Models of Microswimmer Locomotion

by

Frank Nguyen

A dissertation submitted in partial fulfillment of

the requirements for the degree of

Doctor of Philosophy

(Chemical and Biological Engineering)

at the

UNIVERSITY OF WISCONSIN - MADISON

2018

Date of final oral examination: July 30, 2018

This dissertation is approved by the following members of the Final Oral Committee:

Michael D. Graham, Professor, Chemical and Biological Engineering

James B. Rawlings, Professor, Chemical and Biological Engineering

Reid C. Van Lehn, Assistant Professor, Chemical and Biological Engineering

Victor Zavala, Assistant Professor, Chemical and Biological Engineering

Saverio E. Spagnolie, Assistant Professor, Mathematics

Dedication

I'd like to thank my defense committee Profs. Rawlings, Van Lehn, Zavala, and Spagnolie. And of course, I'd like to express the greatest gratitude to my advisor Mike Graham for these last five years of grad school, and for his immense help in setting me up for a strong career to follow.

To everyone else, I love you all so much, and thank you for all that you've done for me.

To my parents, Cuong Nguyen and Nancy Hiep Huynh, a special message: Không gì trong thế giới này quang trọng bằng gia đình. Bố mẹ phải vượt qua quá nhiều sự đao khổ để tạo ra cuộc sống thật tốt cho con và chị. Dù mình là gia đình nghèo, qua 26 năm nay, dù cách xa nhà bao nhiêu, bố mẹ đã lo cho con từ học hành, ăn uống, quần áo, và nhà cửa. Con sẽ không bao giờ quên những gì mà bố mẹ đã làm cho con. Vì con đã xa nhà quá lâu, lúc nào con cũng nhớ nhà, và lúc nào con cũng nghĩ về gia đình. Cảm ơn bố mẹ – con thật sự thương hai ngừơi với hết lòng.

To my sister, Kimberly Nguyen: You are a true hero in every sense of the word, and I am so thankful to have had you as a sister, a role model, a mentor, a friend, and everything in between. Thank you for taking care of me through all these years, for making sure I always had food to eat, for making sure my stuff was always as good as the other kids' stuff, for always encouraging me to give my best efforts, and for yelling at me for being stupid. I hope

I've made you proud through the years. I can never fully repay you everything, but I hope I can start by being a good uncle.

To Anna Pang and Rothpanhar Ross: you are the reason I made it to the Promised Day. Your friendship has meant everything to me. Thank you for all that you've done for me, for always listening to me, and for helping me find my way through the darkest of times. I hope I can do the same for you one day.

To my dearest crew from La Quinta High School, Kevin Tran, Mary Nguyen, Samuel Nguyen, Denise Huynh, Magie Nguyen, and Xuan Dang: a collective love of food and desire to avoid strenuous exercise brought us all together so many years ago, and has kept us together ever since. Thank you for always being there, from high school graduation, seeing me off to college (props to Mary for sticking with me), and then grad school. Thank you for celebrating every Christmas with me, for making sure I never feel alone, and for cheering me on as I write this dissertation. Should the Promised Day pass, Wagyu beef. And lastly, to Matthew Huynh, rest in peace, my dear friend. I will never forget seeing you and Kevin at the bus stop saying good-bye as I left for Berkeley, and that you drove me to the airport at 5:30 AM to see me off to Wisconsin. It breaks my heart to be finishing this journey without you, but your memory will always live on.

To my childhood friends, Christina Nguyen and Nha Tong: we've come a long way from being the kids terrorizing the neighborhood. You're the oldest friends I have. Thanks for sticking with me, and here's to bright futures for all of us.

To my grad school roommates, Parth Mangrolia and Kevin Barnett, who've been with me since the beginning, from East Johnson to East Gorham to North Butler. It's been a blast living with y'all. And a special nod to de facto roommate Michael James Risbeck – thank

you for all your wise counsel through the years, for responding to most of my vague immediate invitations, and for understanding most of the things I say. To Paul Adamczyk, thank you for pushing me to do crazy things and for always providing a listening ear. To all of you, we've shared so many meals, games, holidays, and milestones together, as well as everything in between. Somehow we all made it to the end, and I will miss you all dearly.

To all the other Wisconsin people who've made this a memorable 5 years:

- The remaining fifth year crew: Travis & Hannah Nelson, Joe Chada, Loukas Goulatis, Nestor Hernandez, Dan & Alisa McClellan, Merve Özen.
- To Hector/Aaron Fuster, there aren't enough words to describe all the things you have done for me. You are indeed precious cinnamon, and you live up to the dream of an American melting pot. Thank you for always being there for me, you are truly one of the greatest friends I've had the privilege of knowing.
- To my other favorites: Taylor "Tay Tay" Cook, and Keishla Rivera Dones, also known as Kei$$hla, Ke$$ha, Kesha, Keesha, Keeshla, Caysha, Kiki, Keeshermos, etc.
- Sandy Chen and Alec Joseph Linot - you've been stuck sitting across from me every day now for awhile. Thank you for always letting me complain, and thank you especially for listening to me when it mattered most. No office will be the same without you two. Especially to Alec Joseph, thank you for driving me to Minnesota, for looking out for me, and for helping me start a new chapter in life.
- Jonathan "Mustache" Sheavly, I've never met anyone as perpetually happy as you. Mike "Droops" Jindra, you're one of a kind – keep being you and thanks for looking out for me.
- Tava Das thank you for helping with my transition to Madison, and for making sure I wasn't homeless. Ben Wilson for many a lively discussion.
- Curran Gahan – you're the life of the party now. Make me proud.

- Everyone else: ChEGS, Throwmodynamics, the Van Lehns, and the kids, aka the rambunctious go-getting first years. I'll miss you all.

To my Graham group research members, past and present: Kushal Sinha, Jae Sung Park, Rafael Henriquez Rivera, Sandy Wang (the original), Anubhav, Joshua Duncan, Xiao Zhang, Ashwin Shekar, Eric Yu, Alec Linot, Kevin Zeng, and a big thank you to Sarit Dutta for all the programming help.

To Stephanie To and Christina Hung, my friends since 7th grade at McGarvin, thanks for sticking with me all these years.

To Thomas Dursch, Jr., Clay Radke, John Prausnitz, and everyone at the Radke lab in Berkeley, grad school would not have been possible without you. To another Berkeley alum, Sahil Sheth, you're one of the few people in my life sassier than me. Never change.

To University Health Services, way to come in clutch. Jo Hoese, thank you for all your help.

And lastly, to two very, very special teachers. Alexis Raleigh, my eighth grade science teacher at McGarvin – you made science so much fun, and I have you to thank for starting me on the path toward engineering. Even 13 years later, I still very much enjoy our meals and chats every winter. Barbara Lyon, my senior English teacher at La Quinta – you were a teacher who cared for her students beyond the classroom and beyond high school. Thank you for steering me and so many others on the path to success, and always welcoming us back with open arms.

Abstract

Bacteria and microorganisms are ubiquitous throughout the world, from our bodies to our environments. They provide ideal blueprints to study locomotion in Stokes flow, where viscous drag dominates motion. In Chapter 1, we introduce the mathematics characterizing Stokes flow, and discuss major bacterial locomotion phenomena documented in literature. We emphasize studies of stability and elasticity of bacterial flagella, external appendages aiding locomotion, and the hook joint connecting flagella and cell. We complete our overview examining effects of phase transformations, hydrodynamic interactions, and complex environments. Beyond natural bacteria, we examine artificial and biohybrid microswimmers that motivate our exploratory work.

Chapter 2 examines the relation between hook buckling and uniflagellar locomotion. A key flexibility number measures the hook flexibility relative to flagellar thrust; this parameter and the swimmer geometry determine the stability of straight swimming. We first perform an analytical analysis with two simple toy models, and then perform numerical simulations using a full model with a helical flagellum. In all models, a critical flexibility number separates equilibrium configurations into straight or bent, and for the full model, separates trajectories into straight or helical. With the full model, we also see that for a given body geometry, there exists a flagellar geometry that minimizes the buckling threshold[1].

In Chapter 3, we develop a model for a swimmer consisting of a rigid spherical cell body and an arbitrary number of connected elastic flagella, complete with a flagellar discretization

[1]This portion of the abstract has been adapted from the publication Nguyen et al. *Biophys. J.*, 112, 2017

scheme. For this general model, we incorporate two flexibility numbers for the hook and the flagellar filament itself, as well as hydrodynamic interactions, and develop a numerical projection method to perform simulations. We use this general model to study free and bounded swimming for a bacterium, and a biohybrid swimmer.

Chapter 4 examines the effects of flagellar number and flexibility on locomotion. Experiments show that uniflagellar swimmers typically require a stiff hook and flagellum to move, but multiflagellar swimmers require higher hook and flagellar compliance to swim. Thus for a single body and flagellar geometry, we analyze stability in the parameter space of hook and filament flexibility for a swimmer with one to four flagella. Our simulations indeed show that the uniflagellar swimming is stable only in a small, stiff region of flexibility parameter space, and we capture changes to stability with increasing flagellar number. Bi- and triflagellar swimmers display the ability to with and without bundling. Quadriflagellar swimmers with isotropic flagellar arrangement swim only at high hook flexibility, with swimming robust against flagellar distribution. In a final piece, we demonstrate diminishing speed returns with flagellar multiplicity with a simple model.

We end with two exploratory works. Chapter 5 examines locomotion of a bacterium near an infinite wall. We specify hydrodynamic interactions using image systems, and detail the progress in verifying key experimental phenomena associated with wall swimming. In Chapter 6, we create a detailed model bioybrid driven by beating cardiomyocytes on a flexible (straight) flagellum. We simulate single-tail swimmers to show the range of trajectory speeds and curvatures, and show the potential for further modification with multiple tails, multiple beats, or other external dynamics.

The collective results shown in this work provide deeper insight into understanding the design and function of natural bacteria, particularly on the relationship between flexibility, stability, and flagellar multiplicity. The knowledge provided here can also be used to improve the design and function of artificial swimmers (e.g. biomedicine, microfluidics), and hopefully expand the potential horizons and applications of microswimmers in general.

Contents

List of Figures

Chapter 1

General introduction

Bacteria and small organisms are found in nearly evey aspect of the world and its operations [8, 65]. Human bodies host millions of bacteria responsible for everything from digestion to infection. They are widely used for a variety of industrial applications, ranging from the making of food products such as cheese and vinegar to the formation of biofilms for oil extraction and the transformation of glycerol to ethanol. In agriculture, the movement of bacteria through soil is a key factor in increasing nitrogen quality for improved crop nutrition and yield. Bacteria and microbes are also ubiquitous in the environment. Amoebas move freely through lakes, algae periodically appear in oceans, and bacteria exist on virtually every surface of larger organisms as well as man-made structures. They are known to be some of the longest-lasting lifeforms on this planet, having been in existence for millions of years. Their functions and life cycles have been studied extensively in biological fields but only in the last sixty years have scientists and engineers begun intensive investigation into the physics of bacterial motion [65].

The size of bacteria is on the order of microns, and they move vastly different from macroscopic animals such as fish and humans, even in a simple fluid such as water. In fluid dynamics, the flow regime of an object is governed by a quintessential dimensionless group called the Reynolds number: $Re = \rho v L/\mu$, where ρ is the density of the fluid medium, v the velocity of the swimmer, L the swimmer's characteristic length, and μ the dynamic viscosity

of fluid. The Reynolds number is often interpreted as a ratio of inertial to viscous forces. Microswimmers typically live in a low-Re regime where viscosity dominates. Thus, while a fish may swim by imparting momentum on its surroundings, a bacterium cannot, as the viscous surroundings effectively dampen its momentum [65]. Instead, microswimmers must resort to seemingly unconventional techniques, such as altering their geometry and utilizing appendages, to move [81]. A common type of biological swimmer consists of a cell body and one or more appendages known as flagella, and an embedded motor in the body spins the flagella to swim [11].

While early fluid dynamic studies established the mathematical possibility of low Re swimming, massive improvements in technology have led to experiments confirming these hypotheses as well as revealing previously unknown phenomena. Precise forces and torques on single cells can be measured, swimming trajectories and shape changes can be visualized, and a multitude of environments may be imposed on test microswimmers [65, 68, 71, 89]. Improvements in computation have greatly facilitated numerical simulations of microswimmers as well [50, 93]. With full empirical and theoretical characterizations of microswimmers well within reach, recent attention has also turned to the synthesis of artificial microswimmers whose initial applications range from targeted drug delivery to microscale towing, and potentially allow for a far greater degree of customization and optimization [65].

1.1 Dynamics of low *Re* swimming

1.1.1 Governing equations

Reynolds numbers for bacteria range from 10^{-5} for a small species such as *E. Coli* and 10^{-1} for a large species such as *Paramecium*, thus placing them in a regime where $Re \to 0$ [12, 65, 81]. For this regime, the incompressible Navier-Stokes equation for a Newtonian fluid simplifies to Stokes' equation (momentum conservation) and continuity (mass conservation):

$$-\nabla p + \mu \nabla^2 \mathbf{u} = \mathbf{0}, \quad \nabla \cdot \mathbf{u} = 0 \tag{1.1}$$

where $\mathbf{u}$ is the fluid velocity, p the dynamic pressure, and μ the dynamic viscosity. The fundamental solution of Eq. (1.1) is the stokeslet:

$$\mathbf{u}(\mathbf{r}) = \mathbf{S}(\mathbf{r} - \mathbf{r}') \cdot \mathbf{F}, \quad \mathbf{S}(\mathbf{r}) = \frac{1}{8\pi\mu} \left(\frac{\mathbf{I}}{|\mathbf{r}|} + \frac{\mathbf{rr}}{|\mathbf{r}|^3} \right) \tag{1.2}$$

where $\mathbf{r}$ is the position vector and $\mathbf{S}$ is the Oseen tensor [41]. The linearity of Stokes' equation leads to a simple relation between kinematics (force exerted $\mathbf{F}$, torque exerted $\mathbf{T}$) and kinetics (velocity $\mathbf{v}$, angular velocity $\boldsymbol{\omega}$):

$$\begin{pmatrix} \mathbf{F} \\ \mathbf{T} \end{pmatrix} = \begin{pmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B^T} & \mathbf{C} \end{pmatrix} \begin{pmatrix} \mathbf{U}_\infty - \mathbf{u} \\ \boldsymbol{\Omega}_\infty - \boldsymbol{\omega} \end{pmatrix} \tag{1.3}$$

where the subscript ∞ denotes background flow, and the matrix shown is the *resistance matrix* [59]. Eq.(1.3) is the basis of much theoretical and experimental work. Additionally, since inertia is negligible for swimmers in the Stokes regime, writing Newton's law for the swimmer results in instantaneous force and torque balances:

$$\sum_i \mathbf{F}_i = \mathbf{0}, \quad \sum_i \mathbf{T}_i = \mathbf{0} \tag{1.4}$$

Now that the mathematical framework of low Re swimming is established, the possibility of low-Re motion can be addressed from a rigorous mathematical standpoint, as well as quantitative models characterizing flow and swimmer movement.

1.1.2 Early mathematical developments

An initial answer comes from analyzing idealized representations of flagella. A classic example is Taylor's swimming sheet, an infinite sheet in viscous fluid where small-amplitude transverse waves were propagated down the length of the sheet, similar to a bending wave

passed down the length of a spermatazoa flagellum [106]. By solving Eq. (1.1), Taylor found that material points on the sheet moved opposite to the propagation direction as a consequence of induced vorticity in the flow [65, 106]. Taylor also considered the more specific case of propagating a transverse wave down an infinite cylinder, a necessary step to studying real 3D deformations associated with organisms, finding that backwards propagation also resulted in forward motion [107].

Gray and Hancock applied these results to study the propulsion of sea-urchin spermatozoa in free flow [39]. By explicitly modeling a flagellum as a sine wave, they showed mathematically that the undulation of the flagellum produced a thrust force in the axis of propulsion when drag anistropy exists [39]. By assuming forces on individual flagellar segments were only locally significant and that local drag was simply proportional to velocity, they integrated local over the entire flagellum to obtain the total force. Comparison with Eq. (1.3) yielded drag coefficients, and these calculations formed the basis for *resistive force theory* (RFT) [39, 55, 89]. This is the simplest theory used to calculate resistance tensors in Eq. (1.3). Lighthill developed another set of RFT drag coefficients, but a new theory was required for improved accuracy and refinement [69].

Slender body theory (SBT) was next formulated to reconcile the shortcomings of RFT, namely the omission of hydrodynamic forces between flagellar segments and their cumulative effects on motion [55]. Lighthill highlighted this importance by utilizing a distribution of Stokeslets and source dipoles in the interior flagellar centerline to describe the forces on a flagellum [69]. Since Stokeslets decay as $1/r$, they do induce long-range effects that must be considered, while the dipole flow of order $1/r^3$ is negligible. To avoid the use of singularities in numerical simulations, Cortez et. al. introduced a method of regularized Stokeslets on the outer flagellar surface to approximate forces and avoid singularities [24]. Rodenborn et. al. compared the calculated drag coefficients of Gray and Hancock RFT, Lighthill RFT, Lighthill SBT, and Cortez with experimental values for a rigid helix, concluding that Lighthill SBT and Cortez's model is generally more accurate than RFT theory [89], with improved accuracy

outside the range of typical flagellar geometries. The improvement of drag coefficients for complex flagellar geometries allows Eq. (1.3) to be used without extensive computation and discretization.

1.1.3 Limitations of low Re swimming

While it is useful identifying particular parameters allowing motion, it is equally important to lay out all the necessary conditions for motion to occur at low *Re*. A fundamental consequence of the linearity and time-independence of Stokes equation, Eq. (1.1), is *kinematic reversibility* [65, 81]. Lauga summarizes the two principal results of kinematic reversibility and their mathematical proofs [65]. First, the distance travelled by a swimmer depends only on its geometry, the sequence of shapes it adopts in motion, not on the rate the geometry changes between configurations. The second property is known as the *scallop theorem*: if the swimmer's sequence of configurations is the same viewed either forward or backward in time, then the swimmer will not move on average. Purcell demonstrated the latter theorem with an example of a single-hinged scallop periodically opening and closing [81]. Ultimately, the scallop theorem presents a very real obstacle influencing the design of both natural and artificial swimmers.

From a theoretical standpoint, there exist ways of bypassing the scallop theorem. An object may avoid reciprocal motion if it can cycle through a configuration sequence without symmetry [64, 110]. Such objects can be readily found in nature, and also developed artificially in a laboratory setting.

1.2 Natural microswimmers

1.2.1 Physical description

Bacteria are designated as the quintessential examples of low Reynolds number swimming. As stated earlier, a typical bacterium is compose of a cell body, usually rigid, and one or

multiple flagella. Most bacteria (>90%) are *monotrichous* or uniflagellar, consisting of only a single tail [67]. Examples include *Vibrio alginolyticus*, *Rhodobacter Sphaeroides* [31, 98]. The remainder of bacteria are multiflagellar. The descriptor *peritrichous* is usually reserved for bacteria whose flagella are randomly expressed (and sometimes excessively expressed) on the surface of the cell body, such as in *E. coli*, *B. subtilis*, *S. typhimurium* [31]. Other descriptors exist to classify the placement of flagella relative to each other, e.g. *lophotrichous* for flagella all located at the same pole as in *Chromatium okenii*, *bilophotrichous* for flagella located at two poles as in variants of *Halobacterium salinarium* [31]. There are also species of bacteria that express multi-functional flagella. For instance, *Agrobacterium tumafaciens* can express tufts of polar flagella (i.e. located at poles of body) alongside a few lateral flagella [31]. Electron microscopy images for *Shewanella putrafaciens* and *Bradyrhizobium diazoefficiens* reveal that these species have two distinct sets of flagella – a thick, long polar flagellum, and a series of thinner, shorter lateral flagella [17, 83].

Across the various flagellar morphologies above, the biology and operation of flagella is usually the same. The base of the flagellum is attached to the cell wall via a short elastic filament called a hook. The hook is anchored to a motor composed of a static stator and free-spinning rotor. When the motor turns on a (constant) torque is applied to the hook. Because the hook properties are reminiscent of a universal joint with low bend and high twist resistance [35], the hook transmits most of the motor torque down the length of the flagellar filament [11, 26]. The rotation of the chiral flagellar helix yields no time-reversible symmetry in the sequence of shapes adopted in one rotational cycle, leading to net motion in the presence of a body [65].

1.2.2 Pushing and pulling

Experiments of bacterial movement show that swimmers exhibit an extensive range of motion, rather than solely one-dimensional thrust, both individually and in groups. In free flow, perhaps the most noticeable individual feat is the switching between running and tumbling,

or alternatively, “pulling" and “pushing," occurring when the motor spinning the flagellum suddenly stops and reverses direction [38, 65, 71, 98]. Without loss of generality, imagine a swimmer with a left-handed flagellum. Spinning this flagellum clockwise (when viewed behind the cell), will produce a propulsive force that pulls the cell, and puts the flagellum under tension. Spinning this flagellum counter-clockwise will produce a force that pushes on the cell and puts the flagellum under compression, and with compression comes elastic buckling.

When the flagellum is under tension or compression, the same mode applies to the elastic hook that connects it to the body, which can have a drastic effect on uniflagellar swimmers. For instance, Son et al. saw that when a *V. alginolyticus* cell switches from being pulled by the flagellum to being pushed, the hook buckles under a large force and torque, causing the swimmer to abruptly flick and change direction before stabilizing on a straight path [98]. It is postulated that the buckling mechanism is exploited to allow swift changes in direction in the absence of any other available mechanisms [13, 98]. Liu observed that the forward instability is not limited to a swift flick, as *Caulobacter crescentus* was observed taking a helical trajectory in the pull-to-push transition [71]. This hook buckling phenomenon has been also observed in simulations of swimmers [77, 96], and this subject matter is the focus of Chapter 2. We note that simulations in [51] show that flagellar flexibility also plays a crucial role in hook buckling, and we examine this interplay in a different light in Chapter 4.

The run and tumble motion of multiflagellar organisms such as *E. coli* does not rely on the same elastic instabilities as the uniflagellar case. Flagella are observed to bundle tightly in *E. Coli* and *Rhizobium melitoli* when pushing (usually in a single tight helix), unravel during a tumble, resulting in reorientation of the swimming direction and rebundling of the flagella as the swimmer again pushes [12, 38]. Further changes, such as the shape of the flagella themselves, can be differentiated during the run and tumble stages, as discussed in the next section.

Lastly, beyond the obvious differences in swimming direction, pushing and pulling also change the interaction of the swimmer with the surrounding fluid due to conservation of mass. In a pusher, the flagellum displaces fluid behind it propels the cell, and the body is pushing fluid forward as it moves – the far-field flow is thus a dipole where the swimmer draws in fluid through its sides and expels it along the swimming direction [65]. This far-field effect has been directly measured in experiments by Drescher et al. for [29]. Through the same exercise, a puller in the far-field will pull in fluid in the swimming direction and expel it through the sides [65]. Though the differences in the fluid flows generated by pushers and pullers may not have a large impact on the swimming of an isolated organism in free flow, these differences can have a profound impact when considering hydrodynamic interactions between cells, particularly in suspensions, and in interactions with boundaries and other obstacles, as we summarize later.

1.2.3 Polymorphic phase transformations

The constituent protein of flagella, flagellin, can be arranged into exactly 12 shapes, 2 of which are straight and the remaining 10 helical [114]. Though a flagellum may have a desired equilibrium shape, it can undergo a polymorphic phase transformation and change its shape under certain conditions. This phenomenon is usually strongly associated with the push to pull transitions described above. The classical example from Berg are the polymorphic states of *E. coli* [11, 112]. When an *E. coli* cell tumbles, one of the flagella in the bundle changes shape from the normal left-handed helix to the (curly) right handed helix [11]. More changes occur before the bundle reforms and all the flagella return to their constituent states. The uniflagellar bacterium *R. sphaeroides* exploits polymorphic phase transformations to change directions, rather than exploiting elastic instabilities as mentioned above. Kühn et al. in Ref. [62] found that when the a *Shewanella putrefaciens* bacterium becomes stuck, it will utilize a phase transformation during a push to pull transition to wrap its flagellum around the cell body to escape traps in a corkscrew-like motion. We note though that polymorphic

phase transformations are not possible for all organisms. For example, it is thought that the sheath covering the flagella of *V. alginolyticus* prevents transformations from occurring.

Polymorphic phase transformations are crucial to include in simulations capturing the motion of cells such as *E. coli*. This is typically done by *a priori* incorporating known flagellar geometries as system parameters, and defining separate energy potentials for each geometry, as in Ref. [113]. Theoretical studies of phase transformations include Vogel et al. in Ref. [113] where the authors use force extension curves to find a relation between elastic parameters and phase transformations, and Ko et al. in Ref. [61], where the authors study phase transformations in response to motor reversal and external fluid flow.

1.2.4 Hydrodynamic interactions between cells

Hydrodynamic interactions play an even larger role in observed cell-cell interactions as well as cell-boundary interactionsas a result of induced velocity fields and velocity gradients [65]. Ishikawa et. al. simulated interactions between two spherical squirmers as well as two *Paramecium*, observing that hydrodynamic interactions result in near-field attraction, dramatic reorientation, and subsequent separation with no stable pairwise swimming [47, 49, 50]. For near-field interactions, Guellii's study of magnetotatic bacteria confirmed attraction between two side-by-side pushers, but also showed repulsion between two side-by-side pullers [40]. More detailed phenomenon include phase locking and synchronization, modeled by Elfring for the flagella of two spermatozoa using an extension of Taylor's swimming sheet [32, 106]. Qian et. al. examined this idea experimenting with and modeling two constant-torque rotating paddles in viscous fluid, seeing synchronization due to hydrodynamic interactions with flexibility in paddle supports [82]. The phenomena observed for neighboring cells contributes to swimming phenomena seen when extended to larger populations.

Due to the prevalence of bacterial suspensions, understanding how local swimming properties affect macroscopic behavior such as ordering and rheology is of great interest. Colonies of *Bacillus subtilis* were observed to form temporary whirls and jets on a length scale much

larger than the cell size [75]. Dombrowski et. al. noted in further experiments with *B. subtilis* that the randomness of the high-order structures was analogous to turbulence [28]. Simulations of active suspensions modeling swimmers as self-propelled rods also exhibit complex dynamics – pushers will form large-scale structures, while pullers do not display any interesting organization [44, 93]. Viscosity, another important feature of suspensions, also behaves curiously for active particles. Hatwalne et. al. modeled the viscosity of a dilute active suspension in Newtonian fluid based on Batchelor's theory (neglecting intracellular hydrodynamic interactions) [9, 42]. The effective viscosity contains two contributions, one from resistance to background flow, and another due to active flow generated by swimming [65] and is linear in volume fraction. Ishikawa followed this study by examining non-dilute suspensions and found that transient cell-cell configurations gave rise to a quadratic relation between effective viscosity and volume fraction [48], highlighting effects of hydrodynamic interactions for suspension.

1.2.5 Interaction with boundaries

Finally, swimming behavior near solid boundaries has also been extensively studied due to the prevalence of boundaries in applications such as biofilm formation, bacterial infection sites, confinement, and laboratory experiments in other fields. Lauga's review discerns three important swimming traits near boundaries [65]. First is the competition between increased viscous drag near walls and the resulting drag-based thrust coupled with changing local anisotropy [16, 56, 57]. Second is the change in trajectories for select organisms. For example, straight swimming *E. Coli* move in circles near walls due to a wall induced torque effectively altering the direction of the axisymmetric propulsive force [66]. Third is the wall effect on hydrodynamics, leading to increased attraction for pushers and deflection for pullers [43, 65]. This is further seen in the shear trapping of bacteria near walls (with other implications towards Brownian motion) [92].

The formation of biofilms is perhaps one of the most important and most clear consequences of bacterial attraction to boundaries. Bacteria containing subsets of lateral flagella will often use them to swarm on surfaces. For example, in *Aeromonas* species with lateral flagella will form biofilms, but that this effect was attenuated in various mutants of the wild-type cells [60].

Simulations of near-boundary swimming must satisfy a no-slip boundary condition at the boundary while also solving for motion of the swimmer. For a rigid swimmer, usually of fixed orientation, boundary element methods have been developed and extensively used in numerous studies, as in Refs. [15, 37, 79, 84]. Most studies of this kind rely on constant wave-form dynamics initially prescribed to a single flagellum by Phan-Thien et. al. [79], and do not allow for flagellar deformation. A model by Shum et al. in Ref. [96] also uses a boundary integral method, but allows changes in swimmer configuration via hook bending. In general, it is not particularly convenient or feasible (as we discuss later) to use a fine-mesh boundary integral method to model the deformation and motion of a thin filament such as a bacterial flagellum. As such, Watari et al. in Ref. [120] use a coarse-grain model of a swimmer represented by beads and springs to briefly study near-wall hydrodynamics. We present our own version study of a swimmer with deforming elastic flagella near a wall in Chapter 5.

1.2.6 Complex environments

A great number of experiments and simulations with bacteria and other swimmers are performed in very simple Newtonian fluids such as water. However, the reality is that most bacteria live in rather complex environments. These environments include soil and the human body (e.g. mucus tracts, organs, etc.), and can have great effects on locomotion. For instance, in viscoelastic fluids, Liu et al. in Ref [70] found that the swimming sped of a helix is maximized when the relaxation time of the fluid and the rotation rate of the helix are roughly the same. This same speed enhancement was also observed for helices in Ref. [100].

Swimmers may also encounter solid boundaries, as mentioned above, as well as discrete obstacles. For instance, simulations in Ref. [101] shows that swimmers can become trapped in endless orbit around colloidal-size spheres. Liu et al. in Ref. [72] found that the transport properties of *E. coli* can be heavily affected by placing them in a medium such as sand.

Simulations with complex environments are inherently more difficult than analogous simulations in Newtonian fluid, but much effort has been made on this front. For instance, for viscoelastic fluids, numerical solutions typically require coupling the Stokes equations with a consitutive equation for the fluid stress, as in Ref. [100]. In another example, Wróbel et al. utilized connected Maxwell elements (springs and dashpots) to represent a viscoelastic network reminiscent of cervical mucus, and were able to use regularized Stokeslet to model the hydrodynamics of a sperm cell moving through this network [123, 124]

1.3 Artificial and biohybrid microswimmers

Discussions of overcoming reciprocal motion also extend to artificial swimmer design. The most basic example is Purcell's swimmer, consisting of three rigid links connected by two hinges; it swims by non-reciprocal motions in its arms (similar to wave propagation for small strokes), but it is not very efficient. [10, 81] – attempts to optimize the strokes patterns can be found in Ref. [104] Another example from Najafi consists of three linked spheres with periodically changing separation lengths [76]. A third example from Avron is a system of two connected spheres where the separation distance and the size of one sphere changes simultaneously, and in fact may be more efficient than classical swimmers for large stroke sizes [7]. All have been shown to be capable of propulsive dynamics [7, 10, 76].

In the world of microfabrication, there has been much effort in the design and construction of microswimmers, as these can be subject to external control via constant magnetic fields. In one such endeavor, Zhang et al. [126] attached metal helices to soft, flexible heads to study motion. For another example, a Ding et. al in Ref. [27] developed micro-scale

metal helices roughly the same size as bacterial flagella for the capture and transport of colloidal droplets, with the ultimate goal being the fabrication of a fully motorized swimmer [27]. There have also been increased efforts to use these artificial microswimmers directly in biological environments in conjunction with magnetic control. In one experiment, Dreyfus et al. attached a linear magnetic tail to a red blood cell and successfully propelled the structure using an external field [30]. A swimmer from Huang et al. in Ref. [45] consists of a metal head attached to a flexible hydrogel tail capable of adapting its shape depending on environment and temperature. Medina-Sánchez et al. in Ref. [74] used metal helices to capture low-motility sperm cells and deliver them directly to the surface of an egg cell as a novel approach to fertilization.

Swimmer hybrids of consisting of both natural and artificial components have also been constructed. Ali et al. in Ref. [4] denatured the flagellar constituent protein and fully reconstituted flagella onto the surface of magnetic nanoparticles to test motion control, with emphasis on bundle formation. Other incorporate active swimmers directly into their design. For instance, in Ref. [73], motile sperm are used to propel robotic microswimmers, and in Ref. [103] a similar feat is achieved with motile bacteria. Bacteria themselves can also be converted into delivery systems, as Felfoul et al. in Ref. [34] grafted drug-containing nano-liposomes directly onto magneto-aerotactic bacteria for targeted delivery to tumors. But perhaps the major aim of studying hybrid swimmers is the search for a functioning fully self-propelled swimmer. Cardiomyocytes have proven to be a useful driving force for such swimmers, often in conjunction with thin polymer filaments or films [86, 87] and advanced printing techniques [102]. Williams et al. in Ref. [122] have produced biohybrid swimmers using beating cardiomyocytes grafted onto a flexible hydrogel. The hydrogel filament can then be attached to a static polymeric body to act as a flagellum facsimile or to form a cantilever-esque structure that moves along the base of a wall as in Ref. [19], or even as constituent components of a dolphin-shaped swimmer, as in Ref. [125]. Ultimately, further studies of the detailed dynamics would prove beneficial to the manufacturing of improved

swimmers.

1.4 Brownian motion and stochastic behavior

The locomotion of a single bacterium is generally unaffected by Brownian motion. Because the persistence length of a bacterium is on the order of centimeters, orders of magnitude larger than the micron scale length of the a bacterial flagellum, thermal fluctuations are not expected to play a significant role [65]. Indeed, straight runs of bacteria do not seem to indicate significant trajectory fluctuations [71, 98], and simulations of simple locomotion reproduce similar results while completely neglecting Brownian motion, such as Refs [1, 77, 96]. However, that is not to say that locomotion overall is completely deterministic. For instance, in the simple push-pull sequence in a *V. alginolyticus* trajectory, while the motion of each segment (push or pull) is deterministic, the transition times between push and pull are are stochastic, i.e. the transitions follow a Poisson process [98]. This may give the overall swimmer trajectory over a long time the semblance of a random walk or stochastic diffusion. A similar perspective can be applied to the run-and-tumble sequences of *E. coli* [12]. However, external factors due influence the occurrence (and thus frequency) of motor reversals that lead to these transitions [118]. Rusconi et al. in Ref. [92] showed that rotational diffusion can help a swimmer escape shear-trapping near boundaries. Beyond just locomotion, Vogel et al. in Ref. [114] show that thermal fluctuations can give rise to polymorphic phase transformations.

1.5 General models of swimmers

1.5.1 Single swimmers

The models developed for the study of bacterial locomotion vary greatly in level of detail depending on the problem studied, ranging from molecular dynamics (M.D.) studies to sus-

pension studies. On the smallest end, the aforementioned M.D. studies are not used to study locomotion per se, but rather to understand how the dynamics of molecular-scale bacterial components drive locomotion. A summary of a molecular-level study for the dynamics and behavior of the bacterial hook, including stochastic effects can be found in Ref. [111]. Studies of locomotion involving only a single swimmer are coarser than M.D., but in discretizing the swimmer maintain resolution to the scale of the flagellar filament radius. This is quite evident in simulations of rigid flagella as in Refs. [24, 46, 89], where a flagellar segment is represented by an n-gon of regularized Stokeslets. Boundary integral representations of rigid flagella utilize the same refinement, such as in the work of Shum et al. in Refs. [96, 97], though some earlier boundary integral works do utilize artificially thicker flagella, such as in Ref. [79]. In Refs. [46] and [96], the body is discretized to similar resolution as the flagella. The detailed resolution used in the above studies allows for the accurate calculation of near-field flows generated by swimmer motion, and can also be extended to systems of more than one swimmer, such as in Ref. [50], but not practical for suspensions. The models above are typically solved by formulating systems of linear equations that can be solved quite quickly, with easy interchange between open domains and half domains [15]. However, we do note that many models of the class mentioned above utilize fixed configurations of swimmers, and some also utilize prescribed kinematics, neither of which truly capture the fundamental physics and motion of a general bacterium.

Though the assumption of rigid flagella can be quite good for particular case studies of locomotion, flagella in general are elastic filaments. As we stated earlier, elasticity is important, for instance, in cells that bundle their flagella like *E. coli* or in cells that utilize deformations waves, such as sperm. Rigorously speaking, the surface of a flagellum could be treated as a deformable elastic membrane, and one could use membrane equations such as those in Ref. [80] to simulate flagellar motion. However, in addition to the enormous computational costs of using such a membrane, deformations, the near field flows generated by elastic deformations at the scale of the flagellar radius do not have a particularly important

effect on locomotion on the scale of the body size. Thus, a coarse grain approach may readily capture elastic phenomena with reasonable computational cost, and multiple such approaches exist in literature. One approach developed for fibers, as in Refs. [91, 94], treats an elastic filament as a series of rigid rods connected by torsion springs – this method was also used in Ref. [52] to discretize flagella. Another approach by Watari et al. [120] treats both the cell body and flagellum as small spheres connected by Fene-Frankel springs with corresponding bend, twist, and stretch potentials, though the others do note this is meant to be a qualitative treatment, as spheres will not quantitatively match the hydrodynamics of slender filaments. Perhaps the most common representation is to treat a flagellum as a series of connected nodes (or points) and edges, and to incorporate a classical Kirchhoff elastic energy to govern the twist and conformation along the length of the filament (and stretching energy for inextensibility). This representation has been used in Refs. [116, 117] for the study of helices and helical organisms, and in Refs. [1, 2, 62, 114, 115] for the study of bacterial flagella, though not all studies consider the role of hydrodynamic interactions, particularly between body and flagella.

The coarse-grained approaches above have 2 main drawbacks. First, while a method that uses stiff springs allows one the equations of motion of a swimmer as a full, but stiff, differential equation system, and thus will necessitate time steps much smaller than the rotation period of a flagellum. Second, the rotational friction of a flagellar segment along its axis is *very* small compared to any other friction coefficient, scaling as the square of the filament radius. Thus, when applying constant torque to a flagellum, time steps must still be very small in order to properly resolve the rotational phase of the flagellum. Bergou et al. in Ref. [14] present their Discrete Elastic Rods (DER) algorithm to avoid both issues mentioned above – inextensibility constraints are explicitly enforced, and twist is assumed to propagate instantaneously down the length of the filament since there is practically no resistance – inertial, viscous or otherwise. Jawed et al. in Ref [53] successfully used DER to model buckling of an elastic flagellum, but we note that in their case, the actuation

mechanism was a motor translating the proximal flagellar end in a circular trap, rather than axial rotation of the flagellar end. We are not aware of a fully working DER implementation in simulations of an elastic flagellum utilizing a torque/rotation driven boundary condition, though DER may prove to be a powerful computational tool in this field.

1.5.2 Systems of multiple swimmers

As the number of swimmers in a system continually grow, the near-field flows generated locally by a single swimmer become less and less important, thus allowing the use of more simplified swimmer representations. One widely used example are so-called Janus swimmers, where a swimmer is treated as an ellipsoid with one portion subject to a no-slip boundary condition, and the remaining portion subject to a no-stress boundary condition. This has been used in studies of single swimmers in Ref. [99] and in studies of suspensions by [44, 93], with the latter also incorporating a probability distribution equation for the orientations of individual swimmers. Rusconi et al. in Ref. [92] condense swimmers down to points moving in conjunction with a background flow, again with a stochastic component to govern their swimming orientation.

1.6 Outline of this work

The overarching goal of this dissertation is to provide a model framework for a microswimmer that can be used to build upon all of the work discussed up to this point. In Chapter 2, we develop a very simple model swimmer – linked rigid bodies connected by a flexible hook with no hydrodynamic interactions. This chapter studies locomotion on the most basic level, and allows us to use analytical equations in discussions of swimming and stability. In Chapter 3, we present a general model of a swimmer consisting of a rigid spherical body with to elastic flagella. We also include the dynamics and equations of motion that will be used in the remaining chapters. In Chapter 4, we study the motion of this swimmer with elastic flagella

in open domain, focusing on the relationship between flexibility and locomotion for both uni- and multiflagellar swimmers. The last two chapters present exploratory work building upon our general swimmer model. Chapter 5 studies the locomotion of a bacterial swimmer near a wall. In Chapter 6, we adapt our general swimmer model to simulate the motion of a biohybrid swimmer actuated by beating cells. The final chapter then summarizes the results of this dissertation and provides an outline on future directions for the model and work presented here.

Chapter 2

Linked rigid body swimmer model [1]

2.1 Introduction

To sample and respond to their environments, motile microorganisms display a wide variety of mechanisms for propelling themselves in fluid. Many species, such as *E. coli*, have a large number of long flagella that dynamically assemble into bundles as they rotate, propelling the microorganism. Others, including *Vibrio parahaemolyticus*, may display many flagella under some circumstances but only a single flagellum with different morphology under others. This diversity may confer selective advantages to swimmers utilizing different schemes in different environments, but the origin of this diversity is as yet unknown [18]. An important specific issue is the means by which cells reorient as they swim. *E. coli*, for example, reorients whenever one of the flagellar motors switches directions, causing the flagellar bundle to be temporarily disrupted [11]. This strategy is not available to *monotrichous* bacteria with only a single flagellum. Recent evidence [98] suggests that the singly flagellated organism *Vibrio alginolyticus* exploits a mechanical buckling instability of the hook protein connecting the cell body and flagellum to reorient. In the present chapter we build and analyze a series of simple micromechanical models, with varying complexity, of bacteria propelled by a single flagellum. Our aims are to characterize the parameter regime in which such instabilities

[1] The text of this chapter is adapted from the publication Nguyen et al. *Biophys. J.*, 112, 2017

occur and the dynamics of swimming within these regimes, as well as to find the minimal level of description required to capture the relevant phenomena.

In eukaryotic cells such as spermatozoa, and wild-type *Chlamydomonas*, motors distributed along the flagellar filament actuate the flagellum by generating helical waves or beating patterns [33, 65]. Simpler prokaryotic cells possess only a single rotary motor, composed of a static stator plus rotor, per flagellum. During locomotion, a load torque or "motor torque" is exerted by the motor, causing rotation of the rotor [11] and a corresponding counterrotation of the cell body to conserve angular momentum. The elastic hook that connects motor and flagellum transmits the torque to one end of the flagellar filament, inducing rotation in the entire flagellum. Because a flagellum is helical, the rotation-translation coupling produces a net force when the flagellum rotates around its own axis, thereby pushing or pulling the cell body and hook. Since the resultant drag from the cell body exactly balances the propulsive force, the hook is under tension for pullers and under compression for pushers. Experiments with both elastic and rigid flagella have shown that the hook will buckle under large enough compression, one consequence being misalignment between cell body and flagellum [53, 98]. Buckling has been observed in *Vibrio alginolyticus* when the hook unwinds and significantly weakens during a pull-to-push transition. This buckling manifests only as a brief trajectory "flick," as the hook winds up again and stiffens, realigning the swimmer [98]. Further evidence of buckling was inferred by Liu *et al.* from the helical trajectories of *Caulobacter crescentus*, caused by consistent propulsion off-axis to the cell body [71].

Bacterial motion is often modeled by solving the Stokes equations via boundary integral methods or by placing singularities (Stokeslets or regularized Stokeslets) on the surface of a rigid discretized swimmer with prescribed orientation and kinematics [24, 79, 84]. For instance, in a model of *B. subtilus*, Hyon *et al.* use regularized Stokeslets to examine helical trajectories arising from a number of fixed flagellar arrangements on the cell body [46]. Because assigning configurations to swimmers necessarily precludes the exploration of any hook dynamics or mechanics, other studies instead use various models incorporating elasticity

to examine the stability of the hook and of the overall flagellar filament [1, 52, 115, 120]. Indeed, models of a standalone elastic flagellum show instability in the equilibrium helical shape above a critical applied torque load / angular velocity [53, 115]. However, for flagella attached to cell bodies, experiments show that the hook can buckle even without significant deformation to the flagellar filament, highlighting the key role the local hook flexibility plays in determining the stability of swimming [98]. This latter observation forms the basis of models by Shum *et al.*, utilizing a rigid cell body and rigid helical flagellum connected by an elastic hook modeled explicitly as a Kirchoff rod to combine the evolution of the overall swimmer's configuration with an accurate boundary element method [96]. When the (dimensionless) ratio of the hook stiffness to motor torque is in an intermediate range (below we will call the inverse of this parameter the flexibility number), the swimming configuration is stable and the swimmer moves normally in straight line [96]. Outside this range however, the swimmer either wobbles inefficiently (the hook is too rigid) or the alignment angle between body and flagellum diverges quickly from equilibrium (the hook is too flexible) [96]. Though examining the trajectories associated with very large alignment angles is beyond the scope of their model, Shum *et al.* provides a very comprehensive picture of hook stability.

In the present work, we develop simple models that capture the relevant phenomena pertaining to monotrichous swimmers, namely the buckling of the hook in pushers. We first introduce the basic swimmer described by a rigid cell body and a rigid flagellum connected by a flexible joint. For simplicity, we neglect hydrodynamic interactions between body and flagellum. We progress through three levels of description: a static toy model, a dynamic toy model, and a full model. The first two toy models serve to analytically show the existence of buckled states in both a stationary and moving swimmer by prescribing a simplistic geometry to the flagellum and propelling it with an artificial force. In the full model, the flagellum is treated as a helix, and the application of a motor torque generates propulsion through the rotation/thrust coupling characteristic of chiral objects. Section 2.7 summarizes key results from numerical simulations of the full model, describing properties of straight and buckled

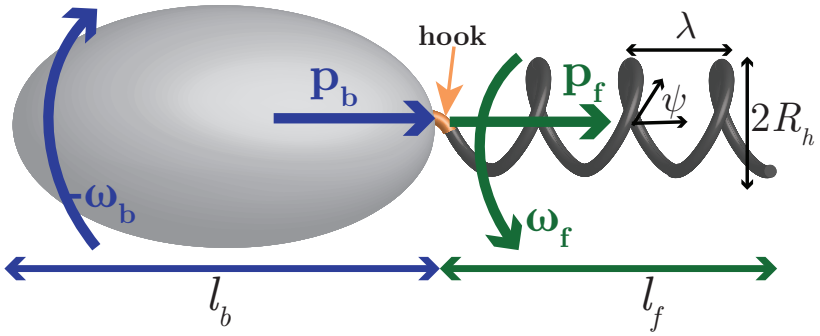

Figure 2.1: Schematic of the main swimmer model consisting of a body connected to a flagellum by a hook. In this illustration, the body is a prolate spheroid and the flagellum is a helix with a taper ($\ll l_f$) at the hook. Vectors $\mathbf{p}_i$ denote orientations and $\boldsymbol{\omega}_i$ rotation directions of the body ($i = b$) and flagellum ($i = f$).

configurations in swimmers. We also compare the stability behavior of the full model with the analytical stability results provided by the toy models. This analysis establishes the minimal criteria necessary to describe instabilities associated with swimmers and may shed insight on how organisms utilizing buckling as part of their locomotion scheme have evolved or adopted certain sizes or shapes and particular mechanisms of motion.

2.2 General linked rigid body model

Fig. 2.1 illustrates a swimmer consisting of a rigid cell body and rigid flagellum linked by a flexible hook. For general rigid shapes, we may write force and torque balances on each component i with $i = b$ (body) and $i = f$ (flagellum). The hook is not modeled explicitly as it is negligible in size relative to body and flagellum. For a cell in unbounded Stokes flow, neglecting hydrodynamic interactions between body and flagellum, these can be written:

$$-\mathbf{A}_i \cdot \dot{\mathbf{r}}_i - \mathbf{B}_i \cdot \boldsymbol{\omega}_i + \sum \mathbf{F}_i^{ext} = 0 \tag{2.1}$$

$$-\mathbf{B}_i^{\mathrm{T}} \cdot \dot{\mathbf{r}}_i - \mathbf{C}_i \cdot \boldsymbol{\omega}_i + \sum \mathbf{T}_i^{ext} = 0 \tag{2.2}$$

Vector $\mathbf{r}_i$ is the center of mass position and $\boldsymbol{\omega}_i$ the angular velocity. The tensors $\mathbf{A}_i$, $\mathbf{B}_i$, and $\mathbf{C}_i$ are the hydrodynamic resistances of component i, with $\mathbf{B}_i$ being the translation-rotation coupling. In these equations the first two terms incorporate the hydrodynamic forces and torques, respectively, exerted on component i by the surrounding fluid, while $\sum \mathbf{F}_i^{ext}$ and $\sum \mathbf{T}_i^{ext}$ are the total non-hydrodynamic forces and torques. The unit orientation vectors $\mathbf{p}_i$ define the orientation of the body and flagellum. Connectivity of the body and flagellum at the hook position requires that:

$$\mathbf{r}_b + \frac{l_b}{2}\mathbf{p}_b - \mathbf{r}_f + \frac{l_f}{2}\mathbf{p}_f = \mathbf{0}. \tag{2.3}$$

The overall configuration of the swimmer is described by a bending angle θ, where

$$\cos\theta = \mathbf{p}_b \cdot \mathbf{p}_f. \tag{2.4}$$

The plane of bending is span$\{\mathbf{p}_b, \mathbf{p}_f\}$, and the unit normal defining this plane is $\mathbf{n} = (\mathbf{p}_b \times \mathbf{p}_f)/\sin\theta$. The equilibrium configuration, denoting alignment between body and flagellum is characterized by $\theta_{eq} = 0$. "Straight swimming" corresponds to a very small (but not necessarily zero) value of θ and we will be interested in the stability of this state. Note that unlike in ref. [96], there is no explicit equation of motion for the hook, but its effects will be present via contributions to the external forces and torques. Before analyzing this model in detail in section 2.5, we will first prescribe simple symmetric geometries ($\mathbf{B}_i = 0$) and external dynamics in two "toy" models (sections 2.3 and 2.4).

2.3 Static toy model

Fig. (2.2) shows a very simple model of cell in compression, consisting of two rigid, slender rods (length l_i, radius a_i, aspect ratio $\epsilon \equiv a_i/l_i$) connected by a torsional spring (hook) of constant modulus κ and initially aligned along the x-axis. An external force $\mathbf{F}_P$ of constant

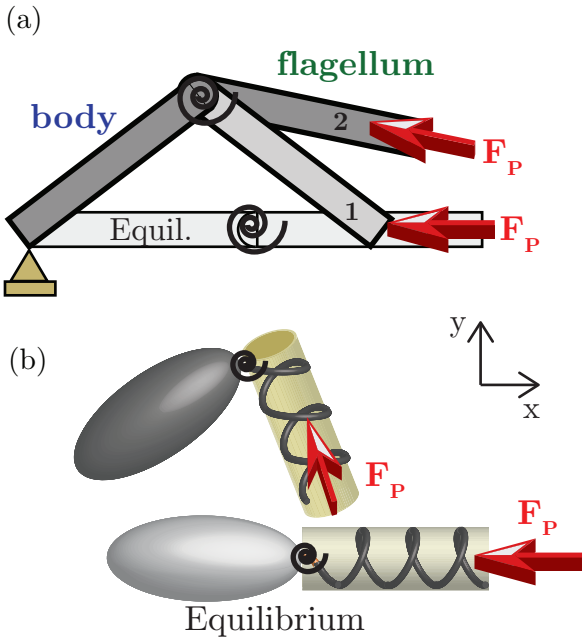

Figure 2.2: (a) Static toy model showing the body and flagellum as two connected rods. Propulsion is either aligned with the x-axis (Case 1) or the flagellum (Case 2) . (b) Dynamic toy model showing a spheroidal body and cylindrical flagellum. The propulsion is aligned with the latter. The cylinder is exaggerated for clarity, and the associated helix is the same tapered one as Fig. 2.1

magnitude F_P pushes the flagellum along the unit vector $\mathbf{p}$ towards the body – this models the reaction force from the fluid to the propulsive force (thus the subscript P) exerted by the flagellum. The "front" end of the body is pinned at the origin while free to rotate. This situation models a cell whose body is stuck to a surface. We will consider this model in two special cases (1): $\mathbf{p} = -\mathbf{e}_x$, with the flagellum end constrained to lie on the x-axis, analogous to compression of a beam with a horizonal force load on both ends, and (2): $\mathbf{p} = -\mathbf{p}_f$, a closer representation of a swimmer generating propulsion aligned with the flagellar axis. Without loss of generality, motion is restricted to the $x - y$ plane, and torques are in the $\mathbf{n} = \mathbf{e}_z$ direction. The segment orientations are parametrized by angles θ_i, i.e. $\mathbf{p}_i = [\cos\theta_i \ \sin\theta_i]^{\mathrm{T}}$, and we define the bending angle $\theta = \theta_f - \theta_b$. From Eq. 2.1, the translational velocity of component i is

$$\dot{\mathbf{r}}_i = \mathbf{A}_i^{-1} \cdot \sum \mathbf{F}_i^{ext} = \mathbf{A}_i^{-1} \cdot \left[(\delta_{ib} - \delta_{if})\mathbf{X} + \delta_{ib}\mathbf{X}_C + \delta_{if}\mathbf{F}_P\right], \tag{2.5}$$

where $\mathbf{X}$ is a constraint force exerted by the hook to maintain connectivity, $\mathbf{X}_C$ is the external constraint fixing the body end position, and δ is the Kronecker delta. For slender rods, the translational resistance is $\mathbf{A}_i = \zeta^{\parallel}\mathbf{p}_i\mathbf{p}_i + \zeta^{\perp}(\boldsymbol{\delta} - \mathbf{p}_i\mathbf{p}_i)$ where $\boldsymbol{\delta}$ is the identity tensor and $\zeta^{\perp} = 4\pi\eta(l_i/2)/\ln(\epsilon^{-1})$ with $2\zeta^{\parallel} = \zeta^{\perp}$. The angular velocities are given by Eq. 2.2:

$$\dot{\theta}_i = \frac{1}{C_i^{\perp}}\sum T_i^{ext} = \frac{1}{C_i^{\perp}}\left[(\delta_{ib} - \delta_{if})\kappa(\theta - \theta_{eq}) + \frac{l_i}{2}\,(\mathbf{p_i} \times (\mathbf{X} - \delta_{ib}\mathbf{X}_C + \delta_{if}\mathbf{F}_P)) \cdot \mathbf{n}\right], \quad (2.6)$$

where $C_i^{\perp} = (2/3)\pi\eta l_i^3/\ln(\epsilon^{-1})$ is the rotational resistance normal to the plane. The first torque contribution is the linear contribution from the spring and the second includes all force moments. Two algebraic constraints arise: the first maintaining connectivity (Eq. 2.3) and the second fixing the body-end position:

$$\mathbf{0} = \mathbf{r}_b - \frac{l_b}{2}\mathbf{p}_b \quad (2.7)$$

Case 1 requires an additional constraint force $\mathbf{X}' = X'\mathbf{e}_y$ fixing the flagellum end to the x-axis so that

$$0 = r_{fy} + \frac{l_f}{2}p_{fy}. \quad (2.8)$$

Eqs. 2.3, 2.5, 2.6, 2.7, and 2.8 are non-dimensionalized with characteristic length $l_b/2$, force F_p, torque $F_P l_b$, time $\zeta^{\parallel}(l_b/2)/F_P$. We define the length ratio between flagellum and cell body as $L = l_f/l_b$. The following dimensionless differential-algebraic equation summarizes the dynamics of the static toy model (* denotes dimensionless variables):

$$\dot{\mathbf{r}}_i^* = \frac{1}{2(l_i/l_b)}(\boldsymbol{\delta} + \mathbf{p}_i\mathbf{p}_i) \cdot [(\delta_{ib} - \delta_{if})\mathbf{X}^* + \delta_{ib}\mathbf{X}_C^* + \delta_{if}\mathbf{p}] \quad (2.9a)$$

$$\dot{\theta}_i = \frac{3}{4(l_i/l_b)^3}\left[\mathrm{Fl}_F^{-1}(\theta - \theta_{eq}) + \frac{l_i}{l_b}\mathbf{p_i} \times (\mathbf{X}^* - \delta_{ib}\mathbf{X}_C^* + \delta_{if}\mathbf{p}) \cdot \mathbf{n}\right] \quad (2.9b)$$

$$\mathbf{0} = \mathbf{r}_b^* - \mathbf{r}_f^* + \mathbf{p}_b + L\mathbf{p}_f \quad (2.9c)$$

$$\mathbf{0} = \mathbf{r}_b^* - \mathbf{p}_b \quad (2.9d)$$

$$0 = r^*_{fy} + Lp_{fy} \tag{2.9e}$$

Here a dimensionless group $\mathrm{Fl}_F = F_P l_b/2\kappa$ arises, which will be called the flexibility number. The subscript F denotes a definition based on force. (Below we will define one based on motor torque.) It characterizes the ratio of the force moment to the bending resistance. We wish to examine the effects of Fl_F on the existence and stability of static equilibrium points arising from Eq. 2.9. Our choice of $\mathbf{p}$ affects the equilibria we find. In either case, a trivial solution to Eq. 2.9 maintaining static equilibrium is $\mathbf{r}_b = \mathbf{e}_x$, $\mathbf{r}_f = (2+L)\mathbf{e}_x$, $\theta_b = \theta_f = 0$, $\mathbf{X} = -\mathbf{X}^C = -\mathbf{e}_x$, $X' = 0$. Linearizing Eq. 2.9 around this point and seeking solutions of the form $\exp(\mu t)$, we find that μ satisfies:

$$\text{Case 1: } 10L^2\mathrm{Fl}_F\mu + 3L + 3 - 6L\mathrm{Fl}_F = 0, \tag{2.10}$$

$$\text{Case 2: } \mu\left(L^3\mathrm{Fl}_F(50+48L)\mu + 36L(1 - L\mathrm{Fl}_F + L) + 15(1 + L^3 - 2L^3\mathrm{Fl}_F)\right) = 0. \tag{2.11}$$

In Case 1, the sole eigenvector has different signs for (θ_b, θ_f), denoting a buckled conformation. This buckling eigenvalue changes sign when Fl_F exceeds a critical value

$$\mathrm{Fl}_{F,crit} = \frac{1}{2}\frac{L+1}{L} \tag{2.12}$$

Because of the up-down mirror symmetry of the system, $\mathrm{Fl}_{F,crit}$ indicates a pitchfork bifurcation. We note that while the loading chosen for Case 1 is chosen to mirror a classical buckling problem, it is not a representative picture of how a swimmer functions.

For the more realistic Case 2 there is always a zero eigenvalue that arises from rotational invariance – every angle the swimmer makes with the anchor point is equivalent. The remaining eigenvalue corresponds to the buckled state but is always negative, unlike Case 1. Thus a swimmer pinned by its "nose" does not buckle. To understand this result, we note that static equilibrium requires $\mathbf{X} = -\mathbf{p}_f$, and since the propulsive force is parallel to $\mathbf{p}_f$, this condition can only hold for $\theta = 0$. Any nonzero value of θ will result in a restoring spring

torque that will drive the system to the state $\theta = 0$. In other words, because the propulsive force is parallel to $\mathbf{p}_f$, it can generate no torque to counteract the restoring torque so any initially bent conformation will straighten over time. With the next model we address how this result changes when the cell can swim freely.

2.4 Dynamic toy model

2.4.1 Formulation

Consider the swimmer shown in Fig. 2.2(b), where the hinge constraining the body has been removed and the swimmer can move freely through the open domain. Here the non-hydrodynamic forces $\sum \mathbf{F}_i$ consist only of the constraint force $\mathbf{X}$ and the propulsive force, aligned with the flagellum in this case:

$$\sum \mathbf{F}_i^{ext} = (\delta_{ib} - \delta_{if})\mathbf{X} - \delta_{if} F_P \mathbf{p}_f. \tag{2.13}$$

The external torques are the same as the static toy model (Eq. 2.6), albeit with only $\mathbf{X}$ contributing a force moment. Any case where an aligned body and flagellum move with constant velocity $\mathbf{v}$ is a steady state solution to the equations of motion. To assess stability we need to be more definite than above about the geometry for the swimmer because the stability criterion now depends on the resistance coefficients. Specifically, we model the body as a prolate spheroid with major axis l_b and eccentricity e_b and the "flagellum" as a cylinder encompassing a helix of end-to-end length l_f, pitch λ, pitch angle ψ, radius $R_h = (\lambda/2\pi)\tan\psi$, and filament radius a_h, as originally illustrated in Fig. 2.1. In this model we will account for the thrust and drag associated with the helical flagellum without incorporating its rotation and the corresponding torque-thrust coupling. This effect is reserved for the full model.

The resistance tensors for the cell body can be decomposed into normal and tangential

components:

$$\mathbf{A}_b = 6\pi\eta(l_b/2)\left[X^A(e_b)\mathbf{p}_b\mathbf{p}_b + Y^A(e_b)(\boldsymbol{\delta} - \mathbf{p}_b\mathbf{p}_b)\right] \tag{2.14a}$$

$$\mathbf{C}_b = 8\pi\eta(l_b/2)^3\left[X^C(e_b)\mathbf{p}_b\mathbf{p}_b + Y^C(e_b)(\boldsymbol{\delta} - \mathbf{p}_b\mathbf{p}_b)\right] \tag{2.14b}$$

where η is the fluid viscosity and $\boldsymbol{\delta}$ is the identity tensor. The scalar resistance functions $X^A(e_i), X^C(e_i), Y^A(e_i)$ and $Y^C(e_i)$ are defined in ref. [59]. The flagellar resistance takes the same form as Eq. 2.14. We determine the normal and tangential drag coefficients of the cylinder from the full resistance tensors of the associated helix using resistive force theory [20, 115].

2.4.2 Calculation of resistance tenors

We assume the friction tensor per unit length, $\bar{\boldsymbol{\zeta}}$ at each point on the helix to be:

$$\bar{\boldsymbol{\zeta}} = \bar{\zeta}^{\parallel}\mathbf{tt} + \bar{\zeta}^{\perp}(\boldsymbol{\delta} - \mathbf{tt}), \tag{2.15}$$

where $\mathbf{t}$ is the local unit tangent vector, $\bar{\zeta}^{\parallel}$ and $\bar{\zeta}^{\perp}$ are the local tangential and normal drag coefficients per unit length for a slender rod, derived in [69] to account for the effects of helicity, and shown to work moderately well for typical flagella by [20, 89]:

$$\bar{\zeta}^{\parallel} = \frac{2\pi\eta}{\ln\frac{0.18\lambda}{a_h\cos\psi} - \frac{1}{2}}, \quad \bar{\zeta}^{\perp} = \frac{4\pi\eta}{\ln\frac{0.18\lambda}{a_h\cos\psi} + \frac{1}{2}} \tag{2.16}$$

Integrating in arclength s over the flagellum shape $\mathbf{r}(s)$ (neglecting the very minimal contribution of the taper to simplify the mathematics) gives the following resistance tensors with the integration limit $S = 2\pi l_f/\lambda$:

$$\mathbf{A}_f = \frac{\lambda}{2\pi\cos\psi}\left(\int_0^S \bar{\boldsymbol{\zeta}}(s)\,\mathrm{d}s\right), \tag{2.17a}$$

$$\mathbf{B}_f = -\frac{\lambda}{2\pi\cos\psi}\left(\int_0^S \bar{\boldsymbol{\zeta}}(s)\times\mathbf{r}(s)\,\mathrm{d}s\right), \tag{2.17b}$$

$$\mathbf{C}_f = -\frac{\lambda}{2\pi\cos\psi}\left(\int_0^S \mathbf{r}(s)\times\bar{\boldsymbol{\zeta}}(s)\times\mathbf{r}(s)\,\mathrm{d}s\right). \tag{2.17c}$$

The explicit forms of these tensors are reported below. Recall $S = l_f/b = 2\pi l_f/\lambda$, $\gamma = \bar{\zeta}^{\parallel}/\bar{\zeta}^{\perp}$. Equations for a left-handed helix oriented along x-axis. The helical amplitude does not decay to zero at the origin. The contributions to the resistance tensors are defined as follows:

$$\mathbf{A}_f = \left(\frac{\lambda}{2\pi}\right)\sec\psi\,\bar{\zeta}^{\perp}[S\boldsymbol{\delta} - (1-\gamma)\mathbf{D}] \tag{2.18}$$

$$\mathbf{B}_f = \left(\frac{\lambda}{2\pi}\right)^2\sec\psi\,\bar{\zeta}^{\perp}[\mathbf{R}_x - (1-\gamma)\mathbf{D}_r] \tag{2.19}$$

$$\mathbf{C}_f = \left(\frac{\lambda}{2\pi}\right)^3\sec\psi\,\bar{\zeta}^{\perp}[\mathbf{R}_{xx} - (1-\gamma)\mathbf{D}_{rr}] \tag{2.20}$$

$$\frac{\mathbf{D}}{\cos^2\psi} = \begin{bmatrix} S & -\tan\psi(1-\cos S) & -\tan\psi\sin S \\ -\tan\psi(1-\cos S) & \frac{1}{2}\tan^2\psi(S-\sin S\cos S) & \frac{1}{2}\tan^2\psi\sin^2 S \\ -\tan\psi\sin S & \frac{1}{2}\tan^2\psi\sin^2 S & \frac{1}{2}\tan^2\psi(S+\sin S\cos S) \end{bmatrix} \tag{2.21}$$

$$\mathbf{R}_x = \begin{bmatrix} 0 & -\tan\psi\cos S & \tan\psi\sin S \\ \tan\psi\cos S & 0 & -\frac{1}{2}S^2 \\ -\tan\psi\sin S & \frac{1}{2}S^2 & 0 \end{bmatrix} \tag{2.22}$$

$$\frac{\mathbf{D}_r}{\sin^2\psi} = \begin{bmatrix} S & \cot\psi(2-S\sin S-2\cos S) & \cot\psi(2\sin S - S\cos S) \\ \tan\psi\cos S & \frac{1}{8}(-4S+3\sin 2S-2S\cos 2S) & (2S^2-2S\sin 2S-3\cos 2S+3) \\ -\tan\psi\sin S & (2S^2+2S\sin 2S+3\cos 2S-3) & -\frac{1}{8}(4S+3\sin 2S-2S\cos 2S) \end{bmatrix} \tag{2.23}$$

$$\frac{\mathbf{R}_{xx}}{\tan^2\psi} = \begin{bmatrix} -S & \cot\psi(\cos S\sin S-1) & \cot\psi(S\cos S-\sin S) \\ \cot\psi(\cos S\sin S-1) & \frac{1}{12}(3\sin 2S-6S-4\cot^2\psi S^3) & -\frac{1}{2}\sin^2 S \\ \cot\psi(S\cos S-\sin S) & -\frac{1}{2}\sin^2 S & \frac{1}{12}(-3\sin 2S-6S-4\cot^2\psi S^3) \end{bmatrix} \tag{2.24}$$

$$
\begin{aligned}
\mathbf{D}_{rr}(1,1) &= -\sin^2\psi\tan^2\psi S \\
\mathbf{D}_{rr}(2,2) &= \sin^2\psi(\tfrac{5}{8}\sin 2S - \tfrac{S}{2} - \tfrac{S^3}{6} - \tfrac{3S}{4}\cos 2S - \tfrac{S^2}{4}\sin 2S) \\
\mathbf{D}_{rr}(3,3) &= \sin^2\psi(\tfrac{3S}{4}\cos 2S - \tfrac{S}{2} - \tfrac{S^3}{6} - \tfrac{5}{8}\sin 2S + \tfrac{S^2}{4}\sin 2S) \\
\mathbf{D}_{rr}(1,2) &= \mathbf{D}_{rr}(2,1) = \sin^2\psi\tan\psi(S\sin S - 4\sin^2\tfrac{1}{2}S) \\
\mathbf{D}_{rr}(1,3) &= \mathbf{D}_{rr}(3,1) = -\sin^2\psi\tan\psi(2\sin S - S\cos S) \\
\mathbf{D}_{rr}(2,3) &= \mathbf{D}_{rr}(3,2) = \tfrac{1}{8}\sin^2\psi(5\cos 2S - 5 + 6S\sin 2S - 2S^2\cos 2S)
\end{aligned}
\tag{2.25}
$$

For general use, the tensors are non-dimensionalized and reoriented along $\mathbf{p}_f$ via rotation matrices $\mathbf{R}_b$ and $\mathbf{R}_f$:

$$\mathbf{A}_f^* = \frac{1}{\zeta_b}\mathbf{R}_b^{\mathrm{T}}\cdot\mathbf{R}_f^{\mathrm{T}}\cdot\mathbf{A}_f\cdot\mathbf{R}_f\cdot\mathbf{R}_b \tag{2.26}$$

$$\mathbf{B}_f^* = \frac{1}{(l_b/2)\zeta_b}\mathbf{R}_b^{\mathrm{T}}\cdot\mathbf{R}_f^{\mathrm{T}}\cdot\mathbf{B}_f\cdot\mathbf{R}_f\cdot\mathbf{R}_b \tag{2.27}$$

$$\mathbf{C}_f^* = \frac{1}{(l_b/2)^2\zeta_b}\mathbf{R}_b^{\mathrm{T}}\cdot\mathbf{R}_f^{\mathrm{T}}\cdot\mathbf{C}_f\cdot\mathbf{R}_f\cdot\mathbf{R}_b \tag{2.28}$$

Two distinguishing features of the resistance tensors are the coupling of motion in all directions and the lack of degeneracy in directions normal to the helical axis. However, in the present dynamic toy model the “flagellum" does not rotate but only exerts a thrust to which the fluid exerts a reaction force, so $\mathbf{B}$ does not enter the formulation. The chosen symmetry also removes coupling between x-y-z directions, and we average the normal drags to obtain the following scalar resistances for the overall cylinder:

$$A_f^{\parallel} = \bar{\zeta}^{\perp} l_f \sec\psi\left[1 - \frac{1}{2}(1-\gamma)\sin^2\psi\right], \quad A_f^{\perp} = \bar{\zeta}^{\perp} l_f \frac{\sin^2\psi}{\cos\psi}(1+\gamma\cot^2\psi), \tag{2.29a}$$

$$
\begin{aligned}
C_f^{\parallel} &= \max\left[\bar{\zeta}^{\perp} l_f \left(\tfrac{\lambda}{2\pi}\right)^2 \tfrac{\sin^2\psi}{\cos\psi}(1+\gamma\cot^2\psi),\ \ 4\pi\eta a_h^2 \tfrac{l_f}{\cos\psi}\right], \\
C_f^{\perp} &= \bar{\zeta}^{\perp} l_f \tfrac{\sin^2\psi}{\cos\psi}\left[\tfrac{1}{2}\left(\tfrac{\lambda}{2\pi}\right)^2(\sec^2\psi+\gamma) + \tfrac{2}{3}l_f^2(2\csc^2\psi - (1-\gamma))\right],
\end{aligned}
\tag{2.29b}
$$

where $\gamma = \bar{\zeta}^{\parallel}/\bar{\zeta}^{\perp}$. The condition imposed on $C_f^{\parallel}$ ensures a finite resistance for the straight-

ened flagellum.

2.4.3 Stability analysis for dynamic toy model

Because the stability of straight swimming is independent of the swimmer's orientation with respect to the laboratory frame, we can write a single evolution equation for θ by differentiating Eq. 2.4:

$$\frac{d\theta}{dt} = (\boldsymbol{\omega}_f - \boldsymbol{\omega}_b) \cdot \mathbf{n}. \tag{2.30}$$

Writing $\boldsymbol{\omega}_i$ explicitly in terms of external forces and torques yields

$$\boldsymbol{\omega}_i = \mathbf{C}_i^{-1} \cdot \sum \mathbf{T}_i = \mathbf{C}_i^{-1} \cdot \left[(\delta_{ib} - \delta_{if})\kappa(\theta - \theta_{eq})\mathbf{n} + \frac{l_i}{2} \left(\mathbf{p_i} \times \mathbf{X}\right) \right], \tag{2.31}$$

We use the same force and length scaling as the static toy model with characteristic time $\zeta_b(l_b/2)/F_P$ based on the body drag coefficient $\zeta_b = 6\pi\eta(l_b/2)$. In dimensionless form, Eq. 2.30 becomes

$$\frac{d\theta}{dt^*} = -\frac{1}{\mathrm{Fl}_F} \left(c_b^{*\perp} + c_f^{*\perp}\right)(\theta - \theta_{eq}) + \left(\left(Lc_f^{*\perp}\mathbf{p}_f - c_b^{*\perp}\mathbf{p}_b\right) \times \mathbf{X}^*\right) \cdot \mathbf{n}, \tag{2.32}$$

where $c_i^{*\perp} = [C_i^{\perp}/(\zeta_b(l_b/2)^2)]^{-1}$ is the scaled rotational mobility. Eqn. (2.32) consists of two pieces: the first term is the collective spring torque always opposing the growth of θ, and the second term summarizes bending moment contributions. To explicitly express the bending moments as functions of θ owing to invariance, we need a suitable expression for the constraint force $\mathbf{X}^*$ which we may obtain from differentiating the algebraic constraint equation 2.3:

$$\mathbf{0} = \dot{\mathbf{r}}_b - \dot{\mathbf{r}}_f + \dot{\mathbf{p}}_b + L\dot{\mathbf{p}}_f \tag{2.33}$$

Recognizing $\dot{\mathbf{p}}_i = \boldsymbol{\omega}_i \times \mathbf{p}_i$ and using Eqs. (2.1) and (2.2) for the velocity terms:

$$\mathbf{X}^* = \mathbf{M}^{-1} \cdot \left[-a_f^{\parallel *}\mathbf{p}_f - c_b^{\perp *} Y(\mathbf{n} \times \mathbf{p}_b) + Lc_f^{\perp *} Y(\mathbf{n} \times \mathbf{p}_f) \right] \tag{2.34}$$

$$\mathbf{M} = m_1\boldsymbol{\delta} + m_2\mathbf{p}_b\mathbf{p}_b + m_3\mathbf{p}_f\mathbf{p}_f \tag{2.35}$$

$$\begin{aligned} m_1 &= a_b^{\perp *} + a_f^{\perp *} + c_b^{\perp *} + c_f^{\perp *}L^2, \quad m_2 = -c_b^{\perp *} + a_b^{\parallel *} - a_b^{\perp *} \\ m_3 &= -c_f^{\perp}L^2 + \tfrac{1}{L}\left(a_b^{\parallel *} - a_b^{\perp *}\right). \end{aligned} \tag{2.36}$$

As in Section 2.4, a_i^* is the scaled mobility, either tangential or normal. The inverse of $\mathbf{M}$ is conveniently written as:

$$\mathbf{M}^{-1} = m_1'\boldsymbol{\delta} - m_2'\mathbf{p}_b\mathbf{p}_b - m_3'\mathbf{p}_f\mathbf{p}_f + m_4'\mathbf{nn} \tag{2.37}$$

$$\begin{aligned} m_1' &= \det(\mathbf{M})^{-1}m_1(m_1+m_2+m_3), & m_2' &= \det(\mathbf{M})^{-1}m_1m_2 \\ m_3' &= \det(\mathbf{M})^{-1}m_1m_3, & m_4' &= \det(\mathbf{M})^{-1}m_2m_3\sin^2\theta \\ \det(\mathbf{M}) &= m_1^3 + m_2m_1^2 + m_3m_1^2 + m_1m_2m_3\sin^2\theta \end{aligned} \tag{2.38}$$

Substituting into Eq. 2.34 and recognizing $\cos\theta = \mathbf{p_b}\cdot\mathbf{p_f}$ and $\mathbf{p_b}\times\mathbf{p_f} = \sin\theta\mathbf{n}$:

$$\begin{aligned} \mathbf{X}^* = m_1'&\left[-a_f^{\parallel *}\mathbf{p}_f - c_b^{\perp *}Y(\mathbf{n}\times\mathbf{p}_b) + c_f^{\perp *}Y(\mathbf{n}\times\mathbf{p}_f)\right] \\ &+m_2'\left[a_f^{\parallel *}\mathbf{p}_b\cos\theta + c_f^{\perp *}Y\sin\theta\mathbf{p}_b\right] + m_3'\left[a_f^{\parallel *}\mathbf{p}_f + c_b^{\perp *}Y\sin\theta\mathbf{p}_f\right]. \end{aligned} \tag{2.39}$$

And finally, substituting for the θ derivative yields:

$$\begin{aligned} \frac{d\theta}{dt^*} = &-(c_f^{\perp *} + c_b^{\perp *})Y + c_f^{\perp *}L\left[m_1'(-c_b^{\perp *}Y\cos\theta + c_fL^{\perp *}Y) - m_2'\left(a_f^{\parallel *}\frac{\sin 2\theta}{2} + c_f^{\perp *}LY\sin^2\theta\right)\right] \\ &-c_b^{\perp *}\left[(m_3' - m_1')a_f^{\parallel *}\sin\theta + c_b^{\perp *}Y(m_3'\sin^2\theta - m_1') + m_1'c_f^{\perp *}LY\cos\theta\right], \end{aligned} \tag{2.40}$$

Linearizing at $\theta = 0$:

$$\begin{aligned} \frac{d\theta}{dt^*} = \Bigg[&\frac{-(c_f^{\perp *} + c_b^{\perp *})}{f} + c_f^{\perp *}L\left(\frac{m_1^0}{f}(-c_b^{\perp *} + c_f^{\perp *}L) - m_2^0a_f^{\parallel *}\right) \\ &-c_b^{\perp *}\left((m_3^0 - m_1^0)a_f^{\parallel *} + \frac{m_1^0(c_f^{\perp *}L - c_b^{\perp *})}{f}\right)\Bigg]\theta \end{aligned} \tag{2.41}$$

where $m_i^0 = m_i'|_{\theta=0}$. The eigenvalue in Eq. 2.41 is a function of Fl, L, e_b, and the helical geometry. The critical flexibility is then solved for in Eq. 2.42.

As above, we perform linear stability analysis of the straight-swimming state $\theta = 0$, finding an the expression for $\mathrm{Fl}_{F,crit}$ in terms of the aspect ratio L and the individual body and flagellar geometries:

$$\mathrm{Fl}_{F,crit} = A_f^{\parallel *} \frac{-(c_f^{\perp *} + c_b^{\perp *}) + (c_f^{\perp *} L m_1^0 - c_b^{\perp *} m_1^0)(-c_b^{\perp *} + c_f^{\perp *} L)}{c_f^{\perp *} L m_2^0 + c_b^{\perp *}(m_3^0 - m_1^0)} \tag{2.42}$$

Here $A_i^{\parallel *} = A_i^{\parallel}/\zeta_b$ is the scaled translational resistance. Eq. 2.42 displays a minimum in L and does not plateau as $L \to \infty$ (provided e_b and helical parameters are constant). The existence of any extrema in the bifurcation parameter may already have implications about the range of length ratios found in natural organisms. In addition, the stability criterion depends on the hydrodynamic resistances – this is an expected result because these enter into the force balance for the straight-swimming state. However, this model still oversimplifies the thrust mechanism; below we will return to the full model in which motor torque rotates the flagellum and the rotation-force coupling in the flagellar hydrodynamic resistance leads to the propulsive force.

2.5 Full model

Equations of motion

Having established the bending behavior of the dynamic toy model, we keep the spheroidal body and replace the cylindrical flagellum in the dynamic toy model with a helix of length l_f, pitch λ, pitch angle ψ, radius $R_h = \lambda/2\pi \tan\psi$ and filament radius a_h, to replicate a real flagellum, as originally shown in Fig. 2.1 The taper on the flagellum reflects the observation of Vogel *et al.* [115] that a flagellum with finite elasticity will self-taper at the hook as it rotates, and makes for convenient analysis later. (Though we also note from [96], dynamics of swimmers with tapered and plain helices are not fundamentally different). An applied motor torque $\mathbf{T}_m$ with constant magnitude T rotates the flagellum, generating the propulsive force

on the swimmer via the rotation-force coupling tensor $\mathbf{B}$. The dynamics are thus governed by Eqs. 2.1 and 2.2 in their full form, with

$$\sum \mathbf{F}_i^{ext} = (\delta_{ib} - \delta_{if})\mathbf{X}, \tag{2.43}$$

$$\sum \mathbf{T}_i^{ext} = (\delta_{i,b} - \delta_{i,f})\left[-\mathbf{T}_p + \kappa(\theta - \theta_{eq})\mathbf{n}\right] + \frac{l_i}{2}\,\mathbf{p_i} \times \mathbf{X}, \tag{2.44}$$

and include the coupling in the resistance tensors as determined by Eq. 2.17. The algebraic constraint, Eq. 2.3 still remains. The quantity $\mathbf{T}_p$ is the torque applied to the flagellum by the motor via the hook. If $\theta = 0$, then $\mathbf{T}_p = \mathbf{T}_m$; otherwise they are not identical but are related via the torque balance on the hook as we describe below. We scale torques by T, forces by $T/(l_b/2)$, time by $\zeta_b(l_b/2)^2/T$, $\mathbf{A}$ by ζ_b. $\mathbf{B}$ by $\zeta_b(l_b/2)$ and $\mathbf{C}$ by $\zeta_b(l_b/2)^2$. Here the flexibility number $\mathrm{Fl}_T = T/\kappa$ characterizes the ratio between the motor torque and bending modulus, with the subscript T denoting a definition based on applied torque (not force moment). The rotation of body and flagellum requires us to track the rotational phase of each via Euler parameters $\mathbf{q}_i$. Details of their definition and evolution are found in Appendix 2.6, along with methods used to solve the full equations of motion.

Torque propagation through the hook

Though we do not model the motor or hook explicitly, we must resolve their effect on our model dynamics via $\mathbf{T}_p$. Fig. 2.3(a) offers a more detailed look at the body-flagellum connection. The connection occurs over three parts: an embedded motor in the cell body, flexible hook, and flagellum. The motor consists of a static stator and spinning rotor; the stator is stationary relative to the body and anchors the freely-spinning rotor, keeping it aligned with $\mathbf{p}_b$. During swimming, the stator applies a constant-magnitude torque $\mathbf{T}_m^{(b)} = T\mathbf{p}_b$ on the rotor, hereby referred to as the motor torque, and a counter-torque $-\mathbf{T}_m^{(b)}$ is exerted on the stator. The hook passes the torque from motor to flagellum, resulting in the torque $\mathbf{T}_m^{(f)} = T\mathbf{p}_f$ on the flagellum if $\mathbf{p}_b = \mathbf{p}_f$ ($\theta = 0$). The sign convention chosen here

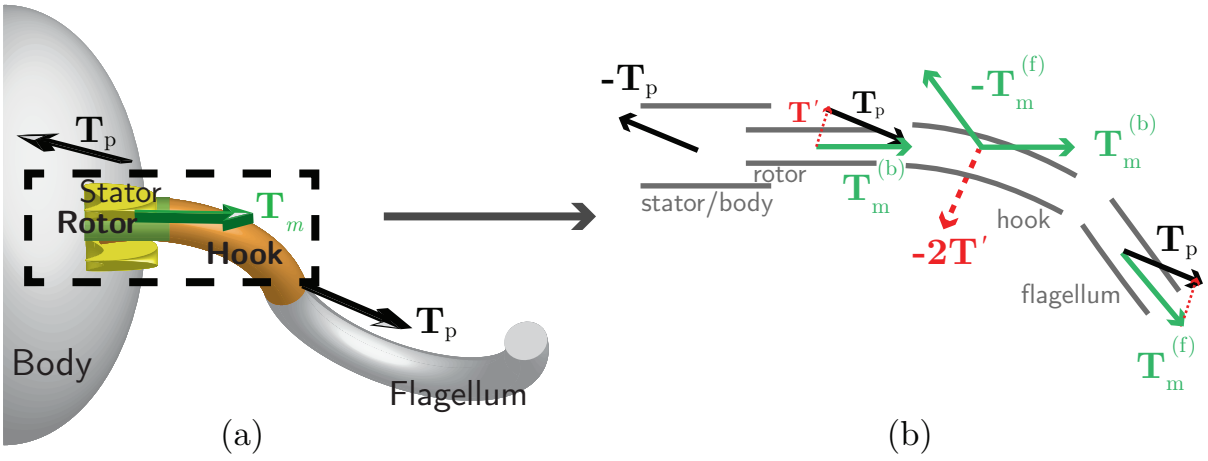

Figure 2.3: (a) Close-up of cell at the body-flagellum connection showing stator, rotor, and hook. Green arrow shows the applied motor torque, and black arrows show $\mathbf{T}_p$. The narrow tapering of the flagellum aligns hook axis with flagellar helical axis $\mathbf{p}_f$ at connection point. (b) Non-hydrodynamic torques **in the bending plane** resulting from motor torque propagation. Components (left to right): stator/body, rotor, hook, flagellum. Arrows: solid green is the motor torque propagation, dotted red is the required offset for a torque-free hook, solid black are the torques explicitly used in equations of motion. Constraint torques and hydrodynamic torques are omitted for clarity.

produces a pusher for a left-handed flagellum, e.g. *V. alginolyticus*. The case $\theta \neq 0$, in which case the motor torque is propagated along a curved hook, requires further attention as we now describe.

Fig. 2.3(b) shows the torques in the bending plane resulting from propagation of $\mathbf{T}_m^{(b)}$ from rotor to flagellum in a bent swimmer. The green arrows show the torque propagation that would occur if the torque simply followed the local axis of the hook and then on to the flagellum. However, since the motor and flagellum are not aligned, the rotor and flagellum must exert an additional torque on the hook so that the sum of torques on the hook is zero. We will denote this additional torque as $-2\mathbf{T}'$ so that the torque balance on the hook becomes

$$T\left(\mathbf{p}_b - \mathbf{p}_f\right) - 2\mathbf{T}' = 0. \tag{2.45}$$

Thus

$$\mathbf{T}' = \frac{1}{2} T\left(\mathbf{p}_b - \mathbf{p}_f\right). \tag{2.46}$$

We call $\mathbf{T}'$ the offset torque. The hook must then exert a total torque of $2\mathbf{T}'$ back onto the

rotor and flagellum, which we assume from symmetry will be equally distributed between the two as shown with dotted red arrows in Fig.2.3(b). Thus the total torque exerted on the flagellum (black arrow) is given by

$$\mathbf{T}_p = T\mathbf{p}_f + \mathbf{T}' = \frac{1}{2}T(\mathbf{p}_b + \mathbf{p}_f). \tag{2.47}$$

As a check, we calculate the same sum for the stator/body. Note that any nonzero $\mathbf{T}'$ acting on the rotor attempts to reorient it from $\mathbf{p}_b$, so the stator must exert a constraint torque equal to $-\mathbf{T}'$ on the rotor to prevent relative motion between the two. Summing torques exerted by stator on rotor gives $\mathbf{T}_m^{(b)} - \mathbf{T}' = \mathbf{T}_p$. Torque balance requires the torque on stator/body to be exactly $-\mathbf{T}^p$ (black arrow). Indeed, we end up (as we must) with equal and opposite torques acting on body and flagellum from the propagation; these will be balanced by hydrodynamic drag. To see how effective the propagation is, we look at the magnitude of $\mathbf{T}^p$:

$$|\mathbf{T}^p| = T\sqrt{\frac{1+\cos\theta}{2}} \tag{2.48}$$

We clearly see that the full torque T is passed on when the swimmer is straight ($\theta = 0$), but the total propagated torque is less than T when $\theta \neq 0$. Having completely established the equations of motion for the full model, we now proceed to characterizing swimming behavior.

2.6 Numerics

The swimmer is treated as a system of two linked rigid segments. Affixed to each segment is a local coordinate system as demonstrated by Ross *et al.* [91]. Any orientation in $\mathbb{R}^3$ can be represented by 4 Euler parameters, q_{ij} [$i = b, f$ and $j = 0, 1, 2, 3$], that explicitly define

transformations between different frames of reference, via rotation matrices $\mathbf{R_i}$,

$$\mathbf{R}_i = 2\begin{bmatrix} q_{i0}^2 + q_{i1}^2 - \frac{1}{2} & q_{i1}q_{i2} + q_{i0}q_{i3} & q_{i1}q_{i3} - q_{i0}q_{i2} \\ q_{i1}q_{i2} - q_{i0}q_{i3} & q_{i0}^2 + q_{i2}^2 - \frac{1}{2} & q_{i2}q_{i3} + q_{i0}q_{i1} \\ q_{i1}q_{i3} + q_{i0}q_{i2} & q_{i2}q_{i3} - q_{i0}q_{i1} & q_{i0}^2 + q_{i3}^2 - \frac{1}{2} \end{bmatrix} \tag{2.49}$$

Refering to Figure 2.1, in the body (i) fixed frames, orientation vectors are given by $\mathbf{e}_x^{(i)}$. Transformations mapping body-fixed frames to the laboratory frame are given by the following:

$$\mathbf{p}_b = \mathbf{R}_b^{\mathrm{T}}\mathbf{e}_x^{(b)},\ \mathbf{p}_f = \mathbf{R}_b^{\mathrm{T}}\mathbf{R}_f^{\mathrm{T}}\mathbf{e}_x^{(b)}. \tag{2.50}$$

The time evolution of Euler parameters is determined by relative angular velocities, $\boldsymbol{\omega_i^*}$, where the * superscript refers to $\boldsymbol{\omega_i}$ in the body (i) frame.[91]:

$$\begin{bmatrix} \dot{q}_{i0} \\ \dot{q}_{i1} \\ \dot{q}_{i2} \\ \dot{q}_{i3} \end{bmatrix} = \frac{1}{2}\begin{bmatrix} 0 & -\omega_{ix}^* & -\omega_{iy}^* & -\omega_{iz}^* \\ \omega_{ix}^* & 0 & \omega_{iz}^* & -\omega_{iy}^* \\ \omega_{iy}^* & -\omega_{iz}^* & 0 & \omega_{ix}^* \\ \omega_{iz}^* & \omega_{iy}^* & -\omega_{ix}^* & 0 \end{bmatrix}\begin{bmatrix} q_{i0} \\ q_{i1} \\ q_{i2} \\ q_{i3} \end{bmatrix} \tag{2.51}$$

$$\boldsymbol{\omega}_b^* = \mathbf{R}_b\boldsymbol{\omega}_b,\ \boldsymbol{\omega}_f^* = \mathbf{R}_f\mathbf{R}_b(\boldsymbol{\omega}_f - \boldsymbol{\omega}_b) \tag{2.52}$$

The angular velocities in the laboratory frame are determined by torque balances on each segment, Eq. 2.2. Writing Eq. 2.51 for b and Fl together with Eqs. (2.1) and (2.3) yields a system of differential algebraic equations (DAE) completely characterizing the swimmer's motion. The dependent variables are $\mathbf{r}_i$, $\mathbf{q}_i$, and $\mathbf{X}$ (17 unknowns counting for dimensions).

The dynamic equations may be expressed in a condensed form

$$\begin{cases} \dot{\mathbf{y}} = \mathbf{F}(\mathbf{y};\mathbf{X}) \\ \mathbf{0} = \mathbf{G}(\mathbf{y}) \end{cases}, \quad \mathbf{y} = \begin{bmatrix} \mathbf{r}_b \\ \mathbf{r}_f \\ \mathbf{q}_b \\ \mathbf{q}_f \end{bmatrix} \tag{2.53}$$

where $\mathbf{F}$ and $\mathbf{G}$ are vector-valued functions, $\mathbf{y}$ is a differential variable, and $\mathbf{X}$ is an algebraic variable (constraint force). Eq. (2.53) is a differential-algebraic equation (DAE) in autonomous, semi-explicit Hessenberg form with index 2, with the exact solution algorithms being rather cumbersome [54]. Instead, an approximate linear solution following Schmid *et al.* is adopted [94]. At each time step, differential variables $\mathbf{y}_n$ are first advanced to $\mathbf{y}_{n+1}$ using the current constraint force $\mathbf{X}_n$, and the updated $\mathbf{y}_{n+1}$ is used to calculate $\mathbf{X}_{n+1}$. The latter step requires an algebraic equation for $\mathbf{X}$ that we showed above in Eq. 2.34. We then obtain an algebraic equation of the form $\mathbf{0} = \mathbf{G}'(\mathbf{y};\mathbf{X})$. For this particular case, $\mathbf{G}'$ is linear in $\mathbf{X}$:

$$\begin{aligned} \mathbf{G}'(\mathbf{y};\mathbf{X}) = \Big[& \mathbf{A}_b^{*-1} + \mathbf{A}_f^{*-1} - \mathbf{p}_b \times \mathbf{C}_b^{*-1} \times \mathbf{p}_b + \left(\mathbf{A}_f^{*-1} \cdot \mathbf{B}_f^{*} - L\mathbf{p}_f \times \right) \\ & \times \left(\mathbf{C}_f^{*} - \mathbf{B}_f^{*\mathrm{T}} \cdot \mathbf{A}_f^{*-1} \cdot \mathbf{B}_f^{*} \right)^{-1} \cdot \left(\mathbf{B}_f^{*\mathrm{T}} \cdot \mathbf{A}_f^{*-1} + L\mathbf{p}_f \times \right) \Big] \cdot \mathbf{X} - \\ & \left(-\mathbf{A}_f^{*-1} \cdot \mathbf{B}_f^{*} + L\mathbf{p}_f \times \right) \cdot \left(\mathbf{C}_f^{*} - \mathbf{B}_f^{*\mathrm{T}} \cdot \mathbf{A}_f^{*-1} \cdot \mathbf{B}_f^{*} \right)^{-1} \cdot (\mathbf{p}_f - \mathbf{Y}) \\ & \mathbf{p}_b \times \mathbf{C}_b^{*-1} \cdot (-\mathbf{p}_b + \mathbf{Y}) \end{aligned} \tag{2.54}$$

As long as the initial guess $\mathbf{y}_0$ satisfies the original algebraic constraint (2.3), then solutions to Eq. (2.54) should be in the neighborhood of the true solution for a small enough step size h. For the first time step t_1, $\mathbf{y}_0$ is advanced using the forward Euler method. The remaining steps $\mathbf{y}_n$ are advanced using the linear two-step Adams-Bashforth method. Since the matrices in Eq. (2.54) are only 3×3, $\mathbf{X}$ can be solved very quickly.

Simulations are run with the swimmer initially aligned in the x direction with $\theta = 0$, and

h is determined from the period of flagellar rotation, T_f. We approximate T_f by calculating the axial rotation ω_f of a standalone helix subject to a torque of magnitude T along its axis, and scaled with our dimensionless time $\zeta_b(l_b/2)^2/T$:

$$T_f = 2\pi \frac{\zeta^{\perp}}{\zeta_b} \lambda^{*2} \sin\psi \tan\psi \left(1 + \gamma \tan^2\psi\right), \tag{2.55}$$

For accurate time resolution, we select $h = T_f/100$.

2.7 Results and discussion

2.7.1 Trajectory characterization of full model

To begin presentation of results for the full model, we show results for two different values of Fl_T. We assign a fixed geometry to the swimmer based on a real bacterium, *V. alginolyticus* with $L = 1$, $e_b = 0.93$, $\lambda^* = \lambda/2\pi(l_b/2) = 0.16$, $\psi = 0.65$ [98] and initialize the configuration with $\theta = 0$ and $\theta_{eq} = 0$. We denote the time-averaged orientation angle as $\bar{\theta}$. Under an applied motor torque, the flagellum rotates counter-clockwise (when viewed from behind), generating the propulsive force that pushes the cell body forward and inducing clockwise counter-rotation in the body. The equations of motion are solved via the method outline in Appendix 2.6. Simulations with small and large Fl_T ($\mathrm{Fl}_T = 1$ and 2, respectively) are shown in Fig. 2.4.

“Straight" swimming

For $\mathrm{Fl}_T = 1$, the cell moves in what appears superficially to be a perfectly straight trajectory as shown in Fig. 2.4(a). Actually, however there is a very slight “wobble", as the steady state value of θ is not exactly zero due to the chirality of the flagellum – the trajectory is actually a helix of very small radius and pitch. The time evolution of θ is shown in Fig. 2.4(c). At $\mathrm{Fl}_T = 1$, the steady-state value of $\theta = 0.012$. With a motor torque of 1000 pN

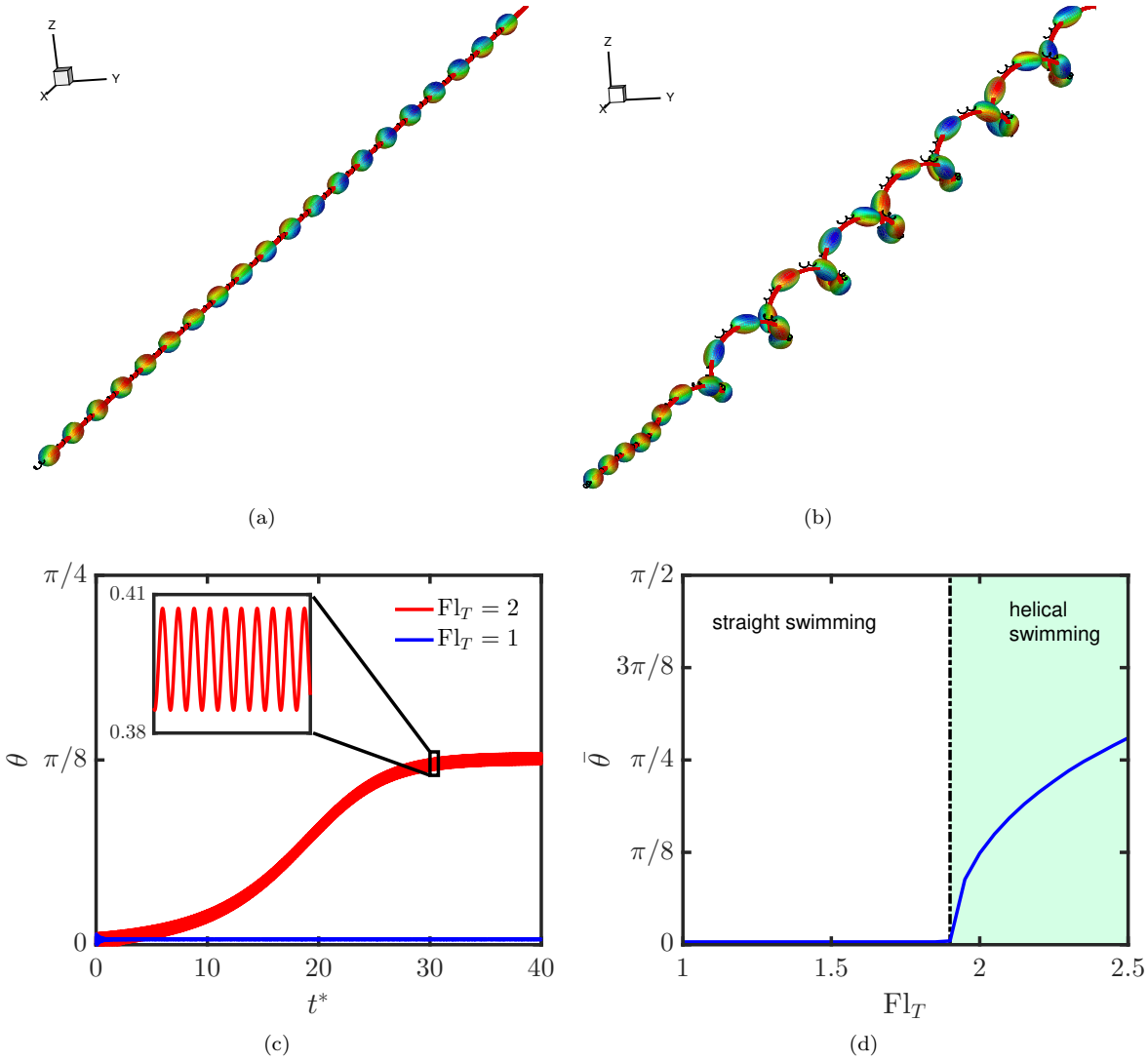

Figure 2.4: (a) Snapshot of straight trajectory with $\mathrm{Fl}_T = 1$. (b) Snapshot of helical trajectory with $\mathrm{Fl}_T = 2$. Solid lines trace the path of the cell body center of mass. (c) Time evolution of the instantaneous bending angle θ. The zoom inset shows oscillations in θ above $\mathrm{Fl}_{T,crit}$ (d) Bifurcation of average bending angle $\bar{\theta}$ in Fl.

nm, $|\mathbf{v}_b| \sim 60$ μm/s, $|\boldsymbol{\omega}_f| \sim 100$ Hz in real units, comparable to experimental results [98]. As Fl_T is increased from 1, $\bar{\theta}$ does not change as shown in Fig. 2.4(d), meaning trajectories still remain straight. However, at a value of $\mathrm{Fl}_T = 1.65$, $\bar{\theta}$ starts to deviates sharply from the straight swimming angle, and we define this point as the bifurcation value, $\mathrm{Fl}_{T,crit}$. Beyond this flexibility, a new trajectory regime arises.

Helical swimming

When Fl_T exceeds $\mathrm{Fl}_{T,crit}$ (flexible hook), we see helical trajectories as shown in Fig. 2.4(b) for $\mathrm{Fl}_T = 2$. In Fig. 2.4(c), we see that $\bar{\theta}$ plateaus to a nonzero value, and the instantaneous angle $\theta(t^*)$ oscillates around $\bar{\theta}$, which is $\sim 22^\circ$ in this case. Because the flexible hook does not generate a sufficient restoring torque (as in the dynamic toy model), the alignment will not stabilize near θ_{eq} and the swimmer's conformation will not remain straight. A helical path with radius R_t and pitch λ_t of $O(l_b)$ is created as a net result of the flagellar thrust pushing the body off-axis (circular motion in bending plane) and the coupling in the flagellar resistances (translation normal to bending plane). Once Fl_T exceeds $\mathrm{Fl}_{T,crit}$, the net speed decreases because of the helical path and also because the propagation of motor torque into flagellar rotation decreases as indicated by Eq. 2.48. Specifically, as seen in Fig. 2.4(d), $\bar{\theta}$ increases monotonically with Fl past $\mathrm{Fl}_{T,crit}$, meaning $|\mathbf{T}_p|$ drops monotonically with Fl_T. At sufficently large Fl_T, we observe $\bar{\theta} > \pi/2$, a conformation unlikely to be observed in real swimmers that arises here because our model neglects excluded volume interactions between body and flagellum.

2.7.2 Dynamics of the straight-helical transition

$\bar{\theta}$ is a useful metric of stability, but the chirality of the helical flagellum limits the direct analysis of either θ or $\bar{\theta}$ in the full model. But following the dynamic toy model, we recognize that θ is invariant in any frame of reference. To more closely examine the dynamics, we

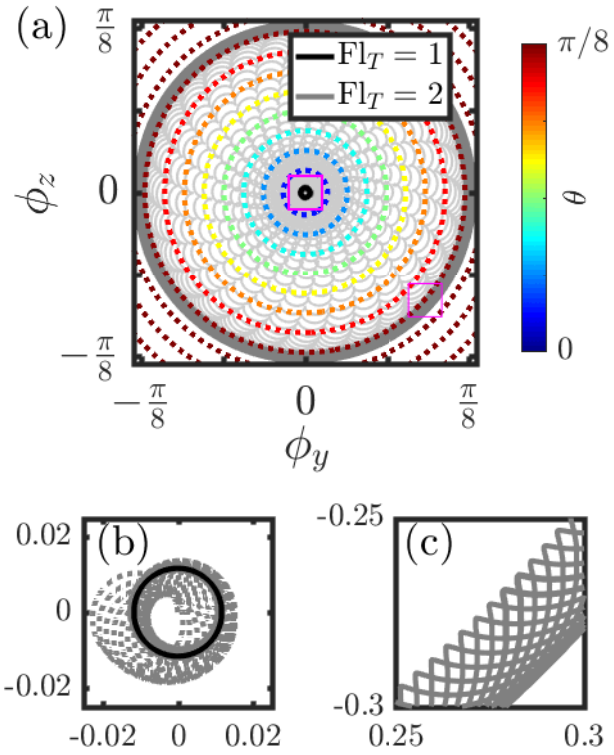

Figure 2.5: (a) Phase space of bending-related Euler angles ϕ_y and ϕ_z. Thick lines are limiting trajectories for $\mathrm{Fl}_T = 1$ and $\mathrm{Fl}_T = 2$, thin gray line shows full transient trajectory for $\mathrm{Fl}_T = 2$, and colored dotted contours show level curves of θ. (b) Close-up of center region showing the stable limit cycle of straight-swimming at $\mathrm{Fl}_T = 1$ in black, with the dotted gray line showing the trajectory evolution from (0,0). (c) Close-up of limiting trajectory showing quasiperiodicity at $\mathrm{Fl}_T = 2$ helical swimming.

rewrite the equations of motion in the body (b) frame of reference with

$$\mathbf{p}_b^{(b)} = \mathbf{e}_x,\ \mathbf{p}_f^{(b)} = \mathbf{R}_f^{\mathrm{T}} \cdot \mathbf{e}_x,\ \mathbf{X}^{(b)} = \mathbf{R}_b \cdot \mathbf{X}. \tag{2.56}$$

As long as the rotation matrix $\mathbf{R}_f^{\mathrm{T}}$ is known, the configuration evolution is completely known. A useful way to visualize it is via the Euler angles $\boldsymbol{\phi} = [\phi_x\ \phi_y\ \phi_z]^{\mathrm{T}}$ describing the extrinsic rotations taken around the $x-y-z$ axes of the (b) frame to obtain the (f) frame. Using $\boldsymbol{\phi}$ we can write

$$\mathbf{p}_f^{(b)} = \begin{bmatrix} \cos\phi_y\ \cos\phi_z \\ \cos\phi_y\ \sin\phi_z \\ -\sin\phi_y \end{bmatrix},\quad \cos\theta = \cos\phi_y\ \cos\phi_z. \tag{2.57}$$

The angle ϕ_x tracks the rotational phase difference between body and flagella, but does not affect bending, which is solely determined by ϕ_y and ϕ_z. Fig. 2.5 shows the evolution of these quantities for the results presented above at $\mathrm{Fl}_T = 1$ and $\mathrm{Fl}_T = 2$. Even for small Fl_T, there is no perfectly straight configuration $\theta = 0$, as Euler angle trajectory evolves to a limit cycle on a contour of small but finite θ, illustrated by the finite radius in the inset of Fig. 2.5. Above the bifurcation value $\mathrm{Fl}_{T,crit}$, the trajectory becomes quasi-periodic (Fig. 2.5, second inset), and θ oscillates around some average value $\bar{\theta}$ much further from zero. The trajectory radius also grows monotonically with Fl_T in this regime. Thus the bifurcation value $\mathrm{Fl}_{T,crit}$ marks the transition from periodicity to quasi-periodicity in Euler angles.

2.7.3 Parameter study of full model and model comparison

Having established the relation between Fl_T and $\bar{\theta}$ for the base geometry described in section 2.7A and the stability criterion defining $\mathrm{Fl}_{T,crit}$, we examine how these change with the 4 main geometric parameters of the full model: L, ψ, λ^*, and e_b. We run simulations altering a single parameter in the base geometry and tracking $\bar{\theta}$ as Fl_T increases. We approximate the stability boundary by marking sharp changes in $\bar{\theta}$ with Fl_T, similar to Figure 2.4(d), and

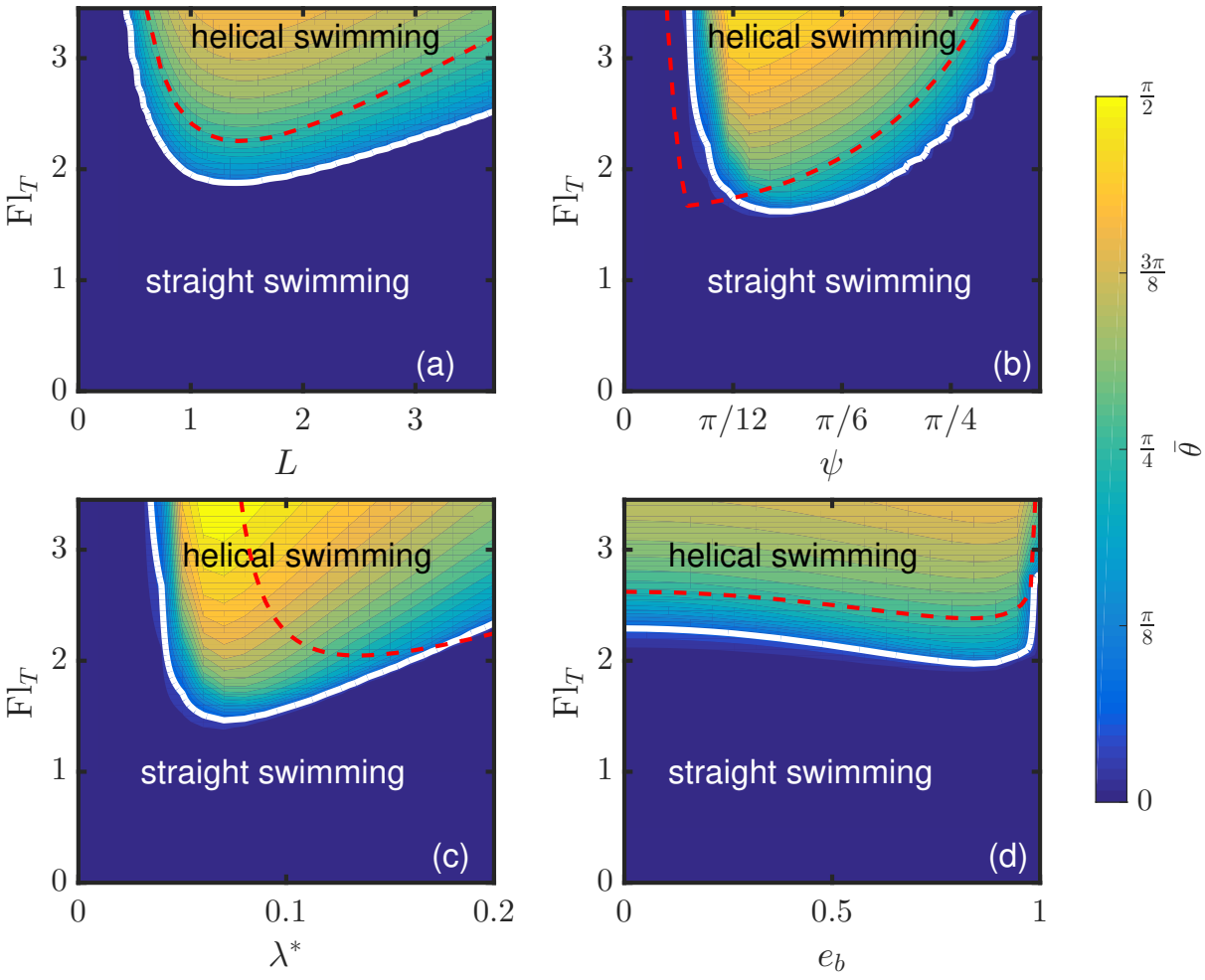

Figure 2.6: Average bending angle $\bar{\theta}$ as a function of (a) (f, L), (b) (f, ψ), (c) (f, λ^*), (d) (f, e_b), obtained by running simulations for an initially straight swimmer. In each case, the constant parameters are taken from the base geometry. The stability boundary is marked by the white line for the full model and dashed red line for the dynamic toy model, denoting the critical flexibility $\mathrm{Fl}_{T,crit}$. The oscillations in the plots are purely numerical, as for certain parameter regimes, the bending timescale is far greater than the timescale of flagellar rotation.

assume this accurately captures the periodic to quasi-periodic transition discussed previously. Because the dynamic toy model provides a simple stability criterion, we compare the analytical predictions to the numerical full model results by converting the propulsive-force-based flexibility number Fl_F to the torque-based flexibility number Fl_F using the following relation:

$$\mathrm{Fl}_T = \frac{T}{F_P(l_b/2)}\mathrm{Fl}_F = \alpha \mathrm{Fl}_F \tag{2.58}$$

We approximate the ratio between the torque and force moment, α by calculating the thrust force from the full model using a constrained 1D swimmer (ignoring wobbles) subject to torque T:

$$\alpha = \frac{T}{F_p(l_b/2)} = \frac{(A_b + A_f)C_f - B_f^2}{(l_b/2)B_f(A_b + A_f)} \tag{2.59}$$

where all scalar resistances are obtained from the (1,1) entries of the corresponding tensor.

Fig. 2.6 shows values of $\bar{\theta}$ (color contours) and stability limits for the full model as functions of parameters as well as stability predictions for the dynamic toy model. For all geometries, $\bar{\theta}$ increases monotonically with Fl_T, as discussed in Section 2.7A for the base geometry. Additionally, for any variation in a single parameter, the stability boundary $\mathrm{Fl}_{T,crit}$ exhibits a minimum. To see how this arises, we note that the compressive force $\mathbf{X}$ responsible for buckling depends on the swimmer's geometry, and there exist particular geometric limits where $\mathbf{X}$ vanishes. When $\mathbf{X} \to \mathbf{0}$, there are no bending moments to sustain bent configurations and $\mathrm{Fl}_{T,crit}$ must be infinitely large for buckling to occur.

In the case of L, as $L \to 0$ or $L \to \infty$, the drag vanishes on the flagellum or body respectively, so in these limits there can be no compressive force on the hook. The limits $\lambda^* \to 0$ or $\psi \to 0$ correspond to a straight flagellum and $\lambda^* \to \infty$ or $\psi \to \pi/2$ to a circular one. In both cases symmetry prevents the generation of any propulsion, and hence no compression. Accordingly, in all these limits the critical flexibility number goes to infinity. Between these limits $\mathbf{X}$ and thus $\mathrm{Fl}_{T,crit}$ must be finite, hence a minimum $\mathrm{Fl}_{T,crit}$ must occur. Lastly, changes in the body parameter $e_b \in (0, 1)$ characterize the transition of body

shape from a sphere to a needle (with constant major axis). For a given Fl_T, decreasing e_b lowers the body drag to make bending easier, but simultaneously reduces the compressive force $\mathbf{X}$. The dynamic toy model is useful in describing this effect, as we can directly write $|\mathbf{X}| = a_f^{\parallel *}(-m_1^0 + m_2^0 + m_3^0)$ for the straight swimmer to see the monotonic decrease in $\mathbf{X}$ as $e_b \to 1$.

Overall, the models show reasonable qualitative agreement with each other. The discrepancy arises because the dynamic toy model misses the force-torque coupling central to the full model and the actual organism biology. Ultimately, the key result here is the apparent minimization of $\mathrm{Fl}_{T,crit}$ over the swimmer's geometry. In physical terms, the minimum $\mathrm{Fl}_{T,crit}$ provides the smallest motor torque, T_{min} required to observe buckling for a given hook strength. This potentially significant result may explain the dimensions of swimming microorganisms and be a governing limit in the design of artificial swimmers and flagella, though we cannot precisely say without further detailed measurements of motor strengths and elastic properties in swimmers.

2.8 Summary

In this chapter we have established a series of concise, computationally inexpensive, linked rigid body models to describe many of the relevant dynamics, kinematics, and characterizations of a uni-flagellated swimmer at low Reynolds number.

A static toy model of a swimmer pinned at its front yields instability only in the unphysical case where the end of the flagellum is constrained to move on a line. A dynamic toy model, for a freely swimming cell but with a simplified propulsion mechanism, does buckle at a critical value of $\mathrm{Fl}_{F,crit}$. The full model rectifies the shortcomings of the toy models by utilizing a rigid helical flagellum that is rotated by a motor torque. The helical chirality incorporates the force-torque coupling central to locomotion, namely generating propulsion via an applied motor torque, as is the case in real organisms. The stability of straight

swimming in this model is reasonably well-predicted by the dynamic toy model, but the full model has more intricate three-dimensional trajectories: “straight" configurations (that actually have a nonzero equilibrium bending angle and a slight wobble) below $\mathrm{Fl}_{T,crit}$ and buckled configurations above $\mathrm{Fl}_{T,crit}$ that yield helical trajectories. The straight-to-buckled bifurcation is a transition from periodic to quasiperiodic evolution of the cell configuration. The swimming paths and kinematics depend on swimmer configuration, with the latter intricately determined by the hook flexibility and the individual body and flagellum geometries. Over a range of geometric parameters, we find a minimum in $\mathrm{Fl}_{T,crit}$ over independent variations of each parameter. This may be a significant result for swimmers relying on buckling instabilities to move, such as *V. alginolyticus*.

A number of simplifications made in the current model can be relaxed by including hydrodynamic interactions and introducing elasticity into the flagellum to expand the model analysis to a more comprehensive range of swimmers. Perhaps the most drastic simplification is the neglect of internal dynamics of the hook, whose winding and unwinding affect its effective mechanical properties and subsequently the course of a swimmer’s motion and trajectories.

Chapter 3

General bacterial swimmer model [1]

3.1 Introduction

As we stated in Chapter 1, numerous swimmer models have been developed to characterize locomotion. In Chapter 2, we devised a linked-rigid-body swimmer consisting of rigid body and flagellum connected by a torsion spring representing the elastic hook [77], with results very comparable to more comprehensive model, as in Ref [96]. However, to consider effects of flagellar elasticity on locomotion, models utilizing a flagellar discretization are required. We note that the formulation of Ref. [85] has been adapted to better represent flagellar shape [1, 62, 115, 117], and we follow the same methodology in this chapter to develop a swimmer model and numerical solution algorithm to be used in the remainder of this work.

3.2 Physical description

Our model swimmer, depicted in Fig. 3.1, consists of a rigid spherical cell body of radius R_b connected to one or more right-handed flagella each via a flexible hook of length L_h, following Nguyen et al. [77]. Each flagellum is a plain, inextensible, elastic helix with uniform

[1] The text of this chapter has been adapted from the manuscript titled "Impacts of multiflagellarity on stability and speed of bacterial locomotion" by F. T. M. Nguyen and M. D. Graham on arXiv and submitted for review to Physical Review E.

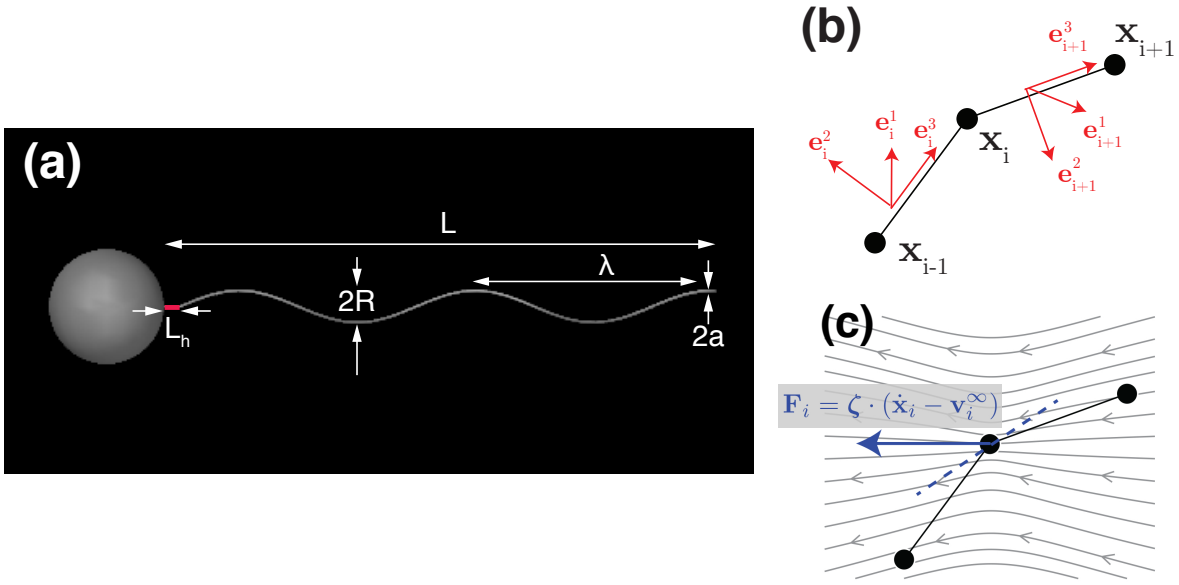

Figure 3.1: (a) Model swimmer with flagellum drawn true to scale The hook segment is shown in red. (b) Discretization of flagellum into nodes $\mathbf{x}_i$ and edges $\mathbf{e}_i$. (c) Flow field generated by a regularized point force (gray lines). The dashed blue line shows the approximation of the local tangent vector at node $\mathbf{x}_i$.

parameters across all flagella. The equilibrium helix geometry is described by radius R, pitch λ, end-to-end length L and filament radius a. [**Note:** in the remaining illustrations in this text, the flagellar thickness will be slightly exaggerated for visual clarity.] The elasticity of the flagella and hooks are described by flexural rigidities K_B and K_{Bh}, respectively. During swimming, motors embedded within the body surface exert torques of constant magnitude T on the hooks, which in turn transmit these torques to their respective flagella.

To understand buckling effects, we note that for a simple Euler beam of length L (with fixed ends under compression and torsion loads), the critical buckling torque would be $2\pi K_B/L$ (with critical force $\pi^2 K_B/L^2$) as in Ref. [108]. Motivated by this observation, we characterize the flexibility of the flagella and hook by dimensionless *flexibility numbers* Fl and Fl_h, defined as

$$\mathrm{Fl} = \frac{T}{K_B/L}, \quad \mathrm{Fl}_h = \frac{T}{K_{Bh}/L_h}. \tag{3.1}$$

We note that Ref. [115] describes an analysis of the bending elasticity of a helical filament

that provides a geometry-dependent correction to the simple estimate K_B.

For a bacterial flagellum, K_B is typically between 1-10 pN μm^2 [53, 98], T between 1-3 pN μm [26], and L between 3-10 μm [90]. This translates to Fl between 0.3 and 30. In our results below we consider Fl up to 6.5; we are not aware of organisms with parameters operating above this limit, as Fl is around 2.5 for *E. coli* (with similar values for *S. typhimurium*) and between 0.2-1.6 for *V. alginolyticus* [26, 98]. Bacterial hooks are roughly 100 nm for uniflagellar organisms and 50-80 nm for peritrichous organisms with very wide ranges of K_{Bh}, ranging from 0.2 pN μm^2 (upper uniflagellar estimate) to 10^{-4} pN μm^2 (peritrichous) [95, 98]. From this data, Fl_h can range from 0.2 to 1000. However, this full range is not needed to observe hook buckling. For example, a *V. alginolyticus cell* has $\mathrm{Fl}_h = 0.2$ in its straight swimming state, and climbs only to $\mathrm{Fl}_h = 1.6$ in its buckled state. We run simulations with $\mathrm{Fl}_h \leq 100$, as this value is sufficient to allow complete hook buckling and there is little additional bending resistance above this value.

3.3 Discretization

The cell body is a rigid sphere with center of mass position $\mathbf{x}_b$ and orientation $\mathbf{q}_b$, a unit quaternion (whose components are Euler parameters). Initially, we align the body orientation with the laboratory frame so that $\mathbf{q}_b = [1\ 0\ 0\ 0]^{\mathrm{T}}$. We select anchor points on the body surface where at each point we attach a hook and flagellum.

Each of the N flagella is discretized into $M-1$ connected straight rods (radius a) which we call *edges*. The edge connected to the body at the anchor point is the hook of length L_h. The hook defines the flagellar axis and is initialized normal to the body surface. The remaining edges are flagellar segments each of length l, constituting a plain helix directly attached to the hook. The points connecting each edge, including endpoints, are called *nodes*. Thus, each flagellum has M nodes $\mathbf{x}_i$ with $i \in [0,\, M-1]$. A schematic is shown in Fig. 3.1(b). Node 0 denotes the anchor point attached to the body surface. For the edge i between nodes

$i-1$ and i, we assign a local orthonormal coordinate system or *triad* $\{\mathbf{e}_i^1, \mathbf{e}_i^2, \mathbf{e}_i^3\}$, where $\mathbf{e}_i^3$ is exactly the edge defined by:

$$\mathbf{e}_i^3 = \frac{\mathbf{x}_i - \mathbf{x}_{i-1}}{|\mathbf{x}_i - \mathbf{x}_{i-1}|}, \quad i \in [1,\, N-1], \tag{3.2a}$$

$$\mathbf{e}_0^3 = \frac{\mathbf{x}_0 - \mathbf{x}_b}{R_b}. \tag{3.2b}$$

Note that $\mathbf{e}_1^3$ defines the hook orientation / flagellar axis. The remaining triad components are defined as follows:

$$\mathbf{e}_i^1 = \frac{\mathbf{e}_{i-1}^3 \times \mathbf{e}_i^3}{|\mathbf{e}_{i-1}^3 \times \mathbf{e}_i^3|}, \tag{3.3a}$$

$$\mathbf{e}_i^2 = \mathbf{e}_i^3 \times \mathbf{e}_i^1. \tag{3.3b}$$

We note that Eqs. 3.2 and 3.3 are used only to initialize the flagellar helix, after which point we integrate the triads directly rather than reconstructing them (as we describe at the end of this section). We do not use Eqs. 3.2 and 3.3 to initialize the edges $\mathbf{e}_0^3$ and $\mathbf{e}_1^3$. We subsequently define a bend angle θ_i and twist angle φ_i that transform triad i to triad $i+1$.

$$\theta_i = \arccos\left(\mathbf{e}_i^3 \cdot \mathbf{e}_{i+1}^3\right), \tag{3.4a}$$

$$\varphi_i = \arctan \frac{\tilde{\mathbf{e}}_i^1 \cdot \mathbf{e}_i^2}{\mathbf{e}_i^1 \cdot \tilde{\mathbf{e}}_i^1}, \tag{3.4b}$$

$$\tilde{\mathbf{e}}_i^1 = [\mathbf{nn} + \cos\theta_i\,(\boldsymbol{\delta} - \mathbf{nn})] \cdot \mathbf{e}_{i+1}^1 - \sin\theta_i\left(\mathbf{n} \times \mathbf{e}_i^1\right), \tag{3.4c}$$

$$\mathbf{n} = \mathbf{e}_i^3 \times \mathbf{e}_{i+1}^3 / \sin\theta_i = \mathbf{e}_{i+1}^1. \tag{3.4d}$$

Here $\boldsymbol{\delta}$ is the identity tensor, and $\tilde{\mathbf{e}}_i^1$ is an intermediate twist-free rotation of $\mathbf{e}_i^1$. For the $N-2$ nodes located between 2 consecutive flagellar edges, we define the generalized curvature, $\boldsymbol{\Omega}_i$ locally at that node:

$$\Omega_i^1 = -\frac{\theta_i}{\sin\theta_i}\mathbf{e}_i^2 \cdot \mathbf{e}_{i+1}^3, \tag{3.5a}$$

$$\Omega_i^2 = +\frac{\theta_i}{\sin\theta_i}\mathbf{e}_i^1 \cdot \mathbf{e}_{i+1}^3, \tag{3.5b}$$

$$\Omega_i^3 = \varphi_i. \tag{3.5c}$$

Eq. 3.5 is also used to find the equilibrium curvature, $\mathbf{\Omega}_{i,\mathrm{eq}}$. With $\mathbf{\Omega}_i$, we can fully describe the flagellar conformation needed to characterize elasticity and its evolution during motion.

3.4 Equations of motion

Our swimmer moves through incompressible Newtonian fluid at zero Reynolds number. The body motion is coupled to that of the elastic flagella. Neglecting inertia, each component of the swimmer (denoted by k), body and individual flagellar elements, is force and torque free. Hence, the overall swimmer is also force and torque free. The hydrodynamic drag (D) balances all external dynamics attributed to elasticity (el), steric interactions (ster), application of motor torque (mot), and constraints (C).

$$\mathbf{F}_k^{\mathrm{D}} + \mathbf{F}_k^{\mathrm{el}} + \mathbf{F}_k^{\mathrm{ster}} + \mathbf{F}_k^{\mathrm{mot}} + \mathbf{F}_k^{\mathrm{C}} = \mathbf{0}, \tag{3.6}$$

$$\mathbf{T}_k^{\mathrm{D}} + \mathbf{T}_k^{\mathrm{el}} + \mathbf{T}_k^{\mathrm{ster}} + \mathbf{T}_k^{\mathrm{mot}} + \mathbf{T}_k^{\mathrm{C}} = \mathbf{0}. \tag{3.7}$$

As we will be discussing body and flagella simultaneously in the remainder of this section, we note that subscript $k = b$ refers to the body and $k = i$ refers to the ith flagellar segment. We adopt two key conventions applied to models used in Refs. [1, 14, 62, 85, 114]: (1) the position and conformation of each flagellum are completely specified from the node positions $\mathbf{x}_i$ and the twist φ_i at any given time; (2) forces translate nodes and torques rotate edges. We adopt these conventions when deriving the forces and torques used in Eqs. 3.6 and 3.7 in the following sections.

3.4.1 Hydrodynamics

We use a discretized slender body theory to treat the hydrodynamics of the flagella: but we must also additionally track the rotational phase φ_i of the filament due to the imposed torque on the flagellum. Each node on a flagellum has velocity $\mathbf{v}_i$ and each edge (with its corresponding triad) rotates with angular velocity $\omega_i = \dot{\varphi}_i$. As a node moves, the surrounding fluid exerts a Stokes law drag force on it. This drag force $\mathbf{F}_i^{\mathrm{D}}$ has three contributions: a local contribution from the motion of node $\mathbf{x}_i$ itself, and far-field contributions induced by the motion of all other flagellar nodes ($\mathbf{v}_{i,\infty}^f$) and the body ($\mathbf{v}_{i,\infty}^b$). In this case

$$\mathbf{F}_i^{\mathrm{D}} = \boldsymbol{\zeta}_i \cdot \left[\left(\mathbf{v}_{i,\infty}^f + \mathbf{v}_{i,\infty}^b - \mathbf{v}_i \right) \right], \tag{3.8}$$

where $\boldsymbol{\zeta}_i$ is an anisotropic friction coefficient for a slender rigid rod of aspect ratio l/a:

$$\boldsymbol{\zeta}_i = \zeta_\perp \boldsymbol{\delta} + (\zeta_\parallel - \zeta_\perp)\mathbf{t}_i\mathbf{t}_i, \tag{3.9}$$

and $\zeta_\perp$ and $\zeta_\parallel$ are the scalar normal and tangential friction coefficients, found in Ref. [90]. We take the axis of the rod at node i to be oriented with the average orientation of edges i and $i+1$: $\mathbf{t}_i = \frac{1}{2}(\mathbf{e}_i^3 + \mathbf{e}_{i+1}^3)$. Using the Stokes drag for a rod rather than a sphere at each node automatically incorporates the anisotropy of drag that leads to flagellar locomotion and allows for a coarser discretization than would be necessary if we had used spheres [52]. We neglect rotlets so that the drag from rotating edge i along its axis is simply:

$$\mathbf{T}_i^{\mathrm{D}} \cdot \mathbf{e}_i^3 = -\zeta_i^r \omega_i, \tag{3.10}$$

following the convention of Refs. [1, 114], where $\zeta_i^r = 4\pi\eta a^2 l$.

The velocity field $\mathbf{v}_{i,\infty}^f$ in Eq. 3.8 arises from the hydrodynamic interactions between the flagella. To compute this field, we treat each node as a regularized point force acting on the fluid, as illustrated in Fig. 3.1(c). The flagellar flow field $\mathbf{v}_{i,\infty}^f$ experienced by node i is

obtained by summing the flows induced by all other flagellar nodes:

$$\mathbf{v}_{i,\infty}^{f} = \sum_{i \neq j} \mathbf{M}_f(\mathbf{x}_i, \mathbf{x}_j) \cdot \mathbf{F}_j, \tag{3.11}$$

where $\mathbf{F}_j = -\mathbf{F}_j^{\mathrm{D}}$ is the force exerted *on* the fluid *by* flagellar node j. The exact expression for the mobility tensor $\mathbf{M}_f$ depends on the swimming domain, i.e. unbounded or bounded fluid, but the general linear form of Eq. 3.11 always holds due to the linearity of Stokes flow. The exact expression for the body induced flow $\mathbf{v}_{i,\infty}^{b}$ will also depend on the swimming domain, but in general, it is linear function of the force and torque exerted *by* the sphere *on* the fluid:

$$\mathbf{v}_{i,\infty}^{b} = \mathbf{M}_b(\mathbf{x}_i - \mathbf{x}_b) \cdot \mathbf{M}_b + \mathbf{M}_b^{r}(\mathbf{x}_i - \mathbf{x}_b) \cdot \mathbf{T}_b, \tag{3.12}$$

With regard to Eq. 3.12, Newton's third law states $\mathbf{F}_b = -\mathbf{F}_b^{\mathrm{D}}$ and $\mathbf{T}_b = -\mathbf{T}_b^{\mathrm{D}}$, and thus we require the body drag to fully calculate flagellar hydrodynamics. In the following chapters, the tensors $\mathbf{M}_f$, $\mathbf{M}_b$, and $\mathbf{M}_{b,r}$ will be explicitly defined for the relevant problem.

The body moves with velocity $\mathbf{v}_b$ and angular velocity $\boldsymbol{\omega}_b$. We write the hydrodynamic force and torque on the sphere using Faxén's laws to account for body-flagella HI as in Ref. [105]:

$$\mathbf{F}_b^{\mathrm{D}} = \zeta_b \left[-\mathbf{v}_b + \left(1 + \frac{R_b^2}{6} \nabla^2 \right) \mathbf{v}_\infty \Big|_{\mathbf{x}_b} \right], \tag{3.13}$$

$$\mathbf{T}_b^{\mathrm{D}} = \zeta_b^{r} \left[-\boldsymbol{\omega}_b + \frac{1}{2} \left(\nabla \times \mathbf{v}_\infty \right) \Big|_{x_b} \right]. \tag{3.14}$$

Here $\zeta_b = 6\pi\eta R_b$ is the translational drag and $\zeta_{b,r} = 8\pi\eta R_b^3$ the rotational drag coefficient for the sphere. The flow $\mathbf{v}_\infty$ has contributions from *all* flagellar points and any additional interactions of the body with the flow domain. Thus for any point $\mathbf{x}$ in the flow domain:

$$\mathbf{v}_\infty(\mathbf{x}) = \sum_j \mathbf{M}_f(\mathbf{x}, \mathbf{x}_j) \cdot \mathbf{F}_j + \mathbf{M}_b^{*} \cdot \mathbf{F}_b + \mathbf{M}_b^{r*} \cdot \mathbf{T}_b. \tag{3.15}$$

Derivatives of $\mathbf{v}_\infty$ in Eqs. 3.13 & 3.14 are evaluated at the body center $\mathbf{x}_b$. In contrast to

Eq. 3.11, all flagellar nodes are included in the summation in Eq. 3.15.

3.4.2 Elasticity

Having previously defined the discrete generalized curvatures $\mathbf{\Omega}_i$ and $\mathbf{\Omega}_{i,\mathrm{eq}}$, we use a discrete version of Kirchhoff's classical theory to write the elastic energy, $\mathcal{E}^{\mathrm{el}}$, of each flagellum:

$$\mathcal{E}^{\mathrm{el}} = \frac{K_{Bh}}{L_h}\left(\theta_0 - \theta_{0,\mathrm{eq}}\right)^2 + \frac{K_B}{l}\sum_{i=1}^{N-2}\left[\sum_{\beta=1}^{2}\left(\Omega_i^{\beta} - \Omega_{i,\mathrm{eq}}^{\beta}\right)^2 + \Gamma\left(\Omega_i^3 - \Omega_{i,eq}^3\right)\right], \tag{3.16}$$

We set $\theta_{0,\mathrm{eq}} = 0$ (straight hook at equilibrium) and assign no twist penalty to allow for free rotation and counter-rotation between body and flagella. We assume the flagellar filament has a circular cross section so that bending is isotropic. The parameter $\Gamma = K_T/K_B$ is a material property called the twist-to-bend ratio, where K_T is the torsional ridity. For a completely incompressible material, $\Gamma = 0.67$. However, Darnton et al. in Ref. [25] found that $\Gamma = 1$ fit their force-extension data for a bacterial flagellum quite well, so we set $\Gamma = 1$ as well.

On each flagellum, derivatives of the elastic energy determine the elastic force on node i of flagellum m and elastic torque on edge i:

$$\mathbf{F}_i^{\mathrm{el}} = -\frac{\partial \mathcal{E}^{\mathrm{el}}}{\partial \mathbf{x}_i}, \tag{3.17}$$

$$\mathbf{T}_i^{\mathrm{el}} = -\frac{\partial \mathcal{E}^{\mathrm{el}}}{\partial \varphi_i}\mathbf{e}_i^3 = -T_i^{\mathrm{el}}\mathbf{e}_i^3. \tag{3.18}$$

The separate treatment of forces and torques in this manner eases the calculation and implementation of twisting and bending deformations on the filament, as noted in Ref. [85]. We also note that these are the forces exerted *by* the filament *on* the fluid, and that the elastic torque is distributed entirely along the bond direction, $\mathbf{e}_i^3$, with all elastic torques normal to $\mathbf{e}_i^3$ decomposed into forces on adjacent nodes.

The elastic dynamics on the body arise from elastic forces acting at flagellar anchor

points:

$$\mathbf{F}_b^{\text{el}} = \sum_{\text{flag.}} \mathbf{F}_0^{\text{el}}, \tag{3.19}$$

$$\mathbf{T}_b^{\text{el}} = R_b \sum_{\text{flag.}} \mathbf{e}_0^3 \ \times \mathbf{F}_0^{\text{el}}, \tag{3.20}$$

where the index 0 denotes an anchor point and the summation is over all flagella.

3.4.3 Steric repulsion

Following Ref. [1], we introduce repulsive steric forces when flagellar nodes approach the body and/or other flagellar nodes too closely. A truncated Lennard-Jones potential is used:

$$U_{LJ} = \frac{F_s\,\sigma}{6}\left[\left(\frac{\sigma}{r_s}\right)^{12} - \left(\frac{\sigma}{r_s}\right)^{6}\right] H\left(2^{1/6}\sigma - r_s\right), \tag{3.21}$$

where H is the heaviside step function, r_s the point of closest approach between two components, σ the cut-off distance (under which steric interactions occur), and F_s the repulsion strength. Following Ref. [1], we set $F_s = 0.8$ pN and set $\sigma = 4a$ for numerical stability. Accounting for every possible interaction among all components (body and flagellar nodes), we write the total steric force on the body and each flagellar node as:

$$\mathbf{F}_b = \sum_i \mathbf{f}_{b,i}, \tag{3.22}$$

$$\mathbf{F}_i^{\text{ster}} = \mathbf{f}_{i,b} + \sum_{j \neq i} \mathbf{f}_{i,j}, \tag{3.23}$$

where the general notation $\mathbf{f}_{p,q}$ denotes the steric force *on* component p *due to* contact with q. If we let $\mathbf{r}_s$ be the vector connecting closest points of contact (magnitude r_s and direction $\hat{r}_s$), we obtain the general force equation from the potential energy (Eq. 3.21):

$$\mathbf{f}_{p,q}^s = -\frac{\mathrm{d}U_{LJ}(r_s)}{\mathrm{d}\mathbf{r}_s}. \tag{3.24}$$

We know describe calculation of $\mathbf{r}_s$ for all interactions.

First we resolve body-flagellum interactions. For the ith flagellar edge, we find the point $\mathbf{x}^*_{i,b}$ on that edge closest to the body center $\mathbf{x}_b$ using a projection:

$$\mathbf{x}^*_{i,b} = \mathbf{x}_{i-1} + \mathcal{F}\left[-(\mathbf{x}_{i-1} - \mathbf{x}_b)\cdot \mathbf{e}_i^3\right]\mathbf{e}_i^3, \tag{3.25}$$

$$\mathcal{F}[h] = \begin{cases} 0, & h < 0 \\ h, & h \in [0, l_{i,\mathrm{eq}}] \\ l_{i,\mathrm{eq}}, & h > l_{i,\mathrm{eq}} \end{cases} . \tag{3.26}$$

Here the truncation function $\mathcal{F}$ ensures that $\mathbf{x}^*_{i,b}$ is on edge i. If $r_s = |\mathbf{x}^*_{i,b} - \mathbf{x}_b| < \sigma$, we apply equal and opposite repulsive forces $\mathbf{f}^s_{i,b} = -\mathbf{f}^s_{b,i}$ on the body and edge i. On the former, we simply write $\mathbf{f}_{b,i} = -\mathbf{f}^s_{i,b}$. For the latter, we track translation only on adjacent nodes, so we decompose the force $\mathbf{F}^s$ acting on $\mathbf{x}^*_{i,b}$ to equivalent forces acting on $\mathbf{x}_{i-1}$ and $\mathbf{x}_i$:

$$\mathbf{f}_{i-1,b} = \frac{h_{i,b}}{l}\mathbf{f}^s_{i,b}, \quad \mathbf{f}_{i,b} = \frac{l - h_{i,b}}{l}\mathbf{f}^s_{i,b}, \tag{3.27}$$

where $h_{i,b} = |\mathbf{x}^*_{i,b} - \mathbf{x}_i|$.

For the same flagellar edge i, we must also calculate the closest contact with **every** other flagellar edge j. We use Eqs. A1-A3 in Ref. [1] along with Eq. 3.26. We define $\mathbf{x}^*_{i,j}$ and $\mathbf{x}^*_{j,i}$ as the closest contacts located on edges i and j respectively. If $r_s = |\mathbf{x}^*_{i,j} - \mathbf{x}^*_{j,i}| < \sigma$ we again apply an equal and opposite steric force $\mathbf{f}^s_{i,j} = -\mathbf{f}^s_{j,i}$ to each edge, and decompose these forces into equivalent forces on adjacent nodes, as we did in Eq. 3.27:

$$\mathbf{f}_{i-1,j} = \frac{h_{i,j}}{l}\mathbf{f}^s_{i,j}, \quad \mathbf{f}_{i,j} = \frac{l - h_{i,j}}{l}\mathbf{f}^s_{i,j}, \tag{3.28}$$

$$\mathbf{f}_{j-1,i} = -\frac{h_{j,i}}{l}\mathbf{f}^s_{i,j}, \quad \mathbf{f}_{j,i} = -\frac{l - h_{j,i}}{l}\mathbf{f}^s_{i,j}, \tag{3.29}$$

where $h_{i,j} = |\mathbf{x}^*_{i,j} - \mathbf{x}_i|$ and $h_{j,i} = |\mathbf{x}_{j,i} - \mathbf{x}_j|$.

We end this appendix with remarks on steric torques. Because each steric force on the body is normal to the body surface, no moments arise and thus $\mathbf{T}_b^{\text{ster}} = \mathbf{0}$. For the flagella, because we apply all steric forces directly on nodes, no moments arise here either, and thus $\mathbf{T}_i^{\text{ster}} = \mathbf{0} \; \forall \; i$.

3.4.4 Motor

In keeping with the propagation of motor torque around a bend presented in Nguyen et al. [77], we write the motor torque on each flagellum as:

$$\mathbf{T}_1^{\text{mot}} = \frac{T}{2}\left(\mathbf{e}_0^3 + \mathbf{e}_1^3\right). \tag{3.30}$$

Note that the motor torque is only applied to the first segment (hook) of each flagellum. When there is no hook bending, i.e. $\theta_0 = 0$, we simply have $\mathbf{T}_1^{\text{mot}} = T\mathbf{e}_1^3$. For the general case of a bent hook, we decompose $\mathbf{T}^{\text{mot}}$ to a torque acting along the hook direction $\mathbf{e}_1^3$ and forces acting on the adjacent nodes 0 and 1.

$$\mathbf{F}_{0,1}^{\text{mot}} = \pm\frac{T}{2L_h}\mathbf{e}_0^3 \times \mathbf{e}_1^3. \tag{3.31}$$

By Newton's third law, a corresponding counter-torque $-\mathbf{T}_1^{\text{mot}}$ is exerted on the body for each flagellum (leading to counter-rotation of the body) – we account for this contribution along with anchor point contributions (latter is zero when $\theta_0 = 0$):

$$\mathbf{T}_b^{\text{mot}} = \sum_{\text{flag.}}\left[-\mathbf{T}_1^{\text{mot}} + R_b\mathbf{e}_0^3 \times \mathbf{F}_0^{\text{mot}}\right], \tag{3.32}$$

$$\mathbf{F}_b^{\text{mot}} = \sum_{\text{flag}}\mathbf{F}_0^{\text{mot}}. \tag{3.33}$$

3.4.5 Constraints

We will enforce constraints explicitly rather than using very stiff springs, as we desire reasonable simulation times when working with complex multi-flagellar swimmers in later chapters. The formulation presented here involves a number of constraints: 1) $\mathbf{q}_b$ remains a unit quaternion, 2) all flagellar anchors undergo rigid body motion, i.e. anchors follow body surface, 3) inextensibility of all flagellar segments. The corresponding equations [14] are, respectively,

$$\mathbf{q}_b \cdot \mathbf{q}_b - 1 = 0, \tag{3.34}$$

$$\mathbf{q}_b \mathbf{x}_0^{(b)} \mathbf{q}_b^{-1} - (\mathbf{x}_0 - \mathbf{x}_b) = \mathbf{0}, \tag{3.35}$$

$$(\mathbf{x}_{i+1} - \mathbf{x}_i) \cdot (\mathbf{x}_{i+1} - \mathbf{x}_i) - l^2 = 0. \tag{3.36}$$

We set l_i to be L_h for a hook and l otherwise. In Eq. 3.35, $\mathbf{x}_0^{(b)}$ is a constant vector denoting the anchor point in the body-fixed frame of reference, and the conjugation operation $\mathbf{q}_b \mathbf{x}_0^{(b)} \mathbf{q}_b^{-1}$ rotates $\mathbf{x}_0^{(b)}$ to the laboratory frame. While the constraint forces and torques $\mathbf{F}_k^{\mathrm{C}}$ and $\mathbf{T}_k^{\mathrm{C}}$ may be obtained analytically through differentiation, it is simpler to calculate them numerically, as we discuss later. An alternative formulation utilizing stiff springs (e.g. as in Ref. [115]) to enforce inextensibility is discussed in Appendix C.

3.5 Mobility formulation

Having completely specified the dynamics for our general swimmer, we proceed to formulate a condensed representation of the swimmer's motion. Following Ref. [14], we condense the swimmer's position variables into a single vector $\mathbf{y}$ and dynamic variables into vector $\mathbf{f}$ as shown below:

$$\mathbf{y} = \begin{bmatrix} \mathbf{q}_b \\ \mathbf{x}_b \\ \mathbf{x}_f \end{bmatrix}, \quad \mathbf{f} = \begin{bmatrix} \mathbf{t}_b \\ \mathbf{F}_b \\ \mathbf{F}_f \end{bmatrix} \tag{3.37}$$

where $\mathbf{t}_b \in \mathbb{R}^4$ is a quaternion multiplication of $[0\,\mathbf{T}_b]^{\mathrm{T}}$ and $\mathbf{q}_b$, with $\mathbf{x}_f$ and $\mathbf{F}_f$ vectors of all flagellar nodes and forces. For NM total nodes, $\mathbf{x}_f$, $\mathbf{F}_f \in \mathbb{R}^{3NM}$ so that $\mathbf{y} \in \mathbb{R}^{3NM+7}$. Exploiting the linearity of Stokes flow, we write the linear relation between the swimmer velocity $\dot{\mathbf{y}}$ and the swimmer forces $\mathbf{f}$ via a grand mobility matrix $\mathcal{M}$:

$$\dot{\mathbf{y}} = \mathcal{M} \cdot \mathbf{f} \tag{3.38}$$

$$\mathcal{M} = \left[\begin{array}{cc|ccc} \mathbf{M}_b^r & \mathbf{M}^\dagger & \mathbf{M}_{b,f1}^r & \dots & \mathbf{M}_{b,fN}^r \\ \mathbf{M}^\ddagger & \mathbf{M}_b & \mathbf{M}_{b,f1} & \dots & \mathbf{M}_{b,fN} \\ \hline \mathbf{M}_{f1,b}^r & \mathbf{M}_{f,b} & & & \\ \vdots & \vdots & & \mathbf{M}_F & \\ \mathbf{M}_{fN,b}^r & \mathbf{M}_{fN,b} & & & \end{array}\right], \tag{3.39}$$

We emphasize that the components of $\mathcal{F}$: $\mathbf{T}_k$ and $\mathbf{F}_k$, are the *non*-hydrodynamic torque and force on component k derived above. The matrix components of $\mathcal{M}$ capture the interaction of component k with the fluid *and* with other swimming components, as seen in Eqs. 3.11 - 3.15. We write the general expressions for some of the matrices below.

We first resolve body rotation and flagellar forcing. We use Faxen's second law to resolve $\mathbf{M}_{b,fi} \in \mathbb{R}^{4\times3}$, the rotation of the body induced by a *single* flagellar node i:

$$\tilde{\boldsymbol{\omega}}_{b,i} = \frac{1}{2}\left(\boldsymbol{\nabla} \times \mathbf{M}_f(\mathbf{x}, \mathbf{x}_i) \cdot \mathbf{F}_i)\right|_{} = \frac{1}{2}\mathbf{M}_{f\times} \cdot \mathbf{F}_i \ \in \mathbb{R}^3, \tag{3.40}$$

where $\mathbf{M}_{f\times} \in \mathbb{R}^{3\times3}$ is the (analytical) tensor expression for the curl of $\mathbf{M}_f$. Note that $\dot{\mathbf{q}}_b \in \mathbb{R}^4$, $\mathbf{M}_{b,fi}$ must also contain a conversion factor, which we obtain from the relation between $\dot{\mathbf{q}}$ and $\boldsymbol{\omega}$. Finally, we have:

$$\mathbf{M}_{b,fi} = \frac{1}{4}\mathbf{Q} \cdot \mathbf{M}_{f\times}, \tag{3.41}$$

where we have expressed right-quaternion multiplication by $\mathbf{q}$ as a left-hand matrix multi-

plication so that

$$\mathbf{Q}(\mathbf{q}) = \begin{bmatrix} -q1 & -q2 & -q3 \\ q0 & q3 & -q2 \\ -q3 & q0 & q1 \\ q2 & -q1 & q0 \end{bmatrix} \tag{3.42}$$

Conversely, the tensor $\mathbf{M}^r_{fi,b} \in \mathbb{R}^{3\times 4}$ characterizes the the force on a flagellar node due to a rotation on the body. From Eq. 3.12, we must have:

$$\mathbf{M}^r_b(\mathbf{x}_i - \mathbf{x}_b) \cdot \mathbf{T}_b = \mathbf{M}^r_{fi,b} \cdot \mathbf{t}_b \tag{3.43}$$

Thus using the relation between $\mathbf{t}_b$ and $\mathbf{T}_b$:

$$\mathbf{M}^r_{fi,b} = \mathbf{M}^r_b(\mathbf{x}_i - \mathbf{x}_b) \cdot \mathbf{Q}^{\mathrm{T}}, \tag{3.44}$$

where we let $\mathbf{Q}^{\mathrm{T}}$ denote right-quaternion multiplication by $\mathbf{q}^{-1}$ as a left-hand matrix multiplication. We now move on to resolving body forcing and flagellar forcing. Faxén's first law simply states:

$$\mathbf{M}_{b,fi} = \left(1 + \frac{R_b^2}{6}\nabla^2\right) \mathbf{M}_f(\mathbf{x}, \mathbf{x}_i)\Big|_{\mathbf{x}_b} \tag{3.45}$$

In the other direction, we have:

$$\mathbf{M}_{fi,b} = \mathbf{M}_b(\mathbf{x}_i - \mathbf{x}_b) \tag{3.46}$$

In the following chapters, components of $\mathcal{M}$ will be defined specific to the contexts of the problem at hand, so that $\mathcal{M}$ may be utilized in conjunction with a solution algorithm, which we discuss next.

3.6 Projection algorithm

Our general solution contains two main parts: in the current state of the swimmer, we first use a projection algorithm to translate the flagellar nodes, and translate and rotate the rigid body. After this constrained motion is applied, we then calculate then update the conformation on each flagellar edge, i.e. the local triads.

Returning to Eq. 3.38, we may split $\mathbf{f}$ into unconstrained (superscript †) and constrained pieces:

$$\begin{aligned}\dot{\mathbf{y}} &= \mathcal{M} \cdot \left[\mathbf{f}^{\dagger} - \nabla \mathbf{C}^{\mathrm{T}} \cdot \mathbf{\Lambda}\right] \\ &= \dot{\mathbf{y}}^{\dagger} - \mathcal{M} \cdot \boldsymbol{\nabla} \mathbf{C}^{\mathrm{T}} \cdot \mathbf{\Lambda}.\end{aligned} \tag{3.47}$$

We calculate $\mathbf{f}^{\dagger}$ from $-\mathbf{T}_k^{\mathrm{D}}$ and $-\mathbf{F}_k^{\mathrm{D}}$ in Eqs. 3.6 and 3.7. The vector $\mathbf{C} \in \mathbb{R}^{N_C}$ contains the N_C constraint equations: 3.34, 3.35, 3.36, and the gradient is taken with respect to $\mathbf{y}$. The vector $\mathbf{\Lambda} \in \mathbb{R}^{N_C}$ contains the Lagrange multipliers enforcing the constraint

$$\mathbf{C} = \mathbf{0}. \tag{3.48}$$

Together, Eqs. 3.47 and 3.48 form a differential-algebraic system of $(3NM+7+N_C)$ equations and $(3NM + 7 + N_C)$ variables ($\mathbf{y}$ and $\mathbf{\Lambda}$). As stated in the main text, we solve this system via projection on the unconstrained velocity $\dot{\mathbf{y}}^{\dagger}$.

For the current swimmer configuration at time t_n, we opt to calculate $\dot{\mathbf{y}}^{\dagger}$ directly from Eqs. 3.8, 3.13, 3.14, along with $\mathbf{v}_0^{\dagger} = \mathbf{v}_b^{\dagger} + \boldsymbol{\omega}_b^{\dagger} \times R_b \mathbf{e}_0^3$ on the flagellar anchors[2]. We then take an unconstrained time step with a forward Euler method:

$$\mathbf{y}^{\dagger}(t_{n+1}) = \mathbf{y}(t_n) + \dot{\mathbf{y}}^{\dagger} \Delta t \tag{3.49}$$

[2]**Remark**: Because we have opted to apply anchor point dynamics directly to the body, our "unconstrained" velocity (†) mirrors the equations outlined in Appendix C for the stiff-spring implementation (full differential equation system) with the stretching energy omitted, in lieu of starting with $\dot{\mathbf{y}} = \mathcal{M} \cdot \mathbf{f}^{\dagger}$. As we show with Fig. 3.2, this unconstrained step still produces valid results.

Generally, $\mathbf{y}^{\dagger}(t_{n+1})$ does not satisfy $\mathbf{C} = \mathbf{0}$, and projection is required to correct this result. Starting with Eq. 3.47, we follow the derivation presented in Ref. [36] where constraints are evaluated at t_{n+1}. Thus, following from the derivation presented in Appendix B, we present the projection equations below:

$$\Delta t \left[\nabla \mathbf{C} \cdot \mathcal{M} \cdot \nabla \mathbf{C}^{\mathrm{T}}\right] \mathbf{\Lambda} = \mathbf{C}. \tag{3.50}$$

$$\delta \mathbf{y} = -\Delta t \, \mathcal{M} \cdot \nabla \mathbf{C}^{\mathrm{T}} \cdot \mathbf{\Lambda}. \tag{3.51}$$

Here $\delta \mathbf{y}$ denotes the difference between the constrained and unconstrained solutions. Eq. 3.50 forms a $N_C \times N_C$ linear system of equations for $\mathbf{\Lambda}$, and Eq. 3.51 forms a $(3NM + 7) \times (3NM + 7)$ linear system for $\delta \mathbf{y}$. The projection steps are as follows:

1. Start with unconstrained soln: $\mathbf{y}^* = \mathbf{y}^{\dagger}(t_{n+1})$.
2. Do projection (steps 3-5) while $|\mathbf{C}(\mathbf{y}^*)| > \epsilon$.
3. Solve Eq. 3.50 for $\mathbf{\Lambda}$.
4. Solve Eq. 3.51 for $\delta \mathbf{y}$.
5. Update: $\mathbf{y}^* + \delta \mathbf{y} \mapsto \mathbf{y}^*$.
6. Constrained solution: $\mathbf{y}(t_{n+1}) = \mathbf{y}^*$.

We set our tolerance to $\epsilon = 10^{-12}$. With the finished projection, we use the constrained solution $\mathbf{y}(t_{n+1})$ to calculate the constrained velocities:

$$\mathbf{v} = \dot{\mathbf{y}} = [\mathbf{y}(t_{n+1}) - \mathbf{y}(t_n)]/\Delta t \tag{3.52}$$

Once we have translated all flagellar nodes, we must update the flagellar conformation at t_{n+1}, i.e. the triads on each edge. One way is to use $\varphi_i(t_{n+1}) = \varphi_i(t_n) + \omega_i \Delta t$ together with Eq. 3.3 to reconstruct the triads from the new node positions and new φ_i's. However, we choose to time-integrate the edge triads directly. Following the formulation in [85], we account for torques along the edge direction $\mathbf{e}_i^3$ and for node displacements. Thus for triad i,

$$\dot{\mathbf{e}}_i^{\beta} = \boldsymbol{\omega}_i \times \mathbf{e}_i^{\beta} \quad (\beta = 1, 2, 3, \quad i \neq 0), \tag{3.53}$$

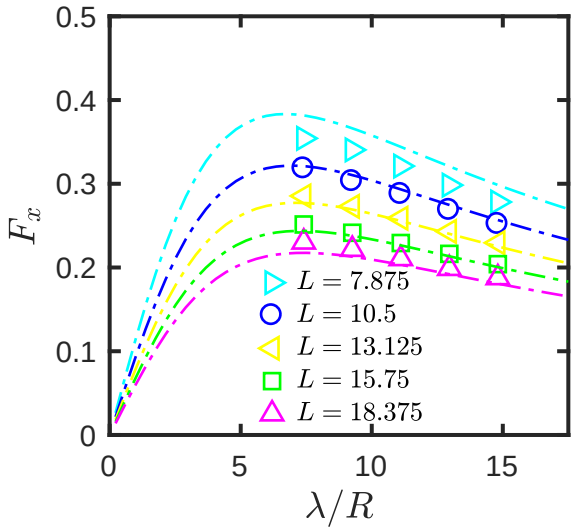

Figure 3.2: Propulsive force on body F_x vs. λ/R at varying flagellar lengths. Symbols are simulations with stiff flagellum and hooks with all hydrodynamic interactions turned off, and lines are the RFT calculations from Chapter 2.

$$\boldsymbol{\omega_i} \cdot \mathbf{e}_i^1 = -|\mathbf{x}_i - \mathbf{x}_{i-1}|^{-1} \, \mathbf{e}_i^2 \cdot (\mathbf{v}_i - \mathbf{v}_{i-1}), \tag{3.54a}$$

$$\boldsymbol{\omega_i} \cdot \mathbf{e}_i^2 = +|\mathbf{x}_i - \mathbf{x}_{i-1}|^{-1} \, \mathbf{e}_i^1 \cdot (\mathbf{v}_i - \mathbf{v}_{i-1}), \tag{3.54b}$$

$$\boldsymbol{\omega_i} \cdot \mathbf{e}_i^3 = \omega_i = \zeta^{r-1} \left[T_i^{\text{el}} + \delta_{i1} (\mathbf{T}_i^{\text{mot}} \cdot \mathbf{e}_i^3) \right], \tag{3.54c}$$

where we have set all other torques to zero, as discussed previously. The general architecture of the code used to execute the projection algorithm can be found in Appendix D. A quick calculation in Fig. 3.2 shows good quantitative agreement for the body propulsion between RFT predictions from Chapter 2 and simulations with our full elastic model using stiff parameters and no hydrodynamic interactions.

3.7 Summary

This chapter has described in detail the treatment and discretization of a rigid body with attached elastic flagella. We also develop the full equations of motion and necessary constraints

that account for the actuation mechanism, elasticity, steric interactions, and hydrodynamics. Other than basic geometric parameters, the problems addressed in the later chapters of this work typically require only the specification of the mobility matrix, as we shall do in Chapters 4 and 5. We also note that the constraints introduced in this chapter are the merely the minimum number needed to ensure inextensibility of the flagellar filaments and their attachment to the body. In general, we can freely apply additional constraints as needed.

Chapter 4

Bacterial swimmer in free fluid [1]

4.1 Introduction

Most bacteria (>90%) are *monotrichous* or uniflagellar, consisting of only one cell body with one attached flagellum [67], e.g. *Vibrio alginolyticus*, *Rhodobacter sphaeroides*, and *Caulobacter crescentus* [65]. Despite their simplicity, uniflagellar locomotion exhibits both normal (straight) trajectories and complex trajectories because these swimmers are able to exploit elastic instabilities in the hook and filament [63, 98]. Hook buckling leads to misalignment between cell body and flagellum resulting in complex trajectories, as discussed in Chapter 2. While an elastic flagellum can also buckle [53], this feature does not seem to be observed in normal bacterial locomotion [63]. Rather, flagellar elasticity is associated with polymorphic phase transformations and bundling of flagella, and, as discussed in Chapter 1, is particularly important for multiflagellar or peritrichous bacteria. The flagella of these peritrichous swimmers require very high hook compliance, much more so than the uniflagellar swimmers, to allow flagellar reorientation around the cell body [26]. Compliance in the flagellar filament itself is especially important in allowing flagella to interact and entangle to form bundles and generate thrust. However, multiflagellar swimmers do not necessarily

[1] Parts of this chapter have been adapted from the manuscript titled "Impacts of multiflagellarity on stability and speed of bacterial locomotion" by F. T. M. Nguyen and M. D. Graham on arXiv and submitted for review to Physical Review E.

swim faster than their uniflagellar counterparts, and thus these swimmers likely utilize their flagella for other purposes as well [26].

In summary, experiments show uniflagellar bacteria require low hook flexibility and low flagellar flexibility to swim, while multiflagellar bacteria require higher flexibilities. Models of bacterial locomotion have captured this phenomenon. Authors in Refs. [77] and [96] both found that for swimmers consisting of a rigid body and flagellum connected by an elastic hook, straight trajectories are not possible when the hook buckles. However, to consider bundling and other effects in elastic flagella, models utilizing a flagellar discretization are required. For instance, Watari et al. in Ref. [120] use a bead-spring model of a swimmer to show bundling of flagella. Janssen et al. in Ref. [52] model flagella as rods connected by torsion springs and show that hydrodynamic interactions induce bundling between anchored elastic flagella, a result not seen for anchored rigid flagella [58]. Adhyapak et al. in Ref. [1] show that elastic flagella attached to a cell body will always bundle given a rotating body and sufficient hook compliance. Riley et al. in Ref. [88] also demonstrate that sufficient hook compliance is necessary for bundling and fast swimming in a multiflagellar swimmer. Authors in Refs. [115] and [53] both showed that buckling of the flagellar filament is possible under high enough torque, signaling limits to flagellar thrust production. Though buckling is expected for general elastic filaments, it is not seen for the range of motor torques and elastic moduli typical of real bacteria [63].

In this chapter, we build upon the elastic models utilized in Refs. [1, 85, 114] to develop a general model describing a rigid spherical cell body with discretized elastic flagella, each attached to the body via a flexible hook. The flagella are driven with constant torque, and we avoid the use of convenient boundary conditions elsewhere in our model. We prescribe hook and flagellar flexibilities independently. First, we verify that a uniflagellar swimmer undergoes straight swimming in the limit of low flexibility to match the results of Refs. [77, 96, 98]. We then vary both flexibility parameters to establish trajectories outside of straight swimming, highlighting the effects of hook and flagellar buckling. We then study

swimmers with two, three and four flagella to determine changes in stability. For the bi- and triflagellar swimmers, we find stability at both high and low flexiblities, and for the isotropic quadriflagellar swimmer, we find stability only at high hook flexibilities. For the quadriflagellar swimmer, we show the robustness of bundling and swimming against flagellar arrangement on the body. Lastly, we examine swimming speed for polar bundles, finding that swimming speed does not scale linearly with flagellar multiplicity.

4.2 Equations of motion

To model our free swimmer, we use the model introduced in Chapter 3 along with the specified equations of motion. Below we specify the tensors used to capture hydrodynamic interactions of the swimmer.

4.2.1 Hydrodynamic interactions in open domain

Recall the mobility relation, Eq. 3.38, defined in the previous chapter:

$$\mathcal{V} = \mathcal{M} \cdot \mathcal{F}$$

Since the calculation of torques and forces was already described in Chapter 3, we then focus on calculating the matrix components of $\mathcal{M}$ for a free swimmer. We start with the body self-mobility. Since rotation and translation are uncoupled for a symmetric body in free flow and the drag on a sphere is isotropic, we may immediately write:

$$\mathbf{M}^{\ddagger} = \mathbf{M}^{\dagger} = \mathbf{0}, \tag{4.1a}$$

$$\mathbf{M}_b^r = \zeta_b^{r-1}\boldsymbol{\delta}, \tag{4.1b}$$

$$\mathbf{M}_b = \zeta_b^{-1}\boldsymbol{\delta}. \tag{4.1c}$$

We next quantify the flagellar mobilities. To each flagellar node $\mathbf{x}_i$, we assign locally the anistropic drag $\boldsymbol{\zeta}_i$ described in Eq. 3.9. To account for far-field effects, we treat the node as a *regularized* point force acting on the fluid. So for the flagella mobility tensor $\mathbf{M}_f$, we write:

$$[\mathbf{M}_f]_{ij} = \begin{cases} \boldsymbol{\zeta}_i, & i = j \\ \mathbf{S}(\mathbf{x}_i - \mathbf{x}_j), & i \neq j \end{cases}, \tag{4.2}$$

where $\mathbf{S}(\mathbf{x})$ is the regularized Stokeslet tensor defined in Appendix A. The regularization parameter ξ for each Stokeslet is chosen so that no flagellar segment can generate a flow faster than its velocity, as in Ref. [52]. We have neglected terms of order $1/r^3$ in Eq. 3.11 following the analysis of Tornberg et al. in Ref. [109] since a is very small relative to all other geometric parameters of our swimmer, and any interactions of $\mathcal{O}(a)$ will be dominated by excluded volume effects. Lastly, we resolve body-flagellum and flagellum-body interactions. The far-field flow generated by each Stokeslet affects the body via Faxéns Laws, Eqs. 3.13 and 3.14, so that:

$$\mathbf{M}^r_{b,fi} = \left[\frac{1}{2}\boldsymbol{\nabla} \times \mathbf{S}(\mathbf{x} - \mathbf{x}_i)\right]_{\mathbf{x}=\mathbf{x}_b} \tag{4.3a}$$

$$\mathbf{M}_{b,fi} = \left[\left(1 + \frac{R_b^2}{6}\nabla^2\right)\mathbf{S}(\mathbf{x} - \mathbf{x}_i)\right]_{\mathbf{x}=\mathbf{x}_b} \tag{4.3b}$$

Derivatives of the Stokeslet are also found in Appendix A. Conversely, we recognize that the body motion will also generate a flow $\mathbf{v}^b_{i,\infty}$ on the flagella. Recall that the flow generated by a single rigid sphere translating and rotating through stagnant viscous fluid can be represented by a linear combination of flow singularities, in this case a Stokeslet, a source dipole ($\mathbf{D}$), and a rotlet ($\mathbf{R}$) (all detailed in Appendix A):

$$\begin{aligned} \mathbf{v}^b_{i,\infty} &= \left[\left(1 + \frac{R_b^2}{6}\nabla^2\right)\mathbf{S}(\mathbf{x} - \mathbf{x}_b)\right]_{\mathbf{x}=\mathbf{x}_i} \cdot \mathbf{F}_b + \mathbf{R}(\mathbf{x}_i - \mathbf{x}_b) \cdot \mathbf{T}_b \\ &= \left[\mathbf{S}(\mathbf{x}_i - \mathbf{x}_b) - \frac{R_b^2}{6}\mathbf{D}(\mathbf{x}_i - \mathbf{x}_b)\right] \cdot \mathbf{F}_b + \mathbf{R}(\mathbf{x}_i - \mathbf{x}_b) \cdot \mathbf{T}_b \end{aligned} \tag{4.4}$$

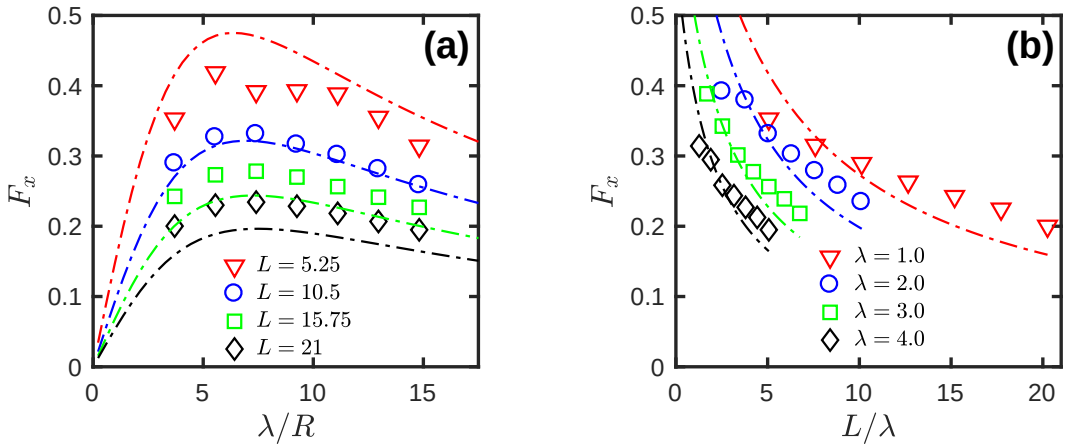

Figure 4.1: Propulsive force on body vs. flagellar helical geometry. The symbols are numerical simulations of the full elastic model with low flexibilities. The dashed lines are the RFT predictions using equations from Chapter 2 and Ref. [77].

We deduce the remaining mobility tensors from the body flow:

$$\mathbf{M}^r_{fi,b} = \mathbf{R}(\mathbf{x}_i - \mathbf{x}_b) \tag{4.5a}$$

$$\mathbf{M}_{fi,b} = \mathbf{S}(\mathbf{x}_i - \mathbf{x}_b) - \frac{R_b^2}{6}\mathbf{D}(\mathbf{x}_i - \mathbf{x}_b) \tag{4.5b}$$

Thus Eqs. 4.1, 4.2, 4.3, and 4.5 completely define $\mathcal{M}$ for the free swimmer.

4.2.2 Model validation with rigid swimmer

Furthermore, we compare low-flexibility uniflagellar simulations using various flagellar geometries to our previous analytical predictions for swimmers with a rigid flagellum in chapter 2 and Ref. [77] using resistive force theory. The propulsive forces on the body along the swimming direction, F_x, are summarized in Fig. 4.1. Our current model results generally follow the same qualitative trend as the RFT calculations, with particularly good quantitative agreement for a few geometries. The moderate quantitative differences arising in all other cases is due to the inclusion of hydrodynamic interactions and elasticity in the current model. For the shorter flagella we test, end effects seem to play a large role, as evidenced

by red triangles in Fig. 4.1(a). In Fig. 4.1(a), $F_x \to 0$ when $\lambda/R \to 0$ or $\lambda/R \to \infty$ as we expect because the flagellum becomes a ring or line, with neither capable of overcoming kinematic reversibility to generate thrust. In Fig. 4.1(b). $F_x \to \infty$ as $L/\lambda \to 0$ because the drag vanishes faster than thrust. On the opposite end, $F_x \to 0$ as $L/\lambda \to \infty$ because there is too much drag on the flagellum.

4.2.3 Simulations and trajectory characterization

As we stated in the introduction, we wish to examine the differences between uniflagellar swimming and multiflagellar swimming, as well as potential transitions in between. We create a simple representation of a peritrichous bacterium using a swimmer with four flagella isotropically distributed across the body surface, whose motors are located at the vertices of a regular tetrahedron inscribed inside the spherical body. We begin our study of a uniflagellar swimmer by placing a single tail at one such vertex. We then create a biflagellar swimmer by adding a second tail at another vertex, then a triflagellar swimmer with a third tail at yet another vertex, until we finally reach the isotropic quadriflagellar swimmer.

To characterize swimming, we first introduce a parameter D capturing the root mean square distance between the body and flagellum, where:

$$D^2 = \sum_{\text{flag}} \frac{1}{N} \sum_{\text{nodes}} |\mathbf{x}_i - \mathbf{x}_b|^2. \tag{4.6}$$

We then define the parameter s_D capturing the deviation in D over time:

$$s_D^2 = \frac{1}{t_2 - t_1} \int_{t_1}^{t_2} D^2 \, \mathrm{d}t - \bar{D}^2, \tag{4.7}$$

$$\bar{D} = \frac{1}{t_2 - t_1} \int_{t_1}^{t_2} D \, \mathrm{d}t, \tag{4.8}$$

where $\bar{D}$ is the time-average of D from t_1 to t_2. The integrals above are evaluated over the last eight time units in a given simulation to filter out any transient effects that may appear

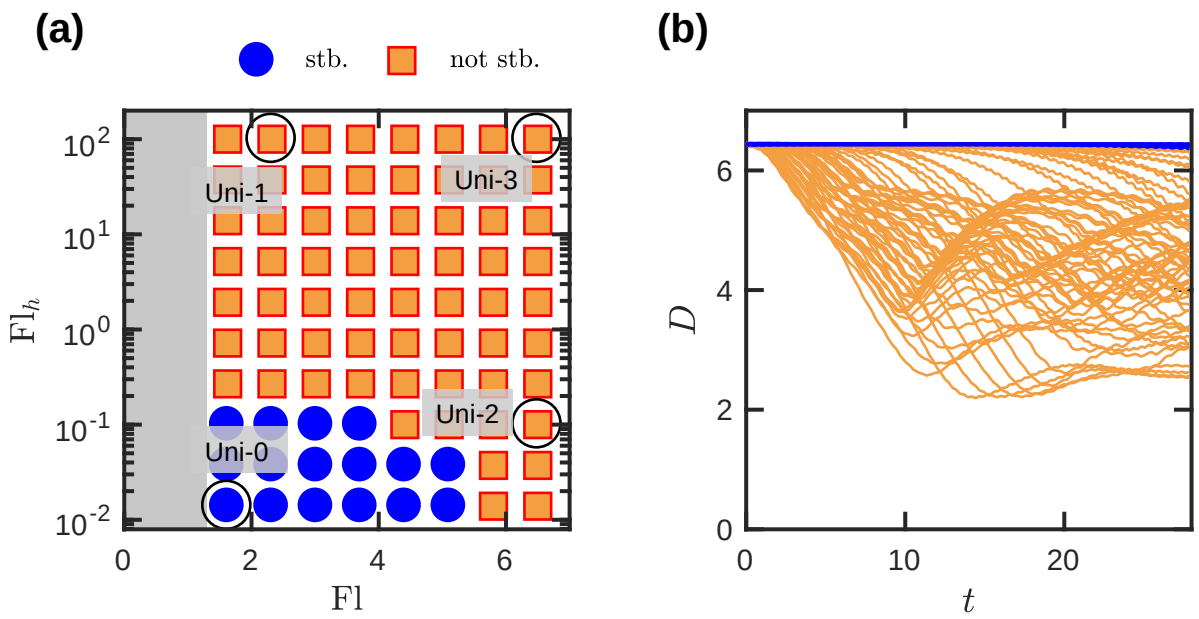

Figure 4.2: (a) Bifurcation diagram for uniflagellar swimmer with $\lambda = 4$. Blue circles denote stable swimming, and orange squares denote non-stable swimming. Black circles denote the cases of (Fl, Fl_h) marked for study: Uni-0 (1.6, 0.014), Uni-1 (2.3, 100), Uni-2 (6.5,0.1), Uni-3 (6.5,100). (b) Flagellar distance from body D vs. time t.

before steady swimming. When $s_D = 0$, there is no change in swimmer shape. For each flagellar multiplicity, we examine D vs. t for each point in flexibility parameter space tested, and group together similar profiles using s_D and $\bar{D}$ to assist in data analysis.

4.3 Uniflagellar swimming

We present the uniflagellar flexibility bifurcation diagram in Fig. 4.2(a). We classify the trajectories into two categories: stable swimming in blue circles, where the swimmer maintains a straight path, and not stable in orange squares, for the remaining cases where straight swimming is not stable. We also show the corresponding time evolution of the distance metric D in Fig. 4.2(b), and point out that we do not see an additional characterization of $D(t)$ other than a straight line or otherwise.

The first case we consider, labeled "Uni-0' in Figure 4.2, is for $\mathrm{Fl} = 1.6, \mathrm{Fl}_h = 0.014$. The motion of this swimmer, which we describe as "stable straight swimming" is depicted by the

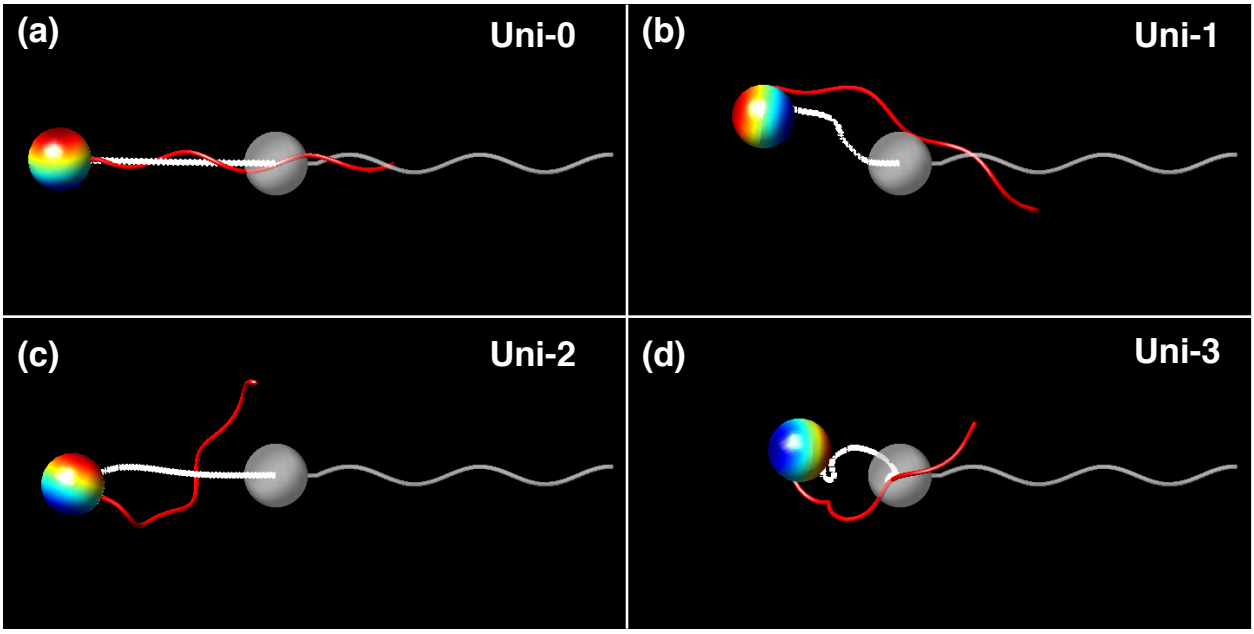

Figure 4.3: Snapshots of uniflagellar swimmer at $t = 0$ (gray) and $t = 28$ (color). The flexibilities (Fl, Fl_h) for each case are: Uni-0 (1.6, 0.014), Uni-1 (2.3, 100), Uni-2 (6.5,0.1), Uni-3 (6.5,100). White lines follow the body center $\mathbf{x}_b$ over 28 time units.

snapshot in Figure 4.3(a). Here the right-handed flagellum rotates clockwise with negligible deformation to produce constant thrust on the body, with the latter slowly rotating counter-clockwise as it translates. The swimming trajectory is linear indicating constant speed (about 0.24 in dimensionless units), and the hook is virtually unbent. We note that at $t = 0$, the flagellar axis is not aligned with the center of a cell because the flagellum is a plain helix. However, once the flagellum begins rotating, the flagellar shape slightly tapers near the body and the flagellar helical axis aligns with the center of the body during flow. Despite this alignment at steady-state, the hook angle θ_0 is not exactly zero (though still very small) due to the generation of thrust normal to the helical axis, as noted in Ref. [77]. As we show in Fig. 4.2(a), straight swimming is stable only in a small region of flexibility parameter space, i.e. only at the lowest flexibilities, and so we analyze swimming behavior away from the stability boundary to understand how locomotion changes.

We next consider case "Uni-1," where $\mathrm{Fl} = 2.3$ and $\mathrm{Fl}_h = 100$. We have increased Fl_h by several orders of magnitude from Uni-0 while maintaining a stiff flagellum. In the Fig.

4.3(b) snapshot, we see that the swimmer enters a slow and broad helical wobble rather than swimming in a straight line, and on average moves about 40% slower than Uni-0. Because Fl_h is quite high, the hook buckles quite readily, with the hook angle θ_0 approaching $\pi/2$ in about 5 time units. When the hook buckles, the flagellar axis bends out of alignment with the body, and we expect flagellar deformations if the flagellar axis changes faster than the characteristic bending time of the flagellar filament, as is the case for Uni-1. This phenomenon has an experimental analogue, as Son et al. showed that a *V. alginolyticus* cell with a buckled hook also exhibits a curved flagellum [98]. Those authors point out that this instability enables a uniflagellar cell to reorient and thus better explore its environment. We see a similar result for case "Uni-3" with (Fl, Fl_h) = (6.5,100), as shown by the snapshot in Fig. 4.3(d).

We now consider the case "Uni-2" with $\mathrm{Fl} = 6.5$ and $\mathrm{Fl}_h = 0.1$. In contrast to Uni-0 and Uni-1, here we have a swimmer with a rather stiff hook and very flexible flagellum. From the Fig. 4.3(c) snapshot, we see that the swimmer maintains a straight path for about 15 time units (about 3 times longer than Uni-1) before deviating from the Uni-0 trajectory. Though it is not apparent from the snapshot, the trajectory does eventually stall at long time. Unlike Uni-1, we do not detect any significant directional change in the body. While we do see flagellar curvature, we attribute this solely to the high flagellar flexibility and not due to a changing flagellar axis. Without hook buckling and reorientation, the flagellum stays behind the cell. The distorted flagellar helix is comparable to the simulations of buckled helices with a fixed torque direction reported in [53, 115], but this particular buckling behavior is, to our knowledge, not seen in nature because unlike hook buckling, pure flagellar buckling impedes swimming without any auxiliary benefits.

Given the limited ability of the uniflagellar swimmer to successfully swim in most regions of parameter space, we turn to swimmers with more flagella to see how stability changes.

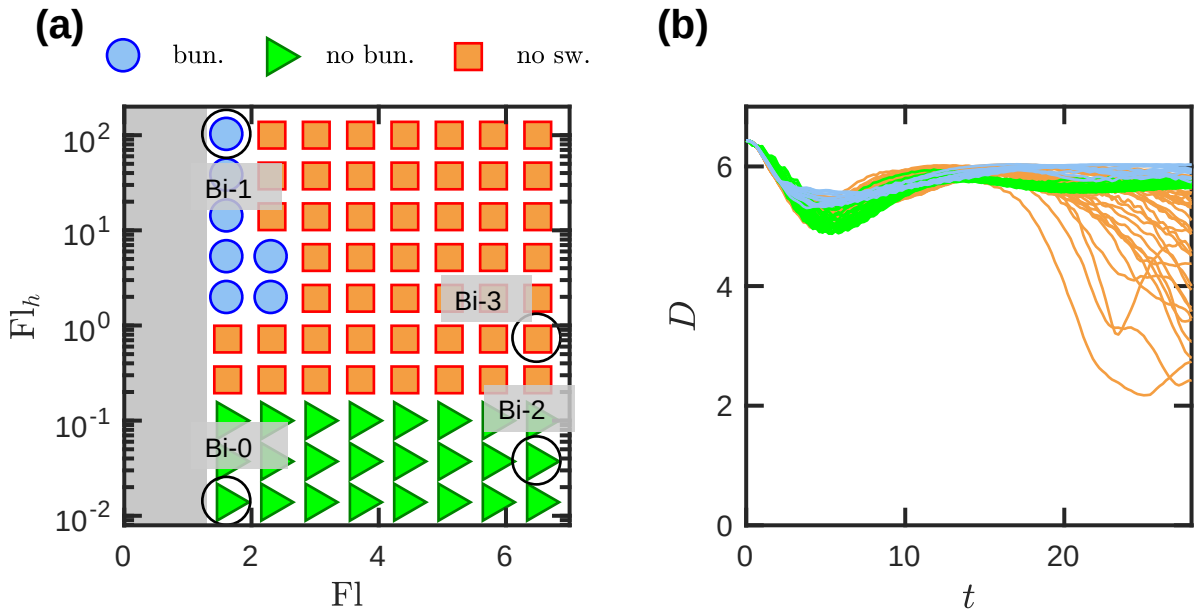

Figure 4.4: (a) Bifurcation diagram for biflagellar swimmer with $\lambda = 4$. Blue circles denote swimming with an intermittent bundle, green triangles denote swimming with no bundling and orange squares denote ineffective swimming. Black circles denote the cases of (Fl, Fl$_h$) marked for study: Bi-0 (1.6, 0.014), Bi-1 (1.6, 100), Bi-2 (6.5,0.038), Bi-3 (6.5,0.72). (b) Flagellar distance from body D vs. time t.

4.4 Biflagellar swimming

We present biflagellar simulation data in Fig. 4.4. Turning our attention first to the plot of $D(t)$ in Fig. 4.4(b), we note that classifying trajectories is not quite as clear-cut as the uniflagellar case. There appears to be a broad range of simulations where D will dip and then level out. However within this range, we find two distinct bands of solutions, shown in blue and green. Simulations in blue exhibit swimming with the formation of intermittent bundles, while simulations in green exhibit swimming with some flagellar deformation, but no bundling at all. Thus for the biflagellar bifurcation diagram shown in Fig. 4.4(a) we classify swimming three categories: blue circles for swimming with bundle, green triangles for swimming with no bundle, and orange squares for everything else that does not swim.

To describe swimming with no bundle, we turn to cases Bi-0 with (Fl, Fl$_h$) = (1.6,0.014) and Bi-2 with (Fl, Fl$_h$) = (6.5,0.038), where the hooks are quite stiff for both cases. The corresponding snapshots shown in Fig. 4.5(a) and (c) show very similar swimmers. In each

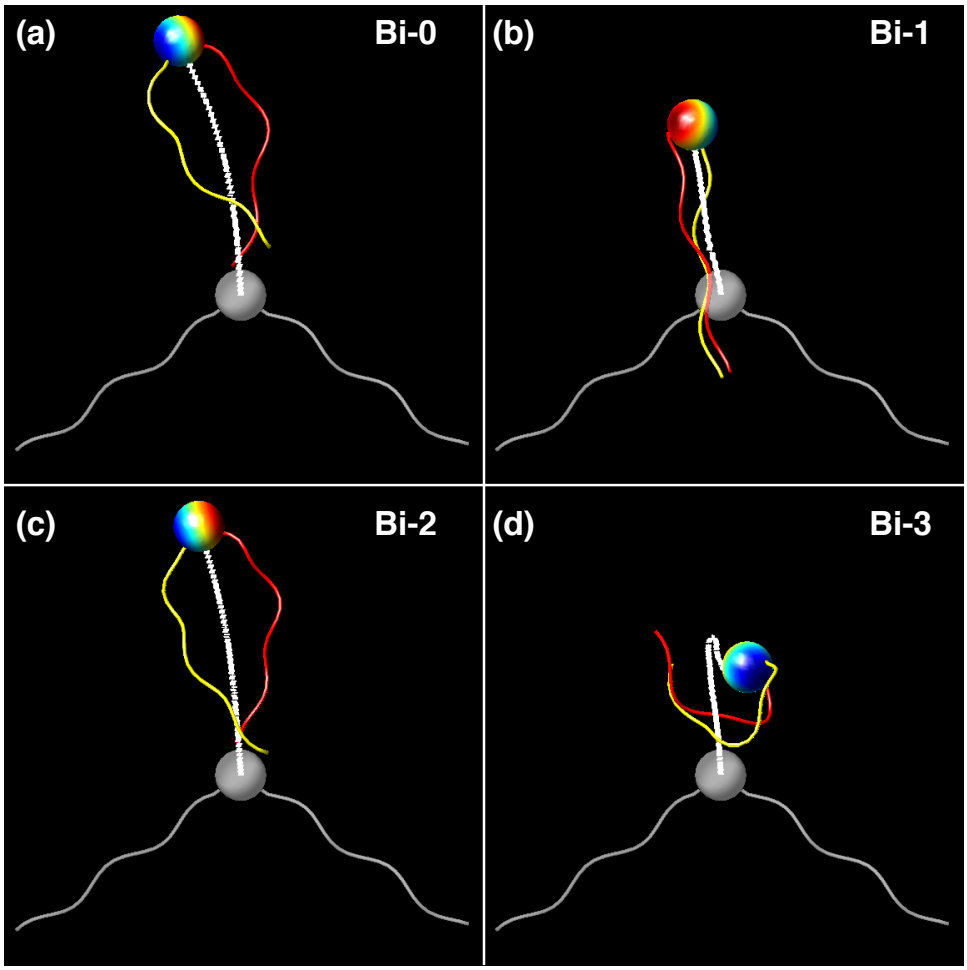

Figure 4.5: Snapshots of quadriflagellar swimmer with flagellar anchors placed at the vertices of a regular tetrahedron at $t = 0$ (gray) and $t = 28$ (color). The flexibilities (Fl, Fl_h) for each case are: Bi-0 (1.6, 0.014), Bi-1 (1.6, 100), Bi-2 (6.5,0.038), Bi-3 (6.5,0.72). White lines follow the body center $\mathbf{x}_b$ over 28 time units.

case, the swimmer translates in a slightly curved path. Though the flagella are in closer proximity than their starting positions (in gray), they do not bundle and remain splayed as the stiff hooks do not permit the flagella to deform near the body. Further away from the body, Bi-2 shows slightly more deformation down its length than Bi-0 (by about 1 helical turn) due to the higher Fl. The speed of both swimmers is faster than Uni-0 by about 50%, but how long this splayed swimmer can maintain this speed remains to be seen.

To see a case where bundling can occur, we examine case Bi-1 with (Fl, Fl_h) = (1.6,100), where the swimmer has stiff flagella and very soft hooks. A snapshot is provided in Fig. 4.4(b), where we see that the flagellar axes now coincide in roughly the same direction, and the flagella exhibit loose bundling in their distal halves. Because the hooks are quite flexible, the flagella will come into contact and bundle intermittently as the swimmer moves. This is because the Bi-1 swimmer actually translates in a helical path rather than a completely straight line once the flagella have gathered, and so bundling alternates between tight and loose as the body wobbles and shifts the flagellar anchors from side to side. The Bi-1 swimmer translates with almost an identical speed to Uni-0, but we note that it is stably swimming at an Fl_h about 4 orders of magnitude larger than Uni-0. Thus, we have found an example of multiflagellarity conferring stability, though the blue stability region at high Fl_h in Fig. 4.4(a) for this particular biflagellar swimmer is small.

We point out that although we see more stable swimming overall for the biflagellar swimmer than the uniflagellar case, ineffective swimming still occupies a significant portion of parameter space. We show one example of a poor swimmer with case Bi-3, (Fl, Fl_h) = (6.5,0.72), whose snapshot is depicted in Fig. 4.5(d). While the swimmer initially appears to swim, it doubles back on its trajectory, and the flagella ultimately end up contorted around the cell body, with no further significant translation detected. We note that the biflagellar swimmer studied here is not a true representation of a peritrichous bacterium, so we turn to even higher flagellar multiplicities to seek stability at higher flexibilities.

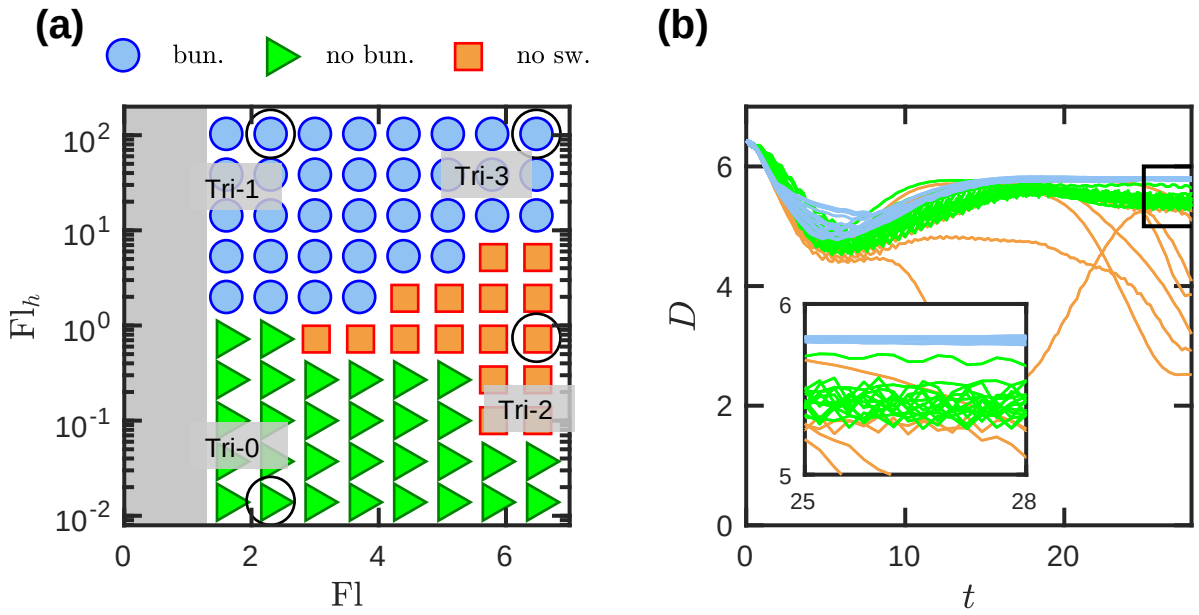

Figure 4.6: (a) (a) Bifurcation diagram for triflagellar swimmer with $\lambda = 4$. Blue circles denote swimming with a stable bundle, green triangles denote swimming with no bundling and orange squares denote ineffective swimming. Black circles denote the cases of (Fl, Fl_h) marked for study: Tri-0 (2.3, 0.014), Tri-1 (2.3, 100), Tri-2 (6.5,0.72), Tri-3 (6.5,100). (b) Flagellar distance from body D vs. time t. The inset shows the demarcation in D for the bundling (blue) and non-bundling (green).

4.5 Triflagellar swimming

We present the triflagellar simulation results in Fig. 4.6. The categorization from $D(t)$ in Fig. 4.6(b) closely follows that of the biflagellar swimmer where we see two distinct plateaus in D, though the distinction is more obvious in this case. As shown in the inset of Fig. 4.6(b), we have a blue lines that collapse almost uniformly near $D = 5.8$, and green lines that show a constant-time average, but with much more noise. We follow the convention for the biflagellar swimmer, blue for swimming with bundles and green swimming without bundles, to construct the bifurcation diagram shown in Fig. 4.6. At a first glance, the stability regions have increased substantially from the uni- and biflagellar cases, as we find swimming in most regions of parameter space.

We first look at swimming with no bundle in case “Tri-0" with (Fl, Fl_h) = (2.3,0.014) and snapshot depicted in Fig. 4.7(a). Analogous to the biflagellar case, the swimmer here

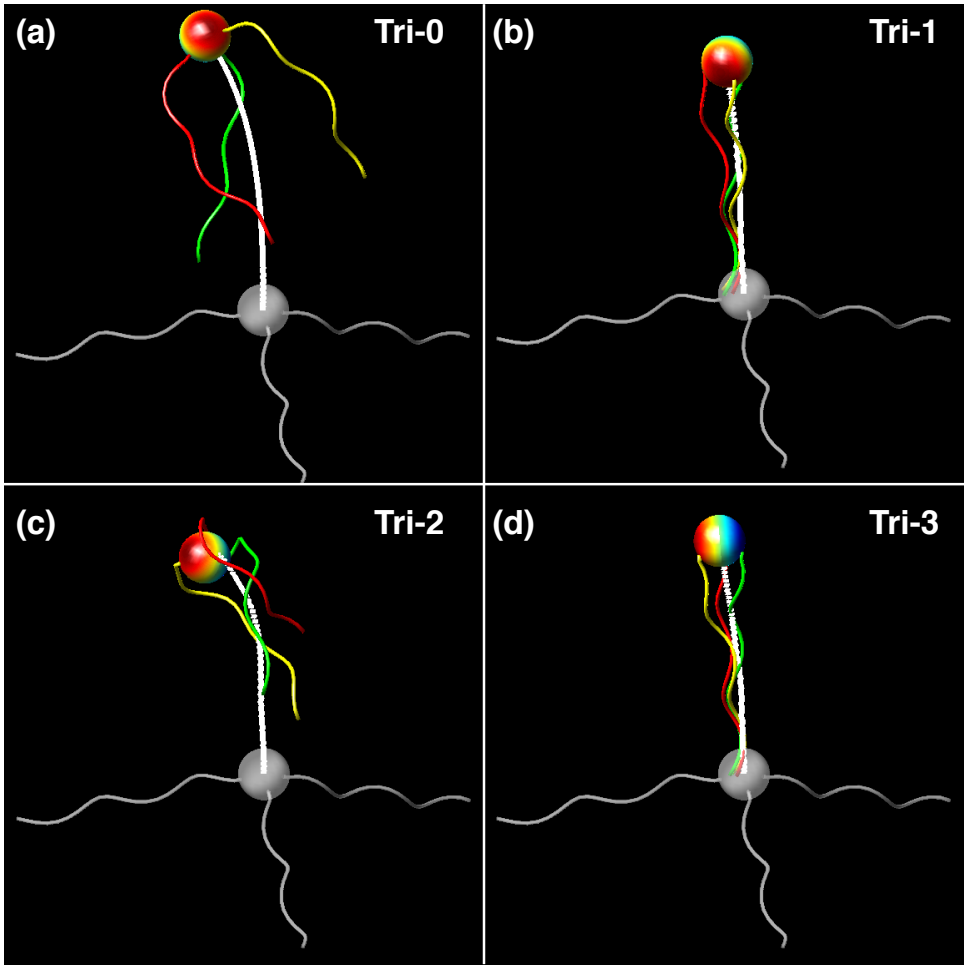

Figure 4.7: Snapshots of triflagellar swimmer at $t = 0$ (gray) and $t = 28$ (color). The flexibilities (Fl, Fl_h) for each case are: Tri-0 (2.3, 0.014), Tri-1 (2.3, 100), Tri-2 (6.5,0.72), Tri-3 (6.5,100). White lines follow the body center $\mathbf{x}_b$ over 28 time units.

swims in a slightly curved path with its flagella splayed in a tripod-like configuration. The swimming speed here is about 60% higher than Uni-0, and not much faster than Bi-0 or Bi-2. Similar results are obtained in the green region of Fig. 4.6(a) even when increasing Fl. Thus at low Fl_h, the three flagella do not seem to confer significantly more stability or speed than two flagella.

At higher Fl_h however, we see a different story on stability. Above a value of $\mathrm{Fl}_h = 1$, all simulations exhibit swimming with a stable bundle, a marked expansion from what we saw for the biflagellar case in Fig. 4.4(a). We show two examples of this stable region in cases "Tri-1" with (Fl, Fl_h) = (2.3,100) and "Tri-3" with (Fl, Fl_h) = (6.5,100) with corresponding snapshots in Figs. 4.7(b) and (d). The swimming process is quite simple – the flagella gather behind the cell from their starting positions and bundle to push the cell forward. This is the classic behavior seen for peritrichous bacteria, as discussed in Chapter 1 and in the introduction to this chapter. Bundling is made easier by the fact that the flagellar motors are distributed on the same hemisphere rather than equatorially. The swimming speed in these cases is about 40% higher than Uni-0, and we emphasize that Fl_h for the cases here are several orders of magnitude higher than Uni-0.

For completeness, we also show one case "Tri-2" with (Fl, Fl_h) = (6.5,0.72) that does not swim effectively due to inhibiting deformations of flagella as shown in Fig. 4.7(c). Overall, we see quite a remarkable stability transformation from two to three flagella, particularly at high Fl values associated with peritrichous bacteria. One issue we must contend with is that most peritrichous organisms do not have as convenient a distribution of flagella as seen for our triflagellar swimmer, and thus we advance to a study of an isotropic flagellar distribution with a quadriflagelar swimmer.

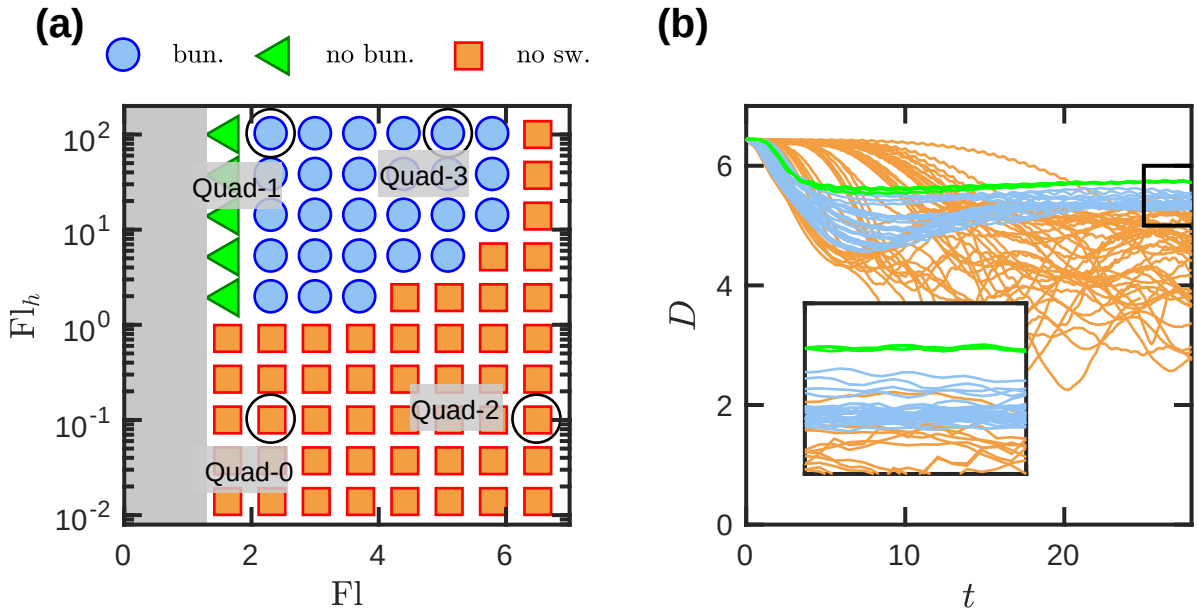

Figure 4.8: (a) Bifurcation diagram for quadriflagellar swimmer with flagellar anchors placed at the vertices of a regular tetrahedron – for each flagellum, $\lambda = 4$. Blue circles denote swimming with a stable bundle, green triangles denote no bundling, and orange squares denote ineffective swimming. Black circles denote the cases of (Fl, Fl_h) marked for study: Quad-0 (2.3, 0.1), Quad-1 (2.3, 100), Quad-2 (6.5,0.1), Quad-3 (5.0,100). (b) Flagellar distance from body D vs. time t. The inset shows the demarcation in D for the bundling (blue) and non-bundling (green).

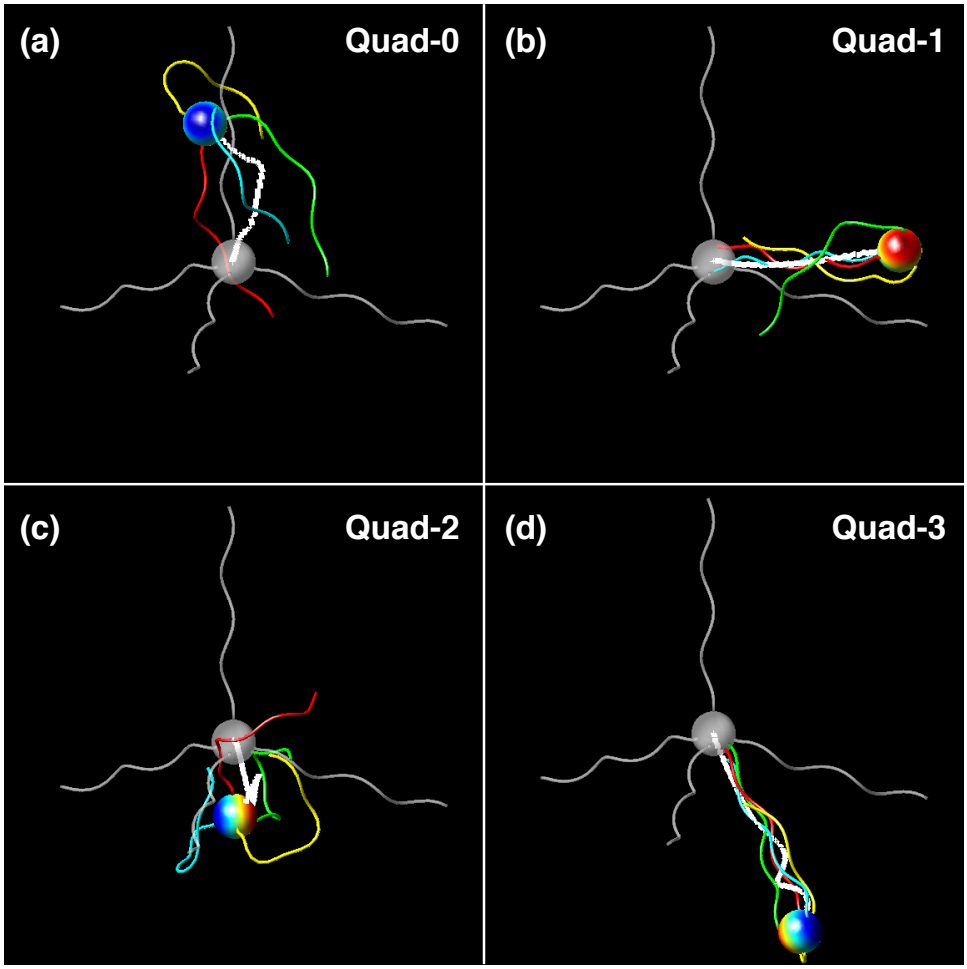

Figure 4.9: Snapshots of quadriflagellar swimmer at $t = 0$ (gray) and $t = 28$ (color). The flexibilities (Fl, Fl_h) for each case are: Tet-0 (2.3, 0.10), Tet-1 (2.3, 100), Tet-2 (6.5,0.1), Tet-3 (5.0,100). White lines follow the body center $\mathbf{x}_b$ over 28 time units.

4.6 Quadriflagellar swimming

We present here our quadriflagellar swimmer with an isotropic distribution of flagellar motors as a simple representation of a peritrichous swimmer. The tetrahedral arrangement maximizes the separation between flagella. Simulation data is shown in Fig. 4.8. Here $D(t)$ shows the same general stratification as the bi- and triflagellar cases, so we again use the same labels to classify swimming. One difference here compared to the triflagellar case is that D for the stable bundled cases can be quite noisy (due to variations in the degree of bundling) while the unbundled cases are quite level. But like the triflagellar case before, the quadriflagellar swimmer is quite stable at high Fl_h, with the same approximate threshold of $\mathrm{Fl}_h = 1$. What we do not see here are many cases of swimming with no bundle, as nearly al points below $\mathrm{Fl}_h = 1$ show ineffective swimming. Thus for this quadriflagellar swimmer, swimmers *require* high hook flexibility to swim.

We first examine case "Quad-0" with (Fl, Fl_h) = (2.3, 0.10), the stiff parameter set yielding stable swimming for Uni-0. From the trajectory snapshot in Fig. 4.9(a), we see that the swimmer's displacement from the origin over 28 time units is comparable to Uni-0. However, the body's trajectory is not straight at all, and furthermore, we see no evidence of bundling. While in the bi- and triflagellar cases the splayed flagella can push the cell body in one direction, this is not the case here due to the isotropic flagellar arrangement. While the body's rotation may attempt to bring the flagella closer together (a process partially augmented by hydrodynamic interactions), the very stiff hooks prevent any of the flagella from wrapping around the body. The stiff filaments further inhibit any flagellar deformations favorable to bundle formation. Without a bundle, it does not appear that the flagella generate propulsion in a consistent direction. We note that even at higher Fl as in case "Quad-2" with (Fl, Fl_h) = (6.5,0.1), increased flagellar deformation will not yield bundles without sufficient hook compliance, as illustrated in Fig. 4.9(c). These observations are quite a stark change from those we noted for Uni-0, where we stiff hooks and filaments are necessary for steady swimming.

We next examine case “Quad-1" with (Fl, Fl_h) = (2.3, 100), where the swimmer has stiff flagella and very weak hooks. Snapshots for this swimmer are shown in Fig. 4.9(b). As the body begins rotating, the hooks will bend, most to $\pi/2$ radians, given the lack of bending resistance, and allow sections of the flagella near the body to wrap around it. Hence, we see that the flagella will also slightly deform near the body as we saw in Uni-1. The flagella proceed to gather on the same side of the body and eventually form a loose bundle that propels the body in a relatively straight path, as shown by the end snapshot of Fig. 4.9(b). The stiffness of the flagella prevents further deformation of the flagella and thus prevents a tighter bundle from forming. Overall, we do not see the same slow-down in motion as we saw for Uni-1, and indeed find that for these flexibility parameters, the tetrahedral arrangement of flagella stabilizes swimming.

Last, we examine case “Quad-3" with (Fl, Fl_h) = (5.0,100), representing highly flexible flagella and hooks. The snapshot in Fig. 4.9(c) shows that for this case we once again recover stable swimming. With these flexibilities, the hooks readily buckle and the flagella easily wrap around the cell body and deform into a collective bundle. Fig. 4.9(d) reveals a tighter bundle for Quad-3 than for Quad-0. However, the flagella are not quite long enough (after gathering to the same side of the body) to physically entangle each other. One interesting feature is that the steady body trajectory is helical, albeit a much tighter one than that observed for the very wobbling case Uni-1.

To summarize, flexibilities allowing for stable uniflagellar swimming will not lead to stable swimming for this tetrahedral arrangement of flagella. Rather, stable quadriflagellar swimming occurs only for flexibilities where uniflagellar swimming is *un*stable. High hook compliance, i.e. high Fl_h is required to allow the flagella to deform from their starting isotropic arrangements and bundle to propel the cell in a consistent direction.

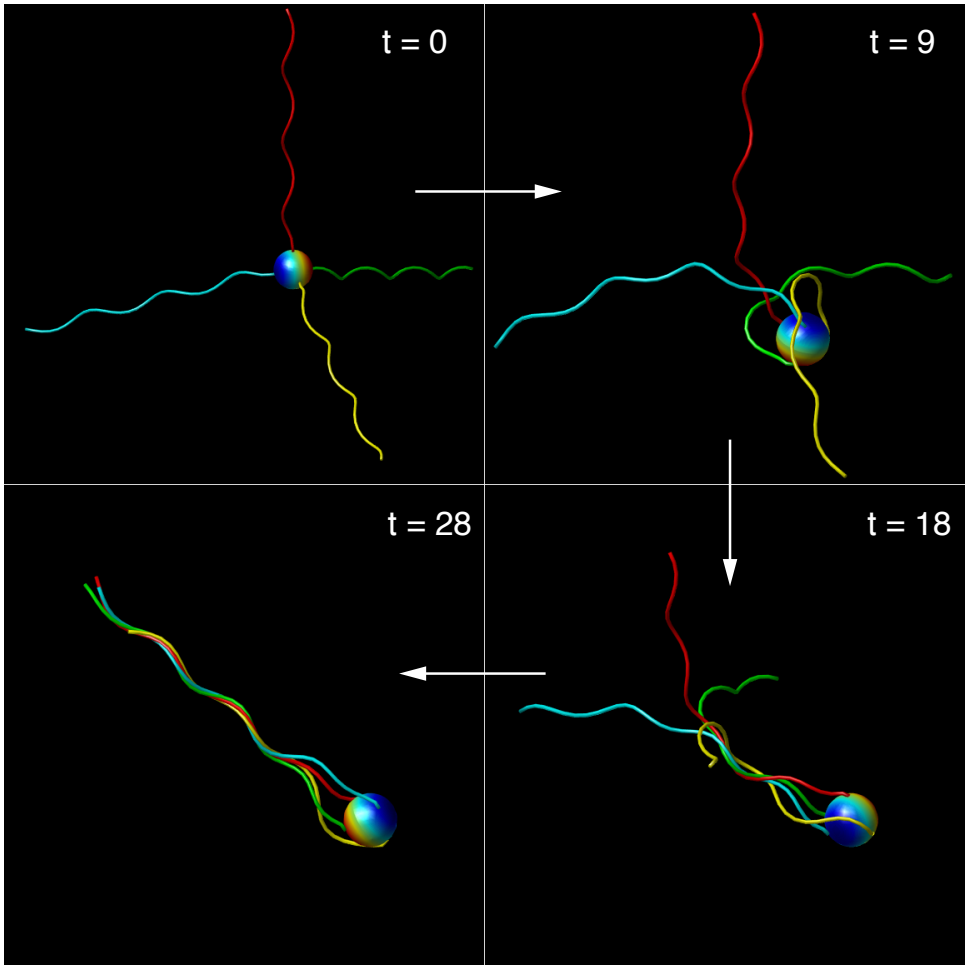

Figure 4.10: Time snapshots of the bundling for a quadriflagellar swimmer with $l = 12$. Note that the panel for $t = 0$ is zoomed out to show the full swimmer in its original isotropic configuration. Flexibilities are (Fl, Fl_h) = (6.5,14), and $L = 12$.

4.6.1 Tighter bundling

In the above analysis using the standard aspect ratio we have selected for the flagellar length relative to the body length, we see coherent bundle forms without the degree of physical entanglement required to form tight bundles. Thus we simply increase the flagellar length by 50% to seek changes in bundle formation. In Fig. 4.10, we show the time evolution of this longer-flagella swimmer for (Fl, Fl_h) = (6.5,14). We see that not only does the body rotation bring the flagella closer, but the longer lengths actually cause the flagella to entangle, as highlighted in the $t = 18$ panel, leading to a tight, full bundle as shown in the $t = 28$ panel. The overall process in Fig . 4.10 is similar to the “zipping and entanglement" effect described by Adhyapak et al. in Ref. [1].

4.6.2 Flagellar arrangement on body surface

To test the robustness of straight swimming against flagellar arrangement, we generate 30 samples of the placement of four flagellar motors by generating random points on the body (vectors on the unit sphere) and placing the motors there. In each sample, the flexibilities are fixed to the same values as Quad-3. In Fig. 4.11(a) we show D vs. t for the samples. With a few exceptions, most runs (33/40) show bundled swimming (blue lines). We show examples of these successful swimmers as snapshots in Fig 4.11(b-c). Overall, our data show that bundling and straight swimming is indeed robust against randomness, as we expect since the average net flagellar separation is smaller compared to the regular tetrahedron and should lead to easier bundling. More importantly, we have again shown that the presence of multiple flagella can stabilize against elastic buckling to produce better locomotion.

4.7 Multiflagellar bundles and swimming speed

In the final part of this chapter, we look at the effects of multiflagellarity on swimming speed. For this study, we arrange equally-spaced flagellar motors around a small circle on the body

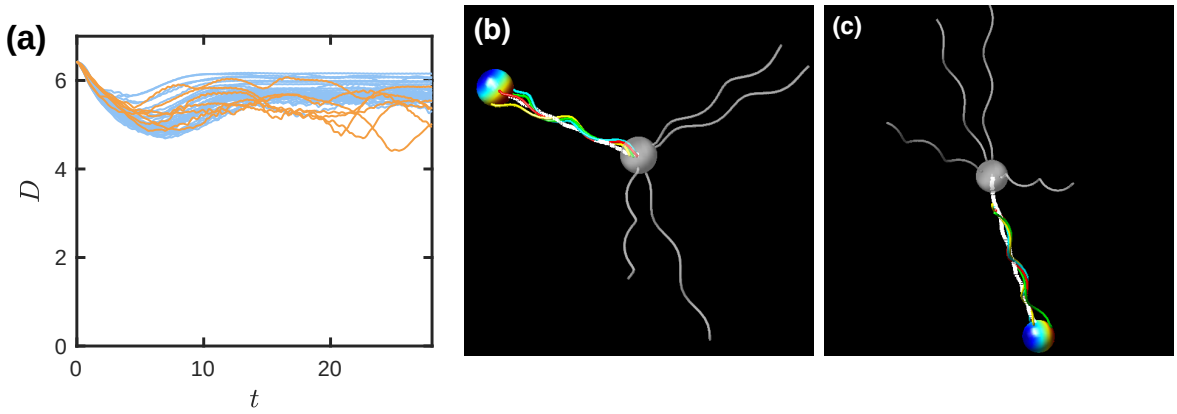

Figure 4.11: (a) Plots of D vs. t in for swimmers with four randomly sampled flagella. Blue lines indicate swimmers whose flagella successfully bundle and swim; orange lines denote otherwise. (b)(c) Snapshots of successful swimmers: $t = 0$ (gray) and $t = 28$ (color).

surface with solid angle $\varphi_{\max} = 0.52$ radians. This angle is small so the flagella readily form bundles. The flexibilities are $Fl = 2.5$ and $\mathrm{Fl}_h = 0.1$ so that uniflagellar swimming is stable. We see from the snapshots in Fig. 4.12(b) that the tripolar and quadripolar flagella interact in a rather tight bundle and the trajectories are straight. Fig. 4.12(a) shows the translational and rotational speeds (relative to Uni-0) as a function of the number of flagella. We note that the swimming speed does not increase proportionally with the number of flagella. However, the body rotation rate does increase almost linearly with the number of flagella. We find a similar speed saturation from from the "peritrichous" cases Uni-0, Bi-1, Tri-3, and Quad-3. These results are closely consistent with experimental data from Darnton *et al.* [26]. Hydrodynamic interactions do not explain this trend, as our polar simulations with the effect turned off, also shown in Fig. 4.12(a), indicates the speed saturates quickly with N. With two straightforward models, we can show that the relative speed levels off because the presence of more flagella adds drag to the overall swimmer.

4.7.1 Simple speed calculation

We follow the toy models presented by Nguyen et al. in Ref. [77] (linked rigid swimmers with no force-torque coupling, hydrodynamics, or sterics), and consider only 1D motion. Suppose

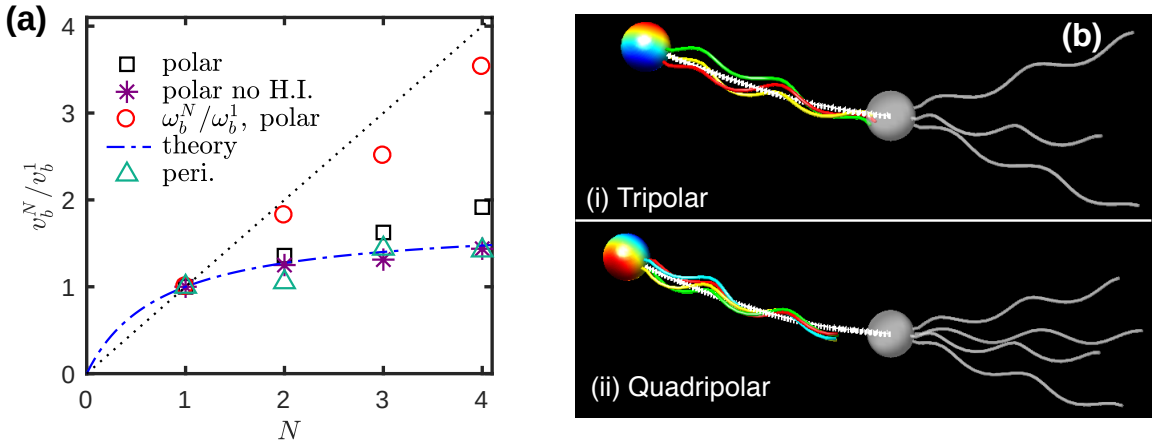

Figure 4.12: (a) Speed of body (normalized to uniflagellar value) vs. number of flagella. For cells with polar flagella ($\mathrm{Fl} = 2.5, \mathrm{Fl}_h = 0.1$), squares are translational speed, v_b^N/v_b^1, and circles rotational speed, ω_b^N/ω_b^1. Asterisks are v_b^N/v_b^1 calculated with no HI. Triangles are v_b^N/v_b^1 for the cases Uni-0, Bi-1, Tri-3 and Quad-3. The dashed black line is a linear trend and the solid blue line is Eq. 4.10. (b) Snapshots of swimming for a polar arrangement of (i) three flagella and (ii) four flagella with $t = 0$ in gray and $t = 28$ in color.

we have N independent flagella connected to the body. Each flagellum exerts a force $-F_p$ on the fluid, which in turn exerts a force $+F_p$ on the swimmer. To write the force balance on the entire swimmer, we assume the swimmer moves with constant speed v_b^N and balance the total propulsion with the total drag (from body and N flagella):

$$-\zeta_b v_b^N - N\zeta_f^{\parallel} v_b^N + NF_p = 0, \tag{4.9}$$

where $\zeta_f^{\parallel}$ is the translational drag on a flagellum moving along its axis. Solving for v_b^N and normalizing by the uniflagellar value v_b^1 yields an expression for relative speed:

$$\frac{v_b^N}{v_b^1} = \frac{N\left(\zeta_b + \zeta_f^{\parallel}\right)}{\zeta_b + N\zeta_f^{\parallel}}. \tag{4.10}$$

Estimating $\zeta_f^{\parallel}$ for our standard flagellum from RFT, the prediction from Eq. 4.10 shown by the blue line in Fig. 4.12(a) matches our simulation data quite well. We recognize in Eq. 4.10 that $v_b^N/v_b^1 < N$ and $v_b^N/v_b^1 \to$ constant as $N \to \infty$, meaning there are diminishing returns on speed with more flagella. Though HI may alter friction tensors, the qualitative result of

Eq. 4.10 will not change. To account for flagellar rotation and body counter-rotation in a real swimmer, we include a more detailed calculation of v_b^N/v_b^1 and the relative body rotation ω_b^N/ω_b^1 below.

4.7.2 More detailed speed calculation

we reference the full model presented by Nguyen et al. in Ref. [77] (rigid body and rigid helix with RFT resistance tensors but no H.I. or sterics). We again consider a 1D example where N independent flagella all push the body in the same direction at the same anchor. To generate this propulsion, a torque T is applied to each of the j flagella to rotate them on their helical axes, with a constraint force X_j maintaining connectivity with the body. Assuming that the entire swimmer translates with constant velocity v_b^N, we write (in mobility formulation) the resulting force and torque balances on body and flagella as the sum of external dynamics and hydrodynamic drag:

$$\begin{pmatrix} \sum X_j \\ -NT \end{pmatrix} - \begin{pmatrix} \zeta_b & 0 \\ 0 & \zeta_{b,r} \end{pmatrix} \begin{pmatrix} v_b^N \\ \omega_b^N \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \tag{4.11}$$

$$\begin{pmatrix} -X_j \\ T \end{pmatrix} - \begin{pmatrix} \zeta_f^{\parallel} & \psi_f \\ \psi_f & \zeta_{f,r}^{\parallel} \end{pmatrix} \begin{pmatrix} v_b^N \\ \omega_j^N \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \tag{4.12}$$

where ω_b^N is the body rotation speed, and we write Eq. 4.12 $\forall$ j with ω_j^N denoting the axial rotational speed, $\zeta_{f,r}^{\parallel}$ the axial rotation friction, and ψ_f the helical force-torque coupling. From Eq. 4.11 we can immediately write:

$$\omega_b^N = -\zeta_{b,r}^{-1} NT \;\; \rightarrow \;\; \frac{\omega_b^N}{\omega_b^1} = N. \tag{4.13}$$

We then solve the remaining linear system of equations for v_b^N, ω_j^N, and X_j. The velocity solutions are then:

$$v_b^N = -\frac{(\psi_f/\zeta_{f,r}^{\parallel})NT}{\zeta_b + N\left[\zeta_f^{\parallel} - \psi_f^2/\zeta_{f,r}^{\parallel}\right]}, \tag{4.14}$$

$$\omega_j^N = \frac{T}{\zeta_{f,r}^{\parallel}} \left[1 + \frac{N\psi_f^2/\zeta_{f,r}^{\parallel}}{\zeta_b + N\left[\zeta_f^{\parallel} - \psi_f^2/\zeta_{f,r}^{\parallel}\right]} \right]. \tag{4.15}$$

The body translation and flagellar rotation relative to the uniflagellar values v_b^1 and ω_b^1 are then:

$$\frac{v_b^N}{v_b^1} = N \frac{\zeta_b + \zeta_f^{\parallel} - \psi_f^2/\zeta_{f,r}^{\parallel}}{\zeta_b + N\left[\zeta_f^{\parallel} - \psi_f^2/\zeta_{f,r}^{\parallel}\right]} < N, \tag{4.16}$$

$$\frac{\omega_j^N}{\omega_j^1} = \left[1 - \frac{\psi_f^2/\zeta_{f,r}^{\parallel}}{\zeta_b + \zeta_f^{\parallel}} \right] \left[1 + \frac{N\psi_f^2/\zeta_{f,r}^{\parallel}}{\zeta_b + N\left[\zeta_f^{\parallel} - \psi_f^2/\zeta_{f,r}^{\parallel}\right]} \right]. \tag{4.17}$$

Because the flagellar mobility matrix is positive definite, $\zeta_f^{\parallel}\zeta_{f,r}^{\parallel} - \psi_f^2 > 0$ and the inequality in Eq. 4.16 always holds. We note that as $N \to \infty$, v_b^N/v_b^1 and ω_j^N/ω_j^1 become constant, meaning we see the same diminishing return on body speed as in Eq. 4.10, although $\omega_b^N \to \infty$. Using the same geometry as the previous section, the relative translation velocity has the same behavior as Eq. 4.10, but we additionally find that $\omega_b^N/\omega_b^1 = N$, consistent with our simulation results, also shown on Fig. 4.12(a).

4.8 Summary

We have developed a model swimmer with a rigid body and one or more elastic flagella. We characterize the stability of straight swimming in the parameter space of hook and flagellar flexibilities. In the uniflagellar case, straight swimming is stable only for a small range of low flexibilities. Modest increases in Fl_h lead to buckling of the hook and/or flagellum, and combined with higher Fl produce widely wobbling trajectories that are much slower than the straight case.

For the bi- and triflagellar cases, we see that swimming is possible at low Fl_h with no bundling and at high Fl_h with bundling. We note that particularly for the triflagellar swimmer, the stability region in flexibility parameter space is substantially larger than that for the uniflagellar case. For the isotropic quadriflagellar swimmer, we see swimming only at high Fl_h, and thus we see that the stability has reversed from that of the uniflagellar

swimmer, highlighting the reasons for the high hook compliance measured in peritrichous swimmers.

To further study the effects of flagellar placement, we test a swimmer with four flagella at maximum separation, finding that given a point in parameter space that produces straight swimming, bundling and swimming are indeed robust against flagellar placement, and we once again observe the stabilizing effects of multiple flagella. Finally, we find the swimming speed does not increase linearly with the number of flagella and present a simple theory whose predictions match the observations.

The work shown here offers insight into the biology and function of natural bacterial swimmers, and our results may contribute to the design of artificial swimming devices.

Chapter 5

Bacterial swimmer near wall

5.1 Introduction

As we stated in Chapter 1, studies of bacterial locomotion near boundaries are quite important. Swimming near walls can produce hydrodynamic interactions can lead to attraction to the wall, and subsequently swarming, and subsequently biofilm formation, with the latter being a practical issue dealt with across many engineering and biological fields. Thus in this chapter, we study the motion of a bacterial swimmer near a wall.

We begin by building upon the model developed in Chapter 4 for the free bacterial swimmer. First, we must define a wall in the 3D fluid domain. We then include wall effects to the mobility matrix capturing hydrodynamic interactions between all swimming components. To validate our model, we must satisfy all appropriate and significant near-wall interactions documented in experiment. Our end goal is to study how the multiflagellar swimming phenomena shown in Chapter 4 change with the addition of a wall.

5.2 Updated model schematic

The model swimmer we use in this chapter is shown in Fig. 5.1. While we still utilize a rigid body and elastic flagella as in previous chapters, note that we have also introduced

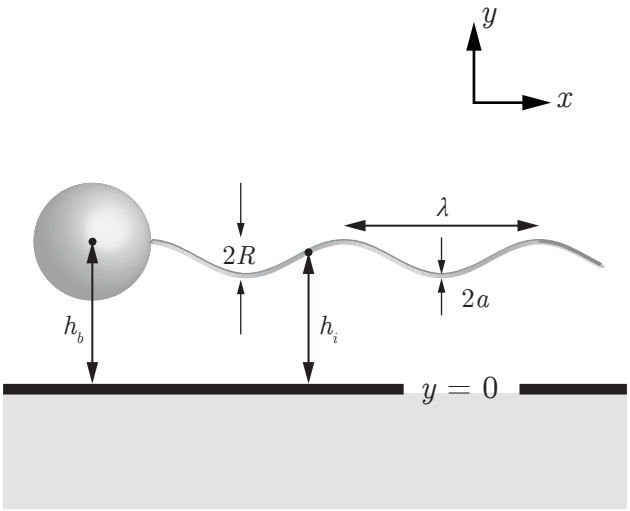

Figure 5.1: Depiction of a swimmer in a flow domain with an inifinite wall. The wall-normal direction is $\mathbf{y}$, with the wall plane located at $y = 0$ (marked by thick black line). The body is a rigid sphere of radius R_b with center located at a height h_b above the wall and each flagellar node at height h_i. The flagellar geometry is specified by radius R, pitch λ, filament radius a, and length L.

a wall into the flow domain corresponding to the plane $y = 0$. The now adopt the same geometry used for the tetrahedral swimmers at the end of Chapter 4, i.e. with $\lambda = 4$. The general equations of motion are the same as those used in previous chapters, with the main exception the addition of hydrodynamic interactions with the wall.

5.3 Hydrodynamic interactions with wall

Recall the mobility relation between kinematics and dynamics from Eq. 3.39: $\mathcal{V} = \mathcal{M} \cdot \mathcal{F}$

$$\mathcal{V} = \begin{bmatrix} \boldsymbol{\omega}_b \\ \mathbf{v}_b \\ \mathbf{v}_i \end{bmatrix}, \; \mathcal{M} = \left[\begin{array}{cc|c} \mathbf{M}_b^r & \mathbf{M}^\dagger & \mathbf{M}_{b,i}^r \\ \mathbf{M}^\ddagger & \mathbf{M}_b & \mathbf{M}_{b,i} \\ \hline \mathbf{M}_{i,b}^r & \mathbf{M}_{i,b} & \mathbf{M}_i \end{array}\right], \; \mathcal{F} = \begin{bmatrix} \mathbf{T}_b \\ \mathbf{F}_b \\ \mathbf{F}_i \end{bmatrix} \tag{5.1}$$

where $\mathcal{V}$ and $\mathcal{F}$ are the composite vectors of velocities and dynamics, and $\mathcal{M}$ is a grand mobility matrix specifying the hydrodynamic interactions between all swimming components. We define the matrices in $\mathcal{M}$ using singularity solutions and their corresponding images.

Each flagellar node is locally represented by its local drag tensor and the Stokeslet image system, and globally as a Stokeslet and its image system.

$$[\mathbf{M}_f]_{i,j} = \begin{cases} \boldsymbol{\zeta}_i^{-1} + \mathbf{S}^{\mathrm{IM}}(\mathbf{x}_i - \mathbf{x}_i^*; \mathbf{e}), & i = j \\ \mathbf{S}(\mathbf{x}_i - \mathbf{x}_j) + \mathbf{S}^{\mathrm{IM}}(\mathbf{x}_i - \mathbf{x}_j^*; \mathbf{e}), & i \neq j \end{cases} \tag{5.2}$$

We represent the body generated flow as a combination of a Stokeslet, a dipole, and their corresponding images so that the effect on the flagella is:

$$\mathbf{M}_{i,b}^r = \mathbf{R}(\mathbf{x}_{ib}) + \mathbf{R}^{\mathrm{IM}}(\mathbf{x}_{ib^*}; \mathbf{e}) \tag{5.3}$$

$$\mathbf{M}_{i,b} = \mathbf{S}(\mathbf{x}_{ib}) + \mathbf{S}^{\mathrm{IM}} - (R_b^2/6)\left[\mathbf{D}(\mathbf{x}_{ib}) + \mathbf{D}^{\mathrm{IM}}(\mathbf{x}_{ib^*}; \mathbf{e})\right] \tag{5.4}$$

The drag force and torque on the spherical body are calculated using Faxén's laws:

$$\mathbf{F}_b^{\mathrm{D}} = \zeta_b \left[-\mathbf{v}_b + \left(1 + \frac{R_b^2}{6}\nabla^2\right) \mathbf{v}_\infty^f \Big|_{\mathbf{x}_b} \right], \tag{5.5}$$

$$\mathbf{T}_b^{\mathrm{D}} = \zeta_b^r \left[-\boldsymbol{\omega}_b + \frac{1}{2}\left(\nabla \times \mathbf{v}_\infty^f\right)\Big|_{x_b} \right]. \tag{5.6}$$

The velocity fields affecting the body motion arise from the flagellar Stokeslets, their images, *and* the image systems for the body. We first focus on the self-mobility of the body and the rotation-translation coupling due to the body images:

$$\mathbf{M}_{b,r} = \zeta_{b,r}^{-1}\boldsymbol{\delta} + \frac{1}{2}\nabla \times \mathbf{R}^{\mathrm{IM}}(\mathbf{x} - \mathbf{x}_b^*; \mathbf{e}) \tag{5.7a}$$

$$\mathbf{M}^\dagger = \frac{1}{2}\nabla \times \left[\mathbf{S}^{\mathrm{IM}}(\mathbf{x} - \mathbf{x}_b^*) - (R_b^2/6)\mathbf{D}^{\mathrm{IM}}(\mathbf{x} - \mathbf{x}_b^*; \mathbf{e})\right] \tag{5.7b}$$

$$\mathbf{M}_b = \zeta_b^{-1}\boldsymbol{\delta} + \left[1 + (R_b^2/6)\nabla^2\right]\left[\mathbf{S}^{\mathrm{IM}}(\mathbf{x} - \mathbf{x}_b^*; \mathbf{e}) - (R_b^2/6)\mathbf{D}^{\mathrm{IM}}(\mathbf{x} - \mathbf{x}_b^*; \mathbf{e})\right] \tag{5.7c}$$

$$\mathbf{M}^\ddagger = \left[1 + (R_b^2/6)\nabla^2\right]\mathbf{R}^{\mathrm{IM}}(\mathbf{x} - \mathbf{x}_b^*; \mathbf{e}) \tag{5.7d}$$

Eqs. 5.7a to 5.7d are evaluated at the body center with $r = 2h_b \geq 2$, and so we make a number of simplifications. mainly neglecting terms of $\mathcal{O}(h_b^{-4})$ following [Kim and Karrila]. In Eq. 5.7b, $\nabla \times \mathbf{S}^{\mathrm{IM}}(\mathbf{x} - \mathbf{x}_b^*)$ is exactly zero at $\mathbf{x}_b$, and thus the leading order correction to the body rotation coming from the $\nabla \times \mathbf{D}^{\mathrm{IM}}$ is $\mathcal{O}(h_b^{-4})$. We assume the weak rolling induced by the wall is dwarfed by the torque applied on the body, which is order 1 when scaled. In Eqs. 5.7c and 5.7d, $\mathbf{D}^{\mathrm{IM}}$ and $\mathbf{R}^{\mathrm{IM}}$ are already $\mathcal{O}(h_b^{-3})$, so we omit calculating the Laplacian of these terms. The Laplacian of $\mathbf{S}^{\mathrm{IM}}$ only requires the additional calculation of $\nabla^2\mathbf{SD}$ since $\nabla^2\mathbf{S} = -\mathbf{D}$ and $\nabla^2\mathbf{D} = \mathbf{0}$.

$$\begin{aligned}\nabla^2\mathbf{SD} =&(\mathbf{x}\cdot\mathbf{e})\left[(4H_2'/r + H_2'' - D_2)\boldsymbol{\delta} - (D_2'/r + D_2''/2)\mathbf{xx}\right] \\ &+ (4H_2'/r + H_2'' - D_2)\mathbf{xe} - (D_2 + 2D_1'/r + D_1''/2 + 2H_2'' + 8H_2'/r)\mathbf{ex}\end{aligned} \tag{5.8}$$

Lastly, we turn our attention to the contributions from flagella on body motion:

$$\mathbf{M}_{b,i}^r = \frac{1}{2}\nabla \times \left[\mathbf{S}(\mathbf{x} - \mathbf{x}_i) + \mathbf{S}^{\mathrm{IM}}(\mathbf{x} - \mathbf{x}_i^*; \mathbf{e})\right] \tag{5.9}$$

$$\mathbf{M}_{b,i} = \left[1 + (R_b^2/6)\nabla^2\right]\left[\mathbf{S}(\mathbf{x} - \mathbf{x}_i) + \mathbf{S}^{\mathrm{IM}}(\mathbf{x} - \mathbf{x}_i^*; \mathbf{e})\right] \tag{5.10}$$

In Eqs. 5.9 and 5.10, we keep terms of $\mathcal{O}(r^{-3})$ meaning we need only the curl of $\mathbf{S}$, $\mathbf{SD}$, and the Laplacian of $\mathbf{S}$.

$$\nabla \times \mathbf{S} = \left(\frac{H_1'}{r} - H_2\right)[\mathbf{x}] \tag{5.11}$$

$$\begin{aligned}\nabla \times \mathbf{SD} =&\frac{[\mathbf{e}]}{2}(2H_2\boldsymbol{\delta} - D_2\mathbf{xx}) + (\mathbf{x}\cdot\mathbf{e})\left(H_2'/r + D_2/2\right)[\mathbf{x}] \\ &\frac{1}{2r}\left(D_1' + 4H_2'\right)(\mathbf{e}\times\mathbf{x})\mathbf{x}^{\mathrm{T}} + \frac{1}{2}(D_1 + 4H_2)[\mathbf{e}]\end{aligned} \tag{5.12}$$

One final remark on hydrodynamics is that we do not have lubrication forces anywhere on our swimmer. We may consider implementing wall repulsive forces using the same Lennard-Jones potential as introduced in Chapter 3 to prevent collisions with the wall. However, we do not that in real systems, asperities on the surface of the swimmer as well as wall roughness will inevitably result in wall collisions for certain cases.

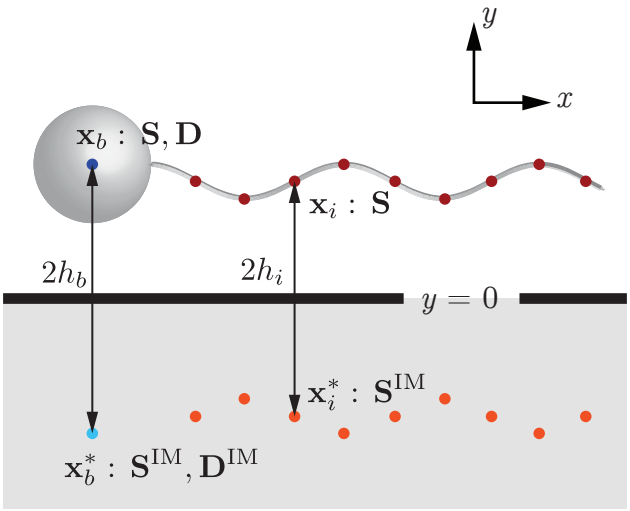

Figure 5.2: Hydrodynamics of a bacterial swimmer represented as a series of flow singularities in the flow domain and their corresponding image systems. For each flagellar node (dark red dots), we place a regularized Stokeslet $\mathbf{S}$ at the node position $\mathbf{x}_i$ and the image system $\mathbf{S}^{\mathrm{IM}}$ at the image point $\mathbf{x}_i^*$ (light red point). For the body, we place a Stokeslet and dipole $\mathbf{D}$ at the body center $\mathbf{x}_b$ (dark blue dot) and the dipole image system $\mathbf{D}^{\mathrm{IM}}$ along with $\mathbf{S}^{\mathrm{IM}}$ at the image point $\mathbf{x}_b$ (light blue point).

5.4 Model validation

The first check is that the swimming dynamics must be the same as that of the free swimmer in Chapter 4 when the wall distance becomes very large. As of writing this chapter, With the current formulation of the code shown in Appendix D, the linear swimming speed comes within 10% of the free swimming speed. [There are more modifications to be made]. The next steps are to verify key phenomena associated with near wall swimming. One easy item to check is that a cell swimming parallel to the wall and pushed by a right-handed helix will turn to the left. Thus in our case, the swimmer initialized to swim in the $-\hat{x}$ direction should start swimming toward the $+\hat{z}$ direction. The second item to check is that a pusher in any initial arbitrary orientation relative to the wall will always align to swim parallel to the wall. The final item, related to both aforementioned phenomena, is to ensure that there are indeed regions of parameter space Fl, Fl_h, $h_{b,0}$ that will yield stable (circular) swimming at a constant height above the wall.

We begin with the first check, the turning of the cell, using simulations of a uniflagellar

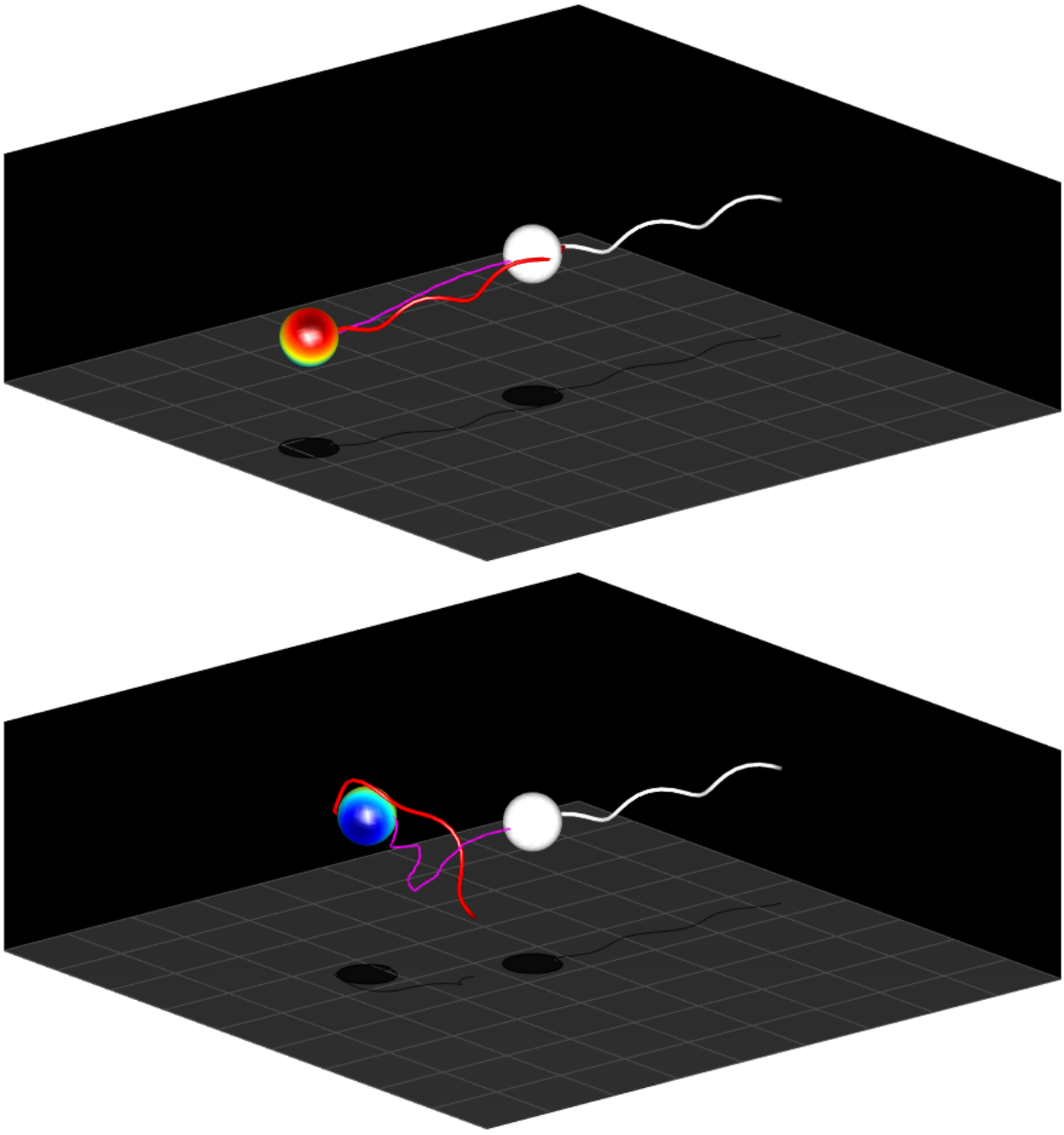

Figure 5.3: Snapshots of a uniflagellar swimmer initially at height $h_{b,0} = 5$ above the wall at $t = 0$ in white and $t = 42$ in color. The magenta line indicates the trajectory of the body center of mass. The parameters are (Top) $\mathrm{Fl} = 2.5$, $\mathrm{Fl}_h = 0.1$. (Bottom) $\mathrm{Fl} = 2.5$, $\mathrm{Fl}_h = 0.5$

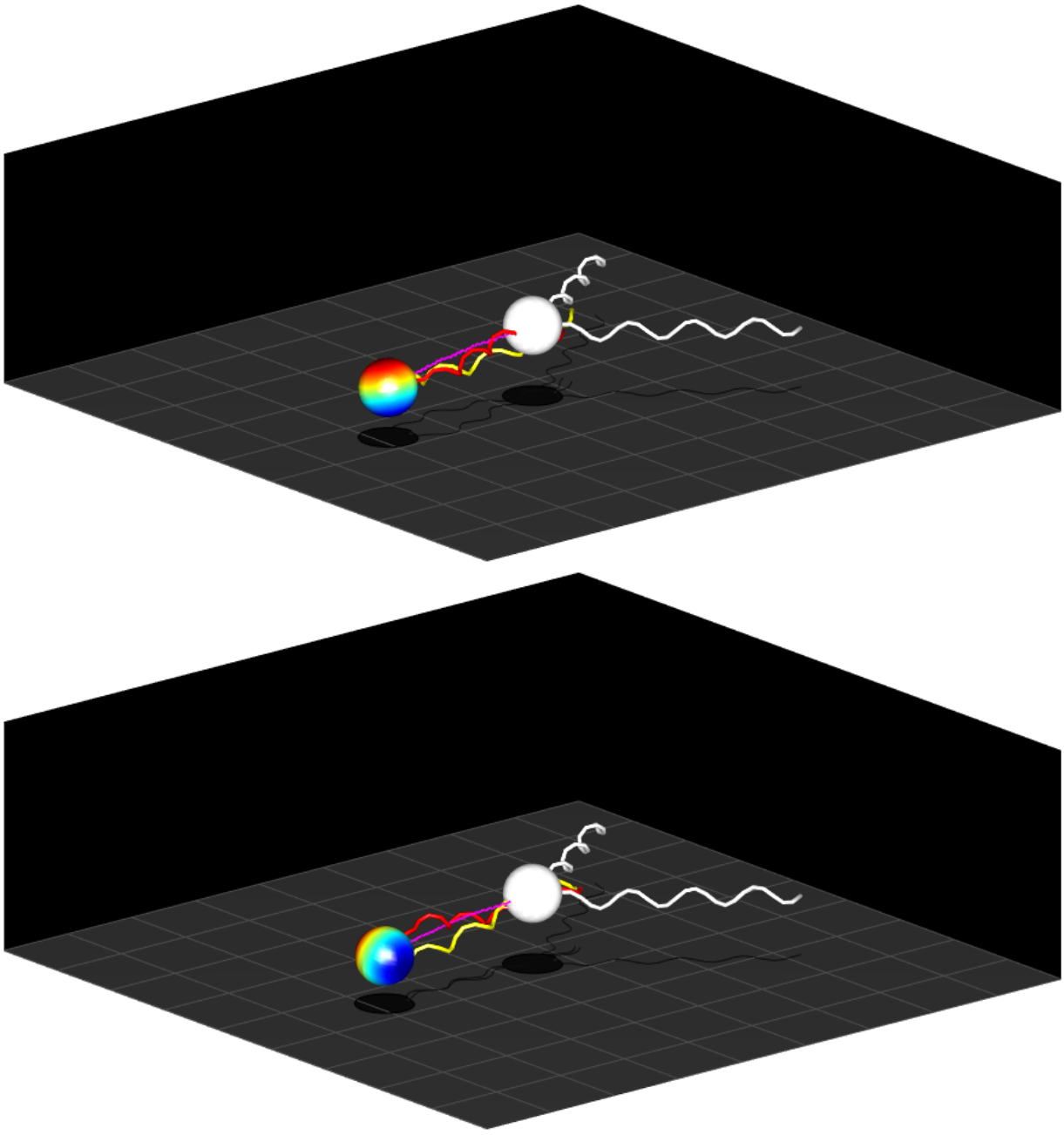

Figure 5.4: Snapshots of a biflagellar swimmer initially at height $h_{b,0} = 2.5$ above the wall at $t = 0$ in white and $t = 21$ in color. The magenta line indicates the trajectory of the body center of mass. The parameters are (Top) $\mathrm{Fl} = 2.5$, $\mathrm{Fl}_h = 0.1$. (Bottom) $\mathrm{Fl} = 2.8$, $\mathrm{Fl}_h = 0.58$

swimmer. Fig. 5.3 shows trajectory snapshots for (Fl, Fl_h) = (2.5,0.1) and (2.5,0.5) over 42 time units, where the latter parameter set is corresponds to our stable free uniflagellar swimmer as in Chapter 4. In this first case, we note that the trajectory does seem to veer in the proper turning direction before turning the opposite direction. We also note that the swimmer descends closer toward the wall, but does not seem to collide into it over the simulation timescale that we test. In the second case, the trajectory does not stay straight for very long, and the flagellum seems rather contorted around the cell as the body erratically moves up away from the wall. Note that for the parameters we used here, we are not able to induce pure hook buckling – in Chapter 4, we saw that at Fl = 2.5, the bending of the hook will also induce (unstable) bending of the flagellum as well. This may not be the case should we run the test with higher flagellar multiplicity, as in Ref. [119]. We may also elect to use much lower stiffnesses, as authors in Ref. [96] showed that a rigid helix and bent hook can produce circular paths, albeit with a very large radius of curvature relative to the body size. However, dropping Fl too low may make our equations to stiff to solve on a reasonable time scale. We also ran test simulations with a biflagellar swimmer slightly closer to the wall, $h_{b,0} = 2.5$ with similar flexibility parameters, (Fl, Fl_h) = (2.5,0.1) and (2.8,0.58), as shown in Fig. 5.4. Though the trajectories seem to turn in the proper direction at the end of the simulation without bending the flagella, whether this turn eventually leads to a circular path remains to be seen with longer-run simulations. Validation of this phenomenon is still ongoing.

We continue to the second test, that a pusher oriented at an arbitrary angle will realign to swim parallel to the wall. We do test runs with both uni-and bi-flagellar swimmers. Our initial test simulations use very stiff parameters, Fl = 2.5, $\mathrm{Fl}_h = 0.1$, with the initial swimming direction tilted about 0.52 radians from the plane $y = 0$. Fig. 5.5 shows snapshots for the swimmers tested. We note that in both cases, the body appears to be quite close to crashing into the wall after 21 time units without any evidence of reorientation in the overall swimmer. We would expect that even with stiff hooks not allowing much bending

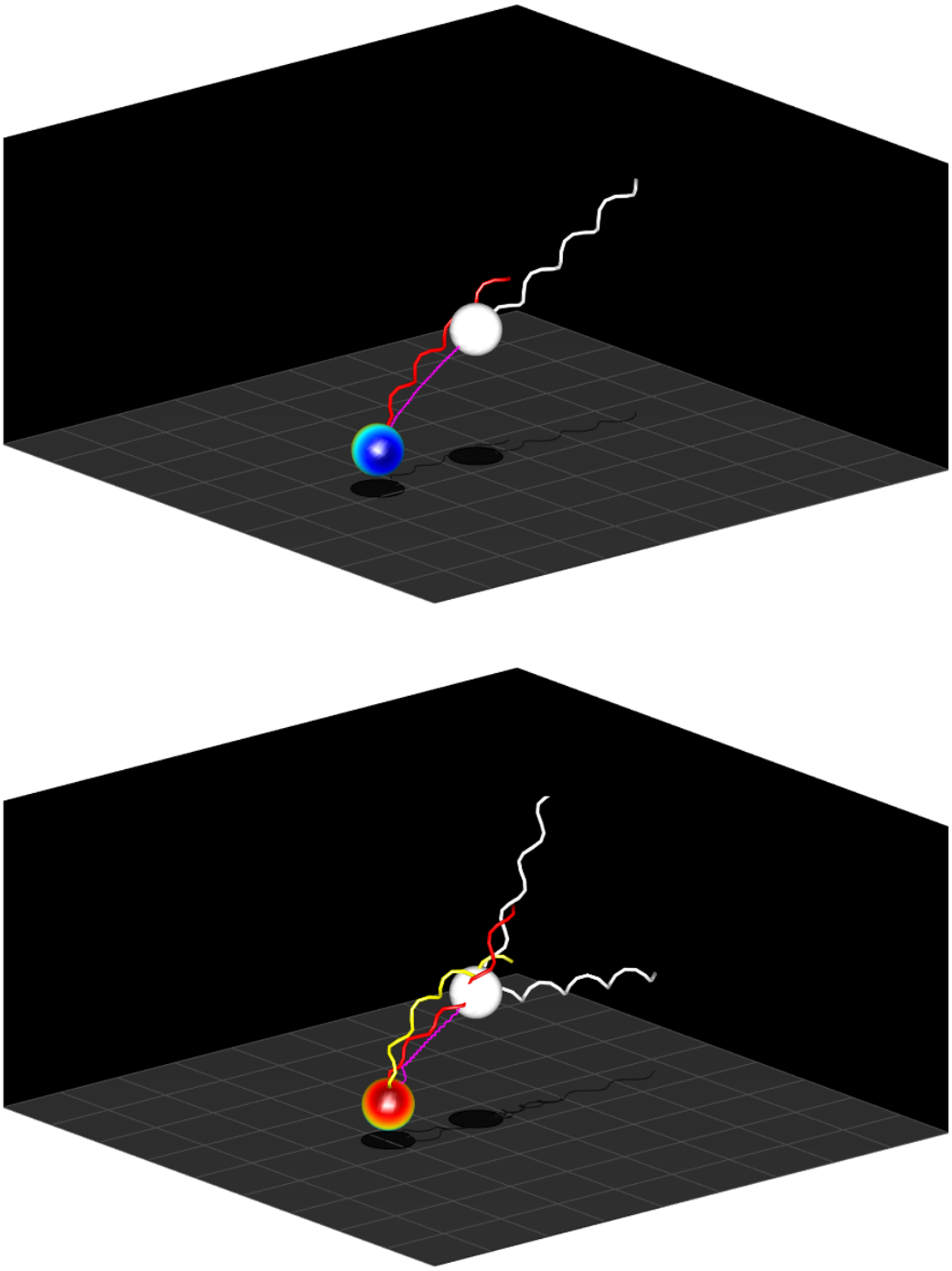

Figure 5.5: Snapshots of a uniflagellar (top) and biflagellar (bottom) swimmer initially at height $h_{b,0} = 5$ above the wall at $t = 0$ in white and $t = 21$ in color. The magenta line indicates the trajectory of the body center of mass. The parameters are $\mathrm{Fl} = 2.5$, $\mathrm{Fl}_h = 0.1$, and the swimming direction is initially oriented about 0.52 radians from the plane $y = 0$. The separation angle for the biflagellar swimmer is $\varphi = 1.1$ radians.

in the swimmer, that the hydrodynamic interactions would be strong enough to just rotate the entire swimmer – however, at this stage of testing, this does not seem to be the case. It may also be that our swimmers were initialized too close to the wall, and did not have enough time to reorient. We note that the drag required to rotate the swimmer orthogonal to its swimming orientation is quite large, and thus the larger time requirement. Validation of this phenomenon is still ongoing.

As for the final test, swimming at a constant distance above the wall, the current results shown in Figs. 5.3, 5.4, and 5.5 are inconclusive. For those simulations where significant elastic buckling has not occured, we see a clear trend toward the wall, but over the simulation times stated, this trend is weak, at best. We note from Ref. [96] that the ability to swim stably above the wall is a rather complicated function of swimmer geometry, configuration, initial height above the wall, hook flexibility, etc., so the preliminary results shown here must be expanded in order to obtain a clearer picture of stable swimming. Validation of this phenomenon is ongoing, and perhaps may be better accomplished with a different swimmer, such as one with multiple flagella to stabilize against elastic instabilities that may arise due to interactions with the wall.

5.5 Multiflagellar swimmers near wall

In this section we detail our initial attempts to understand the relationship between multiflagellarity and wall attraction. We study tri- and quadriflagellar swimmers with the same flexibilities as the tetrahedral swimmer presented in Chapter 4, Fl = 2.5, $\mathrm{Fl}_h = 1$ so that swimming should be robust against orientation. Simulations here are run to 28 time units.

5.5.1 Attraction or escape

To conduct our first sampling study, we initialize our cell body at a height $h_{b,0} = L$ above the wall so that anchor points can be sampled evenly around the surface of the sphere, i.e.

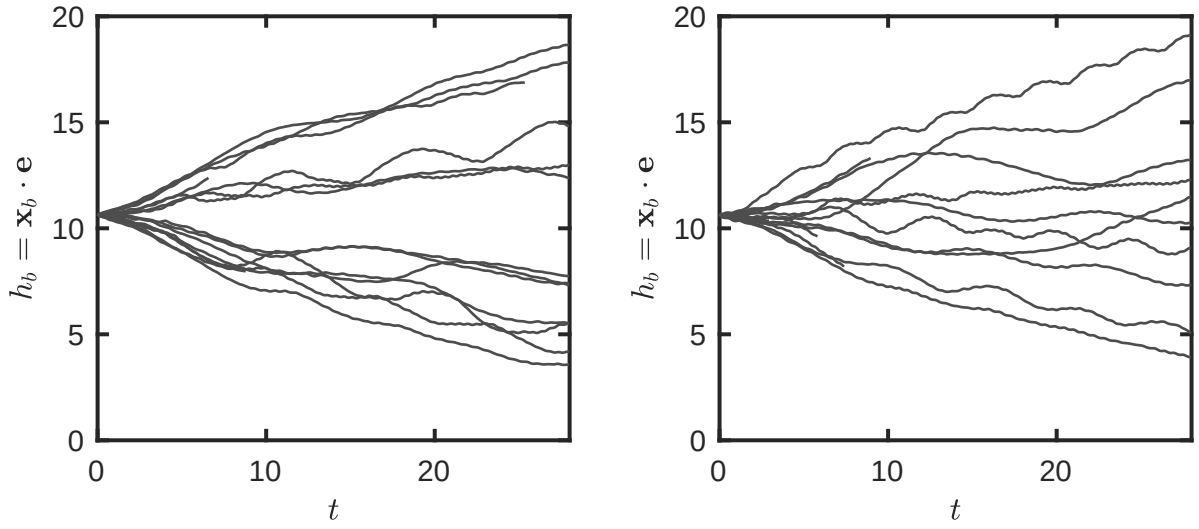

Figure 5.6: Body height h_b vs. time t for samples of (left) triflagellar swimmers and (right) quadriflagellar swimmers over 28 time units. Lines that terminate before $t = 28$ represent failed simulations, usually due to large steric interactions or penetration through the wall.

we wish to avoid initializing flagella whose free ends are below the wall. Fig. 5.6 shows the heights of each sample swimmer above the wall as a function of time. At first glance, both the tri- and quadriflagellar samples show an even split between swimmers moving away from the wall and swimmers moving away from the wall. This is not a terribly surprising result, as we expect equal chances for bundles to form pointing toward and away from the wall. However, we do note that that in each case, there are samples that only seem weakly attracted to or repelled by the wall. In these cases, bundles form and generate propulsion parallel to the wall. Snapshots of all the aforementioned modes of motion are found in Fig. 5.7 for the triflagellar swimmer and Fig. 5.8 for the quadriflagellar swimmer. We note that the Fig. 5.8(right) seems to demonstrate the best example of swimming above the wall at constant height we have obtained thus far. As a last remark, we point out that the random sampling done for this section may not be the most useful tool in formulating a full understanding of wall effects on multiflagellar swimming. It may be better to stick to a fixed distribution of flagella on the body surface, e.g. tetrahedron, and sample a wide range of initial placements / rotations of that swimmer above the wall.

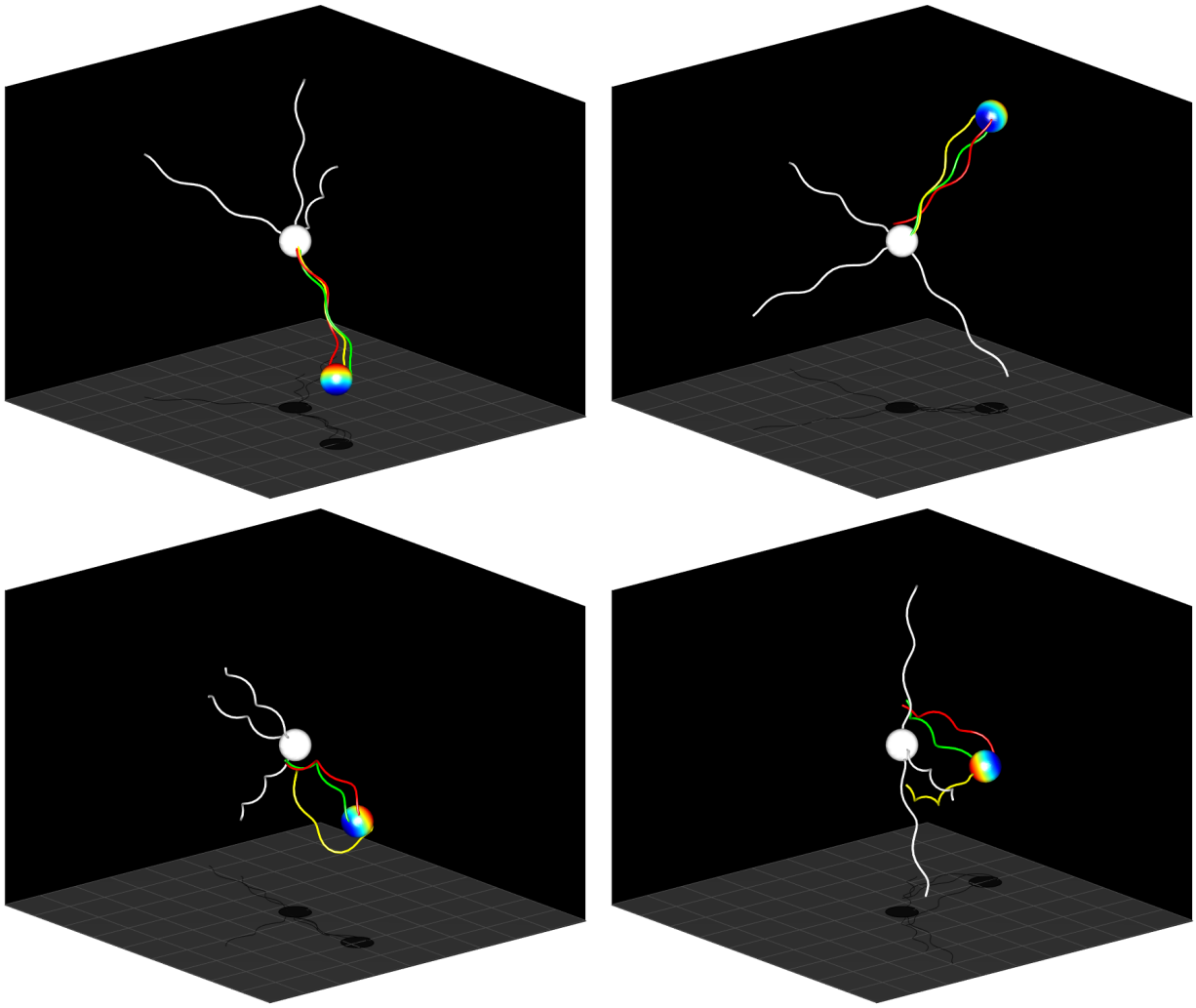

Figure 5.7: Snapshots of triflagellar swimmmer at $t = 0$ in white and $t = 28$ in full color. Orientations were sampled randomly in each case. (Top left) Swimming towards the wall. (Top right) swimming away from wall. (Bottom row) Very weak attraction toward the wall.

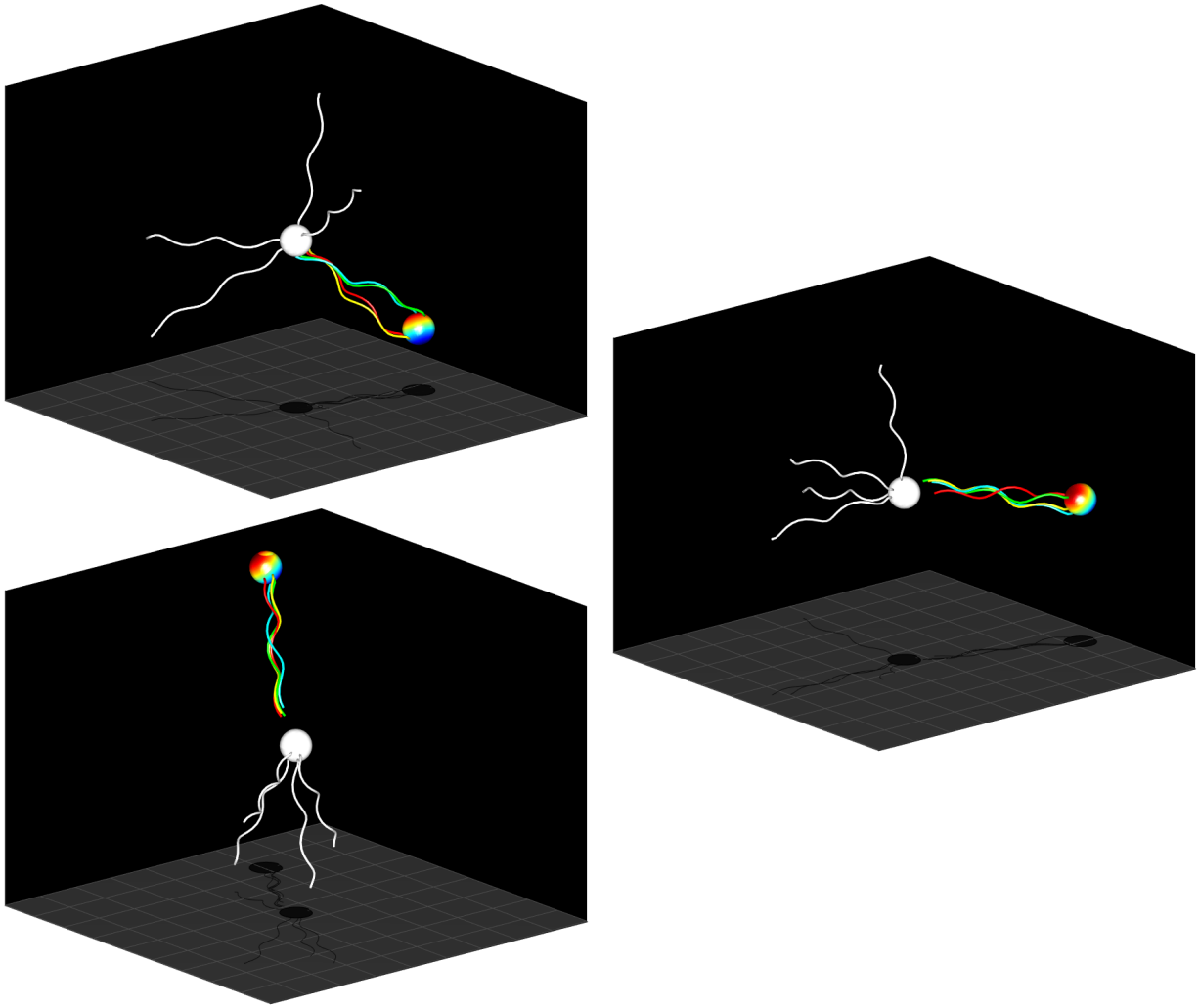

Figure 5.8: Snapshots of triflagellar swimmmer at $t = 0$ in white and $t = 28$ in full color. Orientations were sampled randomly in each case. (Top left) Swimming towards the wall. (Bottom left) swimming away from wall. (Right) Very weak attraction toward the wall. Shadows show 2D projections on wall.

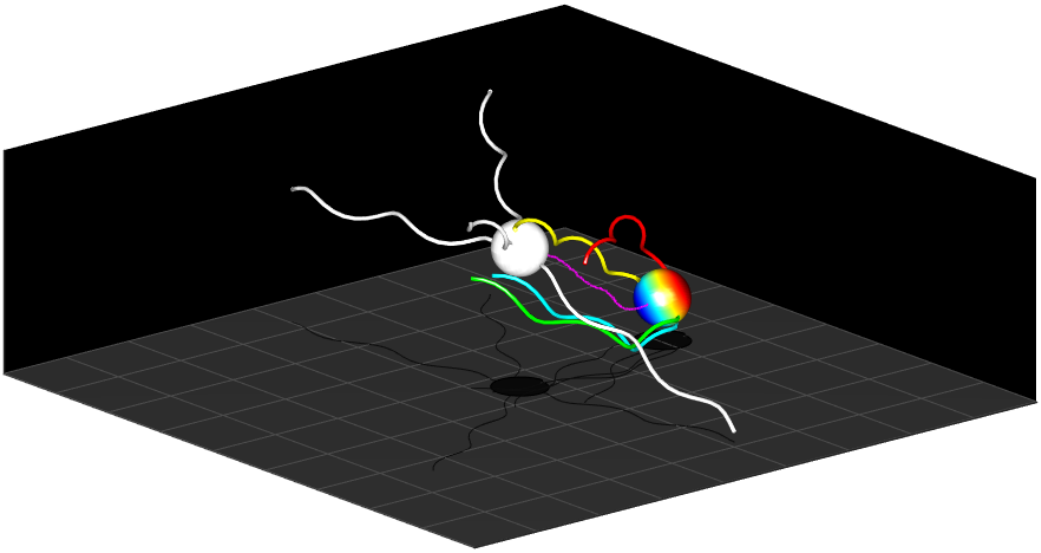

Figure 5.9: Snapshots of a quadriflagellar swimmer with randomized flagellar motors initially at height $h_{b,0} = 5$ above the wall at $t = 0$ in white and $t = 21$ in color. The magenta line indicates the trajectory of the body center of mass. The flexibilities are $\mathrm{Fl} = 2.5$, $\mathrm{Fl}_h = 1$.

5.5.2 Wall effects on bundling

We may also elect to set a lower initial height for our multiflagellar swimmer to better grasp how near-wall effects affect the bundling process. However, one potential issue is that the closer the swimmer is to the wall, the narrower the range of possible configurations we may sample. For instance, if the body center were placed at the wall, we can only sample half of all possible configurations, as the other half would place the flagella in the wall interior rather than the flow domain. We show one example of a successful sample in Fig. 5.9, where we note that this sample swimmer its flagella more or less parallel to the wall initially. When the simulation begins, the flagella will attempt to gather and bundle, but they must shift to the same side of the body without touching the wall. The end snapshot in 5.9 shows that two of the flagella (green and blue) come very close to colliding with the wall, as does the cell body, and the flagella seem to be splayed rather than bundled. To elucidate the real effects of the wall, we must run this same sample in open domain and compare the bundle formation there with this case.

5.6 Summary

Starting from the general model developed in Chapter 3, we have incorporated the necessary hydrodynamics to utilize this model in a fluid domain bounded by an infinite wall. Our initial fidelity tests with uni- and biflagellar swimmers, though not completely successful, show much promise in verifying the key phenomena associated with near-wall swimming of bacteria. We have also begun initial studies probing the hydrodynamic effects of the wall on multi-flagellar swimming and bundling, but there is much work left to be done in this area.

Chapter 6

Biohybrid swimmer

6.1 Introduction

In the previous chapters, we have dealt with modeling the motion of a simple prokaryotic helical flagellum driven by a single motor. However, more complex mechanisms for the actuation of flagella do indeed exist in nature. For instance, a mammalian sperm cell, a large eukaryote, have motors located along the length of the flagellar filament, usually within a flagellar cross-section, that generate relatively large amplitude bending waves that lead to faster propulsion that bacteria [65].

The examples above have provided the foundation for the research and fabrication of artificial swimmers. In particular, the simple helical structure of the various microswimmers mentioned in Chapter 1. For instance, the natural flagella themselves can be sheared from cells, reconstituted onto different substrates such as magnetic nanoparticles, and used to augment motion as in Ref [4]. Another often used swimmer design utilizes a metal helix that can be controlled by an external mechanism, usually a magnetic field as in Refs. [74, 126]. One note on the class of swimmers mentioned here is that they all require a form of external control, and likely need an operator. While this may be the desired mode of operation for many applications, particularly in biomedical settings, the use of fully autonomous swimmers may potentially expand the range of these applications even more.

Notable studies of fully self-propelled swimmers have utilized biological components as the “batteries" driving the motion of artificial objects, hence the biohybrid label. For instance, authors of Refs. [19, 122] directly graft beating cardiomyocytes onto soft flexible substrates, thus generating biohybrid flagella that generate propulsion via bending waves. The cardiomyocytes can be grafted along the entire filament length, as in Ref. [19], or can be localized at particular points, as in Ref. [122] – motion will occur as long as the swimming stroke can overcome kinematic reversibility. For the latter free swimmer from Williams et al. [122], the authors matched their experimental speed and trajectory measurements to a theoretical model derived from a simple 1D 4th order beam equation:

$$(EI)y_{xxxx} + \zeta_{\perp} y_t = M_{xx}, \tag{6.1}$$

where y measures the deflection normal to the filament axis, and M is the bending moment actuating the filament. The authors in the aforementioned study assume bending deflections to be very small in both experiment and model, leading to very slow propulsion, with translation amounting to less that a few percent of the body length in a run. We also note that the despite swimming at speeds comparable to bacteria, the biohybrid of Ref. [122] is much larger due to the fact that the swimmer must accommodate micron size cardiomyocytes. Generally speaking, cardiomyocytes also have other attributes that make them desirable batteries for biohybrid swimmers. For instance the beating cells are able to communicate and synchronize, particularly in response to environmental stimuli such as elastic deformations [6, 21, 22]. Furthermore, these attributes can be exploited so that the cardiomyocytes beat at specific, tunable frequencies [78], and yield better overall control.

In this chapter, we develop a model for a swimming biohybrid analogous to the one introduced in Ref. [122] by adapting the bacteria model introduced in Chapter 3 for the capability of analyzing full 3D motion. Whereas Ref. [122] focus on optimizing the swimmer geometry for a given (low-amplitude) driving mechanism, we fix our geometry and focus

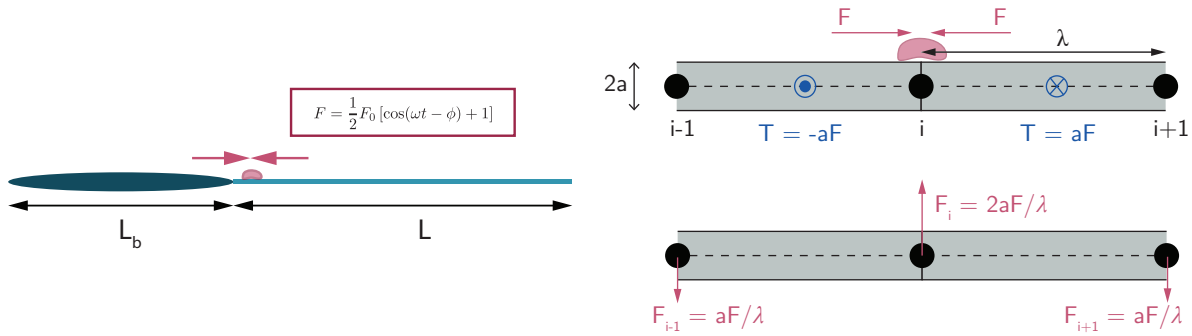

Figure 6.1: (a) Model of biohybrid swimmer with oscillatory beat places on flagellar filament. L_b is the body length and L is the tail length (b) (i) Closer view showing the placement and dynamics of a beat at a bending node. Torques on adjacent edges of node i are shown by blue circles. (ii) The induced moments are resolved as forces on adjacent nodes, denoted by pink arrows.

on optimizing the parameters of the driving mechanism. Because our model was created to allow an arbitrary number of tails, we also investigate the effects of a second tail on the motion of the biohybrid.

6.2 Updated model schematic

6.2.1 Physical description

The swimmer consists of a rigid body, prolate spheroid of length L_b, connected to an elastic flagellum, naturally straight filament of length L with circular cross-section of radius a and bending modulus EI. A beating cell is placed on the filament. The magnitude F of the contractile force exerted by the beating cell is:

$$F = \frac{1}{2}F_0\,[\cos(\omega t - \phi) + 1] \tag{6.2}$$

where F_0 is a constant amplitude, t time, ω the actuation frequency, and ϕ a phase. The constant 1 assures that F is always nonnegative. A bending length scale, L_ω arises from the actuation of the filament:

$$L_\omega = \left(\frac{EI}{\bar{\zeta}\omega}\right)^{1/4} \tag{6.3}$$

Taking L_ω with our remaining parameters, we characterize our swimmer with the following dimensionless groups:

$$\mathrm{Fl} = \frac{T}{EI/L_f} \tag{6.4}$$

$$\mathrm{Sp} = \frac{L}{L_\omega} \tag{6.5}$$

$$Z = \frac{\zeta_b}{\bar{\zeta} L_f} \tag{6.6}$$

$$V = \frac{v}{L_\omega \omega}; V = \frac{v}{T/\zeta_b L_b} \tag{6.7}$$

$$R = \frac{L_f}{L} \tag{6.8}$$

The flexibility number, Fl measures the magnitude of the actuation moment relative to the buckling moment of the flagellum. The sperm number, Sp, is an relates the swimmer length to the characteristic bending length and captures the overall elastohydrodynamics of the swimmer [121]. The quantity Z is the ratio of body to flagellar drag, R is an aspect ratio of flagellar to swimmer length, and V relates the swimming velocity v to some characteristic velocity. While we may define a characteristic velocity $L_\omega \omega$ based on the frequency, we instead define one based on the body properties, $T/\zeta_b L_b$ as this will stay constant as we change the actuation parameters on the tail.

6.2.2 Discretization

The discretization scheme is the same as that introduced in Chapter 3. The flagellum is discretized into N_f nodes $\mathbf{x}_i$ and $N_f - 1$ edges connecting said nodes. To each edge we assign an orthonormal coordinate system $\{\mathbf{e}^1, \mathbf{e}^2, \mathbf{e}^3\}$ where $\mathbf{e}_i^3 = (\mathbf{x}_i - \mathbf{x}_{i-1})/|\mathbf{x}_i - \mathbf{x}_{i-1}|$. The local tangent vector at node $\mathbf{x}_i$ is $\mathbf{t}_i = (\mathbf{e}_i^3 + \mathbf{e}_{i+1}^3)/2$. At each node located between two adjacent edges, we define the local generalized curvature $\mathbf{\Omega} = [\Omega_i^1 \ \Omega_i^2 \ \Omega_i^3]^{\mathrm{T}}$. Specific to this system, care must be taken in choosing the discretization length l, as this will determine the extent of bending for a given beat, as discussed below.

6.3 Equations of motion

The equations of motion for the biohybrid swimmer have the same general contributions as introduced in Chapter 3 with Eqs. 3.6 and 3.7, whose definitions were specified previously. For simplicity, we neglect hydrodynamic interactions between the body and filament. Since we use a prolate spheroid for the body, we must use the appropriate anistropic drag tensors as described in Ref. [59]. We also clarify that the motor contribution does not come from a driving torque embedded in the cell as for bacteria, but comes from the beating cell(s) placed on the exterior of the filament. We now resolve the new $\mathbf{T}^{\text{mot}}$ and $\mathbf{F}^{\text{mot}}$.

6.3.1 Beating dynamics

Fig. 6.1(b)(i) illustrates the beating cell on a *bending joint*, i.e. any node i connected to two other adjacent nodes $i-1$ and $i+1$. The beat is placed on the exterior of the filament at point $\mathbf{x}_i^s$ in the direction of $\mathbf{e}_i^1$, and produces a contractile beating force along the local tangent vector $\mathbf{t}_i$:

$$\mathbf{x}_i^s = \mathbf{x}_i + a\,\mathbf{e}_i^1, \tag{6.9}$$

$$\mathbf{F}_i^- = +F\mathbf{t}_i, \tag{6.10}$$

$$\mathbf{F}_i^+ = -F\mathbf{t}_i, \tag{6.11}$$

with magnitude F given by Eq. 6.2. The superscript notation (-) denotes the side of $\mathbf{x}_i$ toward $i-1$, and (+) the side of $\mathbf{x}_i$ toward $i+1$. This beating force generates equal and oppposite moments of magnitude aF at the centers of edges $\mathbf{e}_i$ and $\mathbf{e}_{i+1}$:

$$\mathbf{x}_i^- = \frac{1}{2}\left(\mathbf{x}_{i-1} + \mathbf{x}_i\right) \tag{6.12}$$

$$\mathbf{x}_i^+ = \frac{1}{2}\left(\mathbf{x}_i + \mathbf{x}_{i+1}\right) \tag{6.13}$$

$$\mathbf{T}_i^- = +\frac{l}{2}\mathbf{e}_i^3 \times \mathbf{F}_i^- \tag{6.14}$$

$$\mathbf{T}_i^+ = -\frac{l}{2}\mathbf{e}_{i+1}^3 \times \mathbf{F}_i^+ \tag{6.15}$$

We resolve the moments on these edges into components perpendicular and parallel to the constituent edges $\mathbf{e}_i^3$ and $\mathbf{e}_{i+1}^3$:

$$\mathbf{T}_i^{-,\perp} = (\boldsymbol{\delta} - \mathbf{e}_i^3\mathbf{e}_i^3) \cdot \mathbf{T}_i^- \tag{6.16}$$

$$\mathbf{T}_i^{+,\perp} = (\boldsymbol{\delta} - \mathbf{e}_{i+1}^3\mathbf{e}_{i+1}^3) \cdot \mathbf{T}_i^+ \tag{6.17}$$

$$T_i^{-,\parallel} = \mathbf{e}_i^3 \cdot \mathbf{T}_i^- \tag{6.18}$$

$$T_i^{+,\parallel} = \mathbf{e}_{i+1}^3 \cdot \mathbf{T}_i^+ \tag{6.19}$$

Note that when the swimming bends only in a single plane, $T_i^{-,\parallel} = T_i^{+,\parallel} = 0$ and there are no torques twisting the filament out of the bending plane. Finally we resolve the above torques as forces (magnitude aF/λ) acting directly on nodes $i-1$, i and $i+1$ (with superscript mot to denote that this is the actuation force).

$$\mathbf{F}_{i-1}^{\text{mot}} = -\frac{1}{l}\left(\mathbf{T}_i^{-\perp} \times \mathbf{e}_i^3\right) \tag{6.20}$$

$$\mathbf{F}_i^{\text{mot}} = \frac{1}{l}\left(\mathbf{T}_i^{-\perp} \times \mathbf{e}_i^3 - \mathbf{T}_i^{+,\perp} \times \mathbf{e}_{i+1}^3\right) \tag{6.21}$$

$$\mathbf{F}_{i+1}^{\text{mot}} = \frac{1}{l}\left(\mathbf{T}_i^{+,\perp} \times \mathbf{e}_{i+1}^3\right) \tag{6.22}$$

Note that when the filament is completely straight, the action of the beat is to bend the filament, as illustrated by the direction of the force arrows in Fig. 6.1(b)(ii).

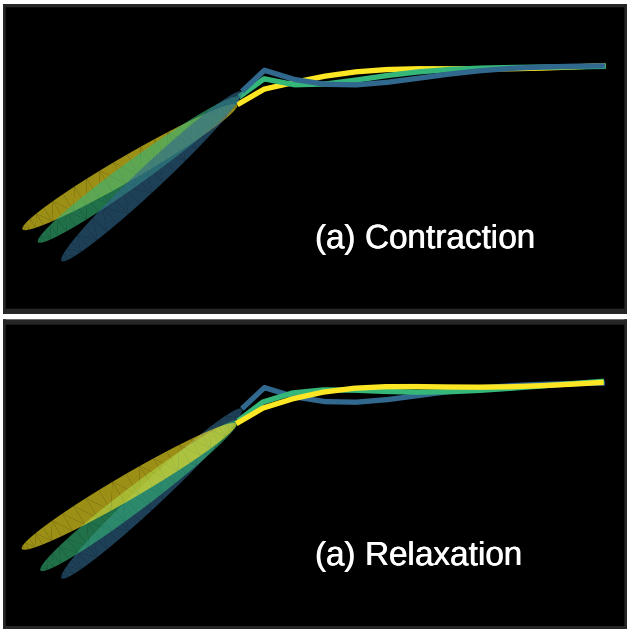

Figure 6.2: Cycle of motion for Fl = 5, Sp = 8. Colors reflect the bending angle. (a) Contraction: starting from the relaxed state in the flagellum bends; time goes from light to dark for a duration of 0.024 time units. (b) Starting from the bent state, the flagellum returns to the relaxed state; time goes from dark to light for a duration of 0.036 time units. The body displacement from beginning to end of cycle is about 0.010.

6.3.2 Additional constraint

Starting with the rigid body motion constraints from Eqs. 3.34, 3.35, and 3.36, we include the following constraint on node 1 (and edge 1):

$$\mathbf{q}_b \mathbf{x}_1^{(b)} \mathbf{q}_b^{-1} - (\mathbf{x}_1 - \mathbf{x}_b) = \mathbf{0}, \tag{6.23}$$

Since we apply a bending boundary condition in the bulk filament rather than applying a torque boundary condition directly on the body, it is more convenient for us to weld edge 1 to the body, true to the convention introduced by Bergou et al. in Ref. [14].

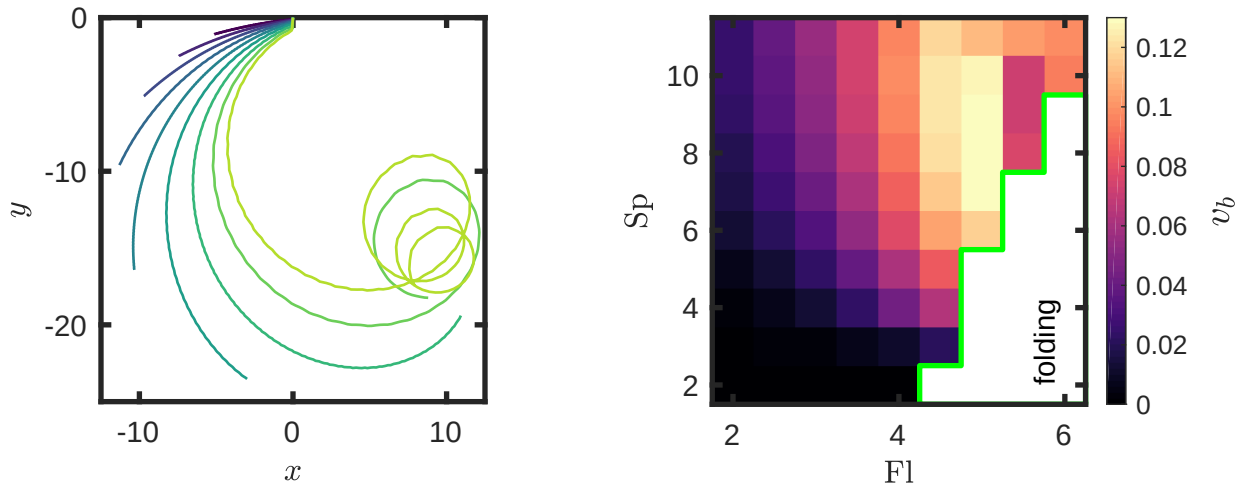

Figure 6.3: (a) Trajectories in $x-y$ plane at fixed Sp = 11 and varying Fl. Lighter color lines denote increasing Fl (b) Average speed v_b as a function of Fl and Sp. The white folding regime denotes extremely large bending fluctuations that do not produce measurable speeds.

6.4 Results

6.4.1 Single tail swimmer

During swimming, the imposed beat goes through a 2-step cycle. First, as in the top panel of Fig. 6.2, *contraction* produces a bend on the first joint of the flagellum, with the coupled body translating in response as well, and next, as in the bottom panel of Fig. 6.2, *relaxation* restores the flagellum to its equilibrium straight state. The elastic response in the flagellum breaks kinematic reversibility and allows the flagellum to propel the cell. The trajectory is 2D because all moments are normal to the plane of bending ($x-y$ in this case). As noted in Ref. [122], this is a slow mechanism, and a single period of contraction and relaxation generates only a miniscule displacement of the swimmer. Their (geometrically) optimized swimmer moved only 0.5% of L_b over several periods of beating. Despite this seemingly slow speed, the swimmer should cover significant distances given that the lifetime of cardiomyocytes has an estimated upper bound of 3-4 days.

We run our simulations much, much longer than those in Ref. [122], finding that *hundreds*, and perhaps *thousands* of periods are required to propel the swimmer one body length, with

far more required to cover longer distances. We show 2D trajectories at a fixed high value of Sp = 11 and varying Fl in Fig. 6.3. Though each simulation is run to the same end time, we note the drastic difference in endpoints for each trajectory. At small Fl, the trajectories are linear and slow. As Fl increases, the trajectories become faster (greater trajectory arclength) and exhibit more curvature. At the highest Fl, the curvature increases with time and the trajectory starts spiraling inward. Overall, we note that the most "effective" motion for a given Sp results from an optimal trade-off between raw speed and curvature.

Reason to use fixed geometry: design restrictions on geometry and aspect ratio, as depending on the application, one may require navigation through small channels, complex environments, etc. For a fixed geometry, we seek the optimal trajectory over a wide range of both Sp and Fl. We simply define an average speed using the net displacement according to the equation below:

$$v_b = \frac{\mathbf{x}_b(T) - \mathbf{x}_b(0)}{T}. \tag{6.24}$$

We show v_b in Sp − Fl parameter space in Fig. 6.3(b). Again, each simulation is run to $T = 200$ time units. We note that v_b should approach zero in several limiting cases: 1) Sp $\to 0$, there is no actuation of the filament; 2) Fl $\to 0$, filament is rigid and kinematic reversibility prevents motion; 3) Sp $\to \infty$, the actuation frequency is too fast for the filament to respond; 4) Fl $\to \infty$, the filament is too flexible and folds upon itself, thereby prohibiting motion. Given these limits and the fact that v_b must be finite somewhere in parameter space, we expect a maximum value of v_b. Indeed, from our speed definition in Eq. 6.24, we note v_b is maximized around Fl = 5 and Sp = 8. The snapshots shown in Fig. 6.2 are generally representative of simulations performed for Fig. 6.3, where the maximum bending angle of each simulation will depend on Fl and Sp.

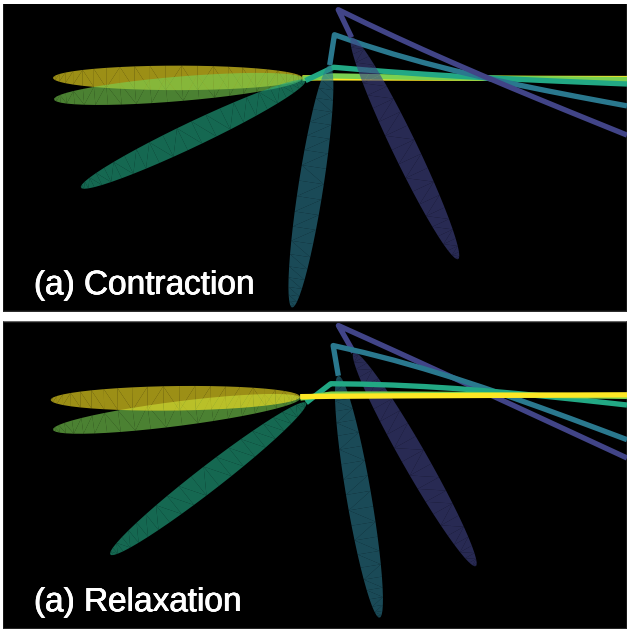

Figure 6.4: Cycle of motion for Fl = 6, $Sp = 2$. Colors reflect the bending angle. (a) Contraction: starting from the relaxed state in the flagellum bends; time goes from light to dark for a duration of 10 time units. (b) Starting from the bent state, the flagellum returns to the relaxed state; time goes from dark to light for a duration of 10 time units. The body displacement from beginning to end of cycle is about 0.020.

6.4.2 Validity of elastic energy formulation

We focus our attention now on the region of low Sp (slow actuation) and high Fl very flexible filaments, represented by the white region in Fig. 6.3. In the limit of large Fl and small Sp, we have calculated v_b to be effectively zero. We shown example snapshots for Fl = 6 and Sp = 2 in Fig. 6.4. We clearly see that indeed the filament has nearly folded on itself, almost to a full 2π radians. We also point out that the actuation is extremely slow due to the low Sp, clocking in at about 20 time units total for contraction and relaxation, about 16 times slower than the example shown in Fig. 6.2. The large deflections here do not generate meaningful movement of the body as $\mathbf{x}_b$ will just *very* slowly oscillate between its initial position and its bent position. As a matter of practicality, large deflections are also undesirable as they are more difficult to control. The question we must ask is whether at this high of angular strain, the restoring moment is still linear in the bending angle. We recall that the Kirchhoff elastic energy presented in Eq. 3.16 is good for deformations of

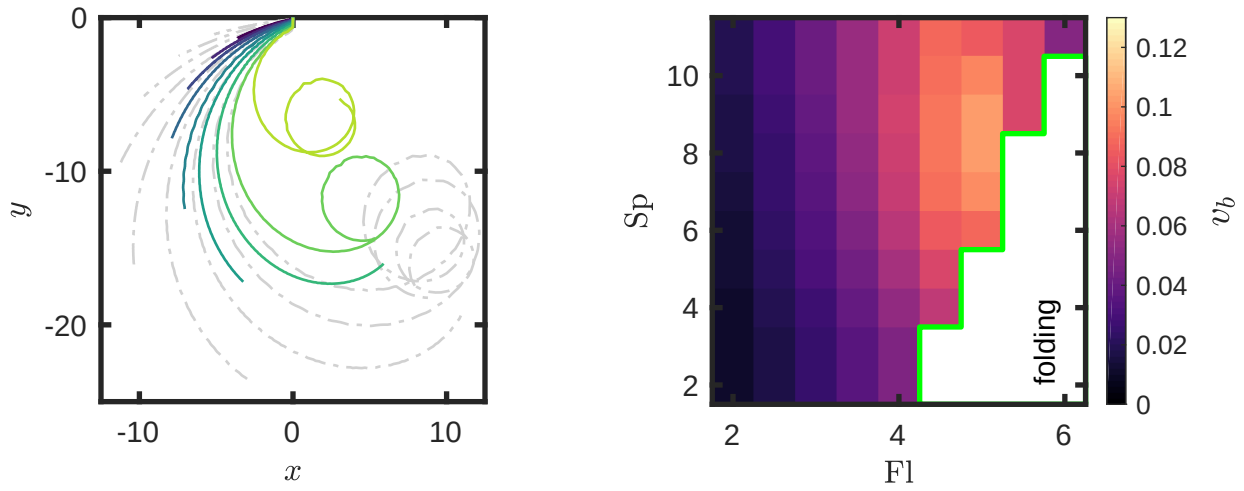

Figure 6.5: Motion of a two-tailed biohybrid swimmer. We fix $(\mathrm{Fl}, \mathrm{Sp}) = (5, 8)$ on first tail and vary it on the other. (a) Trajectories in $x-y$ plane fixing $Sp = 11$ on second tail. Lighter color lines denote increasing Fl on the second tail, and dashed lines are show the trajectories for the single-tail swimmer for comparison (b) Average speed v_b as a function of Fl and Sp. The white folding regime denotes extremely large bending fluctuations that do not produce measurable speeds.

order [], and will require additional corrections at high angular strains. We may also need to consider a non-linear elastic response and/or non-linear deformations in the filament.

6.4.3 Two tail swimmer

We place two tails very close together (about 0.1 radians apart) at the long end of the spheroid. On one tail, we keep Sp and Fl fixed at the optimal value for the single tail swimmer (5 and 8, respectively), and vary Sp and Fl on the second tail. One question we wish to address is whether the second tail can improve the speed of the single tail swimmer. The second question is whether the actuation frequencies of the tail will synchronize, either through hydrodynamic interactions in the fluid or through the mechanical oscillation of the body.

For fixed Sp = 11 on the second tail and varying Fl, we show a number of 2D trajectories in Fig. 6.5(a). With these parameters, we may make direct comparisons with the single tail trajectories in Fig. 6.3(a). Surprisingly, the two-tail trajectories are overall slower than their one-tail counterparts. Even, in the low Fl limit, the difference in endpoints for the

small-curvature trajectories is very pronounced between 1 and 2 tails. At the high Fl end, the presence of the second flexible tail is enough to increase the trajectory curvature to an even greater degree than the one-tail case. Thus, starting from a completely equilibrium straight state, the two-tail swimmer can turn around with less time and distance, and may prove to a more effective steerer than the flexible one-tail swimmer.

We again explore the average swimming speed v_b over Sp − Fl parameter space for the second tail and present the results in Fig. 6.5(b). As we stated above, the overall speeds for the two-tail swimmer are slower. However, we observe one exception to this trend in the region of small Sp and small Fl (note we show high Sp trajectories in Fig. 6.5(a)). We infer from the data that the drag from a non-actuating tail (effectively dead weight) can be overcome from the actuation provided by the second tail, but not by a very significant amount.

6.5 Summary

In this chapter, we adapted the general bacterium model introduced in Chapter 3 to create a general biohybrid model suitable for an arbitrary number of tails and a range of bending angles. In the limit of small deflections of the filament, we are able to reproduce results similar to experiments and predictions from a simple 1D equation. We also observe increasing speeds for increasing bending angles in the filament, but only up to a certain point, as for the cycling time for the largest observable deflections is quite slow, with no significant net displacement in a cycle. Our preliminary results with a two-tail swimmer provide insight on speed and curvature, but perhaps a wider parametric study is required to fully understand the two-tail biohybrid.

We further note the swimmers studied in this chapter had only a single beat on per tail, and the external moment associated with this beat had a fixed actuation pattern that was not responsive to the mechanical motion of the cell or the surrounding fluid flow. A logical

next step is to incorporate the oscillator equations in Refs. [21, 22] to the dynamics driving motion, i.e. $\mathbf{F}^{\text{mot}}$ and $\mathbf{T}^{\text{mot}}$. One further step after this is to add multiple beats to each tail, as was done for filaments studied in Ref. [19]. In theory, when the action of one beat deforms the filament, there should be a mechanical feedback mechanism that affects the actuation from other beats, and thus we should be able to study potential synchronization and understand this effect on the efficiency of motion.

Chapter 7

Conclusions

7.1 General summary

As we stated early on in Chapter 1, bacteria are perhaps the most important natural example of locomotion through Stokes flow, and a detailed understanding of this locomotion behavior is essential to elucidating our knowledge of the world of microswimmers.

Starting with the simple observation of bent configurations in uniflagellar bacteria, we constructed a very simple linked rigid body model where we represent a cell body and a single flagellum as two rigid objects connected by a flexible joint or hook. Using a series of toy models, we are able to analytically predict the critical hook flexibilities that yield buckled configurations of swimmers. These predictions worked reasonably well when compared to full simulations of a torque-driven rigid flagellum. We were also performed a nearly complete parametric study on hook buckling. However, the number of simplifications applied to this swimmer excluded the study of a wide range of locomotion phenomena, so we developed a more detailed model to use in later chapters.

With myriad locomotion phenomena associated with elastic flagella, we sought to develop a general model of a swimmer with a rigid cell body and elastic flagella. The latter point required us to develop a discretization scheme for the flagella, making simulations invariably more expensive but opening the door to a wider range of results. For our general swimmer

model, we incorporate both hook flexibility, governing the evolution of the flagellar axes with respect to the cell body, and flagellar flexibility, governing the degree of deformation in the flagellar filaments themselves. To further improve our model accuracy, we also included hydrodynamic interactions between all swimming components, but limit our swimmer to spherical cell body for simplicity. This general model was used to study both free swimmers and near wall swimmers.

In our study of a free swimmer, we consider changes to swimming stability between uni- and multi-flagellar swimmers in the parameter space of hook and flagellar flexibilities. Our simulations begin with analyzing the stability of a uniflagellar swimmer, and a continually simulate swimmers with additional flagella until reaching the end goal of a quadriflagellar swimmer with an isotropic distribution of flagellar motors. For the former uniflagellar case, we find that stable swimming occurs only in a limited region of parameter space where hook and flagellar flexibilities are quite small, i.e. stiff hooks and filaments are required for swimming. The addition of a second and third flagellum fundamentally alters the stability region, and we observe a number of interesting swimming states with bundled and unbundled flagella, namely that there can be stable swimming both with and without bundles. For the quadriflagellar swimmer, we see a stability region opposite that of the uniflagellar swimmer – that is, multiflagellar swimmers require high flexibilities to swim successfully. However, the additional stability provided by multiflagellar swimmers does not translate to faster swimming speeds, and we in fact show that higher flagellar multiplicities come with diminishing speed returns.

The final two chapters cover initial exploratory work, both using models adapted from our general swimmer. For a swimmer near a wall, we perform simulations by simply modifying the mobility matrix for the free swimmer with the proper image solutions. Though we are not yet able to reproduce all important wall-swimming phenomena with the parameters and swimmers tested thus far, the model shows great promise in accurately capturing near-wall locomotion. We also develop a model biohybrid swimmer that swims via an external beating

moment on the flagellar filament rather than a motor torque supplied from the body. We are able to reproduce baseline results from previous experimental and computational studies of cardiomyocyte driven biohybrid swimmers. Because our model has wider applicability, we also perform studies with moderate to extreme filament deformations and the effects of multiple tails, but more though must be given to these preliminary results. Ultimately, these last two chapters provide a thorough starting foundation for more extensive study on interesting problems of microswimmer locomotion.

7.2 Future work

7.2.1 Code acceleration

We have already implemented a partial code acceleration by enforcing inextensibility of the flagellar filament via explicit constraints instead of stiff springs. Despite this, the equations of motion are still rather stiff for the torque-driven model swimmers, as we must track the rotational phase of each segment of the flagellar filament. Recall that the rotational drag ζ_i^r of each segment scales as the square of the filament radius, a^2. Because this rotational drag is very small compared to the translational drag on any node, the maximum time step is limited to the order of $a^2/100$, which is much smaller than the rotation period of a flagellum. Note that several flagellar rotations are needed to translate the cell body, and with this time limitation, we keep the simulation windows to roughly 7-8 body lengths at the maximum.

One potential remedy we discussed in Chapter 1 is the quasi-static approximation (QSA) associated with the Discrete Elastic Rods (DER) algorithm [14], where twist is propagated instantaneously down the filament since there is little resistance. However, most of the problems tested with DER use clamped or free boundary conditions, i.e. the conformational triad on the edge attached to the endpoint is either fixed or free. Neither of these boundary conditions applies to a torque driven filament where one end is constantly rotated, and thus the QSA may not work as intended. For instance, we performed a preliminary DER

simulation with an anchored flagellum where we applied a constant torque to the anchored edge. However, after applying the QSA, we observed rigid rotation of the helical flagellum, i.e. the anchored portion and free portion of the flagellum moved simultaneously. This is not the case when using the elastic model with no QSA as in Chapter 3. Were we to find a way to remedy this issue with the appropriate adjusted boundary conditions in conjunction with QSA, we should be able to substantially increase the time step and run faster simulations.

7.2.2 Body shape

As stated in Chapter 3, we use a spherical body in order to calculate hydrodynamic interactions via Faxén's laws and avoid an expensive discretization of the cell body. (The number of nodes required to properly discretize a cell body can easily outnumber the total number of flagellar nodes in a swimmer). However, with certain non-spherical body shapes, we can still employ a number of analytical approximations to avoid discretization. For instance, ellipsoidal bodies may prove to be a feasible option, as approximate forms for hydrodynamic interactions exist in the literature. We also note that the hydrodynamics of an ellipsoid can be taken as a reasonable approximation for a myriad of other shapes, for instance in considering bacteria with pill shaped or rod shaped bodies.

7.2.3 Filament shape

For the work shown here, we have opted to use circular cross sections in the slender filaments for simplicity. However, we note that the elastic dynamics can readily handle anistropic bending associated with filaments with a non-circular cross section. For instance, the filaments used to create biohybrid swimmers in experiments in Ref. [122] have rectangular cross section to accommodate the beating cells, and the hydrogel flagellum in Ref. [45] has changing cross sections as it swells and deswells in response to its environment. In short, there existing literature has already highlighted an important role for non-circular filaments in a variety of microswimmer applications, and it may be of great interest to incorporate these attributes

into the current swimmer model.

7.2.4 Stochastic processes

As we discussed in Chapter 1, stochastic processes play in important role in all the aspects of locomotion, from generating random-walk trajectories to initiating polymorphic phase transformations. Our current work does not currently include any stochastic elements, but our models can be readily adapted to include them if needed. For instance, in both the linked rigid body model and full elastic flagella model, we can easily include a switching time for the direction of one or more flagellar motors, as has been done in many studies in the literature. The linked rigid body model provides the easiest starting point for further studies, as it is minimal and fast. For instance, with the mere introduction of a switching time and transient dynamics for the hook, we can use the linked rigid body model to study pseudo-diffusion of a uniflagellar swimmer with a code that runs in a matter of minutes.

Appendix A

Singularity solutions

In this appendix, we show the fundamental solutions to the Stokes equations that are necessary for use in models implementing full hydrodynamic interactions, as used in Chapters 4 and 5.

A.1 Regularized solutions to Stokes equations

The outline presented here roughly follows that presented by Cortez et al. in Ref. [23]. For a point $\mathbf{x}$ in the fluid domain, we write the Stokes equations for *regularized* point force located at $\mathbf{x}_0$:

$$-\boldsymbol{\nabla} p + \mu \nabla^2 \mathbf{u} = \mathbf{f}\phi(r), \tag{A.1a}$$

$$\boldsymbol{\nabla} \cdot \mathbf{u} = 0, \tag{A.1b}$$

where $\mathbf{u}$ is the velocity field, $r = |\mathbf{x} - \mathbf{x}_0|$, and $\phi(r)$ represents a *blob* function with regularization length scale ξ^{-1} controlling its spread. With this regularization, the velocity is finite everywhere and in the limit of $\xi^{-1} \to 0$, $\phi(r)$ becomes the delta function. We consider the Gaussian blob ϕ_s as in Hernandez-Ortiz [43]:

$$\phi_s(r) = \left(\frac{\xi}{\sqrt{\pi}}\right)^3 \exp(-\xi^2 r^2)\left(\frac{5}{2} - \xi^2 r^2\right) \tag{A.2}$$

Then from Eqs. A.1 and A.2, the velocity field generated by the point force is:

$$\mathbf{u}(\mathbf{x}) = \mathbf{S}(\mathbf{x}) \cdot \mathbf{f} \tag{A.3}$$

where $\mathbf{S}(\mathbf{x})$ is the regularized Stokeslet tensor defined below:

$$\mathbf{S}(\mathbf{x}) = \frac{1}{8\pi\mu} \left[H_1(r)\boldsymbol{\delta} + H_2(r)\mathbf{x}\mathbf{x} \right], \tag{A.4}$$

where $\boldsymbol{\delta}$ is the identity tensor and $\mathbf{x} = \mathbf{x}_s - \mathbf{x}_0$. The auxiliary functions H_1 and H_2 are derived from the blob ϕ_s:

$$H_1(r) = \frac{\mathrm{erf}(\xi r)}{r} + \frac{2\xi}{\sqrt{\pi}} \exp(-\xi^2 r^2) \tag{A.5a}$$

$$H_1(r) = \frac{1}{r^2} \left[\frac{\mathrm{erf}(\xi r)}{r} - \frac{2\xi}{\sqrt{\pi}} \exp(-\xi^2 r^2) \right] \tag{A.5b}$$

We note that in the limiting case of $\xi \to \infty$, we indeed recover the well-known singular solution for the Stokeslet (see Ref. [59]):

$$\lim_{\xi \to \infty} \mathbf{S}(\mathbf{x}) = \frac{1}{8\pi\mu} \left(\frac{\boldsymbol{\delta}}{r} + \frac{\mathbf{x}\mathbf{x}}{r^3} \right) \tag{A.6}$$

Following Cortez et al. [23], we also write potential flow solutions to the regularized Stokes solutions by defining a companion blob ϕ_d using the following relation:

$$\phi_s = \frac{1}{2} \left(r\phi_d' + 5\phi_d \right) \tag{A.7}$$

$$\phi_d(r) = \left(\frac{\xi}{\sqrt{\pi}} \right)^3 \exp(-\xi^2 r^2) \tag{A.8}$$

Following the name and vector convention from [Spagnolie], we take derivatives of the regularized Stokeslet to obtain the regularized Stokeslet dipole **SD** and Stokeslet quadrupole **SQ**.

(The Stokeslet descriptor emphasizes that these singularities are derived from the Stokeslet – this will be important later.)

$$\mathbf{SD}(\mathbf{x};\mathbf{e}) = \mathbf{e}\cdot\boldsymbol{\nabla}\mathbf{S}(\mathbf{x}) \tag{A.9}$$

$$\mathbf{SQ}(\mathbf{x};\mathbf{e},\mathbf{d}) = \mathbf{d}\cdot\boldsymbol{\nabla}\mathbf{SD}(\mathbf{x};\mathbf{e}) \tag{A.10}$$

Following Ref. [99], the vectors $\mathbf{e}$ and $\mathbf{d}$ denote the direction that derivatives are taken. The convention of Cortez [23] uses ϕ_s to derive Stokeslets (and derivatives), and a related blob ϕ_d for sources, dipoles, and their derivatives). The reason for using two related blobs is to maintain the same algebraic simplifcations in the regularized case as the singular case [23]. We write the regularized source ($\mathbf{U}$), source dipole ($\mathbf{D}$), and source quadrupole ($\mathbf{Q}$):

$$\mathbf{U}(\mathbf{x}) = \frac{\mathbf{x}}{4\pi\mu r^3}\left(\mathrm{erf}(\xi r) - \frac{2\xi r}{\pi^{3/2}}\right) \tag{A.11}$$

$$\mathbf{D}(\mathbf{x}) = \boldsymbol{\nabla}\mathbf{U}(\mathbf{x}) \tag{A.12}$$

$$\mathbf{Q}(\mathbf{x};\mathbf{d}) = \mathbf{d}\cdot\boldsymbol{\nabla}\mathbf{D}(\mathbf{x}) \tag{A.13}$$

The Stokeslet and source singularities are related by the following:

$$\mathbf{D}(\mathbf{x}) = -\nabla^2\mathbf{S}(\mathbf{x}) = \frac{1}{8\pi\mu}\left[D_1(r)\boldsymbol{\delta} + D_2(r)\mathbf{xx}\right] \tag{A.14}$$

where the dipole coefficients D_1 and D_2 are related to the Stokeslet coefficients H_1 and H_2 via:

$$\frac{2H_1'}{r} + 4H_2 + D_1 = 0 \tag{A.15a}$$

$$D_2 + \frac{2H_2'}{r} = 0 \tag{A.15b}$$

Using Eq. A.15 above, we present the vector explicit forms of the higher order singularities:

$$\mathbf{SD}(\mathbf{x};\mathbf{e}) = \frac{(\mathbf{x}\cdot\mathbf{e})}{2}\left(2H_2\boldsymbol{\delta} - D_2\mathbf{xx}\right) + \frac{H_1'}{r}\mathbf{ex} + H_2\mathbf{xe} \tag{A.16}$$

$$\begin{aligned}\mathbf{SQ}(\mathbf{x};\mathbf{e},\mathbf{e}) = -\frac{1}{2r}\left[(D_1' + 4H_2')(\mathbf{x}\cdot\mathbf{e})^2 - 2H_1'\right]\boldsymbol{\delta} - \frac{1}{2}\left[\frac{D_2'}{r}(\mathbf{x}\cdot\mathbf{e})^2 + D_2\right]\mathbf{xx} \\ + 2H_2\mathbf{ee} - D_2(\mathbf{x}\cdot\mathbf{e})(\mathbf{xe}+\mathbf{ex})\end{aligned} \tag{A.17}$$

$$\mathbf{Q}(\mathbf{x},\mathbf{e}) = \frac{(\mathbf{x}\cdot\mathbf{e})}{r}\left(D_1'\boldsymbol{\delta} + D_2'\mathbf{xx}\right) + D_2\left(\mathbf{ex}+\mathbf{xe}\right) \tag{A.18}$$

We note that all the solutions presented so far correspond to a point force. For a point torque $\mathbf{T}$, solutions to the Stokes equations can be written as:

$$\mathbf{u}(\mathbf{x}) = \frac{1}{8\pi\mu}\frac{\mathbf{T}\times\mathbf{x}}{r^3} = \mathbf{R}(\mathbf{x})\cdot\mathbf{T} \tag{A.19}$$

where $\mathbf{R}$ is defined as the rotlet tensor:

$$\mathbf{R}(\mathbf{x}) = -\frac{[\mathbf{x}]_\times}{8\pi\mu r^3}, \tag{A.20}$$

where $[\cdot]_\times$ denotes the cross product operator in tensor form. For the remainder of this appendix, all singularity solutions are assumed to be regularized unless otherwise stated.

A.2 Image systems

Given the regularized singularity at point $\mathbf{x}_0$ in the fluid domain, we define the image point as the reflection across the wall:

$$\mathbf{x}_0^* = \mathbf{P}\cdot\mathbf{x}_i; \quad \mathbf{P} = \boldsymbol{\delta} - 2\mathbf{ee} \tag{A.21}$$

where $\mathbf{P}$ is the reflection operator. All tensors in the rest of this paper will refer to regularized quantities unless otherwise stated. Following Ref. [3], the image system for a Stokeslet at $\mathbf{x}_0$ consists of a Stokeslet ($\mathbf{S}$), a Stokeslet dipole ($\mathbf{SD}$), a source dipole ($\mathbf{D}$), and 2 rotlets ($\mathbf{R}_d$, $\mathbf{R}_s$). Defining $\mathbf{x} = \mathbf{x}_s - \mathbf{x}_0$ and $\mathbf{x}^* = \mathbf{x}_s - \mathbf{x}_0^*$, we define the image system:

$$\mathbf{S}^{\mathrm{IM}} = -\mathbf{S}(\mathbf{x}^*) + \left[2h\mathbf{SD}(\mathbf{x}^*;\mathbf{e}) + h^2\mathbf{D}(\mathbf{x}^*) + 2h(\mathbf{R}_d - \mathbf{R}_s)(\mathbf{x}^*;\mathbf{e})\right] \cdot (-\mathbf{P}) \tag{A.22}$$

$$2(\mathbf{R}_d - \mathbf{R}_s)(\mathbf{x};\mathbf{e}) = (D_1 + 2H_2)\left[-(\mathbf{x}\cdot\mathbf{e})\boldsymbol{\delta} + \mathbf{ex}\right] \tag{A.23}$$

Following Ref. [23], the regularized image system for a dipole at $\mathbf{x}_0$ consists of a dipole ($\mathbf{D}$), a Stokeslet quadrupole ($\mathbf{SQ}$), a source quadrupole ($\mathbf{Q}$), a second dipole, and a rotlet dipole ($\mathbf{RD}$).

$$\mathbf{D}^{\mathrm{IM}} = -\mathbf{D}(\mathbf{x}^*) + [4\mathbf{SQ}(\mathbf{x}^*;\mathbf{e},\mathbf{e}) + 2h\mathbf{Q}(\mathbf{x}^*;\mathbf{e}) + 2\mathbf{D}(\mathbf{x}^*)] \cdot \mathbf{P} + 4\mathbf{RD}(\mathbf{x}^*;\mathbf{e}) \tag{A.24}$$

$$\mathbf{RD}(\mathbf{x};\mathbf{e}) = \frac{\phi_d' - D_1'}{r}(\mathbf{x}\cdot\mathbf{e})\left[(\mathbf{x}\cdot\mathbf{e})\boldsymbol{\delta} - \mathbf{ex}\right] + (\phi_d - D_1)(\boldsymbol{\delta} - \mathbf{ee}) \tag{A.25}$$

For the rotlet image system, we use the singular solutions provided by [Blake], since $r \geq 1$:

$$\mathbf{R}^{\mathrm{IM}} = \frac{1}{8\pi\mu r^3}\left[[\mathbf{x}^*]_\times + 2h[\mathbf{e}]_\times + (6(\mathbf{x}^*\cdot\mathbf{e}) - 2h)\frac{(\mathbf{x}^*\mathbf{e})\cdot[\mathbf{x}^*]\times}{r^2}\right] \tag{A.26}$$

Appendix B

Derivation of projection method for constrained motion

In this appendix, we derive a projection algorithm unique to our swimmer problem by combining the SHAKE-HI algorithm derived by Alison et al. with the Fast Projection method derived by Goldenthal et al. [5, 36]. Rather than beginning with a conserved energy functional as in Ref. [36], we follow the initial step of SHAKE-HI [5] and start with the total force balance on our swimmer, i.e. viscous drag balances all other external forces when neglecting the inertia of the swimmer:

$$-\mathcal{M}^{-1}\cdot\dot{\mathbf{y}}+\mathbf{f}-\boldsymbol{\nabla}\mathbf{C}^{\mathrm{T}}\cdot\boldsymbol{\Lambda}=\mathbf{0}. \tag{B.1}$$

Recall $\mathcal{M}$, $\dagger$, $\mathbf{f}$ are the composite mobility matrix, velocity, and force for the swimmer defined in Eq. 3.38. The vector $\mathbf{C}$ contains all constraint equations, and $\boldsymbol{\Lambda}$ is the vector of corresponding Lagrange multipliers so that the quantity $-\boldsymbol{\nabla}\mathbf{C}^{\mathrm{T}}\cdot\boldsymbol{\Lambda}$ denotes the constraint force on the swimmer. (Note that Eq. B.1 is simply a more general version of the force balance in Chapter 2). We define the *unconstrained* velocity as:

$$\dot{\mathbf{y}}^{\dagger}=\mathcal{M}\cdot\mathbf{f}, \tag{B.2}$$

and the *constrained* velocity as:

$$\dot{\mathbf{y}} = \mathcal{M} \cdot \left(\mathbf{f} - \boldsymbol{\nabla} \mathbf{C}^{\mathrm{T}} \cdot \boldsymbol{\Lambda} \right). \tag{B.3}$$

We define an explicit forward step in time from t_n to t_{n+1} as:

$$\mathbf{y}_{n+1} = \mathbf{y}_n + \Delta t \, \dot{\mathbf{y}}. \tag{B.4}$$

Letting $\delta \mathbf{y}_{n+1}$ denote the difference between a constrained and unconstrained time step, the true time step must satisfy the following conditions:

$$\delta \mathbf{y}_{n+1} + \Delta t \, \mathcal{M} \cdot \boldsymbol{\nabla} \mathbf{C}^{\mathrm{T}}(\mathbf{y}_{n+1}) \cdot \boldsymbol{\Lambda} = \mathbf{0}, \tag{B.5}$$

$$\mathbf{C}(\mathbf{y}_{n+1}) = \mathbf{0}. \tag{B.6}$$

We emphasize that the constraint and constraint gradient are evaluated at the *end* of the time step, i.e. using the state $\mathbf{y}_{n+1}$. This implicit formulation was introduced by Goldenthal to rectify instabilities in SHAKE-HI caused by common configurations [36]. The remaining derivation closely follows the methodology of [36]. First we define a functional W whose stationary equations are exactly Eqs. B.5 and B.6:

$$W\left(\delta \mathbf{y}_{n+1}, \boldsymbol{\Lambda}\right) = \frac{1}{2\Delta t} (\delta \mathbf{y}_{n+1})^{\mathrm{T}} \cdot \mathcal{M} \cdot \delta \mathbf{y}_{n+1} + \mathbf{C}^{\mathrm{T}}(\mathbf{y}_{n+1}) \cdot \boldsymbol{\Lambda} \tag{B.7}$$

We linearize the constraint term near $\mathbf{y}_n$ to obtain:

$$\mathbf{C}(\mathbf{y}_{n+1}) = \mathbf{C}(\mathbf{y}) + \boldsymbol{\nabla} \mathbf{C}(\mathbf{y}_n) \cdot \delta \mathbf{y}_{n+1}. \tag{B.8}$$

Substituting into W:

$$W\left(\delta\mathbf{y}_{n+1}, \mathbf{\Lambda}\right) = \frac{1}{2\Delta t}(\delta\mathbf{y}_{n+1})^{\mathrm{T}} \cdot \mathcal{M} \cdot \delta\mathbf{y}_{n+1} + \mathbf{C}^{\mathrm{T}}(\mathbf{y}_n) \cdot \mathbf{\Lambda} + (\mathbf{y}_{n+1})^{\mathrm{T}} \boldsymbol{\nabla}\mathbf{C}^{\mathrm{T}} \cdot \mathbf{\Lambda} \tag{B.9}$$

We calculate the stationary equations for W to arrive at equations for $\mathbf{y}_{n+1}$ and $\mathbf{\Lambda}$:

$$\frac{\delta W}{\delta \mathbf{y}_{n+1}} = \frac{1}{\Delta t}\, \mathcal{M} \cdot \delta\mathbf{y}_{n+1} + \boldsymbol{\nabla}\mathbf{C}^{\mathrm{T}}(\mathbf{y}_n) \cdot \mathbf{\Lambda} = 0, \tag{B.10}$$

$$\frac{\delta W}{\delta \mathbf{\Lambda}} = \mathbf{C}(\mathbf{y}_n) + \boldsymbol{\nabla}\mathbf{C}(\mathbf{y}_n) \cdot \delta\mathbf{y}_{n+1} = 0 \tag{B.11}$$

$$\downarrow$$

$$\Delta t \left[\boldsymbol{\nabla}\mathbf{C}(\mathbf{y}_n) \cdot \mathcal{M} \cdot \boldsymbol{\nabla}\mathbf{C}^{\mathrm{T}}(\mathbf{y}_n)\right] \cdot \mathbf{\Lambda} = \mathbf{C}(\mathbf{y}_n) \tag{B.12}$$

$$\delta\mathbf{y}_{n+1} = -\Delta t\, \mathcal{M} \cdot \boldsymbol{\nabla}\mathbf{C}^{\mathrm{T}}(\mathbf{y}_n) \cdot \mathbf{\Lambda} \tag{B.13}$$

Thus, our projection algorithm iteratively solves Eq. B.12 for $\mathbf{\Lambda}$ and Eq. B.13 for $\delta\mathbf{y}_{n+1}$ with the update $\mathbf{y}_{n+1} + \delta\mathbf{y}_{n+1} \mapsto \mathbf{y}_{n+1}$ until $|\mathbf{C}(\mathbf{y}_{n+1})|$ is below our set tolerance, ϵ_p.

One last note on body rotation: to account for the body torque, we follow the convention introduced by Bergou et al. of including the body rotational mobility $\mathbf{M}_{b,r}$ in $\mathcal{M}$, the body torque in $\mathbf{f}$, and the body angular velocity in $\dot{\mathbf{y}}$. [14]. However, the body rotational state or orientation is tracked with a quaternion $\mathbf{q} \in \mathbb{H}^4$ (or $\mathbb{R}^4$) while the aforementioned dynamics reside in a three-dimensional space. To resolve this dimensional conflict, we recall from Chapter 2 the relationship between $\mathbf{q}$ and $\boldsymbol{\omega}$.

$$\dot{\mathbf{q}} = \frac{1}{2}\mathcal{W}(\boldsymbol{\omega}) \cdot \mathbf{q} \tag{B.14}$$

If we rewrite the above expression using quaternion multiplication, we have:

$$\dot{\mathbf{q}} = \frac{1}{2}\boldsymbol{\omega}\mathbf{q} \;\leftrightarrow\; \boldsymbol{\omega} = 2\dot{\mathbf{q}}\mathbf{q}^{-1} \tag{B.15}$$

We emphasize that $\boldsymbol{\omega}_b$ is written here in the laboratory frame. Eq. B.15 can be interpreted in two ways: $\boldsymbol{\omega}_b$ is a quaternion with zero real part: $[0\ \boldsymbol{\omega}_b]^{\mathrm{T}}$ or $\dot{\mathbf{q}}\mathbf{q}^{-1}$ is a vector in $\mathbb{R}^3$. For now, we use the former definition to write the torque balance (with constraints) on the body:

$$\mathbf{M}^{-1} \cdot \dot{\mathbf{q}} = \mathbf{T}\mathbf{q} - \boldsymbol{\nabla}\mathbf{C}^{\mathrm{T}} \cdot \boldsymbol{\Lambda}. \tag{B.16}$$

Here the gradient operator takes derivatives with respect to $\mathbf{q}$. We interpret $-\boldsymbol{\nabla}\mathbf{C}^{\mathrm{T}} \cdot \boldsymbol{\Lambda}$ as a constraint torque (to have consistent units) on the quaternion velocity. The rotational mobility matrix $\mathbf{M}$ must be invertible, and in an unconstrained setting, we must satisfy $\boldsymbol{\omega} = \mathbf{M}_{b,r} \cdot \mathbf{T}$ [recall that $\mathbf{M}_{b,r}$ is the rotational mobility corresponding to the body shape]. With these two conditions, we may simply write:

$$\mathbf{M} = \begin{bmatrix} 1 & 0 \\ 0 & (1/2)\mathbf{M}_{b,r} \end{bmatrix} \tag{B.17}$$

Thus, Eq. B.17 is the actual matrix used as the rotational piece of $\mathcal{M}$, and the remaining derivation follows the same steps as the previous section. One last remark is that because we are operating in the laboratory frame, $\mathbf{M}_{b,r}$ must be written in the laboratory frame, which is the opposite convention than that adopted in Ref. [14].

Appendix C

Stiff springs method

In this appendix, we discuss a common alternative approach used to model flagellar filaments, as seen in references such as [1, 2, 62, 114, 115]. Rather than calculating exact constraint forces to maintain inextensibility, a much used alternative is to connect the nodes via very stiff springs to keep the total extension of a filament below a threshold, typically around 1%. Implementing this method begins with modifying the elastic energy with a stretching term (superscript stretch):

$$\mathcal{E}^{\text{el}} = \mathcal{E}^{\text{curv}} + \mathcal{E}^{\text{stretch}} \tag{C.1}$$

$$\mathcal{E}^{\text{stretch}} = B \sum |L_i - L_{i,\text{eq}}|^2, \quad L_i = |\mathbf{x}_i - \mathbf{x}_{i-1}| \tag{C.2}$$

Here the term $\mathcal{E}^{\text{curv}}$ is the contribution to the elastic energy from the filament curvature as previously defined in Chapter 3. The stretching energy $\mathcal{E}^{\text{stretch}}$ is derived from the displacement of the nodes from their equilibrium length – the springs are assumed to follow Hook's law with stiffness constant B. To maintain inextensibility, B is set very high, typically 100-200 times the characteristic force of a system. We then define the stretching force as [85]:

$$\frac{\partial \mathcal{E}^{\text{stretch}}}{\partial \mathbf{x}_i} = -B(L_i - L_e)\mathbf{e}_i^3 + B(L_{i+1} - L_e)\mathbf{e}_{i+1}^3. \tag{C.3}$$

The above force replaces the constraints previously used in the projection method, and allows us to formulate a pure differential equation system for the swimmer.

$$\boldsymbol{\omega}_b = \sum \mathbf{T}_0 + R_b \sum (\mathbf{e}_0^3 \times \mathbf{F}_0) + \boldsymbol{\omega}_\infty^f \quad \mapsto \quad \dot{\mathbf{q}}_b = \frac{1}{2}\boldsymbol{\omega}\mathbf{q}_b \tag{C.4}$$

$$\mathbf{v}_b = \sum \mathbf{F}_0 + \mathbf{v}_\infty^f \tag{C.5}$$

$$\mathbf{v}_0 = \mathbf{v}_b + R_b(\boldsymbol{\omega}_b \times \mathbf{e}_0^3) \tag{C.6}$$

$$\mathbf{v}_i = \boldsymbol{\zeta}^{-1} \cdot \mathbf{F}_i + \mathbf{v}_{i,\infty}^f + \mathbf{v}_b^\infty, \quad i \neq 0 \tag{C.7}$$

$$\omega_i = \zeta_r T_i \tag{C.8}$$

Note that the attachment of the anchor point (index 0) to the body is manually enforced in any integration scheme. Overall, this system is quite simple in design, but the numerics can become very stiff since B is very large. For a small system such as a uni- or biflagellar swimmer without enormous steric interactions, this system can be solved reasonably quickly using an Adams-Bashforth method. Faster integration algorithms are suggested in Refs. [115] and [62], but note that the problems studied are not exactly analogous to the ones presented in our work and may not have the same potential numerical issues.

Appendix D

Code excerpts

The C++ code below defines the classes for body and flagellum with their corresponding geometric parameters, as well as dynamics and kinematics. We use the Armadillo C++ library for linear algebra, as this provides similar syntax for vector and matrix operations as MATLAB.

The in-class functions mostly deal with initialization of the swimmer and include simple linear integration schemes. We note that at the end of this code we have provided the functions to alternatively utilize the full DER convention described in Ref. [14].

```
1
2  #include <iostream>
3  #include <armadillo>
4  #include "Swimmer.hpp"
5
6  using namespace std;
7  using namespace arma;
8
9  // Default constructor
10 Body::Body () {
11     radius = 1.00;
12     z_inv = 1.00;
13     zr_inv = 0.75;
14     Anchor = 1;
15     numNodes = 1;
16     Node = zeros<mat>(3,1);
17     mat I = eye<mat>(3,3);
18     Axis = I.col(0);
19     Normal = I.col(1);
20     Binormal = I.col(2);
21     Force = zeros<vec>(3,1); Force_steric = Force; Torque = Force;
```

```
22      Velocity = Force; Omega = Force;
23      Velocity_HI = Force; Omega_HI = Force;
24      q = eye<vec>(4,1); dq = 0*q;
25  }
26
27  // Default constructor
28  Flagellum::Flagellum () {
29      helix_radius = 0.28;
30      helix_b = 7.5*helix_radius / (2*datum::pi);
31      length = 10.5;
32      fil_radius = 0.028;
33      h_length = 0.28;
34      seg_length = 15 * fil_radius;
35      numNodes = int (length/seg_length) + 1;
36      A_fil = 10;
37      A_hook = 5;
38      B_stretch = 100;
39      G_twist = 1.00;
40      G_hook = 1.00;
41      T_mot = -1;
42      theta0 = 0;
43
44      double PI = datum::pi;
45      double ln2qr = log ( 0.18 * 2 * PI * sqrt(helix_radius * helix_radius
46          + helix_b * helix_b) / fil_radius  );
47      delta = 1.13 * seg_length / (ln2qr + 0.5);
48      zpar_inv = 3 / seg_length * ln2qr;
49      zperp_inv = 1.5 /seg_length *(ln2qr + 0.5);
50      zr_inv = 1.5 / (fil_radius * fil_radius * seg_length);
51
52      Nodes = zeros<mat>(3,numNodes);
53      E1 = Nodes; E2 = Nodes; E3 = Nodes; W_eq = Nodes;
54      dE1 = Nodes; dE2 = Nodes; dE3 = Nodes;
55      Forces = Nodes; Velocities = Nodes;
56      Forces_steric = Nodes; Velocities_HI = Nodes;
57
58      Fc0 = zeros<vec>(3,1);
59      x_eq = Nodes.col(0); m_eq = x_eq;
60      kb = Nodes; U = Nodes; V = U; M1 = U; M2 = U;
61      W_eq_left = zeros<mat>(2,numNodes); W_eq_right = W_eq_left;
62      W_left = W_eq_left; W_right = W_eq_left;
63      Phi = zeros<vec>(numNodes,1); phi_eq = Phi; dPhi = Phi;
64
65      Beats = zeros<mat>(3,numNodes);
66      BeatsTwist = zeros<mat>(1,numNodes);
67      METHOD = "Normal";
68  }
69
70  void Body::integrateLin1(double dt) {
71      Node += dt * Velocity;
72      Axis += dt * cross(Omega, Axis);
73      Normal += dt * cross(Omega, Normal);
74      Binormal += dt * cross(Omega, Binormal);
75  }
```

```
76
77  void Body::integrateLin2(double dt, field<vec> dB) {
78      Node += dt * ( 1.5 * Velocity - 0.5*dB(0) );
79      Axis += dt * ( 1.5 * cross(Omega, Axis) - 0.5 * cross(dB(1), Axis) );
80      Normal += dt * ( 1.5 * cross(Omega, Normal) - 0.5 * cross(dB(1), Normal) );
81      Binormal += dt * ( 1.5 * cross(Omega, Binormal) - 0.5 * cross(dB(1), Binormal) );
82  }
83
84  field<vec> Body::storeDerivatives(void) {
85      field<vec> Derivatives(1,2);
86      Derivatives(0) = Velocity;
87      Derivatives(1) = Omega;
88      return Derivatives;
89  }
90
91  mat Body::output(void) {
92      return join_vert( Node, join_vert(Axis, Normal) ).t();
93  }
94
95  void Flagellum::integrateLin1(double dt) {
96      //Nodes += dt * Velocities;
97      E1 += dt * dE1; E2 += dt * dE2; E3 += dt * dE3;
98  }
99
100 void Flagellum::integratePositionsLin1(double dt) {
101     Nodes += dt * Velocities;
102 }
103
104 void Flagellum::integratePhiLin1(double dt) {
105     Phi -= dt * zr_inv * dPhi;
106     Phi(1) += dt * zr_inv * T_mot;
107 }
108
109 void Flagellum::integrateLin2(double dt, field<mat> dF) {
110     Nodes += dt * ( 1.5 * Velocities - 0.5*dF(0) );
111     E1 += dt * ( 1.5 * dE1 - 0.5*dF(1) );
112     E2 += dt * ( 1.5 * dE2 - 0.5*dF(2) );
113     E3 += dt * ( 1.5 * dE3 - 0.5*dF(3) );
114 }
115
116 field<mat> Flagellum::storeDerivatives(void) {
117     field<mat> Derivatives(1,4);
118     Derivatives(0) = Velocities;
119     Derivatives(1) = dE1;
120     Derivatives(2) = dE2;
121     Derivatives(3) = dE3;
122     return Derivatives;
123 }
124
125 void Flagellum::initLine(void) {
126     mat I = eye<mat>(3,3);
127     E3 = repmat(I.col(0),1,numNodes);
128     E1 = repmat(I.col(1),1,numNodes);
129     E2 = repmat(I.col(2),1,numNodes);
```

```
130     Nodes = zeros<mat>(3,numNodes);
131     Nodes.row(0) = linspace(0,seg_length*(numNodes-1),numNodes).t();
132     W_eq = zeros<mat>(3,numNodes);
133 }
134 
135 // Initialize the nodes based on geometry and orientation
136 mat Flagellum::initNodes(void){
137 
138     double T = length / sqrt(helix_radius*helix_radius + helix_b*helix_b);
139     vec t = linspace<vec>(0,T,numNodes);
140     mat R = zeros<mat>(3,numNodes);
141     R(0,1) = h_length;
142 
143     double ts;
144     vec tangent = zeros<vec>(3,1);
145 
146     int j;
147     for (j = 2; j < numNodes; j++) {
148         ts = 0.5 * ( t(j-1) + t(j-2) );
149         tangent(0) = helix_b;
150         tangent(1) = -helix_radius * sin(ts);
151         tangent(2) = +helix_radius * cos(ts);
152         tangent = normalise(tangent);
153         R.col(j) = R.col(j-1) + seg_length*tangent;
154 
155     }
156     return R;
157 
158 }
159 
160 void Flagellum::resetInitializers(void) {
161     // Resets placeholders in case numNodes is overloaded
162     mat I0 = zeros<mat>(3,numNodes);
163     dE1 = I0; dE2 = I0; dE3 = I0;
164     Forces = I0; Velocities = I0;
165     Forces_steric = I0; Velocities_HI = I0;
166 
167     kb = I0; U = I0; V = I0; M1 = I0; M2 = I0;
168     Beats = I0; BeatsTwist = I0.row(0);
169 }
170 
171 void Flagellum::resetMobility(void) {
172     // Reset the mobility values in case discretization is changed
173     double PI = datum::pi;
174     double ln2qr = log ( 0.18 * 2 * PI * sqrt(helix_radius * helix_radius
175         + helix_b * helix_b) / fil_radius  );
176     delta = 1.13 * seg_length / (ln2qr + 0.5);
177     zpar_inv = 3 / seg_length * ln2qr;
178     zperp_inv = 1.5 / seg_length *(ln2qr + 0.5);
179     zr_inv = 1.5 / (fil_radius * fil_radius * seg_length);
180 }
181 
182 void Flagellum::initE3(void) {
183     E3 = zeros<mat>(3,numNodes);
```

```
184     mat R = Nodes;
185     for (int j = 1; j < numNodes; j++) {
186         E3.col(j) = normalise( R.col(j) - R.col(j-1) );
187     }
188 }
189 
190 
191 void Flagellum::initE1(void) {
192     E1 = zeros<mat>(3,numNodes);
193     for (int j = 2; j < numNodes; j++) {
194         E1.col(j) = normalise( cross( E3.col(j-1), E3.col(j) ) );
195     }
196     double th = acos(dot(E3.col(1), E3.col(2)));
197     vec n = cross( E3.col(1), E3.col(2) ) / sin(th);
198     E1.col(1) = n ; //- sin(th) * cross(E3.col(0),n);
199 }
200 
201 void Flagellum::initE2(void) {
202     E2 = zeros<mat>(3,numNodes);
203     for (int j = 1; j < numNodes; j++) {
204         E2.col(j) = cross( E3.col(j), E1.col(j) );
205     }
206 }
207 
208 void Flagellum::initW_eq(void) {
209     W_eq = zeros<mat>(3,numNodes);
210     double th, phi;
211     vec n, e1_it;
212     mat I = eye<mat>(3,3);
213     for (int j = 1; j < numNodes-1; j++) {
214         th = acos( dot(E3.col(j), E3.col(j+1) ) );
215         n = cross( E3.col(j), E3.col(j+1) ) / sin(th);
216         e1_it = (n*n.t()) * E1.col(j+1) + cos(th)*(I - n*n.t()) * E1.col(j+1) - sin(th)*
    ↪ cross(n, E1.col(j));
217         phi = atan2( dot(e1_it, E2.col(j)) , dot( E1.col(j), e1_it ) );
218 
219         W_eq(0,j) = -th/sin(th) * dot(E2.col(j), E3.col(j+1));
220         W_eq(1,j) = +th/sin(th) * dot(E1.col(j), E3.col(j+1));
221         W_eq(2,j) = phi;
222     }
223 }
224 
225 mat Flagellum::getTangents(void) {
226     mat tangent = zeros<mat>(3, numNodes);
227     for (int j = 0; j < numNodes; j++) {
228         if (j == numNodes-1) {tangent.col(j) = E3.col(j);}
229         else if (j == 0) {tangent.col(j) = E3.col(j+1);}
230         else {tangent.col(j) = normalise(E3.col(j) + E3.col(j+1));}
231     }
232 
233     return tangent;
234 }
235 
236 void Flagellum::initDER_eq(void) {
```

```
237     kb = zeros<mat>(3,numNodes);
238     W_eq_left = zeros<mat>(2,numNodes);
239     W_eq_right = zeros<mat>(2,numNodes);
240     phi_eq = zeros<vec>(numNodes,1);
241     Phi = phi_eq;
242 
243     int j;
244     for (j = 1; j < numNodes-1; j++) {
245         // Curvature binormal at each node
246         kb.col(j) = 2*cross(E3.col(j), E3.col(j+1)) / (1 + dot( E3.col(j), E3.col(j+1) )
    ↪ ) ;
247         // Equilibrium curvature at each node
248         W_eq_left(0,j) = +dot( kb.col(j), E2.col(j) );
249         W_eq_left(1,j) = -dot( kb.col(j), E1.col(j) );
250         W_eq_right(0,j) = +dot( kb.col(j), E2.col(j+1) );
251         W_eq_right(1,j) = -dot( kb.col(j), E1.col(j+1) );
252     }
253     // Total twist relative to first edge.
254     double phi = 0;
255     for (j = 2; j < numNodes; j++) {
256         phi += W_eq(2,j-1);
257         phi_eq(j-1) = W_eq(2,j-1);
258         Phi(j) = phi;
259     }
260 }
261 
262 void Flagellum::getDER_kb(void) {
263     for (int j = 1; j < numNodes-1; j++) {
264         kb.col(j) = 2*cross(E3.col(j), E3.col(j+1)) / (1 + dot( E3.col(j), E3.col(j+1) )
    ↪ ) ;
265     }
266 }
267 
268 void Flagellum::getDER_Bishop(vec u1) {
269     // Assume all material frames are already initialized
270     // Align the first Bishop frame with the first material frame
271 
272     double phi; vec n; mat I = eye<mat>(3,3);
273     U.col(1) = u1;
274     int j;
275 
276     for (j = 2; j < numNodes; j++) {
277         phi = acos( dot(E3.col(j-1), E3.col(j) )   );
278         if (phi < 1e-8) { U.col(j) = U.col(j-1); }
279         else {
280             n = normalise( cross( E3.col(j-1), E3.col(j) ) );
281             U.col(j) = ( n*n.t() + cos(phi)*(I-n*n.t()) ) * U.col(j-1) + sin(phi) * cross
    ↪ (n,U.col(j-1));
282         }
283     }
284 
285     for (j = 1; j < numNodes; j++) {
286         V.col(j) = cross(E3.col(j), U.col(j));
287     }
```

```
288
289 }
290
291 void Flagellum::getDER_matFrame(void) {
292     int j;
293     for (j = 1; j < numNodes; j++) {
294         M1.col(j) = cos(Phi(j)) * U.col(j) + sin(Phi(j)) * V.col(j);
295         M2.col(j) = cos(Phi(j)) * V.col(j) - sin(Phi(j)) * U.col(j);
296     }
297
298     for (j = 1; j < numNodes-1; j++) {
299         W_left(0,j) = +dot( kb.col(j), M2.col(j) );
300         W_left(1,j) = -dot( kb.col(j), M1.col(j) );
301         W_right(0,j) = +dot( kb.col(j), M2.col(j+1) );
302         W_right(1,j) = -dot( kb.col(j), M1.col(j+1) );
303     }
304
305 }
```

The next code shown below is the main simulation code. 3 input files are required to run this code. The file "simulation_parameters.txt" specifies the number of flagella on the swimmer, the time step size, and the total number of time steps in the first three lines. The final line overload specifies whether (0) we fully construct a swimmer at equilibrium to run the simulation or (1) obtain the initial swimmer from an external file. The file "initial_anchors.txt" specifies the location of the flagella anchors on the unit sphere. Each row of this file should be in [axis angle] format, and the total number of rows must match the number of flagella specified in "simulation_parameters.txt" as a redundant check. The file "initial_axes.txt" specifies the helical axis of each flagellum, but since we attach flagella normal to the surface at equilibrium, this file is usually identical to "initial_anchors.txt". An optional input is "change_parameters.txt," where we alter flagellar parameters according to the keyword mapping shown in the code. If this file is blank, then the default constructor is used to initialize flagella. Once the full swimmer is initialized, we call the solver function to perform the full simulation.

```
1 #include <stdio.h>
2 #include <math.h>
3
4 #include <iostream>
5 #include <armadillo>
6 #include <fstream>
```

```
7  #include <iomanip>
8  #include <map>
9  #include <string>
10
11 #include "Swimmer.hpp"
12 #include "FlagInitialize.hpp"
13 #include "SolverDER.hpp"
14
15 using namespace std;
16 using namespace arma;
17
18 int main () {
19
20     wall_clock timer;
21     timer.tic();
22     double total_time;
23
24     /************ READ IN PARAMETERS ***************/
25     double var[5];
26
27     ifstream infile("input/simulation_parameters.txt");  // construct object and open
       ↪ file
28     string line;
29
30     if (!infile) { cerr << "Error opening file!\n"; return 1; }
31
32     int i = 0;
33     string parameter; double value;
34
35     while (getline(infile, line)) {
36         istringstream iss(line);
37
38         if (!(iss >> parameter >> value)) {
39             cerr << "Invalid line, skipping.\n";
40             continue;
41         }
42         var[i] = value;
43         // cout << value << endl;
44         i++ ;
45     }
46
47     // Simulation parameters
48     int nff = var[0];
49     double dt = var[1];
50     int telem = var[2];
51     double body_rad = var[3];
52     int overload = var[4];
53
54
55     // Initialize classes for body and flagella
56     const int NFlag = nff;
57     Flagellum *flag = new Flagellum[NFlag];
58     Body body;
59
```

```
60      // Load orientations of flagella
61      mat axes;
62      axes.load("input/initial_axes.txt");
63      if (axes.n_rows != NFlag) { cerr << "Dimension␣mismatch!\n"; return 1; }
64
65      mat anchors;
66      anchors.load("input/initial_anchors.txt");
67      if (anchors.n_rows != NFlag) { cerr << "Dimension␣mismatch!\n"; return 1; }
68
69      mat Q = EulerMatrix(axes, NFlag);
70      mat QA = EulerMatrix(anchors, NFlag);
71
72
73      // Overwrite PARAMETERS here *******************************************/
74      // First try to define a map connecting keywords to class pointers
75      typedef double (Flagellum::*flag_value_t);
76      typedef map<string, flag_value_t> flag_map_t;
77
78      flag_map_t mapping;
79      mapping["fila"] = &Flagellum::A_fil;
80      mapping["hook"] = &Flagellum::A_hook;
81      mapping["motor"] = &Flagellum::T_mot;
82      mapping["helb"] = &Flagellum::helix_b;
83      mapping["length"] = &Flagellum::length;
84      mapping["seg"] = &Flagellum::seg_length;
85      mapping["h_length"] = &Flagellum::h_length;
86      mapping["Gamma"] = &Flagellum::G_twist;
87      mapping[óÀÌG_hookóÀÍ] = &Flagellum::G_hook;
88
89      flag_map_t::iterator fit;
90
91      ifstream infilep("input/change_parameters.txt");  // construct object and open file
92      string linep;
93      if (!infilep) { cerr << "Error␣opening␣file!\n"; return 1; }
94
95      int nf;
96      while (getline(infilep, linep)) {
97          istringstream iss(linep);
98
99          if (!(iss >> nf >> parameter >> value)) {
100             cerr << "Invalid␣line,␣skipping.\n";
101             continue;
102         }
103
104         // Search map for matching keywords
105         fit = mapping.find(parameter);
106         if (fit != mapping.end() ) {
107             flag[nf].*(fit->second) = value;
108
109             // Reset the number of nodes if length or seg_length changes
110             if (parameter == "seg") {
111                 flag[nf].numNodes = int (flag[nf].length/flag[nf].seg_length) + 1;
112             }
113             else if (parameter == "length") {
```

```
114                 flag[nf].numNodes = int (flag[nf].length/flag[nf].seg_length) + 1;
115             }
116 
117         }
118     }
119     // ************************************************************************
120 
121     body.radius = body_rad;
122     // Initialize the flagella with overloaded parameters
123     getinitFlag(body, flag, NFlag, Q, QA);
124     double t_sim = 0;
125 
126     cout << "Overload " << overload << endl;
127 
128     // Overload initial conditions here
129     if (overload==1) {
130         body.q.load("input/final/body_q_f");
131         body.Node.load("input/final/body_node_f");
132         mat time_f;
133         time_f.load("input/final/time_f");
134         t_sim = as_scalar(time_f);
135 
136         field<mat> Nodes, E1, E2, E3, W_eq;
137         Nodes.load("input/final/flag_nodes_f");
138         E1.load("input/final/E1_f");
139         E2.load("input/final/E2_f");
140         E3.load("input/final/E3_f");
141         W_eq.load("input/final/W_eq_f");
142 
143         for (nf = 0; nf < NFlag; nf++) {
144             flag[nf].Nodes = Nodes(nf);
145             flag[nf].E1 = E1(nf);
146             flag[nf].E2 = E2(nf);
147             flag[nf].E3 = E3(nf);
148             flag[nf].W_eq = W_eq(nf);
149         }
150     }
151 
152 
153 
154     cout << "Enter the matrix: sentinel activated." << endl;
155 
156     // Send to solver
157     solveSwimmerDER(body, flag, NFlag, t_sim, dt, telem);
158 
159 
160     //cout << ConstraintInextDER(flag[0]) << endl;
161 
162     total_time = timer.toc();
163     cout << "simulation time = " << total_time << endl;
164     return 0;
165 }
```

The final code excerpt is shown below. We first initialize data streams so that output data can be written to file, and explicitly specify the geometric and mechanical parameters used in a separate text file. In the main time integration loop, we take an unconstrained time step as our initial integration, and correct the step using projection. The remaining code shows the construction of the mobility matrix characterizing interactions of the swimmer, and auxiliary functions required to collate variables into a single vector and vice versa.

```
1  #include <stdio.h>
2  #include <math.h>
3
4  #include <iostream>
5  #include <armadillo>
6  #include <fstream>
7  #include <iomanip>
8
9  #include "SolverDER.hpp"
10 #include "ConstraintsDER.hpp"
11 #include "GradientsDER.hpp"
12 #include "SwimmerTranslation.hpp"
13 #include "SwimmerRotation.hpp"
14 #include "DynamicsCurvFil.hpp"
15 #include "FlagInitialize.hpp"
16 #include "FlowFields.hpp"
17
18 using namespace std;
19 using namespace arma;
20
21 void solveSwimmerDER(Body body, Flagellum* flag, const int NFlag, double t_sim, double dt
       ↪ , int telem)
22 {
23     ofstream ofs_nodes, ofs_bodynode, ofs_bodydyn, ofs_flagforces, ofs_parameters;
24     ofstream ofs_bodyq, ofs_energy, RG;
25     ofs_nodes.open("output/flag_nodes");
26     ofs_bodynode.open("output/body_node");
27     ofs_bodydyn.open("output/body_dyn");
28     ofs_flagforces.open("output/flag_forces");
29     ofs_parameters.open("parameters.txt");
30     ofs_bodyq.open("output/body_q");
31     ofs_energy.open("output/elastic_energy");
32
33     int nf;
34     mat CNum = zeros<mat>(1,NFlag);
35
36     for (nf = 0; nf < NFlag; nf++) {
37         // Go ahead and print parameters to file as well.
38         ofs_parameters << "[Flagellum␣␣␣␣" << nf+1 << "]\n";
39         ofs_parameters << "helix_radius␣␣␣␣" << flag[nf].helix_radius << "\n";
40         ofs_parameters << "helix_b␣␣␣␣" << flag[nf].helix_b << "\n";
41         ofs_parameters << "length␣␣␣␣" << flag[nf].length << "\n";
```

```
42          ofs_parameters << "fil_radius␣␣␣␣" << flag[nf].fil_radius << "\n";
43          ofs_parameters << "seg_length␣␣␣␣" << flag[nf].seg_length << "\n";
44          ofs_parameters << "numNodes␣␣␣␣" << flag[nf].numNodes << "\n";
45          ofs_parameters << "A_fil␣␣␣␣" << flag[nf].A_fil << "\n";
46          ofs_parameters << "A_hook␣␣␣␣" << flag[nf].A_hook << "\n";
47          ofs_parameters << "B_stretch␣␣␣␣" << flag[nf].B_stretch << "\n";
48          ofs_parameters << "G_twist␣␣␣␣" << flag[nf].G_twist << "\n";
49          ofs_parameters << "T_mot␣␣␣␣" << flag[nf].T_mot << "\n";
50          ofs_parameters << "\n";
51
52          CNum(nf) = flag[nf].numNodes;
53      }
54      ofs_parameters.close();
55      CNum.save("output/counter",raw_ascii);
56      gatherFlagMatrices(flag, NFlag, "Weq").save("output/final/W_eq_f");
57
58      // Simulation parameters
59      int tt = 0;
60
61      // Declarations and placeholders
62      mat one = eye<mat>(1,1);
63      mat EE = zeros<mat>(1,3);
64      vec positions_out;
65      mat B_inv;
66
67      vec counter = splitSwimmerCounter(flag, NFlag);
68
69      //cout << gatherConstraints(body, flag, NFlag) << endl;
70
71
72      // WRITE THE ACTUAL ALGORITHM HERE
73      for (tt = 0; tt < telem-1; tt++)
74      {
75          if (tt % 25 == 0)
76          {
77              // Calculate and store mobility matrix
78              B_inv = getMobilityMatrix(body, flag, NFlag, counter);
79
80              // Normalize triads
81              for (nf = 0; nf < NFlag; nf++) {
82                  flag[nf].E1 = normalise(flag[nf].E1);
83                  flag[nf].E2 = normalise(flag[nf].E2);
84                  flag[nf].E3 = normalise(flag[nf].E3);
85              }
86          }
87          // Different approach: use old code with fast projection
88
89          // Get the forces translational velocities on nodes, but don't integrate yet
90          getSwimmerTranslation(&body, flag, NFlag, tt);
91
92          //getSwimmerFSoloTranslation(flag, NFlag, tt, t_sim);
93
94          // Now calculate the torque and angular velocity on the body
95          getSwimmerBSoloRotation(&body, flag, NFlag);
```

```
96
97          if (tt % 200 == 0)
98          {
99              gatherFlagMatrices(flag, NFlag, "Positions").print(ofs_nodes);
100             body.q.t().raw_print(ofs_bodyq);
101             body.Node.t().raw_print(ofs_bodynode);
102             join_vert(body.Force, body.Torque).t().raw_print(ofs_bodydyn);
103             // (t_sim*one).print(myfile3);
104             // gatherFlagMatrices(flag, NFlag, "Forces").print(myfile3);
105
106             EE(0) = t_sim;
107             EE(1) = H_total(flag, NFlag);
108             EE(2) = getRadius(body, flag, NFlag);
109             EE.raw_print(ofs_energy);
110         }
111
112
113         // Do an unconstrained time step, first body, then flagella
114         body.q += dt * body.dq;
115
116         body.Node += dt * body.Velocity;
117         for (nf = 0; nf < NFlag; nf++) {
118             //cout << flag[nf].E3.col(0).t();
119             flag[nf].integratePositionsLin1(dt);
120         }
121
122         // Do fast projection to get constraint maintaining Velocity
123         FastProjection(&body, flag, NFlag, B_inv, counter, dt);
124
125         // Break the loop if the Fast Projection breaks
126         if (isnan(body.Node(0)) == 1) {
127             cout << "broken" << endl;
128             break;
129         }
130
131         getSwimmerFSoloRotation(flag, NFlag);
132
133         // Integrate triads on each flagellum
134         for (nf = 0; nf < NFlag; nf++) {
135             flag[nf].integrateLin1(dt);
136             flag[nf].E3.col(0) = normalise(flag[nf].Nodes.col(0) - body.Node);
137         }
138
139         t_sim += dt;
140
141     }
142
143     //(getRadius(body, flag, NFlag)*one).print(RG);
144     ofs_nodes.close();
145     ofs_bodynode.close();
146     ofs_bodydyn.close();
147     ofs_flagforces.close();
148     ofs_parameters.close();
149     ofs_bodyq.close();
```

```
150     ofs_energy.close();
151
152     // Save triads at final time step
153     gatherFlagMatrices(flag, NFlag, "Positions").save("output/final/flag_nodes_f");
154     gatherFlagMatrices(flag, NFlag, "E1").save("output/final/E1_f");
155     gatherFlagMatrices(flag, NFlag, "E2").save("output/final/E2_f");
156     gatherFlagMatrices(flag, NFlag, "E3").save("output/final/E3_f");
157     body.q.save("output/final/body_q_f",raw_ascii);
158     body.Node.save("output/final/body_node_f",raw_ascii);
159
160     mat t_final = t_sim*one;
161     t_final.save("output/final/time_f",raw_ascii);
162
163 }
164
165 void NewtonSolverQS(Flagellum* flag, const int NFlag)
166 {
167     vec CQS, dPhi;
168     mat H;
169     int nf;
170
171     // Apply QS approx on all flagella
172     for (nf = 0; nf < NFlag; nf++) {
173
174         CQS = ConstraintQS(flag[nf]);
175
176         while (norm(CQS) > 1e-8) {
177             mat H = HessianQS(flag[nf]);
178
179             dPhi = solve(H,CQS);
180             flag[nf].Phi.rows(2,flag[nf].numNodes-1) -= dPhi;
181
182             // Then update material frames and recalibrate
183             // Note: nodes are not changed here, so neither do U,V
184             flag[nf].getDER_matFrame();
185             CQS = ConstraintQS(flag[nf]);
186         }
187     }
188
189 }
190
191 void FastProjection(Body* body, Flagellum* flag, const int NFlag, mat B_inv, vec counter,
    ↪  double dt)
192 {
193     vec delta_L, delta_X, x0, X;
194     vec FC;
195
196     int iter = 0;
197     vec CC = gatherConstraints(*body, flag, NFlag);
198     int NC = CC.n_elem;
199     mat grad_CC;
200     x0 = gatherSwimmerPositions(*body, flag, NFlag);
201
202     while (norm(CC) > 1e-12 && iter < 100 ) {
```

```
203         grad_CC = gatherConstraintGradients(*body, flag, NFlag, counter, NC);
204
205         delta_L = solve(dt * grad_CC * B_inv * grad_CC.t() , CC);
206         //delta_X = -dt * B_inv * grad_CC.t() * delta_L;
207         FC = -grad_CC.t()*delta_L;
208         delta_X = dt * B_inv * FC;
209
210         //flag[0].Nodes += reshape(delta_X,3,flag[0].numNodes);
211         splitSwimmerVarupdate(delta_X, body, flag, NFlag, counter, "Positions");
212         splitSwimmerVarupdate(FC, body, flag, NFlag, counter, "Forces");
213
214
215         CC = gatherConstraints(*body, flag, NFlag);
216         iter++;
217         //cout << "iter = " << iter << " " << norm(CC) << endl;
218     }
219
220     // Calculate constraint maintaining velocity
221     X = gatherSwimmerPositions(*body, flag, NFlag);
222     vec delta_V = (X - x0) / dt;
223     splitSwimmerVarupdate(delta_V, body, flag, NFlag, counter, "Velocities");
224
225     if (iter > 90) {cout << "Failed to converge" << endl;}
226
227 }
228
229 mat getMobilityMatrix(Body body, Flagellum* flag, const int NFlag, vec counter)
230 {
231     int nf;
232
233     mat tangent; double zn, zt, delta; mat I = eye<mat>(3,3);
234     // Remember the 7 components needed for the body;
235     int numVar = counter(NFlag);
236     mat B_inv = zeros<mat>(numVar, numVar);
237
238     // cout << numVar << endl;
239
240     // Deal with body first
241     B_inv(0,0) = 1;
242     B_inv(span(1,3), span(1,3)) = 0.5 * body.zr_inv * I;
243     B_inv(span(4,6), span(4,6)) = body.z_inv * I;
244
245     mat pos_flags = gatherFlagPositions(flag,NFlag).t();
246     //int numFNodes = (numVar-7)/3;
247
248     int j, k; int ii = 2; int jj; int nf_k; vec xj, xjj;
249
250     for (nf = 0; nf < NFlag; nf++)
251     {
252         // nf counts the flagellum we are on
253         // j, k = intraflagellar node index
254         // ii, jj count the overall flagellar nodes
255
256         // Make sure to update the edge vectors with each projection step;
```

```
257         flag[nf].initE3();
258         flag[nf].E3.col(0) = normalise(flag[nf].Nodes.col(0) - body.Node);
259         tangent = flag[nf].getTangents();
260         zn = flag[nf].zperp_inv; zt = flag[nf].zpar_inv; delta = flag[nf].delta;
261 
262         for (j = 0; j < flag[nf].numNodes; j++)
263         {
264             xj = flag[nf].Nodes.col(j);
265             // Now get the body interaction at the non-anchoring nodes on flag[nf]
266 
267             {
268                 B_inv( span(0,3), span(3*ii+1, 3*ii+3) ) = 0.5 * QMatrixMultiply(body.q)
    ↪ * tensor_Faxen_body_rot(body.Node, xj, body.radius);
269                 B_inv( span(4,6), span(3*ii+1, 3*ii+3) ) = tensor_Faxen_body( body.Node,
    ↪ xj, body.radius );
270 
271             B_inv( span(3*ii+1, 3*ii+3), span(0,3) ) = 0.5 * tensor_Faxen_body_rot(xj, body.
    ↪ Node, body.radius) * QMatrixMultiply(body.q).t();
272             B_inv( span(3*ii+1, 3*ii+3), span(4,6) ) = tensor_Faxen_body( body.Node, xj, body
    ↪ .radius );
273 
274             }
275 
276             // Now sum over the other flagellar nodes to get HI
277             jj = 2;
278             for (nf_k = 0; nf_k < NFlag; nf_k++) {
279                 for (k = 0; k < flag[nf_k].numNodes; k++) {
280                     // Do not count H.I. involving anchor points
281                     // cout << k << endl;
282                     // if (k == 0) { jj++; }
283                     if (nf_k == nf && k == j) {
284                         B_inv(span(3*ii+1, 3*ii+3), span(3*ii+1,3*ii+3)) = zn*I + (zt-zn)
    ↪ *tangent.col(j)*tangent.col(j).t();
285                         jj++;
286                     }
287                     else {
288                         xjj = pos_flags.rows(3*(jj-2), 3*(jj-2)+2);
289                         B_inv(span(3*ii+1, 3*ii+3), span(3*jj+1,3*jj+3)) = 0.75 *
    ↪ reg_Stokeslet(xj, xjj, delta);
290                         jj++;
291                     }
292                 }
293             }
294             ii++;
295         }
296     }
297     return B_inv;
298 }
299 
300 vec gatherConstraints(Body body, Flagellum* flag, const int NFlag)
301 {
302     // 1 Euler constraint
303     // For each flagellum, there are numNodes + 2 constraints
304 
```

```
305     int NC = 1; int nf;
306     for (nf = 0; nf < NFlag; nf++) {
307         NC += flag[nf].numNodes + 2;
308     }
309 
310     vec CC = zeros<vec>(NC,1);
311     int ix = 1;
312     CC(0) = ConstraintEuler(body);
313 
314     for (nf = 0; nf < NFlag; nf++) {
315         CC.rows(ix, ix+2) = ConstraintRBMDER(body,flag[nf]);
316         CC.rows(ix+3, ix+3+flag[nf].numNodes-2) = ConstraintInextDER(flag[nf]);
317         ix = ix + 3 + flag[nf].numNodes-1;
318     }
319 
320     /* Add in clamping if needed (7 equations) and remove redundant constraint */
321     //CC = join_vert( CC, ConstraintBodyClamp(body) );
322     //CC.shed_row(0);
323 
324     return CC;
325 }
326 
327 mat gatherConstraintGradients(Body body, Flagellum* flag, const int NFlag, vec counter,
        ↪ int NC)
328 {
329     // Count variables properly: 7 for body, and 3*numNodes for each flagellum
330     int numVar = counter(NFlag);
331     int nf;
332 
333     mat grad = zeros<mat>(NC,numVar);
334     mat grad_RBM;
335     int ix = 1;
336     // ix is normally 1. ix is zero if removing redundant constraint
337 
338     grad(span(0,0), span(0,3)) = grad_ConstraintEuler(body);
339     for (nf = 0; nf < NFlag; nf++) {
340         grad_RBM = grad_ConstraintRBMDER(body, flag[nf]);
341         grad(span(ix,ix+2), span(0,6)) = grad_RBM.cols(0,6);
342         grad(span(ix,ix+2), span(counter(nf),counter(nf)+2)) = grad_RBM.cols(7,9);
343         grad(span(ix+3, ix+3+flag[nf].numNodes-2),span(counter(nf),counter(nf) + 3*flag[
        ↪ nf].numNodes-1)) = grad_ConstraintInextDER(flag[nf]);
344         ix = ix + 3 + flag[nf].numNodes-1;
345     }
346 
347     /* Add derivatives of clamping equations if needed (7 rows) and remove redundant
        ↪ constraint */
348     //grad( span(NC-7, NC-1), span(0,6) ) = grad_ConstraintBodyClamp(body);
349     //grad.shed_row(0);
350 
351     return grad;
352 }
353 
354 double getRadius(Body body, Flagellum* flag, const int NFlag)
355 {
```

```
356     double Rg = 0;
357     double a;
358     int j, nf; int N = 0;
359     for (nf = 0; nf < NFlag; nf++)
360     {
361         N += flag[nf].numNodes;
362         for (j = 0; j < flag[nf].numNodes; j++)
363         {
364             a = norm(flag[nf].Nodes.col(j) - body.Node);
365             Rg += a * a;
366         }
367     }
368     Rg = sqrt(Rg/N);
369     return Rg;
370 }
```

References

[1] T. C. Adhyapak and H. Stark. Zipping and entanglement in flagellar bundle of E. coli: Role of motile cell body. *Phys. Rev. E*, 92(5):052701, Nov. 2015.

[2] T. C. Adhyapak and H. Stark. Dynamics of a bacterial flagellum under reverse rotation. *Soft Matter*, 12:5621–5629, June 2016.

[3] J. Ainley, S. Durkin, R. Embid, P. Boindala, and R. Cortez. The method of images for regularized Stokeslets. *Journal of Computational Physics*, 227(9):4600–4616, 2008.

[4] J. Ali, U. K. Cheang, J. D. Martindale, M. Jabbarzadeh, H. C. Fu, and M. Jun Kim. Bacteria-inspired nanorobots with flagellar polymorphic transformations and bundling. *Scientific Reports*, 7(1):1–10, 2017.

[5] S. A. Allison and J. A. McCammon. Transport properties of rigid and flexible macromolecules by brownian dynamics simulation. *Biopolymers*, 23(1):167–187, 1984.

[6] S. V. Anand and T. A. Saif. Emergent dynamics of cardiomyocyte clusters on deformable polymeric substrates. *Extreme Mechanics Letters*, 8:1–5, 2016.

[7] J. E. Avron, O. Kenneth, and D. H. Oaknin. Pushmepullyou: an efficient microswimmer. *New Journal of Physics*, 7:234–42, nov 2005.

[8] R. H. Baltz, A. L. Demain, and J. E. Davies. *Manual of Industrial Microbiology and Biotechnology*. American Society Mic Series. ASM Press, 2010.

[9] G. K. Batchelor. The stress system in a suspension of force-free particles. *J. Fluid. Mech.*, 41(03):545–570, mar 1970.

[10] L. E. Becker, S. a. Koehler, and H. a. Stone. On self-propulsion of micro-machines at low Reynolds number: Purcells three-link swimmer. *J. Fluid. Mech.*, 490:15–35, sep 2003.

[11] H. C. Berg. The rotary motor of bacterial flagella. *Annu. Rev. Biochem.*, 72:19–54, jan 2003.

[12] H. C. Berg. *E. coli in Motion*. Biological and Medical Physics, Biomedical Engineering. Springer, 2004.

[13] H. C. Berg. Cell motility: Turning failure into function. *Nature Physics*, 9(8):460–461, jul 2013.

[14] M. Bergou, M. Wardetzky, S. Robinson, B. Audoly, and E. Grinspun. Discrete elastic rods. *ACM Trans. Graph.*, 27(3):1, 2008.

[15] J. R. Blake. A note on the image system for a stokeslet in a no-slip boundary. *Mathematical Proceedings of the Cambridge Philosophical Society*, 70(02):303–10, oct 2008.

[16] C. Brennen and H. Winet. Fluid mechanics of propulsion by cilia and flagella. *Annual Review of Fluid Mechanics*, 9:339–98, 1977.

[17] S. Bubendorfer, M. Koltai, F. Rossmann, V. Sourjik, and K. M. Thormann. Secondary bacterial flagellar system improves bacterial spreading by increasing the directional persistence of swimming. *Proc. Nat. Acad. Sci. U.S.A.*, 111(31):11485–11490, 2014.

[18] S. Bubendorfer, M. Koltai, F. Rossmann, V. Sourjik, and K. M. Thormann. Secondary bacterial flagellar system improves bacterial spreading by increasing the directional persistence of swimming. *Proc. Nat. Acad. Sci. U.S.A.*, 111(31):11485–11490, 2014.

[19] V. Chan, K. Park, M. B. Collens, H. Kong, T. A. Saif, and R. Bashir. Development of Miniaturized Walking Biological Machines. *Sci. Rep.*, 2, Nov. 2012.

[20] S. Chattopadhyay, R. Moldovan, C. Yeung, and X. L. Wu. Swimming efficiency of bacterium Escherichia coli. *Proc. Nat. Acad. Sci. U.S.A.*, 103(37):13712–7, sep 2006.

[21] O. Cohen and S. A. Safran. Elastic interactions synchronize beating in cardiomyocytes. *Soft Matter*, 12(28):6088–6095, 2016.

[22] O. Cohen and S. A. Safran. Theory of frequency response of mechanically driven cardiomyocytes. *Scientific Reports*, 8(1):1–8, 2018.

[23] R. Cortez and D. Varela. A general system of images for regularized Stokeslets and other elements near a plane wall. *Journal of Computational Physics*, 285:41–54, 2015.

[24] R. Cortez, L. Fauci, and A. Medovikov. The method of regularized Stokeslets in three dimensions: Analysis, validation, and application to helical swimming. *Physics of Fluids*, 17(3):031504, 2005.

[25] N. C. Darnton and H. C. Berg. Force-Extension Measurements on Bacterial Flagella: Triggering Polymorphic Transformations. *Biophys. J.*, 92(6):2230–2236, Jan. 2007.

[26] N. C. Darnton, L. Turner, S. Rojevsky, and H. C. Berg. On torque and tumbling in swimming Escherichia coli. *J. Bacteriol.*, 189(5):1756–1764, 2007.

[27] Y. Ding, F. Qiu, X. Casadevall i Solvas, F. W. Y. Chiu, B. J. Nelson, and A. De Mello. Microfluidic-based droplet and cell manipulations using artificial bacterial flagella. *Micromachines*, 7(2):1–13, 2016.

[28] C. Dombrowski, L. Cisneros, S. Chatkaew, R. E. Goldstein, and J. O. Kessler. Self-Concentration and Large-Scale Coherence in Bacterial Dynamics. *Phys. Rev. Lett.*, 93 (9):0981031–4, aug 2004.

[29] K. Drescher, R. E. Goldstein, N. Michel, M. Polin, and I. Tuval. Direct measurement of the flow field around swimming microorganisms. *Phys. Rev. Lett.*, 105(16):1–4, 2010.

[30] R. Dreyfus, J. Baudry, M. L. Roper, M. Fermigier, H. A. Stone, and J. Bibette. Microscopic artificial swimmers. *Nature*, 437(7060):862–865, Oct. 2005.

[31] M. Eisenbach. Bacterial Chemotaxis. *Encyclopedia of Life Sciences*, (May), 2001.

[32] G. Elfring and E. Lauga. Hydrodynamic Phase Locking of Swimming Microorganisms. *Phys. Rev. Lett.*, 103(8):088101, aug 2009.

[33] L. FAUCI and R. Dillon. Biofluidmechanics of reproduction. *Annu Rev Fluid Mech*, 38:371–394, Jan. 2006.

[34] O. Felfoul, M. Mohammadi, S. Taherkhani, D. de Lanauze, Y. Z. Xu, D. Loghin, S. Essa, S. Jancik, D. Houle, M. Lafleur, L. Gaboury, M. Tabrizian, N. Kaou, M. Atkin, T. Vuong, G. Batist, N. Beauchemin, D. Radzioch, and S. Martel. Magneto-aerotactic bacteria deliver drug-containing nanoliposomes to tumour hypoxic regions. *Nature Nanotechnology*, 11(August):941–949, 2016.

[35] T. Fujii, T. Kato, and K. Namba. Specific Arrangement of alpha-Helical Coiled Coils in the Core Domain of the Bacterial Flagellar Hook for the Universal Joint Function. *Structure*, 17(11):1485–1493, 2009.

[36] R. Goldenthal, D. Harmon, R. Fattal, M. Bercovier, and E. Grinspun. Efficient simulation of inextensible cloth. *ACM Trans. Graph.*, 26(3):49, 2007.

[37] T. Goto, K. Nakata, K. Baba, M. Nishimura, and Y. Magariyama. A fluid-dynamic interpretation of the asymmetric motion of singly flagellated bacteria swimming close to a boundary. *Biophys. J.*, 89(6):3771–9, dec 2005.

[38] R. Gotzt and R. Schmitt. Rhizobium meliloti swims by unidirectional , intermittent rotation of right-handed flagellar helices. *Journal of Bacteriology*, 169(7):3146–3150, 1987.

[39] B. Y. J. Gray and G. J. Hancock. The propulsion of sea-urchin spermatozoa. *Journal of Experimental Biology*, 32:802–14, 1955.

[40] D. C. Guellii, H. B. Ii, R. B. Frankelt, and H. Hartman. Hydrodynamic Forces and Band Formation in Swimming Magnetotactic Bacteria. *Journal of Theoretical Biology*, 135(4):525–542, 1988.

[41] G. J. Hancock. The Self-Propulsion of Microscopic Organisms through Liquids. *Proc. R. Soc. London, Ser. A*, 217(1128):96–121, mar 1953.

[42] Y. Hatwalne, S. Ramaswamy, M. Rao, and R. Simha. Rheology of Active-Particle Suspensions. *Phys. Rev. Lett.*, 92(11):118101, mar 2004.

[43] J. P. Hernandez-Ortiz, C. G. Stoltz, and M. D. Graham. Transport and Collective Dynamics in Suspensions of Confined Swimming Particles. *Phys. Rev. Lett.*, 95(20): 204501, nov 2005.

[44] C. Hohenegger and M. J. Shelley. *Dynamics of complex biofluids*. 2011.

[45] H.-W. Huang, M. S. Sakar, A. J. Petruska, S. Pane, and B. J. Nelson. Soft micromachines with programmable motility and morphology. *Nat Commun*, 7:12263, 2016.

[46] Y. Hyon, Marcos, T. R. Powers, R. Stocker, and H. C. Fu. The wiggling trajectories of bacteria. *J. Fluid Mech.*, 705(-1):58–76, Aug. 2012.

[47] T. Ishikawa and M. Hota. Interaction of two swimming Paramecia. *The Journal of experimental biology*, 209(22):4452–63, nov 2006.

[48] T. Ishikawa and T. J. Pedley. The rheology of a semi-dilute suspension of swimming model micro-organisms. *J. Fluid. Mech.*, 588:399–435, sep 2007.

[49] T. Ishikawa, M. P. Simmonds, and T. J. Pedley. Hydrodynamic interaction of two swimming model micro-organisms. *J. Fluid. Mech.*, 568:119–160, nov 2006.

[50] T. Ishikawa, G. Sekiya, Y. Imai, and T. Yamaguchi. Hydrodynamic interactions between two swimming bacteria. *Biophys. J.*, 93(6):2217–25, sep 2007.

[51] M. Jabbarzadeh and H. C. Fu. Dynamic instability in the hook-flagellum system that triggers bacterial flicks. *Phys. Rev. E*, 97:012402, Jan 2018.

[52] P. J. A. Janssen and M. D. Graham. Coexistence of tight and loose bundled states in a model of bacterial flagellar dynamics. *Phys. Rev. E*, 84:011910, 2011.

[53] M. K. Jawed, N. K. Khouri, F. Da, E. Grinspun, and P. M. Reis. Propulsion and Instability of a Flexible Helical Rod Rotating in a Viscous Fluid. *Phys. Rev. Lett.*, 115 (16):168101, 2015.

[54] L. O. Jay. Solution of index 2 implicit differential-algebraic equations by Lobatto Runge-Kutta methods. *BIT Numerical Mathematics*, 43(1):93–106, 2003.

[55] R. E. Johnson and C. Brokaw. Flagellar hydrodynamics. A comparison between resistive-force theory and slender-body theory. *Biophys. J.*, 25(1):113–127, 1979.

[56] D. F. Katz. On the propulsion of micro-organisms near solid boundaries. *J. Fluid. Mech.*, 64(1):33–49, 1974.

[57] D. F. Katz and J. R. Blake. On the movement of slender bodies near plane boundaries at low Reynolds number. *J. Fluid. Mech.*, 72(3):529–540, 1975.

[58] M. Kim and T. R. Powers. Hydrodynamic interactions between rotating helices. *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics*, 69(6 1):1–5, 2004.

[59] S. Kim, S. J. Karrila, and H. Brenner. *Microhydrodynamics: Principles and Selected Applications*. Butterworth-Heinemann series in chemical engineering. Elsevier Science, 2013.

[60] S. M. Kirov, M. Castrisios, and J. G. Shaw. Aeromonas Flagella (Polar and Lateral) Are Enterocyte Adhesins That Contribute to Biofilm Formation on Surfaces. *Infection and Immunity*, 72(4):1939–1945, 2004.

[61] W. Ko, S. Lim, W. Lee, Y. Kim, H. C. Berg, and C. S. Peskin. Modeling polymorphic transformation of rotating bacterial flagella in a viscous fluid. *Physical Review E*, 95 (6):1–12, 2017.

[62] M. J. Kühn, F. K. Schmidt, B. Eckhardt, and K. M. Thormann. Bacteria exploit a polymorphic instability of the flagellar filament to escape from traps. *Proc. Nat. Acad. Sci. U.S.A.*, 114(24):6340–6345, June 2017.

[63] E. Lauga. Bacterial hydrodynamics. *Annu. Rev. of Fluid Mech.*, 48(1):105–130, 2016.

[64] E. Lauga and D. Bartolo. No many-scallop theorem: Collective locomotion of reciprocal swimmers. *Physical Review E*, 78(3):030901, sep 2008.

[65] E. Lauga and T. R. Powers. The hydrodynamics of swimming microorganisms. *Rep. Prog. Phys.*, 72(9):096601, sep 2009.

[66] E. Lauga, W. R. DiLuzio, G. M. Whitesides, and H. a. Stone. Swimming in circles: motion of bacteria near solid boundaries. *Biophys. J.*, 90(2):400–12, jan 2006.

[67] E. Leifson, B. J. Cosenza, R. Murchelano, and R. C. Cleverdon. MOTILE MARINE BACTERIA I. : Techniques, Ecology, and General Characteristics. *J. Bacteriol.*, 87 (3):652–666, mar 1964.

[68] G. Li and J. X. Tang. Low flagellar motor torque and high swimming efficiency of Caulobacter crescentus swarmer cells. *Biophys. J.*, 91(7):2726–34, oct 2006.

[69] J. Lighthill. Flagellar Hydrodynamics. *SIAM Review*, 18(2):161–230, apr 1976.

[70] B. Liu, T. R. Powers, and K. S. Breuer. Force-free swimming of a model helical flagellum in viscoelastic fluids. *Proceedings of the National Academy of Sciences*, 108 (49):19516–19520, 2011.

[71] B. Liu, M. Gulino, M. Morse, J. X. Tang, T. R. Powers, and K. S. Breuer. Helical motion of the cell body enhances Caulobacter crescentus motility. *Proc. Nat. Acad. Sci. U.S.A.*, 111(31):11252–11256, 2014.

[72] J. Liu, R. M. Ford, and J. A. Smith. Idling Time of Motile Bacteria Contributes to Retardation and Dispersion in Sand Porous Medium. *Environ Sci Technol*, 45(9): 3945–3951, 2011.

[73] V. Magdanz, M. Medina-Sánchez, L. Schwarz, H. Xu, J. Elgeti, and O. G. Schmidt. Spermatozoa as Functional Components of Robotic Microswimmers. *Advanced Materials*, 29(24):1–18, 2017.

[74] M. Medina-Sánchez, L. Schwarz, A. K. Meyer, F. Hebenstreit, and O. G. Schmidt. Cellular Cargo Delivery: Toward Assisted Fertilization by Sperm-Carrying Micromotors. *Nano Lett.*, 16(1):555–561, Jan. 2016.

[75] N. H. Mendelson, A. Bourque, K. Wilkening, K. R. Anderson, and J. C. Watkins. Organized Cell Swimming Motions in. *Journal of Bacteriology*, 181(2):600–609, 1999.

[76] A. Najafi and R. Golestanian. Simple swimmer at low Reynolds number: Three linked spheres. *Physical Review E*, 69(6):062901, jun 2004.

[77] F. T. M. Nguyen and M. D. Graham. Buckling instabilities and complex trajectories in a simple model of uniflagellar bacteria. *Biophys. J.*, 112:1–13, 2017.

[78] I. Nitsan, S. Drori, Y. E. Lewis, S. Cohen, and S. Tzlil. Mechanical communication in cardiac cell synchronized beating. *Nature Physics*, 12(5), 2016.

[79] N. Phan-Thien and T. Tran-Cong. A boundary-element analysis of flagellar propulsion. *J. Fluid Mech.*, 1987.

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

[81] E. M. Purcell. Am J Phys 1977 PURCELL.pdf. *American Journal of Physics*, 45(1): 3–11, 1977.

[82] B. Qian, H. Jiang, D. a. Gagnon, K. S. Breuer, and T. R. Powers. Minimal model for synchronization induced by hydrodynamic interactions. *Physical Review E*, 80(6): 061919, dec 2009.

[83] J. I. Quelas, M. J. Althabegoiti, C. Jimenez-Sanchez, A. A. Melgarejo, V. I. Marconi, E. J. Mongiardini, S. A. Trejo, F. Mengucci, J.-J. Ortega-Calvo, and A. R. Lodeiro. Swimming performance of Bradyrhizobium diazoefficiens is an emergent property of its two flagellar systems. *Sci. Rep.*, 6:23841, Apr. 2016.

[84] M. Ramia, D. L. Tullock, and N. Phan-Thien. The role of hydrodynamic interaction in the locomotion of microorganisms. *Biophys. J.*, 65(2):755–78, aug 1993.

[85] M. Reichert. *Hydrodynamic Interactions in Colloidal and Biological Systems, Dissertation.* PhD thesis, University Konstanz, 2006.

[86] L. Ricotti and T. Fujie. Thin polymeric films for building biohybrid microrobots. *Bioinspiration and Biomimetics*, 12(2), 2017.

[87] L. Ricotti, B. Trimmer, A. W. Feinberg, R. Raman, K. K. Parker, R. Bashir, M. Sitti, S. Martel, P. Dario, and A. Menciassi. Biohybrid actuators for robotics: A review of devices actuated by living cells. *Science Robotics*, 2(12):eaaq0495, 2017.

[88] E. E. Riley, D. Das, and E. Lauga. Swimming of peritrichous bacteria is enabled by an elastohydrodynamic instability. (2000):1–6, 2018.

[89] B. Rodenborn, C.-h. Chen, H. L. Swinney, B. Liu, and H. P. Zhang. Propulsion of microorganisms by a helical flagellum. *Proceedings of the National Academy of Sciences*, 110(5):338–47, 2013.

[90] B. Rodenborn, C.-H. Chen, H. L. Swinney, B. Liu, and H. P. Zhang. Propulsion of microorganisms by a helical flagellum. *Proc. Nat. Acad. Sci. U.S.A.*, 110(5):E338–E347, 2013.

[91] R. F. Ross and D. J. Klingenberg. Dynamic simulation of flexible fibers composed of linked rigid bodies. *The Journal of Chemical Physics*, 106(7):2949–60, 1997.

[92] R. Rusconi, J. S. Guasto, and R. Stocker. Bacterial transport suppressed by fluid shear. *Nature Physics*, 10(3):212–217, feb 2014.

[93] D. Saintillan and M. J. Shelley. Active suspensions and their nonlinear models. *Comptes Rendus Physique*, 14(6):497–517, jun 2013.

[94] C. F. Schmid, L. H. Switzer, and D. J. Klingenberg. Simulations of fiber flocculation: Effects of fiber properties and interfiber friction. *Journal of Rheology*, 44(4):781–809, 2000.

[95] A. Sen, R. K. Nandy, and A. N. Ghosh. Elasticity of flagellar hooks. *J. Elect. Microsc.*, 53(3):305–309, 2004.

[96] H. Shum and E. A. Gaffney. The effects of flagellar hook compliance on motility of monotrichous bacteria: A modeling study. *Phys. Fluids*, 24(6):061901–061901–19, 2012.

[97] H. Shum, E. A. Gaffney, and D. J. Smith. Modelling bacterial behaviour close to a no-slip plane boundary : the influence of. *Proc. R. Soc. London, Ser. A*, 466(December 2009):1725–1748, 2010.

[98] K. Son, J. S. Guasto, and R. Stocker. Bacteria can exploit a flagellar buckling instability to change direction. *Nat. Phys.*, 9(8):494–498, July 2013.

[99] S. E. Spagnolie and E. Lauga. Hydrodynamics of self-propulsion near a boundary: predictions and accuracy of far-field approximations. *J. Fluid. Mech.*, 700:105–147, 2012.

[100] S. E. Spagnolie, B. Liu, and T. R. Powers. Locomotion of helical bodies in viscoelastic fluids: Enhanced swimming at large helical amplitudes. *Phys. Rev. Lett.*, 111(6), 2013.

[101] S. E. Spagnolie, G. R. Moreno-Flores, D. Bartolo, and E. Lauga. Geometric capture and escape of a microswimmer colliding with an obstacle. *Soft Matter*, 11(17):3396–3411, 2015.

[102] M. M. Stanton, C. Trichet-Paredes, and S. Sánchez. Applications of three-dimensional (3D) printing for microswimmers and bio-hybrid robotics. *Lab on a Chip*, 15(7):1634–1637, 2015.

[103] M. M. Stanton, B. W. Park, A. Miguel-López, X. Ma, M. Sitti, and S. Sánchez. Bio-hybrid Microtube Swimmers Driven by Single Captured Bacteria. *Small*, 13(19):1–10, 2017.

[104] D. Tam and A. E. Hosoi. Optimal Stroke Patterns for Purcell's Three-Link Swimmer. *Phys. Rev. Lett.*, 98(6):4, Feb. 2007.

[105] D. Tam and A. E. Hosoi. Optimal feeding and swimming gaits of biflagellated organisms. *Proc. Nat. Acad. Sci. U.S.A.*, 108(3):1001–1006, 2011.

[106] G. Taylor. Analysis of the Swimming of Microscopic Organisms. *Proceedings of the Royal Society of London A: Mathematical and Physical Sciences*, 209(1099):447–461, 1951.

[107] G. Taylor. The Action of Waving Cylindrical Tails in Propelling Microscopic Organisms. *Proc. R. Soc. London, Ser. A*, 211(1105):225–239, feb 1952.

[108] S. Timoshenko. *Theory of Elastic Stability. 2.ed.* McGraw-Hill, 1961.

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

[110] R. Trouilloud, T. Yu, a. Hosoi, and E. Lauga. Soft Swimming: Exploiting Deformable Interfaces for Low Reynolds Number Locomotion. *Phys. Rev. Lett.*, 101(4):048102, jul 2008.

[111] Y. Tu. Biomolecular switches: Driven to peak. *Nature Physics*, 13(7):631–632, 2017.

[112] L. Turner, W. Ryu, and H. BERG. Real-time imaging of fluorescent flagellar filaments. *J. Bacteriol*, 182(10):2793–2801, 2000.

[113] R. Vogel and H. Stark. Force-extension curves of bacterial flagella. *European Physical Journal E*, 33(3):259–271, 2010.

[114] R. Vogel and H. Stark. Force-extension curves of bacterial flagella. *Eur Phys J E*, 33 (3):259–271, 2010.

[115] R. Vogel and H. Stark. Motor-driven bacterial flagella and buckling instabilities. *Eur. Phys. J. E*, 35(2):15, Feb. 2012.

[116] H. Wada and R. R. Netz. Model for self-propulsive helical filaments: Kink-pair propagation. *Phys. Rev. Lett.*, 99(10):5–8, 2007.

[117] H. Wada and R. R. Netz. Hydrodynamics of helical-shaped bacterial motility. *Phys. Rev. E*, 80(2):021921, Jan. 2009.

[118] F. Wang, H. Shi, R. He, R. Wang, R. Zhang, and J. Yuan. Non-equilibrium effect in the allosteric regulation of the bacterial flagellar switch. *Nature Physics*, 13(7): 710–714, 2017.

[119] N. Watari and R. G. Larson. The hydrodynamics of a run-and-tumble bacterium propelled by polymorphic helical flagella. *Biophys. J.*, 98(1):12–17, 2010.

[120] N. Watari and R. G. Larson. The Hydrodynamics of a Run-and-Tumble Bacterium Propelled by Polymorphic Helical Flagella. *Biophys. J.*, 98(1):12–17, June 2010.

[121] C. H. Wiggins, D. Riveline, A. Ott, and R. E. Goldstein. Trapping and wiggling: Elastohydrodynamics of driven microfilaments. *Biophysical Journal*, 74(2 I):1043–1060, 1998.

[122] B. J. Williams, S. V. Anand, J. Rajagopalan, and M. T. A. Saif. A self-propelled biohybrid swimmer at low Reynolds number. *Nat Comms*, 5, Jan. 2014.

[123] J. K. Wrobel, R. Cortez, and L. Fauci. Modeling viscoelastic networks in Stokes flow. *Physics of Fluids*, 26(11), 2014.

[124] J. K. Wróbel, S. Lynch, A. Barrett, L. Fauci, and R. Cortez. Enhanced flagellar swimming through a compliant viscoelastic network in Stokes flow. *J. Fluid Mech.*, 792:775–797, Mar. 2016.

[125] C. Zhang, J. Wang, W. Wang, N. Xi, Y. Wang, and L. Liu. Modeling and analysis of bio-syncretic micro-swimmers for cardiomyocyte-based actuation. *Bioinspiration and Biomimetics*, 11(5):1–13, 2016.

[126] L. Zhang, J. J. Abbott, L. Dong, K. E. Peyer, B. E. Kratochvil, H. Zhang, C. Bergeles, and B. J. Nelson. Characterizing the swimming properties of artificial bacterial flagella. *Nano Letters*, 9(10):3663–3667, 2009.