

# A numerical method for suspensions of articulated bodies in viscous flows



Florencio Balboa Usabiaga<sup>a</sup>, Blaise Delmotte<sup>b,\*</sup>

<sup>a</sup> BCAM - Basque Center for Applied Mathematics, Mazarredo 14, Bilbao, E48009, Basque Country, Spain

<sup>b</sup> LadHyX, CNRS, Ecole Polytechnique, Institut Polytechnique de Paris, Palaiseau, 91120, France

## ARTICLE INFO

### Article history:

Received 23 July 2021

Received in revised form 18 March 2022

Accepted 23 May 2022

Available online 1 June 2022

### Keywords:

Suspensions

Stokes flow

Fluid-structure interactions

Active matter

Constraints

Articulated bodies

## ABSTRACT

An articulated body is defined as a finite number of rigid bodies connected by a set of arbitrary constraints that limit the relative motion between pairs of bodies. Such a general definition encompasses a wide variety of situations in the microscopic world, from bacteria to synthetic micro-swimmers, but it is also encountered when discretizing inextensible bodies, such as filaments or membranes. In this work we consider hybrid articulated bodies, i.e. constituted of both linear chains, such as filaments, and closed-loop chains, such as membranes. Simulating suspensions of such articulated bodies requires to solve the hydrodynamic interactions between large collections of objects of arbitrary shape while satisfying the multiple constraints that connect them. Two main challenges arise in this task: limiting the cost of the hydrodynamic solves, and enforcing the constraints within machine precision at each time-step. To address these challenges we propose a formalism that combines the body mobility problem in Stokes flow with a velocity formulation of the constraints, resulting in a mixed mobility-resistance problem. While resistance problems are known to scale poorly with the particle number, our preconditioned iterative solver is not sensitive to the system size, therefore allowing to study large suspensions with quasilinear computational cost. Additionally, constraint violations, e.g. due to discrete time-integration errors, are prevented by correcting the particles' positions and orientations at the end of each time-step. Our correction procedure, based on a nonlinear minimisation algorithm, has negligible computational cost and preserves the accuracy of the time-integration scheme. The versatility of our method allows to study a plethora of articulated systems within a unified framework. We showcase its robustness and scalability by exploring the locomotion modes of a model microswimmer inspired by the diatom colony *Bacillaria Paxillifer*, and by simulating large suspensions of bacteria interacting near a no-slip boundary. Finally, we provide a Python implementation of our framework in a collaborative publicly available code, where the user can prescribe a set of constraints through a single input file to study a wide spectrum of applications involving suspensions of articulated bodies.

© 2022 Elsevier Inc. All rights reserved.

## Contents

1. Introduction . . . . .	2
---------------------------	---

\* Corresponding author.

E-mail address: [blaise.delmotte@ladhyx.polytechnique.fr](mailto:blaise.delmotte@ladhyx.polytechnique.fr) (B. Delmotte).

2.	Model and formulation . . . . .	4
2.1.	Continuum formulation for rigid bodies . . . . .	4
2.2.	Constrained mobility problem . . . . .	5
2.3.	Rigid multiblob model as a Stokes solver . . . . .	6
2.4.	Constraints with single blobs . . . . .	7
3.	Preconditionner and iterative solver convergence . . . . .	7
3.1.	Block diagonal preconditionner . . . . .	7
3.2.	Convergence results . . . . .	8
3.2.1.	Settling filaments . . . . .	8
3.2.2.	Deformable shells . . . . .	9
4.	Body reconstruction and time-integration . . . . .	10
4.1.	Reconstruction method for general articulated bodies . . . . .	10
4.1.1.	The robot-arm model for linear chains . . . . .	10
4.1.2.	General reconstruction method . . . . .	10
4.2.	Correction step for discrete time-integrators . . . . .	11
4.3.	Reconstruction and correction with Explicit Euler . . . . .	12
4.4.	Convergence in time . . . . .	13
5.	Simulations . . . . .	14
5.1.	Swimming motion of a model diatom chain . . . . .	14
5.2.	Single bacterium . . . . .	15
5.3.	Bacterial suspension . . . . .	16
6.	Conclusions . . . . .	18
	CRedit authorship contribution statement . . . . .	18
	Declaration of competing interest . . . . .	18
	Acknowledgements . . . . .	18
Appendix A.	No net constraint force and torque on articulated bodies . . . . .	18
Appendix B.	Jacobian of the objective function . . . . .	19
Appendix C.	Midpoint scheme . . . . .	19
Appendix D.	Supplementary material . . . . .	20
	References . . . . .	20

## 1. Introduction

An articulated body consists of a set of rigid bodies connected together by constraints that limit the relative motion between pairs of bodies. Articulated bodies are ubiquitous in microscopic systems. Bacteria, the dominant prokaryotic microorganisms, self-propel with one or several helical flagella attached to their rigid head. The head and flagella are articulated together by an inextensible hook connected to a rotary motor to achieve self-propulsion [1–3]. In the oceans and waterways, small unicellular organisms, called diatoms, can assemble into colonies with complex articulations. The species *Bacillaria Paxillifer* forms colonies of stacked rectangular cells that slide along each other while remaining parallel. Their intriguing coordinated motion leads to beautiful and nontrivial trajectories at the scale of the colony [4–6]. Inspired by Nature, scientists have designed articulated systems that can achieve locomotion. Artificial microswimmers use a plethora of swimming gaits such as the undulation of hinged rigid segments, or colloidal beads linked by DNA, under the action of a magnetic field [7–9], articulated arms moving in a non-reciprocal manner to mimic elementary propulsion mechanisms, such as the well-known Purcell’s swimmer [10,11], or more elaborate strategies such as the four-arm breaststroke of the *Copepod* zooplankton [12,13]. Beyond microswimmers, small articulated systems are investigated for the design of smart composite particles and new functional materials. Using DNA functionalized colloids and emulsions respectively, Sacanna et al. [14] and McMullen et al. [15] both showed that polymeric freely-jointed molecules can be assembled at the micron scale.

Joint articulations are also encountered when discretizing inextensible objects, such as fibres or membranes, with numerical methods. As shown in the literature [16–18], the inextensibility condition along a fibre centerline can be discretized as a set of ball-and-socket joints, i.e. only allowing rotation, between discrete degrees of freedom. Regardless of their classification, constraints in articulated bodies can either form open chains, such as filaments, closed-loop chains, such as membranes, or a combination of both.

Simulating suspensions of such a wide variety of articulated bodies requires to solve the hydrodynamic interactions between large collections of rigid objects of arbitrary shape while satisfying multiple constraints. Developing efficient methods to solve for the constrained motion of immersed objects is an active subject of research. One of the main challenges in this task is to build scalable solvers for the hydrodynamic and constraint equations. To solve the hydrodynamic problem, numerical methods, such as the Boundary Element method [19,20] or the rigid multiblob method [21,22], discretize particles of arbitrary shape with discrete degrees of freedom, or markers, constrained to move as a single rigid body. The hydrodynamic interactions between discrete degrees of freedom are materialized by a dense mobility matrix. The mobility matrix is a position-dependent linear operator that relates the forces applied on the markers to their velocities. Thanks to key advances

made in the past decade, the action of the mobility matrix on a vector, a bottleneck in terms of computational cost, can be computed in a fast and scalable way. These approaches rely on fast summation techniques such as the FMM [23–25] or the FFT, and include Ewald summation [26,27], Accelerated and Fast Stokesian Dynamics [28,29], in addition to the Immersed Boundary [30] and similar methods, such as the Force Coupling Method [31,32], that make use of fast, matrix-free, solvers.

Following classical mechanics, the coupling between the constraints and the hydrodynamic part is achieved through a set of Lagrange multipliers that are translated into constraint forces on the bodies through the Jacobian of the constraint equations. The way these Lagrange multipliers are determined depends on the constraint formulation. When the constraints are holonomic, i.e. they depend on the particle positions, orientations and time, finding the Lagrange multipliers requires solving a nonlinear system in the particle positions and orientations [33]. In this case, the constraint forces are given by the product between the vector of Lagrange multipliers and the Jacobian of the constraints with respect to the particle's positions and orientations. If the constraints are nonholonomic, linear and exact, i.e. they depend linearly on the particle velocities and are exact differential equations with respect to time, then one can use the linearity of Stokes equations between forces and velocities to find the Lagrange multipliers directly. When inverting the resulting linear system, one computes the *constraint resistance matrix* that involves the mobility matrix between markers [17]. The corresponding constraint forces are given by the product between the vector of Lagrange multipliers and the Jacobian of the constraints with respect to the particles' velocities.

Up to now, numerical methods for suspensions of articulated bodies have been specifically devoted to the modelling of active and passive inextensible filaments as linear chains of freely-jointed spherical bodies. Most of them are built upon the velocity (i.e. nonholonomic) formulation of the free-joint inextensibility constraints and use analytical expressions of the mobility matrix between the spherical bodies distributed along the filament centerline [34,16,35,36,17]. Even though the velocity formulation combines nicely with the linearity of the mobility problem, these methods rely on direct dense linear algebra tools to find the set of Lagrange multipliers, and therefore scale badly with the number of fibres. Schoeller *et al.* [18] recently overcame that limitation by using holonomic inextensibility constraints with an efficient nonlinear solver to simultaneously find the Lagrange multipliers and the implicit update of the sphere positions and orientations. Their iterative scheme, combined with the Force Coupling Method [31,32] on a grid-based solver, allowed them to simulate large suspensions with up to 1000 filaments for a wide variety of applications [37,18]. Though scalable to many particles, their method is specifically calibrated for linear chains of spherical bodies connected with free-joint constraints. Articulated bodies made of nonspherical units have been employed to study the dynamics of a single bacterium [38,39] or parasite [40] with the Boundary Element Method (BEM). However the constraint formulations used in these works are not generic, in the sense that they are specific to these micro-swimmers, and BEM can bear very small numbers of particles due to high computational costs.

Another drawback of the nonholonomic formulation of the constraints is that, even if the body velocities satisfy the constraints exactly, the local truncation errors due to discrete time-integrators for the body equations of motion, can accumulate and thus violate the position and orientation constraints. Various workarounds have been proposed such as recursive time-step reductions [35], which can significantly increase the computational cost, or successive position adjustments [41].

In this study, we provide a robust, comprehensive and scalable framework to handle large collections of articulated bodies composed of particles with arbitrary shape connected in an arbitrary fashion. The mobility problem between rigid bodies of arbitrary shape is discretized with the rigid multiblob model, which is formulated as a symmetric saddle system involving the mobility matrix between markers [22]. Expanding on our previous work [17], we combine the mobility problem with a velocity formulation of the constraints, resulting in a mixed mobility-resistance problem for the unknown body velocities and constraint Lagrange multipliers. Instead of directly solving the corresponding linear system, we use an iterative method with a block-diagonal preconditioner. Since iterative solvers only require computing the action of the mobility matrix onto a vector, our framework works with any fast method. We demonstrate the efficiency of the preconditioner on various configurations involving articulated bodies with open and closed loops, such as deformable filaments and shells. While resistance problems are known to scale poorly with the particle number, our preconditioned iterative solver is not sensitive to the system size, therefore allowing to simulate large suspensions with quasilinear computational cost. The number of degrees of freedom in the system is then reduced by extending the well-known robot-arm model to the more general case of hybrid chains, so that only one reference position and the bodies orientations are needed to track and uniquely reconstruct an articulated body. Our new reconstruction scheme avoids constraint violations for open chains but does not fully prevent them for closed loops or hybrid chains. In order to prevent any constraint violation, e.g. due to time discretization errors, the particle positions and orientations are corrected at the end of each time-step. Our correction procedure, based on a nonlinear minimisation algorithm, is negligible in terms of computational cost and preserves the accuracy of the time-integration scheme. Besides scaling and convergence tests on shells and filaments, we illustrate the versatility of the method with various simulations of active systems. We first explore the locomotion modes of a model microswimmer made of sliding rigid rods, directly inspired by the diatom colony *Bacillaria Paxillifer*. Very little, if nothing, is known about the swimming mechanisms of this species. Our hydrodynamic simulations, seemingly the first ones of such microorganism, show that, contrary to flagellar dynamics, the swimming speed and direction of the colony depend nonmonotonically on the wavelength of its deformation wave. We also study the swimming speed of a model bacterium, made of a rigid helical flagellum and spherical head, as a function of the helical wave number and compare our results with the theoretical work of Higdon [42]. Finally we simulate the collective dynamics of large, fully resolved, bacterial suspensions near a surface at a scale that, to the best of our knowledge, had never been reached for such systems.

Our framework has been implemented in a collaborative publicly available code on GitHub (<https://github.com/stochasticHydroTools/RigidMultiblobsWall>). Our tool can handle different populations of articulated bodies simultaneously where both passive and time-dependent constraints are directly read from a simple input file. Thanks to its simplicity and flexibility, the user can readily use it to study physical and biological systems involving large collections of articulated bodies.

## 2. Model and formulation

In this section we first outline the equations of the mobility problem that govern the motion of rigid bodies of arbitrary shape freely suspended in a viscous fluid. Then we write the equations of the position constraints that connect these bodies. Expanding on the work of some of us [17], we derive the velocity form of these constraints, which we combine to the mobility problem to propose a new formulation of the *constrained mobility problem*. Finally we show how to use our formulation with existing methods to solve the mobility problem, such as the rigid multiblob model [22].

### 2.1. Continuum formulation for rigid bodies

We first introduce the formulation for a suspension of disconnected rigid bodies and then we generalize it to introduce constraints. Let  $\{\mathcal{B}_p\}_{p=1}^M$  be a set of  $M$  rigid bodies immersed in a Stokes flow. The configuration of each rigid body  $p$  is described by the location of a tracking point,  $\mathbf{q}_p$ , and its orientation represented by the unit quaternion  $\theta_p$ , or in compact notation  $\mathbf{x}_p = \{\mathbf{q}_p, \theta_p\}$ . The linear and angular velocities of the tracking point are  $\mathbf{u}_p$  and  $\boldsymbol{\omega}_p$ . The external force and torque applied to a rigid body are  $\mathbf{f}_p$  and  $\boldsymbol{\tau}_p$ . We will use the concise notation  $\mathbf{U}_p = \{\mathbf{u}_p, \boldsymbol{\omega}_p\}$  and  $\mathbf{F}_p = \{\mathbf{f}_p, \boldsymbol{\tau}_p\}$  as well, while unscripted vectors will refer to the composite vector formed by the variables of all the bodies, e.g.  $\mathbf{U} = \{\mathbf{U}_p\}_{p=1}^M$ . The velocity and pressure,  $\mathbf{v}$  and  $p$ , of the fluid with viscosity  $\eta$  are governed by the Stokes equations [43,19]

$$-\nabla p + \eta \nabla^2 \mathbf{v} = \mathbf{0}, \tag{1}$$

$$\nabla \cdot \mathbf{v} = 0, \tag{2}$$

while at the bodies surface the fluid obeys the no-slip condition [19]

$$\mathbf{v}(\mathbf{r}) = (\mathcal{K}\mathbf{U})(\mathbf{r}) = \mathbf{u}_p + \boldsymbol{\omega}_p \times (\mathbf{r} - \mathbf{q}_p) \text{ for } \mathbf{r} \in \partial\mathcal{B}_p, \tag{3}$$

where we have introduced the geometric operator  $\mathcal{K}$  that transforms rigid body velocities to surface velocities. Since inertia does not play a role in Stokes flows the conservation of linear and angular momentum reduces to the balance of force and torque. For every body  $p$  the balance between hydrodynamic and external stresses is given by [19]

$$\int_{\partial\mathcal{B}_p} \boldsymbol{\lambda}(\mathbf{r}) dS_{\mathbf{r}} = \mathbf{f}_p, \tag{4}$$

$$\int_{\partial\mathcal{B}_p} (\mathbf{r} - \mathbf{q}_p) \times \boldsymbol{\lambda}(\mathbf{r}) dS_{\mathbf{r}} = \boldsymbol{\tau}_p, \tag{5}$$

where  $-\boldsymbol{\lambda}$  is the hydrodynamic traction exerted on the bodies by the fluid. The adjoint of the geometric operator  $\mathcal{K}$  can be used to write the balance of force and torques for all bodies as  $\mathcal{K}^T \boldsymbol{\lambda} = \mathbf{F}$ .

Given the force and torque acting on the rigid bodies, the equations (1)-(5) can be solved for the bodies velocities and the traction. Since the Stokes equations are linear we can write the velocity of  $M$  rigid bodies as

$$\mathbf{U} = \mathbf{N}\mathbf{F}, \tag{6}$$

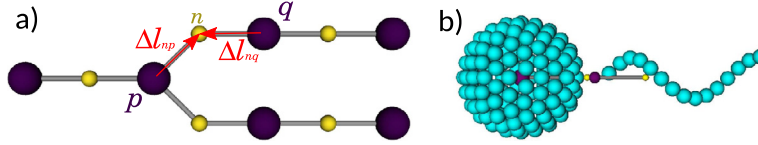
where  $\mathbf{N} = \mathbf{N}(\mathbf{x})$  is a  $6M \times 6M$  mobility matrix that couple the forces and torques acting on the rigid bodies with their velocities. Having  $\mathbf{U}$ , the equations of motion can be integrated in time. Some care is necessary to integrate the quaternion equations, therefore we discuss briefly some of their main properties [44,45]. A unit quaternion,  $\theta = [s, \mathbf{p}]$  with  $s \in \mathbb{R}$  and  $\mathbf{p} \in \mathbb{R}^3$ , represents a rotation around a fixed axis: the finite rotation given by the vector  $\boldsymbol{\gamma}$  has the associated unit quaternion

$$\theta_{\boldsymbol{\gamma}} = [\cos(\gamma/2), \sin(\gamma/2)\boldsymbol{\gamma}/\gamma], \tag{7}$$

where  $\gamma = \|\boldsymbol{\gamma}\|_2$ . Unit quaternions can be combined by the quaternion multiplication

$$\theta_3 = \theta_2 \bullet \theta_1 = [s_2 s_1 - \mathbf{p}_2 \cdot \mathbf{p}_1, s_2 \mathbf{p}_1 + s_1 \mathbf{p}_2 + \mathbf{p}_2 \times \mathbf{p}_1]. \tag{8}$$

Therefore,  $\theta_3$  represents the rotation obtained by the combination of a rotation  $\theta_1$  followed by a rotation  $\theta_2$ . This product rule allows to write the kinematic equations of motion as



**Fig. 1.** a) Example of a branched filament modelled as an articulated body formed by six spherical bodies (large spheres) connect by joints (small spheres). The links connecting two rigid bodies,  $p$  and  $q$ , through the joint  $n$  are labelled in red. b) Bacterium model, its body and flagellum are discretized with blobs (blue spheres) and the tracking points of the body and flagellum (dark spheres) are connected by two links through joints (small spheres). The links fix the flagellum attachment point and the axis of rotation, see Sec. 4.4. (For interpretation of the colours in the figure(s), the reader is referred to the web version of this article.)

$$\frac{d\mathbf{q}_p}{dt} = \mathbf{u}_p, \quad (9)$$

$$\frac{d\theta_p}{dt} = \frac{1}{2} [\mathbf{0}, \boldsymbol{\omega}_p] \bullet \boldsymbol{\theta}_p. \quad (10)$$

## 2.2. Constrained mobility problem

Our constraint formulation is inspired by the robot-arm parametrization [46] generalised to articulated bodies with an arbitrary number of links, including loops. In the following,  $\mathcal{A}_a$  will denote an assembly of  $M_a$  rigid bodies forming an articulated rigid body with  $P_a$  links, so that  $M = \sum_{a=1}^N M_a$  and  $P = \sum_{a=1}^N P_a$  correspond to the total number of rigid bodies and links respectively. We also define  $\mathcal{L}_p$ , the set of links attached to the rigid body  $p$ . Each inelastic link connects two bodies by a hinge joint, see Fig. 1. The links, while inextensible, can rotate. Therefore, the  $n^{\text{th}}$  link, connecting bodies  $p$  and  $q$ , enforces the vectorial constraint

$$\mathbf{g}_n(\mathbf{x}) = \mathbf{q}_p + \mathbf{R}(\theta_p)\Delta\mathbf{l}_{np} - \mathbf{q}_q - \mathbf{R}(\theta_q)\Delta\mathbf{l}_{nq} = \mathbf{0}, \quad (11)$$

where  $\Delta\mathbf{l}_{np}$  is the vector from the body  $p$  to the hinge  $n$  in the body frame of reference. This vector is then rotated to the laboratory frame of reference by the rotation matrix  $\mathbf{R}(\theta_p)$ , see Eq. (B.3). Passive links are time independent in the body frame of reference, however, we will allow for time dependencies, i.e.  $\Delta\mathbf{l}_{np} = \Delta\mathbf{l}_{np}(t)$ , to simulate active links.

For systems with  $M$  rigid bodies and  $P$  links it is convenient to write all the constraints in one equation

$$\mathbf{g} = \mathbf{P}\mathbf{q} + \mathbf{z}(\boldsymbol{\theta}) = \mathbf{0}. \quad (12)$$

The vector  $\mathbf{z}_n(\boldsymbol{\theta}) = \mathbf{R}(\theta_p)\Delta\mathbf{l}_{np} - \mathbf{R}(\theta_q)\Delta\mathbf{l}_{nq}$  carries the orientation dependent terms while the sparse matrix  $\mathbf{P}$ , of size  $3P \times 3M$ , encodes the bodies connectivity. If bonds are neither formed nor broken during a simulation  $\mathbf{P}$  remains constant.

The nonlinearity of (12) with respect to the particles' orientations poses a challenge to integrate the equations of motion [46,18]. However, as the constraints are holonomic their time derivatives are linear in the velocities of the rigid bodies

$$\dot{\mathbf{g}}_n = \mathbf{u}_p + \boldsymbol{\omega}_p \times (\mathbf{R}(\theta_p)\Delta\mathbf{l}_{np}) - [\mathbf{u}_q + \boldsymbol{\omega}_q \times (\mathbf{R}(\theta_q)\Delta\mathbf{l}_{nq})] - \mathbf{B}_n(t) = \mathbf{0}, \quad (13)$$

where  $\mathbf{B}_n(t) = \mathbf{R}(\theta_q)\dot{\Delta\mathbf{l}}_{nq}(t) - \mathbf{R}(\theta_p)\dot{\Delta\mathbf{l}}_{np}(t)$  is zero for passive links. As working with linear constraints is much simpler we will use (13) to derive the equations of motion. Once more we introduce a more compact notation

$$\dot{\mathbf{g}} = \mathbf{C}\mathbf{U} - \mathbf{B}(t) = \mathbf{0}, \quad (14)$$

where the matrix  $\mathbf{C} = (\partial\dot{\mathbf{g}}/\partial\mathbf{U})$  is the constraints' Jacobian. Each constraint exerts a force and torque on the rigid bodies which in the case of passive links,  $\mathbf{C}\mathbf{U} = \mathbf{B}(t) = \mathbf{0}$ , generate no work [33]. This condition is enough to determine the structure of the constraint generalized forces,  $\mathbf{F}^C = \mathbf{C}^T\boldsymbol{\phi}$ , where  $\boldsymbol{\phi}$  is a Lagrange multiplier [17]. It is easy to verify that the generated power,  $dW = \mathbf{U} \cdot \mathbf{F}^C = \boldsymbol{\phi} \cdot \mathbf{B}(t)$ , indeed vanishes for passive links and that the constraint forces do not exert any net force or torque on the whole articulated body, see Appendix A. For articulated bodies the balance of force and torque reads

$$\mathcal{K}^T\boldsymbol{\lambda} - \mathbf{C}^T\boldsymbol{\phi} = \mathbf{F}. \quad (15)$$

Equations (1)-(3) together with (14)-(15) form a linear system that describe the dynamics of articulated rigid bodies immersed in a Stokes flow. At this point, it is worth mentioning that, without loss of generality, the constraint equation (11) can apply to a single spatial component instead of being vectorial. In addition, it can involve the position and/or orientation of a single body instead of two, without changing the formalism presented above.

It is interesting to use (6) to write the linear system with constraints as

$$\begin{bmatrix} -\mathbf{N}\mathbf{C}^T & \mathbf{I} \\ \mathbf{0} & \mathbf{C} \end{bmatrix} \begin{bmatrix} \boldsymbol{\phi} \\ \mathbf{U} \end{bmatrix} = \begin{bmatrix} \mathbf{N}\mathbf{F} \\ \mathbf{B} \end{bmatrix}, \quad (16)$$

where  $\mathbf{I}$  is a  $6M \times 6M$  identity matrix. Any method to solve the Stokes problem, and therefore to apply the mobility matrix  $\mathbf{N}$ , can be used with (16).

It can be enlightening to write the formal solution of the linear system (16) to gain some physical intuition on the equations. First, in the absence of links ( $\mathbf{C} = \mathbf{0}$ ) we can write the velocity of the rigid bodies as  $\hat{\mathbf{U}} = \mathbf{N}\mathbf{F}$ . Then, when the bodies are linked, the velocity can be written as

$$\begin{aligned} \mathbf{U} &= \hat{\mathbf{U}} + \mathbf{N}\mathbf{C}^T\boldsymbol{\phi}, \\ &= [\mathbf{I} - \mathbf{N}\mathbf{C}^T\mathbf{G}\mathbf{C}]\hat{\mathbf{U}} + \mathbf{N}\mathbf{C}^T\mathbf{G}\mathbf{B}(t), \end{aligned} \tag{17}$$

where  $\mathbf{G} = (\mathbf{C}\mathbf{N}\mathbf{C}^T)^{-1}$  is the *constraint resistance matrix*. Both lines of (17) are enlightening. The first line shows that the unconstrained velocity is rectified by the flow generated by the constraint forces. It also shows that the linear system becomes a mixed mobility-resistance problem, as the constraint forces,  $\boldsymbol{\phi}$ , are unknowns of the linear system. The second line shows that for passive links, i.e.  $\mathbf{B}(t) = \mathbf{0}$ , the constrained velocity can be found by projecting  $\hat{\mathbf{U}}$  to the space of admissible velocities. Note that the matrix  $\mathbf{Q} = \mathbf{I} - \mathbf{N}\mathbf{C}^T\mathbf{G}\mathbf{C}$  is a projection operator, i.e.  $\mathbf{Q}^2 = \mathbf{Q}$ .

### 2.3. Rigid multiblob model as a Stokes solver

Solving the mobility-resistance problem applying an iterative solver to (16) could be quite expensive as, in general, computing the action of the mobility  $\mathbf{N}$  requires solving a linear system of its own. We seek a formulation that avoids nested linear systems to reduce the computational cost. For that reason we will solve the Stokes problem and the constraint equations simultaneously. To solve the hydrodynamic problem we rely on the rigid multiblob method [22]. We discretize the surface of each rigid body by a set of markers or *blobs* of finite radius  $a$  and position  $\mathbf{r}_i$ . Fig. 1b shows an example of discretization of a bacterium with a spherical head and a helical flagellum. Each blob is subject to a force,  $\lambda_i$ , that enforces the rigid motion of the body. Thus, the integrals in the balance of force and torque (15) become sums over the blobs

$$\sum_{i \in \mathcal{B}_p} \lambda_i - \sum_{n \in \mathcal{L}_p} \phi_n = \mathbf{f}_p, \tag{18}$$

$$\sum_{i \in \mathcal{B}_p} (\mathbf{r}_i - \mathbf{q}_p) \times \lambda_i - \sum_{n \in \mathcal{L}_p} (\Delta \mathbf{l}_{np} - \mathbf{q}_p) \times \phi_n = \boldsymbol{\tau}_p, \tag{19}$$

where the second sum runs over the links  $\mathcal{L}_p$  attached to the rigid body  $p$  and  $\phi_n$  is the constraint Lagrange multiplier acting on link  $n$ .

The no-slip condition, as in collocation methods [19], is evaluated at each blob

$$\mathbf{v}(\mathbf{r}_i) = \sum_j \mathbf{M}_{ij} \lambda_j = \mathbf{u}_p + \boldsymbol{\omega}_p \times (\mathbf{r}_i - \mathbf{q}_p) \text{ for all } i \in \mathcal{B}_p, \tag{20}$$

where the mobility matrix  $\mathbf{M}$  mediates the hydrodynamic interactions. The term  $\mathbf{M}_{ij}$  couples the force acting on the blob  $j$  to the velocity of blob  $i$ . In a first-kind boundary integral formulation the mobility would be simply the Green's function of the Stokes equation [19,47]. The rigid multiblob method uses instead a regularization of the Green's function, the so-called Rotne-Prager approximation [48], which can be written as the double integral of the Green's function,  $\mathbf{G}(\mathbf{r})$ , over the blobs surface [49]

$$\mathbf{M}_{ij} = \mathbf{M}(\mathbf{r}_i, \mathbf{r}_j) = \frac{1}{(4\pi a^2)^2} \int \delta(|\mathbf{r}' - \mathbf{r}_i| - a) \mathbf{G}(\mathbf{r}', \mathbf{r}'') \delta(|\mathbf{r}'' - \mathbf{r}_j| - a) d^3 r'' d^3 r', \tag{21}$$

where  $\delta(\mathbf{r})$  is the Dirac's delta function and  $a$  is the blob radius. The advantage of (21) over the non-regularized Green's function is that the RPY mobility is always positive definite, even when blobs overlap. This property makes the numerical method quite robust and easy to implement. There are analytical expressions to calculate the RPY mobility in some geometries such as unbounded spaces [48,49] or above an infinite no-slip wall [50]. Moreover, there are fast methods to compute the product  $\mathbf{M}\boldsymbol{\lambda}$  in quasilinear time in those domains [51,52] as well as in periodic domains [27,31].

The whole linear system to solve the constrained mobility problem is found by combining (18)-(20),

$$\underbrace{\begin{bmatrix} \mathbf{M} & -\mathbf{K} & \mathbf{0} \\ -\mathbf{K}^T & \mathbf{0} & \mathbf{C}^T \\ \mathbf{0} & \mathbf{C} & \mathbf{0} \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix} \boldsymbol{\lambda} \\ \mathbf{U} \\ \boldsymbol{\phi} \end{bmatrix}}_{\mathbf{y}} = \underbrace{\begin{bmatrix} \mathbf{u}_s \\ -\mathbf{F} \\ \mathbf{B} \end{bmatrix}}_{\mathbf{b}}. \tag{22}$$

In the linear system the matrix  $\mathbf{K}$  is a discretization of the operator  $\mathcal{K}$ , so it transforms the rigid body velocities to blobs velocities. Note that in the right hand (RHS) side we have included a slip velocity,  $\mathbf{u}_s$ , on the blobs. To solve (22) with

iterative methods, such as GMRES, it is necessary that the implementation of the equations can handle nonzero slips. Besides,  $\mathbf{u}_s$  can be used to model active slip on the bodies, e.g. to model a squirmer [53] or a phoretic particle [54,55]. The solution of (22) is identical to (17) except that the unconstrained body velocities now contain a slip-induced term

$$\widehat{\mathbf{U}} = \mathbf{N}\mathbf{F} - \mathbf{N}\mathbf{K}^T \mathbf{M}^{-1} \mathbf{u}_s \quad (23)$$

where  $\mathbf{N} = (\mathbf{K}^T \mathbf{M}^{-1} \mathbf{K})^{-1}$  is the body mobility matrix given by the rigid multiblob model.

#### 2.4. Constraints with single blobs

For some specific applications it is convenient to represent rigid bodies as single blobs. For example, an inextensible filament can be modelled by discretizing the centerline with blobs [34,17,18] or a sheet can be formed by blobs linked to their nearest neighbours [56].

At this level of approximation the application of the mobility matrix,  $\mathbf{N}$ , simply requires computing hydrodynamic interactions between blobs, including the coupling with the rotational degrees of freedom. This can be achieved with analytical approximations of the generalized RPY tensor [49] or by using grid-based Stokes solvers with the Force Coupling Method [32] or similar techniques. Therefore, the computational cost is much lower and the velocities of the rigid articulated bodies can be found by solving (16) with an iterative method such as GMRES. We will use this approach in Section 3.2 to model filaments and spherical deformable shells.

### 3. Preconditionner and iterative solver convergence

In this section we first present the iterative method used to solve the constrained mobility problem (22). In order to ensure and speed-up the convergence of the iterative solver, we use a new block diagonal preconditioner that generalizes one introduced for freely suspended, unconstrained, rigid bodies [22]. Then we evaluate its effectiveness on linear articulated bodies, such as filaments, and closed ones with loops, such as shells. While these two examples cover a broad range of interesting applications, this section focuses on the solver performances rather than the physical phenomena at stake.

#### 3.1. Block diagonal preconditioner

In the following, we use the preconditioned GMRES method to solve the constrained mobility problem (22) iteratively. Due to the long-ranged nature of hydrodynamic interactions at low Reynolds numbers, the condition number of the matrix  $\mathbf{A}$  in Eq. (22) increases with the blob number and volume fraction, which slows down the convergence of iterative solvers for large and/or concentrated suspensions.

As shown in our previous work [22], the convergence on the unconstrained mobility problem can be improved with a very effective, yet simple, left preconditioner  $\widetilde{\mathbf{A}}^{-1}$  considering non-interacting bodies. Applying  $\widetilde{\mathbf{A}}^{-1}$  to the unconstrained mobility problem with the rigid multiblob method amounts to solving the approximate problem [22]

$$\begin{bmatrix} \widetilde{\mathbf{M}} & -\mathbf{K} \\ -\mathbf{K}^T & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{\lambda} \\ \mathbf{U} \end{bmatrix} = \begin{bmatrix} \mathbf{u}_s \\ -\mathbf{F} \end{bmatrix} \quad (24)$$

where  $\widetilde{\mathbf{M}}^{(pq)} = \delta_{pq} \mathbf{M}^{(pp)}$  is nonzero only for blobs belonging to the same body  $p$ . The corresponding body mobility matrix  $\widetilde{\mathbf{N}}^{(pq)}$  is block diagonal, where each block refers to a single body neglecting all hydrodynamic interactions with other bodies

$$\widetilde{\mathbf{N}}^{(pq)} = \delta_{pq} \left[ \left( \mathbf{K}^{(p)} \right)^T \mathbf{M}^{(pp)} \mathbf{K}^{(p)} \right]^{-1}. \quad (25)$$

Doing so, we found that, for freely suspended bodies, the number of iterations becomes independent of the number of bodies, and slightly increases with the volume fraction [22].

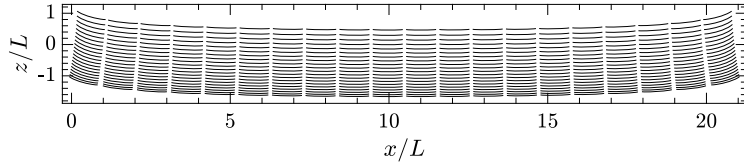
In this work we apply this approach to the constrained system (22), which amounts to solving the approximate problem

$$\begin{bmatrix} \widetilde{\mathbf{M}} & -\mathbf{K} & \mathbf{0} \\ -\mathbf{K}^T & \mathbf{0} & \mathbf{C}^T \\ \mathbf{0} & \mathbf{C} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{\lambda} \\ \mathbf{U} \\ \boldsymbol{\phi} \end{bmatrix} = \begin{bmatrix} \mathbf{u}_s \\ -\mathbf{F} \\ \mathbf{B} \end{bmatrix}. \quad (26)$$

As a result, the approximate problem can be solved for each articulated body separately, and in parallel, since the resulting approximate constraint resistance matrix is block diagonal for each assembly

$$\widetilde{\mathbf{G}}^{(\mathcal{A}_a \mathcal{A}_b)} = \delta_{ab} \left[ \mathbf{C}^{(\mathcal{A}_a)} \widetilde{\mathbf{N}}^{(\mathcal{A}_a \mathcal{A}_a)} \left( \mathbf{C}^{(\mathcal{A}_a)} \right)^T \right]^{-1}. \quad (27)$$

If the matrix  $\mathbf{C}$  does not have full row rank,  $\widetilde{\mathbf{G}}$  is evaluated with the pseudo-inverse.



**Fig. 2.** Snapshot of a sedimenting array of  $20 \times 20 = 400$  filaments, discretized with  $M_a = 15$  spheres, bended by the hydrodynamic drag and the flow disturbances induced by their neighbours [18]. Snapshot at dimensionless time  $t \times U_z/L = 0.09$ , where  $U_z = -3.26\mu\text{m/s}$  is the velocity of an isolated straight filament subject to the gravitational force.

### 3.2. Convergence results

In the following we evaluate the effectiveness of the preconditioned iterative solver on linear articulated bodies, such as filaments, and closed ones with loops and branches such as shells. We show that, just like unconstrained bodies, the number of iterations does not grow with the system size and therefore that our scheme is effective to perform many body simulations of articulated bodies.

#### 3.2.1. Settling filaments

We consider an array of initially straight, inextensible, and deformable filaments (i.e. with zero bending stiffness) sedimenting in the  $(x, z)$ -plane in free space, with gravity pointing in the  $-z$  direction. The dynamics of sedimenting fibres has been thoroughly investigated at the individual [57,17] and collective level [58–60,18,61]. Here we use this example exclusively to investigate the performance of our iterative solver with linear chains of constraints and refer the reader to the cited articles for more details on the modelling aspects and the physics of the system.

Each filament  $\mathcal{A}_a$  is oriented along the  $x$ -axis and discretized with  $M_a = 15$  spheres with hydrodynamic radius  $a = 1\mu\text{m}$ . Discretizing filaments with spheres is very common in the literature and it has proven to accurately and efficiently capture their dynamics at the individual and collective level [16,62,35,17,57,18,63]. The inextensibility constraint of a filament states that the derivative of the centerline position  $\mathbf{q}$  with respect to the arclength  $s$  must correspond to a unit vector  $\hat{\mathbf{t}}$  tangent to the centerline:

$$\frac{\partial \mathbf{q}}{\partial s} = \hat{\mathbf{t}}. \tag{28}$$

After discretizing the filament with  $M_a$  bodies and  $M_a - 1$  links, Eq. (28) becomes

$$\mathbf{q}_{p+1} - \mathbf{q}_p - \frac{l_c}{2} (\hat{\mathbf{t}}_p + \hat{\mathbf{t}}_{p+1}) = \mathbf{0}, \forall p \in \mathcal{A}_a, \tag{29}$$

where  $\hat{\mathbf{t}}_p$  is the unit orientation vector of body  $p$  along the filament centerline, and  $l_c$  is the length of the link connecting the centers of body  $p$  and  $p + 1$ . Using the general notation introduced in Section 2.2, Eq. (29) writes

$$\mathbf{q}_{p+1} + \mathbf{R}(\theta_{p+1}) \cdot \Delta \mathbf{l}_{n,p+1} - \mathbf{q}_p - \mathbf{R}(\theta_p) \cdot \Delta \mathbf{l}_{n,p} = \mathbf{0} \tag{30}$$

where the link vectors in the body frame are  $\Delta \mathbf{l}_{n,p+1} = -\Delta \mathbf{l}_{n,p} = -l_c \mathbf{e}_x/2$  for a straight filament initially aligned along the  $x$ -axis.

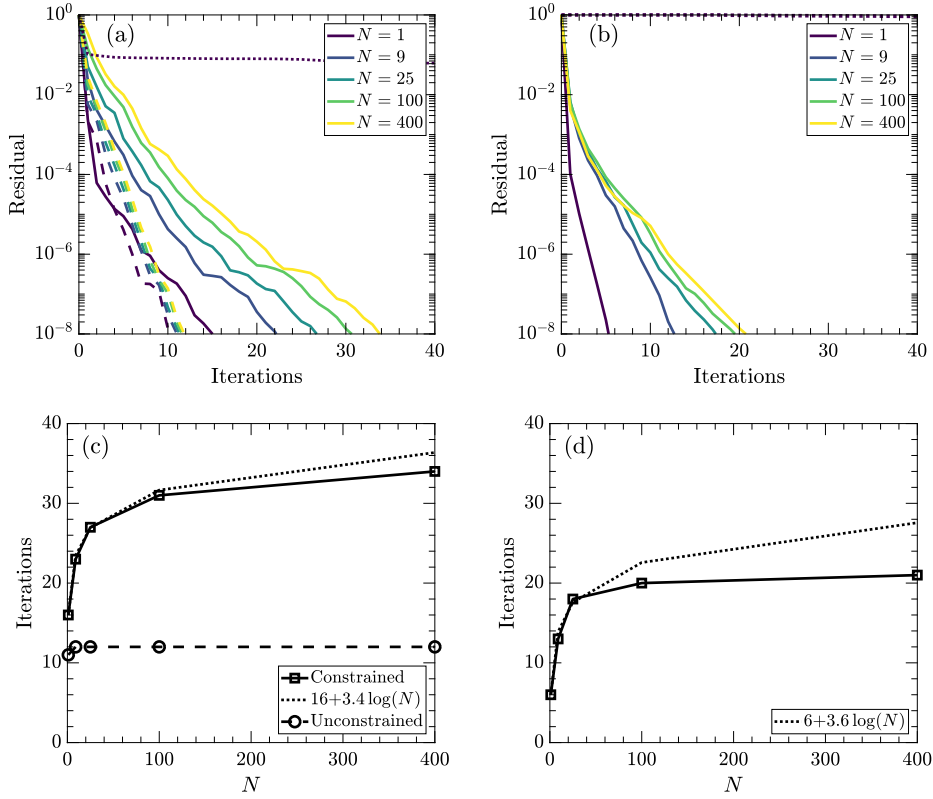
In the following we set the link length between two bodies  $l_c = 2.5a$  so that the filament length and aspect ratio are  $L = (M_a - 1) \times l_c + 2a = 37a$  and  $L/2a = 18.5$  respectively. The total number of filaments in the array is  $N = N_x \times N_z$  where  $N_x$  and  $N_z$  are the number of filaments along each direction. The spacing between filaments along the  $x$  and  $z$  direction is  $4a$ , so that the local area fraction is approximately  $\phi \approx 0.32$ . Each body is subject to a gravitational force along the  $z$ -axis with magnitude  $mg = 2.5 \cdot 10^4 \text{ pN}$ . The solvent viscosity is  $\eta = 10^3 \text{ Pa}\cdot\text{s}$ . Fig. 2 shows a snapshot of the sedimenting array at the area fraction considered in this test.

The spheres making up the filaments are either modelled as single blobs or as a rigid icosahedron made of 12 blobs distributed on the sphere surface. The number of filaments is varied from  $N = 1$  up to  $N = 20 \times 20 = 400$ , which corresponds to a maximum system size of 52, 800 for single blobs and 268, 800 for the multiblob model.

Before evaluating the performance of our preconditioner, we stress that even with only one filament, the GMRES solver converges extremely slowly in the absence of preconditioner, regardless of the discretization (Fig. 3a,b dotted line). We have found that the typical number of iterations required to reach an acceptable tolerance is approximately equal to the system size which leads to a cubic scaling similar to direct methods, hence the need for a good, scalable preconditioner.

Fig. 3 shows the convergence of the preconditioned iterative solver for the two levels of discretization. First we examine the effect of including the inextensibility constraint (30) between spheres on the solver convergence. In the absence of kinematic constraints the spheres are free to move and the preconditioned iterative solver for the multiblob model converges within 10 iterations independently of the number of filaments  $N$  (Fig. 3a), which is expected from our previous work [22]. When the inextensibility constraints are included, the convergence slows down with  $N$ . However, as shown in Fig. 3c, the number of iterations to reach a tight tolerance ( $\epsilon = 10^{-8}$ ) scales sub-logarithmically with the particle number. For the





**Fig. 3.** Linear solver convergence for different number of sedimenting filaments,  $N$ , discretized with  $M_a = 15$  icosahedra (a,c) or blobs (b,d) each. The dashed lines in panels (a,c) correspond to an identical configuration with unconstrained icosahedra. The dotted line in panel (a,b) corresponds to the convergence of GMRES without preconditioner with  $N = 1$  filament. Panels (c,d) show the number of iterations required to reach a tolerance  $\epsilon = 10^{-8}$ . The dotted lines represent a logarithmic fit of the first three data points.

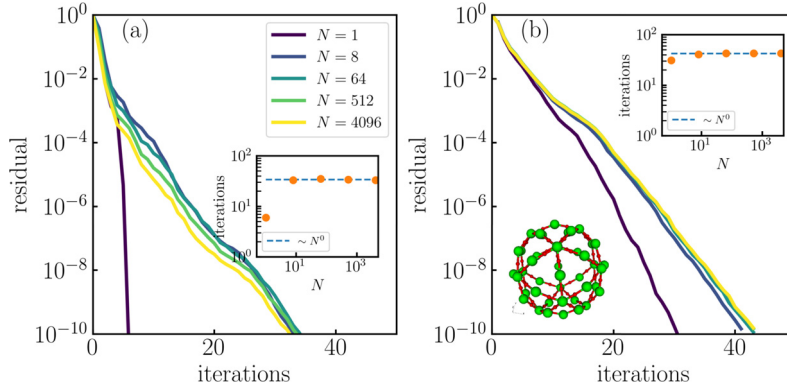
single-blob model (Fig. 3b,d), the convergence rate is faster and quickly becomes independent of the number of filaments. The independence on the particle number seems to be a generic feature of the preconditioner since a similar plateau is observed for arrays of deformable shells, as shown in the next section.

Altogether, these results demonstrate the ability and effectiveness of the preconditioner to handle constrained systems. When all the degrees of freedom of the bodies are constrained, i.e.  $N_c = N_{dof} = 6M$  where  $N_c$  is the number of independent constraints, Eq. (22) becomes a resistance problem, where one needs to invert the whole mobility matrix  $\mathbf{N}$  to obtain the forces corresponding to the prescribed particle velocities. It has been shown that preconditioned iterative solvers for resistance problems scale, at best, sublinearly as  $M^\alpha$ , where  $0 < \alpha < 1$  [64,22]. Our constrained problem is a mixed resistance-mobility problem since the ratio of independent constraints over the number of degrees of freedom of the system is smaller than unity: for a filament discretized with  $M_a \in \llbracket 2, \infty \llbracket$  spheres the ratio is in the range  $N_c/N_{dof} = 3(M_a - 1)/6M_a = 0.25 - 0.5$ . In order to find the constraint forces, one needs to invert the constraint mobility matrix  $\mathbf{CNC}^T$ , which is smaller than  $\mathbf{N}$ . It is therefore interesting to note that when the number of constraints is smaller than the number of degrees of freedom our preconditioned iterative solver becomes independent of the particle number ( $\alpha = 0$ ), which is a significant improvement over sublinear scaling.

Note that in practical applications, a looser tolerance is usually employed ( $\epsilon = 10^{-3} - 10^{-4}$ ), thus lowering the number of iterations to 6 - 14 with the multiblob model and only 3-4 with single blobs, which drastically reduces the computational cost. Note also that, in dynamic simulations, the iterative solver uses the solution from the previous time-step as a first guess. By doing so, the number of iterations is further reduced: in the example shown in Fig. 2 only 1-2 iterations were necessary to reach  $\epsilon = 10^{-4}$  with the single blob discretization.

### 3.2.2. Deformable shells

We further test the robustness of our preconditioner on deformable shells (i.e. with zero bending stiffness) where, contrary to filaments, the links form loops. Such complex constraints arrangements appear in any discretization of inextensible surfaces, like membranes, but here we just focus on the solver performance and not on the physical aspects of the model. Each shell is discretized with  $M_a = 42$  blobs with minimal spacing  $l_c \approx 0.25$  times the shell diameter. The blobs are linked to their first neighbours. Most blobs have only two first neighbours but some have five so that the total number of links per shell is  $P_a = 60$ , see inset of Fig. 4b. For an articulated shell, the ratio of the number of constraints over the number of



**Fig. 4.** Linear solver convergence for different number of shells,  $N$ , under (a) constant force or (b) random forces and torques. Insets show the number of iterations required to reach a tolerance  $\epsilon = 10^{-10}$ . Shells are discretized with  $M_a = 42$  blobs and  $P_a = 60$  links, see inset in (b).

degrees of freedom is  $N_c/N_{dof} = 3 \times 60 / (42 \times 6) = 0.71$  which is closer to a resistance problem (for which this ratio would be exactly one) compared to the filament case above. We consider a cubic lattice of  $N = 1 - 4096$  shells two diameters apart which corresponds to a local volume fraction of  $\phi \approx 0.065$ . The maximum system size of the linear system, reached for  $N = 4096$ , is 1,769,472. Fig. 4 shows the convergence of the preconditioned iterative solver with two types of forcing on the blobs: a constant force (e.g. due to gravity, Fig. 4a) and random forces and torques (e.g. due to thermal fluctuations, Fig. 4b). As in the filament simulations, the number of iterations to reach a given tolerance does not depend on the number of shells regardless of the forcing type (see insets), even though the ratio constraints/degrees of freedom is closer to unity. Owing to its complexity, the random forcing case requires more iterations to converge than the sedimenting case (e.g. 16 vs. 8 iterations to reach a residual of  $10^{-4}$ ). Altogether, these results confirm the robustness of the preconditioner and scalability of the iterative solver for the constrained mobility problem.

#### 4. Body reconstruction and time-integration

As explained in Section 2.2, the kinematic constraints on the velocities are obtained by taking the time derivative of the position constraints. Thanks to this transformation, the constrained problem is reduced to a single linear system for the body velocities that can be efficiently solved with our preconditioned GMRES solver (cf. Section 3). However, satisfying the velocity constraints does not guarantee that the position constraints will be obeyed with the same accuracy. Indeed, when integrating Eqs. (9)-(10) for the particle positions and orientations, errors of order  $O(\Delta t^{q+1})$ , where  $q$  is the order of convergence of the discrete time integrator, will accumulate at each time step. The error in the position constraints may therefore become uncontrolled as time advances. To circumvent that problem, we propose to track the articulated bodies with a new reconstruction method and a correction procedure that satisfy the position constraint up to an arbitrary precision while preserving the  $O(\Delta t^q)$  global error of the time integration scheme.

##### 4.1. Reconstruction method for general articulated bodies

###### 4.1.1. The robot-arm model for linear chains

A well-known approach to prevent error accumulation in linear chains, such as filaments, is the “robot-arm” model. Reconstructing a robot-arm only requires tracking an arbitrary point of the articulated bodies (e.g. the first body) and the orientation of each link composing the arm. In the case of a single robot-arm with tracking point  $\mathbf{q}_1$ , the position of body  $p > 1$  is given by simply adding the link vectors along the chain

$$\mathbf{q}_p = \mathbf{q}_1 + \sum_{q=1}^{p-1} [\mathbf{R}(\theta_q) \cdot \Delta \mathbf{l}_{q+1,q} - \mathbf{R}(\theta_{q+1}) \cdot \Delta \mathbf{l}_{q,q+1}], \tag{31}$$

where  $\Delta \mathbf{l}_{q,q+1}$  (respectively  $\Delta \mathbf{l}_{q+1,q}$ ) is the vector connecting body  $q$  (resp.  $q + 1$ ) to the hinge between  $q$  and  $q + 1$ . Therefore, the robot-arm parametrization ensures that the body positions satisfy the position constraint exactly regardless of the integration error on the tracking point  $\mathbf{q}_1$  and the orientations  $\theta = \{\theta_p\}_{p=1}^M$ . However, the reconstruction (31) does not work for general articulated bodies with branches and/or loops.

###### 4.1.2. General reconstruction method

Here we present our extension of the robot-arm model to arbitrary constraint arrangements. Since articulated bodies  $\{\mathcal{A}_a\}_{a=1}^N$  can have arbitrary connections, a natural choice for the tracking point is their center of mass (COM)  $\mathbf{q}_{COM} = \{\mathbf{q}_{COM}^a\}_{a=1}^N$ . Therefore, instead of integrating the position for all the rigid bodies (9), the equations to be integrated in time are

$$\frac{d\mathbf{q}_{COM}}{dt} = \mathbf{u}_{COM}, \quad (32)$$

$$\frac{d\theta_p}{dt} = \frac{1}{2} [0, \boldsymbol{\omega}_p] \bullet \boldsymbol{\theta}_p, \quad p = 1, \dots, M, \quad (33)$$

where  $\mathbf{u}_{COM} = \{\mathbf{u}_{COM}^a\}_{a=1}^N$  with

$$\mathbf{u}_{COM}^a = \frac{1}{M_a} \sum_{p \in \mathcal{A}_a} \mathbf{u}_p, \quad (34)$$

the mean velocity of articulated body  $\mathcal{A}_a$ .

A simple way to generalize the robot-arm model for articulated bodies including loops and branches is to solve the connectivity problem given by the matrix form of the constraints (12) for each articulated body separately

$$\mathbf{P}^a \mathbf{q}^a = -\mathbf{z}^a(\boldsymbol{\theta}, t), \quad a = 1, \dots, N, \quad (35)$$

where  $\mathbf{P}^a$ ,  $\mathbf{q}^a = \{\mathbf{q}_p\}_{p \in \mathcal{A}_a}$  and  $\mathbf{z}^a$  are the  $3P_a \times 3M_a$  connectivity matrix,  $3M_a \times 1$  body positions and  $3P_a \times 1$  relative orientations associated to articulated body  $a$ . Since each constraint involves pairs of bodies, and because the number of links per articulated body always satisfies  $P_a \geq (M_a - 1)$ , the rank of  $\mathbf{P}^a$  is  $r = 3(M_a - 1)$  regardless of the constraint arrangement. Therefore Eq. (35) has infinitely many solutions. The nullspace of  $\mathbf{P}^a$  is spanned by  $3M_a - r = 3$  basis vectors. Each of this basis vector contains a single position repeated  $M_a$  times, which corresponds to a shift of the COM of the articulated body. In order to remain in the frame attached to the articulated body, we remove all the nullspace contributions by solving (35) with the pseudo-inverse of  $\mathbf{P}^a$

$$\tilde{\mathbf{q}}^a = -(\mathbf{P}^a)^+ \mathbf{z}^a. \quad (36)$$

One can check that the resulting solution has indeed zero COM:  $\tilde{\mathbf{q}}_{COM}^a = \frac{1}{M_a} \sum_{p \in \mathcal{A}_a} \tilde{\mathbf{q}}_p = \mathbf{0}$ . The articulated body is then reconstructed by adding the position of the COM obtained after integrating (32)

$$\mathbf{q}_p = \tilde{\mathbf{q}}_p + \mathbf{q}_{COM}^a, \quad \forall p \in \mathcal{A}_a, \quad a = 1, \dots, N. \quad (37)$$

Since arbitrary translations of the COM are removed with the pseudo-inverse in (36), the reconstruction (37) is unique regardless of the constraint arrangement: there is only one way to construct an assembly with a given set of links and a given reference position. Therefore, for linear chains, our general reconstruction method is strictly identical to the robot-arm model, regardless of the choice of the tracking point.

#### 4.2. Correction step for discrete time-integrators

If (32)-(33) are integrated exactly in time, then  $\tilde{\mathbf{q}}^a$  satisfies the position constraints and so do the reconstructed positions in (37). However, in practical simulations, (32)-(33) are integrated with a discrete time integrator. After integrating over a time-step, the resulting  $O(\Delta t^{q+1})$  error on the body orientations  $\boldsymbol{\theta}_p$ , is transmitted to the RHS of Eq. (35). If the articulated bodies do not have any loops, e.g. branched filaments, the solution (36) exactly obeys the constraints independently of the time-stepping errors in the RHS. However, when the constraints form loops, the solution  $\tilde{\mathbf{q}}$  might violate the position constraints.

We quantify the constraint violation due to a finite time-step by simulating a deformable shell made of  $M = 42$  blobs and  $P = 60$  links. Each blob constituting the shell is subject to random forces and torques. The equations of motion (32)-(33) are integrated with an Explicit Euler scheme over one time-step. Fig. 5a shows the convergence of the position constraint error, here defined as the infinity norm of the constraint vector  $\|\mathbf{g}\|_\infty$ , with  $\Delta t$  for several linear solver tolerances  $\epsilon = 10^{-6} - 10^{-1}$ . As the preconditioner guarantees that the velocities obey the constraints for any  $\epsilon$ , the constraint violations show second order convergence, as expected for an Explicit Euler integrator. Even though the constraint error remains low and decays as  $\Delta t^2$ , it is preferable to prevent its accumulation and to decorrelate it from the time discretization.

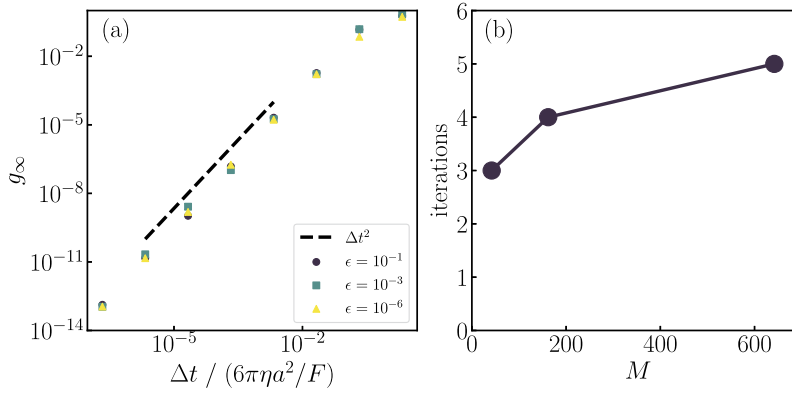
To prevent any constraint violation at the end of the time step we correct the body positions and orientations with small increments,  $\delta \mathbf{q} = \{\delta \mathbf{q}_p\}_{p=1}^M$  and  $\delta \boldsymbol{\theta} = \{\delta \boldsymbol{\theta}_p\}_{p=1}^M$  respectively. These increments are solutions to the minimization problem

$$(\tilde{\delta \mathbf{q}}, \tilde{\delta \boldsymbol{\theta}}) = \arg \min \|\mathbf{g}(\delta \mathbf{q}, \delta \boldsymbol{\theta})\|_2^2 \quad (38)$$

$$\text{s.t. } \|\delta \boldsymbol{\theta}_p\|_2^2 - 1 = 0, \quad \forall p, \quad (39)$$

where

$$\mathbf{g}_n(\delta \mathbf{q}, \delta \boldsymbol{\theta}) = \mathbf{q}_p + \delta \mathbf{q}_p + \mathbf{R}(\delta \boldsymbol{\theta}_p \bullet \boldsymbol{\theta}_p) \Delta \mathbf{l}_{np} - \mathbf{q}_q - \delta \mathbf{q}_q - \mathbf{R}(\delta \boldsymbol{\theta}_q \bullet \boldsymbol{\theta}_q) \Delta \mathbf{l}_{nq}, \quad (40)$$



**Fig. 5.** Constraint errors and nonlinear convergence for a shell formed by blobs under random forces and torques. (a) Infinite norm of the constraint error before the correction step versus the integration time step size  $\Delta t$  for a shell formed by  $M = 42$  blobs and  $P = 60$  links. Using the preconditioner the velocity obeys the constraint for any tolerance  $\epsilon$  and therefore the error decreases as the  $O(\Delta t^2)$  local truncation error of Explicit Euler. (b) Number of iterations in the nonlinear solver to converge to a tolerance  $\delta = 10^{-10}$  versus the number of blobs  $M$ . The number of blobs in the shell is  $M = 42, 162$  and  $642$  and the number of links  $P = 60, 240$  and  $963$  respectively.

is the  $n^{th}$  constraint linking bodies  $p$  and  $q$ . The second equation, (39), constrains the quaternion increments to have unit norm to properly represent rotations.<sup>1</sup>

We solve (38)-(39) using the least squares algorithm implemented in the *Scipy* library [65]. We have observed that the correction is proportional to the local truncation error of the time-integration scheme, thus the nonlinear solver does not affect the integrator global convergence rate. It is possible to add inequalities to the minimization problem to ensure the corrections do not exceed the local truncation error of the time-integration scheme [66]. We use the exact Jacobian of the residual to speedup the convergence, see Appendix B; in this case the number of residual and Jacobian evaluations is equal to the number of iterations plus one. Since the Jacobian is very sparse its use does not increase the memory requirements of the algorithm appreciably. The number of iterations to attain a tight tolerance, e.g.  $\delta = 10^{-10}$ , is almost independent of the size of the articulated body. Fig. 5b shows that the least square solver converges in a small number of iterations even for large articulated bodies with many constraints.

### 4.3. Reconstruction and correction with Explicit Euler

Our reconstruction and correction methods combines effectively with any time integration scheme. For the sake of simplicity, we outline the algorithm for the Explicit Euler integrator. An algorithm with an Explicit Midpoint scheme is provided in Appendix C.

- Initialize positions  $\mathbf{q}(0)$ ,  $\mathbf{q}_{COM}(0)$  and orientations  $\theta(0)$
- Precompute the pseudo-inverse of the connectivity matrix  $\mathbf{P}^+$ .
- Time loop: for  $k = 0, \dots, N_{it} - 1$ 
  1. Solve the linear system (22) at  $t_k = k\Delta t$  to obtain the body velocities  $\mathbf{U}^{(k)}$ .
  2. Compute the translational velocity of the COMs  $\mathbf{u}_{COM}^{(k)}$  using (34).
  3. Update the COM and the orientations of each assembly

$$\mathbf{q}_{COM}^* = \mathbf{q}_{COM}^{(k)} + \Delta t \mathbf{u}_{COM}^{(k)}, \tag{41}$$

$$\theta^* = \theta_{\omega^{(k)} \Delta t} \bullet \theta^{(k)}. \tag{42}$$

4. Reconstruct the articulated bodies using the pseudo-inverse (36), where  $\mathbf{z}^a = \mathbf{z}^a(\theta^*, t_{k+1})$  and then apply Eq. (37) to obtain the body positions  $\mathbf{q}^*$  attached to the updated centers of mass.
5. Evaluate the updated constraint vector  $\mathbf{g}^* = \mathbf{g}(\mathbf{q}^*, \theta^*, t_{k+1})$  and check constraint violation:
  - if  $\|\mathbf{g}^*\|_2^2 < \delta$  then

$$\mathbf{q}^{(k+1)} \leftarrow \mathbf{q}^*, \tag{43}$$

$$\theta^{(k+1)} \leftarrow \theta^*, \tag{44}$$

else

<sup>1</sup> We could have written the minimization problem with rotation vectors  $\mathbf{y}_p$  so that  $\delta\theta_p = \theta \mathbf{y}_p = [\cos(\gamma_p/2), \sin(\gamma_p/2)\hat{\mathbf{y}}_p]$ , the quaternion obtained from the vector  $\mathbf{y}_p = \gamma_p \hat{\mathbf{y}}_p$ , automatically satisfies the unit norm constraint. However, we found that approach to be less effective in terms of computational cost due to the cumbersome expressions of the Jacobian matrix  $\mathbf{J} = \partial \mathbf{g} / \partial \delta \mathbf{x}$ .

- Solve the minimization problem (38)-(39) with the nonlinear solver to find the increments  $(\check{\delta}\mathbf{q}, \check{\delta}\boldsymbol{\theta})$ .
- Correct the body positions and orientations

$$\mathbf{q}^{(k+1)} = \mathbf{q}^* + \check{\delta}\mathbf{q}, \quad (45)$$

$$\boldsymbol{\theta}^{(k+1)} = \check{\delta}\boldsymbol{\theta} \bullet \boldsymbol{\theta}^*. \quad (46)$$

We note that the correction step is not necessary for open chains since our reconstruction method satisfies the constraints regardless of the time discretization errors.

#### 4.4. Convergence in time

Before running dynamic simulations, we confirm that our method preserves the global error of the time integration scheme to which it is combined, here Explicit Euler. For long enough simulations, we expect the linear solver tolerance  $\epsilon$  to shift the value of the global error, without affecting its convergence rate. Indeed, since the body velocities are obtained by solving (22) iteratively with a tolerance  $\epsilon$ , the position and orientation updates acquire an error  $O(\epsilon\Delta t)$  at each Explicit Euler time step. After a simulation time  $t_f$ , the accumulated error due to the iterative solver is therefore  $O(\epsilon t_f)$ , with a prefactor that depends on the system configuration. Below we use two examples, a swimming bacterium and the sedimentation of a pair of shells, to check that our reconstruction and correction steps do not affect the convergence of the discrete time-integrator.

First we consider the motion of a swimming bacterium. The bacterium is formed by a spherical body of radius  $R = 1\ \mu\text{m}$  and a single helical flagellum of length  $L = 10\ \mu\text{m}$ , see Fig. 9. We discretize the spherical body with 162 blobs of radius  $a = 0.131\ \mu\text{m}$  and the flagellum with only 38 blobs. The effective thickness of the flagellum is  $h \approx a/2$ . We incorporate two links between the spherical body and the flagellum,

$$\mathbf{g}_n(\mathbf{x}) = \mathbf{q}_{\text{body}} + \mathbf{R}(\boldsymbol{\theta}_{\text{body}})\Delta\mathbf{I}_{n\text{body}} - [\mathbf{q}_{\text{flagellum}} + \mathbf{R}(\boldsymbol{\theta}_{\text{flagellum}})\Delta\mathbf{I}_{n\text{flagellum}}] = \mathbf{0}, \quad (47)$$

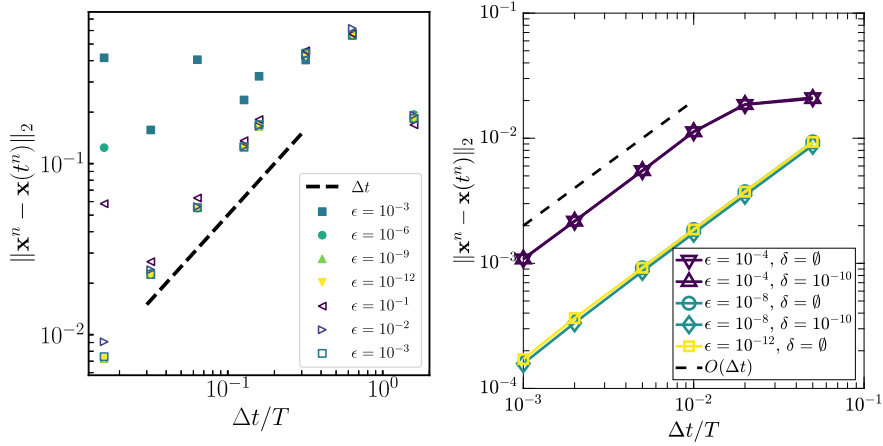
$$\Delta\mathbf{I}_{1\text{body}} = (0, 0, R + a), \quad \Delta\mathbf{I}_{1\text{flagellum}} = (0, 0, -a), \quad (48)$$

$$\Delta\mathbf{I}_{2\text{body}} = (0, 0, 2R + a), \quad \Delta\mathbf{I}_{2\text{flagellum}} = (0, 0, R - a), \quad (49)$$

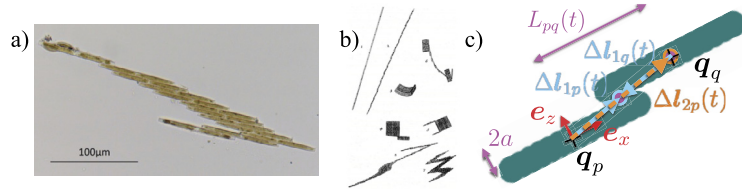
the first keeps the flagellum attached at the body surface while the second, at an arbitrary distance from the first but aligned along the same direction, fixes the axis of rotation. Then we apply equal but opposite torques ( $\tau = 0.4644\ \text{pN} \cdot \mu\text{m}$ ) to the body and the flagellum parallel to the axis of rotation, as the flagellum rotates the bacterium swims forward. We simulate the swimming bacterium for about 16 rotations periods  $T$  which moves the bacterium a distance  $d \approx 2.5R$ . The matrix  $\mathbf{C}$  corresponding to the constraints in Eq. (14) does not have full row rank (rank 5 instead of 6) as the links are not linearly independent. The resulting linear system is not invertible as such and the iterative solver alone cannot converge. As indicated in Section 3.1, the preconditioner computes the block-diagonal constraint resistance matrix using a pseudo-inverse when  $\mathbf{C}$  is not full rank to achieve convergence. However, the resulting solution is sensitive to small constraint violations: a constraint violation as small as  $\|\mathbf{g}\| = 10^{-6}$  at time iteration  $t_k$  can lead to large discrepancies between the preconditioned residual  $\tilde{r} = \|\tilde{\mathbf{A}}^{-1}(\mathbf{Ax} - \mathbf{b})\| \approx 10^{-12}$  and the actual residual  $r = \|\mathbf{Ax} - \mathbf{b}\| \approx 10^{-3}$  at time iteration  $t_{k+1}$ . In figure Fig. 6a we show the time convergence with respect to a reference simulation with tight tolerances  $\epsilon = \delta = 10^{-12}$  and a small time step  $\Delta t/T = 8 \cdot 10^{-3}$ . Due to the error accumulation the expected  $O(\Delta t)$  convergence with Explicit Euler is not observed for a loose nonlinear solver tolerance  $\delta > 10^{-9}$  (filled symbols). However, as soon as the constraint tolerance is tight enough  $\delta \leq 10^{-9}$  (open symbols), the scheme converges as expected, regardless of the linear solver tolerance. Identical tests using an explicit midpoint scheme show similar results: the  $O(\Delta t^2)$  convergence is observed as long as  $\delta$  is small enough (not shown). Using a tight tolerance  $\delta$  for the constraint violation is possible because the cost of the nonlinear solver is negligible compared to solving the linear system, and only a few iterations, typically less than 5, are required per articulated body, see Fig. 5b. When  $\delta \leq 10^{-9}$ , the effect of the linear solver tolerance is visible as the global error slightly decreases with  $\epsilon$  (open symbols).

Our second example is the sedimentation of  $N = 2$  deformable shells with radius  $R$  made of  $M_a = 42$  spheres and  $P_a = 60$  links each. Spheres are discretized as single blobs. In this case the links are all independent and the corresponding matrix  $\mathbf{C}$  has full row rank. The shells are initially placed at positions  $(\pm 1.5, 0, \pm 1.5)R$  and sediment for  $t_f = 5T$  where  $T = R/U_0$  where  $U_0$  is the settling speed at  $t = 0$ . Fig. 6b reports the maximum  $L_2$  error of the final body positions and orientations compared to a reference solution obtained with a very small time-step size,  $\Delta t/T = 10^{-4}$ , and tight tolerances,  $\epsilon = \delta = 10^{-12}$ . Since the constraints are independent, the preconditioned GMRES solver is not sensitive to constraint violations. For this specific configuration the shift due to the  $O(\epsilon t_f)$  error from the linear solver is clearly visible between  $\epsilon = 10^{-4}$  and  $\epsilon = 10^{-8}$ . We also observe that the constraint violations are so small that the effect of the correction scheme is only visible below a tight tolerance: when  $\epsilon \leq 10^{-8}$  the correction step slightly decreases the global error and even allows to obtain more accurate results than uncorrected positions with  $\epsilon = 10^{-12}$ .

Altogether, these results show that our correction procedure preserves the accuracy of the time-integration scheme regardless of the constraint arrangement.



**Fig. 6.** Time convergence of the Explicit Euler scheme quantified by the distance to a reference solution at final time  $t^n = t_f$ . (a) Results for a swimming bacterium. Full symbols: same tolerance for linear and nonlinear solver ( $\epsilon = \delta$ ). Open symbols: tight tolerance for nonlinear solver ( $\delta = 10^{-12}$ ). First order convergence is observed when the constraint violations are small ( $\delta \leq 10^{-9}$ ). (b) Convergence for two sedimenting shells with ( $\delta = 10^{-10}$ ) and without ( $\delta = \emptyset$ ) correction. First order convergence is always observed.



**Fig. 7.** a) Microscope view of a colony made of  $M = 12$  rod-like cells [69]. b) Drawing of the various configurations observed by O.F. Müller, who discovered the species *Bacillaria Paxillifer* in 1783 [4]. c) Schematic of the two links enforcing the kinematic sliding constraint between two adjacent rods  $p$  and  $q$ . Cyan arrows: connections to link 1 (51). Orange arrow: connection to link 2 (52). Magenta circles: location of the links. The basis vectors represent the local frame attached to body  $p$ .

## 5. Simulations

In this Section we simulate two different kinds of active systems to further illustrate the flexibility and versatility of our method. We describe, for the first time, the various locomotion modes of a model swimmer inspired by the diatom chain *Bacillaria Paxillifer*. Then we investigate the swimming speed of a model bacterium, compare our results with theoretical models, and finally exploit the scalability of our solver to simulate large suspensions of bacteria near a no-slip surface.

### 5.1. Swimming motion of a model diatom chain

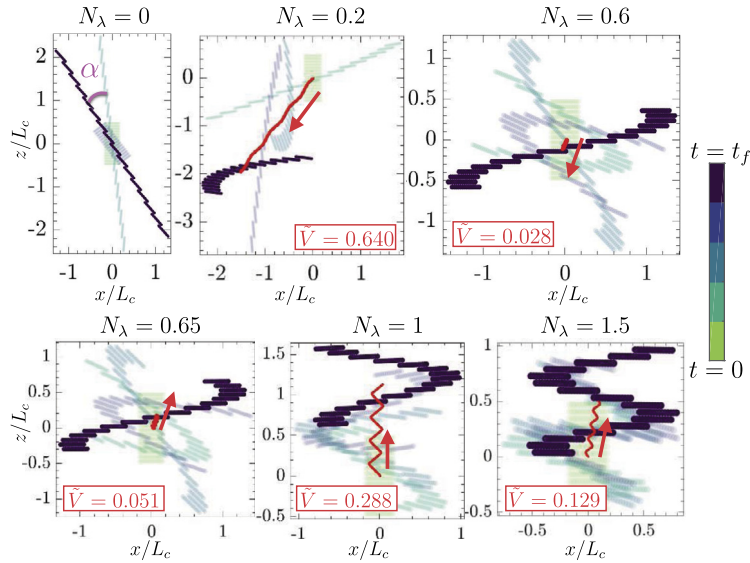
*Bacillaria Paxillifer* is a diatom species found in a wide variety of natural environments such as marine and freshwaters [67]. Cells are rectangular in shape with typical length  $L \approx 70 \mu\text{m}$ , height and width  $H \approx W \approx 10 \mu\text{m}$ , and live in stacked colonies (see Fig. 7a). Colonies of diatoms are phototactic and motile. Members, with their long axes parallel to one another, slide against their neighbours in a coordinated fashion, allowing the structure to expand or contract in many different ways (see Fig. 7a,b). So far, the diversity of deformation sequences, their purpose and their mechanical origin remain a mystery. The literature on these subjects is very scarce. In this work, we carry out the first numerical simulation of such microorganisms and show that hydrodynamic interactions between sliding cells lead to various, nontrivial, locomotion modes.

Our model swimmer inspired by *Bacillaria Paxillifer* is made of  $M = 16$  rigid rods of aspect ratio  $L/H = 6.3$  connected together by kinematic constraints in the  $(x, z)$ -plane. Each rod is discretized with 14 blobs of size  $a = H/2$  separated by a distance  $0.81a$ , which was shown to provide highly accurate predictions of the rod mobility coefficients [68,22]. The length of the stacked colony is therefore  $L_c = M \times H = 16 \times 2a$ . The kinematic constraints prescribe a time-dependent parallel sliding motion and a constant normal distance between pairs of rods. To enforce both of these constraints, two links are incorporated between each pair of adjacent rods  $p$  and  $q$ :

$$\mathbf{g}_n(\mathbf{x}) = \mathbf{q}_p + \mathbf{R}(\theta_p)\Delta \mathbf{l}_{np} - [\mathbf{q}_q + \mathbf{R}(\theta_q)\Delta \mathbf{l}_{nq}] = \mathbf{0}, \quad (50)$$

$$\Delta \mathbf{l}_{1p} = 0.5L_{pq}(t)\mathbf{e}_x + a\mathbf{e}_z, \quad \Delta \mathbf{l}_{1q} = -\Delta \mathbf{l}_{1p}, \quad (51)$$

$$\Delta \mathbf{l}_{2p} = L_{pq}(t)\mathbf{e}_x + 2a\mathbf{e}_z, \quad \Delta \mathbf{l}_{2q} = \mathbf{0}, \quad (52)$$



**Fig. 8.** Trajectory and conformation of the colony for six values of the number of wavelengths along the colony  $N_\lambda = L_c/\lambda = 0 - 1.5$ . The conformation is shown at five different times of the simulation, from green to dark blue (see colorbar), where only the final state is opaque. Solid red line: trajectory of the center of mass. Red arrow: swimming direction.  $\alpha$  is the angle between two fully extended configurations when  $N_\lambda = 0$ . See also Supplementary Movies.

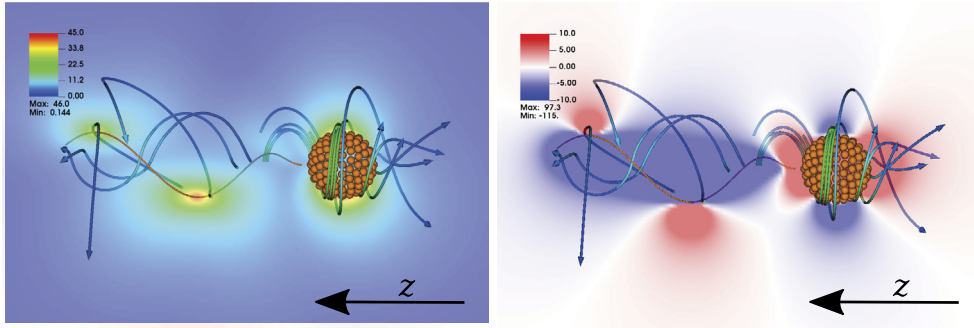
$$\text{with } L_{pq}(t) = A \sin(2\pi ft + \phi_n), \tag{53}$$

where  $A = 1.8L$  is the sliding amplitude, and  $\phi_n = (n - 1)\Delta\phi$ ,  $n = 1, \dots, M - 1$ , is the local phase,  $\Delta\phi$  being the phase shift between two adjacent pairs of sliding rods. The location of the second link in eq. (52) is arbitrary as long as it lies on the axis connecting  $\mathbf{q}_p$  and  $\mathbf{q}_q$  and does not coincide with the first link. This relative sliding motion can be seen as a deformation wave in the frame oriented with the colony:  $z(x_n + ct) \sim \sin(k(x_n + ct))$  with wave number  $k = 2\pi/\lambda = (M - 1)\Delta\phi/L_c$ , speed  $c = 2\pi f/k$  along the  $-x$  direction, and discrete horizontal positions  $x_n = (n - 1)2a$ ,  $n = 1, \dots, M$ . However, unlike typical travelling waves, the amplitude of the deformation wave depends on the wavelength  $\lambda$  because of the relative sliding motion: the shorter the wavelength, the larger the deformation.

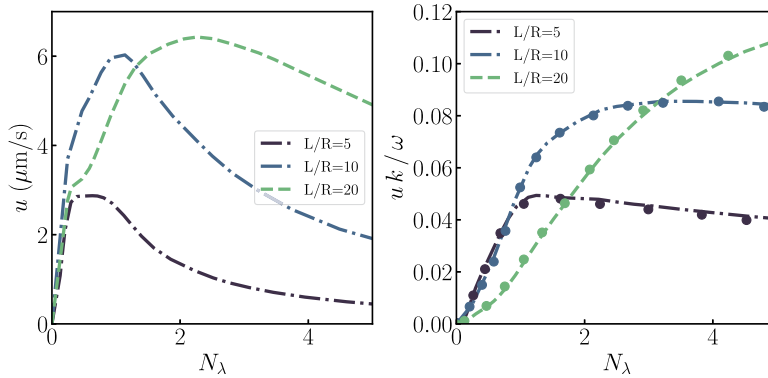
The wavelength of the deformation wave is varied in the range  $N_\lambda = L_c/\lambda = 0 - 1.5$ , where  $N_\lambda$  is the number of wavelengths along the length colony, which amounts to setting  $\Delta\phi = 0 - 2\pi/10$ . Each simulation is run for four sliding periods,  $t_f = 4/f = 4T$ , and is initialized from a rectangular stacked configuration along the  $z$ -axis where the rods are all parallel to the  $x$ -axis. We define  $\tilde{V} = \Delta q_{COM}/L_c$  the number of chain lengths travelled per sliding period, where  $\Delta q_{COM} = 1/3 \sum_{n=1}^3 \|\mathbf{q}_{COM}((n+1)T) - \mathbf{q}_{COM}(nT)\|$ . Fig. 8 shows the center of mass trajectories and snapshots of the colony conformation, for six typical values of  $N_\lambda$  (see also Supplementary Movies generated with Matlab®). When  $N_\lambda = 0$  the diatom chain expands and contracts symmetrically, leading to zero net velocity. Interestingly, the angle between two fully extended configurations is  $\alpha \approx \pi/5$ , which differs dramatically from the value one would obtain without hydrodynamic interactions between the rods:  $\alpha = \pi$ . When  $N_\lambda > 0$ , the colony breaks Stokes's reversibility and exhibits a rich variety of trajectories and velocities that vary nonmonotonically. For  $N_\lambda \leq 0.6$  the chain travels in the  $-x$  direction, which is the direction of the deformation wave, with some angle with respect to the  $x$ -axis set by hydrodynamics. A maximum in the swimming speed is obtained for  $N_\lambda = 0.2$ , for which the colony travels 64% its lengths per sliding period. The net velocity goes back to zero for  $N_\lambda \approx 0.6$ , and switches to the positive  $x$ -direction, i.e. opposite to the wave, for  $N_\lambda \approx 0.65$ . Beyond that value, the colony conformation and the trajectories are similar to the ones of a beating flagellum. The dimensionless speed reaches a secondary maximum at  $N_\lambda \approx 1$ , for which the micro-swimmer travels 29% of its length per sliding period. Altogether these preliminary results show that the swimming dynamics of *Bacillaria Paxillifer* is rich, intricate and differs from flagellar dynamics for which net motion is always opposite to the wave direction and optimal for  $N_\lambda \approx 1$  [70]. A more detailed study on the role of hydrodynamic interactions and the optimal swimming strategies in terms of efficiency [71,72] will be carried out in a forthcoming paper.

### 5.2. Single bacterium

Many unicellular organisms, like bacteria or protists, grow flagella that they use to self-propel [73,74]. The flagellum is a flexible filament around 20 nm in diameter and several micrometers in length [75]. When rotated by the molecular motors at their base, they take a helical form whose rotation allows the microorganisms to swim [76]. Under a constant angular velocity the flagellum reaches a steady state determined by its flexibility. When the flagellum is at the steady state it can be considered a rigid body as the bending forces are perfectly balanced by the fluid drag [1-3]. For this reason many authors have modelled the flagella as rigid objects [38,39,42], we follow this approach here and compare our results



**Fig. 9.** Streamlines around an active bacterium in free space and slice of the velocity magnitude (left) and the vorticity component along the  $z$ -axis (right). The bacterium has a body of radius  $R = 1\ \mu\text{m}$  and a slender flagellum with radius  $h/R = 0.02$ , length  $L = 10\ \mu\text{m}$  and  $N_\lambda = 1.14$  rotating at  $\omega = 100\ \text{Hz}$ .



**Fig. 10.** Swimming speed versus number of wavelengths along the flagellum  $N_\lambda = z_{\text{max}}/k\lambda$  for three flagellum lengths  $L$ . The radius (thickness) of the flagellum is  $h/R = 0.02$  and the angular velocity is  $\omega = 100\ \text{s}^{-1}$ . On the right we normalize the swimming speed with the helical wave speed  $\omega/k$ , with  $k$  the wavenumber of the helical wave, and compare with the results of Higdon (symbols) [42].

to those of Higdon, who employed a method of images and the slender body theory to calculate the swimming speed of bacteria [42].

In this section we will use the model introduced in Section 4.4 to study the dynamics of a single spherical bacterium with one flagellum in free space. The flagellum assumes an helical shape where the equation of the centerline with respect to its attachment point ( $z = 0$ ) is given by [42,77,78]

$$\mathbf{x}(z) = \left( \frac{1}{k}(1 - e^{-k^2 z^2}) \cos(kz), \frac{1}{k}(1 - e^{-k^2 z^2}) \sin(kz), z \right). \tag{54}$$

An example of a flagellum attached to a spherical bacterial body is shown in Fig. 9. In this section we study the speed of the bacterium for different flagellum lengths  $L$  and wavenumbers  $k$ . We will see that the number of wavelengths in the flagellum,  $N_\lambda = z_{\text{max}}/k\lambda$ , controls the speed [42].

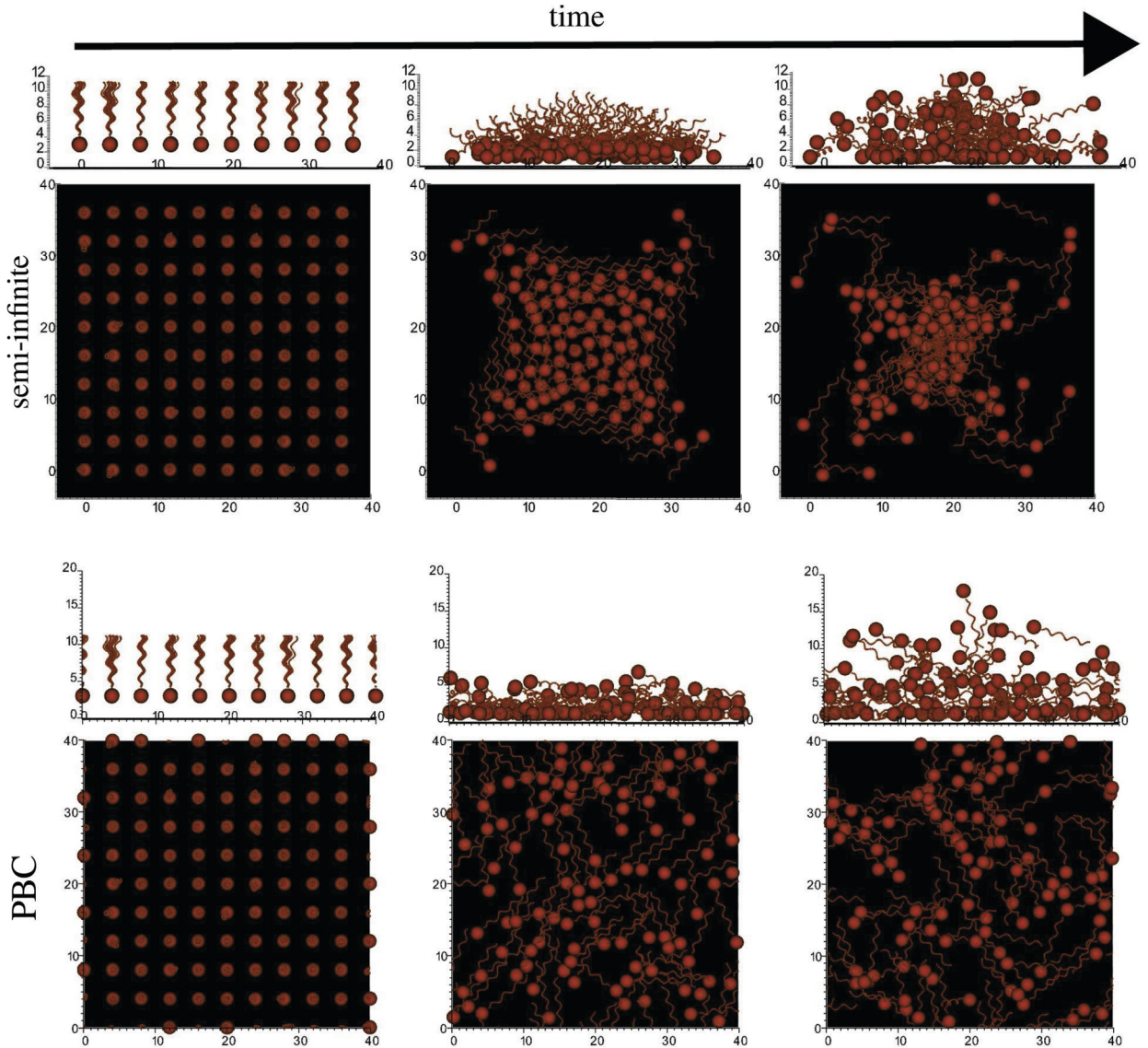
The flagellum can be rotated in two different ways: either by prescribing a constant angular velocity of the tail relative to the body, or by applying equal but opposite torques to the body and flagellum so that the whole bacterium is force and torque-free. For a single bacterium swimming in unbounded space, both options provide equivalent results. Here we fix the relative angular velocity between the spherical body and the flagellum.

Fig. 9 shows the streamlines around a bacterium together with the magnitude of the velocity (left) and vorticity fields (right). It can be observed that the flagellum and the spherical body rotate in opposite directions, as required for a torque-free swimmer. In the absence of walls or other swimmers, the bacterium swims with an average constant speed and direction. In Fig. 10 we plot the swimming speed for different flagellum lengths,  $L$ , and different number of wavelengths  $N_\lambda = z_{\text{max}}/k\lambda$ . As in the seminal work of Higdon [42] the absolute speed is maximized for  $N_\lambda \approx 1$ , i.e. when the helical wave does a full turn along the flagellum. When normalized by the wave speed,  $\omega/k$ , the motion of the flagellum is transmitted better for larger values of  $N_\lambda$ . Our results agree well with those of Higdon [42].

### 5.3. Bacterial suspension

To demonstrate the capabilities of our method we simulate a suspension with  $N = 100$  bacteria swimming above a no-slip wall. Most numerical studies on bacterial suspensions use either continuum formulations [79,80], minimal models based





**Fig. 11.** Top and lateral views of a suspension of  $N = 100$  bacteria above a no-slip wall at times 0, 5 and 12 s. Top row: semi-infinite domain only bounded by the wall. Bottom row: periodic boundary conditions (PBC) in the directions parallel to the wall. See also movies in the supplemental material.

on regularized Stokes dipoles [81] or methods such as Lattice Boltzmann or Multiparticle Collision Dynamics that work at finite Schmidt and Reynolds numbers [82,83]. Instead, we solve the Stokes flow around the whole bacteria, each discretized with 200-blobs (body + flagellum). This is a large system with  $N \times 200 = 20,000$  blobs and 61,800 unknowns. We consider two types of boundary conditions: a semi-infinite system only bounded by the wall and a system with periodic boundary conditions (PBC) in the directions parallel to the wall. We include a short range steric repulsion between the bacteria and between the bacteria with the wall to prevent overlaps. To compute the hydrodynamic interactions between the blobs above a no-slip wall we use a Fast Multipole implementation of the wall-corrected RPY tensor, available in the library STKFMM [52,25], that enables PBC along the lateral directions. In both cases the computational cost scales quasilinearly with the total number of blobs which allows to run long simulations. The tolerance of the linear and nonlinear solvers are set to  $\epsilon = 10^{-6}$  and  $\delta = 10^{-10}$  respectively. The angular velocity of the flagellum is set to  $\omega = 100$  Hz and the time step to  $\Delta t = 0.005$  s, which corresponds to 12 steps per flagellum rotation. We simulate the system for 2500 time steps, which takes about 12 h on a 28-core computer.

Fig. 11 shows a few snapshots of the bacterial suspension, see also Supplementary Movies generated with VisIt [84]. Initially, the bacteria form a square lattice and are oriented towards the wall with small perturbations in their orientations. They start swimming towards the wall but the monolayer soon becomes unstable and forms dense clusters. In the unbounded domain a single cluster is formed, while with PBC the suspension exhibits more complex patterns. Additionally, in the unbounded domain the cluster preserves the tendency of bacterium to swim in circles near walls [85] due to the

torque dipole created by the opposite rotation of body and flagellum and the hydrodynamic interactions with the wall [86]. With PBC the strong interactions between neighbour bacteria suppress this effect. At longer times the flow created by the bacteria destroys the monolayer: some bacteria reorient away from the wall, while others stay tilted towards it.

The collective motion that destabilizes the initial lattice is driven by the flow generated by the microswimmers [79]. The flow induced by a bacterium can be modelled, to a low order approximation, as an extensible force dipole as the flagellum pushes the fluid backwards while the bacterium body pushes it forward. At the same time the fluid is pulled towards the bacterium from its sides to maintain the incompressibility condition. This flow causes a lateral attraction between swimmers, creating clusters and eventually destabilizing the monolayer. Suspensions of swimmers with this characteristic flow signature, known as *pushers*, exhibit a complex dynamics with instabilities and large scale density fluctuations [87,88]. Interestingly, it is well known that many microswimmers, such as bacteria, are hydrodynamically attracted to obstacles and surfaces [89–91]. Such interactions, if strong enough, could stabilize a monolayer of microswimmers near a wall. In this particular experiment the destabilizing effect of the pushers' flow dominates over the hydrodynamic attraction to the wall. Our model, which includes both effects as well as the steric interactions and collisions between bacteria, opens a window to explore the stability of microswimmer suspensions beyond continuum model approximations.

## 6. Conclusions

We have presented a new framework to simulate large suspensions of articulated bodies. Our velocity formulation of the constraints between bodies enables to write the mixed resistance-mobility problem as a single linear system. We solved this linear system with a preconditioned iterative solver that couples effectively with any numerical method to compute hydrodynamic interactions between rigid bodies. Interestingly, the solver convergence is independent of the system size and constraint type, therefore allowing to simulate large suspensions in a scalable fashion. Combining our new reconstruction method, which generalizes the robot-arm model, with a simple and costless correction procedure, one can track articulated bodies with a reduced number of degrees of freedom while avoiding constraint violations. Our method is robust and flexible, it applies to a wide variety of physical and biological systems. We have illustrated some of them involving deformable filaments, shells, and various types of micro-swimmers. Our implementations are freely and publicly available so that others can use its capacity to address new problems. Some applications of interest happen at very small scales, such as the motion of bacteria, membranes, lipid bilayers, actin filaments or molecular motors, where thermal fluctuations play an important role. One future direction is, therefore, to include Brownian motion in our framework [92].

## CRedit authorship contribution statement

**Florencio Balboa Usabiaga:** Conceptualization, Data curation, Investigation, Methodology, Software, Validation, Writing – original draft, Writing – review & editing. **Blaise Delmotte:** Conceptualization, Data curation, Investigation, Methodology, Software, Validation, Writing – original draft, Writing – review & editing.

## Declaration of competing interest

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

## Acknowledgements

F.B.U. acknowledges support from “la Caixa” Foundation (ID 100010434), fellowship LCF/BQ/PI20/11760014, and from the European Union’s Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 847648 and also by the Basque Government through the BERC 2022-2025 program and by the Ministry of Science, Innovation and Universities: BCAM Severo Ochoa accreditation SEV-2017-0718. B.D. acknowledges support from the French National Research Agency (ANR), under award ANR-20-CE30-0006. B.D. also thanks the NVIDIA Academic Partnership program for providing GPU hardware for performing some of the simulations reported here.

## Appendix A. No net constraint force and torque on articulated bodies

We prove here that the constraint forces do not exert any total force or torque on the articulated body. We consider again the linear system (16),

$$\begin{bmatrix} -\mathbf{N}\mathbf{C}^T & \mathbf{I} \\ \mathbf{0} & \mathbf{C} \end{bmatrix} \begin{bmatrix} \boldsymbol{\phi} \\ \mathbf{U} \end{bmatrix} = \begin{bmatrix} \mathbf{N}\mathbf{F} \\ \mathbf{B} \end{bmatrix}, \quad (\text{A.1})$$

The block of the matrix  $\mathbf{C}$  corresponding to the link  $n$  connecting bodies  $p$  and  $q$  is given by

$$\mathbf{C}_n = [\mathbf{0} \cdots \mathbf{I} \quad -\Delta \mathbf{I}_{np}^\times \quad \mathbf{0} \cdots -\mathbf{I} \quad \Delta \mathbf{I}_{nq}^\times \quad \mathbf{0} \cdots], \quad (\text{A.2})$$

and the constraint forces and torques exerted by link  $n$  on those two bodies are

$$\mathbf{F}^C = \mathbf{C}_n^T \boldsymbol{\phi} = (\mathbf{0} \cdots \boldsymbol{\phi}_n \Delta \mathbf{l}_{np} \times \boldsymbol{\phi}_n \mathbf{0} \cdots - \boldsymbol{\phi}_n - \Delta \mathbf{l}_{nq} \times \boldsymbol{\phi}_n \mathbf{0} \cdots). \quad (\text{A.3})$$

The total force on the articulated body exerted by the constraint is therefore  $\mathbf{F}_T^C = \mathbf{F}_p^C + \mathbf{F}_q^C = \boldsymbol{\phi}_n - \boldsymbol{\phi}_n = \mathbf{0}$ . The total torque, here defined around the body  $p$ , is

$$\begin{aligned} \mathbf{T}_T^C &= \mathbf{T}_p^C + \mathbf{T}_q^C + (\mathbf{q}_q - \mathbf{q}_p) \times \mathbf{F}_q^C \\ &= (\mathbf{q}_p + \Delta \mathbf{l}_{np} - \mathbf{q}_q - \Delta \mathbf{l}_{nq}) \times \boldsymbol{\phi}_n \\ &= \mathbf{g}_n \times \boldsymbol{\phi}_n = \mathbf{0}. \end{aligned} \quad (\text{A.4})$$

These results apply to constant and time dependent constraints.

### Appendix B. Jacobian of the objective function

In this appendix we write the expression for the Jacobian of the residual (38). To derive the Jacobian it is enough to focus on Eq. (40) since all the constraints have the same form. To further simplify the equations we drop subscripts and derive the Jacobian for the shorter equation

$$\mathbf{g}(\delta \mathbf{q}, \delta \boldsymbol{\theta}) = \mathbf{q} + \delta \mathbf{q} + \mathbf{R}(\delta \boldsymbol{\theta} \bullet \boldsymbol{\theta}) \Delta \mathbf{l}. \quad (\text{B.1})$$

Note that the variables of the two rigid bodies,  $p$  and  $q$ , enter in (40) in the same way with a plus or minus sign. Therefore the full Jacobian of (40) will have just twice as many nonzero terms, each half multiplied by  $+1$  or  $-1$ . Using the property of the rotation matrices  $\mathbf{R}(\boldsymbol{\theta}_2 \bullet \boldsymbol{\theta}_1) = \mathbf{R}(\boldsymbol{\theta}_2) \mathbf{R}(\boldsymbol{\theta}_1)$ , we can write the residual as

$$\mathbf{g}(\delta \mathbf{q}, \delta \boldsymbol{\theta}) = \mathbf{q} + \delta \mathbf{q} + \mathbf{R}(\delta \boldsymbol{\theta}) \widetilde{\Delta \mathbf{l}}. \quad (\text{B.2})$$

where  $\widetilde{\Delta \mathbf{l}} = \mathbf{R}(\boldsymbol{\theta}) \Delta \mathbf{l}$ . Finally, the rotation matrix associated to the unit quaternion  $\boldsymbol{\theta} = (s, \mathbf{p})$  is [44]

$$\mathbf{R} = 2 \left[ \mathbf{p} \mathbf{p}^T + s \mathbf{p}^\times + \left( s^2 - \frac{1}{2} \right) \mathbf{I} \right], \quad (\text{B.3})$$

where  $\mathbf{p}^\times \mathbf{x} = \mathbf{p} \times \mathbf{x}$  for any vector  $\mathbf{x}$ . Then, the constraint in indicial notation is

$$g_i = q_i + \delta q_i + 2p_i(p_j \widetilde{\Delta l}_j) + s \epsilon_{ijk} p_k \widetilde{\Delta l}_j + \left( s^2 - \frac{1}{2} \right) \widetilde{\Delta l}_i. \quad (\text{B.4})$$

where  $\epsilon_{ijk}$  is the Levi-Civita symbol. From this expression it is easy to compute the elements of the Jacobian

$$J_{iq_j} = \frac{\partial g_i}{\partial q_j} = \delta_{ij}, \quad (\text{B.5})$$

$$J_{is} = \frac{\partial g_i}{\partial s} = \epsilon_{ijk} p_k \widetilde{\Delta l}_j + 2s \widetilde{\Delta l}_i \quad (\text{B.6})$$

$$J_{ip_l} = \frac{\partial g_i}{\partial p_l} = 2\delta_{il}(p_j \widetilde{\Delta l}_j) + 2p_i \widetilde{\Delta l}_l + s \epsilon_{ijl} \widetilde{\Delta l}_j. \quad (\text{B.7})$$

This Jacobian has dimensions  $3 \times 7$ , the Jacobian of the full nonlinear equation (38) will have dimensions  $3P \times 7M$  for  $M$  rigid bodies linked by  $P$  constraints, however, only  $30P$  elements will be nonzero and thus the Jacobian will be sparse.

### Appendix C. Midpoint scheme

Below we outline the correction algorithm embedded in an Explicit Midpoint scheme (or RK2).

Time loop: for  $k = 0, \dots, N_{it} - 1$

1. Solve the linear system (22) at  $t_k = k\Delta t$  to obtain the body velocities  $\mathbf{U}^{(k)}$ ,
2. Compute the translational velocity of the COM's  $\mathbf{u}_{COM}^{(k)}$  using (34),
3. Update the COM and the orientations of each assembly to the midpoint  $t_{k+1/2} = (k + 1/2)\Delta t$

$$\mathbf{q}_{COM}^* = \mathbf{q}_{COM}^{(k)} + \frac{1}{2} \Delta t \mathbf{u}_{COM}^{(k)} \quad (\text{C.1})$$

$$\boldsymbol{\theta}^* = \boldsymbol{\theta}_{\omega^{(k)} \frac{1}{2} \Delta t} \bullet \boldsymbol{\theta}^{(k)}, \quad (\text{C.2})$$

4. Reconstruct the articulated bodies using the pseudo-inverse (36), where  $\mathbf{z}^a = \mathbf{z}^a(\boldsymbol{\theta}^*, t_{k+1/2})$  and then apply Eq. (37) to obtain the body positions around the updated centers of mass  $\mathbf{q}^*$ .
5. Evaluate the updated constraint vector  $\mathbf{g}^* = \mathbf{g}(\mathbf{q}^*, \boldsymbol{\theta}^*, t_{k+1/2})$  and check the constraint violation: if  $\|\mathbf{g}^*\|_2^2 < \delta$  then

$$\mathbf{q}^{(k+1/2)} \leftarrow \mathbf{q}^* \tag{C.3}$$

$$\boldsymbol{\theta}^{(k+1/2)} \leftarrow \boldsymbol{\theta}^* \tag{C.4}$$

else

- Solve the minimization problem (38)-(39) with the nonlinear solver to find the increments  $(\check{\delta}\mathbf{q}, \check{\delta}\boldsymbol{\theta})$
- Correct the body positions and orientations

$$\mathbf{q}^{(k+1/2)} = \mathbf{q}^* + \check{\delta}\mathbf{q}, \tag{C.5}$$

$$\boldsymbol{\theta}^{(k+1/2)} = \check{\delta}\boldsymbol{\theta} \bullet \boldsymbol{\theta}^*. \tag{C.6}$$

6. Solve the linear system (22) at the midpoint  $t_{k+1/2} = (k + 1/2)\Delta t$  to obtain the body velocities  $\mathbf{U}^{(k+1/2)}$ ,
7. Compute the translational velocity of the COM's  $\mathbf{u}_{COM}^{(k+1/2)}$  using (34),
8. Update the COM and the orientations of each assembly to the next time step

$$\mathbf{q}_{COM}^* = \mathbf{q}_{COM}^{(k)} + \Delta t \mathbf{u}_{COM}^{(k+1/2)} \tag{C.7}$$

$$\boldsymbol{\theta}^* = \boldsymbol{\theta}_{\omega^{(k+1/2)\Delta t}} \bullet \boldsymbol{\theta}^{(k)}, \tag{C.8}$$

9. Reconstruct the articulated bodies using the pseudo-inverse (36), where  $\mathbf{z}^a = \mathbf{z}^a(\boldsymbol{\theta}^*, t_{k+1})$  and then apply Eq. (37) to obtain the body positions around the updated centers of mass  $\mathbf{q}^*$ .
10. Evaluate the updated constraint vector  $\mathbf{g}^* = \mathbf{g}(\mathbf{q}^*, \boldsymbol{\theta}^*, t_{k+1})$  and check constraint violation: if  $\|\mathbf{g}^*\|_2^2 < \delta$  then

$$\mathbf{q}^{(k+1)} \leftarrow \mathbf{q}^* \tag{C.9}$$

$$\boldsymbol{\theta}^{(k+1)} \leftarrow \boldsymbol{\theta}^* \tag{C.10}$$

else

- Solve the minimization problem (38)-(39) with the nonlinear solver to find the increments  $(\check{\delta}\mathbf{q}, \check{\delta}\boldsymbol{\theta})$
- Correct the body positions and orientations

$$\mathbf{q}^{(k+1)} = \mathbf{q}^* + \check{\delta}\mathbf{q}, \tag{C.11}$$

$$\boldsymbol{\theta}^{(k+1)} = \check{\delta}\boldsymbol{\theta} \bullet \boldsymbol{\theta}^*. \tag{C.12}$$

#### Appendix D. Supplementary material

Supplementary material related to this article can be found online at <https://doi.org/10.1016/j.jcp.2022.111365>.

#### References

- [1] H.C. Berg, R.A. Anderson, Bacteria swim by rotating their flagellar filaments, *Nature* 245 (5425) (1973) 380–382, <https://doi.org/10.1038/245380a0>, bandiera\_abtest: a Cg\_type: Nature Research Journals Number: 5425 Primary\_atype: Research Publisher: Nature Publishing Group, <https://www.nature.com/articles/245380a0>.
- [2] S.M. Block, D.F. Blair, H.C. Berg, Compliance of bacterial flagella measured with optical tweezers, *Nature* 338 (6215) (1989) 514–518.
- [3] S. Trachtenberg, I. Hammel, The rigidity of bacterial flagellar filaments and its relation to filament polymorphism, *J. Struct. Biol.* 109 (1) (1992) 18–27.
- [4] O.F. Müller, *Kleine Schriften aus der Naturhistorie, vol. 1*, Buchhandlung der Gelehrten, 1782.
- [5] M.R.M. Kapinga, R. Gordon, Cell attachment in the motile colonial diatom bacillaria paxillifer, *Diatom Res.* 7 (2) (1992) 215–220, <https://doi.org/10.1080/0269249X.1992.9705214>, publisher: Taylor & Francis, <https://www.tandfonline.com/doi/citedby/10.1080/0269249X.1992.9705214>.
- [6] R. Gordon, Partial synchronization of the colonial diatom Bacillaria “paradoxa”, *Res. Ideas Outcomes* 2 (2016) e7869, <https://doi.org/10.3897/rio.2.e7869>, publisher: Pensoft Publishers, <https://riojournal.com/article/7869/>.
- [7] R. Dreyfus, J. Baudry, M.L. Roper, M. Fermigier, H.A. Stone, J. Bibette, Microscopic artificial swimmers, *Nature* 437 (7060) (2005) 862–865, <https://doi.org/10.1038/nature04090>, <http://www.nature.com/articles/nature04090>.
- [8] B. Jang, E. Gutman, N. Stucki, B.F. Seitz, P.D. Wendel-García, T. Newton, J. Pokki, O. Ergeneman, S. Pané, Y. Or, et al., Undulatory locomotion of magnetic multilink nanoswimmers, *Nano Lett.* 15 (7) (2015) 4829–4833.
- [9] P. Liao, L. Xing, S. Zhang, D. Sun, Magnetically driven undulatory microswimmers integrating multiple rigid segments, *Small* 15 (36) (2019) 1901197.
- [10] E.M. Purcell, Life at low Reynolds number, *Am. J. Phys.* 45 (1) (1977) 3–11.
- [11] L.E. Becker, S.A. Koehler, H.A. Stone, On self-propulsion of micro-machines at low Reynolds number: Purcell's three-link swimmer, *J. Fluid Mech.* 490 (2003) 15.
- [12] B. Bonnard, M. Chyba, J. Rouot, D. Takagi, Sub-Riemannian geometry, Hamiltonian dynamics, micro-swimmers, copepod nauplii and copepod robot, *Pac. J. Math. Ind.* 10 (1) (2018) 1–27.

- [13] P. Bettioli, B. Bonnard, J. Rouot, Optimal strokes at low Reynolds number: a geometric and numerical study of copepod and purcell swimmers, *SIAM J. Control Optim.* 56 (3) (2018) 1794–1822.
- [14] S. Sacanna, W.T. Irvine, P.M. Chaikin, D.J. Pine, Lock and key colloids, *Nature* 464 (7288) (2010) 575–578.
- [15] A. McMullen, M. Holmes-Cerfon, F. Sciortino, A.Y. Grosberg, J. Brujic, Freely jointed polymers made of droplets, *Phys. Rev. Lett.* 121 (13) (2018) 138002, <https://doi.org/10.1103/PhysRevLett.121.138002>, <https://link.aps.org/doi/10.1103/PhysRevLett.121.138002>.
- [16] R.F. Ross, D.J. Klingenberg, Dynamic simulation of flexible fibers composed of linked rigid bodies, *J. Chem. Phys.* 106 (7) (1997) 2949–2960.
- [17] B. Delmotte, E. Climent, F. Plouraboué, A general formulation of bead models applied to flexible fibers and active filaments at low Reynolds number, *J. Comput. Phys.* 286 (2015) 14–37, <https://doi.org/10.1016/j.jcp.2015.01.026>, <http://www.sciencedirect.com/science/article/pii/S0021999115000303>.
- [18] S.F. Schoeller, A.K. Townsend, T.A. Westwood, E.E. Keaveny, Methods for suspensions of passive and active filaments, *J. Comput. Phys.* 424 (2021) 109846.
- [19] C. Pozrikidis, *Boundary Integral and Singularity Methods for Linearized Viscous Flow*, Cambridge Texts in Applied Mathematics, Cambridge University Press, 1992.
- [20] T.D. Montenegro-Johnson, S. Michelin, E. Lauga, A regularised singularity approach to phoretic problems, *Eur. Phys. J. E* 38 (12) (2015) 139, <https://doi.org/10.1140/epje/i2015-15139-7>, <http://link.springer.com/10.1140/epje/i2015-15139-7>.
- [21] J.W. Swan, G. Wang, Rapid calculation of hydrodynamic and transport properties in concentrated solutions of colloidal particles and macromolecules, *Phys. Fluids* 28 (1) (2016) 011902, <https://doi.org/10.1063/1.4939581>, <http://aip.scitation.org/doi/10.1063/1.4939581>.
- [22] F. Balboa Usabiaga, B. Kallemov, B. Delmotte, A.P.S. Bhalla, B.E. Griffith, A. Donev, Hydrodynamics of suspensions of passive and active rigid particles: a rigid multiblob approach, *Commun. Appl. Math. Comput. Sci.* 11 (2) (2016) 217–296, <https://doi.org/10.2140/camcos.2016.11.217>.
- [23] L. Greengard, V. Rokhlin, A fast algorithm for particle simulations, *J. Comput. Phys.* 73 (2) (1987) 325–348.
- [24] A.-K. Tornberg, L. Greengard, A fast multipole method for the three-dimensional Stokes equations, *J. Comput. Phys.* 227 (3) (2008) 1613–1619.
- [25] W. Yan, R. Blackwell, Kernel aggregated fast multipole method: efficient summation of Laplace and Stokes kernel functions, arXiv preprint, arXiv: 2010.15155, 2020.
- [26] M. Wang, J.F. Brady, Spectral Ewald acceleration of Stokesian dynamics for polydisperse suspensions, *J. Comput. Phys.* 306 (2016) 443–477, <https://doi.org/10.1016/j.jcp.2015.11.042>, <https://linkinghub.elsevier.com/retrieve/pii/S0021999115007822>.
- [27] A.M. Fiore, F. Balboa Usabiaga, A. Donev, J.W. Swan, Rapid sampling of stochastic displacements in Brownian dynamics simulations, *J. Chem. Phys.* 146 (12) (2017) 124116, arXiv: <http://aip.scitation.org/doi/pdf/10.1063/1.4978242>, <http://aip.scitation.org/doi/abs/10.1063/1.4978242>.
- [28] A. Sierou, J.F. Brady, Accelerated stokesian dynamics simulations, *J. Fluid Mech.* 448 (2001) 115–146, <https://doi.org/10.1017/S0022112001005912>, [https://www.cambridge.org/core/product/identifier/S0022112001005912/type/journal\\_article](https://www.cambridge.org/core/product/identifier/S0022112001005912/type/journal_article).
- [29] A.M. Fiore, J.W. Swan, Fast Stokesian dynamics, *J. Fluid Mech.* 878 (2019) 544–597, <https://doi.org/10.1017/jfm.2019.640>, [https://www.cambridge.org/core/product/identifier/S0022112019006402/type/journal\\_article](https://www.cambridge.org/core/product/identifier/S0022112019006402/type/journal_article).
- [30] C. Peskin, The immersed boundary method, *Acta Numer.* 11 (2002) 479–517.
- [31] M. Maxey, B. Patel, Localized force representations for particles sedimenting in Stokes flow, *Int. J. Multiph. Flow* 27 (9) (2001) 1603–1626.
- [32] S. Lomholt, M.R. Maxey, Force-coupling method for particulate two-phase flow: Stokes flow, *J. Comput. Phys.* 184 (2) (2003) 381–405, [https://doi.org/10.1016/S0021-9991\(02\)00021-9](https://doi.org/10.1016/S0021-9991(02)00021-9), <https://linkinghub.elsevier.com/retrieve/pii/S0021999102000219>.
- [33] H. Goldstein, C. Poole, J. Safko, *Classical Mechanics*, 3rd ed., Addison-Wesley, 2001.
- [34] S. Yamamoto, T. Matsuoka, A method for dynamic simulation of rigid and flexible fibers in a flow field, *J. Chem. Phys.* 98 (1) (1993) 644–650, <https://doi.org/10.1063/1.464607>, <http://aip.scitation.org/doi/10.1063/1.464607>.
- [35] S.B. Lindström, T. Uesaka, Simulation of the motion of flexible fibers in viscous fluid flow, *Phys. Fluids* 19 (11) (2007) 113307.
- [36] C.F. Schmid, L.H. Switzer, D.J. Klingenberg, Simulations of fiber flocculation: effects of fiber properties and interfiber friction, *J. Rheol.* 44 (4) (2000) 781–809, <https://doi.org/10.1122/1.551116>, <http://sor.scitation.org/doi/10.1122/1.551116>.
- [37] S.F. Schoeller, E.E. Keaveny, From flagellar undulations to collective motion: predicting the dynamics of sperm suspensions, *J. R. Soc. Interface* 15 (140) (2018) 20170834, <https://doi.org/10.1098/rsif.2017.0834>, <https://royalsocietypublishing.org/doi/10.1098/rsif.2017.0834>.
- [38] H. Shum, E.A. Gaffney, D.J. Smith, Modelling bacterial behaviour close to a no-slip plane boundary: the influence of bacterial geometry, *Proc. R. Soc. A, Math. Phys. Eng. Sci.* 466 (2118) (2010) 1725–1748, <https://doi.org/10.1098/rspa.2009.0520>, <https://royalsocietypublishing.org/doi/10.1098/rspa.2009.0520>.
- [39] H. Shum, J.M. Yeomans, Entrainment and scattering in microswimmer-colloid interactions, *Phys. Rev. Fluids* 2 (11) (2017) 113101, <https://doi.org/10.1103/PhysRevFluids.2.113101>, <https://link.aps.org/doi/10.1103/PhysRevFluids.2.113101>.
- [40] B.J. Walker, R.J. Wheeler, K. Ishimoto, E.A. Gaffney, Boundary behaviours of Leishmania mexicana: a hydrodynamic simulation study, *J. Theor. Biol.* 462 (2019) 311–320, <https://doi.org/10.1016/j.jtbi.2018.11.016>, <https://linkinghub.elsevier.com/retrieve/pii/S002251931830568X>.
- [41] M. Doi, D. Chen, Simulation of aggregating colloids in shear flow, *J. Chem. Phys.* 90 (10) (1989) 5271–5279, <https://doi.org/10.1063/1.456430>, <http://aip.scitation.org/doi/10.1063/1.456430>.
- [42] J. Higdon, The hydrodynamics of flagellar propulsion: helical waves, *J. Fluid Mech.* 94 (2) (1979) 331–351, <https://doi.org/10.1017/S0022112079001051>.
- [43] S. Kim, S.J. Karrila, *Microhydrodynamics: Principles and Selected Applications*, Butterworth-Heinemann Series in Chemical Engineering, Butterworth-Heinemann, Boston, 1991, oCLC: 246965391.
- [44] S. Delong, F. Balboa Usabiaga, A. Donev, Brownian dynamics of confined rigid bodies, *J. Chem. Phys.* 143 (14) (2015) 144107, <https://doi.org/10.1063/1.4932062>, <http://scitation.aip.org/content/aip/journal/jcp/143/14/10.1063/1.4932062>.
- [45] T.A. Westwood, B. Delmotte, E.E. Keaveny, A generalised drift-correcting time integration scheme for Brownian suspensions of rigid particles with arbitrary shape, arXiv:2106.00449, Jun. 2021, <http://arxiv.org/abs/2106.00449>.
- [46] R. Featherstone, *Robot Dynamics Algorithms, Robotics: Vision, Manipulation and Sensors*, vol. 22, Springer US, 1987.
- [47] Y. Bao, M. Rachh, E.E. Keaveny, L. Greengard, A. Donev, A fluctuating boundary integral method for brownian suspensions, *J. Comput. Phys.* 374 (2018) 1094–1119, <https://doi.org/10.1016/j.jcp.2018.08.021>, <http://www.sciencedirect.com/science/article/pii/S0021999118305448>.
- [48] J. Rotne, S. Prager, Variational treatment of hydrodynamic interaction in polymers, *J. Chem. Phys.* 50 (1969) 4831, <https://doi.org/10.1063/1.1670977>.
- [49] E. Wajnryb, K.A. Mizerski, P.J. Zuk, P. Szymczak, Generalization of the Rotne-Prager-Yamakawa mobility and shear disturbance tensors, *J. Fluid Mech.* 731 (2013) R3, <https://doi.org/10.1017/jfm.2013.402>.
- [50] J.W. Swan, J.F. Brady, Simulation of hydrodynamically interacting particles near a no-slip boundary, *Phys. Fluids* 19 (11) (2007) 113306, <https://doi.org/10.1063/1.2803837> (1994–present), <http://scitation.aip.org/content/aip/journal/pof2/19/11/10.1063/1.2803837>.
- [51] Z. Liang, Z. Gimbutas, L. Greengard, J. Huang, S. Jiang, A fast multipole method for the Rotne-Prager-Yamakawa tensor and its applications, *J. Comput. Phys.* 234 (2013) 133–139, <https://doi.org/10.1016/j.jcp.2012.09.021>, <http://www.sciencedirect.com/science/article/pii/S0021999112005529>.
- [52] W. Yan, M. Shelley, Universal image system for non-periodic and periodic Stokes flows above a no-slip wall, *J. Comput. Phys.* 375 (2018) 263–270, <https://doi.org/10.1016/j.jcp.2018.08.041>.
- [53] J. Blake, A spherical envelope approach to ciliary propulsion, *J. Fluid Mech.* 46 (01) (1971) 199–208, <https://doi.org/10.1017/S002211207100048X>.
- [54] Q. Brossseau, F. Balboa Usabiaga, E. Lushi, Y. Wu, L. Ristroph, J. Zhang, M. Ward, M.J. Shelley, Relating rheotaxis and hydrodynamic actuation using asymmetric gold-platinum phoretic rods, *Phys. Rev. Lett.* 123 (2019) 178004, <https://doi.org/10.1103/PhysRevLett.123.178004>, <https://link.aps.org/doi/10.1103/PhysRevLett.123.178004>.

- [55] Q. Brosseau, F.B. Usabiaga, E. Lushi, Y. Wu, L. Ristroph, M.D. Ward, M.J. Shelley, J. Zhang, Metallic microswimmers driven up the wall by gravity, *Soft Matter* 17 (2021) 6597–6602, <https://doi.org/10.1039/D1SM00554E>.
- [56] K.S. Sillmore, M.S. Strano, J.W. Swan, Buckling, crumpling, and tumbling of semiflexible sheets in simple shear flow, *Soft Matter* 17 (18) (2021) 4707–4718, <https://doi.org/10.1039/D0SM02184A>, <http://xlink.rsc.org/?DOI=D0SM02184A>.
- [57] B. Marchetti, V. Raspa, A. Lindner, O. Du Roure, L. Bergougnoux, E. Guazzelli, C. Duprat, Deformation of a flexible fiber settling in a quiescent viscous fluid, *Phys. Rev. Fluids* 3 (10) (2018) 104102.
- [58] B. Herzhaft, É. Guazzelli, Experimental study of the sedimentation of dilute and semi-dilute suspensions of fibres, *J. Fluid Mech.* 384 (1999) 133–158.
- [59] J.E. Butler, E.S. Shaqfeh, Dynamic simulations of the inhomogeneous sedimentation of rigid fibres, *J. Fluid Mech.* 468 (2002) 205–237.
- [60] E. Guazzelli, J. Hinch, Fluctuations and instability in sedimentation, *Annu. Rev. Fluid Mech.* 43 (2011) 97–116.
- [61] E. Nazockdast, A. Rahimian, D. Zorin, M. Shelley, A fast platform for simulating semi-flexible fiber suspensions applied to cell mechanics, *J. Comput. Phys.* 329 (2017) 173–209, <https://doi.org/10.1016/j.jcp.2016.10.026>, <https://linkinghub.elsevier.com/retrieve/pii/S0021999116305241>.
- [62] E. Gauger, H. Stark, Numerical study of a microscopic artificial swimmer, *Phys. Rev. E* 74 (2) (2006) 021907.
- [63] P.J. Žuk, A.M. Stowicka, M.L. Ekiel-Jeżewska, H.A. Stone, Universal features of the shape of elastic fibres in shear flow, *J. Fluid Mech.* 914 (2021).
- [64] K. Ichiki, Improvement of the stokesian dynamics method for systems with a finite number of particles, *J. Fluid Mech.* 452 (2002) 231.
- [65] P. Virtanen, R. Gommers, T.E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S.J. van der Walt, M. Brett, J. Wilson, K.J. Millman, N. Mayorov, A.R.J. Nelson, E. Jones, R. Kern, E. Larson, C.J. Carey, Í. Polat, Y. Feng, E.W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E.A. Quintero, C.R. Harris, A.M. Archibald, A.H. Ribeiro, F. Pedregosa, P. van Mulbregt, SciPy 1.0 contributors, SciPy 1.0: fundamental algorithms for scientific computing in Python, *Nat. Methods* 17 (2020) 261–272, <https://doi.org/10.1038/s41592-019-0686-2>.
- [66] C. Voglis, I. Lagaris, A rectangular trust region dogleg approach for unconstrained and bound constrained nonlinear optimization, in: *WSEAS International Conference on Applied Mathematics*, 2004, p. 7.
- [67] R. Jahn, A.-M.M. Schmid, Revision of the brackish-freshwater diatom genus *Bacillaria* Gmelin (Bacillariophyta) with the description of a new variety and two new species, *Eur. J. Phycol.* 42 (3) (2007) 295–312, <https://doi.org/10.1080/09670260701428864>, <http://www.tandfonline.com/doi/abs/10.1080/09670260701428864>.
- [68] T.T. Bringley, C.S. Peskin, Validation of a simple method for representing spheres and slender bodies in an immersed boundary method for Stokes flow on an unbounded domain, *J. Comput. Phys.* 227 (2008) 5397, <https://doi.org/10.1016/j.jcp.2008.01.048>.
- [69] dskeet, Photo 11624974, <https://www.inaturalist.org/photos/11624974>.
- [70] J.J.L. Higdon, A hydrodynamic analysis of flagellar propulsion, *J. Fluid Mech.* 90 (04) (1979) 685, <https://doi.org/10.1017/S0022112079002482>, [http://www.journals.cambridge.org/abstract\\_S0022112079002482](http://www.journals.cambridge.org/abstract_S0022112079002482).
- [71] S.J. Lighthill, *Mathematical Biofluidynamics*, Society for Industrial and Applied Mathematics, 1975, <http://epubs.siam.org/doi/book/10.1137/1.9781611970517>.
- [72] E. Lauga, T.R. Powers, The hydrodynamics of swimming microorganisms, *Rep. Prog. Phys.* 72 (9) (2009) 096601, <https://doi.org/10.1088/0034-4885/72/9/096601>, <https://iopscience.iop.org/article/10.1088/0034-4885/72/9/096601>.
- [73] J.S. Guasto, R. Rusconi, R. Stocker, Fluid mechanics of planktonic microorganisms, *Annu. Rev. Fluid Mech.* 44 (1) (2012) 373–400, <https://doi.org/10.1146/annurev-fluid-120710-101156>.
- [74] L.T. Nielsen, T. Kjørboe, Foraging trade-offs, flagellar arrangements, and flow architecture of planktonic protists, *Proc. Natl. Acad. Sci.* 118 (3) (2021), <https://doi.org/10.1073/pnas.2009930118>, arXiv: <https://www.pnas.org/content/118/3/e2009930118.full.pdf>, <https://www.pnas.org/content/118/3/e2009930118>.
- [75] L. Turner, W.S. Ryu, H.C. Berg, Real-time imaging of fluorescent flagellar filaments, *J. Bacteriol.* 182 (10) (2000) 2793–2801, <https://doi.org/10.1128/JB.182.10.2793-2801.2000>, arXiv: <https://jb.asm.org/content/182/10/2793.full.pdf>, <https://jb.asm.org/content/182/10/2793>.
- [76] J. Zhang, M. Chinappi, L. Biferale, Base flow decomposition for complex moving objects in linear hydrodynamics: application to helix-shaped flagellated microswimmers, *Phys. Rev. E* 103 (2021) 023109, <https://doi.org/10.1103/PhysRevE.103.023109>, <https://link.aps.org/doi/10.1103/PhysRevE.103.023109>.
- [77] D.J. Smith, A boundary element regularized stokeslet method applied to cilia- and flagella-driven flow, *Proc. R. Soc. A, Math. Phys. Eng. Sci.* 465 (2112) (2009) 3605–3626, <https://doi.org/10.1098/rspa.2009.0295>.
- [78] K. Ishimoto, Bacterial spinning top, *J. Fluid Mech.* 880 (2019) 620–652, <https://doi.org/10.1017/jfm.2019.714>.
- [79] D. Saintillan, M.J. Shelley, Instabilities and pattern formation in active particle suspensions: kinetic theory and continuum simulations, *Phys. Rev. Lett.* 100 (2008) 178103, <https://doi.org/10.1103/PhysRevLett.100.178103>, <https://link.aps.org/doi/10.1103/PhysRevLett.100.178103>.
- [80] E. Lushi, Stability and dynamics of anisotropically tumbling chemotactic swimmers, *Phys. Rev. E* 94 (2016) 022414, <https://doi.org/10.1103/PhysRevE.94.022414>, <https://link.aps.org/doi/10.1103/PhysRevE.94.022414>.
- [81] E. Lushi, H. Wioland, R.E. Goldstein, Fluid flows created by swimming bacteria drive self-organization in confined suspensions, *Proc. Natl. Acad. Sci.* 111 (27) (2014) 9733–9738, <https://doi.org/10.1073/pnas.1405698111>, arXiv: <http://www.pnas.org/content/111/27/9733.full.pdf>, <http://www.pnas.org/content/111/27/9733.abstract>.
- [82] A. Scagliarini, I. Pagonabarraga, Hydrodynamic and geometric effects in the sedimentation of model run-and-tumble microswimmers, *Soft Matter* (2022), <https://doi.org/10.1039/D1SM01594J>.
- [83] K. Qi, E. Westphal, G. Gompper, R.G. Winkler, Emergence of active turbulence in microswimmer suspensions due to active hydrodynamic stress and volume exclusion, *Commun. Phys.* 5 (1) (2022) 1–12, <https://doi.org/10.1038/s42005-022-00820-7>.
- [84] H. Childs, E. Brugger, B. Whitlock, J. Meredith, S. Ahern, D. Pugmire, K. Biagas, M. Miller, C. Harrison, G.H. Weber, H. Krishnan, T. Fogal, A. Sanderson, C. Garth, E.W. Bethel, D. Camp, O. Rübél, M. Durant, J.M. Favre, P. Navrátil, Visit: an end-user tool for visualizing and analyzing very large data, in: *High Performance Visualization—Enabling Extreme-Scale Scientific Insight*, 2012, pp. 357–372, <https://escholarship.org/uc/item/69r5m58v>.
- [85] K. Maeda, Y. Imae, J.I. Shioi, F. Oosawa, Effect of temperature on motility and chemotaxis of *Escherichia coli*, *J. Bacteriol.* 127 (3) (1976) 1039–1046, <https://doi.org/10.1128/jb.127.3.1039-1046.1976>, arXiv: <https://journals.asm.org/doi/pdf/10.1128/jb.127.3.1039-1046.1976>.
- [86] E. Lauga, W.R. DiLuzio, G.M. Whitesides, H.A. Stone, Swimming in circles: motion of bacteria near solid boundaries, *Biophys. J.* 90 (2) (2006) 400–412, <https://doi.org/10.1529/biophysj.105.069401>, <https://www.sciencedirect.com/science/article/pii/S0006349506722214>.
- [87] C. Dombrowski, L. Cisneros, S. Chatkaew, R.E. Goldstein, J.O. Kessler, Self-concentration and large-scale coherence in bacterial dynamics, *Phys. Rev. Lett.* 93 (2004) 098103, <https://doi.org/10.1103/PhysRevLett.93.098103>, <https://link.aps.org/doi/10.1103/PhysRevLett.93.098103>.
- [88] D. Saintillan, M.J. Shelley, Orientational order and instabilities in suspensions of self-locomoting rods, *Phys. Rev. Lett.* 99 (2007) 058102, <https://doi.org/10.1103/PhysRevLett.99.058102>, <https://link.aps.org/doi/10.1103/PhysRevLett.99.058102>.
- [89] A.P. Berke, L. Turner, H.C. Berg, E. Lauga, Hydrodynamic attraction of swimming microorganisms by surfaces, *Phys. Rev. Lett.* 101 (2008) 038102, <https://doi.org/10.1103/PhysRevLett.101.038102>, <http://link.aps.org/doi/10.1103/PhysRevLett.101.038102>.
- [90] S.E. Spagnolie, E. Lauga, Hydrodynamics of self-propulsion near a boundary: predictions and accuracy of far-field approximations, *J. Fluid Mech.* 700 (2012) 105–147, <https://doi.org/10.1017/jfm.2012.101>.
- [91] S.M. Mousavi, G. Gompper, R.G. Winkler, Wall entrapment of peritrichous bacteria: a mesoscale hydrodynamics simulation study, *Soft Matter* 16 (20) (2020) 4866–4875.
- [92] D.C. Morse, Theory of constrained brownian motion, *Adv. Chem. Phys.* 128 (65–189) (2004) 110.