

# Bioinspired swimming simulations



Michel Bergmann<sup>a,b,c,\*</sup>, Angelo Iollo<sup>a,b,c</sup>

<sup>a</sup> Memphis Team, INRIA, F-33400 Talence, France

<sup>b</sup> Univ. Bordeaux, IMB, UMR 5251, F-33400 Talence, France

<sup>c</sup> CNRS, IMB, UMR 5251, F-33400 Talence, France

## ARTICLE INFO

### Article history:

Received 4 March 2016

Received in revised form 18 July 2016

Accepted 19 July 2016

Available online 26 July 2016

### Keywords:

Bioinspired swimming

Level set

Skeleton

Optimal transportation

Cartesian mesh

Penalization

## ABSTRACT

We present a method to simulate the flow past bioinspired swimmers starting from pictures of an actual fish. The overall approach requires i) a skeleton graph generation to get a level-set function from pictures; ii) optimal transportation to obtain the velocity on the body surface; iii) flow simulations realized with a Cartesian method based on penalization. This technique can be used to automate modeling swimming motion from data collected by biologists. We illustrate this paradigm by simulating the swimming of a mackerel fish.

© 2016 Elsevier Inc. All rights reserved.

## 1. Introduction

Autonomous swimming robots can address in the future needs such as search and rescue, environmental monitoring and emergency response in rapidly unfolding scenarios. Fishes have the ability to evolve in complex and unpredictable environments that may abruptly change. They are capable of maneuvering in ways that can hardly be achieved by present engineering devices. In this sense, the design of such devices can significantly benefit from bioinspired principles.

However, quantitative evaluation of the geometrical and dynamical characteristics of swimming in experiments is challenging [1,2]. In particular, precise control of the kinematics and dynamics of the experimental model is difficult; harder still is the accurate measurement of forces and power of a freely swimming model. In this regard, numerical models are well suited for such investigation.

Three-dimensional numerical simulation of swimming has only become viable in the last decade [3–8] with the progress of both numerical schemes and computers architectures. Nonetheless, many of the studies presented in the literature typically deal with prescribed swimmer geometries. Only few start from actual fish geometries, like for example Patankar's group [4,5,9] and Mittal's group [3,10,11].

The aim of this paper is to model a three-dimensional swimmer starting from actual fish pictures. The first step is to build the three-dimensional fish profile starting from two-dimensional data retrieved from one or several subsequent pictures of a fish. To this end, a skeleton technique [12] is used to reconstruct the three-dimensional level set function describing the body surface. Then the skeleton can be deformed using an appropriate swimming law to obtain a sequence of level set functions corresponding to snapshots of the body surface uniformly taken at different instants. Otherwise,

\* Corresponding author at: Univ. Bordeaux, IMB, UMR 5251, F-33400 Talence, France.

E-mail address: [michel.bergmann@inria.fr](mailto:michel.bergmann@inria.fr) (M. Bergmann).

URL: <http://www.math.u-bordeaux.fr/~mbergman/> (M. Bergmann).

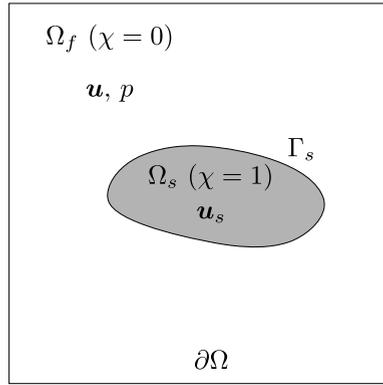


Fig. 1. Sketch of the general flow configuration.

if a sequence of pictures of swimming is available, the body surface can be reconstructed with the same technique for each snapshot. The level set functions give at each instant an implicit description of the surface. This is quite different from usual swimming simulations where the deforming body is fully described in a Lagrangian framework.

With skeleton deformation or with a sequence of actual pictures, we typically can reconstruct only a small number of the snapshots necessary to simulate a swimming stroke. This is due to the fact that the time scale of the simulation is significantly smaller than the time step between two subsequent reconstructed three-dimensional snapshots. Moreover, the surface deformation velocity is required to set the boundary conditions of the flow problem. For this reason it is necessary to build intermediate level-set functions and to compute the deformation velocity field between subsequent reconstructed fish snapshots. Optimal transportation [13] is well suited to achieve this goal providing an objective model to compute intermediate geometries and deformation velocities.

Optimal transportation is an old topic first introduced by Monge in 1781 [14]. It consists in a distance minimization problem that allows to define a mapping between an initial and a final mass distribution. This problem has attracted considerable attention these last years especially from a numerical point of view [15–19]. Here we exploit the scheme introduced in [20] to define a non-linear interpolation mapping between geometries at different instants and to compute the corresponding deformation velocities.

In the following we describe the numerical scheme, the swimmer kinematic modeling and show validations with examples of applications.

## 2. Numerical modeling

### 2.1. Flow problem

The flow configuration is given in Fig. 1 where  $\Omega_s$  is the domain defined by the body,  $\Omega_f$  is the domain filled by the fluid, and  $\Omega = \Omega_f \cup \Omega_s$  is the whole domain. The boundary between  $\Omega_f$  and  $\Omega_s$  is noted  $\Gamma_s$ .

The flow is modeled using the incompressible Navier–Stokes equations written in the fluid domain  $\Omega_f$  as follows

$$\rho \left( \frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} \right) = -\nabla p + \nabla \cdot 2\mu D(\mathbf{u}) + \rho \mathbf{g} \text{ in } \Omega_f, \tag{1a}$$

$$\nabla \cdot \mathbf{u} = 0 \text{ in } \Omega_f, \tag{1b}$$

where  $D(\mathbf{u}) = \frac{\nabla \mathbf{u} + \nabla^T \mathbf{u}}{2}$ ,  $\mathbf{u} = (u, v, w)$  is the three-dimensional velocity field,  $p$  is the pressure field,  $\mu$  is the dynamic viscosity,  $\rho$  is the density and  $\mathbf{g}$  is the gravity. Unsteady boundary conditions are imposed on the body boundary

$$\mathbf{u}(\mathbf{x}, t) = \mathbf{u}_s(\mathbf{x}, t) \text{ on } \Gamma_s, \tag{1c}$$

and on the external boundary

$$\mathbf{u}(\mathbf{x}, t) = \mathbf{u}_{ext}(\mathbf{x}, t) \text{ on } \partial\Omega. \tag{1d}$$

The dynamics of the boundary  $\Gamma_s$  that represents the swimmer will be described later.

Classically, system (1) is discretized using a body fitted mesh such that the boundary condition on the body boundary can be easily and accurately taken into account. However, these kind of meshes are difficult to generate in complex unsteady geometries in three-dimensions as large mesh deformations are required when the body is displaced or is changing topology.

In this study we will use a fixed Cartesian grid. The body boundary is defined and tracked with a level set function [21] and the boundary condition on  $\Gamma_s$  is taken in account in the Navier–Stokes equation using a penalty term [22].

We thus consider the penalized Navier–Stokes equations in the whole domain  $\Omega$  (fluid and body):

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} = -\frac{1}{\rho} \nabla p + \frac{1}{\rho} \nabla \cdot 2\mu D(\mathbf{u}) + \mathbf{g} + \chi \lambda (\mathbf{u}_s - \mathbf{u}) \quad \text{in } \Omega, \quad (2a)$$

$$\nabla \cdot \mathbf{u} = 0 \quad \text{in } \Omega, \quad (2b)$$

where boundary conditions on the body boundary are implicitly imposed through the penalty term  $\chi \lambda (\mathbf{u}_s - \mathbf{u})$ . In the penalty term,  $\chi$  is the characteristic function ( $\chi(\mathbf{x}) = 1$  if  $\mathbf{x} \in \Omega_s$  and  $\chi(\mathbf{x}) = 0$  elsewhere) and  $\lambda$  is a large penalty parameter. The value of  $\lambda$  depends on the numerical scheme used to discretize the penalty term in the momentum equation. If this term is solved in an explicit way,  $\lambda \leq \rho/\Delta t$ , where  $\Delta t$  is the numerical time step. Otherwise, with an implicit discretization of the penalty term, no restrictions are imposed for the value of  $\lambda$ . It has been proven [23] that the solution of the system (2) tends to the solution of the system (1) as  $1/\sqrt{\lambda}$  and that there exist optimal  $\lambda$  with respect to the space discretization. We chose arbitrarily  $\lambda = 10^8$ .

In what follows, the characteristic function will be computed from the signed distance function  $\phi$ , so that  $\chi = H(\phi)$  where  $H$  denotes the Heaviside function.

The system (2) is solved in time using a projection method [24,25] and is discretized in space on a uniform Cartesian mesh using second order centered finite differences for the diffusive term and third-order upwind finite differences for the convective term.

The modeling, the details of the numerical method, as well as some validations of our approach can be found in [26,8]. Other immersed boundary methods based on Lagrange multipliers can impose the non-slip condition on the immersed surface in a different way [27–29]. We have proven that the present scheme is second-order accurate in space for velocity fields (in both  $L_2$  and  $L_\infty$  norms), and first-order accurate for the pressure field [26]. This makes our approach suitable to perform accurate simulations, at least for moderate Reynolds numbers.

The resolution of system (2) requires knowledge of both the unsteady characteristic function  $\chi$  and the body velocity  $\mathbf{u}_s$ . The following sections will explain how to compute these quantities.

The velocity field in a self propelled body can be decomposed such that

$$\mathbf{u}_s(\mathbf{x}, t) = \bar{\mathbf{u}}(\mathbf{x}, t) + \mathbf{u}^\theta(\mathbf{x}, t) + \tilde{\mathbf{u}}(\mathbf{x}, t), \quad (3)$$

where  $\tilde{\mathbf{u}}(\mathbf{x}, t)$ ,  $\bar{\mathbf{u}}(\mathbf{x}, t)$  and  $\mathbf{u}^\theta(\mathbf{x}, t)$  are the deformation, the translation and the rotation velocities, respectively. While the deformation velocity is imposed (by muscles), the translation and rotation velocities are the results of the hydrodynamic forces and torques applied by the fluid onto the body. These latter are computed integrating the Newtons laws. Once the body geometry is obtained (or imposed), we can deform the backbone (the midline) following swimming laws observed in nature. Considering that the backbone of the steady body is  $0 \leq x \leq \ell$ ,  $y = 0$  and  $z = 0$ , we can apply a swimming law in the plan  $(x, z)$  that remains the backbone length to be constant.

The swimming law observed in nature for most of fishes is [30]:

$$z(x, t) = a(x) \sin(2\pi(x/\lambda + ft)), \quad (4a)$$

where the envelop is

$$a(x) = c_0 + c_1 x + c_2 x^2. \quad (4b)$$

This swimming law is defined with a constant phase velocity  $c_p = \lambda f$ , where  $\lambda$  and  $f$  are the wave length and the frequency of the oscillations. The parameters  $c_0$ ,  $c_1$  and  $c_2$  are adjusted to reach a maximal tail amplitude  $A/2$  that is an important parameter for the swimming [31]. It has been shown [32,33] that carangiform swimming mode is close to  $c_0 = 0.02$ ,  $c_1 = -0.12$  and  $c_2 = 0.2$  for a unit length fish with  $0 \leq x \leq 1$ . These parameters can be rescaled when modifying the fish length.

### 3. Generation of level set functions from pictures: the skeleton approach

Numerical simulations of fish-like swimmers [34,8] or jellyfishes [26] are possible with the scheme present above. The swimmer geometries were prescribed to mimic realistic fishes or jellyfishes. The motion of these swimmers was described in a purely Lagrangian way by tracking markers distributed on the swimmer boundary. The objective is now to simulate bioinspired swimmers using geometries and possibly deformation laws obtained by pictures or sequences of pictures. As an example we consider the definition of a swimmer starting from the mackerel fish presented in Fig. 2 (left column) [35].

As mentioned in §2, our numerical approach requires the knowledge of the level set function (distance function) associated with a body to get a second order scheme in space. A characteristic function (mask function) is sufficient for a first order penalty method [34].

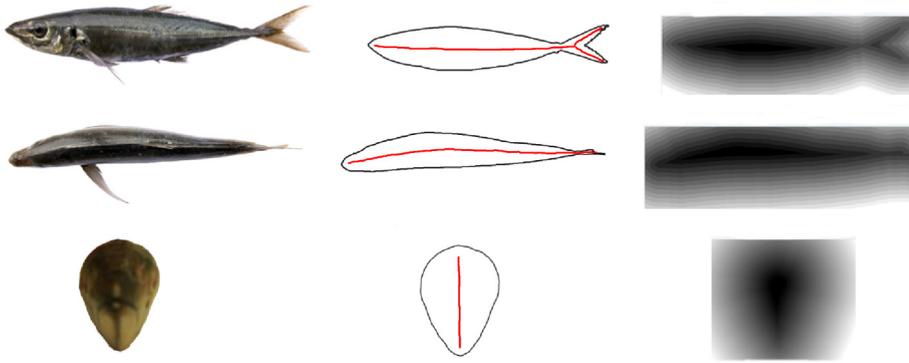


Fig. 2. Skeleton steps: pictures (left), segmentation with skeleton (center) and associated level set functions (right).



Fig. 3. Re-centering (top view).

The process is synthesized in Fig. 2. We start with pictures of a mackerel fish. Top, bottom and side views are necessary. To rebuild fish in 3D from only three pictures (front, side, top), an underlying assumption is made: the whole cross-section of the fish along its length is geometrically similar to the fish on front view (front picture). A dilation takes place in width and another takes place in height. In the tail region of the fish, two connected components can appear on the cross section, each connected component will be geometrically similar to the front view. The scaling factors will be extracted from the thickness of the fish (from the side view) and the width (from the top view), according to the longitudinal position of the transverse cut.

Segmentations are first performed to extract the boundaries (black lines in center column). The distance function (right column) can be computed by solving Eikonal equation. Then the skeleton is computed (red lines in the center column) as being the center points of the inscribed circles. The natural tool to identify this information is the centerline or skeleton of an object as recalled in [12]. It identifies the locus of points equidistant from further points of the boundary. The special feature here is that the relevant skeleton is not constructed from the Euclidean distance. In order to identify the thicknesses and the widths of the fish, the distance is measured in a fixed direction (depending on the height or along the width) of the side view or top view. The skeleton is then simplified compared to the skeleton associated with the Euclidean distance. Thus, the top view will have a skeleton made of a line from the head to the tail of the fish, with a single point of the skeleton on each transverse cut.

The skeleton of the profile view for the vertical distance will consist of a central line splitting in the fish's tail. Finally, for each transverse cut of the profile, the number of connected components of the profile view is the number (1 or 2) of points of the skeleton of this cut. This property is true thanks to the choice of the distance and is not true with Euclidean distance.

For a given cut, let us denote  $a$  a point of the skeleton of the profile view and let  $b$  be the point of the skeleton of the top view. Denote  $d_a$  the vertical distance from  $a$  to the boundary of the fish and  $d_b$  the horizontal distance from  $b$  to the boundary of the fish. The connected component of the cut fish associated to  $(a, b)$  can be rebuilt by a scaling (based on factors  $d_a$  and  $d_b$ ) of the front view and translated according to the position  $(a, b)$ . It also becomes possible to mirror the top view of the fish (or front view) by rectifying the skeleton in a line to correct the defects of shooting fish in the picture, see Fig. 3.

Initial rescaling of pictures has been performed so that the fish length is the same for top and side views, and height is the same for top and side views. A simple 3D reconstruction of the fish could then take place, but the goal is to deform the fish, which will have the effect of losing the notion of directional distance, more complex work is to be implemented. First, the skeleton of the top view (for the vertical distance)  $S_v$  is distorted by law (4). The concept of directional distance is no longer suitable for reconstruction, then the chosen approach is to distort the classic skeleton associated with Euclidean distance, relying on distorted  $S_v$ . This new skeleton allows to reconstruct, for the Euclidean distance, the deformed fish seen from top view. Then, the 3D reconstruction of the fish can be implemented by harnessing both concepts skeleton as shown in Fig. 4 (right). It is thus possible to generate a series of level set functions corresponding to different time  $t$  for the swimming law (4). In what follows, we considered the periodical swimming law (4) and uniformly extract  $N$  snapshots for the level set function over one period (one swimming stroke). Finally, the level set functions have been modified (transported) with a rigid velocity so that we do not add any linear or angular translations during deformation. Linear and angular translations have only to be the results of the hydrodynamic forces and torques.

Note that in what follows, the level set are chosen to be the signed distance functions (positive inside the body).



Fig. 4. Three-dimensional profiles defined by the zero isocontour of the level set function of the actual fish. Undeformed (left) and deformed (right).

#### 4. Computation of intermediate level-sets and deformation velocities: optimal transportation

As mentioned, we have access to  $N$  (here  $N = 20$ ) snapshots. We define by  $\{\varphi^k\}_{k=0,\dots,N-1}$  the level set function over a swimming period  $T$  referred to the center of mass of each image. Here,  $k$  denotes the time  $t = kT i/N$  for the  $i$ th swimming period. The number of available snapshots is of course inessential, but it is typically small compared to the number of snapshots necessary for a numerical simulation. It is then necessary to build missing level set function corresponding to the Navier–Stokes time discretization. Moreover, it is also necessary to compute the fish velocity for the penalty method. This velocity can be decomposed into a deformation velocity, to be computed from the subsequent  $N$  level set functions, and a rigid velocity computed from hydrodynamics effects. The deformation velocity is derived from an optimal transportation model.

Optimal transportation is a model to compute a mapping between two densities. The model assumes that the mass is transported along straight lines with constant initial velocity. The initial velocity is found such that to overall distance measured in a given norm is minimum. The formulation of the  $L^2$  optimal transportation problem is illustrated with respect to generic densities  $\varrho_0$  and  $\varrho_1$ . The main step is to compute the velocity field transforming  $\varrho_0$  into  $\varrho_1$ . Then it is possible to compute any desired intermediate densities for time  $t = rT i/N$  with  $0 < r < 1$ .

Let  $\varrho_0(\xi), \varrho_1(x)$  be two smooth enough non-negative scalar density functions with compact support  $\Omega_0$  and  $\Omega_1$ , where  $\xi, x \in \mathbb{R}^d$  and  $d$  is the space dimension. We assume that

$$\int_{\Omega_0} \varrho_0(\xi) d\xi = \int_{\Omega_1} \varrho_1(x) dx.$$

Let  $X : \Omega_0 \rightarrow \Omega_1$  a smooth one-to-one map such that  $X(\xi)$  realizes the transfer of  $\varrho_0$  onto  $\varrho_1$ , i.e., a map that satisfies the following Jacobian equation:

$$\varrho_0(\xi) = \det(\nabla X(\xi)) \varrho_1(X(\xi)).$$

This equation is under determined with respect to  $X(\xi)$  and a solution is selected among all possible maps by introducing the following  $L^2$  Kantorovich–Wasserstein distance:

$$\inf_X \int_{\Omega_0} \varrho_0(\xi) |X(\xi) - \xi|^2 d\xi.$$

The  $L^2$  Monge–Kantorovich problem (MKP) corresponds to finding a map  $X_0$  such that this infimum is achieved. It has been proved that this problem admits a unique solution [36,37,13], which is the gradient of a.e. convex function  $\Psi_0 : \Omega_0 \rightarrow \mathbb{R}$ :

$$X_0(\xi) = \nabla \Psi_0(\xi).$$

The main idea of solution relies on the assumption that we dispose of a mapping at iteration  $n$  that is a perturbation of the optimal mapping  $X_0(\xi)$  [20]. We derive a linear perturbation equation that is used to iteratively improve the initial guess. Let us assume that the mapping obtained at iteration  $n$  is

$$X^n(\xi) = \nabla_\xi \Psi_0 + \nabla_\xi \Psi_\epsilon^n,$$

where the error  $\Psi_\epsilon^n$  satisfies  $\|\Psi_\epsilon^n\|_2 \approx \epsilon$ . We define  $\varrho_0^n(\xi)$  as the initial density at iteration  $n$  that mapped by  $X^n(\xi)$  gives the exact final density  $\varrho_1(x)$ . Then, taking a first-order Taylor expansion, we have

$$\begin{aligned} \varrho_0^n(\xi) &:= \varrho_1(X^n(\xi)) \det(\nabla_\xi X^n(\xi)) \\ &= \underbrace{\varrho_1(X_0) \det(\nabla_\xi X_0)}_{=\varrho_0(\xi)} + \det(\nabla_\xi X_0) \varrho_1(X_0) \text{Tr}\left((\nabla_\xi X_0)^{-1} \nabla_\xi^2 \Psi_\epsilon^n\right) + \det(\nabla_\xi X_0) \nabla_x \varrho_1(X_0) \cdot \nabla_\xi \Psi_\epsilon^n + o(\epsilon), \end{aligned}$$

where  $\text{Tr}$  denotes the matrix trace operator.

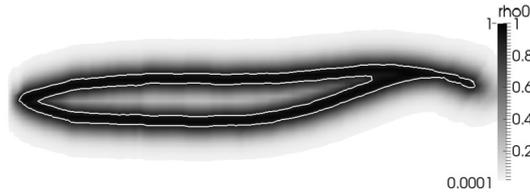


Fig. 5. Gaussian density used in the plan  $y = 0$ . The external white contour corresponds to the fish boundary.

At first order in  $\epsilon$ , we have

$$\frac{\varrho_0^n(\xi) - \varrho_0(\xi)}{\det(\nabla_\xi X_0)} \approx \varrho_1(X_0) \text{Tr}\left((\nabla_\xi X_0)^{-1} \nabla_\xi^2 \Psi_\epsilon^n\right) + \nabla_x \varrho_1(X_0) \cdot \nabla_\xi \Psi_\epsilon^n. \tag{5}$$

The mapping update is then found by the iteration

$$X^{n+1} = X^n - \alpha \nabla \Psi_\epsilon^n,$$

that converges to  $X_0$  as a geometric series, for  $\alpha \in [0, 1]$ , in the linearized approximation [38,17].

If the optimal solution and the approximated mapping are a perturbation of identity, then equation (5), with  $(\nabla_\xi X_0)^{-1} = \text{Id}$  and  $\text{Tr}(\nabla_\xi^2 \Psi_\epsilon^n) = \Delta_\xi \Psi_\epsilon^n$ , reduces to

$$\varrho_0^n(\xi) - \varrho_0(\xi) \approx \nabla_\xi \cdot (\varrho_1(X^n(\xi)) \nabla_\xi \Psi_\epsilon^n) \tag{6}$$

that is equivalent to a semidiscretization in time of the continuity equation written in Eulerian form.

#### 4.1. Choice of the density functions

Since the level set functions corresponding to the  $N$  fish snapshots are not strictly positive, they are not admissible candidates for optimal transportation. A regularized mask function (Heaviside of the level set function) is also not satisfying since the weight of the thin caudal fin in the mapping is very limited and the mapping tends to smear it in comparison to the whole body.

We thus considered Gaussian density functions defined from the level sets  $\{\varphi^k\}_{k=0,\dots,N-1}$ , with a support that is approximately equal to  $\ell/10$  ( $\ell$  being the length of the fish) with maximum near the interface but inside the body. In particular, we take  $\varrho_0 = G(\xi, k) = \exp\left(\frac{(\varphi^k(\xi, t) - \varphi_b)^2}{\gamma}\right) + \epsilon$ , with  $\gamma = 2 \cdot 10^{-5}$ ,  $\varphi_b = 1.5 \cdot 10^{-3}$  and the regularization parameter  $\epsilon = 10^{-3}$ . In a similar way, we take  $\varrho_1 = G(\xi, k + 1)$ . Using these definitions of the densities to be transported, each point of the body surface has a comparable weight in the transportation. We denote by  $G_0$  the value of the Gaussian function corresponding to the swimmer surface. The density  $G(\xi, k)$  is presented in Fig. 5 where the external white line is the body boundary  $G_0$ .

#### 4.2. Computation of intermediate images

Starting from the optimal mapping, the computation of intermediate images is straightforward. Let  $X(\xi, \tau) = \xi + \tau(X_0 - \xi)$  and

$$\varrho(\xi, \tau) = \frac{\varrho_0(\xi)}{\det(\nabla_\xi X(\xi, \tau))}$$

We have  $\varrho(\xi, 0) = \varrho_0(\xi)$ ,  $\varrho(\xi, 1) = \varrho_1(X_0(\xi))$  and for  $0 < \tau < 1$  we obtain the intermediate images.

As for the initial deformation velocity, given the time step  $\Delta t_s$  between two images (here  $\Delta t_s = T/20$ ), the deformation velocity from  $\varrho_0$  to  $\varrho_1$  is

$$\tilde{\mathbf{u}}^0 = \frac{X_0 - \xi}{\Delta t_s} \tag{7}$$

The index 0 highlights the fact that this velocity has to be mapped to the actual position of the swimmer in space, since the level set snapshots are taken with respect to a center of mass fixed in space. Fig. 6 shows the reconstruction the intermediate density for  $\tau = 0.5$  between densities  $\varrho_0$  and  $\varrho_1$ .

The same approach can be applied to the whole series of level sets  $\{\varphi_i\}_{i=0}^N$ , thus computing the corresponding deformation velocities and intermediate densities. Starting from these densities we can compute a characteristic function and a sign distance function using the fast marching approach [39,21] or by solving an Hamilton Jacobi equation [40,41]. In what follows, we recall that  $\varphi(\xi, t)$  are the level set function of the deformed profiles with fixed center of mass and  $\tilde{\mathbf{u}}^0(\xi, t)$  is the deformation velocity.



**Fig. 6.** Construction of an image for  $\tau = 0.5\Delta t_s$  (red) between  $\varrho_0$  (green) and  $\varrho_1$  (blue) in the plan  $y = 0$ . (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

## 5. Dynamics of the swimmer

### 5.1. Computation of the swimmer velocity and position

Since we are interested in a self propelled swimmer, the position of the swimmer is determined with the level set function  $\phi(\xi, t)$  that is an image of the level set function  $\varphi(\xi, t)$  obtained by the transformation characterizing the rigid motion (translation plus rotation),  $X^r(\xi, t)$ . We have

$$\phi(\xi, t) = \varphi(X^r(\xi, t)).$$

The characteristic function  $\chi$  required in the penalty method is

$$\chi(\xi, t) = H(\phi(\xi, t) - G_0),$$

where the constant  $G_0$  is defined in §4.1. We can then impose  $\chi(\xi, t) = 1$  inside the whole body, and compute the signed distance function for second order penalty method [26].

In the same way, the velocity of the swimmer is  $\mathbf{u}^0(\xi, t) = \bar{\mathbf{u}}(\xi, t) + \mathbf{u}^\theta(\xi, t) + \tilde{\mathbf{u}}^0(\xi, t)$ , where  $\tilde{\mathbf{u}}^0(\xi, t)$  is computed from (7). Since the position of the swimmer is modified, the velocity has to act onto the modified position and we have:

$$\mathbf{u}_s(\xi, t) = \mathbf{u}^0(X^r(\xi, t)).$$

The transformation  $X^r(\xi, t)$  as well as the translation velocity  $\bar{\mathbf{u}}(\xi, t)$  and rotation velocity  $\mathbf{u}^\theta(\xi, t)$  are evaluated using the Newton's laws computing the forces and the torques exerted by the fluid onto the body.

### 5.2. Computation of the forces and the torques

The forces and the torques exerted by the fluid onto the body could be computed integrating the stress tensor  $\mathbb{T}(\mathbf{u}, p) = -p\mathbf{I} + \mu(\nabla\mathbf{u} + \nabla\mathbf{u}^T)$  on the body boundary. Let  $\mathbf{n}$  be the unit outward normal vector to the interface  $\Gamma$ , and  $\mathbf{r} = \mathbf{x} - \mathbf{x}_G$  with  $\mathbf{x}_G$  the mass center, the force and the torques write:

$$\mathbf{F} = - \int_{\Gamma_s} \mathbb{T}(\mathbf{u}, p) \mathbf{n} \, d\mathbf{x}, \quad (8a)$$

$$\mathcal{M} = - \int_{\Gamma_s} \mathbf{r} \wedge \mathbb{T}(\mathbf{u}, p) \mathbf{n} \, d\mathbf{x}. \quad (8b)$$

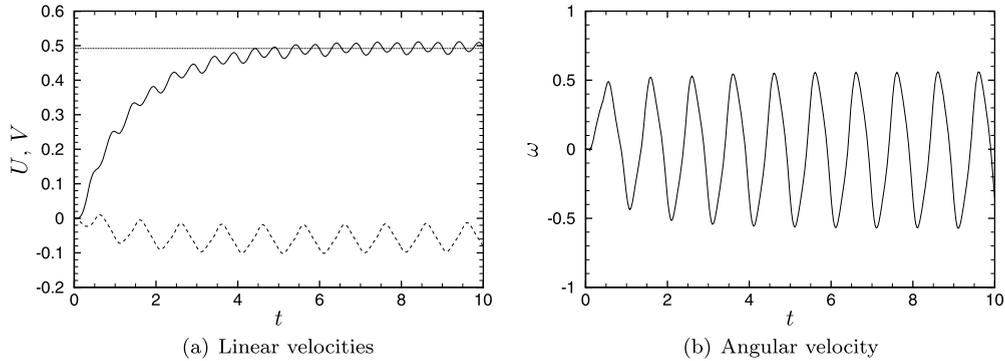
However, since the body boundary is defined in an implicit way as being the zero level of the sign distance function, integration of equations (8) is thus not straightforward. Thus we turn to moment balance as follows.

We consider an arbitrarily subdomain  $\tilde{\Omega}_f$  delimited by the boundaries  $\Gamma_s$  and  $\tilde{\Gamma}_f$  surrounding the body with domain  $\Omega_s$  such that the forces and torques write [42]:

$$\mathbf{F} = - \frac{d}{dt} \int_{\tilde{\Omega}_f(t)} \mathbf{u} \, dV + \int_{\tilde{\Gamma}_f(t)} (\mathbb{T} + (\mathbf{u} - \mathbf{u}_f) \otimes \mathbf{u}) \mathbf{n} \, dS - \int_{\Gamma_s(t)} ((\mathbf{u} - \mathbf{u}_s) \otimes \mathbf{u}) \mathbf{n} \, dS. \quad (9a)$$

$$\mathcal{M} = - \frac{d}{dt} \int_{\tilde{\Omega}_f(t)} \mathbf{r}_i \wedge \mathbf{u} \, dV + \int_{\tilde{\Gamma}_f(t)} \mathbf{r} \wedge (\mathbb{T} + (\mathbf{u} - \mathbf{u}_f) \otimes \mathbf{u}) \mathbf{n} \, dS - \int_{\Gamma_s(t)} \mathbf{r} \wedge ((\mathbf{u} - \mathbf{u}_s) \otimes \mathbf{u}) \mathbf{n} \, dS. \quad (9b)$$

where  $\mathbf{u}_s$  and  $\mathbf{u}_f$  are respectively the velocities of boundaries  $\Gamma_s$  and  $\tilde{\Gamma}_f$ , that might in general be different from that of the fluid. Of course, in many applications (like for swimmers), the integral over the  $\Gamma_s$  boundary vanish due to the fact that we have  $\mathbf{u} = \mathbf{u}_s$ . In the following simulations,  $\tilde{\Gamma}_f$  coincides with the external boundary of the domain where homogeneous boundary conditions are applied, and thus  $\mathbf{u}_f = 0$ .



**Fig. 7.** Evolution of the linear ( $U$  black line,  $V$  dashed line) and angular ( $\omega$ ) swimming velocities, normalized by the wave velocity  $\ell f$  for the eel test case presented in [43] with respect to the time, which is normalized by  $1/f$ .

## 6. Validation and application to actual pictures

### 6.1. Validation

#### 6.1.1. Validation for free swimmer defined by markers

The numerical solver has been already intensively validated for several bodies where the geometry is defined by markers and the body followed in a Lagrangian way [34,26,8]. We now present a validation test case for a free swimming three-dimensional eel. This test case has been set up using conforming mesh by Kern and Koumoutsakos [43] from original two-dimensional simulations done by Carling et al. [44]. This three-dimensional eel has also been studied numerically in [45] and [46] using immersed boundary leading to a good benchmarking test. We prescribe the deformation kinematics of the eel midline while the translation and rotation velocities are computed from fluid-structure interactions. Both the geometry profile and the deformation kinematics of the eel are fully described in [43]. The curvature of the backbone is prescribed along the arc length  $s$  of the midline of length  $\ell$  as

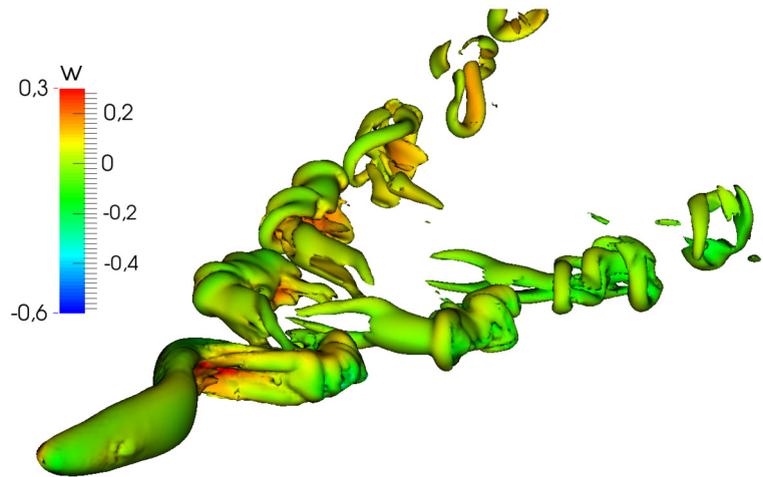
$$\kappa(s, t) = \frac{\partial \alpha}{\partial t} = K(s) \sin(2\pi(ft - \tau(s))) \tanh(\pi ft). \quad (10)$$

The amplitude envelope  $K(s)$  is defined by a cubic spline that takes the values  $K_0$ ,  $K_1$ ,  $K_2$  and  $K_3$  at  $s = 0$ ,  $\ell/3$ ,  $2\ell/3$  and  $\ell$ , respectively,  $\tau(s) = s\tau_{tail}/\ell$ ,  $f$  is the frequency of the traveling wave along the body and  $\alpha$  is the angle made by the tangent vector to the midline and the axis defined by the initial body midline at rest. The Cartesian coordinates can then be easily obtained. The transient deformation kinematics during the initial period  $1/f$  is modeled by  $\tanh(\pi ft)$ . The computational domain is  $8\ell \times 4\ell \times \ell$  which is uniformly discretized by  $1024 \times 512 \times 128$  grid points leading to 128 mesh points along the body length. We have shown [8] that 128 points along the swimmer represents the best compromise between accuracy and computational costs with our numerical method. We chose the same physical constants and deformation kinematics parameters given in [43] to maximize the eel velocity, i.e.  $K_0 = 1.51$ ,  $K_1 = 0.48$ ,  $K_2 = 5.74$ ,  $K_3 = 2.73$ , and  $\tau_{tail} = 1.44$ . The frequency is set to  $f = 1$  Hz and the body length is  $\ell = 1$  m. The adaptive time step is chosen with a classical CFL condition with  $cfl = 0.2$ . The temporal evolutions of the linear and angular swimming velocities for the eel test case presented in [43] are plotted in Fig. 7. The results are normalized using the body length  $\ell$  and the stroke frequency  $f$ , see [46]. The linear velocities represent thus body length per stroke. These velocities are computed using Newtons laws from forces and torques computed from (8). Despite the normalized axial swimming velocity at steady state slightly exceeds the ones obtained in [43] and [46] (0.51 vs. 0.47), we note good agreements between the overall behavior of normalized linear and angular velocities, especially with [46] where the same transient is used. Note that, as in [46], the  $V$ -velocity does not oscillate with zero mean value due to the initial stroke. Fig. 8 shows the isovorticity surface with magnitude 4. The overall behavior is consistent with previous studies [43] and [46].

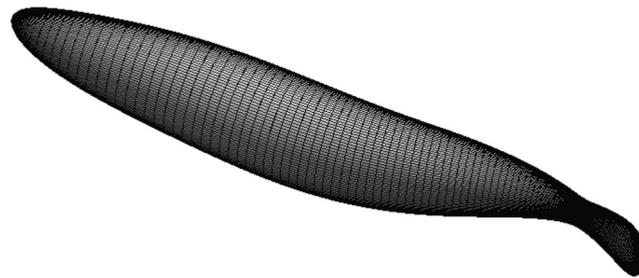
#### 6.1.2. Validation for free swimmer defined by pictures

Our goal is then to validate the proposed method using optimal transportation comparing it with a well defined test case. All the following numerical simulations are performed using Cartesian mesh with  $\Delta x = \Delta y = \Delta z = \ell/128$  on 256 CPUs.

In this section we do not consider an actual sequence of fish snapshots. We consider a fish model where the geometry is known and defined by markers. The geometry is however chosen to fit as best as possible the mackerel fish, except the lunate tail. Fig. 9 presents the fish geometry for this test case as well as the markers (note that we actually use two times more markers in each directions).



**Fig. 8.** Three-dimensional isovorticity structure with magnitude 4 colored by z-velocity field,  $w$ . The same representation parameters as in [46] is used. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)



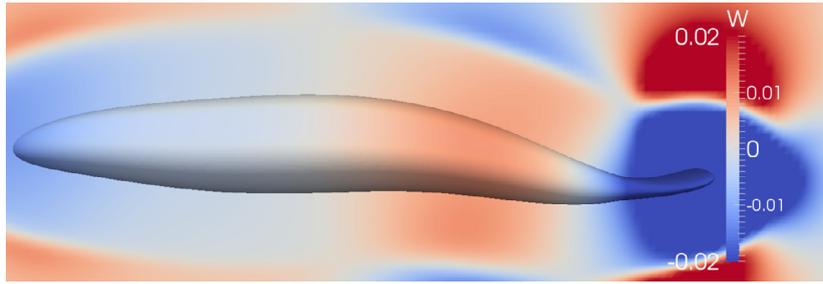
**Fig. 9.** Geometry profile used to validate the method.

Our goal is then to compare the two numerical models for the fish just defined:

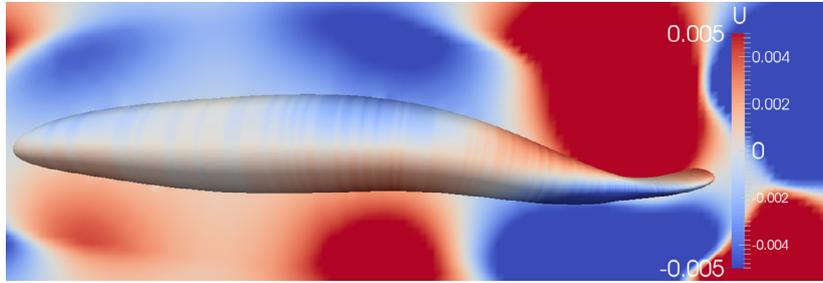
- Case I (“Lagrangian”): the level set functions and the deformation velocities are given at each Navier–Stokes time step by following the Lagrangian markers on the body surface;
- Case II (“Eulerian”): we suppose that we only know 20 snapshots of the level set function over one stroke period. We have thus to compute all the missing level set as well as the deformation velocity using optimal transportation.

In both cases the length fish is  $\ell = 10$  cm and the swimming law (4) is defined with frequency  $f = 4$  Hz and  $c_0 = -0.002$ ,  $c_1 = -0.12$ ,  $c_2 = 2$  and  $\lambda = \ell$ . These parameters are close to those observed in nature [33]. We defined the swimming law in the plan  $(x, z)$  leading to associated deformation velocities fields  $(\tilde{u}^0, \tilde{w}^0)$  computed from (7). A comparison of the deformation velocity fields  $\tilde{u}^0$  and  $\tilde{w}^0$  obtained in cases I (“exact”) and II (approximated using optimal transportation) is presented in Fig. 10. The Lagrangian velocity is only defined inside the body while the Eulerian one is defined in the whole domain where optimal transportation is computed. A very good agreement is obtained for the velocity  $\tilde{w}^0$ : the Eulerian deformation velocity field and the Lagrangian one continuously match at the body surface. Some difference are however visible for the  $\tilde{u}^0$  component. Due to the fish configuration, the velocity  $\tilde{u}^0$  can be considered to be the velocity that is tangential to the body. Since optimal transportation is irrotational by definition, it is not able to model any skin rotation. The tangential velocity can not be computed if no other constraints are taken into account, such as done for example in [18,19] using a rigid motion constraint or a divergence-free constraint.

These Lagrangian and Eulerian velocities are then used to simulate the self propelled swimming in the both cases. According to equation (3), the swimming velocities of the center of mass  $\mathbf{x}_G$  are  $\mathbf{u}_s^G(t) = \mathbf{u}_s(\mathbf{x}_G, t)$ . These self propelled velocities are presented in Fig. 11. As expected, the swimming velocity  $w_s^G$  shows good agreement between cases I and II. The velocity  $u_s^G$  is influenced by the differences observed for the deformation velocity  $\tilde{u}^0$  in Fig. 10. However, considering all the modeling hypothesis, the Eulerian forward swimming velocity correctly approaches the actual Lagrangian forward velocity from which it has been derived.

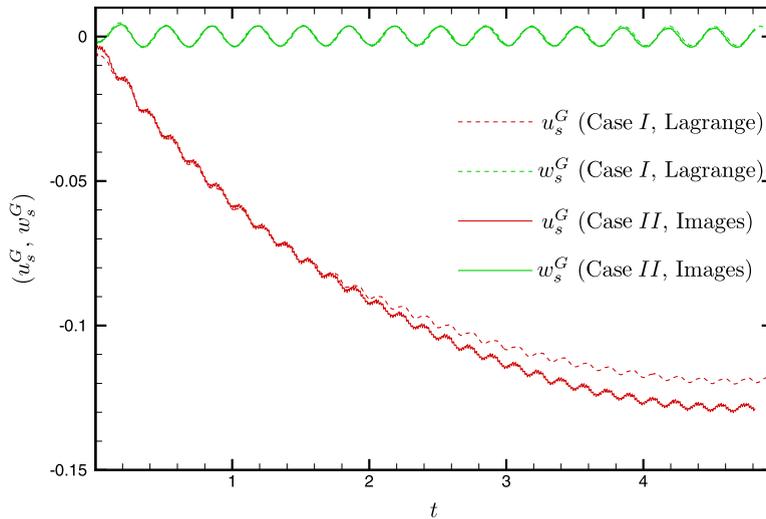


(a) Deformation velocity field ( $W = \tilde{w}^0$ ) in the  $z$ -direction, *i.e.* normal to the swimming direction



(b) Deformation velocity field ( $U = \tilde{u}^0$ ) in the  $x$ -direction, *i.e.* in the swimming direction

**Fig. 10.** Comparison of the deformation velocity fields obtained in cases *I* (“exact”, presented inside the fish) and *II* (approximated using optimal transportation, represented outside the fish).



**Fig. 11.** Comparison of the self propelled velocities obtained in cases *I* and *II*. The swimming velocity is  $u_s^G$  and the lateral velocity is  $w_s^G$ .

### 6.2. Mackerel fish swimming from pictures

We have applied the method described above to recover the geometry of a Mackerel, to deform its skeleton by a swimming law and to simulate swimming. Fig. 12 shows the wake flow generating by the fish built from pictures. We chose the same parameters as before for the swimming law, *i.e.* the fish length is 10 cm and the swimming frequency is  $f = 4$  Hz. The swimming velocity of the mackerel fish is slightly higher than the previous test cases ( $u_s^G \approx -0.15$  m/s). This difference can be explained by a different shape, and moreover by the lunate tail for the real fish.

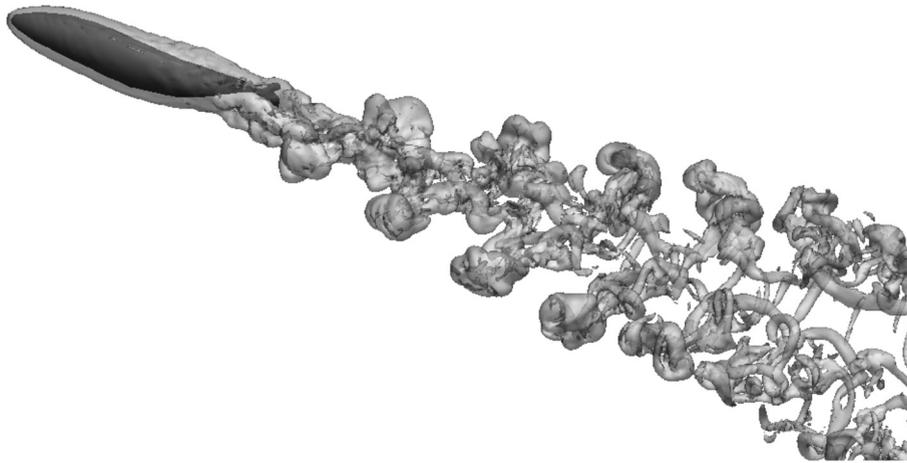


Fig. 12. Isovorticity representation of the wake generated by the mackerel fish starting from actual pictures.

## 7. Conclusions

In this work we have described a procedure to define a bioinspired geometry for a swimmer starting either from one image of a fish or from a sequence of images. This approach can be seen as a step closer in converting observed swimming pictures to a realistic numerical model, reducing arbitrary assumptions and time spent in post-processing images for data fitting to get body kinematics. Using this procedure it is possible to recover an Eulerian deformation velocity on the base of a limited number of modeling hypothesis, namely those at the base of  $L^2$  optimal transportation. We have shown an *in silico* example where we compare the swimming velocity of a swimmer whose geometry and deformation law is completely determined a priori, to that obtained by the identification method we propose. The results in terms of deformation velocity show that the irrotational hypothesis at the base of optimal transportation induces an error in the tangential velocity of the body. This error coupled to the discretization inaccuracies due to the fully Eulerian treatment of the geometry obtained from Lagrangian data reflects in an overestimation of the actual velocity of the swimmer of less than 10%. As an example of an actual application we have modeled a swimmer using pictures of a Mackerel. Future improvements of the results may be obtained by constrained optimal transportation.

## Acknowledgements

The authors thank C. Galusinski and C. Nguyen for contributions on fish construction. This study has been carried out with financial support from the French State, managed by the French National Research Agency (ANR) in the frame of the Investments for the future Programme IdEx Bordeaux-CPU (ANR-10-IDEX-03-02). The simulations presented in this paper were carried out using the PLAFRIM experimental parallel testbed, being developed under the Inria PLAFRIM development action with support from LABRI and IMB and other entities: Conseil Régional d'Aquitaine, FeDER, Université de Bordeaux and CNRS (see <https://plafirim.bordeaux.inria.fr/>).

## References

- [1] M. Triantafyllou, G. Triantafyllou, D. Yue, Hydrodynamics of fishlike swimming, *Annu. Rev. Fluid Mech.* 32 (2000) 33–53.
- [2] L. Wen, G. Lauder, Understanding undulatory locomotion in fishes using an inertia-compensated flapping foil robotic device, *Bioinspir. Biomim.* 8 (4) (2013) 046013.
- [3] R. Mittal, H. Dong, M. Bozkurttas, F. Najjar, A. Vargas, A. von Loebbecke, A versatile sharp interface immersed boundary method for incompressible flows with complex boundaries, *J. Comput. Phys.* 227 (10) (2008) 4825–4852.
- [4] A.A. Shirgaonkar, M.A. MacIver, N.A. Patankar, A new mathematical formulation and fast algorithm for fully resolved simulation of self-propulsion, *J. Comput. Phys.* 228 (7) (2009) 2366–2390.
- [5] O.M. Curet, I.K. AlAli, M.A. MacIver, N.A. Patankar, A versatile implicit iterative approach for fully resolved simulation of self-propulsion, *Comput. Methods Appl. Mech. Eng.* 199 (2010) 2417–2424.
- [6] S. Hieber, P. Koumoutsakos, An immersed boundary method for smoothed particle hydrodynamics of self-propelled swimmers, *J. Comput. Phys.* 227 (19) (2008) 8636–8654.
- [7] M. Gazzola, P. Chatelain, W.M. van Rees, P. Koumoutsakos, Simulations of single and multiple swimmers with non-divergence free deforming geometries, *J. Comput. Phys.* 230 (19) (2011) 7093–7114.
- [8] M. Bergmann, A. Iollo, R. Mittal, Effect of caudal fin flexibility on the propulsive efficiency of a fish-like swimmer, *Bioinspir. Biomim.* 9 (4) (2014) 046001.
- [9] A. Bhalla, R. Bale, B. Griffith, N. Patankar, A unified mathematical framework and an adaptive numerical method for fluid-structure interaction with rigid, deforming, and elastic bodies, *J. Comput. Phys.* 250 (2013) 446–476.
- [10] M. Bozkurttas, R. Mittal, H. Dong, Low-dimensional models and performance scaling of a highly deformable fish pectoral fin, *J. Fluid Mech.* 613 (2009) 311–342.

- [11] H. Dong, M. Bozkurttas, R. Mittal, P. Madden, G.V. Lauder, Computational modelling and analysis of the hydrodynamics of a highly deformable fish pectoral fin, *J. Fluid Mech.* 645 (2010) 345–373.
- [12] C. Galusinski, C. Nguyen, Skeleton and level set for channel construction and flow simulation, *Eng. Comput. (Lond.)* (2015) 289–303.
- [13] C. Villani, *Optimal Transport, Old and New*, 1st edition, Springer-Verlag, 2009.
- [14] G. Monge, *Memoire sur la theorie des deblais et des remblais*, *Histoire de l'Academie des Sciences de Paris*.
- [15] J.-D. Benamou, Y. Brenier, A computational fluid mechanics solution to the Monge–Kantorovich mass transfer problem, *Numer. Math.* 84 (2000) 375–393.
- [16] A. Iollo, D. Lombardi, A lagrangian scheme for the solution of the optimal mass transfer problem, *J. Comput. Phys.* 230 (2011) 3430–3442.
- [17] L.-P. Saumier, M. Agueh, B. Khouider, An efficient numerical algorithm for the l2 optimal transport problem with periodic densities, *IMA J. Appl. Math.*
- [18] D. Lombardi, E. Maitre, Eulerian models and algorithms for unbalanced optimal transport, *ESAIM: M2AN* 49 (6) (2015) 1717–1744.
- [19] R. Hug, E. Maitre, N. Papadakis, Multi-physics optimal transportation and image interpolation, *ESAIM: M2AN* 49 (6) (2015) 1671–1692.
- [20] A. Bouharguane, A. Iollo, L. Weynans, Numerical solution of the Monge–Kantorovich problem by density lift-up continuation, *ESAIM: M2AN* 49 (6) (2015) 1577–1592.
- [21] J.A. Sethian, *Level Set Methods and Fast Marching Methods*, Cambridge University Press, Cambridge, UK, 1999.
- [22] P. Angot, C. Bruneau, P. Fabrie, A penalization method to take into account obstacles in a incompressible flow, *Numer. Math.* 81 (4) (1999) 497–520.
- [23] R.N.V. Yen, D. Kolomenskiy, K. Schneider, Approximation of the Laplace and Stokes operators with Dirichlet boundary conditions through volume penalization: a spectral viewpoint, *Numer. Math.* 128 (2) (2014) 301–338.
- [24] A. Chorin, Numerical solution of the Navier–Stokes equations, *Math. Comput.* 22 (1968) 745–762.
- [25] R. Temam, Sur l'approximation de la solution des equations de Navier–Stokes par la methode des pas fractionnaires ii, *Arch. Ration. Mech. Anal.* 32 (1969) 377–385.
- [26] M. Bergmann, J. Hovnanian, A. Iollo, An accurate cartesian method for incompressible flows with moving boundaries, *Commun. Comput. Phys.* 15 (5) (2014) 1266–1290.
- [27] R. Glowinski, T.-W. Pan, T. Hesla, D. Joseph, J. Periaux, A fictitious domain approach for the direct simulation of incompressible fluid flow past moving rigid bodies: application to particulate flow, *J. Comp. Physiol.* 169 (2001) 363–426.
- [28] K. Taira, T. Colonius, The immersed boundary method: a projection approach, *J. Comput. Phys.* 225 (2) (2007) 2118–2137.
- [29] B. Kallemov, A. Bhalla, B. Griffith, A. Donev, An immersed boundary method for rigid bodies, *Commun. Appl. Math. Comput. Sci.* 11 (1) (2016) 79–141.
- [30] D. Barrett, M. Triantafyllou, D. Yue, M. Grosenbauch, M. Wolfgang, Drag reduction in fish-like locomotion, *J. Fluid Mech.* 392 (1999) 182–212.
- [31] M. Lighthill, Aquatic animal propulsion of high hydrodynamical efficiency, *J. Fluid Mech.* 44 (1970) 265–301.
- [32] J.J. Videler, F. Hess, Fast continuous swimming of two pelagic predators, saithe (*pollachius virens*) and mackerel (*scomber scombrus*): a kinematic analysis, *J. Exp. Biol.* 109 (1984) 209–228.
- [33] C.-L. Yu, S.-C. Ting, M.-K. Yeh, J.-T. Yang, Three-dimensional numerical simulation of hydrodynamic interactions between pectoral-fin vortices and body undulation in a swimming fish, *Phys. Fluids* 23 (9) (2011) 091901.
- [34] M. Bergmann, A. Iollo, Modeling and simulation of fish-like swimming, *J. Comput. Phys.* 230 (2) (2011) 329–348.
- [35] J. Hovnanian, *Methode de frontieres immergees pour la mecanique des fluides. Application a la simulation de la nage*, PhD thesis, Universite de Bordeaux 1, Bordeaux, France, 2012.
- [36] Y. Brenier, Polar factorization and monotone rearrangement of vector-valued functions, *Commun. Pure Appl. Math.* 64 (1991) 375–417.
- [37] C. Villani, *Topics in Optimal Transportation*, 1st edition, American Mathematical Society, 2003.
- [38] G. Loeper, F. Rapetti, Numerical solution of the Monge–Ampere equation by a Newton's algorithm, *C. R. Acad. Sci. Paris, Ser. I* 340 (2005) 319–324.
- [39] J. Sethian, A fast marching level set method for monotonically advancing fronts, *Appl. Math.* 93 (1996) 1591–1595.
- [40] M. Sussman, P. Smereka, S. Osher, A level set approach for computing solutions to incompressible two-phase flow, *J. Comput. Phys.* 114 (1) (1994) 146–159.
- [41] F. Luddens, M. Bergmann, L. Weynans, Enablers for high-order level set methods in fluid mechanics, *Int. J. Numer. Methods Fluids* 79 (2015) 654–675, <http://dx.doi.org/10.1002/flid.4070>.
- [42] F. Noca, *On the evaluation of time-dependent fluid-dynamic forces on bluff bodies*, PhD thesis, California Institute of Technology, 1997.
- [43] S. Kern, P. Koumoutsakos, Simulations of optimized anguilliform swimming, *J. Exp. Biol.* 209 (2006) 4841–4857.
- [44] J. Carling, T.L. Williams, G. Bowtell, Self-propelled anguilliform swimming: simultaneous solution of the two-dimensional Navier–Stokes equations and Newton's laws of motion, *J. Exp. Biol.*
- [45] A.P.S. Bhalla, B.E. Griffith, N.A. Patankar, A forced damped oscillation framework for undulatory swimming provides new insights into how propulsion arises in active and passive swimming, *PLoS Comput. Biol.* 9 (6) (2013).
- [46] A.P.S. Bhalla, R. Bale, B.E. Griffith, N.A. Patankar, Fully resolved immersed electrohydrodynamics for particle motion, electrolocation, and self-propulsion, *J. Comput. Phys.* 256 (2014) 88–108.