilon = 10^{-4}$ and $\Delta t/T = 1/30$ failed to converge.

cost scaling linearly with the total number of segments.

The wall time per iteration as the number of segments is varied is indicated in Fig. 3(d). For JFB-g with $\boldsymbol{P}_{\Delta t}$ as the preconditioner, or no preconditioner at all, the computational time increases at a more rapid rate of approximately N^4. This rapid increase in cost is a result of the approximate Jacobian system size growing with N and poor GMRES convergence. For example, we find that for a JFB-g simulation of $M = 512$ filaments each with $N = 15$ segments, without preconditioning, 94% of the wall time per iteration is spent in the GMRES routine to solve the Jacobian system. We do see, however, using an effective preconditioner, in this case $\boldsymbol{J}_0$, can bring down this cost. For EJB-b and JFB-g with preconditioner $\boldsymbol{J}_0$, we observe a slightly faster than linear growth in computational time with the number of segments. Despite the reduction due to preconditioning, EJB-b still outperforms JFB by approximately a factor of 2.

In addition to being the more computationally efficient method, EJB-b is demonstrably more robust. In practice, for JFB-g, we found that the convergence of Broyden's method relied heavily on the quality of the solution given by GMRES. Consequently, a low residual tolerance was typically required. We also observed that GMRES could be prone to stagnation and found that GMRES convergence is sensitive to the choice of δ in Eq. (65).

3.3.3 Effects of tolerance and timestep size on performance

The Broyden iteration tolerance and the timestep size not only affect the accuracy of the simulation but also the computation time. In particular, while increasing the timestep size reduces the number of timesteps, larger timestep sizes require more Broyden iterations in order to reach the prescribed tolerance as initial guesses are extrapolated from solutions at past timesteps.

Measurements of the wall time for a full settling time, T, at different timestep sizes and different Broyden's tolerances are presented in Fig. 4 for EJB-b simulations of a concentrated suspension (area fraction 12.5%). In general, increasing the Broyden's tolerance decreases the wall time, though with reduced effect at the smallest timestep sizes. Due to differences in iteration numbers, we find that decreasing the timestep from $\Delta t/T = 1/30$ to $1/300$ increases the computation time by only a factor of ~ 3. We also note that for $\Delta t/T = 1/30$, Broyden's method fails to converge in the case where the tolerance was $\varepsilon = 10^{-4}$. Reducing the timestep further to $\Delta t/T = 1/3000$, however, does not lead to additional reductions in the number of Broyden iterations, and consequently, the computation times for these simulations are approximately 10 times greater than those with $\Delta t/T = 1/300$.

3.3.4 Effects of segment and filament number on performance

In general, the walltime impact of changing the number of segments, N, or filaments, M, can be complicated. In practice, we found that increasing local concentrations of filaments can lead to higher

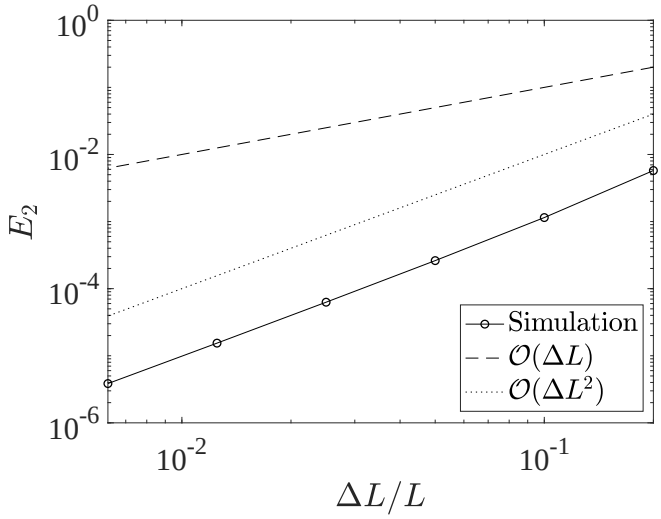


Figure 5: Error between the simulated steady-state filament shape and the analytical solution given by Landau & Lifshitz (1986) for different values of segment separation ΔL.

numbers of Boyden iterations. For each iteration, the predominant computations under EJB-b are the hydrodynamic mobility solve and the inversion of the approximate Jacobian, $\boldsymbol{J}_0$. Given that $\boldsymbol{J}_0$ is block diagonal, the complexity of this operation is at worst $\mathcal{O}(N^3)$. For filaments with $N \lesssim 50$, we have found that the hydrodynamic mobility solve is the more costly aspect of the computation.

The total number of segments impacts the computational work required to apply the mobility matrix. In Section 2.7, we discussed the two hydrodynamic models used in this work: the force-coupling method (FCM) for periodic domains and RPY for unbounded domains. As we use a direct pairwise summation, the computational work with our implementation of RPY scales like $\mathcal{O}((NM)^2)$. For FCM, the computational work scales as $\mathcal{O}(NM)$, and is linked to the cost of constructing the body-force in the Stokes equations used to represent the forces the segments exert on the fluid. This scaling is seen in Fig. 3(c, d). It is important to note that for FCM this cost is in addition to the fixed cost of the FFT-based Stokes solver that scales like $\mathcal{O}(N_g \log N_g)$, where N_g is the total number of grid points. Thus, the computational work associated with FCM will scale linearly with the number of segments only once the filament density is sufficiently high.

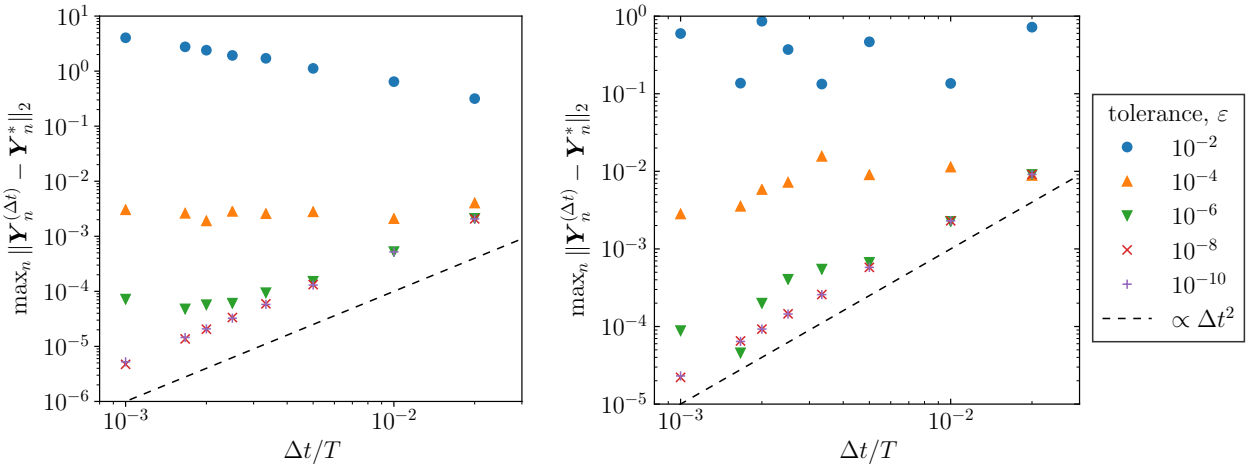
4 Accuracy and convergence

In this section, we present results from numerical experiments that demonstrate the finite difference scheme's second-order convergence in space, as well as the second-order convergence in time for the 2D and 3D implementations of the BDF schemes. In addition, we provide a comparison between the RPY version of the hydrodynamic model and numerical solutions to the boundary integral equations for the hydrodynamic forces experienced by a straight filament.

4.1 Convergence in space

To demonstrate the second-order convergence of the finite difference scheme used to discretise the force and moment balances, Eqs. (10) to (12), we consider an initially horizontal filament that is clamped at one end ($s = 0$) and subject to a vertical force of magnitude $F = 1.93K_B/L^2$ at the free end ($s = L$). The filament bends in the direction of the force, and after some time, reaches an equilibrium shape described by $\theta(s)$, the angle the filament makes with the vertical. After integrating the continuous force and moment balances, one finds that the equilibrium angle satisfies (Landau & Lifshitz, 1986, §19)

$$s = \sqrt{\frac{K_B}{2F}} \int_{\theta(s)}^{\pi/2} \frac{\mathrm{d}\phi}{\sqrt{\cos(\theta(L)) - \cos(\phi)}}. \tag{67}$$



(a) 2D convergence: $M = 2$ settling filaments

(b) 3D convergence: $M = 4$ settling filaments

Figure 6: Maximum filament position error at $t_f = 20T$ for two (in 2D) or four (in 3D) settling filaments with $N = 30$ and $B = 10^3$. Units are such that the RPY radius of each segment is $a = 1$. The error is estimated by comparing to the very accurate solution $\boldsymbol{Y}^*(s, t_f)$ computed with tolerance $\varepsilon = 10^{-12}$ and $\Delta t/T = 10^{-4}$. Simulations were performed using BDF2 and in both cases, we obtained the expected convergence rate for appropriately chosen tolerances.

Fig. 5 shows the L^2 error,

$$E_2 = \sqrt{\Delta L \sum_{n=1}^{N} (\theta^n - \theta(s_n))^2}, \tag{68}$$

where θ^n is the angle for segment n given by the finite difference scheme, as a function of ΔL. We observe second-order convergence to the analytical solution as we decrease ΔL.

4.2 Convergence in time

Here, we confirm the second-order convergence of the BDF2 schemes for two- and three-dimensional filament motion. To test the 2D scheme, we consider a pair of filaments that are initially straight, oriented vertically, and separated horizontally. The filaments are then allowed to settle as a result of a constant vertical force per unit length applied to each. For the 3D scheme, we consider the same settling problem, but for four initially straight filaments that are aligned vertically and arranged such that when viewed from above, they are at the corners of a square (cf. Fig. 19). In all 2D and 3D simulations, we use pairwise hydrodynamics via RPY and the dimensionless elasto-gravitational number is $B = L^3 W/K_B = 1000$, where W is the force per unit length acting on each filament. Each filament consists of $N = 30$ segments and the simulations are run for $t_f = 20T$. The settling time is denoted as $T = \eta L/W$. With these parameters, the filaments bend as they settle and move apart from one another until they each reach a horseshoe-like shape (cf. Fig. 15).

Fig. 6 shows the error based on the maximum Euclidean distance of the positions $\boldsymbol{Y}^{(\Delta t)}(s, t_f)$ to the very accurate solution $\boldsymbol{Y}^*(s, t_f)$ that was computed using 10^4 timesteps per settling time and a tolerance of 10^{-12}. We remind the reader that the positions along the length of the filament are computed using the values of the angles (in 2D) or quaternions (in 3D), and thus their accuracy depends on that of the BDF scheme. For both the 2D and 3D schemes, we do observe second-order accuracy provided that the tolerance for the Broyden's iterations is sufficiently low. Thus, a key parameter in ensuring accuracy of the solution is not only the timestep, but the tolerance to which the resulting system of equations is solved.

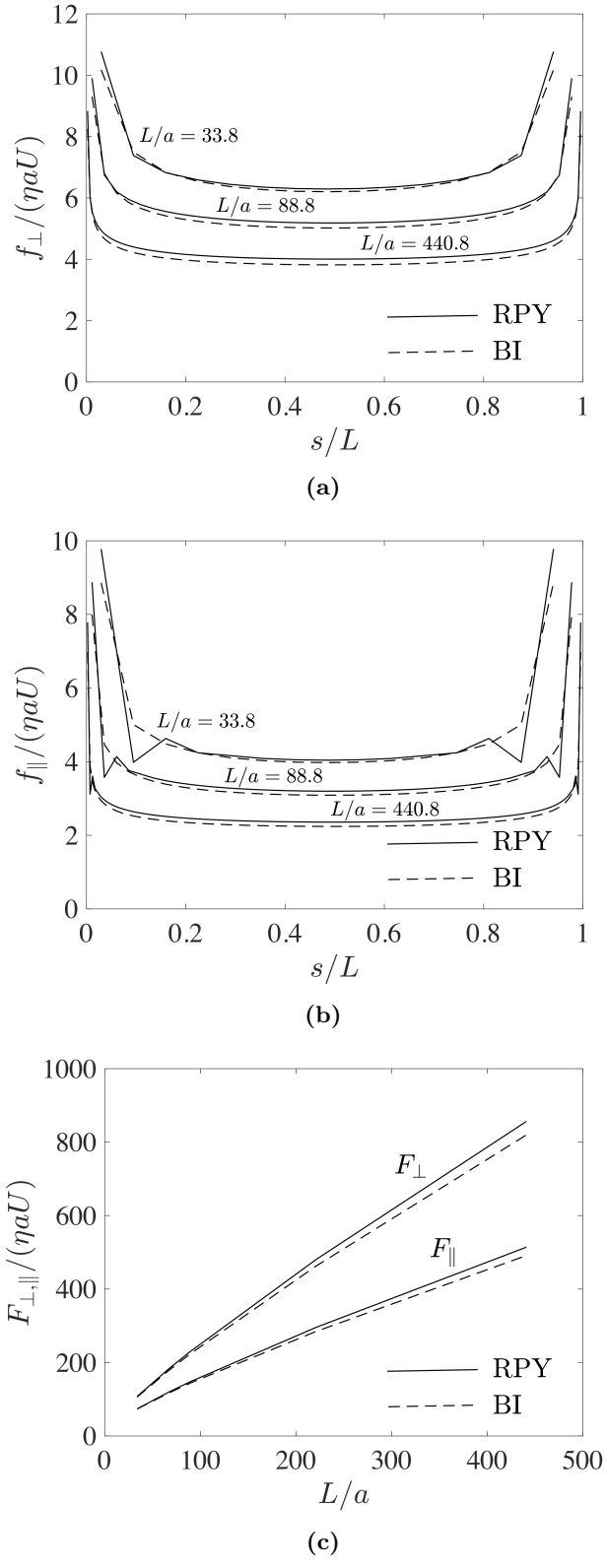


Figure 7: A comparison between the RPY model and boundary integral (BI) computations for (a) the segment forces as a function of s for translation perpendicular to the filament axis, (b) the segment forces as a function of s for translation parallel to the filament axis, and (c) the total force on the filament as a function of L/a for both perpendicular and parallel translation.

4.3 Accuracy of the hydrodynamic model

In this section, we examine the accuracy of the hydrodynamic models that we employ in this work to solve the mobility problem and obtain the translational and angular velocity of the segments. In this test, we consider the simplest case of a translating rigid, straight filament with a circular cross-section of radius a. We compute the force on each segment using the RPY model described above and compare these values with those given by a second-order discretisation of the boundary integral (BI) equations for rigid body motion (Keaveny & Shelley, 2011). For the boundary integral solver, we take the surface of the filament to be that of a cylinder with radius a that has spherical caps. The spacing between the segments with the RPY model is $\Delta L = 2.2a$. We only compare the RPY model with the BI values since both solvers are configured for an unbounded fluid, while our implementation of FCM requires the imposition of periodic boundary conditions. We do expect, though, that for an unbounded fluid, FCM will yield force values similar to those of RPY as the expressions (Maxey & Patel, 2001; Lomholt & Maxey, 2003) for the FCM fluid velocity and particle mobility are asymptotic to their RPY counterparts.

Fig. 7(a) shows the force on the filament segments as a function of s given by the RPY model and BI computations for the case where the filament translates with speed U in a direction perpendicular to its axis. The segment force values are shown for filaments with different aspect ratios, L/a. Similarly, Fig. 7(b) shows the force on the segments given by the two methods when the filament translates with speed U in the direction of its axis, while Fig. 7(c) shows the total force as a function of aspect ratio. We see that while the RPY model slightly overestimates the segment and total force values with a maximum relative error of approximately 4%, it does reproduce the general dependence of the segment force on s given by the BI computations. We also observe that the error is largely independent of the filament aspect ratio.

5 Simulations

In this section, we present results from several simulations that range from a single tethered filament whose position is fixed at one end, to many interacting filaments moving freely in the surrounding fluid.

5.1 Implementations

In the simulations that follow, we used one of three implementations of the method:

1. **MATLAB with RPY**: For the single filament simulations, we have used a MATLAB implementation of the method in which the hydrodynamic interactions are accounted for through the RPY mobility matrices. We have made this implementation available on GitHub (Schoeller *et al.*, 2019).

   For our simulations involving tethered filament arrays, we have accelerated the RPY evaluation by using MATLAB's MEX functionality and performing the pairwise computation in C.

2. **MATLAB with FCM**: For the two-dimensional simulations in periodic domains involving many sedimenting filaments or active swimmers, we use a MATLAB–MEX implementation of the method with FCM resolvubg the hydrodynamic interactions between the segments. Specifically, since with FCM we must solve the Stokes equations, we use C functions to loop through the grid points when assembling the FCM fluid forcing, and to loop through the wave numbers when inverting the Stokes operator. The remaining aspects of the computation, including the FFTs, are performed in MATLAB.

3. **C++ with RPY:** The final simulations of sedimenting filament clouds were performed with a more powerful C++ implementation of the method that takes advantage of the Armadillo linear algebra library (Sanderson & Curtin, 2016, 2018). In addition, the RPY computation, though still performed pairwise, was parallelised using OpenMP.

5.2 Tethered filaments

5.2.1 Rotational dynamics of a single tethered filament

In Coq *et al.* (2008, 2009), an elastic filament is immersed in a tank of pure glycerine. The filament is connected to a motor at its base such that at rest, the filament forms an angle of 15° with the motor axis. The base of the filament is displaced from the motor axis by $\delta_0 = 2\,\text{mm}$ (see Fig. 8a). The filament has radius $a = 435\,\mu\text{m}$, and its length is varied between 2 cm and 10 cm. A dimensionless Sperm number, *Sp*, that characterises the ratio of viscous to elastic forces is introduced as

$$Sp = \frac{\zeta_\perp \omega L^4}{K_B}, \tag{69}$$

where $\zeta_\perp = 4\pi\eta/(\log(L/a) + 1/2)$ is the transverse drag coefficient for a straight rod, ω is the rotation rate of the motor, and K_B is the bending modulus of the elastic filament. By varying ω, a range of *Sp* can be explored and the steady-state distance, d, of the free end from the rotation axis is recorded. At lower *Sp* the filament barely deforms from its straight configuration and undergoes a nearly rigid body rotation such that $d \approx \delta_0 + L\sin(15°)$. At larger *Sp*, the filament slowly collapses onto the rotation axis and $d \approx 0$.

In order to simulate this experiment, the system of equations Eq. (61) is supplemented with the additional constraints such that for timestep j,

$$\boldsymbol{Y}_1 = \begin{pmatrix} \delta_0 \cos(j\omega\Delta t) \\ \delta_0 \sin(j\omega\Delta t) \\ 0 \end{pmatrix}, \tag{70}$$

to tether the filament base to the motor, and

$$\boldsymbol{u}_1^j = \begin{pmatrix} 0 \\ 0 \\ \omega\Delta t \end{pmatrix}, \tag{71}$$

to rotate the filament base with the motor. These constraints are associated with two new vector Lagrange multipliers $\boldsymbol{\lambda}_1$ and $\boldsymbol{\lambda}_2$, and consequently the first segment experiences additional constraint forces and torques given by

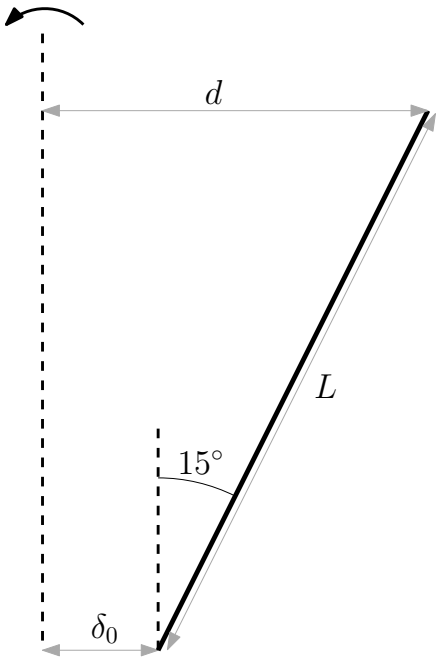
$$\boldsymbol{F}^C = \boldsymbol{\lambda}_1, \tag{72}$$

$$\boldsymbol{T}^C = \boldsymbol{D}^\top \boldsymbol{\lambda}_2, \tag{73}$$

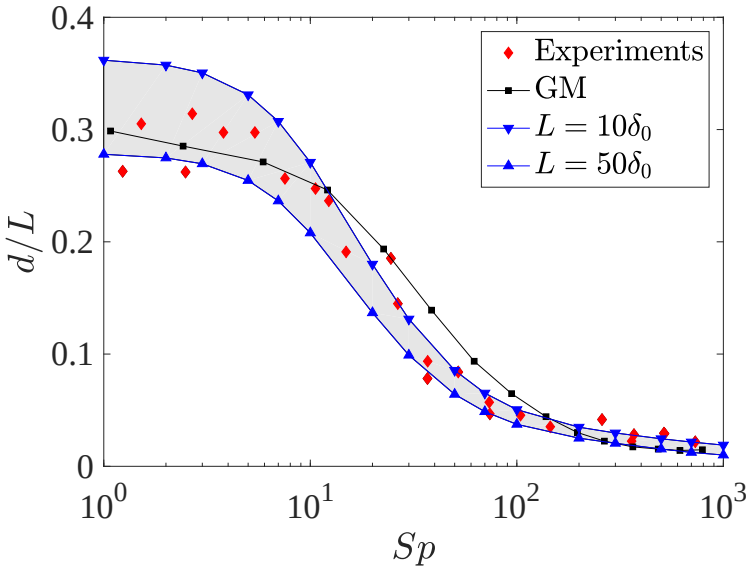
where $\boldsymbol{D}$ is the matrix such that $\boldsymbol{D}\boldsymbol{v} = \text{dexp}^{-1}_{\boldsymbol{u}_1^j}(\boldsymbol{v})$. The total differential algebraic system is modified by substituting these new constraints, Eqs. (70) and (71), into the update equations for the first segment.

To compare our simulations to the experimental results, we match the ratio $\delta_0/a = 4.5977$ and the length range $10\delta_0 \leq L \leq 50\delta_0$ by varying the number of segments, N. For each length considered, the simulation values of ω or K_B are varied to explore the range of *Sp*. In addition, the simulations are run using the RPY mobility matrices (see Section 2.7.1) to capture segment hydrodynamic interactions.

In Fig. 8b, we present the filament end distance, d/L, as a function of *Sp* from our simulations for the range of filament lengths. Along with this data, we show the values measured in the experiments in Coq *et al.* (2008). Though the exact filament lengths associated with the data points were not indicated in Coq *et al.* (2008), the values of d/L measured in the experiments lie broadly within the range given by our simulations. For comparison, we have also included similar results given by the gears model (GM) from Delmotte *et al.* (2015). While these results largely coincide with those from our simulations and the experiments, there are some differences that may be attributed to the interpretation of the offset distance, δ_0, as discussed in Delmotte *et al.* (2015).



(a) A diagram of the rotating filament simulation. The distance of the base from the axis of rotation, δ_0, is a fixed value across all of our simulations. The distance of the filament end from the axis is d.



(b) The normalised end distance, d/L, as a function of Sp. Results from our simulations are compared to those of the Gears Model (GM) (Delmotte *et al.*, 2015) and the experimental results of (Coq *et al.*, 2008). The two blue curves correspond to values obtained using the longest and shortest filaments in both our simulations and the experiments. The enclosed shaded area indicates the values at intermediate filament lengths.

Figure 8: Deflection of a tethered rotating filament.

5.2.2 Coordination in cilia arrays

In this section, we extend our study of a single tethered elastic filament to the simulation of a model ciliary array. Cilia are biologically occurring elastic filaments that undergo time-periodic motion and tend to exhibit metachronal waves and coordination (Brennen & Winet, 1977; Guo *et al.*, 2018). A simplified model of cilia dynamics involves the "geometric switch" (Guo *et al.*, 2018; Kim & Netz, 2006), wherein a driving torque is applied to the base of an elastic filament until some fixed angle $\theta_{\max}$ is reached with respect to the vertical. At this point, the direction of the torque is reversed until the filament reaches an angle of $-\theta_{\max}$, when it switches again and so on. In Kim & Netz (2006), both the direction and the magnitude of the driving torque change once the critical angle is reached (i.e. $\tau_{\text{slow}} \mapsto \tau_{\text{fast}} = -\alpha\tau_{\text{slow}} \mapsto \tau_{\text{slow}} \mapsto \cdots$ with $\alpha \geq 1$), allowing for distinct "effective" and "recovery" strokes. The resulting filament motion, which is non-reciprocal due to its elasticity, allows for the net pumping of the fluid.

Using our methodology, we first simulate a single model cilium comprised of $N = 20$ segments. The position of the base segment is constrained to remain fixed to where it is attached to a no-slip wall. The hydrodynamic effects of the wall are included through the RPY tensor for a half-space given in Swan & Brady (2007). In addition, for numerical stability of our implicit scheme, we allow for a smooth transition between the torque values once $\theta_{\max}$ or $-\theta_{\max}$ are reached. This is accomplished by introducing a transition period of several timesteps during which the driving torque is taken from a sigmoid curve connecting τ_{slow} to τ_{fast}. Fig. 9a shows the shape of the cilium over a period of oscillation given by our simulations. The differences in filament shape and speed due to the different values of τ_{fast} and τ_{slow} can be clearly observed.

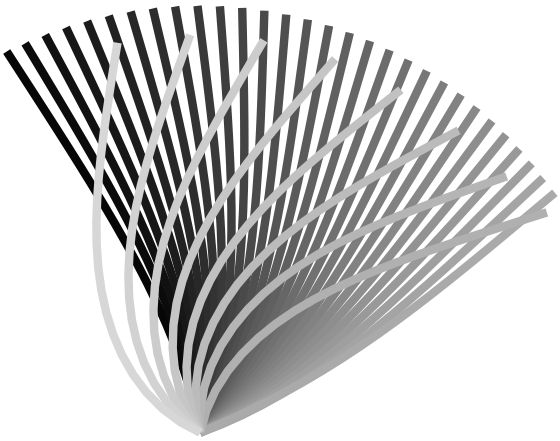
Since the switching between τ_{fast} and τ_{slow} is determined by the tangent vector at the model cilium's base, Kim & Netz (2006) observed that hydrodynamically interacting pairs of model cilia can phase-lock regardless of their initial phase difference. Using our model, we performed similar simulations of two interacting model cilia that are separated by a distance L at the base. Each model cilium is again comprised of $N = 20$ segments and has torque ratio $\alpha = 3$ and $\theta_{\max} = 2\pi/5$. The torque applied at the base of the model cilia is perpendicular to both the base-to-base separation vector and the surface normal. Figure 9b shows θ_2, the angle between the surface normal and base tangent for the second filament, when the first filament switches from the fast to the slow stroke. Similar to the kind of coordination observed in Kim & Netz (2006), we find that regardless of the initial phase difference, θ_2 reaches the same value of $\theta_2 \approx -\pi/5$ for each simulation.

Expanding from the pair simulations, we now consider arrays of these cilia and investigate the effect of inter-filament spacing on coordination. We arranged N_c^2 filaments on an $N_c \times N_c$ grid at a base-to-base separation of L_1 in the plane of motion and L_2 in the orthogonal plane (see Fig. 10). Fixing $L_1 = L$, we vary the ratio L_2/L_1 and explore its effect on cilia self-organisation. Building from the coordination parameter Q of Guo *et al.* (2018), we introduce the time-dependent parameter

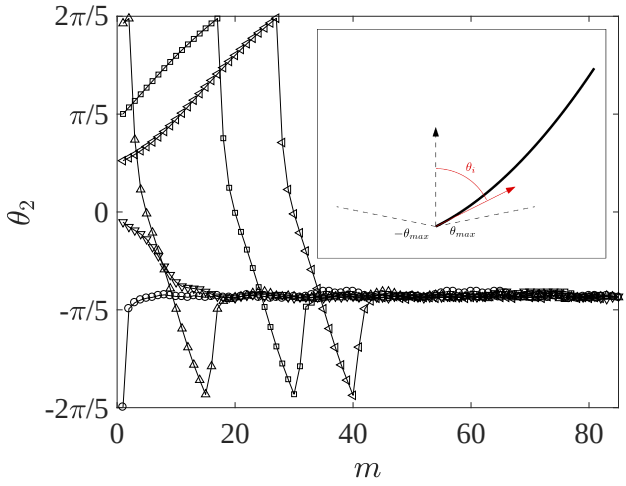
$$Q(t) = \frac{1}{t_0}\int_{t-t_0}^{t} \left(\frac{\sum_{i=1}^{N_c^2}\sum_{j=1}^{N_c^2} \alpha_i(t')\alpha_j(t') - N_c^2}{N_c^2(N_c^2-1)} \right) \mathrm{d}t' \tag{74}$$

to measure the coordination of N_c^2 cilia, where $\alpha_i(t') \in \{-1, 1\}$ is the sign of the driving torque on cilium i at time t'. We set the moving window size, t_0, to be $t_0 = 5T_c$, where T_c is the period of an isolated cilium. When $Q = 1$, there is perfect synchrony. Values of $Q(t)$ close to zero indicate a lack of coordination amongst the cilia.

In Fig. 11, we show Q over time for arrays of $N_c^2 = 36$ cilia with different values of L_2/L_1 ranging from $L_2/L_1 = 0.5$ to 1.5. Each array was initialised with all cilia in-phase and the simulations are run to a final time of approximately $120T_c$. Overall, we find that denser arrays provide a greater level of coordination, as indicated by the higher values of Q. The decrease in coordination with L_2/L_1 is also evident in Fig. 12, where we show snapshots of the array near the final time for the different cases of L_2/L_1. At the extremes of the L_2/L_1 range, $Q(t)$ quickly reaches a value about which it exhibits fairly small fluctuations. For intermediate L_2/L_1, however, we see more complex behaviour where $Q(t)$ fluctuates over a wide range with intermittent periods of coordinated and uncoordinated motion.



(a) Dynamics of a model cilium over the course of one period. The colour fades from black to white with time. Snapshots are taken at fixed time intervals and the fewer lines for the recovery stroke indicate its faster speed.



(b) The long term phase-locking of a pair of model cilia for different initial phase differences. The base angle, θ_2, of filament 2 at the times, indexed by m, the base angle of filament 1 reaches $\theta_1 = -\theta_{\max} = -2\pi/5$ and switches from recovery to effective stroke. These simulations used a torque magnitude ratio of $\alpha = 3$ and the cilia have a base-to-base distance of L.

Figure 9: Dynamics and phase-locking of model cilia.

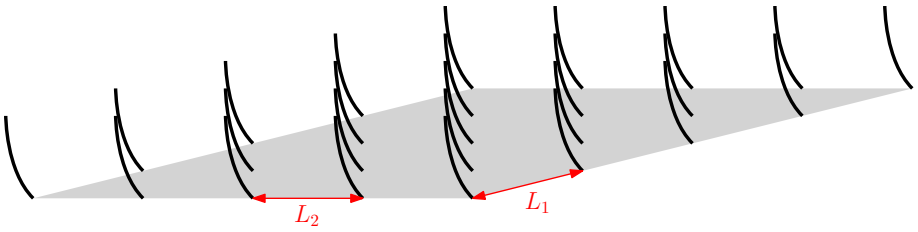


Figure 10: A diagram of a model cilia array, including the definitions of the lengths L_1 and L_2.

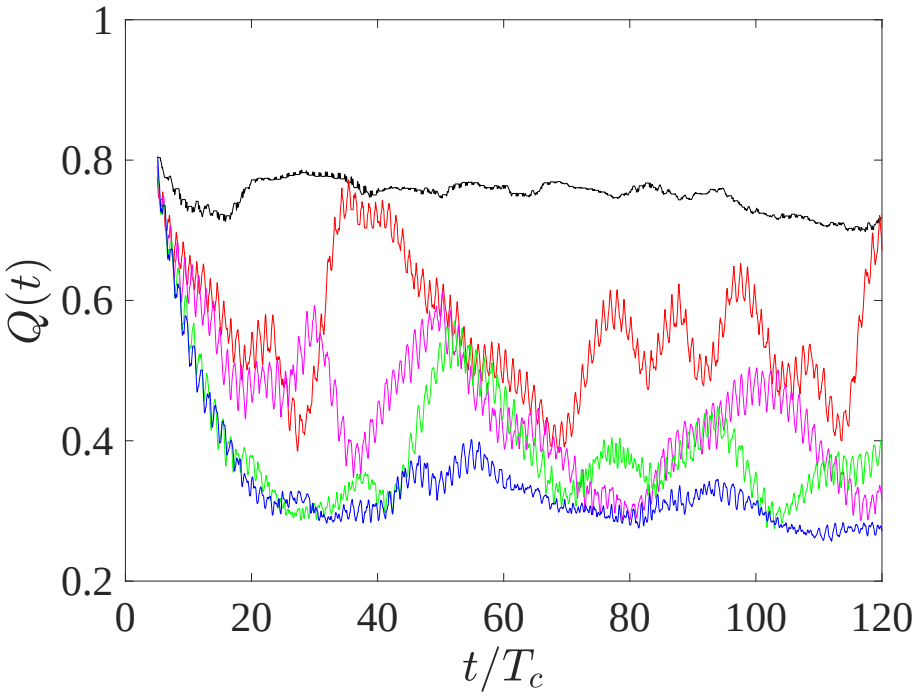
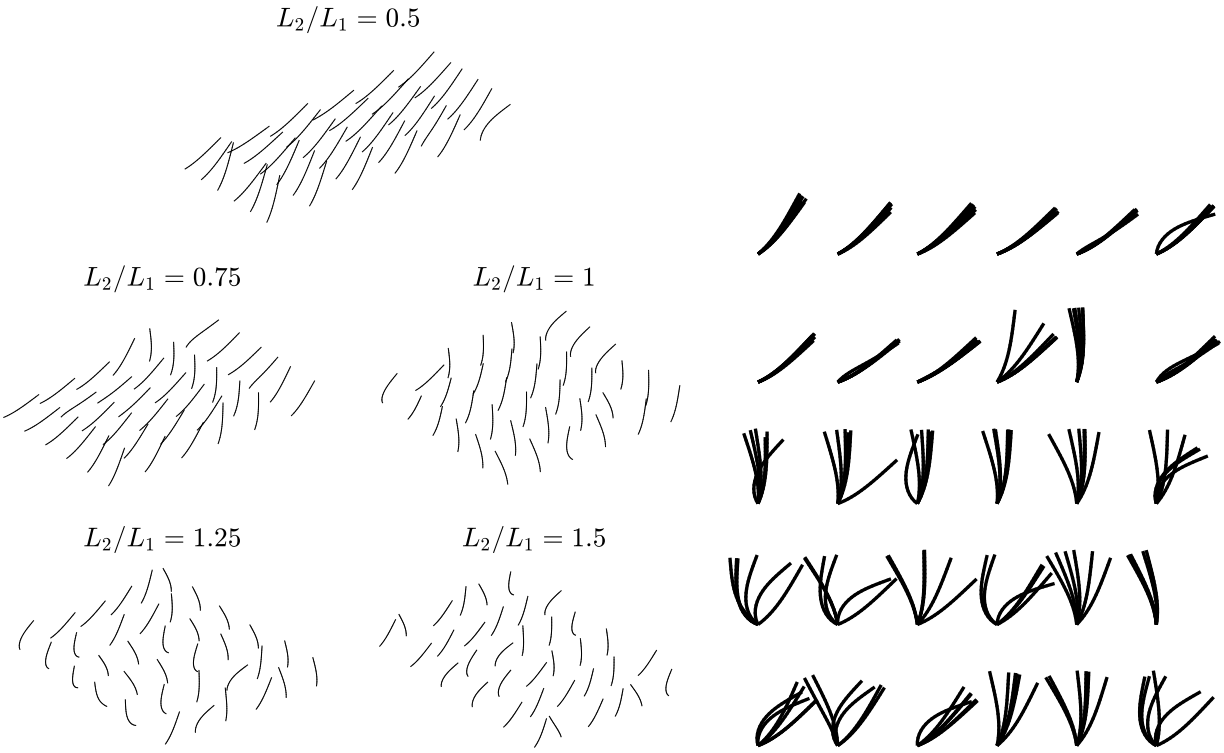


Figure 11: The coordination parameter, Q, for differently spaced cilia arrays over time. We fix the distance $L_1 = L$ and vary L_2 to examine the behaviour of arrays with $L_2/L_1 = 0.5$ (——), 0.75 (——), 1 (——), 1.25 (——) and 1.5 (——).



(a) Cilia array snapshots illustrating different levels of co-ordination.

(b) A side view of the arrays in Fig. 12a. L_2/L_1 increases from top to bottom.

Figure 12: Cilia array simulations at $t/\Delta t = 6.2 \times 10^4$.

5.3 Bidisperse suspension of undulatory swimmers

In this section, we employ the 2D version of our computational model to simulate a suspension of slender undulatory swimmers moving in the xy-plane. The suspension consists of two populations (indexed by $p = 1$ and $p = 2$) that have different swimming gaits. Similar simulations (Agrawal & Babu, 2018) have been performed using multiparticle collision dynamics for two swimmer populations with different frequencies, but the same waveform.

All swimmers have length L and are formed of $N = 30$ segments. The motion of the swimmers is driven by a time-dependent preferred curvature,

$$\kappa_\nu^{(p)}(s,t) = K_0 \sin\left(\frac{2\pi k^{(p)}}{L}s - \omega t + \phi\right), \tag{75}$$

in Eq. (5), where K_0 is the amplitude, $k^{(p)}$ is the wavenumber for population p, $\omega = 2\pi/T$ is the undulation frequency with T as the undulation period, and ϕ is the phase. The two different swimming gaits are prescribed by specifying different $k^{(1)}$ and $k^{(2)}$.

In our simulations, the curvature amplitude is $K_0 = 10.61/L$, while the undulation frequency is set such that the ratio of viscous to elastic forces is $(4\pi\omega\eta/K_B)^{1/4}L = 10$. The phase, ϕ, for each swimmer is drawn randomly from a uniform distribution. To achieve sufficiently different waveforms, we set $k^{(1)} = 1$ and $k^{(2)} = 3$. Figure 13 shows waveforms for isolated $p = 1$ and $p = 2$ swimmers with these parameters. With the swimming speeds $V_1/(L\omega) = 0.01$ and $V_2/(L\omega) = 0.0024$, the swimmers from population $p = 2$ travel at less than a quarter of the speed of those from $p = 1$.

In our simulations, we consider a suspension of 100 swimmers with 50 in population $p = 1$ and 50 in $p = 2$. The swimmers are all initially straight and distributed uniformly and isotropically in the centre-plane of a periodic domain of size $4.7L \times 4.7L \times 0.29L$. FCM is used to resolve the hydrodynamic interactions with the fluid flow being solved on a $1024 \times 1024 \times 64$ grid. Steric interactions are implemented as described in Section 2.6.

The configuration of the suspension at three times is shown in Fig. 14. After 150 undulation periods,

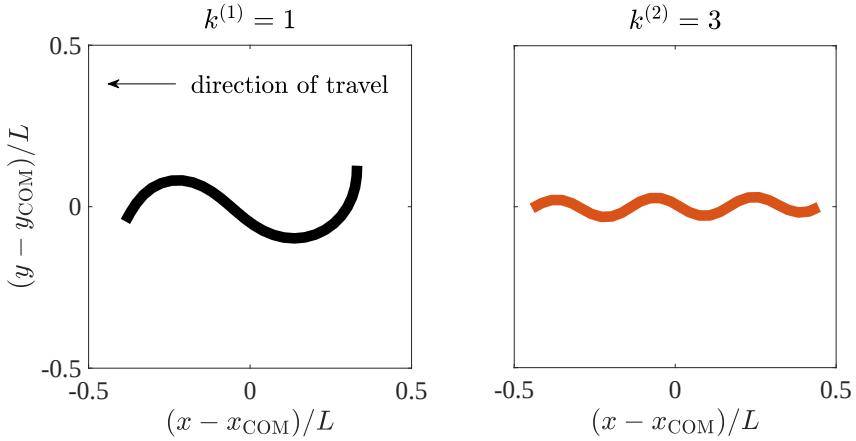


Figure 13: Swimmer shapes for the two different wavenumbers. Figures are centred at the filament centre of mass $(x_{\mathrm{COM}}, y_{\mathrm{COM}})$ and the axes are normalised by the swimmer length, L.

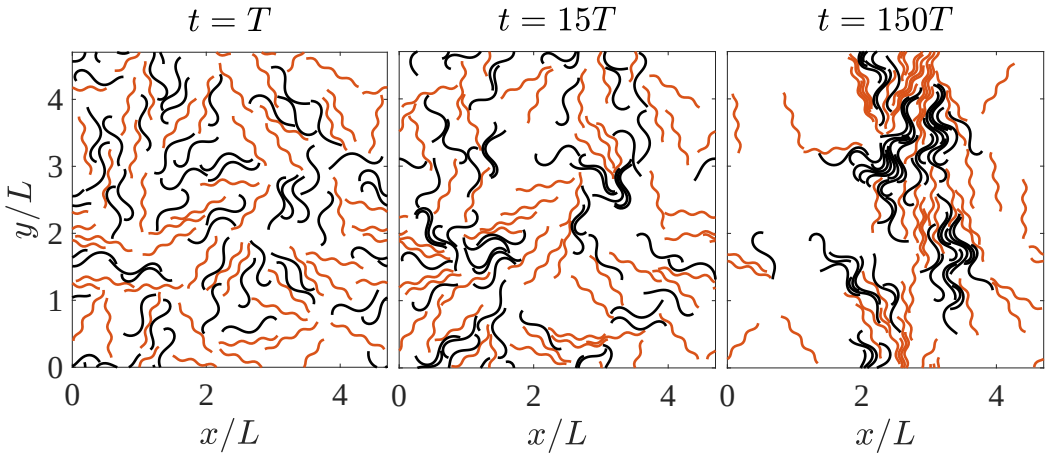


Figure 14: Snapshots at three different times of 100 active filaments in a periodic domain of size $4.7L \times 4.7L \times 0.29L$. Filaments with $k^{(1)} = 1$ are coloured black, while those with $k^{(2)} = 3$ are coloured red, as in Fig. 13. Filaments are initially straight and randomly oriented.

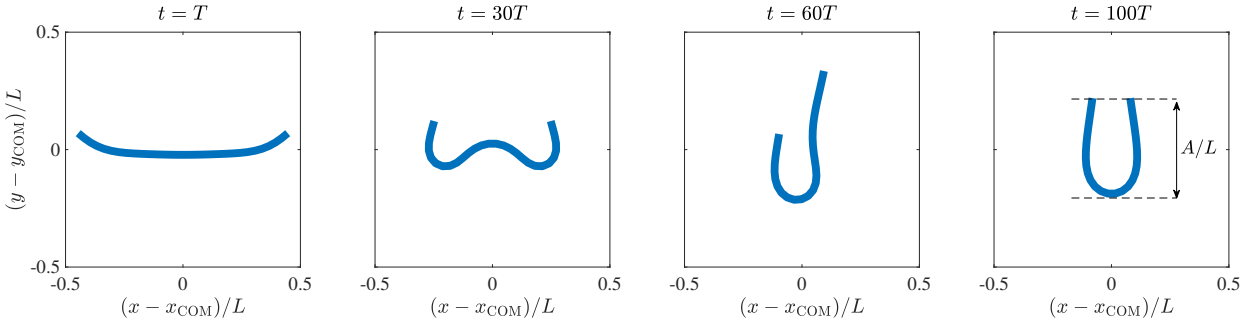


Figure 15: A filament with $N = 31$ and $B = 10^4$, is placed horizontally and allowed to sediment vertically under a constant force per unit length. Upon reaching its final steady shape, the filament's normal deflection, A, is measured. Figures are centred at the filament centre of mass $(x_{\mathrm{COM}}, y_{\mathrm{COM}})$ and are normalised by the length of the filament, L.

we see that all swimmers have clustered, with swimmers tending to cluster with those from their own population, i.e. with a similar waveform. Clustering is also observed in planar monodisperse suspensions of model sperm cells (Schoeller & Keaveny, 2018; Yang *et al.*, 2008) and slender undulatory swimmers (Yang *et al.*, 2010), as well as in planar bidisperse suspensions (Agrawal & Babu, 2018) of undulatory swimmers. We first observe swimmers from $p = 1$ forming clusters with each other, while the slower swimmers from $p = 2$ remain dispersed in the surrounding fluid. Once $p = 1$ clusters are established, we begin to see $p = 2$ swimmers joining these clusters, but tending to remain close to other $p = 2$ swimmers. Interestingly, filament softness has contributed to the development of these slightly mixed clusters. We sometimes observed high-wavenumber $p = 2$ swimmers deforming to match the shape of neighbouring low-wavenumber $p = 1$ swimmers.

5.4 Sedimentation

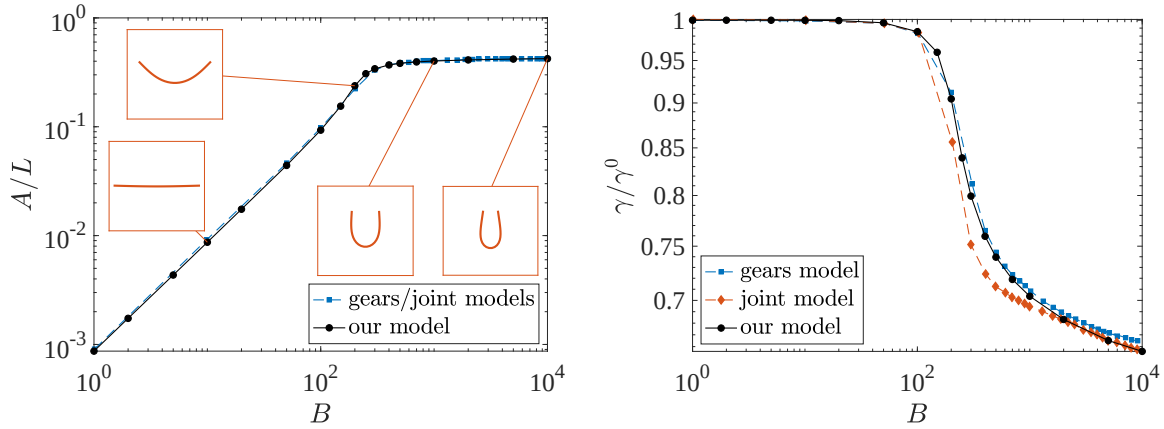
In this section, we use our methodology to study filament sedimentation. We both revisit the case of a single filament falling under gravity, as well as explore cases of multiple filaments, ranging from small ensembles to clouds and suspensions. We address these situations in cases where filament motion is restricted to a plane, as well as those where their motion is completely three-dimensional.

5.4.1 Settling filament

We first revisit the case of a single filament settling under gravity and measure its deformation and effective drag coefficient. This problem, first investigated numerically by Cosentino Lagomarsino *et al.* (2005) and more recently in experiments by Marchetti *et al.* (2018), has also been used previously as a test problem for filament models (Keaveny, 2008; Delmotte *et al.*, 2015).

In our simulations, we consider an initially straight and horizontal filament formed of $N = 31$ segments in an unbounded fluid with the hydrodynamic interactions resolved using the RPY tensor with hydrodynamic radius, a. As in Section 3.3, a constant force per unit length of magnitude W is applied vertically to the filament. Using the parameters arising in the problem, we can define the settling time as $T = \eta L/W$, as well as the elasto-gravitational number $B = L^3W/K_B$. Running the simulation with timestep size $\Delta t = T/30$, the filament is allowed to sediment until it reaches a steady-state shape and steady-state settling speed, V_s. The filament at various points in time is shown in Fig. 15 for the case where $B = 10^4$. Once the steady state is reached, we measure the normal deflection A, defined as the distance between the highest and lowest points of the filament (see Fig. 15), and the effective drag coefficient γ/γ^0, where $\gamma = WL/V_s$ and γ^0 is the value of γ in the limit $B \to 0$ corresponding to a straight filament.

The values of A/L and γ/γ^0 as a function of B are presented in Fig. 16. We compare our results with those presented in Delmotte *et al.* (2015) for both their gears model, where filaments are formed of 34 particles with no separation between the particles, and a joint model, formed of 31 particles with gaps of $0.2a$ between the particles. As we have $\Delta L = 2.2a$, our spacing is equivalent to that used in the



(a) Normalised deflection, A/L, as a function of the elasto-gravitational number, B. The final filament shapes at certain values of B are shown in the inset plots.

(b) Normalised effective drag coefficient γ/γ^0, as a function of the dimensionless elasto-gravitational number, B. Here, γ^0 is the drag coefficient for a rigid rod (i.e. at $B \to 0$).

Figure 16: Normal deflection and effective drag coefficients for a single sedimenting filament as a function of B. Results are compared to the gears and joint model results from Delmotte *et al.* (2015). Simulations were performed with a filament formed of $N = 31$ segments in an infinite domain.

joint model. In all cases, the RPY mobility matrices are used to capture hydrodynamics interactions.

All models show excellent agreement in their respective measurements of normal deflection, A/L. In all cases, the models predict the same initial linear increase of A/L with B, followed by a plateau to the same value at high B. For the drag coefficient, γ/γ^0, the differences, though still small, are more pronounced. In particular, our measurements find good agreement with the gears model for $B \lesssim 10^3$ as the sudden decay in γ/γ^0 given by both models coincide. For higher B, however, we observe better agreement with the joint model, which yield a similar decay with increasing B. We suspect that the agreement with the gears model for lower B is due to the better resolution of the bending moments given by these models. At higher B, we attribute the agreement between our results and those given by the joint model to the similar viscous force on the filament that results from setting $\Delta L = 2.2a$.

5.4.2 Sedimentation of large 2D suspensions

Here, we investigate the sedimentation of a large, planar suspension of filaments in a periodic domain. This set of simulations was inspired by Gustavsson & Tornberg (2009); Tornberg & Gustavsson (2006), where a similar simulation was also performed in a periodic domain, but for a suspension of rigid filaments that can move in all three dimensions. We note as they do, that sedimentation in periodic domains, while a canonical problem, is quite different from that in unbounded or bounded systems and can exhibit dependencies on domain sizes. Nevertheless, these simulations do provide a context with which to explore the effects of filament flexibility on sedimentation.

The simulations presented here are larger versions of the test problem described in Section 3.3. We also run these simulations for longer times. Specifically, we consider monolayers of 100, 500, and 1000 filaments, each formed of $N = 15$ segments. The dimensionless elasto-gravitational number is set at two values, $B = 1$ and 1000, to examine the effect of deformation on suspension dynamics as the filaments settle for as long as $500T$. The timestep is chosen such that $\Delta t = T/300$. The filaments are initially straight and are distributed uniformly and isotropically in the centre-plane of a periodic domain of size $18.8L \times 18.8L \times 1.2L$. FCM is used to resolve the hydrodynamic interactions between the filaments, with the Stokes equations being solved on a grid of $2048 \times 2048 \times 128$ points. Steric interactions are implemented as in Section 2.6.

Snapshots of the sedimenting suspensions for $M = 1000$ with $B = 1$ and $B = 1000$ are shown in Fig. 17. In general, we find that both stiff ($B = 1$) and flexible ($B = 1000$) filaments over time form a single large cluster surrounded by isolated filaments that are found to tumble, but are most often

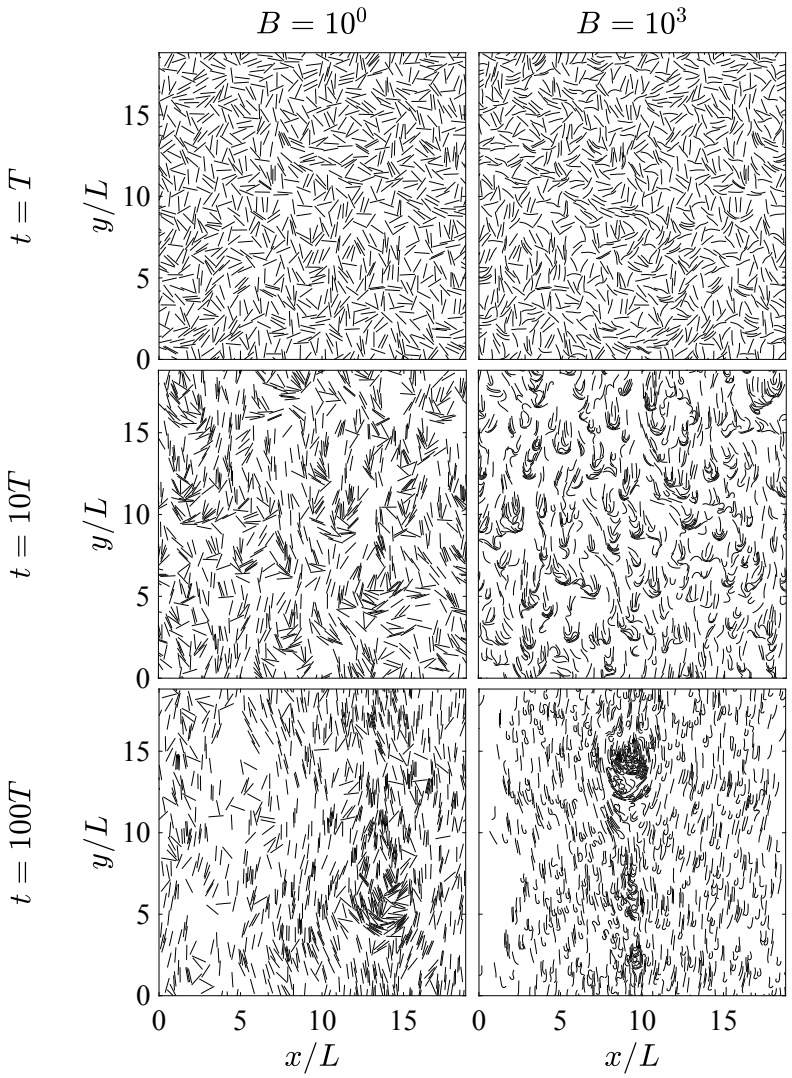


Figure 17: Snapshots of 1000 filaments sedimenting in a periodic domain of size $18.8L \times 18.8L \times 1.2L$, at three different times for two different values of B. Filaments are initially straight and randomly oriented.

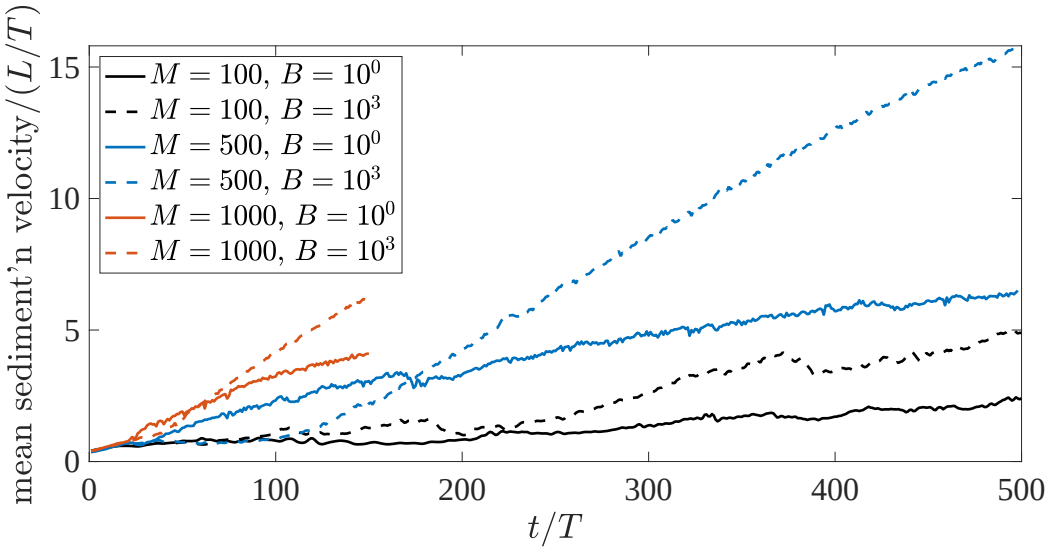


Figure 18: Mean sedimentation velocity, $\overline{w}$, of a suspension of M ($N = 15$) filaments in a periodic domain of size $18.8L \times 18.8L \times 1.2L$. Simulations performed for two values of B.

oriented in the direction of gravity. We do see, however, that for the flexible filaments, the cluster formed is of much higher density, and the filaments surrounding it can buckle and flex, as well as tumble, due to the shear induced by the falling cluster.

Following Gustavsson & Tornberg (2009), for each of the simulations that we have performed, we compute the mean sedimentation velocity,

$$\overline{w}(t) = \sum_{m=1}^{M} w_m(t), \tag{76}$$

where $w_m(t)$ is the vertical centre-of-mass velocity of filament m. The resulting values over time are shown in Fig. 18. We observe that for each concentration, the settling velocity and its growth rate increase with filament flexibility (higher B). We suspect that this is a result of two factors. First, as we observed in Section 5.4.1, flexibility leads to a reduction in the filaments' effective drag coefficient. Second, we also see in Fig. 17 that deformable filaments form denser clusters. This increase in local filament density in turn enhances interactions, yielding higher sedimentation velocities.

In previous work (Mackaplow & Shaqfeh, 1998; Gustavsson & Tornberg, 2009) on rigid fibre sedimentation in periodic domains, the onset of a plateau in the mean settling velocity coincided with the formation of a single cluster in the centre of the domain that is surrounded by an upwelling of clear fluid. Additionally, observed fluctuations about the plateau value corresponded to cluster break-up and reformation events that occur periodically. In our case, we find that even after $500T$, the mean settling velocity continues grow with time. For each case of M, the setting velocity growth rate is higher for the more flexible filaments, but perhaps more surprising is that the growth rate appears to be independent of M. Indeed, increasing the filament density by setting $M = 1000$, we see that over the $150T$ that for both B, the mean sedimentation velocity parallels that for $M = 500$. The continued growth may be attributed to the observation that there many filaments have yet to join the cluster. Additionally, we suspect that the continued growth in sedimentation velocity as compared to the fully three-dimensional cases (Mackaplow & Shaqfeh, 1998; Gustavsson & Tornberg, 2009) is due to both the larger lateral distances in our simulation that the filaments must traverse, as well as their restriction to move only in two dimensions. We suspect also that the reduced out-of-plane thickness enhances the hydrodynamic interactions between the filaments, allowing for the higher growth rates in our simulations.

5.4.3 Tumbling, settling and buckling of a filament square

Gustavsson & Tornberg (2009) have shown that rigid fibres evenly distributed along a circle can periodically tumble as they settle. Similar periodic orbits have been seen both numerically (Claeys & Brady,

1993) and experimentally (Jung *et al.*, 2006) for ellipsoids and other rigid structures released from an initially symmetric configuration. Here, we explore how the introduction of filament flexibility can affect these dynamics.

In our simulations, we consider four filaments whose centres coincide with the corners of a horizontal square of side length $D/L = 0.268$. The filaments are initially aligned vertically in the direction of gravity. We discretise the filaments into $N = 30$ segments and, to allow for an unbounded fluid, we resolve their hydrodynamic interactions using the RPY mobility matrix. Steric interactions are implemented as in Section 2.6. The timestep is chosen such that $\Delta t = T/300$. The elasto-gravitational number, B, is varied from $B = 10^{-1}$ (very stiff filaments) to $B = 10^4$ (very soft filaments). The long-term behaviour reveals significant differences between these cases, including symmetry breaking (Fig. 19). We note that similar observations have been made previously Llopis *et al.* (2007); Saggiorato *et al.* (2015).

At the lowest value of $B = 10^{-1}$, we observe periodic tumbling (Fig. 19a) where the filaments first separate and then reapproach when they return to their initial configuration. This matches the periodic orbits seen in Gustavsson & Tornberg (2009); Jung *et al.* (2006). For $B = 1$ and $B = 10$, while we again observe the filaments returning very close to their initial configuration, we see that the time it takes to return to this configuration decreases with B, as does the distance between the filaments. For the case $B = 10^2$, when returning after tumbling, the filaments, whose deformation is now apparent, approach each other so closely that they collide and experience steric interactions (Fig. 19b). At this point, the filaments cease tumbling and instead begin to settle as a group, rearranging their relative positions as they move downwards. At $B = 10^3$, we see that filaments do not exhibit any tumbling and instead adopt a horseshoe-like shape and a steady configuration as they settle, see Fig. 19c. Finally, for $B = 10^4$, while we see that at early times the dynamics are similar to those for $B = 10^3$, we observe also that this state is unstable. The accumulation of round-off errors are sufficient to trigger the instability (Fig. 19d) and the onset of more complex filament behaviour.

5.4.4 Sedimentation of flexible filament clouds

As a final demonstration of our methodology, we consider the dynamics of a cloud of flexible filaments settling under gravity. For clouds of rigid, spherical particles (Nitsche & Batchelor, 1997; Metzger *et al.*, 2007), as the cloud settles, the particles within the cloud move along trajectories reminiscent of the streamlines within a settling spherical drop (Batchelor, 1967). As the cloud settles, it sheds particles at the cloud axis, until it becomes a torus, leading to the cloud dividing. While similar studies have been performed for clouds of rigid fibres (Park *et al.*, 2010; Nazockdast *et al.*, 2017), showing how particle anisotropy accelerates cloud break-up, the role of flexibility, to the best of our knowledge, has not been explored.

In our fully three-dimensional cloud simulations, the filaments are distributed uniformly and isotropically in cubes (Figs. 20 and 21) of side length $2.5L$ and $10L$ using the initialisation method describe in Appendix A.5. The filaments are discretised into $N = 20$ segments and the timestep size is $\Delta t = T/200$. The hydrodynamic interactions in the unbounded domain are resolved using the RPY motility matrix and steric interactions are implemented as in Section 2.6. To highlight the effect of filament flexibility, simulations were performed with elasto-gravitational numbers $B = 1$ and $B = 10^2$.

Fig. 20 shows snapshots from the cloud simulations with $M = 120$ filaments for both $B = 1$ and $B = 10^2$. The initial conditions for these simulations are identical. At short times, we observe that both clouds exhibit very similar behaviour. Later in the simulations, however, marked differences arise. The cloud of stiff filaments ($B = 1$) splits into two clusters as a result of filament depletion at its centre. This behaviour matches results from Park *et al.* (2010). The cloud consisting of more flexible filaments ($B = 10^2$) breaks up later, and in a much more complicated way. We see that it disintegrates into several clusters, rather than splitting in two. We suspect that this could be connected to individual filaments' orbits within clouds, as well as their sedimentation velocities. We have also performed a larger, though less dense, cloud simulation with $M = 360$ filaments and $B = 10^2$. As shown in Fig. 21, we find that the features of the cloud's initial shape persist as the cloud overturns and only starts to disappear once the cloud begins to break up.

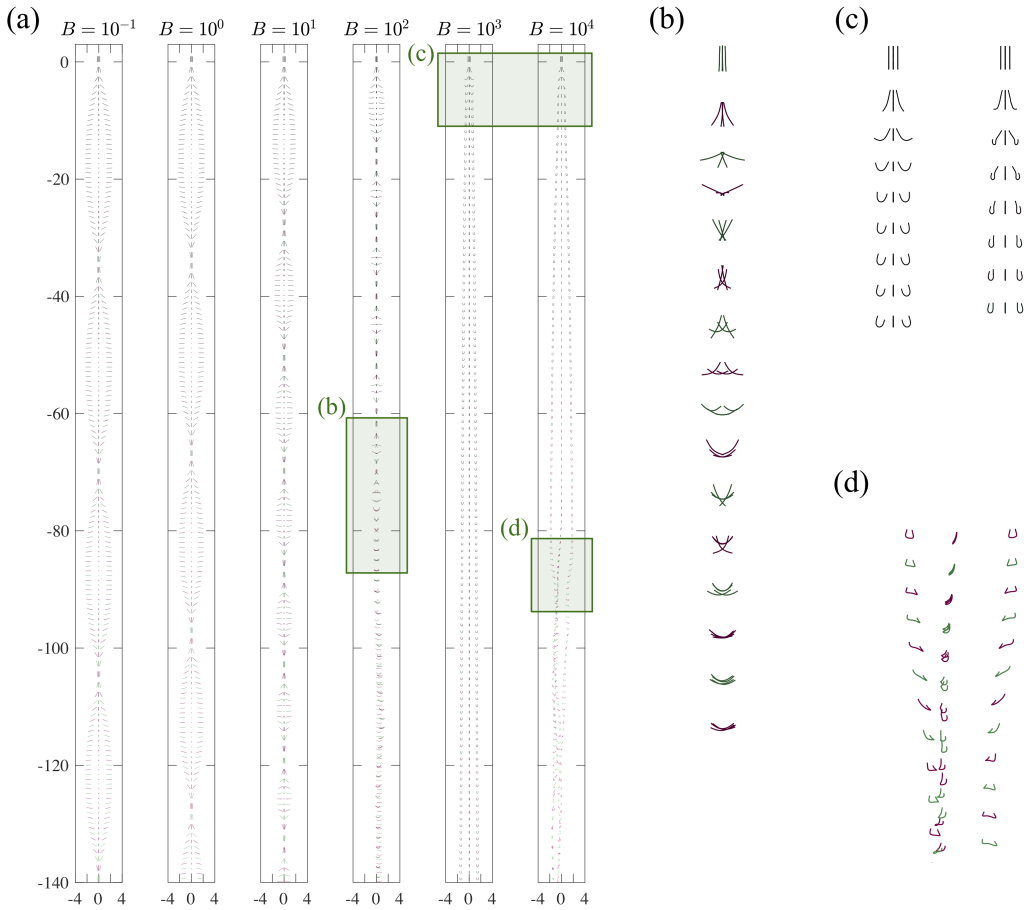


Figure 19: Settling dynamics of four, initially vertical filaments for different values of B. The filaments are shown at every $2T$, with the colours alternating between green and magenta. (a) Stiffer filaments ($B \ll 10^2$) tumble periodically. Though bending is not appreciable for these cases, the tumbling period decreases as B increases. Softer filaments ($B = 10^3$) deform considerably and do not tumble, but may buckle if $B \gg 10^3$. (b) For $B = 10^2$, filaments interact through steric forces, leading to symmetry breaking. (c) For $B = 10^3$ (left) filaments immediately bend at their centres, while for $B = 10^4$ (right), the filaments first buckle at their lower end and this bend propagates to their centres as they fall. (d) For $B = 10^4$, the configuration achieved after the filaments deform is unstable. As perturbations grow, symmetry is broken and more complex trajectories emerge.

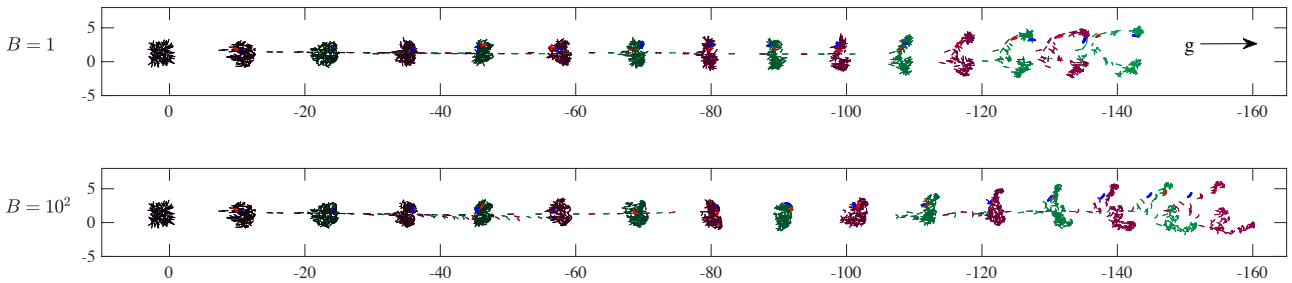


Figure 20: Sedimentation and break-up of a dense, three-dimensional cloud of $M = 120$ filaments with $B = 1$ and $B = 10^2$. Filament positions are initially in a cube of linear size $2.5L$. The cloud is shown every $3T$ with the colour alternating between magenta and green and fading with time. Two filaments are highlighted (red and blue). The arrow (g) indicates the direction of gravity.

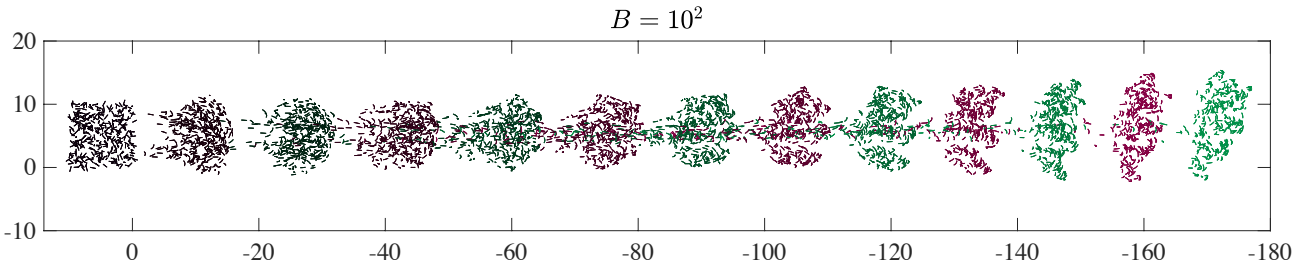


Figure 21: Sedimentation of a three-dimensional cloud of $M = 360$ filaments with $B = 10^2$. Initially, all filaments have at least one end in a cube with side $10L$. The cloud is shown every $4T$ with the colour alternating between magenta and green and fading with time.

6 Discussion and conclusions

In this paper, we have presented a scalable, efficient and flexible method for performing large-scale simulations of passive and active filaments and their suspensions. In particular, we have shown how to describe the fully 3D filament deformations due to bending and twisting through the use of quaternions, and in doing so, demonstrated how quasi-Newton methods coupled with an implicit, geometric multi-step scheme can be used to overcome the numerical stiffness associated with deformation while ensuring constraints on filament motion are satisfied. Our framework can be easily integrated with many existing models and methods for resolving hydrodynamic interactions. In addition, the second-order spatial discretisation could be substituted for a higher-order finite difference method on the equispaced grid, though new expressions for the approximate Jacobian would need to be established given the change in stencil. Our usage of iterative schemes for the nonlinear systems that arise ensures that efficient, matrix-free methods for the hydrodynamics are also applicable, which we have also demonstrated. We have shown that our method can be used to study many relevant problems in biofluid dynamics, as well as suspension mechanics, potentially at scales previously inaccessible. We encourage the reader to experiment with the MATLAB/Octave implementation (Schoeller *et al.*, 2019) of the method to study similar problems or expand upon it to explore new directions.

While we have developed and explored the method exclusively in the limit where fluid inertia is negligible, it would be of interest to understand how our framework could be adapted for use with fluid solvers where inertia is included (Wiens & Stockie, 2015), to study, for example, filament suspensions in turbulent flows (Rosti *et al.*, 2018). Additionally, it could be of interest to explore filament motion in non-Newtonian fluids, such as viscoelastic fluids, especially with respect to modelling flagellar motion and cell swimming (Teran *et al.*, 2010). Our method could be used in the context where the filaments are components in more complex physical systems, such as networks, or, by incorporating Brownian motion (Keaveny, 2014; Delong *et al.*, 2015), suspended polymers in solution, or entangled polymers in melts.

Acknowledgements

EEK and AKT gratefully acknowledge support from EPSRC Grant EP/P013651/1. SFS gratefully acknowledges funding by an Imperial College President's PhD scholarship. TAW is thankful for funding through an EPSRC Studentship (Ref: 1832024). All authors also thank Smitha Maretvadakethope for a detailed reading of the manuscript.

A Quaternions

A.1 Quaternion initialisation

In this section we detail how to initialise quaternion encoding the rotation from the standard basis $\{\widehat{\boldsymbol{e}}_x, \widehat{\boldsymbol{e}}_y, \widehat{\boldsymbol{e}}_z\}$ to the material frame $\{\widehat{\boldsymbol{t}}, \widehat{\boldsymbol{\mu}}, \widehat{\boldsymbol{\nu}}\}$. Since $\widehat{\boldsymbol{\nu}} = \widehat{\boldsymbol{t}} \times \widehat{\boldsymbol{\mu}}$, we need only provide initial conditions for

the tangent and one of normal vectors. The main idea is to determine the quaternion which rotates a given unit vector to another and then apply this twice in succession by first mapping $\widehat{\boldsymbol{e}}_x$ to $\widehat{\boldsymbol{t}}$ and then mapping the image of $\widehat{\boldsymbol{e}}_y$ under this first rotation to $\widehat{\boldsymbol{\mu}}$.

Let $\widehat{\boldsymbol{a}}$ and $\widehat{\boldsymbol{b}}$ be unit vectors. Recall that $\widehat{\boldsymbol{a}} \cdot \widehat{\boldsymbol{b}} = \cos\theta$, $\widehat{\boldsymbol{a}} \times \widehat{\boldsymbol{b}} = \sin\theta\, \widehat{\boldsymbol{u}}$, where $\widehat{\boldsymbol{u}}$ is normal to the plane spanned by $\widehat{\boldsymbol{a}}$ and $\widehat{\boldsymbol{b}}$, and θ is the angle between the two vectors as measured anticlockwise about $\widehat{\boldsymbol{u}}$. Recalling Eq. (44), we see that the quaternion $(\widehat{\boldsymbol{a}} \cdot \widehat{\boldsymbol{b}},\ \widehat{\boldsymbol{a}} \times \widehat{\boldsymbol{b}})$ represents the rotation of 2θ about the vector $\widehat{\boldsymbol{u}}$. Since we need only rotate $\widehat{\boldsymbol{a}}$ by θ to align with $\widehat{\boldsymbol{b}}$, we seek the quaternion $\boldsymbol{q}$ for which $\boldsymbol{q}^2 = \boldsymbol{q} \bullet \boldsymbol{q} = (\widehat{\boldsymbol{a}} \cdot \widehat{\boldsymbol{b}},\ \widehat{\boldsymbol{a}} \times \widehat{\boldsymbol{b}})$. By expanding and rearranging $\boldsymbol{q}^2 = \boldsymbol{p}$ we find that a square root of a quaternion may be defined as

$$(\cdot)^{1/2} : (p_0, \boldsymbol{p}) \mapsto \begin{cases} \left(\sqrt{\frac{p_0+1}{2}}, \frac{\boldsymbol{p}}{\sqrt{2(p_0+1)}}\right) & p_0 \neq -1, \\ (0,0,0,1) & p_0 = -1. \end{cases} \tag{77}$$

It is readily checked that for all $\boldsymbol{p} \in \mathbb{U}$: (a) $\boldsymbol{p}^{1/2} \in \mathbb{U}$, and (b) $\boldsymbol{p}^{1/2} \bullet \boldsymbol{p}^{1/2} = \boldsymbol{p}$. With this defined, we produce the quaternion initial conditions as follows:

1. Let $\widehat{\boldsymbol{t}} = (\widehat{t}_1, \widehat{t}_2, \widehat{t}_3)$ and $\widehat{\boldsymbol{\mu}} = (\widehat{\mu}_1, \widehat{\mu}_2, \widehat{\mu}_3)$ be the initial tangent and normal vectors to the filament centreline.
2. Evaluate the quaternion which maps $\widehat{\boldsymbol{e}}_x \mapsto \widehat{\boldsymbol{t}}$ as $\boldsymbol{p}_1 = (\widehat{\boldsymbol{e}}_x \cdot \widehat{\boldsymbol{t}},\ \widehat{\boldsymbol{e}}_x \times \widehat{\boldsymbol{t}})^{1/2} = (\widehat{t}_1, 0, -\widehat{t}_3, \widehat{t}_2)^{1/2}$.
3. Let $\widehat{\boldsymbol{n}}' = \boldsymbol{R}(\boldsymbol{p}_1)\widehat{\boldsymbol{e}}_y$ be the image of the y-axis direction under this first rotation.
4. Evaluate the rotation mapping $\widehat{\boldsymbol{n}}' \mapsto \widehat{\boldsymbol{\mu}}$ as $\boldsymbol{p}_2 = (\widehat{\boldsymbol{n}}' \cdot \widehat{\boldsymbol{\mu}},\ \widehat{\boldsymbol{n}}' \times \widehat{\boldsymbol{\mu}})^{1/2}$
5. Assign the quaternion an initial value of $\boldsymbol{p}_2 \bullet \boldsymbol{p}_1$.

A.2 Internal moments in terms of quaternions

In this section, we derive an expression for the internal moments, Eq. (5), expressed solely in terms of the unit orientation quaternion. Recall Eq. (5):

$$\boldsymbol{M} = K_B \left(\widehat{\boldsymbol{\mu}} \left(\widehat{\boldsymbol{t}} \cdot \frac{\partial \widehat{\boldsymbol{\nu}}}{\partial s} - \kappa_\mu \right) + \widehat{\boldsymbol{\nu}} \left(\widehat{\boldsymbol{\mu}} \cdot \frac{\partial \widehat{\boldsymbol{t}}}{\partial s} - \kappa_\nu \right) \right) + K_T \widehat{\boldsymbol{t}} \left(\widehat{\boldsymbol{\nu}} \cdot \frac{\partial \widehat{\boldsymbol{\mu}}}{\partial s} - \gamma_0 \right). \tag{78}$$

Since $\widehat{\boldsymbol{t}} = \boldsymbol{R}(\boldsymbol{q})\widehat{\boldsymbol{e}}_x$, $\widehat{\boldsymbol{\mu}} = \boldsymbol{R}(\boldsymbol{q})\widehat{\boldsymbol{e}}_y$, $\widehat{\boldsymbol{\nu}} = \boldsymbol{R}(\boldsymbol{q})\widehat{\boldsymbol{e}}_z$, we have that

$$\boldsymbol{M} = \boldsymbol{R}\left(\boldsymbol{q}\right) \begin{pmatrix} K_T(\widehat{\boldsymbol{\nu}} \cdot \partial\widehat{\boldsymbol{\mu}}/\partial s - \gamma_0) \\ K_B(\widehat{\boldsymbol{t}} \cdot \partial\widehat{\boldsymbol{\nu}}/\partial s - \kappa_\mu) \\ K_B(\widehat{\boldsymbol{\mu}} \cdot \partial\widehat{\boldsymbol{t}}/\partial s - \kappa_\nu) \end{pmatrix}. \tag{79}$$

Next, considering $\widehat{\boldsymbol{t}}(s,t)$, a function of arclength s and time t, we observe that

$$\left(0, \frac{\partial \widehat{\boldsymbol{t}}}{\partial s}\right) = \frac{\partial}{\partial s}\left(0, \widehat{\boldsymbol{t}}\right) \tag{80a}$$

$$= \frac{\partial}{\partial s}\left[\boldsymbol{q} \bullet (0, \widehat{\boldsymbol{e}}_x) \bullet \boldsymbol{q}^*\right] \tag{80b}$$

$$= \frac{\partial \boldsymbol{q}}{\partial s} \bullet \boldsymbol{q}^* \bullet \left(0, \widehat{\boldsymbol{t}}\right) - \left(0, \widehat{\boldsymbol{t}}\right) \bullet \frac{\partial \boldsymbol{q}}{\partial s} \bullet \boldsymbol{q}^*, \tag{80c}$$

where we have used $\boldsymbol{q} \bullet \partial\boldsymbol{q}^*/\partial s = -\partial\boldsymbol{q}/\partial s \bullet \boldsymbol{q}^*$, a consequence of $\boldsymbol{q} \bullet \boldsymbol{q}^* = \boldsymbol{q}^* \bullet \boldsymbol{q} = \boldsymbol{I}_q$. Now, since $\|\boldsymbol{q}\| = 1$, we have that $\partial\boldsymbol{q}/\partial s \bullet \boldsymbol{q}^*$ is a pure quaternion (i.e. one with zero scalar part) and thus may be unambiguously identified with its vector part $[\partial\boldsymbol{q}/\partial s \bullet \boldsymbol{q}^*]_{\mathbb{R}^3}$. Observing that for pure quaternions we have $(0,\boldsymbol{v}) \bullet (0,\boldsymbol{w}) - (0,\boldsymbol{w}) \bullet (0,\boldsymbol{v}) = (0, 2\boldsymbol{v} \times \boldsymbol{w})$, the above yields

$$\frac{\partial \widehat{\boldsymbol{t}}}{\partial s} = 2\left[\frac{\partial \boldsymbol{q}}{\partial s} \bullet \boldsymbol{q}^*\right]_{\mathbb{R}^3} \times \widehat{\boldsymbol{t}}. \tag{81}$$

Analogous expressions hold for $\widehat{\boldsymbol{\mu}}$ and $\widehat{\boldsymbol{\nu}}$. Using the invariance under circular shift of the scalar triple product, we find that

$$\widehat{\boldsymbol{\mu}} \cdot \frac{\partial \widehat{\boldsymbol{t}}}{\partial s} = 2 \left[\frac{\partial \boldsymbol{q}}{\partial s} \bullet \boldsymbol{q}^* \right]_{\mathbb{R}^3} \cdot \widehat{\boldsymbol{\nu}}, \tag{82}$$

$$\widehat{\boldsymbol{\nu}} \cdot \frac{\partial \widehat{\boldsymbol{\mu}}}{\partial s} = 2 \left[\frac{\partial \boldsymbol{q}}{\partial s} \bullet \boldsymbol{q}^* \right]_{\mathbb{R}^3} \cdot \widehat{\boldsymbol{t}}, \tag{83}$$

$$\widehat{\boldsymbol{t}} \cdot \frac{\partial \widehat{\boldsymbol{\nu}}}{\partial s} = 2 \left[\frac{\partial \boldsymbol{q}}{\partial s} \bullet \boldsymbol{q}^* \right]_{\mathbb{R}^3} \cdot \widehat{\boldsymbol{\mu}}. \tag{84}$$

Thus, we simplify,

$$2 \frac{\partial \boldsymbol{q}}{\partial s} \bullet \boldsymbol{q}^* = \left(0, \left(\widehat{\boldsymbol{\nu}} \cdot \frac{\partial \widehat{\boldsymbol{\mu}}}{\partial s} \right) \widehat{\boldsymbol{t}} + \left(\widehat{\boldsymbol{t}} \cdot \frac{\partial \widehat{\boldsymbol{\nu}}}{\partial s} \right) \widehat{\boldsymbol{\mu}} + \left(\widehat{\boldsymbol{\mu}} \cdot \frac{\partial \widehat{\boldsymbol{t}}}{\partial s} \right) \widehat{\boldsymbol{\nu}} \right) \tag{85a}$$

$$= \boldsymbol{q} \bullet \left(0, \widehat{\boldsymbol{\nu}} \cdot \frac{\partial \widehat{\boldsymbol{\mu}}}{\partial s}, \widehat{\boldsymbol{t}} \cdot \frac{\partial \widehat{\boldsymbol{\nu}}}{\partial s}, \widehat{\boldsymbol{\mu}} \cdot \frac{\partial \widehat{\boldsymbol{t}}}{\partial s} \right) \bullet \boldsymbol{q}^*; \tag{85b}$$

which yields

$$2 \boldsymbol{q}^* \bullet \frac{\partial \boldsymbol{q}}{\partial s} = \left(0, \widehat{\boldsymbol{\nu}} \cdot \frac{\partial \widehat{\boldsymbol{\mu}}}{\partial s}, \widehat{\boldsymbol{t}} \cdot \frac{\partial \widehat{\boldsymbol{\nu}}}{\partial s}, \widehat{\boldsymbol{\mu}} \cdot \frac{\partial \widehat{\boldsymbol{t}}}{\partial s} \right). \tag{86}$$

Defining $\boldsymbol{D} = \mathrm{diag}(K_T, K_B, K_B)$ we find that Eq. (79) reduces to

$$\boldsymbol{M} = \boldsymbol{R}(\boldsymbol{q}) \, \boldsymbol{D} \left(2 \left[\boldsymbol{q}^* \bullet \frac{\partial \boldsymbol{q}}{\partial s} \right]_{\mathbb{R}^3} - \begin{pmatrix} \gamma_0 \\ \kappa_\mu \\ \kappa_\nu \end{pmatrix} \right). \tag{87}$$

A.3 The quaternion ODE

In this section, we show that if a frame vector $\boldsymbol{r}$ satisfies $\mathrm{d}\boldsymbol{r}/\mathrm{d}t = \boldsymbol{\Omega} \times \boldsymbol{r}$, where $\boldsymbol{r}$ is defined through a unit quaternion $\boldsymbol{q}$ and an initial vector $\boldsymbol{r}_0$ (i.e. $(0, \boldsymbol{r}) = \boldsymbol{q} \bullet (0, \boldsymbol{r}_0) \bullet \boldsymbol{q}^*$), then the quaternion $\boldsymbol{q}$ must satisfy $\mathrm{d}\boldsymbol{q}/\mathrm{d}t = \frac{1}{2}(0, \boldsymbol{\Omega}) \bullet \boldsymbol{q}$. Employing the same simplifications as in Eq. (80), we observe that

$$\frac{\mathrm{d}}{\mathrm{d}t}(0, \boldsymbol{r}) = \frac{\mathrm{d}\boldsymbol{q}}{\mathrm{d}t} \bullet \boldsymbol{q}^* \bullet (0, \boldsymbol{r}) - (0, \boldsymbol{r}) \bullet \frac{\mathrm{d}\boldsymbol{q}}{\mathrm{d}t} \bullet \boldsymbol{q}^*. \tag{88}$$

This implies

$$\frac{\mathrm{d}\boldsymbol{r}}{\mathrm{d}t} = 2 \left[\frac{\mathrm{d}\boldsymbol{q}}{\mathrm{d}t} \bullet \boldsymbol{q}^* \right]_{\mathbb{R}^3} \times \boldsymbol{r}. \tag{89}$$

Given that this holds true for all the vectors of the frame, we obtain $2[\mathrm{d}\boldsymbol{q}/\mathrm{d}t \bullet \boldsymbol{q}^*]_{\mathbb{R}^3} = \boldsymbol{\Omega}$, and since $\mathrm{d}\boldsymbol{q}/\mathrm{d}t \bullet \boldsymbol{q}^*$ is a pure quaternion we have

$$(0, \boldsymbol{\Omega}) = 2 \frac{\mathrm{d}\boldsymbol{q}}{\mathrm{d}t} \bullet \boldsymbol{q}^*. \tag{90}$$

Dividing by 2 and right-multiplying by $\boldsymbol{q}$ yields the desired result.

A.4 BDF2 in the Lie algebra

In this section, we derive the form of the BDF2 update equation which applies to the Lie algebra elements. This derivation follows the more general framework laid out in Faltinsen *et al.* (2001). Let $\boldsymbol{u}^m_{(k)}$ denote the Lie algebra element corresponding to the quaternion at timestep m expressed in the coordinate system from timestep k, that is,

$$\boldsymbol{q}^m = \exp\left(\boldsymbol{u}^m_{(k)} \right) \bullet \boldsymbol{q}^k. \tag{91}$$

To integrate Eq. (58) using BDF2 we need to solve

$$\boldsymbol{u}_{(n)}^{n+1} = \frac{4}{3}\boldsymbol{u}_{(n)}^{n} - \frac{1}{3}\boldsymbol{u}_{(n)}^{n-1} + \frac{2}{3}\Delta t \operatorname{dexp}^{-1}_{\boldsymbol{u}_{(n)}^{n+1}}\left(\boldsymbol{\Omega}^{n+1}\right), \tag{92}$$

given $\boldsymbol{u}_{(n-1)}^{n}$ and $\boldsymbol{u}_{(n-1)}^{n-1}$. The coordinate transform between tangent spaces needs to be such that a given Lie algebra element represents the same quaternion independent of the current coordinate system, hence

$$\exp\left(\boldsymbol{u}_{(k+1)}^{m}\right) \bullet \boldsymbol{q}^{k+1} = \boldsymbol{q}^{m} = \exp\left(\boldsymbol{u}_{(k)}^{m}\right) \bullet \boldsymbol{q}^{k} \tag{93}$$

needs to hold.

To this end, we use the Baker–Campbell–Hausdorff formula $\mathcal{B}$ which is defined to satisfy

$$\exp\left(\mathcal{B}\left(\boldsymbol{X}, \boldsymbol{Y}\right)\right) = \exp\left(\boldsymbol{X}\right) \bullet \exp\left(\boldsymbol{Y}\right). \tag{94}$$

While in our case ($\mathfrak{so}(3)$ with BDF2), there is a closed-form expression for the Baker–Campbell–Hausdorff formula (Engø, 2001), it will only be necessary to use the identity $\mathcal{B}(\boldsymbol{X}, -\boldsymbol{X}) = \boldsymbol{0}$.

For Eq. (93) to hold, the Lie algebra elements need to be transferred from one tangent space to the next. It follows from the condition $\exp(\boldsymbol{u}_{(k+1)}^{m}) \bullet \boldsymbol{q}^{k+1} = \exp(\boldsymbol{u}_{(k)}^{m}) \bullet \boldsymbol{q}^{k}$ that this can be achieved via

$$\boldsymbol{u}_{(k+1)}^{m} = \mathcal{B}\left(\boldsymbol{u}_{(k)}^{m}, -\boldsymbol{u}_{(k)}^{k+1}\right). \tag{95}$$

Substituting this into Eq. (92) we obtain

$$\boldsymbol{u}_{(n)}^{n+1} = \frac{4}{3}\mathcal{B}\left(\boldsymbol{u}_{(n-1)}^{n}, -\boldsymbol{u}_{(n-1)}^{n}\right) - \frac{1}{3}\mathcal{B}\left(\boldsymbol{u}_{(n-1)}^{n-1}, -\boldsymbol{u}_{(n-1)}^{n}\right) + \frac{2}{3}\Delta t \operatorname{dexp}^{-1}_{\boldsymbol{u}_{(n)}^{n+1}}\left(\boldsymbol{\Omega}^{n+1}\right). \tag{96}$$

Observing in the right-hand side of this expression that

$$\mathcal{B}\left(\boldsymbol{u}_{(n-1)}^{n}, -\boldsymbol{u}_{(n-1)}^{n}\right) = \boldsymbol{0}, \tag{97}$$

and, since $\boldsymbol{u}_{(n-1)}^{n-1} = \boldsymbol{0}$ from Eq. (91), that

$$\mathcal{B}\left(\boldsymbol{u}_{(n-1)}^{n-1}, -\boldsymbol{u}_{(n-1)}^{n}\right) = -\boldsymbol{u}_{(n-1)}^{n}, \tag{98}$$

we obtain

$$\boldsymbol{u}_{(n)}^{n+1} = \frac{1}{3}\boldsymbol{u}_{(n-1)}^{n} + \frac{2}{3}\Delta t \operatorname{dexp}^{-1}_{\boldsymbol{u}_{(n)}^{n+1}}\left(\boldsymbol{\Omega}^{n+1}\right). \tag{99}$$

Noticing that the Lie algebra elements only appear here expressed according to the coordinate system in which they were first produced, we may unambiguously write the BDF2 update equation as

$$\boldsymbol{u}^{n+1} = \frac{1}{3}\boldsymbol{u}^{n} + \frac{2}{3}\Delta t \operatorname{dexp}^{-1}_{\boldsymbol{u}^{n+1}}\left(\boldsymbol{\Omega}^{n+1}\right). \tag{100}$$

A.5 Filament initialisation in cloud simulations

The isotropic initialisation of straight filaments works as follows. We generate uniform random points in $[0, S]^3$, where S is the linear box size. These points set the positions of the filament segments at the $s = 0$ ends. We then generate four-dimensional random vectors drawn from the standard normal distribution. After normalising, these vectors provide the initial quaternions for all segments comprising each filament. We then initialise the remaining segment positions for each filament in straight lines along directions based on the initial quaternions. If there is any overlap between filaments, one is discarded and we reinitialise its position and quaternion. This process generates a uniform distribution of filament orientations (Karney, 2007).

B Form of the approximate Jacobian

In this section, we detail how we generate an analytic approximation to the Jacobian, $\partial \boldsymbol{f}/\partial \boldsymbol{X}$, by including only local physics for each particle – a local Stokes drag model, without hydrodynamic or steric interactions. The resulting approximate Jacobian retains the essence of the full Jacobian, but can be rapidly generated and posesses a block diagonal structure.

The supplementary code (Schoeller *et al.*, 2019) implements the generation of the Jacobian. Here, we demonstrate how the expressions are derived.

B.1 In 2D

The system $\boldsymbol{f}(\boldsymbol{X}) = \boldsymbol{0}$ as described in Eq. (41) is formed of the position update for particle $n = 1$ in each filament, the orientation updates for all n, $1 \le n \le N$, and the modified constraints for $1 < n \le N$:

$$\boldsymbol{0} = (\boldsymbol{f}^1)_{n=1}^{j+1} \coloneqq \boldsymbol{Y}_n^{j+1} - \frac{4}{3}\boldsymbol{Y}_n^j + \frac{1}{3}\boldsymbol{Y}_n^{j-1} - \frac{2}{3}\Delta t\, \boldsymbol{V}_n^{j+1}, \tag{101a}$$

$$0 = (f^2)_{1\le n\le N}^{j+1} \coloneqq \theta_n^{j+1} - \frac{4}{3}\theta_n^j + \frac{1}{3}\theta_n^{j-1} - \frac{2}{3}\Delta t\, \Omega_n^{j+1}, \tag{101b}$$

$$\boldsymbol{0} = (\boldsymbol{f}^3)_{1<n\le N}^{j+1} \coloneqq \boldsymbol{Y}_1^{j+1} + \frac{\Delta L}{2}\sum_{m=2}^{n}\left(\widehat{\boldsymbol{t}}_{m-1}^{j+1} + \widehat{\boldsymbol{t}}_m^{j+1}\right) - \frac{4}{3}\boldsymbol{Y}_n^j + \frac{1}{3}\boldsymbol{Y}_n^{j-1} - \frac{2}{3}\Delta t\, \boldsymbol{V}_n^{j+1}. \tag{101c}$$

This system is repeated for each filament.

In general, the velocities and angular velocities of the filament segments are found by solving the mobility problem,

$$\begin{pmatrix}\boldsymbol{V}\\ \boldsymbol{\Omega}\end{pmatrix} = \mathcal{M}\cdot\begin{pmatrix}\boldsymbol{F}\\ \boldsymbol{T}\end{pmatrix}, \tag{102}$$

where $\boldsymbol{V}^\top = (\boldsymbol{V}_1^\top, \ldots, \boldsymbol{V}_{NM}^\top)$ is the vector of all segments' velocity components, similarly $\boldsymbol{\Omega}$ of all angular velocity components, $\boldsymbol{F}$ of all force components and $\boldsymbol{T}$ of all torque components acting on the filaments' segments.

In the approximate Jacobian, we implement a local Stokes drag model, where the mobility matrix is diagonal, giving $\boldsymbol{V}_n = \boldsymbol{F}_n/6\pi\eta a$ and $\boldsymbol{\Omega}_n = \boldsymbol{T}_n/8\pi\eta a^3$. The only force we apply is the constraint force, Eq. (16), and the torques applied are the elastic torque, Eq. (15), and the constraint torque, Eq. (17).

Recall that for planar motion, without loss of generality, we take the third frame vector $\widehat{\boldsymbol{\nu}}$ to be aligned with the fixed vector $\widehat{\boldsymbol{e}}_z$. This direction is always normal to the plane of motion.

Together, we have

$$\boldsymbol{0} = (\boldsymbol{f}_0^1)_{n=1}^{j+1} \coloneqq \boldsymbol{Y}_1^{j+1} - \frac{4}{3}\boldsymbol{Y}_1^j + \frac{1}{3}\boldsymbol{Y}_1^{j-1} - \frac{2}{3}\frac{\Delta t}{6\pi\eta a}\left(\boldsymbol{\Lambda}_{3/2}^{j+1} - \boldsymbol{\Lambda}_{1/2}^{j+1}\right), \tag{103a}$$

$$0 = (f_0^2)_{1\le n\le N}^{j+1} \coloneqq \theta_n^{j+1} - \frac{4}{3}\theta_n^j + \frac{1}{3}\theta_n^{j-1} - \frac{2}{3}\frac{\Delta t}{8\pi\eta a^3}\times \tag{103b}$$

$$\widehat{\boldsymbol{e}}_z\cdot\left[K_B\left(\frac{\widehat{\boldsymbol{t}}_n^{j+1}\times\left(\widehat{\boldsymbol{t}}_{n-1}^{j+1}+\widehat{\boldsymbol{t}}_{n+1}^{j+1}\right)}{\Delta L} - \widehat{\boldsymbol{e}}_z\kappa_\nu\right) - \frac{\Delta L}{2}\widehat{\boldsymbol{t}}_n^{j+1}\times\left(\boldsymbol{\Lambda}_{n+1/2}^{j+1}+\boldsymbol{\Lambda}_{n-1/2}^{j+1}\right)\right],$$

$$\boldsymbol{0} = (\boldsymbol{f}_0^3)_{1<n\le N}^{j+1} \coloneqq \boldsymbol{Y}_1^{j+1} + \frac{\Delta L}{2}\sum_{m=2}^{n}\left(\widehat{\boldsymbol{t}}_{m-1}^{j+1} + \widehat{\boldsymbol{t}}_m^{j+1}\right) - \frac{4}{3}\boldsymbol{Y}_n^j + \frac{1}{3}\boldsymbol{Y}_n^{j-1} - \frac{2}{3}\frac{\Delta t}{6\pi\eta a}\left(\boldsymbol{\Lambda}_{n+1/2}^{j+1} - \boldsymbol{\Lambda}_{n-1/2}^{j+1}\right), \tag{103c}$$

repeated for each filament. Since the approximate Jacobian assumes no interaction between filaments, we can build the approximate Jacobian for a multifilament system from M independent blocks, placed along the diagonal, with each block holding a single-filament Jacobian. In the following we discuss the derivation of a single filament's approximate Jacobian without loss of generality.

Having written out the system $\boldsymbol{f}_0 = (\boldsymbol{f}_0^1, \boldsymbol{f}_0^2, \boldsymbol{f}_0^3)$ in Eq. (103), it can be differentiated with respect to the state variable, $\boldsymbol{X}^{j+1} = (\boldsymbol{Y}_1^{j+1}, \theta_{1\ldots N}^{j+1}, \boldsymbol{\Lambda}_{3/2\ldots N-1/2}^{j+1})$, to form the Jacobian

$$\boldsymbol{J}_0 = \frac{\partial \boldsymbol{f}_0}{\partial \boldsymbol{X}^{j+1}} = \begin{pmatrix} \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{J}^{13} \\ \boldsymbol{0} & \boldsymbol{J}^{22} & \boldsymbol{J}^{23} \\ \boldsymbol{I} & \boldsymbol{J}^{32} & \boldsymbol{J}^{33} \end{pmatrix}. \tag{104}$$

The remaining expressions for the submatrices relating segments m and n are given by, after some algebraic manipulation and dropping the time (superscript) indices for clarity,

$$\boldsymbol{J}_{1n}^{13} = \frac{\partial (\boldsymbol{f}_0^1)_1}{\partial \boldsymbol{\Lambda}_{n-1/2}} = \frac{2}{3}\frac{\Delta t}{6\pi\eta a}(\delta_{1n} - \delta_{2n})\boldsymbol{I}, \tag{105a}$$

$$\begin{aligned} J_{mn}^{22} = \frac{\partial (f_0^2)_m}{\partial \theta_n} &= \delta_{mn} - \frac{2}{3}\frac{\Delta t}{8\pi\eta a^3}\left[\frac{K_B}{\Delta L}\Big(- \delta_{mn}[\cos(\theta_m - \theta_{m-1}) + \cos(\theta_{m+1} - \theta_m)] \right. \\ &\qquad \left. + \delta_{(m-1)n}\cos(\theta_m - \theta_{m-1}) + \delta_{(m+1)n}\cos(\theta_{m+1} - \theta_m)\Big) \right. \\ &\qquad \left. - \frac{\Delta L}{2}\delta_{mn}\left(\sin\theta_m[\Lambda_{m+1/2}^y + \Lambda_{m-1/2}^y] + \cos\theta_m[\Lambda_{m+1/2}^x + \Lambda_{m-1/2}^x]\right)\right], \end{aligned} \tag{105b}$$

$$\boldsymbol{J}_{mn}^{23} = \frac{\partial (f_0^2)_m}{\partial \boldsymbol{\Lambda}_{n-1/2}} = \frac{2}{3}\frac{\Delta t}{8\pi\eta a^3}\frac{\Delta L}{2}(-\sin\theta_m \widehat{\boldsymbol{e}}_x + \cos\theta_m \widehat{\boldsymbol{e}}_y)(\delta_{mn} + \delta_{(m+1)n}), \tag{105c}$$

$$\boldsymbol{J}_{mn}^{32} = \frac{\partial (\boldsymbol{f}_0^3)_m}{\partial \theta_n} = \frac{\Delta L}{2}\sum_{r=2}^{m}\left[(-\sin\theta_{r-1}\delta_{(r-1)n} - \sin\theta_r\delta_{rn})\widehat{\boldsymbol{e}}_x + (\cos\theta_{r-1}\delta_{(r-1)n} + \cos\theta_r\delta_{rn})\widehat{\boldsymbol{e}}_y\right], \tag{105d}$$

$$\boldsymbol{J}_{mn}^{33} = \frac{\partial (\boldsymbol{f}_0^3)_m}{\partial \boldsymbol{\Lambda}_{n-1/2}} = \frac{2}{3}\frac{\Delta t}{6\pi\eta a}(\delta_{mn} - \delta_{(m+1)n})\boldsymbol{I}, \tag{105e}$$

where $\boldsymbol{\Lambda} = (\Lambda^x, \Lambda^y, 0)$ and δ_{ij} is the Kronecker delta.

B.2 In 3D using Lie algebra elements

Here we discuss the differences in the construction of the approximate Jacobian in 3D compared to 2D. Again we consider the system $\boldsymbol{f}(\boldsymbol{X}) = \boldsymbol{0}$, this time represented by Eq. (63). The standard BDF2 update for the first segment in the filament is handled analogously to the 2D case, Eq. (103a), so we restrict our attention to the Lie algebra update equations and the modified constraints. Substituting in the Stokes drag approximation, each filament-level sub-block of the Jacobian is based on the equations

$$\boldsymbol{0} = (\boldsymbol{f}_0^2)_{1\le n\le N}^{j+1} \coloneqq \boldsymbol{u}_n^{j+1} - \frac{1}{3}\boldsymbol{u}_n^j - \frac{\Delta t}{12\pi\eta a^3}\exp_{\boldsymbol{u}_n^{j+1}}^{-1}\left(\boldsymbol{T}_n^{j+1}\right), \tag{106a}$$

$$\boldsymbol{0} = (\boldsymbol{f}_0^3)_{1<n\le N}^{j+1} \coloneqq \boldsymbol{Y}_1^{j+1} + \frac{\Delta L}{2}\sum_{m=2}^{n}\left(\widehat{\boldsymbol{t}}_{m-1}^{j+1} + \widehat{\boldsymbol{t}}_m^{j+1}\right) - \frac{4}{3}\boldsymbol{Y}_n^j + \frac{1}{3}\boldsymbol{Y}_n^{j-1} - \frac{\Delta t}{9\pi\eta a}\left(\boldsymbol{\Lambda}_{n+1/2}^{j+1} - \boldsymbol{\Lambda}_{n-1/2}^{j+1}\right), \tag{106b}$$

where the Lie algebra update equations are for all segments $1 \le n \le N$, and the modified constraints are for segments $1 < n \le N$.

For the sake of legibility, we discard any terms with a time index other than $j+1$, as they will have zero derivative in all cases, and drop the time superscript. Hence, to produce the Jacobian it suffices to consider the equations

$$\boldsymbol{0} = (\boldsymbol{f}_0^2)_{1\le n\le N} \coloneqq \boldsymbol{u}_n - \frac{\Delta t}{12\pi\eta a^3}\exp_{\boldsymbol{u}_n}^{-1}(\boldsymbol{T}_n), \tag{107a}$$

$$\boldsymbol{0} = (\boldsymbol{f}_0^3)_{1<n\le N} \coloneqq \boldsymbol{Y}_1 + \frac{\Delta L}{2}\sum_{m=2}^{n}\left(\widehat{\boldsymbol{t}}_{m-1} + \widehat{\boldsymbol{t}}_m\right) - \frac{\Delta t}{9\pi\eta a}\left(\boldsymbol{\Lambda}_{n+1/2} - \boldsymbol{\Lambda}_{n-1/2}\right). \tag{107b}$$

The Jacobian then corresponds to the system $\boldsymbol{f}_0 = (\boldsymbol{f}_0^1, \boldsymbol{f}_0^2, \boldsymbol{f}_0^3)$ in Eqs. (103a) and (107), differentiated with respect to the state variable, $\boldsymbol{X} = (\boldsymbol{Y}_1, \boldsymbol{u}_{1\ldots N}, \boldsymbol{\Lambda}_{3/2\ldots N-1/2})$, forming

$$\boldsymbol{J}_0 = \frac{\partial \boldsymbol{f}_0}{\partial \boldsymbol{X}} = \begin{pmatrix} \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{J}^{13} \\ \boldsymbol{0} & \boldsymbol{J}^{22} & \boldsymbol{J}^{23} \\ \boldsymbol{I} & \boldsymbol{J}^{32} & \boldsymbol{J}^{33} \end{pmatrix}. \tag{108}$$

The expressions for $\boldsymbol{J}^{13}$ and $\boldsymbol{J}^{33}$ are the same as in the 2D case, Eqs. (105a) and (105e). We therefore consider the remaining three blocks.

We first observe that

$$\boldsymbol{J}^{32}_{mn} = \frac{\partial (\boldsymbol{f}_0^3)_m}{\partial \boldsymbol{u}_n} = \frac{\Delta L}{2} \sum_{k=2}^{m} \left(\delta_{(k-1)n} \frac{\partial \widehat{\boldsymbol{t}}_{k-1}}{\partial \boldsymbol{u}_n} + \delta_{kn} \frac{\partial \widehat{\boldsymbol{t}}_k}{\partial \boldsymbol{u}_n} \right), \tag{109}$$

where δ_{ij} is the Kronecker delta. By taking Eq. (58) and replacing $\mathrm{dexp}_{\boldsymbol{u}}^{-1}(\boldsymbol{\Omega})$ by the first term of its Taylor series expansion,

$$\mathrm{dexp}_{\boldsymbol{u}}^{-1}(\boldsymbol{v}) \approx \boldsymbol{v} - \frac{1}{2}\boldsymbol{u} \times \boldsymbol{v} + \frac{1}{12}\boldsymbol{u} \times (\boldsymbol{u} \times \boldsymbol{v}), \tag{110}$$

we can approximate $\mathrm{d}\boldsymbol{u}_n/\mathrm{d}t \approx \boldsymbol{\Omega}$. Combining this with $\mathrm{d}\widehat{\boldsymbol{t}}_n/\mathrm{d}t = \boldsymbol{\Omega} \times \widehat{\boldsymbol{t}}_n$ from Eq. (54), we have

$$\frac{\partial \widehat{\boldsymbol{t}}_n}{\partial \boldsymbol{u}_n} \approx \left[\times \widehat{\boldsymbol{t}}_n \right], \tag{111}$$

where we have introduced the notation $[\boldsymbol{v}\times]$ for the skew-symmetric matrix satisfying $[\boldsymbol{v}\times]\boldsymbol{x} = \boldsymbol{v} \times \boldsymbol{x}$ (and $[\times \boldsymbol{v}] = -[\boldsymbol{v}\times] = [\boldsymbol{v}\times]^\top$ satisfying $[\times \boldsymbol{v}]\boldsymbol{x} = \boldsymbol{x} \times \boldsymbol{v}$). Hence,

$$\boldsymbol{J}^{32}_{mn} = \frac{\partial (\boldsymbol{f}_0^3)_m}{\partial \boldsymbol{u}_n} \approx \frac{\Delta L}{2} \sum_{k=2}^{m} \left(\delta_{(k-1)n} + \delta_{n,k} \right) \left[\times \widehat{\boldsymbol{t}}_n \right]. \tag{112}$$

For the final blocks, we employ all three terms written in the Taylor series expansion in Eq. (110) to reduce Eq. (106a) to

$$(\boldsymbol{f}_0^2)_m \approx \boldsymbol{u}_m - \frac{\Delta t}{12\pi\eta a^3} \left(\boldsymbol{T}_m - \frac{1}{2}\boldsymbol{u}_m \times \boldsymbol{T}_m + \frac{1}{12}\boldsymbol{u}_m \times (\boldsymbol{u}_m \times \boldsymbol{T}_m) \right). \tag{113}$$

From this we can see that the matrices of interest take the forms

$$\boldsymbol{J}^{23}_{mn} = \frac{\partial (\boldsymbol{f}_0^2)_m}{\partial \boldsymbol{\Lambda}_{n-1/2}} \approx -\frac{\Delta t}{12\pi\eta a^3} \left(\boldsymbol{I} - \frac{1}{2}[\boldsymbol{u}_m\times] + \frac{1}{12}[\boldsymbol{u}_m\times]^2 \right) \frac{\partial \boldsymbol{T}_m}{\partial \boldsymbol{\Lambda}_{n-1/2}}, \tag{114}$$

$$\boldsymbol{J}^{22}_{mn} = \frac{\partial (\boldsymbol{f}_0^2)_m}{\partial \boldsymbol{u}_n} \approx -\frac{\Delta t}{12\pi\eta a^3} \left(\boldsymbol{I} - \frac{1}{2}[\boldsymbol{u}_m\times] + \frac{1}{12}[\boldsymbol{u}_m\times]^2 \right) \frac{\partial \boldsymbol{T}_m}{\partial \boldsymbol{u}_n}, \tag{115}$$

for $m \neq n$, and otherwise,

$$\begin{aligned} \boldsymbol{J}^{22}_{mm} = \frac{\partial (\boldsymbol{f}_0^2)_m}{\partial \boldsymbol{u}_m} \approx \boldsymbol{I} - \frac{\Delta t}{12\pi\eta a^3} \Bigg\{ & \frac{\partial \boldsymbol{T}_m}{\partial \boldsymbol{u}_m} - \frac{1}{2} \left([\times \boldsymbol{T}_m] + [\boldsymbol{u}_m\times] \frac{\partial \boldsymbol{T}_m}{\partial \boldsymbol{u}_m} \right) \\ & + \frac{1}{12} \left([\times (\boldsymbol{u}_m \times \boldsymbol{T}_m)] + [\boldsymbol{u}_m\times] [\times \boldsymbol{T}_m] + [\boldsymbol{u}_m\times]^2 \frac{\partial \boldsymbol{T}_m}{\partial \boldsymbol{u}_m} \right) \Bigg\}. \end{aligned} \tag{116}$$

The problem of constructing these matrices thus reduces to that of evaluating the derivatives of the torque on segment m. Segment m experiences elastic and constraint torques

$$\boldsymbol{T}_m = \frac{\Delta L}{2} \widehat{\boldsymbol{t}}_m \times \left(\boldsymbol{\Lambda}_{m+1/2} + \boldsymbol{\Lambda}_{m-1/2} \right) + \boldsymbol{M}_{m+1/2} - \boldsymbol{M}_{m-1/2}, \tag{117}$$

where $\boldsymbol{M}_{m+1/2}$ is the elastic moment between segments m and $m+1$. Since the elastic moments do not depend on the Lagrange multipliers, we have

$$\frac{\partial \boldsymbol{T}_m}{\partial \boldsymbol{\Lambda}_{n-1/2}} \approx \frac{\Delta L}{2} \left(\delta_{(m+1)n} + \delta_{mn} \right) \left[\widehat{\boldsymbol{t}}_m \times \right]. \tag{118}$$

Now considering the derivatives with respect to the Lie algebra elements, we have

$$\frac{\partial \boldsymbol{T}_m}{\partial \boldsymbol{u}_n} \approx \frac{\Delta L}{2}\delta_{mn}\left[\times\left(\boldsymbol{\Lambda}_{m+1/2}+\boldsymbol{\Lambda}_{m-1/2}\right)\right]\left[\times\widehat{\boldsymbol{t}}_m\right]+\frac{\partial \boldsymbol{M}_{m+1/2}}{\partial \boldsymbol{u}_n}-\frac{\partial \boldsymbol{M}_{m-1/2}}{\partial \boldsymbol{u}_n}. \tag{119}$$

If we use a frame vector approximation for the elastic moment,

$$\boldsymbol{M}_{m+1/2} \approx K_B\left(\frac{\widehat{\boldsymbol{t}}_m\times\widehat{\boldsymbol{t}}_{m+1}}{\Delta L}-\frac{\kappa_\mu}{2}\left(\widehat{\boldsymbol{\mu}}_m+\widehat{\boldsymbol{\mu}}_{m+1}\right)-\frac{\kappa_\nu}{2}\left(\widehat{\boldsymbol{\nu}}_m+\widehat{\boldsymbol{\nu}}_{m+1}\right)\right)+\frac{K_T\beta}{2}\left(\widehat{\boldsymbol{t}}_m+\widehat{\boldsymbol{t}}_{m+1}\right), \tag{120}$$

where $\beta=\frac{1}{2\Delta L}(\widehat{\boldsymbol{\mu}}_{m+1}\cdot\widehat{\boldsymbol{\nu}}_m-\widehat{\boldsymbol{\mu}}_m\cdot\widehat{\boldsymbol{\nu}}_{m+1})-\gamma_0$, then we can write

$$\begin{aligned}\frac{\partial \boldsymbol{M}_{m+1/2}}{\partial \boldsymbol{u}_m} \approx & K_B\left(\frac{1}{\Delta L}\left[\times\widehat{\boldsymbol{t}}_{m+1}\right]\left[\times\widehat{\boldsymbol{t}}_m\right]-\frac{\kappa_\mu}{2}\left[\times\widehat{\boldsymbol{\mu}}_m\right]-\frac{\kappa_\nu}{2}\left[\times\widehat{\boldsymbol{\nu}}_m\right]\right)\\ &+\frac{K_T}{2}\left[\beta\left[\times\widehat{\boldsymbol{t}}_m\right]+\frac{1}{2\Delta L}\left(\widehat{\boldsymbol{t}}_m+\widehat{\boldsymbol{t}}_{m+1}\right)\left(\widehat{\boldsymbol{\nu}}_m\times\widehat{\boldsymbol{\mu}}_{m+1}-\widehat{\boldsymbol{\mu}}_m\times\widehat{\boldsymbol{\nu}}_{m+1}\right)\right],\end{aligned} \tag{121}$$

where the last term is a dyadic product. Similarly,

$$\begin{aligned}\frac{\partial \boldsymbol{M}_{m+1/2}}{\partial \boldsymbol{u}_{m+1}} \approx & K_B\left(\frac{1}{\Delta L}\left[\widehat{\boldsymbol{t}}_m\times\right]\left[\times\widehat{\boldsymbol{t}}_{m+1}\right]-\frac{\kappa_\mu}{2}\left[\times\widehat{\boldsymbol{\mu}}_{m+1}\right]-\frac{\kappa_\nu}{2}\left[\times\widehat{\boldsymbol{\nu}}_{m+1}\right]\right)\\ &+\frac{K_T}{2}\left[\beta\left[\times\widehat{\boldsymbol{t}}_{m+1}\right]-\frac{1}{2\Delta L}\left(\widehat{\boldsymbol{t}}_m+\widehat{\boldsymbol{t}}_{m+1}\right)\left(\widehat{\boldsymbol{\nu}}_m\times\widehat{\boldsymbol{\mu}}_{m+1}-\widehat{\boldsymbol{\mu}}_m\times\widehat{\boldsymbol{\nu}}_{m+1}\right)\right].\end{aligned} \tag{122}$$

The derivative with respect to any other Lie algebra element is zero, and thus we have provided all of the elements necessary to form an approximate Jacobian. We find that level of approximation employed here works well in our implementation.

References

Agrawal, A. & Babu, S. B. 2018 Self-organization in a bimotility mixture of model microswimmers. *Physical Review E* **97** (2), 020401.

Allen, M. P. & Tildesley, D. J. 2017 *Computer Simulation of Liquids*, 2nd edn. Oxford University Press, Oxford.

Ascher, U. M. & Petzold, L. R. 1998 *Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations*, vol. 61. SIAM.

Baker, E. L., Bonnecaze, R. T. & Zaman, M. H. 2009 Extracellular matrix stiffness and architecture govern intracellular rheology in cancer. *Biophysical Journal* **97** (4), 1013–1021.

Batchelor, G. K. 1967 *An Introduction to Fluid Dynamics*. Cambridge University Press.

Brady, J. F. & Bossis, G. 1988 Stokesian Dynamics. *Annual Review of Fluid Mechanics* **20** (1), 111–157.

Brennen, C. E. & Winet, H. 1977 Fluid mechanics of propulsion by cilia and flagella. *Annual Review of Fluid Mechanics* **9**, 339–398.

Broyden, C. G. 1965 A class of methods for solving nonlinear simultaneous equations. *Mathematics of Computation* **19** (92), 577–593.

Chelakkot, R., Winkler, R. G. & Gompper, G. 2010 Migration of semiflexible polymers in microcapillary flow. *Europhysics Letters* **91** (1), 14001.

Claeys, I. L. & Brady, J. F. 1993 Suspensions of prolate spheroids in Stokes flow. Part 1. Dynamics of a finite number of particles in an unbounded fluid. *Journal of Fluid Mechanics* **251**, 411–442.

Coq, N., du Roure, O., Fermigier, M. & Bartolo, D. 2009 Helical beating of an actuated elastic filament. *Journal of Physics: Condensed Matter* **21** (20), 204109.

Coq, N., du Roure, O., Marthelot, J., Bartolo, D. & Fermigier, M. 2008 Rotational dynamics of a soft filament: Wrapping transition and propulsive forces. *Physics of Fluids* **20** (5), 051703.

Cortez, R. 2001 The method of regularized Stokeslets. *SIAM Journal on Scientific Computing* **23** (4), 1204–1225.

Cortez, R. 2018 Regularized stokeslet segments. *Journal of Computational Physics* **375**, 783–796.

Cortez, R., Fauci, L. J. & Medovikov, A. A. 2005 The method of regularized Stokeslets in three dimensions: Analysis, validation, and application to helical swimming. *Physics of Fluids* **17** (031504).

Cosentino Lagomarsino, M., Pagonabarraga, I. & Lowe, C. P. 2005 Hydrodynamic induced deformation and orientation of a microscopic elastic filament. *Physical Review Letters* **94** (14), 148104.

Dance, S. L., Climent, E. & Maxey, M. R. 2004 Collision barrier effects on the bulk flow in a random suspension. *Physics of Fluids* **16** (3), 828–831.

Delmotte, B., Climent, E. & Plouraboué, F. 2015 A general formulation of bead models applied to flexible fibers and active filaments at low Reynolds number. *Journal of Computational Physics* **286**, 14–37.

Delong, S., Balboa Usabiaga, F. & Donev, A. 2015 Brownian dynamics of confined rigid bodies. *The Journal of Chemical Physics* **143** (14), 144107.

Derakhshandeh, B., Kerekes, R. J., Hatzikiriakos, S. G. & Bennington, C. P. J. 2011 Rheology of pulp fibre suspensions: A critical review. *Chemical Engineering Science* **66** (15), 3460–3470.

du Roure, O., Lindner, A., Nazockdast, E. N. & Shelley, M. J. 2017 Dynamics of flexible fibers in viscous flows and fluids. *Annual Review of Fluid Mechanics* **51** (1), 539–572.

Dunn, F. & Parberry, I. 2011 *3D Math Primer for Graphics and Game Development*, 2nd edn. A K Peters/CRC Press, Boca Raton, Florida.

Elgeti, J. & Gompper, G. 2013 Emergence of metachronal waves in cilia arrays. *Proceedings of the National Academy of Sciences of the United States of America* **110** (12), 4470–4475.

Elgeti, J., Winkler, R. G. & Gompper, G. 2015 Physics of microswimmers – single particle motion and collective behavior: A review. *Reports on Progress in Physics* **78** (5), 056601.

Engø, K. 2001 On the BCH-formula in $\mathfrak{so}(3)$. *BIT Numerical Mathematics* **41** (3), 629–632.

Faltinsen, S., Marthinsen, A. & Munthe-Kaas, H. Z. 2001 Multistep methods integrating ordinary differential equations on manifolds. *Applied Numerical Mathematics* **39** (3), 349–365.

Faubel, R., Westendorf, C., Bodenschatz, E. & Eichele, G. 2016 Cilia-based flow network in the brain ventricles. *Science* **353** (6295), 176–178.

Fauci, L. J. & Peskin, C. S. 1988 A computational model of aquatic animal locomotion. *Journal of Computational Physics* **77** (1), 85–108.

Guo, H., Fauci, L. J., Shelley, M. J. & Kanso, E. 2018 Bistability in the synchronization of actuated microfilaments. *Journal of Fluid Mechanics* **836**, 304–323.

Gustavsson, K. & Tornberg, A.-K. 2009 Gravity induced sedimentation of slender fibers. *Physics of Fluids* **21** (12), 123301.

Hall-McNair, A. L., Gallagher, M. T., Montenegro-Johnson, T. D., Gadêlha, H. & Smith, D. J. 2019 Efficient Implementation of Elastohydrodynamics via Integral Operators. *arXiv:1903.03427 [physics]* .

Heck, T., Smeets, B., Vanmaercke, S., Bhattacharya, P., Odenthal, T., Ramon, H., Van Oosterwyck, H. & Van Liedekerke, P. 2017 Modeling extracellular matrix viscoelasticity using smoothed particle hydrodynamics with improved boundary treatment. *Computer Methods in Applied Mechanics and Engineering* **322**, 515–540.

Hwang, S. H., Litt, M. & Forsman, W. C. 1969 Rheological properties of mucus. *Rheologica Acta* **8** (4), 438–448.

Iserles, A., Munthe-Kaas, H. Z., Nørsett, S. P. & Zanna, A. 2000 Lie-group methods. *Acta Numerica* **9**, 215–365.

Jung, S., Spagnolie, S. E., Parikh, K., Shelley, M. J. & Tornberg, A.-K. 2006 Periodic sedimentation in a Stokesian fluid. *Physical Review E* **74** (3), 035302.

Kamal, A. & Keaveny, E. E. 2018 Enhanced locomotion, effective diffusion and trapping of undulatory micro-swimmers in heterogeneous environments. *Journal of The Royal Society Interface* **15** (148), 20180592.

Karney, C. F. F. 2007 Quaternions in molecular modeling. *Journal of Molecular Graphics and Modelling* **25** (5), 595–604.

Keaveny, E. E. 2008 *Dynamics of structures in active suspensions of paramagnetic particles and applications to artificial micro-swimmers*. PhD Thesis, Brown University, United States.

Keaveny, E. E. 2014 Fluctuating force-coupling method for simulations of colloidal suspensions. *Journal of Computational Physics* **269**, 61–79.

Keaveny, E. E. & Shelley, M. J. 2011 Applying a second-kind boundary integral equation for surface tractions in stokes flow. *Journal of Computational Physics* **230** (5), 2141–2159.

Kim, Y. W. & Netz, R. R. 2006 Pumping fluids with periodically beating grafted elastic filaments. *Physical Review Letters* **96** (15), 158101.

Kirchhoff, G. 1859 Über das Gleichgewicht und die Bewegung eines unendlich dünnen elastischen Stabes. *Journal für die Reine und Angewandte Mathematik* **56**, 285–313.

Knoll, D. A. & Keyes, D. E. 2004 Jacobian-free Newton–Krylov methods: A survey of approaches and applications. *Journal of Computational Physics* **193** (2), 357–397.

Kvaalen, E. 1991 A faster Broyden method. *BIT Numerical Mathematics* **31** (2), 369–372.

Lai, S. K., Wang, Y.-Y., Wirtz, D. & Hanes, J. 2009 Micro- and macrorheology of mucus. *Advanced Drug Delivery Reviews* **61** (2), 86–100.

Landau, L. D. & Lifshitz, E. M. 1986 *Theory of Elasticity*, 3rd edn., vol. 7. Elsevier.

Larson, R. G. 1999 *The Structure and Rheology of Complex Fluids*, 1st edn. Oxford University Press, New York.

Lauga, E. & Powers, T. R. 2009 The hydrodynamics of swimming microorganisms. *Reports on Progress in Physics* **72** (9), 096601.

Li, L., Manikantan, H., Saintillan, D. & Spagnolie, S. E. 2013 The sedimentation of flexible filaments. *Journal of Fluid Mechanics* **735**, 705–736.

Liang, Z., Gimbutas, Z., Greengard, L., Huang, J. & Jiang, S. 2013 A fast multipole method for the Rotne–Prager–Yamakawa tensor and its applications. *Journal of Computational Physics* **234**, 133–139.

Lim, S. 2010 Dynamics of an open elastic rod with intrinsic curvature and twist in a viscous fluid. *Physics of Fluids* **22** (2), 024104.

Lim, S., Ferent, A., Wang, X. S. & Peskin, C. S. 2008 Dynamics of a closed rod with twist and bend in fluid. *SIAM Journal on Scientific Computing* **31** (1), 273–302.

Liu, D., Keaveny, E. E., Maxey, M. R. & Karniadakis, G. 2009 Force-coupling method for flows with ellipsoidal particles. *Journal of Computational Physics* **228** (10), 3559–3581.

Llopis, I., Pagonabarraga, I., Cosentino Lagomarsino, M. & Lowe, C. P. 2007 Sedimentation of pairs of hydrodynamically interacting semiflexible filaments. *Physical Review E* **76** (6), 061901.

Lomholt, S. & Maxey, M. R. 2003 Force-coupling method for particulate two-phase flow: Stokes flow. *Journal of Computational Physics* **184**, 381–405.

Mackaplow, M. B. & Shaqfeh, E. S. G. 1998 A numerical study of the sedimentation of fibre suspensions. *Journal of Fluid Mechanics* **376**, 149–182.

Majmudar, T., Keaveny, E. E., Zhang, J. & Shelley, M. J. 2012 Experiments and theory of undulatory locomotion in a simple structured medium. *Journal of The Royal Society Interface* **9** (73), 1809–1823.

Marchetti, B., Raspa, V., Lindner, A., du Roure, O., Bergougnoux, L., Guazzelli, E. & Duprat, C. 2018 Deformation of a flexible fiber settling in a quiescent viscous fluid. *Physical Review Fluids* **3** (10), 104102.

Maxey, M. R. & Patel, B. K. 2001 Localized force representations for particles sedimenting in Stokes fow. *International Journal of Multiphase Flow* **27**, 1603–1626.

Metzger, B., Nicolas, M. & Guazzelli, E. 2007 Falling clouds of particles in viscous fluids. *Journal of Fluid Mechanics* **580**, 283–301.

Moreau, C., Giraldi, L. & Gadêlha, H. 2018 The asymptotic coarse-graining formulation of slender-rods, bio-filaments and flagella. *Journal of The Royal Society Interface* **15** (144), 20180235.

Nazockdast, E. N., Rahimian, A., Zorin, D. & Shelley, M. J. 2017 A fast platform for simulating semi-flexible fiber suspensions applied to cell mechanics. *Journal of Computational Physics* **329**, 173–209.

Nitsche, J. M. & Batchelor, G. K. 1997 Break-up of a falling drop containing dispersed particles. *Journal of Fluid Mechanics* **340**, 161–175.

Olson, S. D., Lim, S. & Cortez, R. 2013 Modeling the dynamics of an elastic rod with intrinsic curvature and twist using a regularized Stokes formulation. *Journal of Computational Physics* **238**, 169–187.

Park, J. & Chung, W.-K. 2005 Geometric integration on Euclidean group with application to articulated multibody systems. *IEEE Transactions on Robotics* **21** (5), 850–863.

Park, J., Metzger, B., Guazzelli, E. & Butler, J. E. 2010 A cloud of rigid fibres sedimenting in a viscous fluid. *Journal of Fluid Mechanics* **648**, 351–362.

Peskin, C. S. 2002 The immersed boundary method. *Acta Numerica* **11**, 479–517.

Pettersson, T., Hellwig, J., Gustafsson, P.-J. & Stenström, S. 2017 Measurement of the flexibility of wet cellulose fibres using atomic force microscopy. *Cellulose* **24** (10), 4139–4149.

Powers, T. R. 2010 Dynamics of filaments and membranes in a viscous fluid. *Reviews of Modern Physics* **82** (2), 1607–1631.

Pozrikidis, C. 1992 *Boundary Integral and Singularity Methods for Linearized Viscous Flow*. Cambridge University Press.

Purcell, E. M. 1977 Life at low Reynolds number. *American Journal of Physics* **45** (1).

Quraishi, M. S., Jones, N. S. & Mason, J. 1998 The rheology of nasal mucus: A review. *Clinical Otolaryngology & Allied Sciences* **23** (5), 403–413.

Ross, R. F. & Klingenberg, D. J. 1997 Dynamic simulation of flexible fibers composed of linked rigid bodies. *The Journal of Chemical Physics* **106** (7), 2949–2960.

Rosti, M. E., Banaei, A. A., Brandt, L. & Mazzino, A. 2018 Flexible fiber reveals the two-point statistical properties of turbulence. *Physical Review Letters* **121** (4), 044501.

Saad, Y. & Schultz, M. H. 1986 GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal on Scientific and Statistical Computing* **7** (3), 856–869.

Saggiorato, G., Elgeti, J., Winkler, R. G. & Gompper, G. 2015 Conformations, hydrodynamic interactions, and instabilities of sedimenting semiflexible filaments. *Soft Matter* **11** (37), 7337–7344.

Saintillan, D., Darve, E. & Shaqfeh, E. S. G. 2005 A smooth particle-mesh Ewald algorithm for Stokes suspension simulations: The sedimentation of fibers. *Physics of Fluids* **17** (3), 033301.

Sanderson, C. & Curtin, R. 2016 Armadillo: A template-based C++ library for linear algebra. *The Journal of Open Source Software* **1**, 26.

Sanderson, C. & Curtin, R. 2018 A User-Friendly Hybrid Sparse Matrix Class in C++. *Lecture Notes in Computer Science* **10931**, 422–430.

Schoeller, S. F. & Keaveny, E. E. 2018 From flagellar undulations to collective motion: Predicting the dynamics of sperm suspensions. *Journal of The Royal Society Interface* **15** (140), 20170834.

Schoeller, S. F., Townsend, A. K., Westwood, T. A. & Keaveny, E. E. 2019 GitHub repository. `https://github.com/ekeaveny/filaments/`.

Sheehan, J. K. & Carlstedt, I. 1984 Hydrodynamic properties of human cervical-mucus glycoproteins in 6M-guanidinium chloride. *The Biochemical Journal* **217** (1), 93–101.

Shelley, M. J. 2016 The dynamics of microtubule/motor-protein assemblies in biology and physics. *Annual Review of Fluid Mechanics* **48** (1), 487–506.

Sierou, A. & Brady, J. F. 2001 Accelerated Stokesian Dynamics simulations. *Journal of Fluid Mechanics* **448**, 115–146.

Simons, J., Fauci, L. J. & Cortez, R. 2015 A fully three-dimensional model of the interaction of driven elastic filaments in a Stokes flow with applications to sperm motility. *Journal of Biomechanics* **48** (9), 1639–1651.

Smith, D. J. 2009 A boundary element regularized Stokeslet method applied to cilia- and flagella-driven flow. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* **465** (2112), 3605–3626.

Smith, D. J., Gaffney, E. A. & Blake, J. R. 2007 Discrete cilia modelling with singularity distributions: Application to the embryonic node and the airway surface liquid. *Bulletin of Mathematical Biology* **69** (5), 1477–1510.

Smith, D. J., Montenegro-Johnson, T. D. & Lopes, S. S. 2019 Symmetry-breaking cilia-driven flow in embryogenesis. *Annual Review of Fluid Mechanics* **51** (1), 105–128.

Stockie, J. M. & Green, S. I. 1998 Simulating the motion of flexible pulp fibres using the immersed boundary method. *Journal of Computational Physics* **147** (1), 147–165.

Supatto, W. & Vermot, J. 2011 From cilia hydrodynamics to zebrafish embryonic development. *Current Topics in Developmental Biology* **95**, 33–66.

Swan, J. W. & Brady, J. F. 2007 Simulation of hydrodynamically interacting particles near a no-slip boundary. *Physics of Fluids* **19** (11), 113306.

Teran, J., Fauci, L. & Shelley, M. 2010 Viscoelastic fluid response can increase the speed and efficiency of a free swimmer. *Physical Review Letters* **104** (3), 038101.

Tornberg, A.-K. & Gustavsson, K. 2006 A numerical method for simulations of rigid fiber suspensions. *Journal of Computational Physics* **215** (1), 172–196.

Tornberg, A.-K. & Shelley, M. J. 2004 Simulating the dynamics and interactions of flexible fibers in Stokes flows. *Journal of Computational Physics* **196** (1), 8–40.

van de Rotten, B. A. 2003 *A limited memory Broyden method to solve high-dimensional systems of nonlinear equations*. PhD Thesis, Leiden University, Netherlands.

Vince, J. 2011 *Quaternions for Computer Graphics*. Springer-Verlag, London.

Wajnryb, E., Mizerski, K. A., Zuk, P. J. & Szymczak, P. 2013 Generalization of the Rotne–Prager–Yamakawa mobility and shear disturbance tensors. *Journal of Fluid Mechanics* **731**.

Wiens, J. K. & Stockie, J. M. 2015 Simulating flexible fiber suspensions using a scalable immersed boundary algorithm. *Computer Methods in Applied Mechanics and Engineering* **290**, 1–18.

Yamamoto, S. & Matsuoka, T. 1995 Dynamic simulation of fiber suspensions in shear flow. *The Journal of Chemical Physics* **102** (5), 2254–2260.

Yang, Y., Elgeti, J. & Gompper, G. 2008 Cooperation of sperm in two dimensions: Synchronization, attraction, and aggregation through hydrodynamic interactions. *Physical Review E* **78** (061903).

Yang, Y., Marceau, V. & Gompper, G. 2010 Swarm behavior of self-propelled rods and swimming flagella. *Physical Review E* **82** (031904).

Yeo, K. & Maxey, M. R. 2010 Simulation of concentrated suspensions using the force-coupling method. *Journal of Computational Physics* **229** (6), 2401–2421.

Zuk, P. J., Wajnryb, E., Mizerski, K. A. & Szymczak, P. 2014 Rotne–Prager–Yamakawa approximation for different-sized particles in application to macromolecular bead models. *Journal of Fluid Mechanics* **741**.

Zupan, E., Saje, M. & Zupan, D. 2009 The quaternion-based three-dimensional beam theory. *Computer Methods in Applied Mechanics and Engineering* **198**, 3944–3956.