

## MICROROM: AN EFFICIENT AND ACCURATE REDUCED ORDER METHOD TO SOLVE MANY-QUERY PROBLEMS IN MICRO-MOTILITY

NICOLA GIULIANI<sup>1</sup>, MARTIN W. HESS<sup>1</sup>, ANTONIO DESIMONE<sup>1,2</sup>  
AND GIANLUIGI ROZZA<sup>1,\*</sup> 

**Abstract.** In the study of micro-swimmers, both artificial and biological ones, many-query problems arise naturally. Even with the use of advanced high performance computing (HPC), it is not possible to solve this kind of problems in an acceptable amount of time. Various approximations of the Stokes equation have been considered in the past to ease such computational efforts but they introduce non-negligible errors that can easily make the solution of the problem inaccurate and unreliable. Reduced order modeling solves this issue by taking advantage of a proper subdivision between a computationally expensive offline phase and a fast and efficient online stage. This work presents the coupling of Boundary Element Method (BEM) and Reduced Basis (RB) Reduced Order Modeling (ROM) in two models of practical interest, obtaining accurate and reliable solutions to different many-query problems. Comparisons of standard reduced order modeling approaches in different simulation settings and a comparison to typical approximations to Stokes equations are also shown. Different couplings between a solver based on a HPC boundary element method for micro-motility problems and reduced order models are presented in detail. The methodology is tested on two different models: a robotic-bacterium-like and an Eukaryotic-like swimmer, and in each case two resolution strategies for the swimming problem, the split and monolithic one, are used as starting points for the ROM. An efficient and accurate reconstruction of the performance of interest is achieved in both cases proving the effectiveness of our strategy.

**Mathematics Subject Classification.** 65N38, 76D07, 65D99.

Received August 7, 2020. Accepted April 15, 2022.

### 1. INTRODUCTION AND MOTIVATION

Many interesting phenomena depend on the swimming behavior of motile cells, so there is a growing need of accurate, reliable and efficient computational methods to be applied in the study of micro-organisms swimming in a fluid. Understanding the behavior of micro-swimmers can give interesting insights on many complex biological processes: the spread of a pathogen [32], the reproductive efficiency of sperm cells [22,31], the ability to change a motility strategy depending on the environment [45,48,58], and many others. An accurate and reliable simulation of the swimming mechanisms makes also possible the rational optimization and design of artificial micro-robots, mimicking such behaviors [2, 10, 26, 34, 46, 64]. Eukaryotic swimmers use complex flagellar beatings to achieve

---

*Keywords and phrases.* Micro-motility, BEM, reduced order modeling, optimization, many-query problems.

<sup>1</sup> SISSA – International School for Advanced Studies, Via Bonomea 265, 34136 Trieste, Italy.

<sup>2</sup> Sant’Anna – Scuola Universitaria Superiore Pisa, Piazza Martiri della Libertà, 56127 Pisa, Italy.

\*Corresponding author: [grozza@sissa.it](mailto:grozza@sissa.it)

motion. For example *Chlamydomonas Reinhardtii* exploits a couple of flagella that can beat symmetrically to produce an oscillating motion [24, 35], and *Euglena Gracilis* uses a single flagellum [54, 63] executing a non planar beat.

Given the very small characteristic length scale of the problem, inertia is negligible and the fluid can be well approximated using the Stokes flow [51, 52], for which a vast variety of simulation tools exist, among the others we recall the shape optimization results of [16, 62]. Swimmers move both in bounded and unbounded domains and their swimming mechanisms usually involve very large geometrical deformations to achieve the net rigid motion [40, 51]. This kind of deformations make the use of standard Finite Element Method (FEM) very challenging. A possible alternative is the Immersed Boundary Method (IBM) which has been successfully applied to simulate micro-swimmers in bounded channels [57]. The simulation of swimmers in unbounded domains poses a challenge even for IBM. The Boundary Element Method (BEM) [61] can deal with such large deformations efficiently if compared to other simulation techniques such as Finite Element or Finite Volume Methods. While parallelized, high-performance solvers allow very accurate simulations, many-query problems like optimization and inverse problems in a repetitive computational environment still remain a challenge. Optimization procedures which address the different possible configurations and the reconstruction of the movements of a biological swimmer require to run many simulations quickly increasing the computational cost of the overall algorithm. Efficient BEM implementations based on open source High Performance Computing (HPC) libraries, are available [19–21], but given the fact that the computational cost of BEMs scales at least quadratically with the problem complexity, the overall time required by many simulations quickly becomes unbearable even on modern computational architectures. A possible solution comes from rigorous mathematical approximations of the convolution integrals defining the BEM, among the others we recall the Fast Multipole Method [21, 23], the Sparse Cardinal Sine Decomposition [1, 3] and Reduced Order Modeling (ROM) which takes advantage of the classical many-queries settings employing a decomposition between a slow offline phase and a fast online evaluation, see [55]. We aim to solve many-queries problems in micro-motility so we couple ROM with BEM.

In the present work we consider two different test-cases: the optimization of a bacterium-like artificial swimmer and the stroke reconstruction of an Eukaryotic-like organism. These examples present different challenges and they highlight the many-query scenarios that can be studied. Many simplified models exist to ease the computational costs, such as Resistive Force Theory [22, 41] or neglecting the hydrodynamic interactions of different parts of the swimmer [52], but the errors introduced are often not acceptable [19, 53].

The motility of micro-swimmers can be understood using accurate simulations of the underlying fluid-dynamics. To enable many-query computations, reduced order methods are used upon a parallelized high-performance simulation software. Many-query computations then allow to address simulation problems of larger scale, such as iterative optimization and inverse problems. Figure 1 depicts the many query scenario coming from the observation of micro-swimmer and the solution we propose.

The coupling of BEMs and Reduced Order Modeling (ROM), see [55], can reduce dramatically the computational costs of these many-query simulations. ROM have been recently coupled with Hybridisable discontinuous Galerkin Methods (HDG) to solve geometrically parametrized Stokes flows around micro-swimmers in [56]. ROM and BEM have been coupled to study the airflow around objects in 2D [42] but, as far as the authors know, this is the first time these methods are combined to solve micro-swimmer problems on arbitrary geometries in 3D. The model order reduction procedure relies on an offline-online decomposition [55]. During a computationally intensive offline phase, full-order solutions are sampled, an affine parameter-dependency of the system matrices is approximated and low-order quantities for the ROM are pre-computed. Due to the large computational load, the offline phase is typically done on a High Performance Computing cluster. The online phase is then used for fast many-query evaluations with the reduced order model.

In contrast to the standard reduced basis approach, we do not completely decouple all high-order dependencies from the reduced order model. This is to allow a maximum of geometric flexibility of the micro-swimmer. In particular, the mesh is being generated at each new parameter value during the reduced order solve. The alternative would be to approximate mesh movement in an affine fashion, but this would limit the possible application scenarios significantly. To reach the maximum possible flexibility the mesh is being generated with

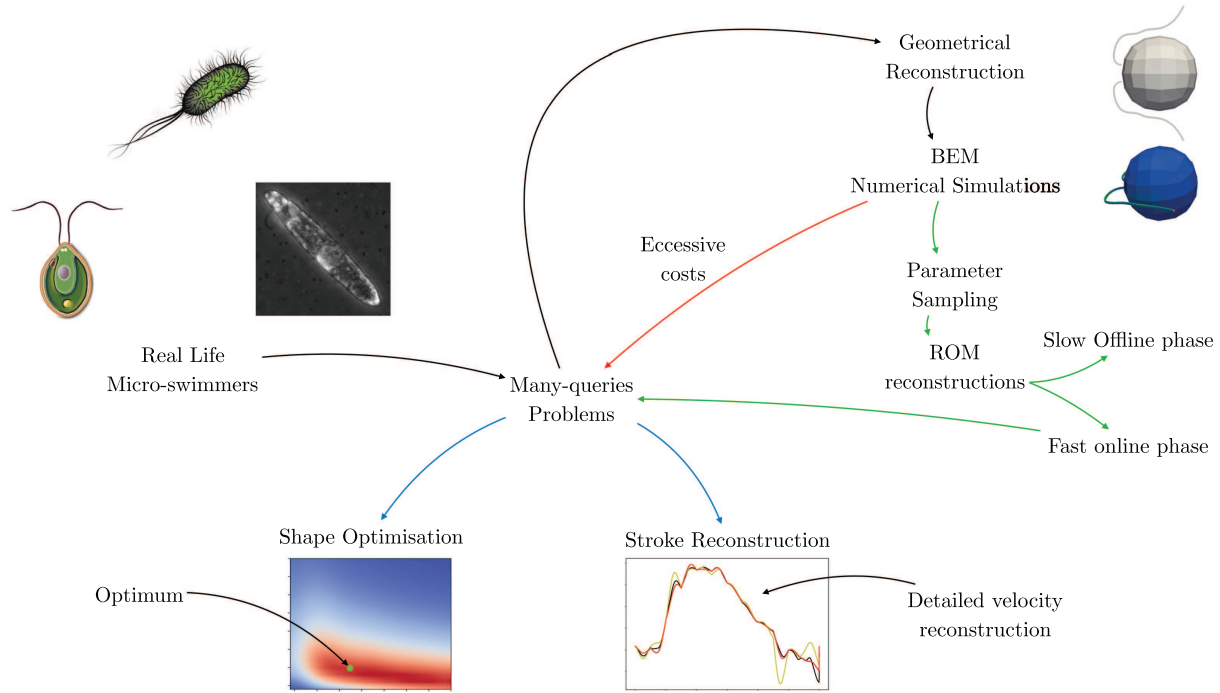


FIGURE 1. The observation of real micro-swimmers as bacteria or more complex Eukaryotic species (as *Chlamydomonas Reinhardtii* or *Euglena Gracilis*) reveals the presence of different many-query problems. We work on two of these problems: shape optimization and the detailed reconstruction of the swimming stroke. Even on modern computational architectures the direct numerical resolution of these problems is unfeasible. We propose a different approach which couples Reduced Order Modeling and the Boundary Element Method.

the software *Blender* [7]. This provides only single precision accuracy, which limits a bit the maximum accuracy that the model reduction can attain. Nevertheless we believe the use of *Blender* to be a strategic advantage since (i) it allows to generate complex arbitrary geometric parameterization in a straightforward way, (ii) it is an open source software, (iii) it has a large community of users.

The two standard approaches of reduced basis model reduction are explored, *i.e.*, the proper orthogonal decomposition (POD) and the Greedy approach [55]. The empirical interpolation method (EIM) [6] is used to approximate an affine parameter dependency of the system matrices. The focus is on the POD method, which is employed in both examples for all the full order resolution strategies. The Greedy approach is tested on one example to have a comparison with the POD available.

The Eukaryotic-like example exhibits a time-dependent boundary control. The reduced order modeling of time-depending problems usually makes use of a combined POD-Greedy approach, *i.e.*, POD in time and greedy in other parameters [27, 30, 44]. This is useful, as most time-dependent PDEs involve inertia, *i.e.*, a time-derivative. In the micro-swimming scenario, a time-derivative is not present and time can thus be treated as any other physical or geometrical parameter from the model reduction point of view.

Altogether our approach produces very accurate results while achieving a drastic reduction in computational cost. We demonstrate this result on two different test cases of interest: the optimization of artificial micro-robots and the reconstruction of the stroke starting from a reduced set of geometrical configurations.

The organization of this work is as follows. Section 2 introduces the general boundary element framework in the continuous and discrete setting, while Section 3 presents two model swimmers mimicking real-life

microorganisms. Section 4 explains the reduced modeling procedure and Section 5 presents detailed numerical results. Section 6 summarizes, concludes and provides further perspectives.

## 2. PROBLEM DESCRIPTION

We aim to solve the swimming problem, *i.e.*, to recover the swimming motion from a given history of shape changes. To do so, we define the mathematical and numerical modeling required to properly study microorganisms swimming in a fluid medium. In particular, the swimming model is composed by a kinematic part describing the possible motions for the swimmer and a dynamic part defining the conservation laws for the fluid domain. The final resolution strategy couples these two models and is solved numerically either in a monolithic approach (2.20) or in a split approach.

### 2.1. Swimming model

Following [11, 12] the swimmer is represented as a time-dependent bounded open set  $B_t \in \mathbb{R}^3$ . The map  $\chi : \bar{B}_0 \subset \mathbb{R}^3 \times [0, T] \rightarrow \mathbb{R}^3$  defines the position  $x$  at time  $t$  of a material point  $X$  of the swimmer, namely

$$x(X, t) = \chi(X, t) = q(t) + R(t)s(X, t), \quad (2.1)$$

where  $q(t)$ ,  $R(t)$ ,  $s(X, t)$  are the position of a point of the swimmer, the rotation of the system local frame and the shape at time  $t$ . The time derivative of (2.1) defines the velocity of any material point on the swimmer,

$$\begin{aligned} u_s = \dot{x} &= \frac{\partial \chi(X, t)}{\partial t} \\ &= \frac{dq}{dt} + R(t) \frac{\partial s(X, t)}{\partial t} + \frac{dR(t)}{dt} s(X, t) \\ &= \dot{q}(t) + R(t) \dot{s}(X, t) + \omega(t) \wedge (R(t)s(X, t)). \end{aligned} \quad (2.2)$$

We assume  $s(X, t)$  to be known, we define “shape velocity” the quantity  $R(t)\dot{s}(X, t)$ , therefore the unknowns are the rigid movements  $q(t)$ ,  $R(t)$ , which define the linear and angular velocities  $\dot{q}(t)$ ,  $\omega(t)$ . In the case of self-propelled swimmers, only viscous drag is acting on the swimmer so the usual momentum balance laws read

$$\int_{\Gamma} f(x) d\gamma(x) = 0, \quad (2.3a)$$

$$\int_{\Gamma} f(x) \wedge (x - x_0) d\gamma(x) = 0. \quad (2.3b)$$

The viscous tractions  $f$  are given by the action of the Cauchy stress tensor  $\sigma$ , see [25], namely

$$f = \sigma(u, p)n, \quad (2.4)$$

where  $n$  is the outer unit normal vector to the surface, and  $u, p$  represent the velocity and the pressure in the fluid.

### 2.2. Fluid model

We study micro-swimmers immersed in a Newtonian incompressible fluid, by writing the non-dimensional formulation of the incompressible Navier–Stokes equation in a generic domain  $\Omega$ , see [25], as

$$\nabla \cdot u = 0 \quad \text{in } \Omega \quad (2.5a)$$

$$\text{Re} \left( \bar{\sigma} \frac{\partial u}{\partial t} + (\nabla u)u \right) = \Delta u - \nabla p \quad \text{in } \Omega, \quad (2.5b)$$

where  $u, p$  are non dimensional velocity and pressure and  $\text{Re}$  is the Reynolds number defined as

$$\text{Re} = \frac{\rho UL}{\mu}, \quad (2.6)$$

while

$$\bar{\sigma} = \frac{\omega L}{U} \quad (2.7)$$

is the Womersley number. We consider the typical length, velocity and frequency for a micro-swimmer as  $L = 10^{-5} \text{ m}$ ,  $U = 10^{-5} \text{ m/s}$  and  $\omega = 10^2 \text{ Hz}$ , we obtain  $\text{Re} = 10^{-4}$  and  $\bar{\sigma}\text{Re} = 10^{-2}$ . Therefore the system of equations governing the flow is the Stokes system,

$$\nabla \cdot u = 0 \quad \text{in } \Omega \quad (2.8a)$$

$$\Delta u - \nabla p = 0 \quad \text{in } \Omega. \quad (2.8b)$$

We identify the boundary of the fluid domain with the boundary of the swimmer  $\partial\Omega = \Gamma$ , we write Dirichlet's boundary conditions for (2.8) as

$$u = u_s, \quad (2.9)$$

where  $u_s$  denotes the velocity of the swimmer on the boundary defined in (2.2). The system (2.8) with Dirichlet boundary conditions is well posed and admits a unique solution  $(u, p) \in ((H^1(\Omega))^3, L_0^2(\Omega))$ . Following [50] we rewrite (2.8) using the fundamental solution to get the representation formula for the Stokes system

$$u_i(x) - \int_{\Gamma} W_{ijk}(x, y) n_k(y) u_j(y) d\gamma_y = \int_{\Gamma} G_{ij}(x, y) f_j(y) d\gamma_y \quad \forall x \in \mathbb{R}^d \setminus \Gamma, \quad (2.10)$$

where  $G, W$  are the first two fundamental solution for the Stokes system. We take the trace of (2.10) to obtain the Boundary Integral Equation (BIE) of the Stokes system as

$$\alpha(x) u_i(x) - \int_{\Gamma}^{PV} T_{ijk}(x, y) n_k(y) u_j(y) d\gamma_y = \int_{\Gamma} G_{ij}(x, y) f_j(y) d\gamma_y \quad \forall x \in \Gamma, \quad (2.11)$$

where the integral on the left is computed in the Cauchy principal value sense, and  $\alpha$  represents its Cauchy principal value. We define the single and double layer operator  $V, H$  as

$$H : \left(H^{\frac{1}{2}}(\Gamma)\right)^3 \rightarrow \left(H^{\frac{1}{2}}(\Gamma)\right)^3, \quad (2.12a)$$

$$V : \left(H^{-\frac{1}{2}}(\Gamma)\right)^3 \rightarrow \left(H^{\frac{1}{2}}(\Gamma)\right)^3, \quad (2.12b)$$

where  $\left(H^{\frac{1}{2}}(\Gamma)\right)^3$  represents the space of the traces of the functional  $(H^1(\Omega))^3$  and  $\left(H^{-\frac{1}{2}}(\Gamma)\right)^3$  is its dual space. We rewrite (2.11) using the operators (2.12a) and (2.12b) as

$$[\alpha I - H]u = -Ku = -Vf. \quad (2.13)$$

We remark that we use the complete traction  $f$  as given by (2.4). For this reason, differently from standard Finite Element simulations of (2.5), we don't need to solve for the pressure  $p$  and the problem is not a saddle point problem. The usage of the fundamental solutions  $G, T$  guarantees that the velocity in the fluid domain is automatically divergence free, see [61] for greater details. For these reasons operator  $V$  is coercive, similarly to what happens to the classic Stokes system (2.5) if we consider a functional space which is divergence free for the velocity field.

### 2.3. Numerical resolution of the swimming problem

To solve the swimming problem means to recover the translation  $q(t)$  and rotation  $R(t)$  given the history of shape changes. For this purpose we use the numerical methodology presented and validated in [19] to discretize (2.11) into a Boundary Element Method (BEM)<sup>1</sup>. We discretize both the geometry of the swimmer and the unknowns  $f, u$  using Lagrangian finite element and we apply a collocation method to derive the BEM linear system

$$\hat{K}\hat{u} = \hat{V}\hat{f}, \quad (2.14)$$

where  $\hat{\cdot}$  represents the finite dimensional counterpart of the operators and functionals represented in (2.13). Following [61] we express the tractions as a function of the velocity introducing the discretized Dirichlet to Neumann map

$$\hat{T} = [\hat{V}^{-1}\hat{K}], \quad (2.15)$$

so that  $\hat{f} = \hat{T}\hat{u}$ . We will drop the  $\hat{\cdot}$  in the rest of the paper for the sake of simplicity. Following [19] we rewrite (2.2) using a suitable set of basis functions for the rigid velocity  $P(X, t)$  and introducing the shape velocity in the swimmer body frame  $v(X, t) = R(t)\dot{s}(X, t)$ , as

$$\begin{aligned} u_s &= \dot{q}(t) + \omega(t) \wedge R(t)s(X, t) + v(X, t) \\ &= \sum_{i=1}^{N_r} p_i(X, t)\dot{p}_i(t) + v(X, t) \\ &= P(X, t)\dot{p}(t) + v(X, t). \end{aligned} \quad (2.16)$$

We assume  $\dot{s}(X, t)$  to be known, so at time  $t$ , given  $R(t)$ , we compute  $v(X, t)$  and we write

$$f_{\text{shape}} = Tv, \quad (2.17a)$$

together with

$$F_{\text{rigid}} = TP. \quad (2.17b)$$

We obtain the rigid velocity coefficients  $\dot{p}(t)$  using (2.3) as

$$P^T M F_{\text{rigid}} \dot{p} + P^T M f_{\text{shape}} = 0, \quad (2.18)$$

where  $M$  is the mass matrix taking care of the surface integrals of (2.3). We then solve (2.18) to get

$$\dot{p} = -[P^T M F_{\text{rigid}}]^{-1} P^T M f_{\text{shape}}. \quad (2.19)$$

We remark that  $[P^T M F_{\text{rigid}}]$  is a symmetric negative definite matrix called grand resistive matrix, so its inversion in (2.19) is legit. We call this resolution strategy, where we compute the tractions associated to any possible velocity and then combine them, the *split approach*.

Another possible resolution strategy is the so-called *monolithic approach* [19, 48], which is based on a single resolution to obtain the overall traction  $f$  together with the rigid velocity coefficients. We write the discretized BIE together with the constraint (2.18) as

$$\begin{bmatrix} [V] & -[K]P \\ P^T[M] & 0 \end{bmatrix} \begin{bmatrix} f \\ \dot{p} \end{bmatrix} = \begin{bmatrix} [K]v \\ 0 \end{bmatrix}, \quad (2.20)$$

the solution of (2.20) is equivalent to the split approach. The differences lie in the fact that the monolithic approach only needs one resolution while the split needs seven. Moreover even the spectral properties of the

<sup>1</sup>In particular we use the open source software **BEMStokes** which is freely available under LGPL license v.2.1 on github [20]

matrices change if we impose the constraint during the BEM resolution. We now apply ROM to both full order approaches to highlight pros and cons of the two resolution strategies.

Standard numerical integration of  $\dot{p}(t)$  provides the rigid displacements and rotations. The **BEMStokes** library efficiently combines several algorithms and ideas in a flexible, modular, and extendible way. It exploits distributed memory parallelism (MPI) using an automatic splitting of the workload at algebraic level using on the graph partitioning tool METIS [33], with the high performance computing libraries Trilinos [29], and **deal.II** [4], used to tackle distributed linear algebra. A similar combination has been successfully applied to achieve high computational efficiency in fluid dynamics, as demonstrated in ASPECT [36] and  $\pi$ -BEM [21].

### 3. APPLICATION: MICRO-SWIMMERS TEST CASES

We consider two models inspired directly from microswimmers found in nature. At the micron scale, there exists a wide variety of organisms [5, 24, 28], which swim using the motion of flagella. The resulting flow can be well modelled using Stokes inertia-free equations [31, 40, 51]. The mathematical model swimmers presented in this section are inspired one from a bacterium and one from the Eukaryotic swimmer *Chlamydomonas Reinhardtii*. The key difference lies in the motion of the flagellum: while bacteria, sketched in Figure 2 can only rotate the root of the flagellum, which is otherwise a passive structure, see [58, 60], Eukaryotic swimmers can exploit a more complex flagellar architecture, called axoneme, to control the shape of the flagellum [49]. We sketch these different configuration in Figure 3.

In particular, for the bacterium presented in Section 3.1 we consider two geometrical parameters, the head radius and the number of flagella windings. The Eukaryotic swimmer of Section 3.2 undergoes a complex flagella movement in a parametrized time trajectory, *i.e.*, a time-dependent boundary control is considered.

#### 3.1. Bacterium-like microswimmer

Following [15, 47, 48, 59] we prescribe a robotic micro-swimmer to be composed by a rigid spherical head of radius  $R$  and a helical tail of width  $b$  which is rotating with respect to the head with constant velocity  $\omega$ . The centerline of the tail is given by

$$r = (x, y, z) = (x, bE(x) \cos(kx - \omega t), bE(x) \sin(kx - \omega t)), \quad (3.1)$$

with

$$E(x) = 1 - e^{-(k_E x)^2}. \quad (3.2)$$

Figure 2 depicts the micro-swimmer under consideration. Following [15] we assume the pitch of the helix  $\lambda = 2\pi/k$ ,  $k_E = k$ , the flagellum has a total number of turns of  $N_\lambda$  and the width of the flagellum is  $b = \lambda/(2\pi)$ . The flagellum thickness is given by

$$d = 0.02 \frac{\pi}{4} R. \quad (3.3)$$

We discretize both the geometry and the unknown functional of (2.14) using linear Lagrangian finite elements for an overall number of  $N_\delta = 2430$  degrees of freedom. We remark that we are considering organisms swimming in free space, in this case a rotational symmetry exists during the stroke (the rotation of the tail w.r.t the head), therefore we simulate the complete stroke of the bacterium-like model solving a single time instant and then we rotate the corresponding instantaneous results to get the results of a complete stroke.

We consider a two-dimensional parameter domain, with varying number of helix turns  $N_\lambda \in [0.4, 4.0]$  and head radius  $R_{\text{head}} \in [0.4, 4.0]$ , so the parameter domain is  $\mathcal{P} = [0.4, 4.0]^2$ . Following [15] we use (3.3) to let the thickness of the flagellum increase proportionally together with the head radius. We aim to use ROM to find the optimal parameter set  $\mu \in \mathcal{P}$  maximizing a chosen performance measure. Following [47, 52] we consider an energetic efficiency, called Lighthill efficiency. Following [15] we compute the axial velocity  $U_{\text{axial}}$  as the projection of the velocity along the angular velocity of the tail, we write

$$U_{\text{axial}} = \dot{q} \cdot \frac{\omega - \Omega}{|\omega - \Omega|}, \quad (3.4)$$





FIGURE 2. Sketch representing a simple robotic-like bacterium composed by a spherical head of radius  $R$  and a helical tail of pitch  $\lambda$  and amplitude  $b$ .

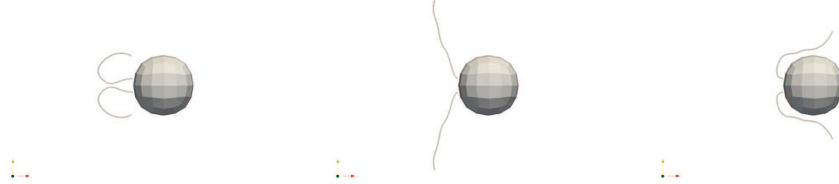


FIGURE 3. Sketch of an Eukaryotic swimmer representing *Chlamydomonas Reinhardtii*. We consider 1200 frames to represent the complete stroke: on the left frame 0, on the center frame 400 and on the right frame 800.

where  $\dot{q}, \Omega$  are the rigid velocities of the swimmer and  $\omega$  is the relative angular velocity between head and tail. Using (3.4) we write the Lighthill efficiency as

$$\eta_{\text{Lighthill}} = \frac{\text{effective power}}{\text{input power}} = \frac{D_{\text{head}} U_{\text{axial}}}{T_{\text{motor}} \omega} = \frac{K_{\text{head}} U_{\text{axial}}^2}{T_{\text{motor}} \omega}, \quad (3.5)$$

where  $K_{\text{head}}$  is the drag coefficient for the head considered alone in free space. We highlight that (3.5) is the ratio between the power expanded to move the payload (the head) and the overall power expanded by the swimmer.

### 3.2. Eukaryotic-like microswimmer

To prove the versatility of our proposed methodology we apply reduced order modeling to a different kind of swimmer, namely we consider an Eukaryotic swimmer mimicking a *Chlamydomonas Reinhardtii* specimen. This organism has been extensively studied in recent years [13, 14, 17, 24] and it is composed by two flagella and a spherical cellular body. The flagella are activated in different ways thanks to the axoneme structure [39] and we focus on the symmetric beating of the two cilia producing a “breast stroke” like movement in the swimmer. We have interpolated the experimental observation of [24] using a standard cubic spline interpolation to obtain a continuous synthetic stroke. For further details on the numerical procedure the reader is referred to [18]. We parametrize the flagellum centerlines as two NURBS beating symmetrically with respect to the body centerline. We represent the periodic stroke using 1200 equi-distanced frames  $\phi$ , we plot the numerical mesh at three different times during the stroke in Figure 3. We consider a spherical cell body having diameter of  $5 \mu\text{m}$  with two symmetrical beating flagella with length  $8 \mu\text{m}$  and thickness  $0.1 \mu\text{m}$ , we use standard linear Lagrangian finite elements for an overall total of  $N_\delta = 1902$  degrees of freedom. This problem is a time-dependent boundary control problem and we remark that since the equations of motion (2.11) do not have any time dependence we can consider the frame number as a geometric parameter  $\mu$ , indexing the current shape. We aim to reconstruct the complete history of rigid body velocities, and consequently the fluid flow around the swimmer, depending on the shape changes, in an efficient and reliable way, exploiting a ROM built from the knowledge of a set of frames. This is also a typical problem from an experimental point of view since the usual frame rates only allows for the reconstruction a discrete subset of shape changes during the stroke.



#### 4. REDUCED ORDER MODELS FOR MICROSWIMMERS

A reduced order model (ROM) is a low-dimensional surrogate model, which approximates the original high-dimensional model over a parameter range of interest. In the following, parameter-dependent quantities are indicated with the parameter vector  $\boldsymbol{\mu}$ . The ROMs are created by means of proper orthogonal decomposition (POD) and reduced basis (RB) methods, see [55] for a general overview and [38] for an overview with a focus on fluid dynamics. The computational speed-up is achieved through an offline-online decomposition, *i.e.*, a compute-intensive offline phase generates a ROM, while a fast online phase is used for many-query ROM evaluations for optimization purposes in a repetitive computational environment. A pre-requisite for the offline-online decomposition is an affine parameter dependency. Since the parameter-dependencies considered here are not affine, an affine parameter dependency is approximated by means of *empirical interpolation method* (EIM) [6, 8]. A projection space is determined to project the high-dimensional equations onto the ROM. During the online phase a low-order approximate solution is computed, independent of the high-order discretization size. Two algorithms are used to compute the projection space: the POD and a residual-based greedy sampling. Both approaches are established in the area of ROM for fluid-structure interactions, see [9, 37].

##### 4.1. Offline-Online decomposition

Conceptually, reduced order modeling uses an offline-online decomposition, in which a compute-intensive offline phase determines an accurate reduced order model (ROM), while the online phase performs fast evaluations. This is beneficial in a many-query or real-time context, since the ROM can be evaluated at low computational cost and accurately recovers the high-dimensional solution.

After applying the *empirical interpolation method* in the geometry parameter, the parameter dependency is cast in an affine form. Therefore, there exists an affine expansion of the matrices  $V(\boldsymbol{\mu})$  and  $K(\boldsymbol{\mu})$  in the parameter  $\boldsymbol{\mu}$ . To achieve fast reduced order solves, the offline-online decomposition computes the parameter-independent projections offline, which are stored as small-sized matrices of the order  $N \times N$ , with  $N$  depending on the velocity and traction projection spaces. When solving for a new parameter online, the affine form (4.1) is evaluated and the reduced order solution (4.4) computed.

In contrast to the typical ROM approach, the dependence on the high-order discretization size is not fully removed in the online phase. The mesh points are generated for the full order problem at each new parameter but only the mesh points relevant for the EIM are taken into account for further computations. This allows to have a maximum of geometric flexibility, needed for these problems, as affinely parametrizing the mesh points would be only possible for special cases of movement. Moreover we do not apply any reduction in the resolution of the momentum balances (2.18) since the computation of the full order method is already efficient, therefore we focus the reduction strategy on the discretization of the BIE (2.14).

##### 4.2. Empirical interpolation method

Since parametric variations in geometry affect the BEM system (2.13) in a nonlinear way, *empirical interpolation method* (EIM) [6, 8] computes an approximate affine parameter dependency. The matrix discrete empirical interpolation [43] computes the decompositions

$$\sum_{i=1}^{Q_V} \Theta_V^i(\boldsymbol{\mu}) V_i \approx V(\boldsymbol{\mu}), \quad \sum_{i=1}^{Q_K} \Theta_K^i(\boldsymbol{\mu}) K_i \approx K(\boldsymbol{\mu}). \quad (4.1)$$

with scalar parameter-dependent coefficient functions  $\Theta_V^i(\boldsymbol{\mu})$  and  $\Theta_K^i(\boldsymbol{\mu})$  and parameter-independent matrices  $V_i$  and  $K_i$ . In particular, each coefficient function corresponds to a single matrix entry of  $V$  or  $K$ , respectively. Since the assembly of only a few (here  $Q_a < 260$ ) matrix entries can be implemented efficiently, an approximation of  $V$  and  $K$  is readily available for each new  $\boldsymbol{\mu}$ . As in the determination of the projection spaces  $U_u, U_f$ , at least 99.99% of the POD energy is used to approximate the system matrices from the collected matrices during snapshot computation. The EIM is performed for each of the two BEM matrices  $V, K$  and the reduced mode

are stored considering the 6 sets of rigid velocities and the shape velocity for an overall total of 14 reduced matrices. The affine expansions on the other hand, are the same for each rigid mode and the shape velocity.

### 4.3. Proper orthogonal decomposition

The POD samples uniformly distributed Stokes solutions  $u, f$  over the parameter domain. The solutions are often called *snapshots* in reduced order modeling. A *singular value decomposition* of the snapshots is computed and the most dominant modes are chosen as the projection space. The most dominant modes corresponding to at least, 99.99% of the POD energy form the projection matrix  $U \in \mathbb{R}^{N_s \times N}$  and implicitly define the low-order space  $V_N = \text{span}(U)$  and set the reduced model size  $N$ . It is possible to use less than 99.99% of the POD energy, which results in smaller ROMs, but less accurate ROM projections. When the POD energy threshold is too low, then the POD projection error will become dominant over other error sources like the EIM. Choosing 99.99% of the POD energy is usually sufficient for at least 5 digits of accuracy in the vast majority of applications.

### 4.4. Greedy sampling

The greedy sampling builds the projection space iteratively from snapshot solutions. The parameter location of the next snapshot is chosen as the maximum of a residual-based error indicator, see [55] for details. In particular we implement a classic residual-based error indicator to choose the snapshots for the force samples selection, while we use a projection-based error indicator for the selection of the velocity samples. The choice of different samples for all the possible snapshots allows for a greater accuracy and faster ROM computation. The decay of the maximum residual over the iterations determines a stopping criterion of the greedy sampling. As in the POD case, a projection matrix  $U \in \mathbb{R}^{N_s \times N}$  is obtained after orthonormalizing the snapshots.

For the monolithic system the residual  $r(\mu)$  is based on equation (2.20) by plugging in the low order solution  $f_N$  and  $\dot{p}_N$  as

$$r(\mu) = \left\| \begin{bmatrix} [V] & -[K]P \\ P^T[M] & 0 \end{bmatrix} \begin{bmatrix} f_N \\ \dot{p}_N \end{bmatrix} - \begin{bmatrix} [K]v \\ 0 \end{bmatrix} \right\|, \quad (4.2)$$

while for the split approach, the total residual  $r(\mu)$  is formed as the sum of the residual of the 6 rigid velocity modes and the shape velocity, which is expressed using equation (2.17) as

$$r(\mu) = \left\| [V]f_N^{\text{shape}} - [K]v \right\| + \sum_{i=1}^6 \left\| [V]F_N^i - [K]P^i \right\|, \quad (4.3)$$

where  $f_N^{\text{shape}}$  and  $F_N^i$  are the low order tractions associated with the shape velocity  $v$  and the  $i$ th rigid mode  $P^i$ .

Both residual computations equations (4.2) and (4.3) can be expanded in the parameter  $\mu$  provided by the affine decomposition from the EIM and the residual computations can be sped-up further by precomputing the parameter-independent parts online, see [30, 55].

### 4.5. Projection for the split approach

In fact, two projection spaces are built, namely  $U_u$  from sampling  $u$  in (2.14) and  $U_f$  from sampling  $f$  in (2.17a). The large-scale system (2.14) is then projected onto the reduced order space:

$$U_f^T [K](\mu) U_u U_u^T u(\mu) = U_f^T [V](\mu) U_f U_f^T f(\mu). \quad (4.4)$$

The low order solutions  $u_N(\mu), f_N(\mu)$  approximates the large-scale solution as  $u(\mu) \approx U_u u_N(\mu), f(\mu) \approx U_f f_N(\mu)$ . In particular, the dimensions of the projection spaces  $U_u$  and  $U_f$  can be chosen independently from each other for every momentum and rigid velocity leading to the creation of 7 different reduced order systems as the one depicted in (4.4). Choosing them independently has two major advantages: in case of the greedy sampling, less snapshot solutions need to be computed, and the evaluation of the ROM becomes faster since

the ROM dimensions are smaller. Once we solve the 7 different ROM systems we can use (2.3) to obtain the six rigid velocities, in particular we use the full order mass matrix  $M(\boldsymbol{\mu})$  together with the rigid modes to obtain

$$P^T(\boldsymbol{\mu})[M(\boldsymbol{\mu})][U_f^1 f_N^1(\boldsymbol{\mu}) | \dots | U_f^6 f_N^6(\boldsymbol{\mu})]p(\boldsymbol{\mu}) = -P^T(\boldsymbol{\mu})[M(\boldsymbol{\mu})]U_f^s f_N^s(\boldsymbol{\mu}) \quad (4.5)$$

where  $P(\boldsymbol{\mu})$ ,  $M(\boldsymbol{\mu})$  represent the full order rigid modes and mass matrix respectively. The assembling time of the full order mass matrix is very fast given its sparsity with respect to the one required by the two BEM operators, so we choose to use the full order operator to have the maximum possible accuracy.

#### 4.6. Projection for the monolithic approach

The full order model presented in Section 2.3 is based on seven different application of the Dirichlet to Neumann map  $T$  to retrieve the stress fields associated to shape and rigid velocities and then combine them to fulfill (2.3) and retrieve the actual rigid velocity coefficients. Alternatively, we use the “so-called” monolithic approach [19, 48] that solves the Stokes system imposing the constraints specified by the momentum balances (2.3) at the same time. We write the monolithic system as

$$\begin{bmatrix} [V](\boldsymbol{\mu}) & -[K](\boldsymbol{\mu})P(\boldsymbol{\mu}) \\ P^T(\boldsymbol{\mu})[M](\boldsymbol{\mu}) & 0 \end{bmatrix} \begin{bmatrix} f(\boldsymbol{\mu}) \\ \dot{p}(\boldsymbol{\mu}) \end{bmatrix} = \begin{bmatrix} [K](\boldsymbol{\mu})v(\boldsymbol{\mu}) \\ 0 \end{bmatrix}, \quad (4.6)$$

where  $[V](\boldsymbol{\mu})$ ,  $[K](\boldsymbol{\mu})$  are the Stokes operators defined in Section 2.3 and  $[M](\boldsymbol{\mu})$  is the Mass matrix that takes care of the surface integration described in (2.3),  $f(\boldsymbol{\mu})$  represents the tractions on the boundary of the swimmer,  $\dot{p}(\boldsymbol{\mu})$  are the rigid velocity coefficients while  $v(\boldsymbol{\mu})$  is the shape velocity. We remark that the system (4.6) consists of  $N_\delta + 6$  equation in  $N_\delta + 6$  unknowns. We repeat the procedure described in Section 4.1 and we introduce the projection spaces  $U_f, U_u$  obtaining

$$\begin{bmatrix} U_f^T[V](\boldsymbol{\mu})U_f & -U_f^T[K](\boldsymbol{\mu})U_u P_N(\boldsymbol{\mu}) \\ P^T(\boldsymbol{\mu})[M](\boldsymbol{\mu})U_f & 0 \end{bmatrix} \begin{bmatrix} \tilde{f}(\boldsymbol{\mu}) \\ \dot{p}(\boldsymbol{\mu}) \end{bmatrix} = \begin{bmatrix} U_f^T[K](\boldsymbol{\mu})U_u v_N(\boldsymbol{\mu}) \\ 0 \end{bmatrix}, \quad (4.7)$$

where  $f_N(\boldsymbol{\mu}) = U_f^T f(\boldsymbol{\mu})$  is the reduced order representation of the tractions,  $P_N(\boldsymbol{\mu}) = U_u^T P(\boldsymbol{\mu})$  is the projection of the rigid velocities and  $v_N(\boldsymbol{\mu}) = U_u^T v(\boldsymbol{\mu})$  is the projection of the shape velocity. The system (4.7) consists of  $N + 6$  equation in  $N + 6$  unknowns, we remark that  $[V_N](\boldsymbol{\mu}) = U_f^T[V](\boldsymbol{\mu})U_f$  and  $[K_N](\boldsymbol{\mu}) = U_f^T[K](\boldsymbol{\mu})U_u$  are computed using the empirical interpolation method described in Section 4.2. The spaces  $U_f$  and  $U_u$  are obtained using a POD of samplings for traction and velocity field, the choice of the samplings has a key-role in determining the stability of the reduced monolithic system. We do not address this issue in detail in the present work, but, judging from our numerical experiments, we believe the best choice to be samplings of all the possible tractions and velocities the system experiences, namely  $U_f$  is obtained from the POD of the shape traction  $f_{\text{shape}}(\boldsymbol{\mu})$  and the rigid tractions  $F_{\text{rigid}}(\boldsymbol{\mu})$  while  $U_u$  comes from the sampling of both  $v(\boldsymbol{\mu})$  and  $P(\boldsymbol{\mu})$ .

#### 4.7. Expected ROM accuracy

Since the modeling and meshing is performed with the software *Blender* v2.79 [7], the input data for numerical simulation and ROM simulations are limited to single precision. We believe the choice of a wide-spread well documented open-source software to be of capital importance in order to guarantee flexibility to the micro-motility solver. It is expected that there is in each step of (i) empirical interpolation, (ii) projection and (iii) solving in ROM space, a slight degradation in the accuracy. We thus expect an accuracy between ROM and FOM of about four digits and an accuracy in an output quantity of about three to four digits. Typically, the input data are double precision and a higher accuracy between ROM and FOM can be expected.

#### 4.8. Optimization using ROM, the coarse-fine approach

The task is to optimize the swimming performance  $\eta_{\text{Lighthill}}$  using a reduced order model built considering the head radius ( $R_{\text{head}}$  and number of windings  $N_{\lambda}$ ) over a two-dimensional parameter domain  $\mathcal{P}$ . To obtain an optimal speed-up, we propose a two-step method. In the first step, a coarse sampling of the parameter domain is used to obtain the approximate location of the maximum swimming efficiency  $\eta_{\text{Lighthill}}$ . In the second step, a focused parameter domain  $\mathcal{P}_{\text{focus}} \subset \mathcal{P}$  is determined, where a fine sampling of the parameter domain is used to obtain a highly accurate ROM solution. This could induce a further speed up with respect to the full order model as the focused parameter domain may require less snapshot basis functions for an accurate ROM. A reference solution is computed for the real maximum of  $\eta_{\text{Lighthill}}$  using the full order BEM.

#### 4.9. Stroke reconstruction

To reconstruct the motion of the swimmer, we need to solve the complete time-dependent stroke. Since the equation of motion are time-independent in the low Reynolds number regime, the time is simply indexing the geometrical variations  $\mu$  for the shape changes. Time thus behaves as any physical parameter for the model reduction purpose, which is in contrast to PDEs involving a time-derivative, where time is treated differently than other parameters. We assume complete knowledge of the geometry of the swimmer at all time instances and the solution of training snapshots at the sample points.

### 5. NUMERICAL RESULTS

We present the numerical results obtained from coupling the boundary element method and reduced order modeling. Firstly, we present the convergence of the ROM to the full order solution with increasing dimension of the reduced model: the main focus is on the POD approximation [30], which is tested on both applications in Section 5.1. In Section 5.2 we compare the POD and Greedy approach on the Eukaryotic-like swimmer, where the POD approximation turns out to be less accurate. Then we present two applications: the shape optimization of the bacterium-like swimmer, see Section 5.3, and the stroke reconstructions for the Eukaryotic-like swimmer, see Section 5.4.

#### 5.1. POD approximations of BEM solution

In Sections 4.5 and 4.6 we introduced the split approach and the monolithic approach for projection-based ROMs for micro-swimmers. We compare the two approaches both for the bacterium-like and Eukaryotic-like swimmer, we consider both convergence to the full order solutions and online timings required to reach a prescribed accuracy. For this analysis we consider the maximum accuracy for the EIM and we let the number of modes gradually increase to see the convergence of the ROMs. The split approach allows for 7 different projection spaces for the different tractions, we report only the analysis for the traction associated to the shape force  $f_{\text{shape}}$  since it is the most demanding one in terms of accuracy and number of modes.

##### 5.1.1. Robotic bacterium

We compare split and monolithic POD approaches on the robotic-like bacterium presented in Section 3.1. Firstly we compare the convergence of the ROMs to the full order solution, in particular we analyze the errors relative to the traction vector  $f$  in the two different models. Figure 4 shows the convergence rate for the split approach on the left and for the monolithic approach on the right, red, green and blue depicts minimum, mean and maximum error computed over 10 randomly selected non training snapshots. We remark that the monolithic approach is built considering all the possible  $N_{\text{rigid}} + 1$  tractions so this approach has an overall number of modes that is  $N_{\text{rigid}} + 1$  times bigger than the split approach that requires  $N_{\text{rigid}} + 1$  expansions instead. The convergence rates are very similar between the two different models, and they both reach a similar maximum accuracy between  $10^{-5}$  and  $10^{-3}$ . If we increase further the number of modes a clear plateau emerges in the convergence and it is mainly due to the errors introduced by the EIM. We tested the convergence of the

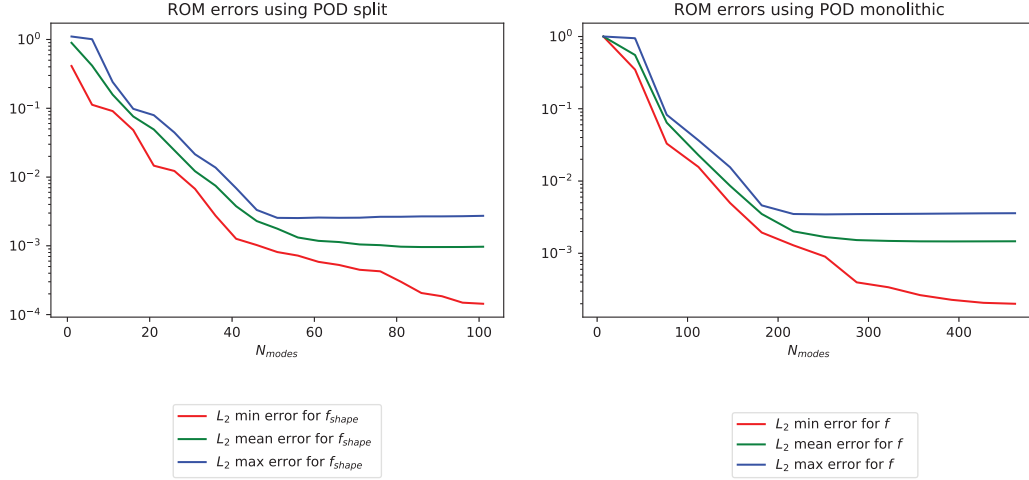


FIGURE 4. ROM convergence analysis for the robotic-like swimmer computed for 10 random non training snapshots using POD. On the left the errors of the shape traction considering the split approach, on the right the final traction errors using the monolithic approach. Red, green and blue represent the minimum, mean and maximum error respectively.

TABLE 1. Summary of the ROM performance indicators for the POD approximations on the bacterium-like test case. Shown are the smallest basis sizes, where a mean approximation error below 5% and 0.5% is attained. The online timing is referred to the approximation needed to have 0.5% accuracy.

	Split mean	Monol mean	Split max	Monol max
Err. less 5%	21	77	26	112
Err. less 0.5%	41	182	46	182
Online timing	1.322	1.446	1.371	1.446

ROM without considering the matrix approximations and we managed to achieve between four and five digits of accuracy without such an evident plateau.

In Table 1 we compare the performances of the two POD models, we report the number of modes required to reach a prescribed accuracy of 5% and 0.5% both for the mean and maximum error. The split approach requires less number of modes than the monolithic one, however we remind that the latter approach requires a single resolution to get the final solution while the split approach solves 7 different linear systems. From this analysis we see that the split approach has a slight advantage with respect to the monolithic one. We believe that this is due to a better memory handling of the split approach, in fact a deeper analysis reveals that the monolithic approach requires more time to compute the BEM entries to be used in the EIM, see Section 4.2. The two approaches require the same entries to be computed but the reduced number of modes of the split approach makes available more memory to the BEM computation and this results in this slight speed up. Apart this minor difference we believe the two approaches to be equivalent.

#### 5.1.2. Eukaryotic swimmer

We compare the convergence to the full order model and the performances of the two POD approaches on the Eukaryotic-like swimmer presented in Section 3.2. Figure 5 compares the convergence of the split approach (on the left) and the monolithic approach (on the right). Red, green and red represent minimum, mean and

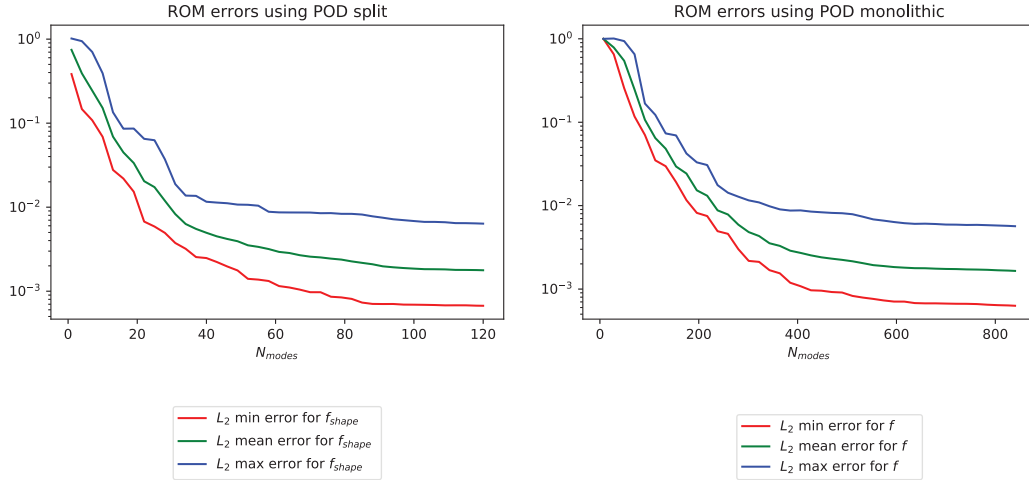


FIGURE 5. ROM convergence analysis for the Eukaryotic-like swimmer computed for 10 random non training snapshots using POD. On the left the errors of the shape traction considering the split approach, on the right the final traction errors using the monolithic approach. Red, green and blue represent the minimum, mean and maximum error respectively.

TABLE 2. Summary of the ROM performance indicators for the POD approximations for the Eukaryotic-like swimmers. Shown are the smallest basis sizes, where a mean approximation error below 5% and 0.7% is attained. The online timing is referred to the approximation needed to have 0.7% accuracy.

	Split mean	Monol mean	Split max	Monol max
Err. less 5%	16	133	28	175
Err. less 0.7%	34	280	100	553
Online timing	0.6637	0.6631	0.6792	1.107

maximum errors computed on 10 randomly chosen non training snapshots. We see that the two ROMs present the same convergence to the full order solution, the error in both cases converges between  $10^{-3}$  and  $10^{-2}$ . We don't see any clear plateau in this test-case and this is due to the increase complexity in the geometry for the Eukaryotic like movement we consider. For this reason the EIM plateau of Section 5.1.1 is not reached.

Table 2 compares the performances of the two approaches considering the number of modes required to reach 5% and 0.7% accuracy. As expected, the split approach requires less modes to reach the target accuracy but we note again that the major part of the online time is required to reconstruct the matrices using EIM, and specifically it is needed to compute the BEM entries. If the number of modes remains low the timings are the same while for the last test case when we require 553 modes we notice a difference in the timings. We believe that this phenomenon confirms our speculation about the worse memory handling of the code when the number of modes increases leaving less memory available to the BEM computations. Apart from this non-linear minor effect we see that the two methods are almost equivalent both in terms of convergence and performances.

## 5.2. Greedy and POD: comparison of approximation accuracy

In Section 5.1 we saw that the two POD approaches are mostly equivalent on both the test-cases presented in Section 3. However we noticed that the geometry shape change of the Eukaryotic-like swimmer, represented in

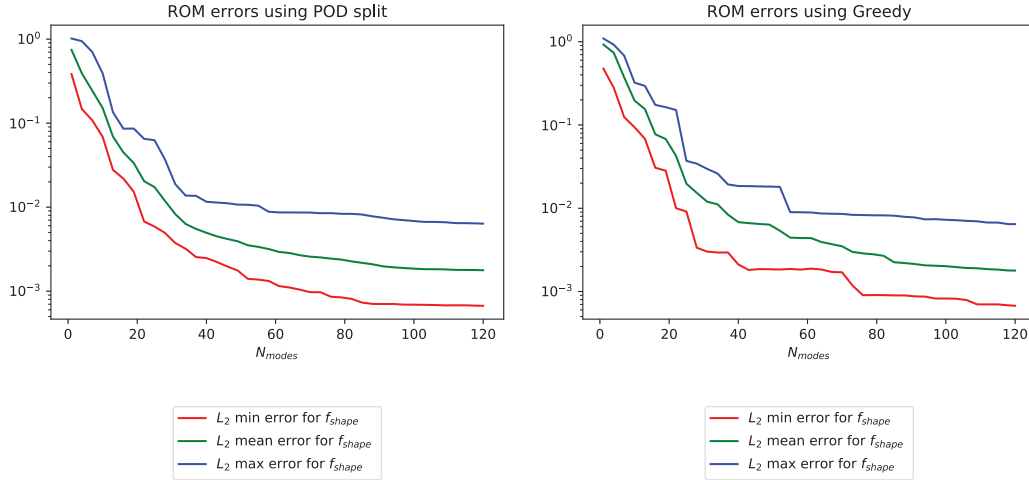


FIGURE 6. Comparison of ROM convergence for the Eukaryotic-like swimmer computed for 10 random non training snapshots using POD or Greedy approach. On the left the POD errors of the shape traction, on the right the shape traction errors using Greedy. Red, green and blue represent the minimum, mean and maximum error respectively.

Figure 3, is more complicated than the rotation of the bacterium-like one. This induces a worse convergence to the full order solution. For this reason we compare the split POD approach to the Greedy approach introduced in Section 4.4 on this test-case. Figure 6 compares the convergence of the POD approach (on the left) and the Greedy approach (on the right), red, green and blue represent minimum, mean and maximum errors computed on 10 randomly chosen non training snapshots. We see that the two approaches have a very similar convergence to the exact solution, we notice that the POD approach is smoother but they both reach the same final accuracy. Table 3 compares the performances of the two approaches and we see that the POD and the Greedy are equivalent even from this point of view. To better understand the Greedy selection procedure we represent in Figure 7 the first indices selected to approximate the shape velocity  $v$  (on the left) and the corresponding traction force  $f_{\text{shape}}$  (on the right). The indices cluster around two different points: the first point is located at  $\phi \sim 200$  which corresponds to the transition between backward and forward motion, the second one is at  $\phi \sim 900$  which corresponds to the two flagella moving very close to the body. Since inertia is negligible the transition between backward and forward motion is immediate as soon as the flagellar beat allows it and the Greedy procedure selects frame near the transition. When the flagella are close to the body there is a lot of interaction between different body parts and the Greedy procedure selects frames in this region to have a better approximation. To better understand this scenario we compare the magnitude of the traction when flagella are very near the body in Figure 8. Even if the differences are moderate the solution with one quarter of the possible Greedy modes depicted on the left is not able to recover the traction pattern of the reference solution on the right. Consequently, the Greedy algorithm tries to increment the accuracy by selecting more modes for this situation, the plot at the centre shows the most accurate possible solution which has a slight increase in the traction representation.

From our analysis on the Eukaryotic-like swimmer we conclude that Greedy and POD have the same convergence and timings. The results of Sections 5.1.1 and 5.1.2 show the same behavior for all the POD on both application so we expect Greedy and POD to behave similarly with the monolithic approach and the bacterium-like application. This grants flexibility for further use cases as either POD/Greedy and split/monolithic approaches can achieve speed-ups of several magnitudes.



TABLE 3. Summary of the ROM performance indicators for the POD and Greedy approximation on the Eukaryotic-like test-case. Shown are the smallest basis sizes, where a mean approximation error below 5% and 0.7% is attained. The online timing is referred to the approximation needed to have 0.7% accuracy.

	POD mean	Greedy mean	POD max	Greedy max
Err. less 5%	16	22	28	25
Err. less 0.7%	34	40	100	109
Online timing	0.6637	0.6547	0.6792	0.6814

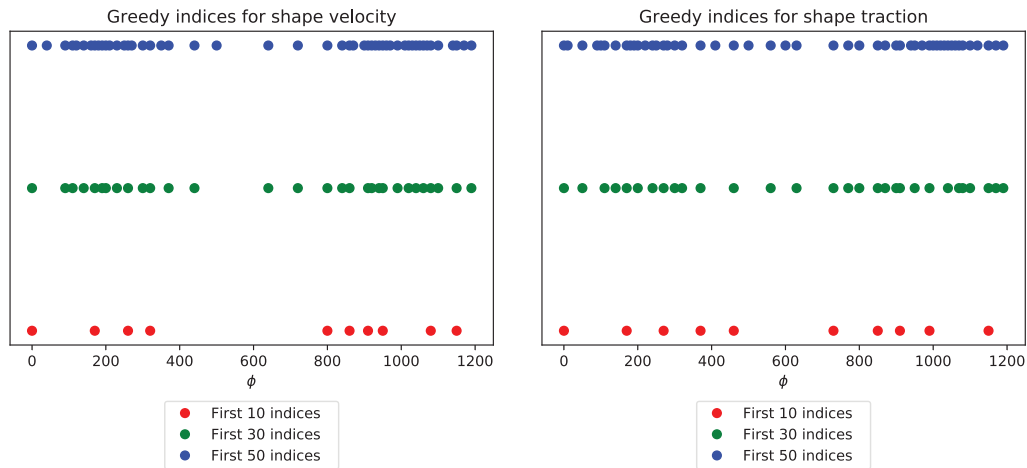


FIGURE 7. Greedy snapshot selection for shape velocity (on the left) and shape traction (on the right) approximations. Red, Green and blue shows the first 10, 30, 50 selected modes.

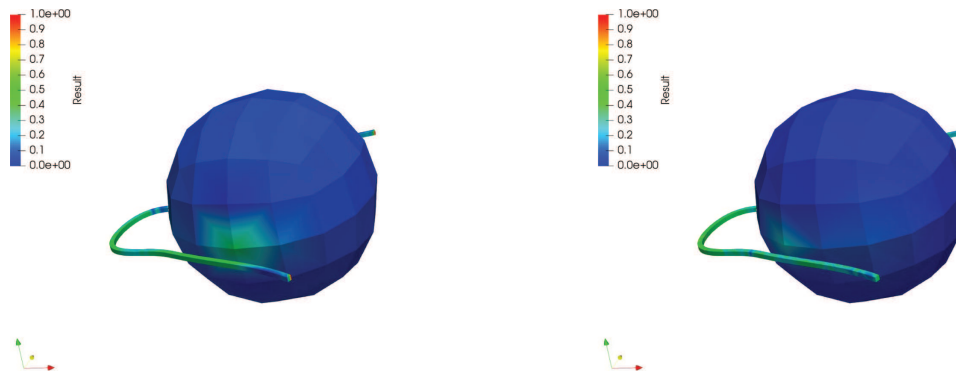


FIGURE 8. Comparison of the traction error magnitude for the Eukaryotic swimmer when the flagella are very near the cell body. On the left the reconstructed solution using 25% of the available modes, on the right the most accurate reduced order solution. The scale is the same on the two plots and it is set so that 1 represent the maximum error of the two approximations.

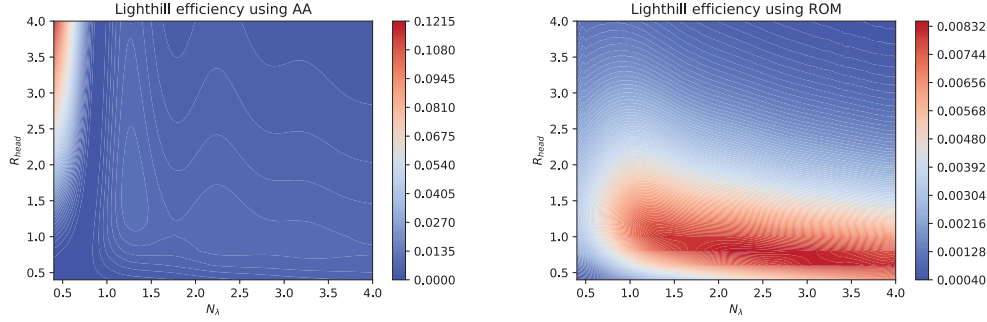


FIGURE 9. Lighthill efficiency (3.5) analysis  $N_\lambda = 0.4 - 4.0$ ,  $R_{\text{head}} = 0.4 - 4.0$ . AA on the left, ROM on the right.

This procedure is not tied to the BEM discretization technique. In fact, this ROM approach has proven to work with finite element methods or finite volume methods in many application areas, see [30, 55] and the references therein.

### 5.3. Two step shape optimization of robotic bacterium

In this Section we study the actual shape optimization of the robotic micro-swimmer introduced in Section 3.1. As performance measure we consider the energetic efficiency  $\eta_{\text{Lighthill}}$  introduced in (3.5). We find the optimal value of the parameter  $\mu = (N_\lambda, R_{\text{head}})$  in the parameter domain  $\mathcal{P} = [0.4, 4.0]^2$ . We only use one of the possible ROMs we presented since they proved to have very similar accuracy and performances. A common way to study and optimize the performance of the considered swimmer is to study the hydrodynamics of the separate component (head and flagellum) to infer the swimming behavior of the complete swimmer [51, 52]. For a complete analysis of this “additive approach” (AA) the reader is referred to [19].

In Figure 9 we compare  $\eta_{\text{Lighthill}}$  analysis using AA (on the left) and ROM (on the right) letting the radius of the head  $R_{\text{head}}$  to vary with a step of 0.2 and the number of turns for the tail with a step 0.02. AA shows the optimum ( $\eta_{\text{Lighthill}} = 0.12138$ ) for  $N_\lambda = 0.4, R = 4.0$ . Considering the analysis of [19] we believe this optimal value to be erroneous since a big head radius creates a considerable wake effect, which is completely neglected by AA, worsening the performances of the real complete swimmer. ROM shows instead an optimum ( $\eta_{\text{Lighthill}} = 0.008406$ ) for  $N_\lambda = 2.36, R = 0.8$  and we see that also AA shows a local maximum close to  $N_\lambda = 2.4$  and  $R = 0.7$ . We compute the real maximum for  $\eta_{\text{Lighthill}}$  using the full order BEM on the same parameter space and we obtain the optimum at  $N_\lambda = 2.38, R = 0.8$  with a relative error for the efficiency of 1.41% for the ROM.

To increase the accuracy of our predictions we apply a second optimization step using a ROM build considering a parameter grid centered on the maximum value region  $\mathcal{P}_{\text{focus}} \subset \mathcal{P}$ , namely we consider a new set of training  $\mu = (N_\lambda, R_{\text{head}})$  in the parameter domain  $\mathcal{P}_{\text{focus}} = [0.6, 1.1] \times [2.0, 2.6]$ . Figure 10 shows the results on the fine grid both for AA (on the left) and ROM (on the right). AA predicts an optimum ( $\eta_{\text{Lighthill}} = 0.00684121$ ) for  $N_\lambda = 2.4333, R_{\text{head}} = 0.6833$ , ROM shows the optimum ( $\eta_{\text{Lighthill}} = 0.0085464$ ) for  $N_\lambda = 2.42, R_{\text{head}} = 0.7667$ , while the full order BEM displays the optimum ( $\eta_{\text{Lighthill}} = 0.00854549$ ) for  $N_\lambda = 2.4067, R_{\text{head}} = 0.7667$ . The simplified AA is not able to recover a good result, especially for what concerns the head radius, while a proper ROM can be effectively used to compute a configuration which is very close to the optimal one for a robotic micro-swimmer. Table 4 sums up the various predictions for the maximum  $\eta_{\text{Lighthill}}$  for all the different scenarios and we clearly see that the ROM is able to recover very good solutions for the optimal shape.

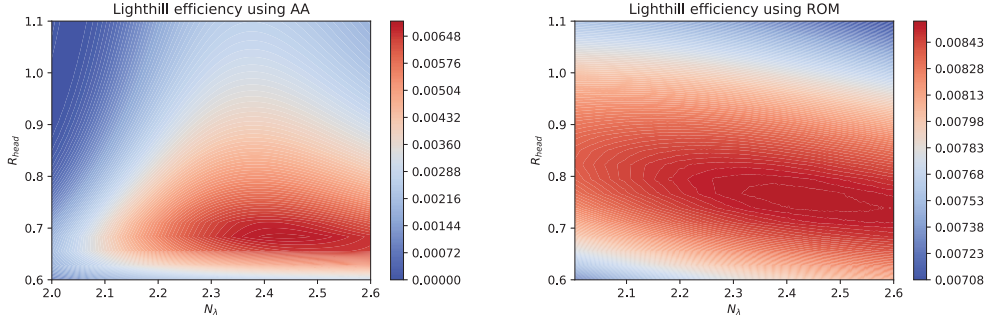


FIGURE 10. Lighthill efficiency (3.5) analysis  $N_\lambda = 2.0 - 2.6$ ,  $R_{\text{head}} = 0.6 - 1.1$ . AA on the left, ROM on the right.

TABLE 4.  $\eta_{\text{Lighthill}}$  predictions and errors using both additive approach and ROM.

Method	$R_{\text{head}}$	$N_{\text{turns}}$	$\eta_{\text{Lighthill}}$	Error
BEM coarse	0.8	2.38	0.008526	
AA coarse	4.0	0.4	0.12138	1323%
ROM coarse	0.8	2.36	0.008406	1.41%
BEM fine	0.7667	2.4067	0.00854549	
AA fine	0.6833	2.4333	0.00684121	19.94%
ROM fine	0.7667	2.42	0.0085464	0.0104%

#### 5.4. Stroke reconstruction of a micro-swimmer

Another possible application of the ROM procedures described in this work is the reconstruction of the rigid body velocities of the swimmer during the complete stroke starting from some precomputed snapshots. We apply this procedure to the Eukaryotic-like swimmer presented in Section 3.2. We have already shown that all the presented ROMs leads to very similar results, so in this Section we only use one of them, in particular we use the POD split approach.

On the left of Figure 11 we use ROM to reconstruct the rigid velocities during the complete stroke starting from the knowledge of the system at the training snapshots. The artificial stroke that we want to reconstruct consists of 1200 different frames, we consider an increasing number of equi-spaced training snapshots ( $N_{\text{training}} = 6, 12, 40, 120$ ) and we see that the solution converges to the full order solution increasing the complexity of the ROM. ROM has proved to be an effective tool to reconstruct the complete stroke starting from a limited number of training snapshots. The right plot of Figure 11 clearly shows the convergence of the stroke to the proper one as we increase the number of training snapshots.

## 6. CONCLUSIONS

This work shows that ROM enhanced BEM simulations of microswimmers can achieve accurate results while significantly reducing the runtime. The many-query setting, which is typical to ROM, also holds when resolving models depending only on time as a single parameter. This is because inertia, *i.e.*, a time-derivative, is not present and the fluid “reaction” is quasi instantaneous. This widens the applicability of ROM methods to microswimmers significantly, as each use-case can be treated separately in a ROM or several use-cases can be further parametrized so that ROM can be applied to a whole family of microswimmers. Another feature of the presented approach is the geometric flexibility, which applies EIM to online-generated mesh data. Using the

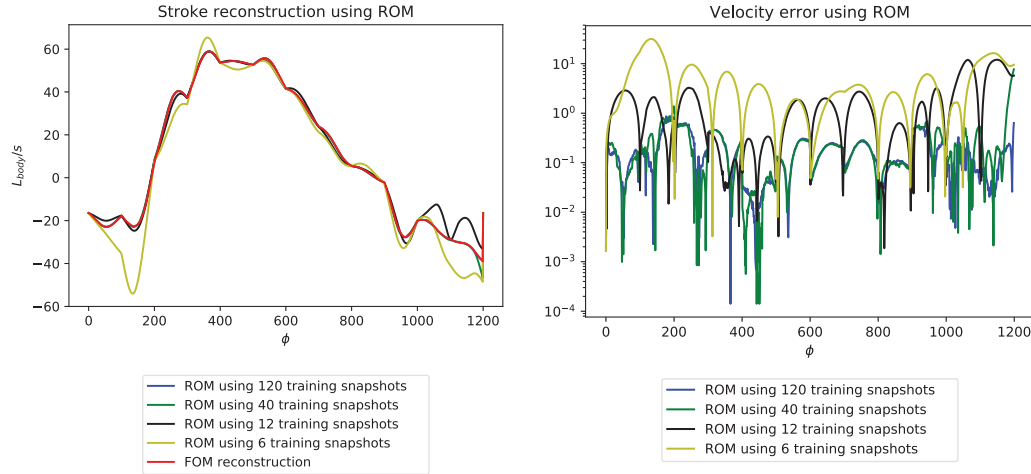


FIGURE 11. Stroke reconstruction of an Eukaryotic-like swimmer. On the left we plot the velocity along the longitudinal axis. On the right we report the error of the ROM approximations with respect to the full order model. Yellow, black, green and blue lines define ROM approximations built considering an increasing number of training snapshots (6, 12, 40, 120), the red line represents the reference solution obtained with the full order BEM.

high-order mesh generation in the online phase allows significantly more geometric flexibility than the restriction to affine geometry transformations, and it allows to capture also large and non-linear deformations.

Both tested formalisms of the BEM systems (*i.e.*, split and monolithic approach) and standard reduced basis ROM algorithms (*i.e.*, POD and greedy) are capable to mitigate the computational costs with an acceptable approximation accuracy. Each approach reaches engineering standards of on average three digits of accuracy. The methodology is applied on two different test-cases, and the maximum reduced order solution error is below 1% in both models, the robotic and eukaryotic.

To be able to further increase the accuracy of the presented methodology a meshing software which provides more than single digit accuracy is necessary. With the presented approach of EIM, projection, solving the reduced order system and approximating the output quantity, on average 3 to 4 digits of accuracy on the output quantity can be achieved.

*Acknowledgements.* This work has been supported by the FP7 ERC Advanced Grant 2013 project 340685 “MicroMotility” and H2020 ERC Consolidator Grant 2015 AROMA-CFD project 681447 “Advanced Reduced Order Methods with Applications in Computational Fluid Dynamics”. Nicola Giuliani thanks Dr. Giovanni Stabile, for useful discussions. The support of INDAM GNCS is also acknowledged.

## REFERENCES

- [1] F. Alouges and M. Aussal, The sparse cardinal sine decomposition and its application for fast numerical convolution. *Numer. Algorithms* **70** (2015) 427–448.
- [2] F. Alouges, A. DeSimone, L. Giraldo and M. Zoppello, Can magnetic multilayers propel artificial microswimmers mimicking sperm cells? *Soft Rob.* **2** (2015) 117–128.
- [3] F. Alouges, M. Aussal, A. Lefebvre-Lepot, F. Pigeonneau and A. Sellier, Application of the sparse cardinal sine decomposition to 3d stokes flows. *Int. J. Comput. Methods Exp. Meas.* **5** (2017) 387–394.

- [4] G. Alzetta, D. Arndt, W. Bangerth, V. Boddu, B. Brands, D. Davydov, R. Gassmoeller, T. Heister, L. Heltai, K. Kormann, M. Kronbichler, M. Maier, J.-P. Pelteret, B. Turcksin and D. Wells, The `deal.II` Library, Version 9.0. *J. Numer. Math.* **26** (2018) 173–183.
- [5] J. Archibald, Handbook of the Protists. Springer, Cham (2017).
- [6] M. Barrault, Y. Maday, N.C. Nguyen and A.T. Patera, An “empirical interpolation” method: application to efficient reduced-basis discretization of partial differential equations. *C. R. Math.* **339** (2004) 667–672.
- [7] Blender Online Community, Blender – A 3D Modelling and Rendering Package. Blender Foundation, Blender Institute, Amsterdam (2019).
- [8] S. Chaturantabut and D.C. Sorensen, Nonlinear model reduction via discrete empirical interpolation. *SIAM J. Sci. Comput.* **32** (2010) 2737–2764.
- [9] C.M. Colciago and S. Deparis, Reduced numerical approximation of reduced fluid-structure interaction problems with applications in hemodynamics. *Frontiers Appl. Math. Stat.* **4** (2018) 18.
- [10] B. Dai, J. Wang, Z. Xiong, X. Zhan, W. Dai, C.-C. Li, S.-P. Feng and J. Tang, Programmable artificial phototactic microswimmer. *Nat. Nanotechnol.* **11** (2016) 1087–1092.
- [11] G. Dal Maso, A. DeSimone and M. Morandotti, An existence and uniqueness result for the motion of self-propelled microswimmers. *SIAM J. Math. Anal.* **43** (2011) 1345–1368.
- [12] G. Dal Maso, A. DeSimone and M. Morandotti, One-dimensional swimmers in viscous fluids: dynamics, controllability, and existence of optimal controls. *ESAIM: Control Optim. Calc. Var.* **21** (2015) 190–216.
- [13] K. Drescher, K.C. Leptos and R.E. Goldstein, How to track protists in three dimensions. *Rev. Sci. Instrum.* **80** (2009) 014301.
- [14] 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** (2010) 168101.
- [15] T. Fujita and T. Kawai, Optimum shape of a flagellated microorganism. *JSME Int. J. Ser. C* **44** (2001) 952–957.
- [16] I. Fumagalli, N. Parolini and M. Verani, Shape optimization for Stokes flows: a finite element convergence analysis. *ESAIM: Math. Modell. Numer. Anal.* **49** (2015) 921–951.
- [17] V.F. Geyer, F. Jülicher, J. Howard and B.M. Friedrich, Cell-body rocking is a dominant mechanism for flagellar synchronization in a swimming alga. *Proc. Nat. Acad. Sci. USA* **110** (2013) 18058–18063.
- [18] N. Giuliani, *Modelling fluid structure interaction problems using boundary element method*. Ph.D. thesis, SISSA, Scuola Internazionale Superiore di Studi Avanzati (2017).
- [19] N. Giuliani, L. Heltai and A. DeSimone, Predicting and optimizing microswimmer performance from the hydrodynamics of its components: the relevance of interactions. *Soft Rob.* **5** (2018) 410–424.
- [20] N. Giuliani, L. Heltai and A. DeSimone, BEMStokes: a boundary element method solver for micro-swimmers. <https://github.com/mathLab/BEMStokes> (2020).
- [21] N. Giuliani, A. Mola and L. Heltai,  $\pi$  – BEM: a flexible parallel implementation for adaptive, geometry aware, and high order boundary element methods. *Adv. Eng. Softw.* **121** (2018) 39–58.
- [22] J. Gray and G. J. Hancock, The propulsion of sea-urchin spermatozoa. *J. Exp. Biol* **32** (1955) 802–814.
- [23] L. Greengard and V. Rokhlin, A fast algorithm for particle simulations. *J. Comput. Phys.* **73** (1987) 325–348.
- [24] J.S. Guasto, K.A. Johnson and J.P. Gollub, Oscillatory flows induced by microorganisms swimming in two dimensions. *Phys. Rev. Lett.* **105** (2010) 168102.
- [25] M. Gurtin, An Introduction to Continuum Mechanics (Mathematics in Science and Engineering). Academic Press (1982).
- [26] E. Gutman and Y. Or, Optimizing an undulating magnetic microswimmer for cargo towing. *Phys. Rev. E* **93** (2016) 1–8.
- [27] B. Haasdonk and M. Ohlberger, Reduced basis method for finite volume approximations of parametrized linear evolution equations. *ESAIM: Math. Modell. Numer. Anal.* **42** (2008) 277–302.
- [28] E. Harris, The Chlamydomonas Sourcebook: A Comprehensive Guide to Biology and Laboratory Use. Elsevier Science, Burlington (1989).
- [29] M.A. Heroux, E.T. Phipps, A.G. Salinger, H.K. Thornquist, R.S. Tuminaro, J.M. Willenbring, A. Williams, S.S. Kendall, R.A. Bartlett, V.E. Howle, R.J. Hoekstra, J.J. Hu, T.G. Kolda, R.B. Lehoucq, K.R. Long and R.P. Pawlowski, An overview of the Trilinos project. *ACM Trans. Math. Softw.* **31** (2005) 397–423.
- [30] J.S. Hesthaven, G. Rozza and B. Stamm, Certified Reduced Basis Methods for Parametrized Partial Differential Equations. Springer International Publishing (2016).
- [31] K. Ishimoto, H. Gadêlha, E.A. Gaffney, D.J. Smith and J. Kirkman-Brown, Coarse-graining the fluid flow around a human sperm. *Phys. Rev. Lett.* **118** (2017) 1–5.
- [32] C. Josenhans and S. Suerbaum, Motility in bacteria. *Int. J. Med. Microbiol.* **291** (2002) 605–614.
- [33] G. Karypis and V. Kumar, A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM J. Sci. Comput.* **20** (1998) 359–392.
- [34] E.E. Keaveny, S.W. Walker and M.J. Shelley, Optimization of chiral structures for microscale propulsion. *Nano Lett.* **13** (2013) 531–537.
- [35] G.S. Klindt and B.M. Friedrich, Flagellar swimmers oscillate between pusher- and puller-type swimming. *Phys. Rev. E – Stat. Nonlinear Soft Matter Phys.* **92** (2015) 1–6.

- [36] M. Kronbichler, T. Heister and W. Bangerth, High accuracy mantle convection simulation through modern numerical methods. *Geophys. J. Int.* **191** (2012) 12–29.
- [37] T. Lassila, A. Quarteroni and G. Rozza, A reduced basis model with parametric coupling for fluid-structure interaction problems. *SIAM J. Sci. Comput.* **34** (2012) A1187–A1213.
- [38] T. Lassila, A. Manzoni, A. Quarteroni and G. Rozza, Model order reduction in fluid dynamics: challenges and perspectives. In: *Reduced Order Methods for Modelling and Computational Reduction, MS&A – Modeling, Simulation and Applications*, edited by A. Quarteroni and G. Rozza. Vol. 9. Springer Cham (2014) 235–273.
- [39] E. Lauga and T.R. Powers, The hydrodynamics of swimming microorganisms. *Rep. Prog. Phys.* **72** (2009) 096601.
- [40] E. Lauga, W.R. DiLuzio, G.M. Whitesides and H.A. Stone, Swimming in circles: motion of bacteria near solid boundaries. *Biophys. J.* **90** (2006) 400–12.
- [41] J. Lighthill, Flagellar hydrodynamics. *SIAM Rev.* **18** (1976) 161–230.
- [42] A. Manzoni, F. Salmoiraghi and L. Heltai, Reduced Basis Isogeometric Methods (RB-IGA) for the real-time simulation of potential flows about parametrized NACA airfoils. *Comput. Methods Appl. Mech. Eng.* **284** (2015) 1147–1180.
- [43] F. Negri, A. Manzoni and D. Amsallam, Efficient model reduction of parametrized systems by matrix discrete empirical interpolation. *J. Comput. Phys.* **303** (2015) 431–454.
- [44] N.C. Nguyen, G. Rozza and A.T. Patera, Reduced basis approximation and a posteriori errorestimation for the time-dependent viscous burgers equation. *Calcolo* **46** (2009) 157–185.
- [45] G. Noselli, A. Beran, M. Arroyo and A. DeSimone, Swimming euglena respond to confinement with a behavioural change enabling effective crawling. *Nat. Phys.* **15** (2019) 496–502.
- [46] E. Passov and Y. Or, Dynamics of Purcell’s three-link microswimmer with a passive elastic tail. *Eur. Phys. J. E* **35** (2012) 1–9.
- [47] N. Phan-Thien, T. Tran-Cong and M. Ramia, A boundary-element analysis of flagellar propulsion. *J. Fluid Mech.* **184** (1987) 533.
- [48] D. Pimponi, M. Chinappi, P. Gualtieri and C.M. Casciola, Hydrodynamics of flagellated microswimmers near free-slip interfaces. *J. Fluid Mech.* **789** (2016) 514–533.
- [49] M.E. Porter and W.S. Sale, The 9 + 2 axoneme anchors multiple inner arm dyneins and a network of kinases and phosphatases that control motility. *J. Cell Biol.* **151** (2000) F37–F42.
- [50] C. Pozrikidis, *Boundary Integral and Singularity Methods for Linearized Viscous Flow*. Vol. 36. Cambridge University Press, Cambridge (1992).
- [51] E.M. Purcell, Life at low Reynolds number. *Am. J. Phys.* **45** (1977) 3–11.
- [52] E.M. Purcell, The efficiency of propulsion by a rotating flagellum. *Biophysics* **94** (1997) 11307–11311.
- [53] 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. USA* **110** (2013) E338–E347.
- [54] M. Rossi, G. Cicconofri, A. Beran, G. Noselli and A. DeSimone, Kinematics of flagellar swimming in euglena gracilis: helical trajectories and flagellar shapes. *Proc. Nat. Acad. Sci.* **114** (2017) 13085–13090.
- [55] G. Rozza, D.B.P. Huynh and A.T. Patera, Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations. *Arch. Comput. Methods Eng.* **15** (2008) 229–275.
- [56] R. Sevilla, L. Borchini, M. Giacomini and A. Huerta, Hybridisable discontinuous Galerkin solution of geometrically parametrised Stokes flows. *Comput. Methods Appl. Mech. Eng.* **372** (2020) 113397.
- [57] L. Shi, S. Čanić, A. Quaini and T.W. Pan, A study of self-propelled elastic cylindrical micro-swimmers using modeling and computation. *J. Comput. Phys.* **314** (2016) 264–286.
- [58] H. Shum and E.A. Gaffney, Hydrodynamic analysis of flagellated bacteria swimming near one and between two no-slip plane boundaries. *Phys. Rev. E* **91** (2015) 033012.
- [59] H. Shum, E.A. Gaffney and D.J. Smith, Modelling bacterial behaviour close to a no-slip plane boundary: the influence of bacterial geometry. *Proc. R. Soc. A: Math. Phys. Eng. Sci.* **466** (2010) 1725–1748.
- [60] K. Son, J.S. Guasto and R. Stocker, Bacteria can exploit a flagellar buckling instability to change direction. *Nat. Phys.* **9** (2013) 494–498.
- [61] O. Steinbach, *Numerical Approximation Methods for Elliptic Boundary Value Problems*. Springer New York, New York, NY (2008).
- [62] T.T.M. Ta, V.C. Le and H.T. Pham, Shape optimization for stokes flows using sensitivity analysis and finite element method. *Appl. Numer. Math.* **126** (2018) 160–179.

- [63] A.C.H. Tsang, A.T. Lam and I.H. Riedel-Kruse, Polygonal motion and adaptable phototaxis via flagellar beat switching in the microswimmer *Euglena gracilis*. *Nat. Phys.* **14** (2018) 1216–1222.
- [64] D. Walker, M. Kübler, K.I. Morozov, P. Fischer and A.M. Leshansky, Optimal length of low Reynolds number nanopropellers. *Nano Lett.* **15** (2015) 4412–4416.

## Subscribe to Open (S2O)

### A fair and sustainable open access model



This journal is currently published in open access under a Subscribe-to-Open model (S2O). S2O is a transformative model that aims to move subscription journals to open access. Open access is the free, immediate, online availability of research articles combined with the rights to use these articles fully in the digital environment. We are thankful to our subscribers and sponsors for making it possible to publish this journal in open access, free of charge for authors.

#### **Please help to maintain this journal in open access!**

Check that your library subscribes to the journal, or make a personal donation to the S2O programme, by contacting [subscribers@edpsciences.org](mailto:subscribers@edpsciences.org)

More information, including a list of sponsors and a financial transparency report, available at: <https://www.edpsciences.org/en/maths-s2o-programme>